

DOCTOR OF PHILOSOPHY

Speciality: Fluid Mechanics



presented by

Martin Petitfrere

EOS based simulations of thermal and compositional flows in porous media

Directed by Dan V. NICHITA and Igor BOGDANOV

Defended on 12th September 2014 with a Jury:

M. Igor BOGDANOV	CHLOE	Co-Director
M. Jean-Luc DARIDON	UPPA	Examiner
M. Eric HENDRIKS	SHELL	Reporter
M. Jean-Noël JAUBERT	ENSIC Nancy	Reporter
M. Claude LEIBOVICI	CFL CONSULTANT	Examiner
M. François MONTEL	TOTAL	Examiner
M. Dan Vladimir NICHITA	CNRS/UPPA	Director
M. Denis VOSKOV	STANFORD University	Examiner

ACKNOWLEDGEMENTS

I would first thank all the jury members for accepting to evaluate my work.

Along this thesis, I have been working within different organisms; the LFCR (Laboratoire des Fluides Complexes et de leurs Rservoirs), within the UPPA (Universit de Pau et des Pays de lAdours), TOTAL S.A. and CHLOE (Centre dhuiles Lourdes ouvertes et exprimentales). I naturally acknowledge all those organisms which made this work possible.

In the LFCR, I would naturally thank D.V. Nichita, my PhD director. I arrived at the beginning with almost no knowledge on thermodynamics, Dan succeeded in teaching me the different aspects. He directed the thesis really well, implying himself in every chapter, giving good advices even outside thermodynamics, and pushing me when he had to. He oriented the work perfectly to work on feasible subjects and challenging works. Finally I would like to thank him for the conferences, the moments we spent on top of the hill, and all the fun we had during this period.

I would like to thank the members of the LFCR department: the professors, and especially Guillaume Galliero for his advices and Jean-Luc Daridon, in charge of the structure. I would like to thank all the students with whom I spent some really good time and finally a special thank to Catherine and Veronique for all the work I made them do.

Besides, I stayed in the CHLOE laboratory for 2 years and a half. I found a really nice team I really got along with. First, I would like to thank Arian Kamp, the former manager of CHLOE for creating, and organizing the beginning of the thesis. I would like to thank Igor for the technical advices along my thesis, for the chapter 5, in which he helped me a lot. I would like to thank Alain for the organization after the departure of Arian. A special thanks to Mareylise, Alfredo for there advices in heavy oils, for the fun we had and the cakes you made for us! Muchas gracias por el viernes de espanol. Finally I would like to thank Brigitte, Isabelle and Delphine for their sympathy and all for the administrative tasks they have been doing for me. Thank you Delphine who has always been there to open the door to someone who was always forgetting its badge.. A special thank to Sebastien and Clement for the mathematical discussions and for all the discussions and football we made! (Ps: thank you Clement for the latex/inkscape help !!). I will miss the football bets, even if i loose against interns (Mathieu...), working in CHLOE was a lot of fun.

Finally within Total, I have been working within three different departments. I would like to thank Xavier Britsch who supervised the whole thesis from the beginning. Always giving good ideas and directing me really well. My industrial supervisor was Franois Montel. He taught me a lot in thermodynamics, explaining me different thermodynamics aspects. He gave me really good ideas and a good supervision, leaving me a good freedom in my work. I have learned a lot from him in many different aspects. He seems to know everything and it was really nice working with him.

Finally I would like to thank the intersim for the nice atmosphere. I would like to thank Bernard Faissat and Gilles Darshes for accepting me within the department. I would like to acknowledge the whole team but more specifically Corentin Rossignon for its expertise in computer science, even if the valgrind faults were not mine.. And thank you for teaching me Inkscape (my favourite software now!) Leonardo Pattachini for all the advices in reservoir simulation and equilibrium flash calculations. He was here to guide the orientation of the thesis and I thank him. It was nice working around him.

Finally a really special thank to Alexandre Lapene. He was there to supervise my thesis as well. He gave me really good advices especially for the thermal part of the reservoirs. He is the one who initiated the collaboration with Stanford.

Last but not least, I would like to acknowledge Stanford University, for allowing me to come to Stanford

and to use AD-GPRS for the presented thesis. More particularly I would like to thank D.Voskov and R.Zaydullin for the exchanges and the collaboration that has been created. And for all the extra work (restaurants, meal at Deniss place, wine testing). Hope the collaboration will go on!

Finally, I would like to thank my family and my friends. My parents have always been there and I owe them a lot. I wouldn't have come to the Total interview without them. They've always helped me in the difficult situations. It was also really nice to spend these years close to my brothers and sisters. They are always there for me too, and it was nice to be around them. I would like to thank my best friends : Jeremy and Guillaume who supported me in the difficult times. Sophie, for the Reunion team building! The friends from the badminton who helped me evacuating pressure, stress (special thanks to my partner Julien, Orni and Alex, Pierrot, Elo, JB, Audrey, Romain). The friends from the UPPA who I spent a lot of time with: Angie, Pamela, Ariane, Georgia, the two Erics, Julien, Julek, Magalie. All the lunch and parties we had were really nice, and i hope we will keep on seeing each other!

ABSTRACT

EOS based simulations of thermal and compositional flows in porous media

Three to four phase equilibrium calculations are in the heart of tertiary recovery simulations. In gas injection, micro-emulsion flooding, steam-injections processes, additional phases emerging from the oil-gas system are added to the set and have a significant impact on the oil recovery. The most important computational effort in many chemical process simulators and in petroleum compositional reservoir simulations is required by phase equilibrium and thermodynamic properties calculations. In chemical process simulators, the high number of components to deal with makes the equilibrium calculations time-consuming; in reservoir simulations, the number of components is limited (typically to a dozen), but a huge number of phase equilibrium calculations is required in field scale simulations. Generally, pseudo-components are generated to decrease the dimensionality of the system leading to approximations of the original problem. For all these reasons, calculation algorithms must be robust and time-saving. In the literature, many simulators based on different equations of state (EoS) have been designed but few of them are applicable to thermal recovery processes such as steam injection. To the best of our knowledge, no fully compositional thermal simulation of the steam injection process has been proposed with extra-heavy oils; these simulations are essential and will offer improved tools for predictive studies of the heavy oil fields. Thus, in this thesis different algorithms of improved efficiency and robustness for multiphase equilibrium calculations are proposed, able to handle conditions encountered during the simulation of steam injection for heavy oil mixtures.

Most of the phase equilibrium calculations are based on the Newton method and use conventional independent variables. These algorithms are first investigated and different improvements are proposed. Michelsens (Fluid Phase Equilibria 9 (1982) 21-40) method for multiphase-split problems is modified in order to take full advantage of symmetry (in the construction of the Jacobian matrix and the resolution of the linear system). The reduction methods introduced by Michelsen (Ind. Eng. Chem. Process Des. Dev. 25 (1986) 184-188) and Hendriks (Ind. Eng. Chem. Res. 27 (1988) 1728-1732) enable to reduce the space of study from nc (the number of components) for conventional variables to M (with $M \ll nc$) and are already used in some commercial reservoir simulators. A reduction method based on the multi-linear expression of the logarithm of fugacity coefficients (Nichita and Graciaa, Fluid Phase Equilibria. 302 (2011) 226-233) is extended to phase stability analysis and multiphase-split calculations. Unlike previous reduction methods, the set of variables is unbounded and the convergence path is the same as in conventional methods using the logarithm of equilibrium constants as variables.

The Newton method requires a positive definite Hessian for convergence. Other kinds of minimization methods are investigated which overcome this constraint; the Quasi-Newton and Trust-region methods always guarantee a descent direction. These methods represent an interesting alternative since they can reach supra-linear steps even when the Hessian is non-positive definite, and can reach quadratic steps (Trust-Region) or nearly quadratic steps (Quasi-Newton) otherwise. A new set of independent variables is proposed (designed to ensure a better scaling of the problem) for a modified BFGS (which ensures the positive definiteness of the approximation of the Hessian matrix) algorithm and a Trust-Region method is also proposed for the stability-testing and phase-split problems.

Subsequently, by assuming the fluid composition as semi-continuous, a methodology based on a Gaussian quadrature is proposed to mathematically compute a set of pseudo-components capable of representing the fluid behavior. The methodology can be seen as a lumping-delumping procedure, applicable to any number of quadrature points and to any feed distribution (even in cases when no distribution function can model the feed composition, or several distribution functions are needed to model different portions of the mixture.).

In a last part, a general multiphase flash procedure implementing all the developed algorithms is presented, and tested against experimental and literature data. Three- and four phase CO₂ injection simulations demonstrate the capability of the program to handle any number of phases. Simulations of steam flooding are performed for highly heterogeneous reservoirs. Finally, a fully compositional simulation of the steam assisted gravity drainage (SAGD) process is realized. To the best of our knowledge, this is the first simulation of the kind for heavy oil mixtures.

Keywords : phase equilibrium calculations, multiphase flash, reduction method, trust-region, quasi-Newton, convergence, number of iteration, robustness, semi-continuous, Gaussian quadrature, characterization, thermodynamics, SAGD, compositional, thermal, simulation, heavy oil, steam, injection, steam flooding

Simulation compositionnelle thermique d'écoulements en milieux poreux, utilisant une équation d'état

Les calculs d'équilibres triphasiques et quadriphasiques sont au coeur des simulations de réservoirs impliquant des processus de récupérations tertiaires. Dans les procédés d'injection de gaz, de balayage par microémulsion et d'injection de vapeur, le système huile-gaz est enrichi d'une phase additionnelle qui joue un rôle important dans la récupération de l'huile en place. Les calculs d'équilibres de phases représentent la majeure partie des temps de calculs dans les simulateurs de procédés chimiques de par le grand nombre de composants impliqués, ainsi que dans les simulations de réservoir compositionnelles qui, contrairement aux simulations de procédés, ne requièrent qu'un nombre limité de composants (typiquement une dizaine), mais impliquent un nombre conséquent de calculs d'équilibre. En général, des pseudo-composants permettant d'approximer le comportement du fluide sont générés, afin de réduire la dimensionnalité du système. Pour toutes ces raisons, il est important de concevoir des algorithmes de calculs d'équilibre qui soient fiables, robustes et rapides. Dans la littérature de nombreux simulateurs de réservoirs basés sur des équations d'état ont été conçus, mais peu d'entre eux sont applicables aux procédés de récupération thermique tels que l'injection de vapeur. A notre connaissance, il n'existe pas de simulation thermique complètement compositionnelle du procédé d'injection de vapeur pour des cas d'applications aux huiles lourdes. Ces simulations apparaissent essentielles et pourraient offrir des outils améliorés pour aider la récupération améliorée de certains champs pétroliers. Finalement, dans cette thèse, des algorithmes robustes et efficaces de calculs des équilibres multiphasiques sont proposés permettant de surmonter les difficultés rencontrés durant les simulations d'injection de vapeur pour des huiles lourdes.

La plupart des algorithmes d'équilibre de phases sont basés sur la méthode de Newton et utilisent les variables conventionnelles comme variables indépendantes. Dans un premier temps, des améliorations de ces algorithmes sont proposées. Les variables réduites introduites par Michelsen (Ind. Eng. Chem. Process Des. Dev. 25 (1986) 184-188) et Hendriks (Ind. Eng. Chem. Res. 27 (1988) 1728-1732), permettent de réduire la dimensionnalité du système de nc (le nombre de composants) dans le cas des variables conventionnelles, à M (avec $M \ll nc$), et sont déjà utilisées dans certains simulateurs de réservoirs commerciaux. Une méthode de réduction basée sur l'expression multilinéaire du logarithme des coefficients de fugacités (Nichita and Graciaa, Fluid Phase Equilibria. 302 (2011) 226-233) est étendue à l'analyse de stabilité et aux calculs d'équilibres multiphasiques. A l'inverse des précédentes méthodes de réduction, les variables ne sont pas bornées et le chemin de convergence est le même que pour les méthodes conventionnelles utilisant le logarithme des constantes d'équilibres comme variables indépendantes.

La méthode de Newton nécessite une Hessienne définie positive pour pouvoir être utilisée. D'autres méthodes de minimisations sont testées qui permettent de s'affranchir de cette contrainte; les méthodes Quasi-Newton et Trust-Region garantissent une direction de descente à chaque itération. Ces méthodes présentent un grand intérêt puisqu'elles permettent de réaliser des pas supra-linéaires même lorsque la Hessienne n'est pas définie positive, et de réaliser des pas quadratiques (Trust-Region) ou proches de quadratiques (Quasi-Newton)

dans le cas contraire. Un nouveau vecteur de variables indépendantes est proposé (construit afin d'obtenir une meilleure mise échelle du problème) et utilisé au sein d'un algorithme BFGS modifié (qui, par construction, approxime la matrice Hessienne par une matrice définie positive). De même, une méthode de Trust-Region est développée pour les problèmes de tests de stabilités et d'équilibres multiphasiques.

Ensuite, considérant le fluide comme semi-continu, une méthodologie basée sur une procédure de quadrature Gaussienne est proposée pour calculer mathématiquement des pseudo-composants capables de représenter le comportement du fluide. La méthodologie peut être vue comme une procédure de groupement/dégroupement, applicable pour tout nombre de points de quadratures et toute composition du mélange (même dans les cas où aucune distribution ne peut modéliser la composition du mélange, où lorsque plusieurs distributions sont nécessaires afin de modéliser les différentes portions de mélange). Dans une dernière partie, un algorithme général pour le calcul des équilibres multiphasiques est présenté incluant tous les algorithmes développés. Cet algorithme est aussi testé et validé pour des données expérimentales et de la littérature. Des simulations triphasiques et quadriphasiques d'injection de CO_2 démontrent la capacité du programme à traiter un nombre arbitraire de phases. Des simulations de balayages par la vapeur sont réalisées pour des réservoirs montrant d'importantes hétérogénéités. Finalement, une simulation totalement compositionnelle du processus de Steam Assisted Gravity Drainage (SAGD) est réalisée. A notre connaissance, il s'agit de la première simulation de la sorte pour des cas d'applications d'huiles lourdes.

Mots clés: calculs d'équilibres, multiphasiques, méthode de réduction, trust-region, quasi-Newton, convergence, nombre d'itérations, robustesse, semi-continue, quadrature Gaussienne, caractérisation, thermodynamique, SAGD, compositionnelle, thermique, simulation, huile lourde, vapeur, injection, balayage par la vapeur

CONTENTS

Acknowledgements	i
Abstract	iii
Contents	vi
French description of the thesis	1
0 Introduction	1
0.1 Les huiles lourdes comme un moyen de répondre à la demande d'énergie croissante . .	1
0.2 Récupération assistée du pétrole	1
0.3 Amélioration de la thermodynamique et des simulateurs de réservoirs existant	3
0.3.1 Un besoin d'améliorer les simulateurs de réservoirs	3
0.3.2 Calculs d'équilibre multiphasiques	3
0.4 Plan de thèse	4
1 Chapitre 1: Thermodynamique fondamentale	6
2 Chapitre 2: La procédure globale de calculs d'équilibre multiphasiques	6
3 Chapitre 3: Amélioration des calculs d'équilibre	7
4 Chapitre 4: Amélioration de la caractérisation des huiles lourdes par la thermodynamique semi-continue	7
5 Chapitre 5 : Simulation de réservoir	9
6 Conclusions et Perspectives	11
6.1 Conclusions	11
6.2 Perspectives	13
Introduction	15
1 The heavy oils, a viable alternative to the increasing energy demand	15
2 Enhanced oil recovery	16
3 Improved reservoir simulation and thermodynamics	17
3.1 A need to improve the reservoir simulators	17
3.2 Multiphase equilibrium calculations	19
4 Overview of the thesis	20
1 Fundamental thermodynamics	22
1.1 Thermodynamics functions	22
1.1.1 Internal Energy	22
1.1.1.1 First Law	22
1.1.1.2 Second law	23
1.1.2 Gibbs free energy	24
1.1.3 Helmholtz free energy	25
1.1.4 Enthalpy	25
1.1.5 Residual energy	25
1.1.6 Maximization of the entropy	25
1.1.7 Fugacity and fugacity coefficients	26

1.1.8	Condition for equilibrium at constant pressure and temperature	28
1.2	Equation of State Calculations	29
1.2.1	Introduction	29
1.2.2	Cubic equations of state	29
1.2.2.1	Solving the cubic polynomial equation	30
1.2.3	Fugacity coefficient	31
1.2.4	Molar Gibbs free energy	31
2	Global multiphase flash procedure	32
2.1	Introduction	32
2.2	Stability Analysis	33
2.2.1	The stationary points	34
2.2.2	Initial guesses	35
2.2.3	General minimization algorithm	36
2.2.4	Bypass acceleration method	36
2.3	Multiphase-split calculations	37
2.3.1	Solving the Rachford-Rice equation	39
2.3.1.1	Two phase-split calculations	39
2.3.1.2	Multiphase equilibrium calculations	40
2.3.2	Phase-split calculations	42
2.4	General multiphase equilibrium algorithm	43
3	Improvements in equilibrium calculations	44
3.1	Presentation of conventional methods	45
3.1.1	Stability testing	45
3.1.1.1	Mole numbers as independent variables	45
3.1.1.2	$\ln Y_i$ as independent variables	46
3.1.1.3	α as independent variable	46
3.1.2	Multiphase flash calculations	47
3.1.2.1	Mole numbers as independent variables	48
3.1.2.2	Improved mole numbers as independent variables	49
3.1.2.3	$\ln K$ as independent variables	49
3.2	Reduction methods	51
3.2.1	Presentation of existing reduction based methods	52
3.2.1.1	Reduction parameters	52
3.2.1.2	Two phase-split calculations in the reduction approach	54
3.2.1.3	Phase stability in the reduction approach	55
3.2.1.4	Pseudo-reduces methods	55
3.2.2	A new reduction method for the stability analysis problem	56
3.2.2.1	Introduction	56
3.2.2.2	Proposed reduction method	57
3.2.2.3	Results	61
3.2.2.4	Discussion	64
3.2.2.5	Conclusion	66
3.2.3	A comparison of conventional and reduction approaches for phase equilibrium calculations	66
3.2.3.1	Introduction	66
3.2.3.2	Results	67
3.2.3.3	Discussion	77
3.2.3.4	Conclusion	79

3.2.4	A new reduction method for multiphase equilibrium calculations	80
3.2.4.1	Introduction	80
3.2.4.2	Reduction multiphase flash	81
3.2.4.3	Results	85
3.2.4.4	Discussion	90
3.2.4.5	Conclusions	92
3.3	Phase equilibrium calculations with quasi-Newton methods	93
3.3.1	Introduction	93
3.3.2	The BFGS quasi-Newton method	93
3.3.2.1	Newton	93
3.3.2.2	The BFGS update	94
3.3.2.3	The Line search procedure	94
3.3.2.4	Ammar and Renon BFGS implementation	96
3.3.3	Proposed method	98
3.3.3.1	Two-phase split calculations	98
3.3.3.2	Stability analysis	100
3.3.4	Results	100
3.3.4.1	Tests on different BFGS methods	101
3.3.4.2	Mixtures used in this study	101
3.3.4.3	Error and stopping criteria	101
3.3.4.4	Results	101
3.3.4.5	Discussion	107
3.3.4.6	Conclusions/Perspectives	108
3.4	Robust and efficient Trust-Region based stability analysis and multiphase flash calculations .	108
3.4.1	Introduction	108
3.4.2	Trust-Region methods	109
3.4.2.1	The Trust-Region subproblem	110
3.4.2.2	Solving the Trust-Region subproblem	110
3.4.2.3	Algorithm for the Trust-Region subproblem	114
3.4.2.4	The Trust-Region size	115
3.4.3	Results	117
3.4.3.1	Phase stability testing	118
3.4.3.2	Multiphase flash calculations	121
3.4.4	Conclusions	130
3.5	Multiphase flash, using $\ln K$ and phase mole fractions as independent variables	130
3.5.1	Introduction	130
3.5.2	New proposed method	130
3.5.3	Results	132
3.5.3.1	Computational time	133
3.5.3.2	Convergence behavior	134
3.5.3.3	Discussion	135
4	Improvements in the characterization of heavy oils : Semi-continuous thermodynamics	136
4.1	Introduction	136
4.2	A new distribution function	138
4.3	Gaussian quadrature	140
4.4	Semi-continuous description	141
4.5	Calculation procedure	144
4.5.1	Discrete equilibrium flash calculations	144

4.5.2	Reconstruction of the initial solution	144
4.5.3	Algorithm	145
4.6	Results	145
4.6.1	Two-phase equilibrium	147
4.6.2	Three-phase equilibrium hydrocarbon mixture-water	149
4.6.3	Three-phase equilibrium hydrocarbon mixture-carbon dioxide	151
4.6.4	Influence of the number of quadrature points on accuracy	152
4.7	Conclusion	154
5	Reservoir Simulation	156
5.1	Introduction	156
5.2	Flow of fluids through porous media	158
5.2.1	Mass conservation	159
5.2.1.1	Accumulation term	159
5.2.1.2	Advective term	159
5.2.1.3	Diffusive and dispersive term	159
5.2.1.4	Mass conservation equation	160
5.2.2	Quantity of movement	160
5.2.3	Equation of energy conservation	160
5.2.3.1	Accumulation	160
5.2.3.2	Heat conduction	161
5.2.3.3	Convective term in energy transfer	161
5.2.3.4	Energy conservation equation	161
5.3	Fluid properties	162
5.3.1	Phase viscosities	162
5.3.1.1	Light-Medium fluid model	162
5.3.1.2	Heavy oil	162
5.3.2	Mass densities	163
5.3.2.1	STARS liquid densities	163
5.3.2.2	Volume translated Peng-Robinson	163
5.3.3	Relative permeability	163
5.3.3.1	Two- and three-phase relative permeabilities	163
5.3.3.2	Four-phase relative permeabilities	164
5.3.4	Enthalpy	164
5.3.4.1	EoS based enthalpy of the oil and gas phases	164
5.3.4.2	Enthalpy of the water phase	165
5.4	Three-phase bypass	166
5.4.1	Introduction	166
5.4.2	Mathematical background	166
5.4.2.1	Working space	166
5.4.2.2	Continuity of a simplex	167
5.4.2.3	Gibbs energy analysis	168
5.4.3	Parameterization	168
5.4.3.1	Parameterization of the compositional space	168
5.4.3.2	Parameterization in temperatures and pressures	170
5.4.3.3	Projection in a tie-simplex space	170
5.4.4	Status identification	171
5.4.5	Pressure and Temperature parameterization and interpolation	171
5.5	Simulations	172

5.5.1	Two phase case: match with Eclipse (commercial reservoir simulator from Schlumberger)	172
5.5.1.1	Mixture/Reservoir properties	172
5.5.1.2	Simulation/Results	173
5.5.2	Full isothermal three phase compositional simulations: CO ₂ gas injection	173
5.5.2.1	Three-phases, CO ₂ injection	174
5.5.2.2	Four-phases, CO ₂ injection	177
5.5.3	Full thermal three-phase compositional simulations for steam injection	179
5.5.3.1	Light oil cases	179
5.5.3.2	Medium oil in a highly heterogeneous reservoir	181
5.5.3.3	Extra-heavy oil	185
5.6	Comparisons between different compositional acceleration procedures	189
5.7	Conclusion	190
6	Conclusions and Perspectives	192
1	Conclusions	192
2	Perspectives	194
	Appendix 1: Reduction variables	196
A	Elements of the Jacobian matrix/partial derivatives for the reduced stability analysis	196
B	Minimization of the modified TPD function in the reduction method	197
C	Equivalence of conventional and reduction stability methods	200
D	Reduction flash as a constrained minimization problem	201
E	Equivalence of conventional and reduction flash calculation methods	204
F	Link between reduction methods Q-red. and h-red. for phase stability testing	205
G	Pseudo-reduction methods for phase stability testing	206
H	Partial derivatives in the direct extension of Nichita and Gracia[2011]’s reduction method	207
I	Gradient vector in multiphase reduction	208
J	Elements of the Hessian matrix in multiphase reduction	209
	Appendix 2: Semi-continuous description of mixture composition	212
K	Computation of the alpha and beta	212
L	Calculation of quadrature nodes and weights	213
	List of Figures	215
	List of Tables	219
	References	220

FRENCH DESCRIPTION OF THE THESIS

0 Introduction

0.1 Les huiles lourdes comme un moyen de répondre à la demande d'énergie croissante

Les besoins en énergie des économies émergentes, et particulièrement en Asie (Chine, Inde) sont de plus en plus importants. Dans ce contexte, la production d'énergie s'accroît chaque année, et l'IEO2013 (International Energy Outlook 2013) projette que la consommation d'énergie mondiale augmentera de 56% entre 2010 et 2040. Les énergies fossiles représentent la plus grande source d'énergie, et leur production doit être augmentée pour satisfaire à la hausse en demande d'énergie.

Les huiles peuvent être classifiées en différentes catégories selon leur API gravity; qui est une mesure du poids de l'huile par rapport à celui de l'eau:

- Si elle est plus grande que 10, l'huile est moins dense que l'eau et flotte.
- Si elle est plus petite, l'huile coule;

Les huiles conventionnelles ont un degré API supérieur à 25. Due à leur faible densité et viscosité, elles peuvent généralement être récupérées facilement et leur coût de production reste assez faible. C'est pourquoi elles ont été exploitées jusqu'à présent. Cependant, les huiles conventionnelles deviennent de plus en plus rares. Leur production décroît de 5% par an et les réserves prouvées peuvent encore subvenir à 40 ans de productions en gardant le même débit.

Les huiles lourdes ont souvent un degré API se situant entre 5 et 22 (voir fig. 1). Due à leur importante viscosité, des méthodes d'EoR (Enhanced Oil Recovery) doivent être utilisées afin d'obtenir un rendement suffisant, ce qui les rend leur coût de production plus coûteuses que pour des huiles conventionnelles. Cependant, le prix du baril augmentant avec la demande, l'exploitation de champs d'huiles lourdes devient de plus en plus rentable et pourrait jouer un rôle important dans le futur. Avec un volume d'huile en place estimé entre 3000 et 4000 milliards de barils et des réserves potentielles autour de 500 000 milliards de barils, les huiles lourdes représentent près de 60% des réserves globales en huiles conventionnelles et représentent 20 à 25% des ressources de pétrole globales. Leur exploitation pourrait étendre les réserves d'énergies mondiales pour environ 15 ans. Différentes méthodes d'EOR existent pour permettre la récupération d'huiles lourdes.

0.2 Récupération assistée du pétrole

Le développement de méthodes d'EOR modernes est aujourd'hui vu comme un moyen d'étendre la production des réserves récupérables. Des estimations ont montré qu'une simple augmentation de 1% de la récupération d'huile pourrait augmenter les réserves d'huiles conventionnelles autour de 88 000 milliards de barils (3 fois la production actuelle).

De plus, non seulement les méthodes d'EOR permettent d'étendre la production d'huiles conventionnelles (qui étaient traditionnellement opérées par le biais de méthodes de récupérations primaires ou secondaires (fig. 2)), mais elles pourraient aussi être utilisées à la production d'huiles non-conventionnelles telles que les

huiles lourdes, qui ne peuvent être récupérées directement par simple pompage.

Différentes méthodes d'EOR existent (quelques-unes sont listées fig. 2), les principales méthodes sont:

- L'injection de gaz: en injectant du CO_2 ou du N_2 dans un réservoir, le gaz se dissout dans l'huile. La viscosité de l'huile diminue, ce qui rend l'huile mobile et plus simple à récupérer.
- Les méthodes chimiques
 - L'injection de surfactant peut créer de la microémulsion à l'interface entre l'huile et l'eau, ce qui réduit la tension interfaciale et mobilise l'huile résiduelle. Ce mécanisme permet entre autre de récupérer une partie de l'huile résiduelle localisée dans les pores.
 - L'injection de polymères est une amélioration du processus de récupération par injection d'eau. En co-injectant du polymère, la mobilité de l'huile est réduite ce qui crée un front plus large et qui permet une plus grande zone de balayage.
- Les méthodes thermiques représentent la plupart des projets d'huiles lourdes. Elles sont actuellement en production et joueront surement un rôle important dans le futur. En augmentant la température, l'huile est chauffée, ce qui réduit sa viscosité. Ce procédé augmente la mobilité (fig. 3) en réduisant la tension de surface et en augmentant la perméabilité. L'huile chauffée peut aussi se vaporiser et condenser pour créer une huile améliorée, plus facile à récupérer. Les méthodes thermiques les plus utilisées sont la combustion In-Situ, l'injection continue de fluides chauds tels que la vapeur, de l'eau ou des gaz ainsi que les méthodes cycliques. Au sein de ces méthodes, l'injection de vapeur représente la principale méthode de récupération thermique d'huiles.

Trois méthodes d'injection de vapeur existent principalement dans l'industrie:

- Avec deux puits verticaux (un producteur et un injecteur) séparés par une certaine distance. Ce processus fonctionne pour des huiles à viscosités moyennes (fig. 4a). Différentes régions peuvent être observée fig. 4b.
 - * Dans la zone de vapeur, près du puit injecteur, trois phases coexistent: le gaz, l'huile et l'eau. La température est assez uniforme, de même que la saturation en huile.
 - * Un peu plus loin, la température décroît, l'eau et les composants légers de l'huile condensent au contact de la matrice froide.
 - * Ensuite, l'huile est déplacée par l'eau (balayage par l'eau) dans une troisième zone.
 - * Enfin, loin du front d'injection, les conditions sont identiques à celle du réservoir initial.
- Avec un puit vertical qui joue à la fois le rôle de producteur et d'injecteur. Le procédé est appelé stimulation cyclique de vapeur (méthode Huff and Puff) et est assez efficace pour des huiles à hautes viscosités. Dans un premier temps, de la vapeur est injectée pour chauffer l'huile et réduire sa viscosité. Ensuite, l'huile est produite par flux naturelles et par pompage. Ces deux phases sont répétées alternativement.
- Enfin, le procédé SAGD (Steam Assisted Gravity Drainage) est très efficace pour récupérer les huiles lourdes avec une très grande viscosité. Le procédé existe déjà en production. Le procédé SAGD est représenté fig. 5a. La vapeur est injectée dans le puit injecteur (situé au-dessus du puit producteur). Avec la température, la viscosité de l'huile diminue et l'huile devient mobile. Par gravité, l'huile coule le long de la chambre de vapeur vers le puit producteur (fig. 5b).

0.3 Amélioration de la thermodynamique et des simulateurs de réservoirs existant

0.3.1 Un besoin d'améliorer les simulateurs de réservoirs

La simulation des procédés thermiques n'est pas simple. L'injection de vapeur crée des effets de composition tels que la distillation de vapeur, la condensation et la vaporisation qui sont essentielles à prendre en compte pour ce type de récupération. La région tri-phasique est au cur du procédé (eau, gaz, huile) et devrait être modélisé de façon précise.

La plupart des simulateurs de réservoirs compositionnels (ECLIPSE 300/ INTERSECT) traitent l'eau comme une phase pure et ne prennent pas en compte la solubilité de l'eau dans les hydrocarbures. Les calculs d'équilibres de phases sont effectués entre les phases huiles et gaz, avec l'eau traitée séparément, ce qui peut conduire à d'importantes approximations.

Il a été montré que la solubilité de l'eau dans l'huile pouvait être significative pour des hautes températures. Fig. 6 montre des données expérimentales de [McKetta and Katz [1948]] témoignant de la solubilité de l'eau dans la phase riche en hydrocarbures pour différentes pressions et températures. Les valeurs de solubilités changent la viscosité, la densité et d'autres propriétés, ce qui peut affecter les performances de production.

Dans le simulateur STARS (CMG), l'eau est traitée dans les calculs d'équilibres. Cependant, le simulateur est basé sur des modèles K-value (c'est aussi le cas pour ECLIPSE ou INTERSECT pour les modèles basés sur les K-value). L'approche K-value peut amener des erreurs importantes dans la simulation de procédés avec de grands changements de compositions, tels que les procédés thermiques. Réalisant les mêmes simulations pour le procédé d'injection de vapeur, [Varavei [2009]] a montré que les différences entre un modèle basé sur les K-value et un modèle basé sur une équation d'état pouvaient être très significatives.

Ces dernières années, des simulations tri-phasiques complètement compositionnelles du procédé de balayage par la vapeur ont été réalisées [Brantferger [1991]], Voskov et al. [2009], Varavei and Sepehrnoori [2009]] et plus récemment [Zaydullin et al. [2014]]. Cependant, aucune simulation complètement compositionnelle d'injection de vapeur sur des huiles extra-lourdes n'a encore été proposé dans la littérature. Ces simulations apparaissent essentielles et pourraient permettre d'aider à la prise de décision de l'exploitation d'un champ, par prédiction de la possible rentabilité.

Les calculs d'équilibres tri- et quadri-phasiques sont au coeur des procédés de récupération tertiaires. Pour les procédés d'injection de gaz, de microémulsion, d'injection de vapeur, une phase additionnelle au système gaz/huile apparaît, jouant un rôle important dans la récupération de l'huile. En chimie des procédés, le nombre important de composants rend les calculs d'équilibre prohibitifs. Généralement des pseudo-composants sont générés afin de diminuer la dimensionnalité du système, ce qui amène à des approximations du problème d'origine. Pour toutes ces raisons, des algorithmes rapides et robustes sont nécessaires.

0.3.2 Calculs d'équilibre multiphasiques

La résolution des calculs d'équilibres multiphasiques est basée sur la minimisation de l'énergie libre de Gibbs. Différents types d'algorithmes ont été proposés pour traiter ce problème. Une méthodologie a été développée [Michelsen [1982b] et Michelsen [1982a]] utilisant alternativement l'algorithme d'analyse de stabilité basé sur la méthode du plan tangent (*TPD*) et le calcul d'équilibre pour un nombre donné de phases.

Deux types d'algorithmes ont été développés dans la littérature.

- La première plus sûre, mais plus couteuse utilise est basé sur une procédure de minimisation globale. [Sun and Seider [1995]] a développé une méthode basée sur les intervalles-Newton et garantie de trouver le minimum global. [Stadtherr et al. [1995]] a développé une méthode basée sur une méthode homotopique. [Lucia et al. [2000]] a utilisé le calcul du plan tangent minimisé à l'aide d'une méthode SQP (Sequential Quadratic Programming), pour tester toutes les paires de composants. [Nichita et al. [2002]] ont développé une méthode de tunneling.
- Les méthodes de minimisation locales sont plus rapides, mais requièrent des initialisations spécifiques et multiples pour garantir une certaine probabilité d'obtenir le minimum global. Pour les calculs d'équilibres, [Michelsen [1982a]] proposa une procédure qui converge généralement vers le minimum global. Dans un contexte où les temps de calculs doivent être extrêmement restreints, cette méthodologie est aujourd'hui la plus utilisée en simulation de réservoir.

Les calculs d'équilibres multiphasiques ont été améliorés afin d'assurer la convergence dans des régions très difficiles (comme proches de singularités: points critiques pour les calculs d'équilibres, la limite du locus de stabilité pour l'analyse de stabilité). [Risnes et al. [1981]] fut le premier à proposer la méthode de substitution successives (SS) pour les calculs multiphasiques. Depuis de nombreux auteurs ont travaillé sur le sujet ([Nghiem and Heidemann [1982]], [Mehra et al. [1982]], [Michelsen [1994]]). [Michelsen [1982b]] fournit un ensemble de solutions initiales pour les calculs d'équilibres multiphasiques qui furent ensuite étendues par [Li and Firoozabadi [2012]] qui proposèrent une stratégie générale pour traiter des calculs multiphasiques pour 2 et 3 phases. Cependant des améliorations sont encore nécessaires, et particulièrement près des conditions difficiles où les algorithmes actuels ont des difficultés.

Le travail de recherche effectué au cours de cette thèse s'est concentré principalement sur l'amélioration des calculs d'équilibres de phases afin de pouvoir proposer des simulations complètement compositionnelles d'injection de vapeur avec des huiles extra-lourdes, sous des temps raisonnables.

0.4 Plan de thèse

Dans la simulation de réservoir, les équations de conservations doivent être résolues à chaque pas de temps. Des équilibres locaux sont considérés au sein de chaque cellule et un nombre important de calculs d'équilibres de phases est requis, basé sur la minimisation de l'énergie de Gibbs. Une erreur dans l'obtention du minimum peut ensuite être propagée, menant à des solutions non physiques. Il est donc impératif de développer des algorithmes efficaces et robustes. Dans cette thèse, des améliorations de calculs d'équilibre multiphasiques sont proposés afin de simuler le procédé d'injection de vapeur. Cette thèse est organisée en cinq parties.

Le premier chapitre présente rapidement les équations thermodynamiques à résoudre dans les calculs d'équilibre ainsi que le modèle d'équations cubiques utilisé pour ce travail. Les équations d'état cubiques fournissent une description raisonnable du comportement de phases pour les composants pures et les mélanges, ne nécessitant que les propriétés critiques et les facteurs acentriques de chaque composant. Ces modèles sont très utilisés dans la simulation de réservoir.

Dans un deuxième chapitre, l'algorithme de minimisation de l'énergie de Gibbs est présenté ([Michelsen [1982a]]) . Le test de stabilité et les calculs d'équilibre sont décrits de même que la procédure globale de minimisation.

Dans un troisième chapitre, des améliorations aux algorithmes de calculs d'équilibres sont présentées. La plupart des algorithmes sont basés sur la méthode de Newton-Raphson et utilisent les variables conventionnelles comme variables indépendantes. Dans un premier temps des améliorations directes de ces algorithmes sont proposées. Le logarithme des constantes d'équilibres ($\ln \mathbf{K}$) semble être le meilleur choix de variables

conventionnelles indépendantes pour les calculs d'équilibre. Dans ce travail, l'algorithme de [Michelsen [1982a]] est étendu aux problèmes multiphasiques, en écrivant la matrice Jacobienne comme le produit de deux matrices symétriques, le système linéaire peut être résolu par une factorisation de Cholesky afin de réduire le nombre d'opérations arithmétiques effectuées à chaque itération.

Les méthodes de réduction introduites par [Michelsen [1986] and Hendriks [1988]] permettent de réduire l'espace de travail de nc (le nombre de composants) to M (avec $M < nc$). Une extension de la méthode de réduction proposée par [Nichita and Graciaa [2011]], basée sur l'expression multilinéaire du logarithme des coefficients de fugacités est proposée pour l'analyse de stabilité et les calculs d'équilibres multiphasiques. De plus, des comparaisons entre différentes méthodes de Newton basées sur des variables conventionnelles et réduites sont effectuées pour l'analyse de stabilité et les calculs d'équilibre multiphasiques. Jusqu'à présent, les comparaisons observées dans la littérature concernaient le temps CPU global de minimisation de l'énergie de Gibbs. Dans ce travail, les comparaisons sont effectuées de façon indépendante pour chaque problème: stabilité, calculs d'équilibre diphasiques et multiphasiques. De plus le conditionnement ainsi que le chemin de convergence sont aussi analysés pour obtenir des comparaisons plus détaillées.

La méthode de Newton requiert une Hessienne définie positive pour être applicable. Dans une nouvelle section, d'autres types de méthodes de minimisations sont analysées qui peuvent surmonter ce problème. Les méthodes Quasi-Newton par exemple garantissent une direction de descente à chaque itération. Ces méthodes représentent une alternative intéressante car elles permettent de réaliser des pas proches de quadratiques sans avoir à calculer les dérivées secondes, et ne nécessitent pas de résoudre de système linéaire. Une approche BFGS (Broyden-Fletcher-Goldfarb-Shanno) appliquée avec un nouvel ensemble de variables indépendantes est ici proposée pour les problèmes d'équilibre diphasiques. De plus, en ajoutant un élément sur la diagonale de la matrice Hessian, les méthodes de Trust-Region permettent de réaliser des pas supra-linéaires jusqu'à quadratiques, même lorsque la Hessienne n'est pas définie positive. Une direction de descente est toujours garantie ce qui les rend des méthodes très attractives. Une procédure de Trust-Region est proposée dans ce même chapitre et appliquée à la fois à l'analyse de stabilité, et aux calculs d'équilibre multiphasiques.

La génération de pseudo-composants capables de représenter précisément le comportement du fluide est nécessaire afin de réaliser des simulations de récupérations améliorées. Dans un quatrième chapitre, un algorithme est proposé pour améliorer la caractérisation du fluide basé sur la thermodynamique semi-continue. En considérant la composition du fluide comme semi-continue, une nouvelle méthodologie basée sur une méthode de quadrature Gaussienne est proposée pour calculer mathématiquement un ensemble de pseudo-composants capables de représenter le comportement du fluide. La méthodologie peut être vue comme une méthode de lumping-delumping.

Au cours de cette thèse, un programme de calculs d'équilibre multiphasiques pouvant fonctionner sous un nombre arbitraire de phases a été développé. Ce programme, regroupant les algorithmes présentés dans le chapitre 3 a été couplé à divers simulateurs de réservoirs: TPP (simulateur de réservoir interne à Total S.A.) et AD-GPRS (Automatic Differentiation General Purposes Reservoir Simulator) développé par l'université de Stanford. Dans un dernier chapitre, le programme est testé par rapport des données expérimentales et de la littérature en stand-alone. Ensuite, différentes simulations sont réalisées pour simuler des méthodes de récupération améliorées. Une comparaison avec le simulateur commercial ECLIPSE (Shlumberger) est effectuée pour un cas tri-phasique de balayage par l'eau (avec l'eau traité comme pure). Ensuite des simulations complètement compositionnelles tri-phasiques et quadri-phasiques d'injection de CO₂ sont présentées révélant ainsi la capacité du programme de calculs d'équilibre développé, à traiter des cas avec un nombre arbitraire de phases.

Enfin, des simulations complètement compositionnelles d'injection de vapeur sont réalisées. Dans un premier temps, des simulations du balayage par la vapeur dans des réservoirs hétérogènes et homogènes sont effectuées. Ensuite, des simulations du procédé SAGD sur des huiles extra-lourdes sont présentées. Dans une

dernière partie, le programme de calculs d'équilibre est testé contre différentes techniques qui permettent de s'affranchir des calculs de stabilités dans le cadre de la simulation de réservoir. Les temps de calculs sont présentés pour les différents cas traités.

1 Chapitre 1: Thermodynamique fondamentale

La thermodynamique est au coeur des procédés de récupération thermiques. Dans ce chapitre, une description des différentes fonctions thermodynamiques utilisées dans cette thèse est développée.

Commençant par les premières et secondes lois, les expressions de l'énergie interne, des énergies libres de Gibbs et d'Helmholtz sont dérivées.

Les conditions d'équilibres sont aussi obtenues en recherchant le minimum de l'énergie libre de Gibbs.

Enfin, la correcte modélisation des différentes phases est un paramètre important dans la simulation de réservoir. De nombreuses équations d'état existent qui relient les différentes variables thermodynamiques pour une phase donnée. Dans cette thèse, nous proposons le développement d'algorithmes globaux pour résoudre les calculs d'équilibres multiphasiques. Les équations d'états cubiques représentent un moyen efficace pour modéliser les différentes propriétés thermodynamiques requises dans la simulation de réservoir (envelopes de phases, enthalpies, densités...) et permettent d'obtenir des résultats convenables pour des mélanges hydrocarbures. Dans un contexte où les temps de calculs sont importants, les équations cubiques offrent un bon compromis entre efficacité et précision. Ces équations d'états sont donc aujourd'hui au coeur de tous les simulateurs compositionnels, et en particulier sont celles utilisées dans cette thèse. Au cours de ce chapitre une description des équations d'état cubiques est effectuée.

2 Chapitre 2: La procédure globale de calculs d'équilibre multiphasiques

Un calcul d'équilibre $P - T$ consiste à calculer les fractions molaires de phases θ , ainsi que les compositions \mathbf{x} qui minimisent l'énergie libre de Gibbs pour une composition globale \mathbf{z} , une température T et une pression p (voir fig. 2.1).

Dans la simulation de réservoir, un nombre très important de calculs d'équilibres doivent être effectués. Les outils de minimisation globale ne peuvent être utilisés car les temps de calculs associés y sont très élevés. Ainsi les algorithmes de minimisation locale sont préférés. Une méthodologie proposée par [Michelsen [1982a]] est basée sur ce dernier type d'optimisation. Elle combine l'analyse de stabilité et les calculs d'équilibre et est aujourd'hui le standard utilisé dans l'industrie.

- L'analyse de stabilité [Michelsen [1982b]] permet de savoir si un mélange est stable pour un nombre donné de phases, ou s'il devrait se scinder en un plus grand nombre de phases.
- Les calculs d'équilibre calculent les compositions \mathbf{x} et les fractions molaires de phases θ qui minimisent l'énergie libre de Gibbs pour un nombre donné de phases.

Cette procédure mène généralement au minimum global et est particulièrement efficace. Elle est aujourd'hui implémentée dans la plupart des simulateurs de réservoir compositionnels et est utilisée dans ce travail. Dans ce chapitre, les deux problèmes seront présentés en détails. La méthodologie globale des calculs d'équilibre de phases pour minimiser l'énergie libre de Gibbs sera aussi présentée.

3 Chapitre 3: Amélioration des calculs d'équilibre

Les calculs d'équilibre multiphasiques (calculs d'équilibres, analyse de stabilité) jouent un rôle majeur dans la chimie des procédés et les réservoirs pétroliers. En effet, dans ces deux procédés, ils représentent la majeure partie du temps de calcul. De plus, la moindre erreur est susceptible d'affecter les résultats ou de mener à la non-convergence d'une simulation. Dans ce contexte, il devient essentiel de développer des algorithmes qui soient à la fois robustes et rapides, et d'utiliser des algorithmes adaptés pour chaque cas. En simulation de réservoir, le nombre de composants est très limité (souvent moins de douze) et un nombre important de calculs sont effectués. A l'inverse, en chimie des procédés, le nombre de composants peut être très important (de l'ordre de la centaine).

Dans ce chapitre, des améliorations relatives aux algorithmes de calculs d'équilibre sont proposées.

Dans un premier temps, un algorithme développé par [Michelsen [1982a]] permet d'utiliser une factorisation de Cholesky pour résoudre le système linéaire obtenu par la méthode de Newton, en utilisant le logarithme des constantes d'équilibres comme variables indépendantes. Cette méthodologie est ici étendue aux calculs d'équilibre multiphasiques et est aussi appliquée à la méthode proposée par [Haugen et al. [2011]], où les fractions molaires de phases et les constantes d'équilibres sont utilisés comme variables indépendantes.

De plus, les méthodes de réductions permettent de réduire l'espace de travail de nc (le nombre de composants) à M (avec $M < nc$). Une nouvelle méthode est ici proposée à la fois pour l'analyse de stabilité et pour les calculs d'équilibre multiphasiques, basée sur l'expression multilinéaire du logarithme des coefficients de fugacité. Ensuite, des comparaisons de différentes méthodes de Newton basées sur différentes variables conventionnelles et réduites sont présentées. Les tests portent à la fois sur les temps de calculs, mais aussi sur le conditionnement et le chemin de convergence. De plus, chaque problème est traité de façon indépendante (stabilité, calculs d'équilibres diphasiques et multiphasiques).

Dans un souci de développer un programme qui se veut modulaire et général, des algorithmes indépendants de l'équation d'Etat sont aussi testés. Une méthode BFGS (Broyden-Fletcher-Goldfarb-Shanno) est appliquée pour les calculs d'équilibre diphasiques avec un nouveau vecteur de variables indépendantes. De plus, une méthode de Trust-Region est appliquée à l'analyse de stabilité et les calculs d'équilibre multiphasiques.

4 Chapitre 4: Amélioration de la caractérisation des huiles lourdes par la thermodynamique semi-continue

Différents mélanges d'intérêts dans l'industrie, tels que les mélanges d'hydrocarbures contiennent un nombre très important de composants. Parce qu (i) un grand nombre de composants peut rendre les calculs d'équilibres coûteux (par exemple dans la simulation de réservoir) et (ii) il est impossible d'identifier tous les composants par des analyses chimiques standards (les fractions lourdes sont les plus difficiles à caractériser), des pseudo-composants (obtenus en regroupant différents composants individuels) sont utilisés afin de diminuer la dimensionnalité du problème de calculs d'équilibres.

Généralement, un mélange est lumpé en pseudo-composants en utilisant des critères de proximité pour regrouper les composants [Montel and Gouel [1984], Newley and Merrill [1991], Lin et al. [2008]]. Les propriétés critiques (température et pression critiques, facteurs acentriques) et les paramètres d'interactions des particules (BIPs) sont calculés en moyennant les propriétés pour chaque pseudo-composant.

Une alternative élégante aux méthodes classiques de lumping est l'utilisation de la thermodynamique semi-continue, qui est basée sur une approximation de la composition du mélange par une distribution continue. En thermodynamique semi-continue, les composants individuels sont traités de façon discrète (en

général, les fractions légères d'hydrocarbures et les composants non-hydrocarbures: CO₂, N₂, H₂S, H₂O, etc), alors que les composants restants sont traités de façon continue. Les principes de thermodynamique continue et semi-continue furent en premier temps développés par [Ratzsch and Kehlen [1983] et par Cotterman and Prausnitz [1985]] respectivement. Après les années 1980, différents auteurs travaillèrent à développer des algorithmes de calculs d'équilibres basés sur ces types de thermodynamique: [Cotterman et al. [1985], Behrens and Sandler [1986], Shibata et al. [1986], Willman and A.S. [1986], Willman and A.S. [1987a], Willman and A.S. [1987b], Ratzsch et al. [1988]]. Ensuite la thermodynamique semi-continue a été appliquée à une variété de calculs d'équilibres: les calculs d'équilibres sous différentes spécifications [Chou and J.M. [1986]], l'analyse de stabilité en utilisant la méthode du plan tangent [Browarzik et al. [1998], Monteagudo et al. [2001b]], les calculs de points critiques [Rochocz et al. [1997]], les gradients de compositions [Lira-Galeana et al. [1994], Esposito et al. [2000]], les équilibres liquides-solides [Labadie and Luks [2003]], la précipitation d'asphaltènes [Monteagudo et al. [2001a]], les calculs d'équilibre liquides-vapeurs en utilisant des équations d'état avec des contributions de groupes [Baer et al. [1997]], etc.

Les méthodes les plus utilisées se basent sur une quadrature généralisée de Gauss-Laguerre pour convertir la concentration molaire (distribution continue) en distribution discrète. Plus récemment, des méthodes ont été présentées utilisant des méthodes plus spécifiques, telle que la quadrature générale de Gauss-Stieltjes qui permet de calculer des points et poids de quadratures pour n'importe quelle distribution [Nichita et al. [2001]], des polynômes orthogonaux [Liu and Wong [1997]], ou la méthode de quadrature basée sur les moments [Lage [2007]].

Cependant, la plupart des approches présentées dans la littérature sont basées sur des distributions standards. Si la composition globale du mélange ne peut pas être modélisée par une distribution standard, ou si elle est irrégulière, la plupart des méthodes ne fonctionnent pas correctement. La méthode semi-continue basée sur la méthode des moments développée par [Lage [2007]], utilise la composition globale du mélange comme fonction poids et fonctionne avec n'importe quelle composition. Cependant, dans la formulation de la méthode QMoM, la quadrature est résolue en utilisant un algorithme de Gordon PDA (Product-Difference Algorithm) [Gordon [1968]], qui ne fonctionne de façon précise que pour un nombre restreint de points de quadratures. [John and Thein [2012]] a comparé les performances de la méthode QMoM en utilisant le PDA avec une méthode LQMDA (long quotient-modified difference algorithm) [Sack and Donovan [1972]] et l'algorithme de Golub-Welsch [Golub and Welsch [1969]]. Ils ont montré que dans certaines situations, la procédure de PDA échouait à calculer les points de quadrature (à partir de 8 points de quadratures dans leurs exemples) alors que les deux autres méthodes testées fonctionnaient correctement. [Gautschi [2004]] a montré que le problème est mal conditionné et que le nombre de conditionnement grandissait exponentiellement avec le nombre de points de quadratures.

Dans ce travail, la méthode QMoM est appliquée aux calculs d'équilibres multiphasiques pour des mélanges réels en utilisant une équation d'Etat cubique avec des BIPs non-nuls. Le calcul de la quadrature se base sur la procédure proposée par [Gautschi [1994]] (ORTHOPOL), qui permet d'éviter le mauvais conditionnement (intrinsèque au problème) et qui est adapté pour tous les nombres de points de quadratures (à l'inverse de la méthode QMoM couplée avec l'algorithme PDA). Dans certaines applications, il est important d'utiliser un nombre de pseudo-composants supérieur à sept (ce qui semble être la limite pour l'algorithme PDA).

Le chapitre se structure comme suit: dans un premier temps, une nouvelle distribution est introduite; après un bref rappel sur les quadratures gaussiennes, la description du fluide semi-continue est effectuée. Ensuite, l'algorithme général est présenté, pour enfin montrer des résultats obtenus pour des calculs d'équilibres diphasiques et tri-phases sur une huile lourde mélangée à du dioxyde de carbone et de l'eau. Les détails du calcul de la quadrature sont donnés en appendices.

5 Chapitre 5 : Simulation de réservoir

L'un des objectifs de cette thèse est de réaliser des simulations complètement compositionnelles du procédé de balayage par la vapeur. Les premiers simulateurs de réservoir traitant de ce processus ont été conçus dans les années soixante (Spillette and Nielsen [1968], Shutler [1969], Shutler [1970]). Les premiers modèles étaient basés sur un modèle linéaire tri-phasique [Shutler [1969]], puis étendue à deux dimensions [Shutler [1970]]. Plus tard [Vinsome [1974]], introduisit une méthode IMPES pour simuler le processus de balayage par la vapeur. [Coats [1976]] proposa le premier simulateur modélisant à la fois la partie thermique et la composition. [Coats [1978]] présenta une extension de [Coats [1976]], développant un simulateur compositionnel en trois dimensions pour simuler l'injection de vapeur. Plus tard, [Ishimoto [1985]] proposèrent de calculer les propriétés des hydrocarbures et de l'huile au moyen d'une équation d'état. La loi de Raoult était utilisée pour calculer la solubilité de l'eau dans l'huile. Puis, [citeChein] développa un simulateur compositionnel général qui pouvait traiter avec des options thermiques. Le simulateur était basé sur deux modèles compositionnelles: l'un basé sur une EOS, l'autre sur un modèle K-value. Une fois de plus, la phase aqueuse était supposée idéale.

Plus récemment, [Cicek and Ertekin [1996]] et plus tard Cicek [2005]] développèrent un simulateur compositionnel multiphasique et Fully Implicit pour simuler les problèmes d'injection de vapeur. Dans leur formulation, l'eau est traitée au sein des calculs d'équilibre de phases. Le simulateur est basé sur un modèle K-value. Généralement les tables de K-value sont fonctions d'un seul composant huile, la pression et la température. L'approche K-value ne permet pas d'obtenir les solubilités des composés hydrocarbures dans l'eau et la solubilité de l'eau dans la phase hydrocarbure de façon précise.

Cependant, il a été observé (fig. 6) que pour des hautes températures, la solubilité du composant eau dans la phase riche en hydrocarbure n'était pas négligeable. Les simulateurs commerciaux actuels font des approximations pour simuler les problèmes d'injection de vapeur. Ces approximations sont même plus importantes dans le cadre d'huiles lourdes. ECLIPSE 500 et INTERSECT par exemple, sont des simulateurs thermiques fonctionnant avec des modèles K-value, ou (pour INTERSECT) qui traitent l'eau séparément des calculs d'équilibres, ce qui conduit à des approximations.

Pour ces raisons, des auteurs ont commencé à développer des simulateurs basés sur des équations d'états pour réaliser des simulations compositionnelles thermiques du procédé d'injection de vapeur. En utilisant le fait que la solubilité des hydrocarbures dans l'eau est négligeable pour une certaine gamme de pressions-températures, des modèles free-water de calculs d'équilibres ont été développés [Luo and Barrufet [2005]], et plus récemment appliqués pour les problèmes d'huiles lourdes [Heidari [2014]]. La même méthodologie a été appliquée pour traiter de l'upgrading In-Situ [Lapene [2010]].

Pour des problèmes d'injection de CO₂ froid, les simulateurs de réservoirs ont été étendus pour intégrer des calculs d'équilibres multiphasiques (quatre phases) [Varavei and Sepehrnoori [2009] et Okuno [2009]]. Dans leur cas, ils n'ont pas inclus l'eau dans les calculs d'équilibre.

Cependant, l'eau issue des réservoirs est généralement plus proche d'une eau salée que d'une eau pure et l'hypothèse free-water n'est donc pas toujours valide. [Brantferger [1991]] développa un simulateur basé sur une équation d'état pour calculer les propriétés thermodynamiques de chaque phase (même la phase eau). Ils proposèrent de traiter le problème avec un flash isenthalpique, choisissant l'enthalpie comme variable primaire au lieu d'utiliser la température. [Voskov et al. [2009]] développèrent un simulateur général y intégrant une méthode d'accélération aux calculs de stabilités, en paramétrant l'espaces des tie-lines. [Varavei and Sepehrnoori [2009]] et plus récemment [Zaydullin et al. [2014]] proposèrent des simulations tri-phasiques d'injection de vapeur.

Enfin [Feizabadi [2013]] étendit les simulations compositionnelles pour simuler des procédés d'injection de solvants quadri-phasiques basés sur une équation d'état.

Il semblerait qu'il n'existe pas de cas de simulations complètement compositionnelles basées sur une équation d'état pour simuler le procédé d'injection de vapeur sur des huiles lourdes, en traitant l'eau de façon complète (au sein du calcul d'équilibre, sans hypothèse simplificatrice). Ces simulations apparaissent essentielles et pourraient permettre d'aider à la prise de décision de l'exploitation d'un champ, par prédiction de la possible rentabilité.

Dans ce chapitre, une description des différentes équations utilisées en simulation de réservoirs sont montrées, ainsi que les différents modèles pour calculer les propriétés du fluide.

Dans le chapitre 3, de nouveaux algorithmes pour résoudre les calculs d'équilibres ont été présentés. Un programme (Mflash) a été développé combinant ces méthodes et un programme de calculs d'équilibres multiphasiques, pouvant traiter un nombre arbitraire de phases a été développé.

Dans un premier temps, des tests en stand-alone sont proposés pour comparer les résultats obtenus avec Mflash, avec des données expérimentales et de la littérature, pour ensuite montrer des résultats au sein d'un simulateur de réservoir. Au cours de cette thèse, Mflash a été intégré à deux simulateurs de réservoirs: TPP (simulateur interne à Total S.A.) et AD-GPRS simulateur de l'université de Stanford [Younis and Aziz [2007], Voskov et al. [2009], Zhou et al. [2011]].

Différents procédés sont simulés dans ce chapitre. Premièrement, une simulation diphasique d'injection de vapeur, où l'eau est traitée indépendamment du calcul d'équilibre. Une comparaison avec le simulateur commercial ECLIPSE est aussi présentée.

Ensuite, des simulations complètement compositionnelles, tri-phasiques et quadri-phasiques d'injection de CO_2 sont montrées. Dans différentes simulations, il a été remarqué que la co-injection de solvant avec de la vapeur pouvait être à l'origine d'une nouvelle phase riche en solvant. L'importance de cette phase dans la récupération de l'huile n'a pas été analysée, mais dans un but de développer un programme thermodynamique robuste pour simuler le procédé SAGD, une simulation quatre phase a été réalisée et est présentée dans ce chapitre pour un cas d'injection isothermique.

Enfin, des simulations complètement compositionnelles du procédé de balayage par la vapeur sont réalisées pour des simulateurs hétérogènes et homogènes. Enfin, des simulations complètement compositionnelles du procédé SAGD pour des huiles lourdes est présentée.

Dans une dernière partie, différentes méthodologies pour améliorer les temps de calculs d'équilibre dans les simulateurs de réservoirs sont proposés pour tous les cas présentés. Les tests portent sur les variables réduites utilisées pour l'analyse de stabilité, la procédure de bypass [Rasmussen et al. [2006]] qui évite certains calculs de stabilités lorsqu'une cellule est dans l'état monophasique, la procédure développée par [Voskov and Tchelepi [2008], Voskov and Tchelepi [2009a], Voskov and Tchelepi [2009b], Iranshahr et al. [2010a]] et plus récemment [Zaydullin et al. [2013]] où l'espace d'étude est paramétré, pour réutiliser l'information de précédents calculs thermodynamiques.

6 Conclusions et Perspectives

6.1 Conclusions

Dans cette thèse, l'amélioration des algorithmes d'équilibre a rendu possible la réalisation de simulations complètement compositionnelles d'injection de vapeur avec des huiles lourdes.

La plupart des calculs d'équilibre sont basés sur la méthode de Newton-Raphson et utilisent les variables conventionnelles comme variables indépendantes. Des extensions de ces méthodes sont ici proposées. Le logarithme des constantes d'équilibre ($\ln K$) est le meilleur choix de variables indépendantes conventionnelles pour les calculs d'équilibre multiphasiques. Ces variables mènent vers des systèmes les mieux conditionnés et les méthodes de Newton basée sur ($\ln K$) convergent généralement avec moins d'itérations que pour les autres variables. Dans ce travail, la méthode de [Michelsen [1982a]] est étendue aux problèmes multiphasiques. La méthode permet d'écrire la matrice Jacobienne comme produit de deux matrices symétriques et de résoudre le système linéaire par une factorisation de Cholesky. Cette méthode a prouvé qu'elle pouvait réduire le temps de calculs d'environ 30% dans la plupart des cas.

Initiées par [Michelsen [1986] et Hendriks [1988]], les méthodes de réductions ont été développées et ont prouvé être très efficaces. Dans ce travail, une nouvelle méthode de réduction appliquée au test de stabilité et les calculs d'équilibre a été présentée. Elle est basée sur l'expression multilinéaire du logarithme des coefficients de fugacité comme fonctions des coefficients \mathbf{f} (variables indépendantes non-bornées). La dimensionnalité du problème ne dépend uniquement que du nombre de composants ayant des BIPs différents de zéro avec les autres, et non pas du nombre de composants du mélange, comme pour les procédures conventionnelles. La méthode proposée a un rayon de convergence plus large et l'algorithme de Newton peut être utilisée directement pour l'analyse de stabilité (sans besoin de substitutions successives). Elle a le même chemin de convergence que la procédure conventionnelle utilisant le logarithme des constantes d'équilibres comme variables indépendantes, et a montré être la meilleure méthode de Newton pour l'analyse de stabilité (en terme d'efficacité et de nombre de conditionnement).

Des comparaisons entre différentes méthodes conventionnelles et réduites ont été effectuées pour les problèmes d'analyse de stabilité et des calculs d'équilibres multiphasiques. Jusqu'à présent, les comparaisons observées dans la littérature ne portaient que sur le temps de calcul global pour calculer l'ensemble du problème de minimisation de l'énergie de Gibbs. Dans ce travail, les comparaisons ont été effectuées indépendamment pour chaque problème (stabilité, flash diphasiques et multiphasiques). De plus, le nombre de conditionnement ainsi que le chemin de convergence sont aussi analysées, amenant à des conclusions plus précises:

- Pour le problème de stabilité, la procédure de réduction proposée est la plus efficace est préférable pour des applications en simulation de réservoir.
- En revanche, pour les calculs d'équilibres multiphasiques, les méthodes basées sur les variables conventionnelles sont plus adaptés pour la simulation de réservoir. Dans le cadre de la chimie des procédés impliquant un grand nombre de composants, la méthode proposée est largement supérieure comparé aux autres méthodes.

Ensuite, une procédure de Trust-Region (TR) a été proposée pour traiter des problèmes d'analyse de stabilité et des calculs d'équilibre multiphasiques. TR peut être vu comme une 'extension' d'une méthode de Newton. Elle permet de réaliser des pas supra-linéaires lorsque la Hessienne n'est pas définie positive et des pas quadratiques sinon, en ajoutant un élément diagonal à la matrice. Une méthode hybride a été proposée. Au lieu de réaliser des itérations TR directement après l'utilisation d'une méthode de premier ordre (SSI), des itérations de Newton sont testées dans un premier temps et la méthode TR n'est utilisée uniquement que si la fonction objective augmente entre deux itérations successives. La méthode hybride a permis de décroître

les temps de calculs de façon remarquables, tout en gardant la robustesse intrinsèque de la procédure TR. Les tests ont été réalisés sur un ensemble complet de régions multiphasiques et convergence a été obtenue même pour des conditions très difficiles, telles que proche de points bi-critiques.

De plus, une nouvelle méthodologie a été développée pour trouver des variables permettant d'obtenir une matrice Hessienne avec une bonne mise à l'échelle. Cette procédure mène à la variable α introduite par [Michelsen [1982b]] pour l'analyse de stabilité et vers une nouvelle variable pour les calculs d'équilibres. Une méthode quasi-Newton avec un retour vers SSI (lorsque l'énergie de Gibbs s'accroît entre deux itérations) a été développée utilisant cette nouvelle variable et a montré que la convergence était possible, même très proche des frontières de phase, où les algorithmes rencontrent souvent des problèmes de convergence. L'algorithme a montré être plus efficace (en terme d'itérations et de nombre d'évaluations de fonctions) que le code de Nocedal LBFGS-B et d'autres méthodes BFGS proposées dans la littérature. Par rapport à l'algorithme de Newton, quelques itérations supplémentaires sont nécessaires pour obtenir la convergence, ce qui la rend très attractive. Cette méthode est très adaptée pour des modèles d'EoS plus complexes.

En combinant tous les algorithmes présentés, un programme général de calculs d'équilibre a été développé (Mflash). Il a été construit afin de pouvoir traiter un nombre arbitraire de phases. Les résultats obtenus avec ce programme ont montré être en accord avec des données provenant de la littérature et des données expérimentales, validant l'approche proposée.

En plus d'améliorer la robustesse et l'efficacité des calculs d'équilibre, une nouvelle méthodologie pour générer les pseudo-composants donnant une représentation précise du comportement du fluide a été développée. Elle est basée sur la méthode de quadrature des moments (QMoM) qui est ici appliquée aux calculs d'équilibre multiphasiques pour des huiles réelles (combinées avec des composants non-hydrocarbures tels que l'eau, le dioxyde de carbone), en utilisant une équation d'état cubique avec des BIPs différents de zéro. La quadrature est résolue à l'aide d'une procédure (ORTHOPOL) qui permet d'éviter les problèmes dus au mauvais conditionnement du problème. Cette méthode est adaptée pour n'importe quel nombre de points de quadratures (à l'inverse de précédentes formulations).

QMoM permet de relier la composition initiale détaillée et discrète, comportant un grand nombre de composants à un petit nombre de composants qui approximent la portion continue du fluide. Elle peut être vue en un certain sens comme une méthode de lumping. Elle peut être appliquée même dans les cas où aucune distribution standard ne peut modéliser la composition globale, ou lorsque plusieurs distributions sont nécessaires afin de modéliser différentes portions du fluide. Dans ce travail, il a été montré que cette méthodologie couplée à une méthode de delumping permettait de représenter très précisément le comportement de fluides réels.

La méthode QMoM dans notre implémentation a été testée pour différentes compositions initiales de mélanges hydrocarbures pour des calculs d'équilibres diphasiques, et en présence d'eau et de dioxyde de carbone pour des calculs tri-phasiques. Dans tous les cas, les frontières de phase, les compositions et fractions molaires de chaque phase étaient bien respectées, en utilisant la procédure semi-continue présentée (permettant de calculer une dizaine de pseudo-composants). Dans le cadre de la simulation de réservoir, cet outil apparaît essentiel car le temps de calcul pour chaque pas de temps est proportionnel à la puissance trois du nombre d'équations à résoudre [Burger et al. [1985]].

Ensuite, le programme de calculs d'équilibre a été implémenté au sein de deux simulateurs de réservoirs: TPP (développé en interne à Total S.A.) et AD-GPRS (simulateur de Stanford).

Pour des problèmes d'injection de vapeur tri-phasiques, les mêmes courbes de productions pour les différentes phases ont été obtenues en utilisant le simulateur ECLIPSE et en utilisant le programme (Mflash) intégré dans TPP. Ceci témoigne de la capacité de Mflash à converger vers la solution correspondant au

minimum d'énergie. De plus, des cas d'injection de CO_2 ont été présentées (connus pour être difficiles). Des simulations compositionnelles isothermiques, tri-phasiques et quadri-phasiques ont été réalisées jusqu'à la percée du CO_2 . Ces simulations montrent la capacité du code de calcul à traiter plus de trois phases, ce qui peut être très difficile d aux infimes différences d'énergies qu'il peut y avoir entre un état à trois phases et un état à quatre phases.

Des simulations d'injection de vapeurs ont aussi été réalisées. Le réservoir SPE10 (connu pour être très hétérogène) a été utilisé pour tester des simulations en trois dimensions de balayage par la vapeur jusqu'à la percée d'eau. Dans ces conditions (3D, hétérogène), un nombre important de chemins thermodynamiques sont accédés durant la simulation. Encore une fois, ces exemples témoignent de la robustesse de l'algorithme.

Enfin, des simulations complètement compositionnelles du procédé SAGD ont été réalisées avec des huiles extra-lourdes. A notre connaissance, il n'existe pas de simulation de la sorte dans la littérature (avec une modélisation complète de l'eau au sein des calculs d'équilibres). Les simulations thermiques sont très complexes car on y observe d'importants changements de conditions (composition, température, pression). De plus, la gravité joue aussi un rôle sur la composition.

Le développement du code a aussi été associé avec différentes méthodes d'accélération telles que la méthode Three-Phase-Bypass présentée par [Zaydullin et al. [2013]]. Des simulations complètement compositionnelles, thermiques tri-phasiques ont pu être réalisées pour simuler le procédé de balayage par la vapeur. Les calculs d'équilibres représentaient moins de 10% du temps de simulation total à l'aide de cette méthode. L'un des objectifs de la thèse était de réaliser des simulations complètement compositionnelles du problème d'injection de vapeur avec un pourcentage des calculs d'équilibres inférieurs à 50% de la simulation totale. Cet objectif est donc atteint.

Au travers de cette thèse, trois papiers ont été publiés [Nichita and Petitfrere [2013], Petitfrere and Nichita [2014a], Petitfrere and Nichita [2014b]]; un quatrième papier a été soumis [Petitfrere et al. [2014]] et trois autres papiers sont en préparation et seront soumis prochainement (ils correspondent aux sections 3.2.4, 3.3 and 3.5).

6.2 Perspectives

A travers le travail réalisé, de nouvelles perspectives apparaissent. Dans cette thèse, des algorithmes de minimisation ont été implémentés et ont montré être plus efficaces que les méthodes préexistantes (Trust-Region, BFGS). Ces algorithmes sont indépendants du modèle d'équation d'état. Lorsque la complexité du modèle d'équation d'état augmente, les calculs des fugacités et de leurs dérivées deviennent de plus en plus prohibitifs. Les méthodes permettant de diminuer le nombre d'itérations par rapport aux méthodes classiques (exemple: Trust-Region) deviennent donc de plus en plus intéressantes. Ainsi, les méthodes quasi-Newton et Trust-Region fonctionneraient plus efficacement avec des modèles d'EoS plus complexes; les méthodes de Trust-Region parce qu'elle diminue le nombre d'itérations et la méthode BFGS car elle ne requiert pas les dérivées de la fugacité.

Ensuite, la méthode de BFGS a été développée pour les calculs d'équilibres diphasiques et une extension aux problèmes multiphasiques est envisagée. Avec un plus grand nombre de phases, la dimensionnalité du système à résoudre, à chaque itération de Newton s'accroît, et la procédure de BFGS (où aucun système linéaire n'est résolu) pourrait devenir très compétitive face à la méthode de Newton.

De nouvelles perspectives apparaissent aussi dans la simulation de réservoir compositionnelle. Le programme de calculs d'équilibre développé a montré être très efficace et robuste et pourrait être utilisé pour simuler de nouveaux procédés. En ajoutant un terme source aux équations de conservation de masse et

d'énergie, des réactions chimiques irréversibles (basées sur des énergies d'activation) pourraient être modélisées. Des simulations du procédé d'Upgrading In-Situ pourraient être effectuées en utilisant cette méthodologie [Lapene [2010]].

Différentes méthodes d'accélération ont aussi été présentées. L'une d'entre elles est la méthode Three-Phase bypass [Zaydullin et al. [2013]] . Elle pourrait être utilisée pour aider à l'identification des phases. Lorsque deux phases liquides sont présentes, il est parfois très difficile d'identifier chacune d'entre elles. La paramétrisation permettrait d'identifier par continuité chaque phase sur l'ensemble de l'espace des compositions.

Une autre méthode d'accélération a été proposée par [Gaganis and Varotsis [2012]] et pourrait être très efficace dans le cadre de simulations compositionnelles pour des procédés complexes (tel que l'injection de CO_2). En utilisant une méthode de machine-learning, ils entraînent le système à générer des fonctions capables de fournir le signe de la fonction du plan tangent à son minimum, en quelques opérations. Il pourrait être intéressant de comparer la méthode face à Three-Phase bypass.

Enfin, les simulations présentées pourraient être améliorées en utilisant un modèle d'équation d'états plus précis. Il est bien connu que les paramètres volumiques et que la plupart des propriétés dérivées ne sont pas modélisées correctement par les équations d'état cubiques. Des équations plus complexes pourraient être utilisées pour réaliser les mêmes simulations d'injection de vapeur proposées dans ce document.

INTRODUCTION

1 The heavy oils, a viable alternative to the increasing energy demand

Emerging economies, and particularly in Asia such as China and India are more and more energy dependent. In this context, the need in energy production is rising every year and the International Energy Outlook 2013 (IEO2013) projects that world energy consumption will grow by 56 percent between 2010 and 2040. Fuels represent the largest source of energy, and their production must be increased to satisfy the growing energy demand.

Oil can be classified in different categories depending on their API gravity; a measure of the weight of liquid petroleum compared to water:

- If the API gravity is higher than 10, the oil floats on water.
- If it is lower, the oil sinks in water.

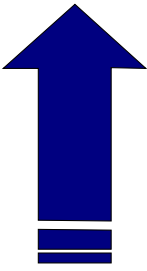
API Gravity	Liquid	Relative Density
0-10	Extra Heavy Crude	Heaviest
10	Water	
10-22	Heavy Crude Oil	
22-38	Medium Crude Oil	
>33	Light Crude Oil	
>55	Condensates	
		Lightest

Figure 1: API gravity

The conventional oils have an API gravity above 25. Due to their relatively low density and viscosity, they generally can be recovered easily and their production cost remain quite small. This is why they have been exploited for many years. Yet, the conventional oils and gas exploration targets become increasingly rare; their production declines at a rate of about 5 percent and the proved global reserves can provide energy for the next 40 years at the current production rate.

Heavy oils can be found with an API gravity between 5 and 22 (see fig. 1). Since they have high viscosities, Enhanced Oil Recovery (EOR) techniques have to be used to be able to reach a sufficient yield, which makes them more expensive than the conventional oils. Nevertheless, the price of the barrel growing with the demand, the exploitation of heavy oils becomes more and more profit making and could play a role in reserve replacement. With a volume of oil-in-place estimated between 3 and 4 trillion barrels and potential reserves of around 500 billion barrels, the heavy oils are equivalent to 60 percent of the global reserves of conventional crude oil and account for 20 up to 25 percent of the global petroleum resources. Their exploitation could extend the world's energy reserves by 15 years. Various EOR techniques exist to enable the heavy oils recovery.

2 Enhanced oil recovery

The development and commercial deployment of modern EORs is today seen as the way to extend the oil production in recoverable reserves. When implemented, EORs make possible to expand the companies' resources. Some estimates suggest that the growth of just 1 percent of the global Oil Recovery Factor could increase conventional oil reserves by around 88 billion barrels (three times the current production).

Moreover, not only could EOR methods be a solution to expend the conventional oil production (which were traditionally operated using Primary or Secondary recoveries (fig. 2)), but they could be used for non-conventional oils such as heavy oils which cannot be recovered by standard depletion methods (as it has been mentionned earlier).

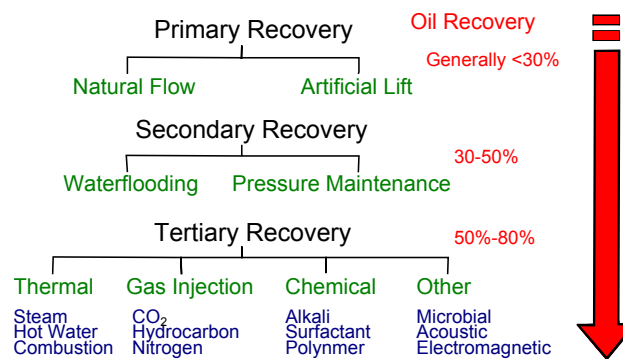


Figure 2: Enhanced oil Recovery [Larry [1989]]

Different type of EORs exist (some are listed in fig. 2), the main methods are given by:

- Gas injection: the introduction of CO_2 or N_2 . By injecting one of those gas into the reservoir, the gas dissolves in oil. The viscosity of any hydrocarbon is then reduces which makes the oil easier to recover.
- Chemical Methods
 - Surfactant flooding can create micro emulsions at the interface between crude oil and water, thus reducing the interfacial tension. This mobilizes the residual oil and allows recovering trapped oil within porous media.
 - Polymer flooding is an augmented water flooding. By co-injecting polymer, the oil/mobility ratio decreases which creates a larger front, thus sweeping a larger volume of oil.
- Thermal recovery methods represent most of the heavy oil projects, they are currently used in production and should probably play a huge role in the future. By raising the temperature the oil is heated with reduces its viscosity. This process increases its mobility (fig. 3) by reducing the surface tension and increasing the permeability. The heated oil may also vaporize and then condense to form an improved oil. The most widely used thermal technics are in-situ combustion, continuous injection of hot fluids such as steam, water or gases, and cyclic operations. Among them, the steam injection processes represent the main type of thermal stimulation of oil reservoirs.

Three main designs of steam injection exists in the industry:

- With two vertical wells (one producer and one injector) separated from a certain distance, for oils with medium viscosities (fig. 4a). Different regions can be seen in fig. 4b.
 - * In the steam zone, close to the injection well, three phase coexist: the gas, oil and water phase. The temperature is quite uniform, as well as the oil saturation.

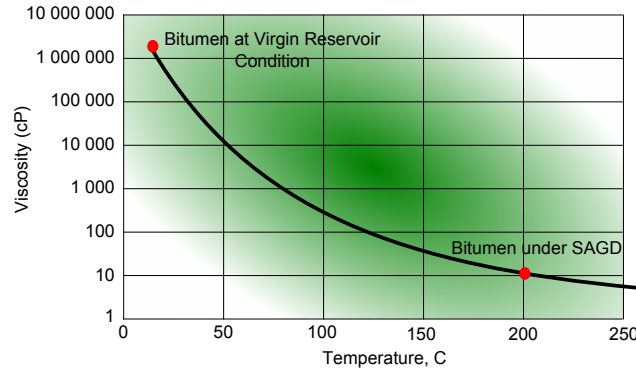


Figure 3: Viscosity versus Temperature

- * Farther, the temperature decreases, water and the light oil components condense in the contact of the cold matrix.
- * Then, the oil is displaced by water (water flooding) in a third zone.
- * Finally far from the front, the conditions are identical to the initial reservoir conditions. The fluid contains the original fluid saturation.

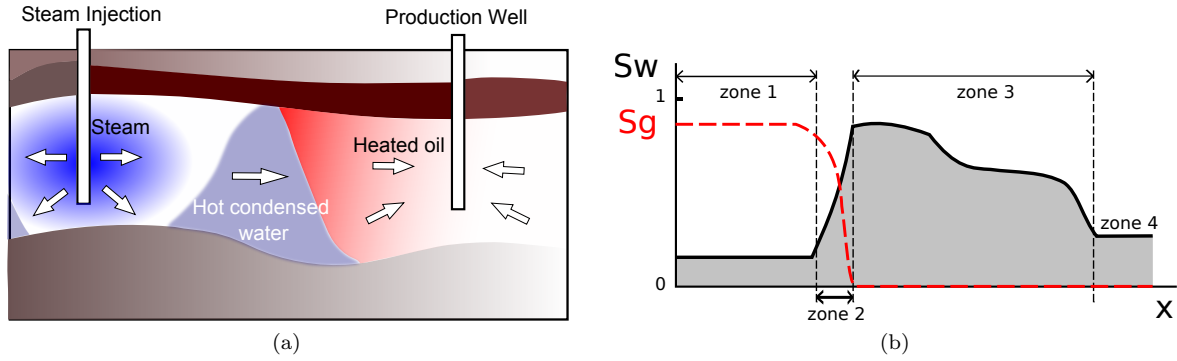


Figure 4: a) Steam injection with vertical wells and b) Steam injection analysis [Baker [1969] and Mandl and Volek [1969]]

- With one vertical well plays both the role of producer and injector. The process is called cyclic steam stimulation (or Huff and Puff method) and is quite efficient for oils with extra heavy viscosities. First steam is injected to heat the oil to reduce its viscosity. Then oil is produced first by means of natural flow, then by artificial lift. The two phases are repeated several times.
- Finally, the SAGD process (Steam Assisted Gravity Drainage), is really efficient for heavy oils with extremely high viscosities. This process is already used in production. In fig. 5a a SAGD process is represented. Steam is injected in a well located upper a production well. With the temperature, the viscosity of the oil decreases and becomes mobile. By gravity, the oil flows towards the production well along the steam chamber (see fig. 5b).

3 Improved reservoir simulation and thermodynamics

3.1 A need to improve the reservoir simulators

Simulations of thermal processes are not easy. Steam injection creates compositional effects such as steam distillation, condensation and vaporization which are essential to take into account. Three phase regions are in the heart of the processes (water, gas, oil) and should be modeled accurately.

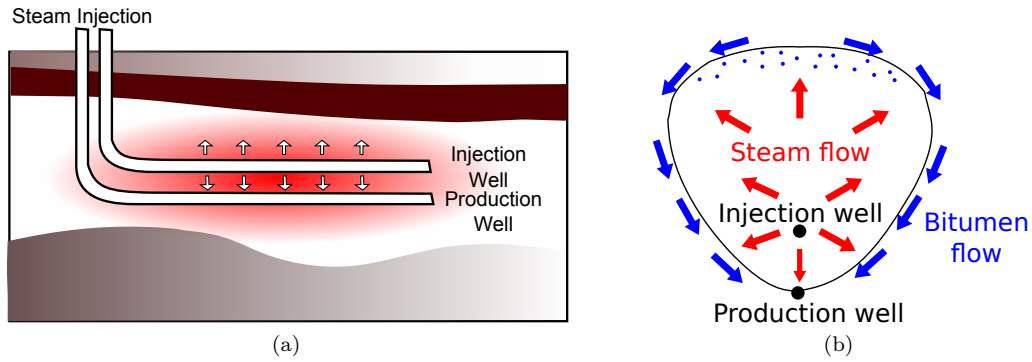
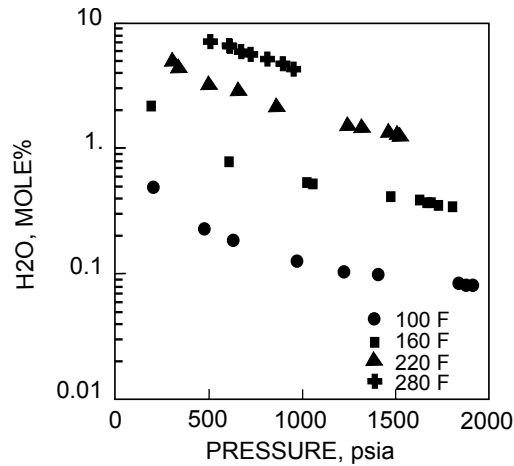


Figure 5: a) SAGD process and b) SAGD analysis

Most of the commercial compositional reservoir simulators (for instance ECLIPSE 300, INTERSECT) treat the water phase as pure and do not account for the water solubility in the hydrocarbon phase. Those compositional thermal simulators treat water separately from the equilibrium calculation which can lead to important inaccuracies.

Indeed, the solubility of water in the oil can become significant for high temperatures. Fig. 6 gives experimental data from [McKetta and Katz [1948]] showing the solubility of the water in the hydrocarbon-rich phase for different pressures and temperatures. Besides, [Wang [1990]] showed that the solubility of water in hydrocarbon-rich liquid can climb up to 50 percent at high temperatures. The solubility values change the viscosity, the density and other properties and may affect the production performance.

Figure 6: Water solubility in gas phase for the C1/nC4/H₂O system, after [McKetta and Katz [1948]]

In STARS (simulator from CMG), water is treated within the equilibrium calculation problem. However the simulator is based on the K-value model (this is also the case in ECLIPSE or INTERSECT when using K-value models). K-value tables can be tuned to match phase-diagrams. However, the tables are generally functions of one oil component, the pressure and the temperature. The K-values approach can lead to important inaccuracies when dealing with processes with high compositional changes, such as thermal processes. [Varavei [2009]] showed significant differences in the oil production performing the same simulation with a K-value model and an equation of state (EoS)-based approach for a steam injection test case.

In the literature, many authors proposed simulations of the steam flooding process. Starting in the late sixties [Spillette and Nielsen [1968], Shutler [1969], Shutler [1970]] were the first to propose numerical models to simulate this process. Progressively, more sophisticated reservoir simulators were designed including more

complex physical phenomena and numerical improvements [Coats [1976], Coats [1978]], [Ishimoto [1985], Chein et al. [1989]]. In most of the simulators, either water was treated as an ideal solution in compositional simulators or a K-value model was used.

In the last few years, some authors started to perform full three phase compositional simulations of the steam flooding process. [Brantferger [1991]], Voskov et al. [2009], Varavei and Sepehrnoori [2009]] and more recently [Zaydullin et al. [2014]] developed full three phase flash simulations and showed simulations of steam injection problems.

However, no case of fully compositional simulations of extra-heavy oils steam injection or SAGD processes have been proposed in the literature yet. Those simulations appear essential and will offer better tools to predict the possible profitability in the exploitation of heavy oil fields.

Three to four phase multiphase calculations are in the heart of tertiary recovery simulations. In gas injection, micro-emulsion, steam-injections processes, an additional phase from the oil-gas system is added to the set, which plays a huge impact on the oil recovery. Thermodynamics is in the heart of those processes and it is important to simulate accurately these multiphase systems. In chemical process simulators, the high number of components to deal with makes the equilibrium calculations time-consuming. Generally pseudo-components are generated to decrease the dimensionality of the system leading to approximations of the original problem. For all these reasons, robust time-saving algorithms are necessary to deal with those applications.

3.2 Multiphase equilibrium calculations

The resolution of the multiphase phase-split problem is based on the minimization of the Gibbs free energy. Different types of algorithms have been developed to model this problem. [Michelsen [1982b]] and Michelsen [1982a]] developed a methodology in which stability analysis, based on the tangent plane distance (*TPD*) minimization, is used alternatively with a phase-split algorithm.

Two different kinds of algorithms have been developed in the literature.

- A safe but time-consuming procedure uses global minimization algorithms. [Sun and Seider [1995]] developed an interval-Newton procedure guaranteeing to find the global minimum. [Stadtherr et al. [1995]] developed a method based on homotopy-continuation functions. [Lucia et al. [2000]] used binary tangent plane distance to test all pairs of components based on a Sequential Quadratic Programming (SQP) algorithm. [Nichita et al. [2002]] developed a tunneling based method.
- Local minimization procedures are faster, but they require sculptured initializations. For phase equilibrium calculations, [Michelsen [1982a]] proposed a procedure which generally converges to the global minimum. In a context where time is crucial, this methodology has been widely used in the reservoir simulation.

In the multiphase equilibrium calculation problem, calculation algorithms were improved ensuring convergence even close to difficult regions (such as in the vicinity of singularities: critical points for flash calculations and the stability test limit locus for stability analysis) and to increase the probability to converge to the global minimum. Starting with [Risnes et al. [1981]] who proposed a successive substitution method for multiphase equilibrium problems, many improvements were proposed to improve multiphase equilibrium calculations ([Nghiem and Heidemann [1982]], [Mehra et al. [1982]], [Michelsen [1994]]). [Michelsen [1982b]] provided initial guesses for multiphase problems which were further extended by [Li and Firoozabadi [2012]] who proposed a general strategy for Stability Testing and Phase-split Calculations in two and three phases. However, new improvements are still necessary, namely in the difficult regions, where a high number of iterations are still required by the current equilibrium algorithms to reach convergence.

The research work undertaken in this thesis focuses on multiphase equilibrium calculations able to simulate different compositional simulation problems such as steam injection with extra-heavy oils, within acceptable times.

4 Overview of the thesis

In compositional thermal reservoir simulations, conservation equations must be solved at each time step. Local equilibrium is assumed in each cell and a huge amount of phase equilibrium calculations is involved, based on the minimization of the Gibbs free energy. A single failure may cause significant error propagation leading to false solutions. Thus, it is imperative that calculation algorithms are efficient and highly robust: reliability has to be kept ensuring in the same time a reasonably fast convergence even in very difficult cases. In this thesis, improvements in multiphase equilibrium calculations are proposed in order to simulate the steam injection process. The thesis is organized in five parts

The first chapter briefly presents some fundamental thermodynamics and the Cubic EoS model. The cubic equations of state provide a reasonable description of the volumetric and phase behavior of pure components and mixtures, requiring only the critical properties and the acentric factors of each component. They have been widely used in the reservoir simulation framework and there are also used in this work.

In a second chapter, the minimization algorithm (from [Michelsen [1982a]]) is presented. The stability testing and the phase-split problems are described as well as the general multiphase equilibrium calculation procedure to minimize the Gibbs free energy.

In a third chapter, multiphase equilibrium algorithms improvements are presented. Most of the equilibrium calculations are based on the Newton-Raphson method and use conventional independent variables. These algorithms are first investigated and different extensions are proposed. The logarithm of the equilibrium constants ($\ln \mathbf{K}$) seems to be the best choice of conventional independent variables for multiphase equilibrium calculations. In this work, by extending [Michelsen [1982a]] to multiphase-split problems, we propose to calculate the Jacobian matrix as a product of two symmetric matrices, and to solve the linear system using a Cholesky factorization in order to reduce the number of arithmetic operations for each iteration.

The reduction methods introduced by [Michelsen [1986] and Hendriks [1988]] enable to reduce the space of study from nc (the number of components) for conventional variables to M (with $M < nc$). An extension of the new reduction method proposed by [Nichita and Graciaa [2011]], based on the multi-linear expression of the logarithm of fugacity coefficients is done for stability analysis and multiphase-split calculations. Furthermore, comparisons between different conventional and reduction methods are performed for the stability testing and the multiphase-split calculation problems. Until now, the comparisons from the literature have focused on the global CPU times to compute the whole multiphase equilibrium problem. In this work, comparisons are performed independently for each problem (stability analysis, two phase-split and multiphase-split calculations). Besides, the condition number and convergence paths are also investigated leading to more detailed and extensive comparisons.

The Newton method requires a positive definite Hessian to be applicable. In a new section, other kinds of minimization methods are investigated which could overcome this constraint and be applicable for each iteration. The Quasi-Newton procedures for example guarantee a descent direction for each iteration. These methods represent an interesting alternative since they approach second order methods without the need to compute the second derivatives and no linear system needs to be solved explicitly. A BFGS (Broyden-Fletcher-Goldfarb-Shanno) approach applied with a new set of variables is here proposed for two-phase split calculations. By adding a diagonal element to the Hessian, the Trust-Region methods enable to perform supra-linear up to quadratic steps even when the Hessian is not positive definite. A descent direction is always guaranteed which makes it an attractive procedure. A Trust Region based method is also proposed in

this same chapter and applied both for stability testing and multiphase-split calculation problems.

The generation of pseudo-components capable of representing the fluid behavior accurately is necessary to obtain accurate simulations of an enhanced oil recovery process. In a forth chapter, an algorithm is proposed to improve the characterization of a fluid based on the semi-continuous thermodynamics. Assuming the fluid composition as semi-continuous, a new methodology based on a Gaussian quadrature is proposed to mathematically compute a set of pseudo-components capable of representing the fluid behavior. The methodology can be seen as a lumping-delumping procedure.

Through this thesis, a program including all the presented new algorithms has been developed to deal with multiphase equilibrium calculations with an arbitrary number of phases. Besides, this program was included within two different reservoir simulator; the TPP platform (internal reservoir simulator from Total S.A.) and AD-GPRS (Automatic Differentiation General Purpose Reservoir Simulator) from Stanford University. In the last chapter, tests of the program in stand-alone computations are performed. Comparisons are presented between the results obtained with the developed program and literature and experimental data. Then different reservoir simulations are carried out to simulate different enhanced oil recovery processes. A comparison with ECLIPSE is performed for a three-phase water injection problem (with the water treated outside the equilibrium calculations). Then fully three- and four-phase compositional reservoir simulations of isothermal CO₂ injection problems are presented revealing the capability of the algorithm to handle more than three-phase simulations.

Finally, fully compositional thermal simulations of steam injections are carried out. In a first step, simulations of the steam flooding process in heterogeneous and homogeneous conditions is realized. In a second step, a simulation of the SAGD process is presented with an extra heavy mixture from Canada. In this last part, the developed equilibrium program is tested with different stability bypass procedures and computational times are presented for all the steam injection cases.

CHAPTER 1

Fundamental thermodynamics

Contents

1.1 Thermodynamics functions	22
1.1.1 Internal Energy	22
1.1.2 Gibbs free energy	24
1.1.3 Helmholtz free energy	25
1.1.4 Enthalpy	25
1.1.5 Residual energy	25
1.1.6 Maximization of the entropy	25
1.1.7 Fugacity and fugacity coefficients	26
1.1.8 Condition for equilibrium at constant pressure and temperature	28
1.2 Equation of State Calculations	29
1.2.1 Introduction	29
1.2.2 Cubic equations of state	29
1.2.3 Fugacity coefficient	31
1.2.4 Molar Gibbs free energy	31

An accurate modeling of the different equilibrium phases is highly important in reservoir simulations. Many equations of state exist relating the thermodynamic variables. In this thesis, the development of a general algorithm to solve the multiphase flash problem is proposed. Cubic EoS provide an efficient way to model the different thermodynamic properties required in a reservoir simulation (enthalpies, densities), as well as the phase envelope construction and they generally lead to an acceptable accuracy for hydrocarbon mixtures. In the reservoir simulation context, where computation times should remain acceptable, cubic EoS have been widely used, and are used in this thesis. A description of the cubic EoS is presented in this chapter and the multiphase equilibrium calculation framework is presented, as well as the thermodynamic functions that need to be computed in the reservoir simulation.

1.1 Thermodynamics functions

1.1.1 Internal Energy

1.1.1.1 First Law

The first law of thermodynamics corresponds to the conservation of the energy for a system.

For a closed system, for any processes, an infinitesimal change in the internal energy is considered due to a combination of heat added to the system Q and work done by the system W . Taking δU as an infinitesimal change in internal energy, and considering only the internal energy variations, the work done by pressure and the thermal exchanges, the first law can be written [Vidal [1997]]:

$$dU = \delta Q + \delta W \quad (1.1)$$

In case of equilibrium mixtures, only the work due to the pressure is taken into account (electrical forces are negligible, no mechanical forces,...). It can be expressed as [Vidal [1997]]

$$\delta W = -pdV \quad (1.2)$$

1.1.1.2 Second law

The entropy change is related to the heat added to the system and the temperature. During a transformation, the entropy variation can be splitted into two different terms [Vidal [1997]]

$$dS = dS_i + dS_e \quad (1.3)$$

where dS_e is related to heat exchanges,

$$dS_e = \frac{\delta Q}{T} \quad (1.4)$$

For reversible transformations, the second law, states that

$$dS_i \geq 0 \quad (1.5)$$

Therefore,

$$dS \geq \frac{\delta Q}{T} \quad (1.6)$$

Using (eq. 1.1),

$$dU + PdV - TdS \leq 0 \quad (1.7)$$

In a reversible process, an infinitesimal variation of the entropy is due to an infinitesimal variation of heat divided by the temperature of the system. If the work done to the system is limited to the work done by pressure one can write [Vidal [1997]]

$$dS = \frac{\delta Q}{T} \quad (1.8)$$

Combining (eq. 1.1, eq. 1.2, eq. 1.8), one obtains

$$dU = TdS - pdV \quad (1.9)$$

U is here a function of S and V . The total differential of the internal energy U can be written as:

$$dU = \left(\frac{\partial U}{\partial S} \right)_V dS + \left(\frac{\partial U}{\partial V} \right)_S dV \quad (1.10)$$

which gives by using (eq. 1.9) in (eq. 1.10),

$$\left(\frac{\partial U}{\partial S} \right)_V = T \quad \left(\frac{\partial U}{\partial V} \right)_S = -p \quad (1.11)$$

For a mixture with different components, the total internal energy of a system can be expressed as a function of S , V and the mole numbers $\mathbf{n} = [n_1, n_2, \dots, n_{nc}]$ [Vidal [1997]]. By definition the total differential of the internal energy U can be written as:

$$dU = \left(\frac{\partial U}{\partial S} \right)_{V, \mathbf{n}} dS + \left(\frac{\partial U}{\partial V} \right)_{S, \mathbf{n}} dV + \sum_{i=1}^{nc} \left(\frac{\partial U}{\partial n_i} \right)_{S, V, n_{j \neq i}} dn_i \quad (1.12)$$

with the chemical potential of the component i given by

$$\mu_i = \left(\frac{\partial U}{\partial n_i} \right)_{n_{j \neq i}, V, S} \quad (1.13)$$

Moreover, the derivatives in (eq. 1.11) still holds for mixtures since V , S and \mathbf{n} are independent variables of U . (eq. 1.11) becomes

$$\left(\frac{\partial U}{\partial S} \right)_{V, \mathbf{n}} = T \quad \left(\frac{\partial U}{\partial V} \right)_{S, \mathbf{n}} = -p \quad (1.14)$$

Combining (eq. 1.12, eq. 1.13 and eq. 1.14) leads to

$$dU = TdS - pdV + \sum_{i=1}^{nc} \mu_i dn_i \quad (1.15)$$

Moreover, as all the independent variables of U (S , \mathbf{n} and V) are extensive variables, using Euler's theorem with (eq. 1.14) leads to

$$U(S, V, \mathbf{n}) = TS - pV + \sum_{i=1}^{nc} \mu_i n_i \quad (1.16)$$

1.1.2 Gibbs free energy

The Gibbs free energy is given in [Vidal [1997]]:

$$G(p, T, \mathbf{n}) = U + pV - TS \quad (1.17)$$

by differentiating this expression:

$$dG(p, T, \mathbf{n}) = dU + pdV + Vdp - TdS - SdT \quad (1.18)$$

and using (eq. 1.15) in (eq. 1.18), one obtains

$$dG(p, T, \mathbf{n}) = \sum_{i=1}^{nc} \mu_i dn_i + Vdp - SdT \quad (1.19)$$

which gives in terms of partial derivatives:

$$\left(\frac{\partial G(p, T, \mathbf{n})}{\partial p} \right)_{\mathbf{n}, T} = V \quad \left(\frac{\partial G(p, T, \mathbf{n})}{\partial T} \right)_{\mathbf{n}, p} = -S \quad \left(\frac{\partial G(p, T, \mathbf{n})}{\partial n_i} \right)_{p, T, n_{j \neq i}} = \mu_i(p, T, \mathbf{n}) \quad (1.20)$$

Using (eq. 1.16) in (eq. 1.17),

$$G(p, T, \mathbf{n}) = \sum_{i=1}^{nc} \mu_i n_i \quad (1.21)$$

and taking the total differentiation of (eq. 1.21),

$$dG = \sum_{i=1}^{nc} n_i d\mu_i + \sum_{i=1}^{nc} \mu_i dn_i \quad (1.22)$$

Using (eq. 1.22 and eq. 1.19), the Gibbs-Duhem equation is obtained,

$$\sum_{i=1}^{nc} n_i d\mu_i = V dp - S dT \quad (1.23)$$

1.1.3 Helmholtz free energy

The Helmholtz free energy is given by [Vidal [1997]]:

$$A(V, T, \mathbf{n}) = U - TS \quad (1.24)$$

Differentiating this equation, and using (eq. 1.15), one obtains

$$dA = \sum_{i=1}^{nc} \mu_i dn_i - S dT - p dV \quad (1.25)$$

which leads to the following partial derivatives:

$$\left(\frac{\partial A(T, V, \mathbf{n})}{\partial V} \right)_{T, \mathbf{n}} = -p \quad \left(\frac{\partial A(T, V, \mathbf{n})}{\partial T} \right)_{V, \mathbf{n}} = -S \quad \left(\frac{\partial A(T, V, \mathbf{n})}{\partial n_i} \right)_{T, V, n_{j \neq i}} = \mu_i(T, V, \mathbf{n}) \quad (1.26)$$

1.1.4 Enthalpy

The enthalpy is given by means of a Legendre transformation [Vidal [1997]]

$$H(p, S, \mathbf{n}) = U + pV \quad (1.27)$$

1.1.5 Residual energy

A residual quantity (M^r) is the difference between the property of a real mixture (M) and that of an ideal gas at the same pressure, temperature and composition (M^*) (or at the same conditions of volume temperature and composition):

$$M(T, p, \mathbf{n}) = M^*(T, p, \mathbf{n}) + M^r(T, p, \mathbf{n}) \quad (1.28)$$

$$M(T, V, \mathbf{n}) = M^*(T, V, \mathbf{n}) + M^r(T, V, \mathbf{n}) \quad (1.29)$$

1.1.6 Maximization of the entropy

The second law states that

$$dS \geq 0 \quad (1.30)$$

At equilibrium in an isolated system, the system does not exchange heat or work with the surroundings,

$$dW = 0 \quad dQ = 0 \quad (1.31)$$

Therefore,

$$dU = 0 \quad dV = 0 \quad (1.32)$$

Following [Vidal [1997]] procedure, let split the system within two subsystems A and B separated with an 'artificial barrier', we have

$$dU = dU_A + dU_B = 0 \quad dS = dS_A + dS_B \quad (1.33)$$

and

$$dV_A + dV_B = 0 \quad (1.34)$$

Using (eq. 1.15), one obtains

$$dS = \left(\frac{1}{T_A} - \frac{1}{T_B} \right) dU_A + \left(\frac{P_A}{T_A} - \frac{P_B}{T_B} \right) dV_A \quad (1.35)$$

At equilibrium, $dS = 0$, the second law states that equilibrium is obtained at the maximum of the entropy, which corresponds here to the equality of the temperatures and pressures.

Moreover, using (eq. 1.7) at constant pressure and temperature, it leads to

$$dG \leq 0 \quad (1.36)$$

At equilibrium,

$$dG = 0 \quad (1.37)$$

Using (eq. 1.36) and (eq. 1.37), the equilibrium corresponds to the minimum of the Gibbs free energy.

1.1.7 Fugacity and fugacity coefficients

In the case of an ideal gas, the ideal gas law can be applied:

$$V = \frac{nRT}{p} \quad (1.38)$$

Moreover, using (eq. 1.20), the Maxwell relation gives:

$$\left(\frac{\partial \mu}{\partial p} \right)_T = \left(\frac{\partial V}{\partial n} \right)_T \quad (1.39)$$

Considering the fluid as a pure component, the left side of the equation (eq. 1.39) corresponds to the molar volume V_m . For a perfect gas, $V_m = RT/p$, (eq. 1.39) becomes at constant temperature

$$\int_{p_0}^p d\mu = \int_{p_0}^p \frac{RT}{p} dp \quad (1.40)$$

Which leads to

$$\mu^*(T, p) = \mu^*(T, p_0) + RT \ln \frac{p}{p_0} \quad (1.41)$$

In case of real fluid, [Lewis and Randall [1923]] introduced the fugacity f to act as an effective pressure. The fugacity represents the value that must be substituted for pressure in order to preserve the expressions for ideal gas [Vidal [1997]]. The relation (eq. 1.40) becomes for real fluids [Michelsen and Mollerup [2007]]

$$\mu(T, p) - \mu^*(T, p_0) = RT \ln \frac{f(T, p)}{p_0} \quad (1.42)$$

which gives taking the difference of (eq. 1.42) and (eq. 1.41),

$$\mu(T, p) - \mu^*(T, p) = RT \ln \frac{f(T, p)}{p} \quad (1.43)$$

The term within the logarithm corresponds to the fugacity coefficient:

$$\phi = \frac{f}{p} \quad (1.44)$$

which is the ratio that indicates how the fugacity of a real substance deviates from that of an ideal gas. For ideal mixtures, the partial pressure is given by [Vidal [1997]] $p_i = x_i p$, this gives

$$\mu_i^*(T, p) = \mu^*(T, p_0) + RT \ln \frac{x_i p}{p_0} \quad (1.45)$$

For real mixtures, the expression of the fugacity is given for each components:

$$\mu_i(T, p, \mathbf{n}) = \mu^*(T, p_0) + RT \ln \frac{f_i(T, p, \mathbf{n})}{p_0} \quad (1.46)$$

Taking the difference (eq. 1.46 - eq. 1.45),

$$\mu_i(T, p, \mathbf{n}) = \mu_i^*(T, p) + RT \ln \frac{f_i}{x_i p} \quad (1.47)$$

The expression of the fugacity coefficient is generalized to

$$\phi_i = \frac{f_i(T, p, \mathbf{n})}{x_i p} \quad (1.48)$$

Using the expression of partial derivatives for ideal mixtures $p_i = x_i p$, one can write

$$\mu_i^*(T, p) = \mu^*(T, p) + RT \ln \frac{x_i p}{p} = \mu^*(T, p) + RT \ln x_i \quad (1.49)$$

Using (eq. 1.47, 1.49),

$$\mu_i(p, T, \mathbf{n}) = RT \ln \phi_i + \mu_i^*(T, p) + RT \ln x_i = RT \ln f_i + \mu_i^*(T, p_0) - RT \ln p_0 \quad (1.50)$$

Now let's find an expression for the fugacity coefficient, taking the Gibbs-Duhem equation (eq. 1.23) for a single component fluid, at constant temperature,

$$d\mu = V_m dp \quad V_m = V/n \quad (1.51)$$

Moreover, by definition of the fugacity, at constant temperature, it can easily be shown that [Vidal [1997]]

$$d\mu = RT d \ln f \quad (1.52)$$

which means that by definition of the coefficients (eq. 1.44),

$$RT d \ln \phi = V_m dp - RT d \ln p = V_m dp - RT \frac{dp}{p} \quad (1.53)$$

When $p \rightarrow 0$, the fluid becomes an ideal state and $\phi \rightarrow 1$, therefore, by integrating (eq. 1.53),

$$\int_{p_0}^p d \ln \phi = \int_{p_0}^p \left(\frac{V_m}{RT} - \frac{1}{p} \right) dp \quad (1.54)$$

Taking $p_0 = 0$,

$$\ln \phi = \int_0^p \left(\frac{V}{nRT} - \frac{1}{p} \right) dp \quad (1.55)$$

For mixtures, at constant temperature, the fugacity definition gives

$$d\mu_i = RT d \ln f_i \quad (1.56)$$

Moreover, using the Maxwell relation with the Gibbs free energy,

$$\left(\frac{\partial \mu_i}{\partial V}\right)_{T, \mathbf{n}} = -\left(\frac{\partial P}{\partial n_i}\right)_{T, V, n_{j \neq i}} \quad (1.57)$$

which leads to

$$d\mu_i = -\left(\frac{\partial P}{\partial n_i}\right)_{T, V, n_{j \neq i}} dV \quad (1.58)$$

Combining (eq. 1.56, eq. 1.58), and eq. 1.48),

$$RT d \ln \phi_i = -\left(\frac{\partial P}{\partial n_i}\right)_{T, V, n_{j \neq i}} dV - RT d \ln p x_i \quad (1.59)$$

with the compressibility factor Z which indicates the deviation from the ideal gas behavior:

$$Z = \frac{pV}{nRT} \quad (1.60)$$

$$RT d \ln \phi_i = -\left(\frac{\partial P}{\partial n_i}\right)_{T, V, n_{j \neq i}} dV + RT d \ln V - RT d \ln \left(\frac{nRT}{x_i}\right) - RT d \ln Z \quad (1.61)$$

when $V \rightarrow \infty$, the system acts as a perfect gas, $\phi_i \rightarrow 1$ and $Z \rightarrow 1$, with $d \ln V = dV/V$, it can be shown that

$$\ln \phi_i = \frac{1}{RT} \int_{\infty}^V \left[\frac{RT}{V} - \left(\frac{\partial P}{\partial n_i}\right)_{T, V, n_{j \neq i}} \right] dV - \ln Z \quad (1.62)$$

1.1.8 Condition for equilibrium at constant pressure and temperature

The Gibbs free energy is given by:

$$G(p, T, \mathbf{n}) = \sum_{i=1}^{nc} \mu_i n_i \quad (1.63)$$

For a closed system, at constant pressure and temperature, the equilibrium is obtained for the state giving the minimum Gibbs free energy. Using equation (eq. 1.50) in (eq. 1.63), minimizing the Gibbs free energy at constant p and T is equivalent to minimize:

$$\frac{G(\mathbf{n})}{RT} = \sum_{i=1}^{nc} n_i \ln f_i \quad (1.64)$$

Which can be extended for multiphase problems:

$$\frac{G(\mathbf{n})}{RT} = \sum_{j=1}^{np} \sum_{i=1}^{nc} n_{ij} \ln f_{ij} \quad (1.65)$$

In this subsection, the mole numbers are taken as independent variables. For each phase, the Gibbs-Duhem relation (eq. 1.23) leads at constant pressures and temperatures,

$$\sum_{i=1}^{nc} n_{ij} \frac{\partial \ln f_{ij}}{\partial n_{kl}} = 0 \quad \forall k = 1, nc \quad \forall l, j = 1, np \quad (1.66)$$

Let z_i be the feed composition of the component i . Taking np as the phase of reference,

$$n_{i, np} = z_i - \sum_{j=1}^{np-1} n_{ij} \quad (1.67)$$

the minimum is found when the elements of the gradient vector are equal to zero.

$$\frac{\partial}{\partial n_{ij}} \left(\frac{G(\mathbf{n})}{RT} \right) = \ln f_{ij} - \ln f_{i,np} = 0 \quad \forall j = 1, np - 1 \quad \forall i = 1, nc \quad (1.68)$$

And therefore, at the equilibrium,

$$f_{ij} = f_{i,np} \quad \forall j = 1, np - 1 \quad \forall i = 1, nc \quad (1.69)$$

1.2 Equation of State Calculations

1.2.1 Introduction

For a real mixture, the compressibility factor Z indicates the deviation from the ideal gas behavior. In the equilibrium problems, at a given temperature and pressure, one needs to compute Z . However, V is unknown. One equation is missing to close the system.

Cubic equations of state (EoS) have been introduced in this attempt. They are simple equations relating pressure, volume and temperature (pVT). They provide a reasonable description of the volumetric and phase behavior of pure compounds and mixtures, requiring only critical properties and acentric factors of each component.

Through this thesis, a program was developed to solve multiphase calculations problem. Cubic equations of state were used in this study, but the program was designed so that other Eos models could be integrated.

1.2.2 Cubic equations of state

The general form of two-cubic EoS is

$$p = \frac{RT}{\nu - b} - \frac{a}{(\nu + \delta_1 b)(\nu + \delta_2 b)} \quad (1.70)$$

where $\delta_1 = 0$ and $\delta_2 = 1$ for the SRK Eos [Soave [1972], Redlich and Kwong [1949]] and $\delta_{1,2} = 1 \pm \sqrt{2}$ for the PR EoS [Peng and Robinson [1976]].

The implicit form (in compressibility factor Z) of the Eos is obtained by substituting $A = \frac{ap}{R^2T^2}$, $B = \frac{bp}{RT}$ and $Z = \frac{p\nu}{RT}$ into (eq. 1.70).

$$\begin{aligned} Z^3 &+ [(\delta_1 + \delta_2 - 1)B - 1] Z^2 + [A + \delta_1 \delta_2 B^2 - (\delta_1 + \delta_2)B(B + 1)] Z \\ &- [AB + \delta_1 \delta_2 B^2(B + 1)] = 0 \end{aligned} \quad (1.71)$$

The van der Waals one-fluid mixing rules are used for the energy A , and for the volume B , parameters of the cubic EoS

$$A = \sum_{i=1}^{nc} \sum_{j=1}^{nc} x_i x_j A_{ij} \quad (1.72)$$

and

$$B = \sum_{i=1}^{nc} B_i x_i \quad (1.73)$$

where

$$A_{ij} = \sqrt{A_i} \sqrt{A_j} (1 - C_{ij}); \quad i, j = 1, nc \quad (1.74)$$

$$A_i = \Omega_a \frac{p_{ri}}{T_{ri}^2} \left[1 + m(\omega_i) \left(1 - \sqrt{T_{ri}} \right) \right]^2 \quad i = 1, nc \quad (1.75)$$

and

$$B_i = \Omega_b \frac{p_{ri}}{T_{ri}}; \quad i = 1, nc \quad (1.76)$$

C_{ij} is an element of the binary interaction parameter matrix (BIP). $T_{ri} = \frac{T}{T_{ci}}$ is the reduced temperature (with T_{ci} the critical temperature), $p_{ri} = \frac{p}{p_{ci}}$, the reduced pressure and ω_i the acentric factor of the component i .

For the PR EoS, $\Omega_a = 0.45724$, $\Omega_b = 0.0778$, $m(\omega) = 0.37464 + 1.54226\omega - 0.26992\omega^2$ for $w \leq 0.49$, and $m(\omega) = 0.379642 + 1.48503\omega - 0.164423\omega^2 + 0.016667\omega^3$ for $w > 0.49$ [Robinson and Peng [1978]]. For the SRK EoS, $\Omega_a = 0.42748$, $\Omega_b = 0.08664$ and $m(\omega) = 0.48 + 1.574\omega - 0.176\omega^2$.

1.2.2.1 Solving the cubic polynomial equation

The Cardan procedure enables to compute analytically the roots of the third order polynomial:

$$Z^3 + a_1 Z^2 + a_2 Z + a_3 = 0 \quad (1.77)$$

Taking

$$Q = \frac{3a_2 - a_1^2}{9} \quad (1.78)$$

$$R = \frac{a_2 a_1}{6} - \frac{a_3}{2} - \frac{2a_1^3}{27} \quad (1.79)$$

the Discriminant is given by:

$$D = Q^3 + R^2 \quad (1.80)$$

- If $D > 0$, there is only one real root:

$$Z = \sqrt{3}R + \sqrt{D} + \sqrt{3}R - \sqrt{D} - \frac{a_1}{3} \quad (1.81)$$

- If $D = 0$, there are three real roots, with two equal:

$$Z_1 = 2\sqrt{3}R - \frac{a_1}{3} \quad (1.82)$$

$$Z_2 = Z_3 = -\sqrt{3}R - \frac{a_1}{3} \quad (1.83)$$

- Else if $D < 0$ there are three real roots:

Introducing θ

$$\theta = \arccos \left(\frac{R}{\sqrt{-Q^3}} \right) \quad (1.84)$$

the expression of the roots becomes

$$Z_1 = 2\sqrt{-Q} \cos\left(\frac{\theta}{3}\right) - \frac{a_1}{3} \quad (1.85)$$

$$Z_2 = 2\sqrt{-Q} \cos\left(\frac{\theta}{3} + \frac{2\pi}{3}\right) - \frac{a_1}{3} \quad (1.86)$$

$$Z_3 = 2\sqrt{-Q} \cos\left(\frac{\theta}{3} + \frac{4\pi}{3}\right) - \frac{a_1}{3} \quad (1.87)$$

However, even if the Cardan method provides an analytical solution, in some situations numerical instabilities can occur [Monroy-Loperena [2012]]. In these cases, a Newton procedure is used to refine the roots.

When multiple roots are found, in two phase equilibrium calculations, generally, the smallest root is taken for liquid and the largest is taken for gas. However, in this work, for multiphase calculations, the root which minimizes the Gibbs free energy is selected for each phase.

1.2.3 Fugacity coefficient

Combining (eq. 1.62 and eq. 1.70), the fugacity coefficient can be expressed:

$$\begin{aligned} \ln \phi_i &= (Z-1) \frac{B_i}{B} - \ln(Z-B) - \frac{A}{(\delta_1 - \delta_2)B} \\ &\quad * \left(2 \frac{\psi_i}{A} - \frac{B_i}{B} \right) \ln \left(\frac{Z + \delta_1 B}{Z + \delta_2 B} \right) \end{aligned} \quad (1.88)$$

with

$$\psi_i = \sum_{j=1}^C A_{ij} x_j; \quad i = 1, nc \quad (1.89)$$

1.2.4 Molar Gibbs free energy

The dimensionless molar Gibbs free energy g can be expressed as the sum of the ideal and excess molar Gibbs free energy

$$g = g_I + g_E \quad (1.90)$$

with

$$g_I = \sum_{i=1}^{nc} x_i \ln x_i \quad (1.91)$$

and

$$g_E = \sum_{i=1}^{nc} x_i \ln \phi_i(\mathbf{x}) \quad (1.92)$$

The last expression becomes for cubic EoS models,

$$g_E = (Z-1) - \ln(Z-B) - \frac{A}{\Delta B} \ln E \quad (1.93)$$

where $E = (Z + \delta_1 B)/(Z + \delta_2 B)$ and $\Delta = \delta_1 - \delta_2$

CHAPTER 2

Global multiphase flash procedure

Contents

2.1	Introduction	32
2.2	Stability Analysis	33
2.2.1	The stationary points	34
2.2.2	Initial guesses	35
2.2.3	General minimization algorithm	36
2.2.4	Bypass acceleration method	36
2.3	Multiphase-split calculations	37
2.3.1	Solving the Rachford-Rice equation	39
2.3.2	Phase-split calculations	42
2.4	General multiphase equilibrium algorithm	43

2.1 Introduction

A $P-T$ equilibrium calculation problem consists in finding the phase mole fractions θ and the compositions \mathbf{x} which minimize the Gibbs free energy for a given feed composition \mathbf{z} , temperature T and pressure p (see fig. 2.1).

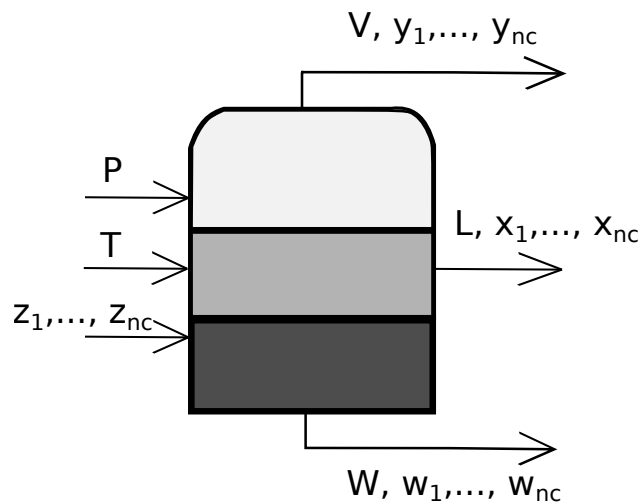


Figure 2.1: Multiphase flash, input/output

In the reservoir simulation framework, many equilibrium calculations must be performed. Global minimization cannot be applied because the algorithms are too time consuming, hence local minimization procedures are preferred. [Michelsen [1982a]] proposed a methodology combining the stability analysis procedure [Michelsen [1982b]] and phase split-calculations:

- The stability analysis problem [Michelsen [1982b]], enables to know if a mixture is stable for a given number of phases or should split into more phases.
- The phase-split algorithm computes the compositions \mathbf{x} and the phase mole fractions $\boldsymbol{\theta}$ which minimize the Gibbs free energy for a given number of phases.

This procedure generally leads to the global minimum and has proven to be highly efficient. It has been implemented in most of the compositional reservoir simulators, and has been used in this work. In this chapter, both problems will be presented in details, as well as the global minimization algorithm used to solve the multiphase equilibrium calculations.

In fig. 2.2 a scheme of the multiphase equilibrium calculation procedure is represented.

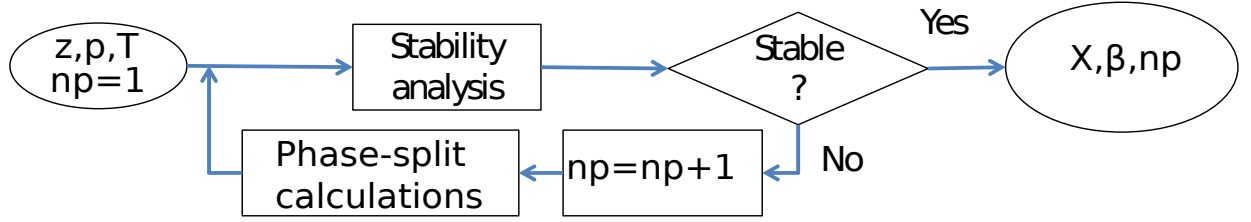


Figure 2.2: Multiphase equilibrium calculation procedure

2.2 Stability Analysis

[Michelsen [1982b]] proposed to solve the stability analysis, based on the tangent plane distance algorithm. Let's now present the method.

At a given temperature and pressure (T_0, P_0) , consider a nc component mixture with component mole fractions $(z_1, z_2, \dots, z_{nc})$. The Gibbs energy of the phase is given by:

$$G_0 = \sum_i^{nc} z_i \mu_i^0 \quad (2.1)$$

with μ_i^0 the chemical potential of component i .

Assume that an infinitesimal amount ϵ of a new phase II is added to the mixture. Let the mole fractions of the new phase be $(x_1, x_2, \dots, x_{nc})$ and let N be the total mole numbers of the single phase. The change in Gibbs energy is:

$$\Delta G = G_I + G_{II} - G_0 = G(N - \epsilon) + G(\epsilon) - G_0 \quad (2.2)$$

A Taylor series expansion of $G(N - \epsilon)$ gives:

$$G(N - \epsilon) = G(N) - \epsilon \sum_i x_i \left(\frac{\partial G}{\partial n_i} \right)_N = G_0 - \epsilon \sum_i x_i \mu_i^0 \quad (2.3)$$

and so,

$$\Delta G = G(\epsilon) - \epsilon \sum_i x_i \mu_i^0 = \epsilon \sum_i x_i (\mu_i(x) - \mu_i^0) \quad (2.4)$$

The right-hand side of the expression gives the tangent plane distance (TPD) function:

$$TPD(x) = \sum_i x_i (\mu_i(x) - \mu_i^0) \quad (2.5)$$

The system always tries to minimize the energy. This means that if ΔG is negative, the system will change to a two phase system. On the other hand, if $\Delta G \geq 0$, the system will remain in one phase. From (eq. 1.50), the dimensionless TPD function becomes [Michelsen [1982b]]

$$D(\mathbf{x}) = \sum_{i=1}^{NC} x_i (\ln x_i + \ln \phi_i(\mathbf{x}) - d_i(\mathbf{z})) \quad (2.6)$$

with

$$d_i = \ln(z_i) + \ln(\phi_i(\mathbf{z})); \quad (2.7)$$

If at all stationary points the TPD function is non negative, the system remains in one phase. When reaching the stationary point:

$$\ln(x_i) + \ln(\phi_i) - d_i = k \quad \forall i = 1, \dots, nc \quad (2.8)$$

By changing the variable: $Y_i = x_i e^{-k}$, it can be noticed that

$$x_i = \frac{Y_i}{Y_T} \quad (2.9)$$

with $Y_T = \sum_{i=1} Y_i$, thus Y_i can be viewed formally as mole numbers. (eq. 2.8) becomes:

$$\ln(Y_i) + \ln(\phi_i) - d_i = 0 \quad \forall i = 1, \dots, nc \quad (2.10)$$

Instead of minimizing the tangent plane distance, [Michelsen [1982b]] proposed a modified TPD function:

$$D^* = \left(1 - \sum_{i=1}^{NC} Y_i\right) + \sum_{i=1}^{NC} Y_i (\ln Y_i + \ln \phi_i(\mathbf{Y}) - d_i) \quad (2.11)$$

The function D^* has the same stationary points and the same sign as the original TPD function D [Michelsen [1982b]]. When testing if a monophasic fluid is stable, the stability is applied to the feed mole fractions \mathbf{z} . For mixtures with M phases, only one phase need to be tested to assume the mixture as stable/unstable in M phases. In this work, following [Li and Firoozabadi [2012]], the phase selected is the one with the highest molecular weight.

2.2.1 The stationary points

At the solution, the stationary points (x_{SP}) are obtained.

- if $\min_{\mathbf{x}_{SP}} TPD(x_{SP}) < 0$, the mixture is unstable, a new phase $\mathbf{y} \neq \mathbf{z}$ is found which will be used as initial guess for the equilibrium
- if $\min_{\mathbf{x}_{SP}} TPD(x_{SP}) = 0$, at the phase boundary, the minimum of the TPD is found for a phase $\mathbf{y} \neq \mathbf{z}$. The mixture is stable.
- if $\min_{\mathbf{x}_{SP}} TPD(x_{SP}) > 0$, the minimum of the TPD is found for a phase $\mathbf{y} \neq \mathbf{z}$. The mixture is stable.
- if $\min_{\mathbf{x}_{SP}} TPD(x_{SP}) = 0$, far from the phase boundary, the minimum of the TPD corresponds to the trial solution \mathbf{z} . The mixture is stable.

An example is given in fig. 2.3a where the tangent plane distance function is plotted at the solution of the stability analysis for the vapor trial phase, for the Y8 mixture [Yarborough [1972]]. In fig. 2.3b, the tangent plane distance of the liquid trial phase is given. In light grey (respectively dark grey) the area converging to a stationary solution (x_{SP}) leading to a negative TPD function (respectively positive TPD function, with a non-trivial solution) is represented. Elsewhere, the problem converges to the trivial solution.

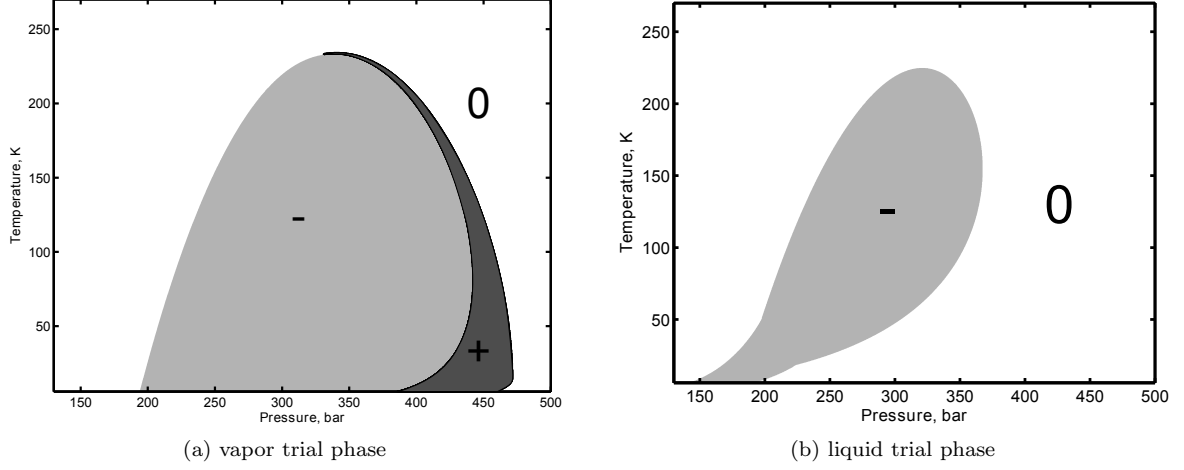


Figure 2.3: Sign of the tangent plane distance for the Y8 mixture

2.2.2 Initial guesses

It has been mentioned that local minimization algorithms were preferred since they are more efficient than the global procedures. However, these procedures do not guarantee to find the global minimum. To increase the probability to reach the global minimum, multiple initial guesses are used which generally cover the whole phase envelope.

[Wilson [1969]] proposed a relation to approximate the equilibrium constants for given pressures and temperatures:

$$K_i^{Wilson} = \frac{p_{c,i}}{p} \exp \left(5.373(1 + w_i) \left(1 - \frac{T_{c,i}}{T} \right) \right) \quad (2.12)$$

For two equilibrium phase calculations, [Michelsen [1982b]] proposed the use the two-sided initialization based on [Wilson [1969]] equilibrium constants:

1. $Y^1 = Z * K_{Wilson}$ (Vapor trial)
2. $Y^2 = Z / K_{Wilson}$ (Liquid trial)

For multiphase equilibrium calculations, [Li and Firoozabadi [2012]] proposed to add other sets of initial guesses:

1. $Y^1 = Z * K_{Wilson}$ (Vapor)
2. $Y^2 = Z / K_{Wilson}$ (Liquid)
3. $Y^3 = Z \sqrt[3]{K_{Wilson}}$ (Vapor)
4. $Y^4 = Z / \sqrt[3]{K_{Wilson}}$ (Liquid)
5. $\forall i / Mw_i < 50, Y_i^k = 0.9, Y_{j \neq i}^k = 0.1 / (nc - 1)$ (Pure components)
6. $\forall i, Y_i^k = 0.9, Y_{j \neq i}^k = 0.1 / (nc - 1)$ (Pure components)

Where Mw_i stands for the molecular weight of the component i .

Of course, the stability analysis is not performed with all the initial guesses. If a negative TPD function is found, the mixture is unstable and there is no need to perform the stability analysis on the remaining initial guesses.

2.2.3 General minimization algorithm

The minimum of the TPD function is obtained when the elements of the gradient are zero.

The elements of the gradient with respect to the mole numbers (Y_i) are given by:

$$\frac{\partial TPD}{\partial Y_i} = \ln Y_i + \ln \phi_i(\mathbf{y}) - d_i \quad (2.13)$$

Newton is a non-linear quadratic method, which enables converging to the solution within 2 or 3 iterations. However, the method requires having a positive definite Jacobian. This is generally not the case for early iterations. The derivatives of the fugacity coefficients can be ill-conditioned far from the solution.

A more robust procedure is needed. In the literature, generally, SSI (Successive Substitution Iterations) are performed at the beginning [Michelsen [1982a]]. The method has been proven to be unconditionally stable [Heidemann and Michelsen [1995]]. Then, close to the solution, the Hessian becomes positive definite and a switch to Newton is carried out.

For the Stability analysis, the SSI update is given by:

$$\ln Y_i^{k+1} = d_i - \ln \phi_i(\mathbf{y}^k) \quad (2.14)$$

It will be costly to compute the Jacobian at each iteration whereas the SSI method does not require any derivatives. Therefore, some switch procedures have been developed for this problem [Hoteit and Firoozabadi [2006]].

In this thesis, the error norm is based on the Euclidean norm of the gradient:

$$S_s = \sqrt{\sum_{i=1}^{nc} (\ln Y_i + \ln \phi_i(\mathbf{y}) - d_i)^2} \quad (2.15)$$

Algorithm 2.1 Stability analysis algorithm using the SSI method

```

Compute  $\ln \phi_i^z$ 
Initialize  $Y_i$  from one of the initial guesses (eq. 2.2.2)
while ( $S_s > \epsilon$ ) do
    Calculate the fugacity coefficients  $\phi_i(\mathbf{y})$ 
    Update  $\mathbf{Y}$  from (eq. 2.15)
end while

```

2.2.4 Bypass acceleration method

Most of the grid cells in the reservoir simulations appear to be in the single phase state. Besides, when a fluid is single phase in a cell, it is most likely to remain monophasic at the following time step.

In the standard procedure, stability analysis is performed at each time step with different initial guesses. If the fluid is monophasic in a cell, [Rasmussen et al. [2006]] proposed an efficient method to skip future stability analysis as long as the conditions (p, T, \mathbf{z}) remain sensibly the same.

However, one needs to make sure that the conditions are sufficiently far from a phase boundary. For a single phase mixture at a given time step (no stationary point of the stability analysis gives a negative TPD function).

- if one of the initial guesses converges to a positive TPD function (with a solution different from the

trivial solution), the conditions are close to a phase boundary and no stability can be bypassed in the future step.

- if all the stationary points converge to the trivial solution, [Michelsen [1982a]] provides a function to evaluate the distance with a two phase boundary:

$$H_{ij} = \delta_{ij} + \sqrt{n_i n_j} \left(\frac{\partial \ln \phi_i}{\partial n_j} \right)_{P,T} \quad (2.16)$$

At the solution, the minimum eigenvalue b of the matrix is computed. Then at the next time step in the reservoir, if the conditions remain sensibly the same, stability analysis can be bypassed because the system is most likely to remain as a one phase mixture.

The conditions to skip calculations are:

$$|z_i^n - z_i^{n-1}| \geq \frac{b^{n-1}}{10} \quad (2.17)$$

$$|P^n - P^{n-1}| \geq \frac{b^{n-1} P^n}{10} \quad (2.18)$$

$$|T^n - T^{n-1}| \geq 10b^{n-1} \quad (2.19)$$

In fig. 2.4, the minimum of the smallest eigenvalue b of the Hessian (eq. 2.16) is given at the solution of both stability analysis (two sided Wilson initialization), for the Y8 mixture [Yarborough [1972]]. It can be noticed that getting away from the two phase boundary, b becomes smaller.

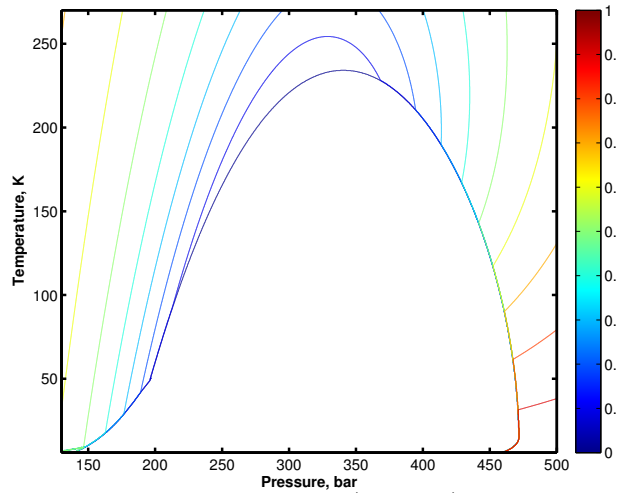


Figure 2.4: Smallest eigenvalue of (eq. 2.16), for the Y8 mixture

2.3 Multiphase-split calculations

Given

- \mathbf{z} the feed composition of each component
- T the temperature
- P the pressure
- A number of phases np

The equilibrium calculation problem solves the phase mole fractions θ as well as the composition for each phase: \mathbf{x} which minimize the Gibbs free energy:

$$G = \frac{Gibbs}{RT} = \sum_{j=1}^{np} \sum_{i=1}^{nc} n_{ij} \ln f_{ij} \quad (2.20)$$

with

$$n_{ij} = x_{ij} \theta_j \quad \forall j = 1, np \quad (2.21)$$

The mass balance equation must also be satisfied,

$$z_i = \sum_{j=1}^{np} \theta_j x_{ij} \quad \forall i = 1, nc \quad (2.22)$$

as well as some closure equations:

$$\sum_{j=1}^{np} \theta_j = 1 \quad (2.23)$$

$$\sum_{i=1}^C x_{ij} = 1 \quad \forall j = 1, np \quad (2.24)$$

Deriving (eq. 2.20) with respect to mole number n_{ij} , the gradient is obtained:

$$\ln f_{ij} - \ln f_{i,np} = 0 \quad i = 1, nc \quad \forall j = 1, np - 1 \quad (2.25)$$

To solve the equilibrium calculations, $nc(np-1)$ independent variables are necessary. Combining equations (eq. 2.24, eq. 2.23 and eq. 2.22), one obtains the Rachford-Rice equation:

$$\sum_{i=1}^{nc} (x_{ij} - x_{i,np}) = 0 \quad \forall j = 1, np - 1 \quad (2.26)$$

Using the equilibrium constant defined by,

$$K_{ij} = \frac{x_{ij}}{x_{i,R}} \quad (2.27)$$

where R denotes the reference phase, the mole fractions $\mathbf{x}_k = (x_{1k}, \dots, x_{nc,k})^T$; $k = 1, np$ are calculated by:

$$x_{ik} = \frac{z_i K_{ik}}{E_i} \quad (2.28)$$

with

$$E_i = 1 + \sum_{\substack{k=1 \\ k \neq R}}^{np} \theta_k (K_{ik} - 1); \quad i = 1, nc \quad (2.29)$$

the Rachford-Rice equation can be obtained:

$$R_j = \sum_{i=1}^{nc} \frac{z_i (K_{ij} - 1)}{1 + \sum_{l=1}^{np-1} \theta_l (K_{il} - 1)} = 0 \quad \forall j = 1, np - 1 \quad (2.30)$$

The Rachford-Rice equations are solved in an inner loop of flash calculations for given K-values, which are updated in the outer loop.

2.3.1 Solving the Rachford-Rice equation

2.3.1.1 Two phase-split calculations

The Rachford-Rice is a function of the phase mole fractions θ . An iterative procedure enables to solve the equation for θ . At the end of the calculation, the composition \mathbf{x} of each phase can be calculated from the mass balance equation.

In two phases, let x_i stands for the liquid composition of the component i , y_i the vapor composition of the component i and θ the vapor mole fraction. The equation becomes,

$$h(\theta) = \sum_{i=1}^C \frac{z_i(K_i - 1)}{1 + \theta(K_i - 1)} = 0 \quad (2.31)$$

The function represents a strictly decreasing function within the negative flash window.

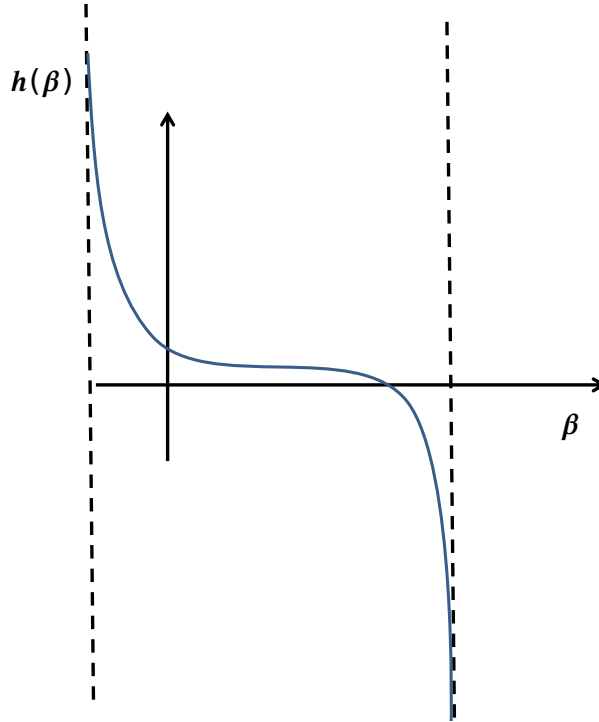


Figure 2.5: Rachford-Rice function

To solve this equation [Michelsen and Mollerup [2007]] proposes the Newton procedure coupled with the dichotomy method. The gradient of the function is given by:

$$h'(\theta) = - \sum_{i=1}^C \frac{z_i(K_i - 1)^2}{[1 + \theta(K_i - 1)]^2} \quad (2.32)$$

When θ has been computed, the compositions can be recovered:

$$x_i = \frac{z_i}{1 - \theta + \theta K_i}; \quad y_i = x_i K_i \quad (2.33)$$

[Nichita and Leibovici [2013]] proposed a new algorithm (used in this thesis) which proved to be really

efficient. The method is now presented.

By introducing:

$$c_i = \frac{1}{1 - K_i} \quad a = \frac{\theta - c_1}{c_n - \theta} \quad (2.34)$$

with c_1 (respectively c_n) corresponding to the maximum (respectively minimum) equilibrium constant ($c_1 = \frac{1}{1 - K_{max}}$) and

$$d_i = \frac{c_1 - c_i}{c_n - c_1} \quad \forall i = 2, nc - 1 \quad (2.35)$$

$$d_1 = 0 \quad (2.36)$$

$$d_n = -1 \quad (2.37)$$

Using a change of variables, the Rachford-Rice equation can be rewritten:

$$h(a) = z_1 + \sum_{i=2}^{nc-1} \frac{z_i a}{d_i + a(1 + d_i)} - z_n a = 0 \quad (2.38)$$

Moreover, for $a \neq 0$, $c_1 \neq \theta$ and $c_n \neq \theta$, a new function was introduced

$$G(a) = \frac{a + 1}{a} h(a) \quad (2.39)$$

$\forall a > 0$, $G(a)$ is convex and monotonically decreasing within the negative flash window, and $G(a)G''(a) > 0 \quad \forall a_0 \in [0, a^*]$. With a_0 the initial a and a^* the root $G(a) = 0$. Furthermore, a new function was introduced:

$$L(a) = -ah(a) \quad (2.40)$$

$\forall a > 0$, $L(a)$ is convex within the negative flash window and that $L(a)L''(a) > 0 \quad \forall a_0 \in [a^*, \infty]$. With a_0 the initial a and a^* the root $L(a) = 0$. Based on h , G and L , an algorithm was proposed to switch between both pairs of convex functions(G and L). No bisection method is necessary in this case.

Moreover, in the procedure, θ is bounded ($\theta \in [\theta_{min}, \theta_{max}]$ [Leibovici and Neoschil [1992]]). In the paper, [Nichita and Leibovici [2013]], decrease the window by finding initial boundaries for the variable $a \in [a_L, a_R]$:

$$a_L = \frac{z_1}{1 - z_1} \quad a_R = \frac{1 - z_n}{z_n} \quad (2.41)$$

with a_L and a_R independent of the equilibrium constants.

2.3.1.2 Multiphase equilibrium calculations

2.3.1.2.1 Multiphase positive Rachford-Rice [Michelsen [1994]]

[Michelsen [1994]] proposed an elegant way to solve the Rachford-Rice equation for multiphase problems. He introduced a function proposed to replace the resolution of the nonlinear system (eq. 2.30) by the minimization of a convex function Q over the set of variables θ (written in terms of constant equilibrium ratios [Leibovici and Nichita [2008]])

$$Q(\theta) = \sum_{j=1}^{np} \theta_j - \sum_{i=1}^C z_i \ln E_i \quad (2.42)$$

subject to $\theta_m \geq 0$; $m = 1, np$, with

$$E_i = \sum_{k=1}^{np} \frac{\theta_k}{\phi_{ik}} \quad (2.43)$$

Algorithm 2.2 Algorithm FGH [Nichita and Leibovici [2013]]

```

Calculate the solution window,  $a_L$  and  $a_R$ 
Calculate the initial guess  $a = a_0$ 
NOTCONV = TRUE
while NOTCONV do
  Calculate  $h(a)$  and  $h'(a)$ 
  INWIND = FALSE
  if  $F'(a) \leq 0$  then
    Newton step  $a^{k+1} = a^k - h(a_k)/h'(a_k)$ 
    Test convergence
    if  $\|h(a)\| < \epsilon$ ,  $V^* = V(a^*)$  then
      NOTCONV = FALSE
      STOP
    end if
    if  $a^{k+1} \in [a_L, a_R]$  then
      INWIND = TRUE
    end if
  end if
  if INWIND = FALSE then
    if  $F(a) > 0$  then
       $a^{k+1} = a^k - G(a_k)/G'(a_k)$ 
    else
       $a^{k+1} = a^k - L(a_k)/L'(a_k)$ 
    end if
  end if
end while

```

A Newton procedure is used with an approximated line search method to guarantee convergence. The gradient is given by :

$$\frac{\partial Q}{\partial \theta_i} = 1 - \sum_{k=1}^{nc} \frac{z_k}{E_k \phi_{ki}} \quad (2.44)$$

and the Hessian is

$$\frac{\partial^2 Q}{\partial \theta_i \partial \theta_j} = \sum_{k=1}^{nc} \frac{z_k}{E_k^2 \phi_{ki} \phi_{kj}} \quad (2.45)$$

The minimum of Q subject to the constraints $\theta_j \geq 0$ is given by:

$$\begin{aligned} \frac{\partial Q}{\partial \theta_j} &= 0 & \theta_j &\geq 0 \\ \frac{\partial Q}{\partial \theta_j} &> 0 & \theta_j &= 0 \end{aligned} \quad (2.46)$$

A procedure of appearance/disappearance of the phases is used to keep all the molar phase fractions in the physical boundaries $([0, 1])$. It is based on (eq. 2.46). At each iteration, if a phase mole fraction θ_i goes out of the physical bound, it is removed from the set. At the convergence, a phase is 're-activated' if

$$\frac{\partial Q}{\partial \theta_j} \leq 0 \quad (2.47)$$

At the solution, \mathbf{x} can be calculated from

$$x_{ij} = \frac{z_i}{E_i \phi_{ij}} \quad \forall i = 1, nc; \quad \forall j = 1, np \quad (2.48)$$

Multiphase positive procedures enable converging to liquid/liquid systems. The equilibrium calculation generally converges to a liquid/liquid state by performing a three phase calculation which then converges to

the two liquid phases.

2.3.1.2.2 Multiphase negative Rachford-Rice

Negative flashes have been investigated by different authors. Starting with [Leibovici and Neoschil [1992]] and then [Leibovici and Nichita [2008]] who proposed the first negative multiphase Rachford-Rice, [Iranshahr et al. [2010b]] proposed a dichotomy method and showed the robustness with complex systems. More recently, [Okuno et al. [2010c]] proposed a faster algorithm using Newton combined with a line search procedure. Finally, [Yan and Stenby [2012]] extended [Michelsen [1994]] formulation to the negative flash. This is the one used in this thesis. Here is a description of their methodology.

A variable θ_i is taken as a dependent variable. It depends on the other molar fractions by means of:

$$\theta_{np} = 1 - \sum_{i=1}^{np-1} \theta_i \quad (2.49)$$

The same convex function as in (eq. 2.42) Q is used, which leads to:

$$Q(\theta) = 1 - \sum_{i=1}^C z_i \ln E_i \quad (2.50)$$

Minimizing Q with respect of θ corresponds to minimize the convex function:

$$Q'(\theta) = - \sum_{i=1}^C z_i \ln E_i \quad (2.51)$$

Once again, a Newton procedure is used with an approximated line search method to guarantee convergence. Using (eq. 2.43) The gradient vector is

$$\frac{\partial Q'}{\partial \theta_i} = - \sum_{k=1}^{nc} \frac{z_k}{E_k} \left(\frac{1}{\phi_{ki}} - \frac{1}{\phi_{knp}} \right) \quad (2.52)$$

and the Hessian is

$$\frac{\partial^2 Q'}{\partial \theta_i \partial \theta_j} = \sum_{k=1}^{nc} \frac{z_k}{E_k^2} \left(\frac{1}{\phi_{ki}} - \frac{1}{\phi_{knp}} \right) \left(\frac{1}{\phi_{kj}} - \frac{1}{\phi_{knp}} \right) \quad (2.53)$$

In the negative flash procedure, it is important to test if the mole fractions remain inside a valid bounded domain, that is to say $E_i \geq 0 \quad \forall i = 1, nc$. If a solution goes outside of the bound the step length is decreased to remain inside the domain. At the solution, \mathbf{x} can be calculated from

$$x_{ij} = \frac{z_i}{E_i \phi_{ij}} \quad \forall i = 1, nc; \quad \forall j = 1, np \quad (2.54)$$

2.3.2 Phase-split calculations

The SSI exposed in the stability analysis section has also been developed for phase-split calculations [Michelsen [1982a]]. It is generally used before switching to the Newton method. A switch procedure can be found in [Nghiem et al. [1983]].

The SSI update is given by:

$$\ln K_{ij}^{k+1} = \ln \phi_{i,np}^k - \ln \phi_{i,j}^k \quad \forall i = 1, nc \quad \forall j = 1, np - 1 \quad (2.55)$$

The error is estimated based on the Euclidean norm of the fugacity difference:

$$S = \sqrt{\sum_{i=1}^{nc} (f_{ij} - f_{i,np})^2} \quad \forall j = 1, np - 1 \quad (2.56)$$

Algorithm 2.3 Algorithm for the multiphase-split problem (with SSI)

From the initial guess \mathbf{X} from Stability analysis + previous multiphase flash solutions
 Compute the fugacities (eq. 1.88)
 Calculate θ , \mathbf{x} , from the multiphase Rachford-Rice.
while NOTCONV **do**
 Compute $\phi(\mathbf{x})$, \mathbf{f}
 Compute S (eq. 2.56)
 if $S > \epsilon$ **then**
 Update \mathbf{x} and θ from the Multiphase Rachford-Rice procedure
 end if
end while

2.4 General multiphase equilibrium algorithm

In reservoir simulation, in most of the cells, the conditions remain sensibly the same between two consecutive time steps. Therefore, the solution provided at the time n represents a good initial guess for the time $n + 1$. A multiphase flash can directly be used based on the previous time step initial guess, which allows skipping many stability analyses. The global algorithm used in this study is presented in fig. 2.6.

In the next chapter, new algorithms will be presented to improve robustness and efficiency of the both stability testing and multiphase-split problems.

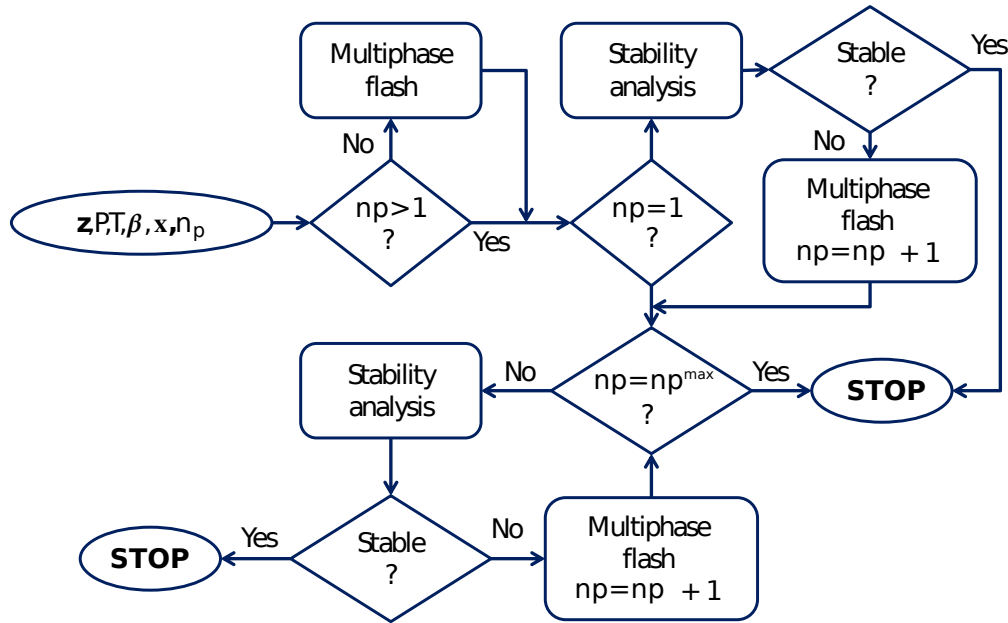


Figure 2.6: Multiphase flash calculation procedure

CHAPTER 3

Improvements in equilibrium calculations

Contents

3.1	Presentation of conventional methods	45
3.1.1	Stability testing	45
3.1.2	Multiphase flash calculations	47
3.2	Reduction methods	51
3.2.1	Presentation of existing reduction based methods	52
3.2.2	A new reduction method for the stability analysis problem	56
3.2.3	A comparison of conventional and reduction approaches for phase equilibrium calculations	66
3.2.4	A new reduction method for multiphase equilibrium calculations	80
3.3	Phase equilibrium calculations with quasi-Newton methods	93
3.3.1	Introduction	93
3.3.2	The BFGS quasi-Newton method	93
3.3.3	Proposed method	98
3.3.4	Results	100
3.4	Robust and efficient Trust-Region based stability analysis and multiphase flash calculations	108
3.4.1	Introduction	108
3.4.2	Trust-Region methods	109
3.4.3	Results	117
3.4.4	Conclusions	130
3.5	Multiphase flash, using $\ln K$ and phase mole fractions as independent variables	130
3.5.1	Introduction	130
3.5.2	New proposed method	130
3.5.3	Results	132

Phase equilibrium calculations (phase split and stability testing) are very important in chemical process and petroleum reservoir engineering. In process simulation and reservoir simulation, an important part of the computational time is spent by phase equilibrium calculations. Moreover, a failure in phase equilibrium routines may affect the results to a high extent or lead to the non-convergence of the simulation. In this context, it becomes essential that algorithms for phase equilibrium calculations are both efficient and robust; it is desirable to use for a given simulation type the most suitable algorithm. In petroleum reservoir compositional simulation, the number of components in the mixture is limited (typically less than a dozen) and a very large number of phase equilibrium calculations are performed in full-field scale simulations (in most or all the blocks

of the discretization network, at each time step and at each iteration level). On the contrary, in process simulations, the number of components can be very large (on the order of hundreds, with several feed mixed together).

In this chapter, improvements relative to equilibrium flash calculations will be described.

First, [Michelsen [1982a]] developed an algorithm which uses the Cholesky factorization to solve the linear system obtained with Newton taking the logarithm of the equilibrium constants as independent variables. This methodology will be extended to multiphase calculations, and will then be applied to [Haugen et al. [2011]]'s method where both the phase mole fractions and the equilibrium constant are updated at each iteration.

Also, the reduction methods introduced in multiphase equilibrium calculations enable to reduce the space of study from nc (the number of components) for conventional variables to M (with $M < nc$). A new reduction method will be proposed both for stability analysis and multiphase-split calculations, based on the multi-linear expression of the logarithm of fugacity coefficients.

Moreover, comparisons of different Newton methods based on conventional and reduction variables will be shown. The tests will focus on the computation time, but also on the condition number and the convergence path. All the problems (stability analysis, two-phase split and multiphase-split calculations) will be treated independently.

In an attempt to develop a general program, algorithms independent of the EOS will also be investigated. A BFGS (Broyden-Fletcher-Goldfarb-Shanno) approach applied with a new set of variables will be proposed for two-phase split calculations. Besides, a Trust-Region based method which could extend the Newton method to be applied to more difficult problems will be presented both for the stability testing and the multiphase-split problems.

3.1 Presentation of conventional methods

3.1.1 Stability testing

A mixture is stable if the TPD function is non-negative at all its stationary points; otherwise, the mixture splits into two or more phases. Michelsen proposed a modified TPD function [Michelsen [1982b]]

$$D^* = \left(1 - \sum_{i=1}^{nc} Y_i\right) + \sum_{i=1}^{nc} Y_i (\ln Y_i + \ln \phi_i(\mathbf{Y}) - d_i) \quad (3.1)$$

3.1.1.1 Mole numbers as independent variables

If Y_i are the independent variables, the elements of the gradient vector are

$$g_i = \frac{\partial D^*}{\partial Y_i} = \ln Y_i + \ln \phi_i - d_i \quad ; i = 1, nc \quad (3.2)$$

The iteration equation in the Newton method is

$$\mathbf{H} \Delta \mathbf{Y} = -\mathbf{g} \quad (3.3)$$

and the elements of the Hessian matrix \mathbf{H} are

$$H_{ij} = \frac{\partial^2 D^*}{\partial Y_i \partial Y_j} = \frac{\partial g_i}{\partial Y_j} = \frac{\delta_{ij}}{Y_i} + \frac{\partial \ln \phi_i}{\partial Y_j} \quad ; i, j = 1, nc \quad (3.4)$$

where

$$\frac{\partial \ln \phi_i}{\partial Y_j} = \frac{1}{Y_T} \left[\frac{\partial \ln \phi_i}{\partial x_j} - \sum_{k=1}^{nc} x_k \frac{\partial \ln \phi_i}{\partial x_k} \right] \quad (3.5)$$

and the partial derivative $\frac{\partial \ln \phi_i}{\partial x_j}$ is evaluated analytically for cubic EoS.

3.1.1.2 $\ln Y_i$ as independent variables

If the nonlinear system of equations $\mathbf{g}(\mathbf{Y}) = \mathbf{0}$ is solved with $\ln Y_i$ as independent variables, the Newton iterations are

$$\mathbf{J} \Delta \ln \mathbf{Y} = \mathbf{H} \mathbf{U}^{-1} \Delta \ln \mathbf{Y} = -\mathbf{g} \quad (3.6)$$

Where $\mathbf{J} = \mathbf{H} \mathbf{U}^{-1}$ is the Jacobian matrix with elements

$$J_{ij} = \frac{\partial g_i}{\partial \ln Y_j} = \delta_{ij} + \frac{\partial \ln \phi_i}{\partial \ln Y_j} \quad ; i, j = 1, nc \quad (3.7)$$

and

$$\Delta \ln \mathbf{Y} = \mathbf{U} \Delta \mathbf{Y} \quad (3.8)$$

In this thesis, the Newton method taking $\ln Y_i$ as independent variable will be called $\ln \mathbf{Y}$. The elements of the matrix \mathbf{U} and of its inverse are

$$U_{ij} = \frac{\partial \ln Y_i}{\partial Y_j} = \frac{\delta_{ij}}{Y_i} \quad ; i, j = 1, nc \quad (3.9)$$

and

$$U_{ij}^{-1} = \frac{\partial Y_i}{\partial \ln Y_j} = \delta_{ij} Y_i \quad ; i, j = 1, nc \quad (3.10)$$

The Jacobian matrix is non-symmetric, but a Cholesky decomposition of \mathbf{H} can be performed to solve for

$$\Delta \ln \mathbf{Y} = -\mathbf{U} \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{g} \quad (3.11)$$

The equilibrium constants for two-phase flash calculations are defined as: $K_i = y_i/x_i = \phi_{iL}(\mathbf{x})/\phi_{iV}(\mathbf{x})$, which corresponds to the equilibrium (zero gradient) condition, $f_{iL} = f_{iV}$. Equilibrium constant for phase stability, \overline{K}_i , can be defined in a similar way (that is, fulfilling (eq. 2.14), rather than the ratio of mole fractions. For liquid phase stability (vapor trial phase)

$$\overline{K}_i = \frac{Y_i}{z_i} = \frac{\phi_{iz}(\mathbf{z})}{\phi_i(\mathbf{Y})} \quad (3.12)$$

and for vapor phase stability (liquid trial phase)

$$\overline{K}_i = \frac{z_i}{Y_i} = \frac{\phi_i(\mathbf{Y})}{\phi_{iz}(\mathbf{z})} \quad (3.13)$$

Note that \overline{K}_i differs from K_i by a factor of Y_T . In the rest of this thesis, the formalism will be presented for liquid phase stability; the case of stability testing of a vapor phase is similar.

Using $\ln \overline{\mathbf{K}}$ instead of $\ln \mathbf{Y}$ as independent variables gives identical results, since $\Delta \ln \mathbf{Y} = \Delta \ln \overline{\mathbf{K}}$ and the gradient and the Hessian are the same for both sets of variables.

3.1.1.3 α as independent variables

In addition to phase stability calculations using the above sets of independent variables, Michelsen's variables ([Michelsen [1982b]]) $\alpha_i = 2\sqrt{Y_i}$ were also used to establish a comparison of conventional methods. In this

case the Newton equation is $\bar{\mathbf{H}}\Delta\boldsymbol{\alpha} = -\bar{\mathbf{g}}^*$, in which the gradient vector is

$$\bar{g}_i^* = \frac{\partial D^*}{\partial \alpha_i} = g_i \sqrt{Y_i} \quad (3.14)$$

and the Hessian matrix is

$$\bar{H}_{ij} = \frac{\partial^2 D^*}{\partial \alpha_i \partial \alpha_j} = \delta_{ij} + \sqrt{Y_i} \sqrt{Y_j} \frac{\partial \ln \phi_i}{\partial Y_j} + \frac{\delta_{ij} \bar{g}_i^*}{2} \quad (3.15)$$

in which the second term is of low effective rank (depending on the number of components with non-zero BIPs) and the last term vanishes at the solution.

Taking the mole numbers as independent variables, for small values of Y_i , large diagonal elements are obtained for the Hessian, which reveals an ill-scaled Hessian. On the contrary, with α_i , the Hessian obtained is well scaled, with diagonal elements close to the identity.

The relationship between $\bar{\mathbf{g}}$ and \mathbf{g} , is

$$\bar{\mathbf{g}} = \mathbf{U}^{-1/2} \mathbf{g} \quad (3.16)$$

and the relation between $\bar{\mathbf{H}}$ and \mathbf{H} is

$$\bar{\mathbf{H}} = \mathbf{U}^{-1/2} \mathbf{H} \mathbf{U}^{-1/2} + \bar{\mathbf{D}} \quad (3.17)$$

where

$$\bar{D}_{ij} = \frac{\delta_{ij} \bar{g}_i^*}{2} \quad (3.18)$$

The most difficult conditions for phase stability testing are those in the vicinity of the singularity, namely the stability test limit locus (STLL) [Hoteit and Firoozabadi [2006], Nichita and Petitfrere [2013], Rasmussen et al. [2006], Nichita et al. [2007c], Petitfrere and Nichita [2014a]], also called 'limit of parallel tangent plane' Whitson and Michelsen [1990] or limit of the 'shadow region' [Rasmussen et al. [2006]].

3.1.2 Multiphase flash calculations

The objective function for multiphase flash calculations is the dimensionless Gibbs free energy, given by

$$G = \sum_{i=1}^{nc} \sum_{k=1}^{np} n_{ik} \ln f_{ik} \quad (3.19)$$

The function G has to be minimized with respect to mole numbers. The material balance must be satisfied at each step, that is to say,

$$z_i = \sum_{k=1}^{np} \theta_k x_{ik}; \quad i = 1, nc \quad (3.20)$$

with

$$\sum_{p=1}^{np} \theta_p = 1 \quad (3.21)$$

For each component i , one phase is taken as reference phase (its mole numbers are dependent variables),

$$n_{iR} = z_i - \sum_{\substack{k=1 \\ k \neq R}}^{np} n_{ik}; i = 1, nc \quad (3.22)$$

3.1.2.1 Mole numbers as independent variables

The elements of the gradient vector \mathbf{g} are:

$$(g_i)_m = \frac{\partial G}{\partial n_{im}} = \ln K_{im} + \ln \phi_{im} - \ln \phi_{iR}; i = 1, nc; m = 1, np; m \neq R \quad (3.23)$$

where the equilibrium constants are $K_{im} = x_{im}/x_{iR} = \phi_{iR}/\phi_{im}; i = 1, nc; m = 1, np; m \neq R$ and the mole fractions are $x_{im} = n_{im}/\sum_{j=1}^{nc} n_{jm}; i = 1, nc; m = 1, np$

The stationary point condition of G is

$$\frac{\partial G}{\partial n_{im}} = 0; i = 1, nc; m = 1, np; m \neq R \quad (3.24)$$

The iteration equation for the Newton method is

$$\mathbf{H}\Delta\mathbf{n} = -\mathbf{g} \quad (3.25)$$

where $\mathbf{n} = (n_1^T, \dots, n_k^T, \dots, n_{np}^T)^T; k \neq R$, with $\mathbf{n}_k = (n_{1k}, \dots, n_{nc,k})^T; k = 1, np$. and \mathbf{H} is the Hessian matrix

$$(H_{ij})_{mk} = \frac{\partial^2 G}{\partial n_{im} \partial n_{jk}} = \delta_{mk} \left[\frac{1}{\theta_m} \left(\frac{\delta_{ij}}{x_{im}} - 1 \right) + \frac{\partial \ln \phi_{im}}{\partial n_{jm}} \right] + \left[\frac{1}{\theta_R} \left(\frac{\delta_{ij}}{x_{iR}} - 1 \right) + \frac{\partial \ln \phi_{iR}}{\partial n_{jR}} \right] \quad (3.26)$$

$\forall i, j = 1, nc; k, m = 1, np; k, m \neq R$

Let V stands for the vapor mole fraction (respectively L the liquid mole fraction). For two-phase cases, the matrix becomes

$$H_{ij} = \frac{1}{LV} \left(\frac{\delta_{ij} z_i}{x_i y_i} - 1 \right) + \frac{\partial \ln \phi_{iL}}{\partial n_{jL}} + \frac{\partial \ln \phi_{iV}}{\partial n_{jV}} \quad (3.27)$$

The matrix \mathbf{H} can be divided into two different terms:

$$\mathbf{H} = \mathbf{U} + \Phi \quad (3.28)$$

where

$$(\Phi_{ij})_{mk} = \frac{\partial^2 G_E}{\partial n_{im} \partial n_{jk}} = \delta_{mk} \frac{\partial \ln \phi_{im}}{\partial n_{jm}} + \frac{\partial \ln \phi_{iR}}{\partial n_{jR}} \quad (3.29)$$

and the elements of the matrix \mathbf{U} (which is always positive-semidefinite [Mehra et al. [1983], Ammar and Renon [1987]]) are for the two phase-split problem,

$$U_{ij} = \frac{\partial \ln K_i}{\partial n_{iV}} = \frac{1}{V(1-V)} \left(\frac{\delta_{ij}}{u_i} - 1 \right); \quad \forall i, j = 1, nc \quad (3.30)$$

with

$$u_i = \frac{x_i y_i}{z_i} \quad (3.31)$$

In this work, the extension to the multiphase-split problem is proposed:

$$(U_{ij})_{mk} = \frac{\ln K_{im}}{\partial n_{jk}} = \delta_{mk} \frac{1}{\theta_m} \left(\frac{\delta_{ij}}{x_{im}} - 1 \right) + \frac{1}{\theta_R} \left(\frac{\delta_{ij}}{x_{iR}} - 1 \right) \quad (3.32)$$

Newton iterations using mole numbers as independent variables are denoted in this work as **NLV**.

3.1.2.2 Improved mole numbers as independent variables

In the expression of the Hessian matrix, some off-diagonal elements contain the term $1/n_{jR}$. This means that if for certain components $n_{jR} \approx 0$, the off-diagonal elements may tend to $+\infty$ and the Hessian may be really ill-conditioned. It is then important to ensure that n_{jR} remains sufficiently large for all the components. [Michelsen [1982a]] proposed to take the reference phase (hence the dependent variables) different for each component. For a component i , the reference phase R is taken to be the one with the highest molar amount, that is, $n_{iR} = \max_{m=1,np} (n_{im})$. This requires further calculations but it gives a considerably better conditioned Hessian than **NLV**, better suited to both Newton and Trust-Region methods. This procedure is denoted here as **NLVM**.

3.1.2.3 $\ln K_{im}$ as independent variables

If $\ln K_{im}$ are taken as independent variables, the nonlinear system $\mathbf{g}(\mathbf{n}(\ln \mathbf{K})) = \mathbf{0}$ can be solved by means of a Newton procedure:

$$\mathbf{J} \Delta \ln \mathbf{K} = -\mathbf{g} \quad (3.33)$$

Where

$$(J_{ij})_{mk} = \frac{\partial^2 G}{\partial n_{im} \partial \ln K_{jk}}; \quad i, j = 1, nc; \quad k, m = 1, np \quad k, m \neq R \quad (3.34)$$

The matrix \mathbf{J} can be expressed as (from eq. 3.28):

$$\mathbf{J} = \mathbf{H} \mathbf{U}^{-1} = \mathbf{I} + \Phi \mathbf{U}^{-1} \quad (3.35)$$

3.1.2.3.1 Two-phase split problem (existing methods)

For the two phase-split problem, the inverse \mathbf{U}^{-1} is obtained by applying the Sherman-Morrison-Woodbury for a rank-one update of a diagonal matrix [Michelsen [1982a], [Ammar and Renon [1987]]

$$U_{ij}^{-1} = \frac{\partial n_{iV}}{\partial \ln K_j} = V L u_i \left(\delta_{ij} + \frac{u_j}{s} \right); \quad i, j = 1, nc \quad (3.36)$$

with $s = 1 - \sum_{i=1}^{nc} u_i$.

The Jacobian matrix is non-symmetric, but is the product of two symmetric matrices, \mathbf{H} and \mathbf{U}^{-1} . As proposed by [Michelsen [1982a]], the Hessian matrix can be decomposed as $\mathbf{H} = \mathbf{L} \mathbf{L}^T$, then the linear system is solved for $\Delta \mathbf{n}_{im} = -\mathbf{L}^{-T} \mathbf{L}^{-1}$, and finally $(\Delta \ln \mathbf{K} = \mathbf{U} \Delta \mathbf{n}_{im})$. [Michelsen [1982a]] used an \mathbf{LDL}^T rather than an \mathbf{LL}^T Cholesky decomposition). Using this procedure, the time required to solve the linear system is reduced by about a half (as compared with an LU decomposition of \mathbf{J}). This procedure will be called **lnK – chol.**

3.1.2.3.2 Multiphase-split problem (new proposed procedure)

For the multiphase-split problem, the dimensionality of the problem is $nc \times (np - 1)$; Matrices \mathbf{H} and \mathbf{J} have a block structure, with $(np - 1) \times (np - 1)$ blocks. The Jacobian matrix is non-symmetric, but one can take advantage of symmetry in constructing it, by calculating only about half of the elements of the symmetric matrices Φ and \mathbf{U}^{-1} . Unlike in two-phase equilibrium, it is very difficult to inverse the matrix \mathbf{U} analytically. The elements of \mathbf{U}^{-1} are

$$(U_{ij}^{-1})_{kp} = \frac{\partial n_{ik}}{\partial \ln K_{jp}}; \quad \forall i, j = 1, nc; \quad \forall k, p = 1, np; \quad k, p \neq R \quad (3.37)$$

Using the chain rule

$$\frac{\partial n_{ik}}{\partial \ln K_{jp}} = \left(\frac{\partial n_{ik}}{\partial \ln K_{jp}} \right)_{\theta} + \sum_{\substack{m=1 \\ m \neq R}}^{nc} \left(\frac{\partial n_{ik}}{\partial \theta_m} \right)_{\mathbf{K}, \theta_l \neq m} \left(\frac{\partial \theta_m}{\partial \ln K_{jp}} \right); \quad (3.38)$$

$$i, j = 1, nc; \quad k, p = 1, np; \quad k, p \neq R \quad (3.39)$$

The partial derivatives in the above equation are

$$\left(\frac{\partial n_{ik}}{\partial \ln K_{jp}} \right)_{\theta} = \delta_{ij} n_{ik} \left(\delta_{kp} - \frac{n_{ip}}{z_i} \right); \quad i, j = 1, nc; \quad k, p = 1, np; \quad k, p \neq R \quad (3.40)$$

and

$$\left(\frac{\partial n_{ik}}{\partial \theta_m} \right)_{\mathbf{K}, \theta_s \neq m} = \delta_{kp} x_{ik} - n_{ik} w_{ip}; \quad i, j = 1, nc; \quad k, p = 1, np; \quad k, p \neq R \quad (3.41)$$

where

$$w_{ik} = \frac{x_{ik} - x_{iR}}{z_i}; \quad k = 1, np, \quad k \neq R \quad (3.42)$$

and the partial derivatives $\partial \theta_m / \partial \ln K_{jp}$ are obtained by differentiating the Rachford-Rice equations, (eq. 2.30)

$$\sum_{\substack{m=1 \\ m \neq R}}^{nc} \left(\frac{\partial R_k}{\partial \theta_m} \right)_{\mathbf{K}, \theta_l \neq m} \left(\frac{\partial \theta_m}{\partial \ln K_{jp}} \right) = - \left(\frac{\partial R_k}{\partial \ln K_{jp}} \right)_{\theta}; \quad j = 1, nc; \quad k, p = 1, np; \quad k, p \neq R \quad (3.43)$$

The linear system of equations is

$$\mathbf{S} \left(\frac{\partial \theta}{\partial \ln \mathbf{K}} \right)_{jp} = (\mathbf{f})_{jp}; \quad j = 1, nc; \quad p = 1, np; \quad p \neq R \quad (3.44)$$

The $(np - 1) \times (np - 1)$ matrix \mathbf{S} is symmetric (since it is a Hessian, corresponding to the minimization of Michelsen's Q function (eq. 2.42) with respect to $np - 1$ θ variables). The matrix \mathbf{S} and its decomposition are already available from the last iteration in the resolution of the Rachford-Rice system of equations; its elements are

$$S_{km} = - \left(\frac{\partial R_k}{\partial \theta_m} \right)_{\mathbf{K}, \theta_s \neq m} = \sum_{i=1}^{nc} z_i w_{ik} w_{im}; \quad k, m = 1, np; \quad k, m \neq R \quad (3.45)$$

and the elements of the RHS vector in (eq. 3.44) are

$$(f_k)_{jp} = \left(\frac{\partial R_k}{\partial \ln K_{jp}} \right)_{\theta} = x_{jR} (\delta_{kp} - \theta_p w_{jk}); \quad j = 1, nc; \quad k, p = 1, np; \quad k, p \neq R \quad (3.46)$$

Moreover, from the proposed expression of the matrix \mathbf{U} for multiphase problems (eq. 3.32), the procedure **lnK – chol.** (from [Michelsen [1982a]], introduced in two phases) can be extended to multiphase calculations. The same procedure can be used for stability analysis with $\ln \mathbf{Y}$ variables and will be called **lnY – Chol.** in this thesis).

To the best of our knowledge, no reports can be found in the literature on how to use symmetry in multiphase flash calculations with $\ln \mathbf{K}$ as independent variables.

The Jacobian matrix \mathbf{J} in **Newton – lnK** iterations is better scaled than the Hessian for **NLVM** iterations; the linear system (eq. 3.33) is better conditioned than the system (eq. 3.25).

3.2 Reduction methods

In order to reduce the computer time in equilibrium calculations, an attractive alternative to the widely used technique of lumping individual components into pseudo-components is given by the so-called reduction methods. In the reduction method the dimensionality of the phase equilibrium problem can be significantly lowered, and the number of independent variables does not depend on the number of components in the mixture, but only on the number of non-zero binary interaction parameters (BIPs) in the equation of state (EoS): the use of a detailed fluid composition is thus allowed. The reduction methods are particularly efficient for mixtures with many components and few non-zero BIPs (such as naturally occurring hydrocarbon systems or certain feeds in refining problems involving several different crude oils that could bring the total number of fractions to the order of hundreds).

The first reduction method for flash calculations was proposed by [Michelsen [1986]]. Using EoS parameters as independent variables, together with the fraction of one equilibrium phase instead of conventional variables used in the compositional space, he obtained a nonlinear system of only three equations, whatever the number of components in the mixture. Even though this method is not applicable in practical cases (being limited to all BIPs equal to zero), Michelsen’s simplified flash formulation is very important since it showed for the first time that the solution phase equilibrium problems can be sought in a space of reduced dimensionality. A few years later, [Hendriks [1988]] presented the ‘Reduction Theorem’, stating the circumstances under which the dimensionality of several phase equilibrium problems can be reduced; he explained that various formulations apparently very different reflect a common underlying mathematical structure. In any reduction method for phase equilibrium calculations with two-parameter cubic EoS, there are M reduction parameters [Hendriks [1988]], $\mathbf{Q} = (Q_1, Q_2, \dots, Q_M)^T$, given by

$$Q_\alpha = \sum_{i=1}^{nc} q_{\alpha i} x_i \quad ; \forall \alpha = 1, M \quad (3.47)$$

Where $q_{\alpha i}$ are the elements of the reduction matrix [Hendriks [1988]].

Different sets of reduced variables/reduction procedures have been proposed [Hendriks and van Bergen [1992], Nichita and Minescu [2004], Li and Johns [2006], Nichita and Graciaa [2011], Gaganis and Varotsis [2013]], and a number of papers have presented calculation procedures for two-phase flash calculations [Hendriks and van Bergen [1992], Nichita and Minescu [2004], Li and Johns [2006], Nichita and Graciaa [2011], Jensen and Fredenslund [1987], Kaul and Thrasher [1996], Pan and Firoozabadi [2003], Nichita et al. [2007a]], phase stability analysis [Firoozabadi and Pan [2002], Nichita et al. [2002], Nichita et al. [2006a], Hoteit and Firoozabadi [2006]], multiphase flash calculations [Nichita et al. [2006a], Okuno et al. [2010a]], critical point calculation [Nichita [2005], Nichita [2006b]], phase envelope construction [Nichita [2008]], or the application of the reduction concept to solve related problems, such as pseudo-component delumping [Nichita and Leibovici [2006]]. The implementation of reduction methods in compositional reservoir simulators has also been evaluated for IMPEC simulator [Okuno et al. [2010a], Okuno et al. [2010b]], in fully implicit [Pan and

Tchelepi [2011a] and Pan and Tchelepi [2011b]), and is also implemented in INTERSECT. Two- parameter cubic equations of state are usually used, but it should be noticed that the applicability of the method is restricted by the mixing rules and not by the form of the equation of state (EoS parameters for mixtures must be linear forms or decomposable into linear forms [Hendriks [1988]]).

In this work, a presentation of the reduction parameters will be made in a first part. Then, a new reduction method will be presented for stability analysis and multiphase flash calculations (based on [Nichita and Graciaa [2011]] methodology (who developed an algorithm for two phases)). Comparisons will be made with previous variables (conventional and previous reductions) for the different problems.

3.2.1 Presentation of existing reduction based methods

3.2.1.1 Reduction parameters

Any procedure [Hendriks and van Bergen [1992], Nichita and Minescu [2004], Li and Johns [2006], Nichita and Graciaa [2011], Gaganis and Varotsis [2013]] to decompose the quadratic form A (eq. 1.72, eq. 1.73) can be used to obtain the reduction parameters and the reduction matrix [Hendriks [1988]]. The spectral decomposition [Hendriks and van Bergen [1992]] of the matrix $\bar{\mathbf{C}} = \{1 - C_{ij}\}_{i,j=1,nc}$ (usually of rank $r < nc$),

$$1 - C_{ij} = \sum_{\alpha=1}^m \lambda_{\alpha} q'_{\alpha i} q'_{\alpha j} \quad (3.48)$$

where λ_{α} are the $m = r$ non-zero eigenvalues, and $q'_{\alpha i}; \alpha = 1, m \quad i = 1, nc$ are the elements of the corresponding eigenvectors. The elements of the reduction matrix are related to eigenvectors by $q_{\alpha i} = q'_{\alpha i} \sqrt{A_i}; \quad \alpha = 1, m \quad i = 1, nc$. From (eq. 1.74), A_{ij} can be written as

$$A_{ij} = \sum_{\alpha=1}^m \lambda_{\alpha} q_{\alpha i} q_{\alpha j} \quad (3.49)$$

and, using (eq. 3.47) for $\alpha = 1, m$, (eq. 3.49 and eq. 1.72, eq. 1.73), we obtain $A(\mathbf{Q})$

$$A = \sum_{i=1}^{nc} \sum_{j=1}^{nc} x_i x_j \left(\sum_{\alpha=1}^m \lambda_{\alpha} q_{\alpha i} q_{\alpha j} \right) = \sum_{\alpha=1}^m \lambda_{\alpha} \left(\sum_{i=1}^{nc} q_{\alpha i} x_i \right) \left(\sum_{j=1}^{nc} q_{\alpha j} x_j \right) = \sum_{\alpha=1}^m \lambda_{\alpha} Q_{\alpha}^2 \quad (3.50)$$

and $\psi_i(\mathbf{Q})$

$$\psi_i = \sum_{i=1}^{nc} x_i \left(\sum_{\alpha=1}^m \lambda_{\alpha} q_{\alpha i} q_{\alpha j} \right) = \sum_{\alpha=1}^m \lambda_{\alpha} q_{\alpha i} \left(\sum_{j=1}^{nc} q_{\alpha j} x_j \right) = \sum_{\alpha=1}^m \lambda_{\alpha} q_{\alpha i} Q_{\alpha}; \quad \forall i = 1, nc \quad (3.51)$$

is obtained from (eq. 3.47, eq. 3.49 and eq. 1.89).

The last reduction parameter is (eq. 3.47 for $\alpha = m + 1 = M$)

$$Q_M \equiv B = \sum_{i=1}^{nc} B_i x_i \quad (3.52)$$

\mathbf{h} variables were introduced by [Nichita and Minescu [2004]].

By introducing (eq. 3.50, eq. 3.51 and eq. 3.52) in (eq. 1.88), after some arrangement the fugacity coefficients are obtained as:

$$\ln \phi_i = \sum_{\alpha=1}^{M+1} q_{\alpha i} h_{\alpha}; \quad i = 1, nc \quad (3.53)$$

where $q_{\alpha i}$; $\alpha = 1, M+1$; $i = 1, nc$ are the elements of the reduction matrix [Hendriks [1988]], with $q_{Mi} = B_i$ and $q_{M+1,i} = 1$.

The coefficients h_α (eq. 3.53) are function only of reduction parameters, $h_\alpha = h_\alpha(\mathbf{Q})$

$$h_\alpha(\mathbf{Q}) = -\frac{2\lambda_\alpha Q_\alpha}{\Delta Q_M} \ln \left(\frac{Z(\mathbf{Q}) + \delta_1 Q_M}{Z(\mathbf{Q}) + \delta_2 Q_M} \right); \quad \alpha = 1, m \quad (3.54)$$

$$h_M(\mathbf{Q}) = \frac{Z(\mathbf{Q}) - 1}{Q_M} + \frac{\sum_{\alpha=1}^m \lambda_\alpha Q_\alpha^2}{\Delta Q_M^2} \ln \left(\frac{Z(\mathbf{Q}) + \delta_1 Q_M}{Z(\mathbf{Q}) + \delta_2 Q_M} \right) \quad (3.55)$$

$$h_{M+1}(\mathbf{Q}) = -\ln(Z(\mathbf{Q}) - Q_M) \quad (3.56)$$

If all BIPs are zero, $m = 1$ ($M = 2$), $q_{1i} = \sqrt{A_i}$, $q_{2i} = B_i$ and $h_\alpha = h_\alpha(A, B)$; $\alpha = 1, 2$.

In matrix form (eq. 3.53) reads

$$\ln \phi = \mathbf{C}^T \mathbf{h} \quad (3.57)$$

where the reduction matrix \mathbf{C} is an $[(M+1) \times nc]$ matrix; it also relates reduction parameters to mole fractions ($\mathbf{Q} = \mathbf{C}\mathbf{x}$; (eq. 3.47)).

In terms of reduction parameters, the excess part of the Gibbs free energy is

$$g_E(\mathbf{Q}) = \sum_{\alpha=1}^{M+1} Q_\alpha h_\alpha(\mathbf{Q}) \quad (3.58)$$

with $Q_{M+1} = 1$ for convenience, and its derivatives with respect to the reduction parameters are

$$\frac{\partial g_E}{\partial Q_\alpha} = h_\alpha \quad (3.59)$$

From equations (eq. 3.53, eq. 3.54 to eq. 3.56 and eq. 3.58), fugacity coefficients and molar excess Gibbs free energy can be viewed as functions of \mathbf{h} , as well as functions of reduction parameters \mathbf{Q} via \mathbf{h} , and not on phase compositions as in conventional methods.

For two phase-split problems, [Nichita and Graciaa [2011]] developed a new algorithm based on the exposed \mathbf{h} variables:

(eq. 3.53) can be expressed for each phase of a multiphase mixture:

$$\ln \phi_{ik} = \sum_{\alpha=1}^{M+1} q_{\alpha i} h_{k\alpha}; \quad i = 1, nc \quad (3.60)$$

Using [3.60], the equilibrium constants $\ln K_i = \ln \phi_{iL} - \ln \phi_{iV}$ are

$$\ln K_i = \sum_{\alpha=1}^{M+1} q_{\alpha i} h_\alpha; \quad i = 1, nc \quad (3.61)$$

or

$$\ln \mathbf{K} = \mathbf{C}^T \mathbf{h} \quad (3.62)$$

with

$$h_\alpha = h_{L\alpha} - h_{V\alpha} \quad (3.63)$$

In reduction methods for phase equilibrium calculations, either the reduction parameters \mathbf{Q} [Hendriks

and van Bergen [1992], Pan and Firoozabadi [2003], Nichita and Minescu [2004], Li and Johns [2006] and Nichita et al. [2007b]] (the algorithms are denoted **Q – red.** through this thesis) or the coefficients **h** from (eq. 3.54, eq. 3.55, eq. 3.56) [Kaul and Thrasher [1996], Pan and Firoozabadi [2003], Nichita and Graciaa [2011]] (which are Lagrange multipliers; algorithms are denoted **h – red.**) can be chosen as independent variables.

3.2.1.2 Two phase-split calculations in the reduction approach

3.2.1.2.1 Reduction parameters as independent variables

In [Nichita and Minescu [2004]], a methodology for the two phase-split problem was developed. $M + 1$ independant variables were used whose corresponding $M + 1$ error equations are

$$e_\alpha = \sum_{i=1}^{nc} q_{\alpha i} y_i - Q_{V,\alpha} = 0 \quad \forall \alpha = 1, M \quad (3.64)$$

$$e_{M+1} = \sum_{i=1}^{nc} (y_i - x_i) = 0 \quad (3.65)$$

A Newton procedure can be used to update the variables. The Newton iteration equation is

$$\mathbf{J}^{R-Q} \Delta \zeta = -\mathbf{e} \quad (3.66)$$

where $\zeta = [Q_{V,1}, \dots, Q_{V,M}, V]^T$ is the vector of independent variables (QBV iterations [Nichita et al. [2007b]], used if the mixture is predominantly liquid; otherwise QBL iterations are used [Nichita et al. [2007b]], and $\zeta = [Q_{L,1}, \dots, Q_{L,M}, L]^T$).

The elements of the Jacobian matrix are

$$J_{\alpha\beta}^{R-Q} = \frac{\partial e_\alpha}{\partial Q_\beta}; \quad \alpha, \beta = 1, M + 1 \quad (3.67)$$

All required partial derivatives are given in [Nichita et al. [2007b]]. The Jacobian matrix is non-symmetric.

3.2.1.2.2 Lagrange multipliers as independent variables

In the reduction method of [Nichita and Graciaa [2011]], the idea is to consider $\ln \phi_i = \ln \phi_i(\mathbf{h}(\mathbf{Q})) = \ln \phi_i(\mathbf{h})$, which suggests to use $\mathbf{h} = [h_1, \dots, h_{M+1}]$ as the vector of independent variables (which are not bounded) and \mathbf{Q} as dependent variables, according to the following sequence:

Given $\mathbf{h} \rightarrow \ln \mathbf{K}$ (eq. 3.61) $\rightarrow V$ (Rachford-Rice) $\rightarrow \mathbf{x} \rightarrow \mathbf{Q}$ (eq. 3.47) \rightarrow New \mathbf{h} (eq. 3.54, eq. 3.55, eq. 3.56).

In this case the error equations are

$$e_\alpha = h_\alpha + h_{V\alpha} - h_{L\alpha} = 0; \quad \forall \alpha = 1, M + 1 \quad (3.68)$$

and the elements of the Jacobian matrix are

$$J_{\alpha\beta} = \frac{\partial e_\alpha}{\partial h_\beta} = \delta_{\alpha\beta} + \frac{\partial h_{V\alpha}}{\partial h_\beta} - \frac{\partial h_{L\alpha}}{\partial h_\beta} \quad (3.69)$$

where

$$\frac{\partial h_{k\alpha}}{\partial h_\beta} = \sum_{\gamma=1}^M \frac{\partial h_{k\alpha}}{\partial Q_{k\gamma}} \frac{\partial Q_{k\gamma}}{\partial h_\beta}; \quad \forall \alpha, \beta = 1, M+1, \quad k = L, V \quad (3.70)$$

The partial derivatives required in (eq. 3.70) can be found in [Nichita and Graciaa [2011]].

Previously, \mathbf{h} have been used as independent variables by [Kaul and Thrasher [1996]] (for all BIPs equal to zero) and [Pan and Firoozabadi [2003]]. [Nichita and Graciaa [2011]] reduction method was used by [Michelsen et al. [2013b], Haugen and Beckner [2013], Gorucu and Johns [2013], Gorucu and Johns [2014] and Gaganis and Varotsis [2014]]. In [Nichita and Graciaa [2011]] it was observed (but not proved) that $\mathbf{h} - \mathbf{red}$. Newton iterations have the same convergence path as the conventional Newton iterations with $\ln \mathbf{K}_i$ as variables. A proof of this statement is given in Appendix E. In Appendix D, it is shown that the elements of \mathbf{h} are Lagrange multipliers associated to a constrained minimization of the Gibbs free energy with respect to component mole numbers and subject to linear equality constraints, and that the error equations and the Jacobian matrix in [Nichita and Graciaa [2011]] are related to this constrained minimization.

3.2.1.3 Phase stability in the reduction approach

3.2.1.3.1 Reduction parameters as independent variables

[Nichita et al. [2006a]] showed that the tangent plane distance (eq. 3.1) could also be considered with the reduction variables. They showed that (eq. 3.2) was equivalent to the system of M equations, with M unknowns ($\mathbf{Q} = [Q_1, \dots, Q_M]^T$):

$$F_\alpha(\mathbf{Q}) = \sum_{i=1}^{nc} q_{\alpha i} Y_i - Q_\alpha \sum_{i=1}^{nc} Y_i = 0 \quad \alpha = 1, M \quad (3.71)$$

The Newton iteration equation is:

$$\mathbf{J}^{R-Q} \Delta \mathbf{Q} = -\mathbf{F} \quad (3.72)$$

From (eq. 2.14),

$$\frac{\partial Y_i}{\partial Q_\beta} = -Y_i \frac{\partial \ln \phi_i(\mathbf{Q})}{\partial Q_\beta}; \quad i = 1, nc \quad \beta = 1, M \quad (3.73)$$

thus,

$$J_{\alpha\beta}^{R-Q} = \frac{\partial F_\alpha}{\partial Q_\beta} = \delta_{\alpha\beta} \sum_{i=1}^{nc} Y_i + Q_\alpha \sum_{i=1}^{nc} \frac{\partial Y_i}{\partial Q_\beta} - \sum_{i=1}^{nc} q_{\alpha i} \frac{\partial Y_i}{\partial Q_\beta} \quad \alpha, \beta = 1, M \quad (3.74)$$

The variables \mathbf{Q} are bounded by $\min_{i=1, nc} q_{\alpha i}$; $\alpha = 1, M$. Only Newton iterations can be performed; if a variable hits its bounds or if the TPD function increases between two subsequent Newton iterations, iterations are reverted to SSI [Hoteit and Firoozabadi [2006]].

For the stability analysis problem, the procedure has been developed in this thesis and will be exposed in the next session.

3.2.1.4 Pseudo-reduces methods

[Michelsen et al. [2013b]] proposed recently a method called here 'pseudo-reduced'. The idea is to build a system with a reduced dimensionality using some reduction principles, but starting from the gradient vector and the Hessian matrix constructed in the compositional space.

For flash calculations, the gradient vector with respect to \mathbf{h} is

$$g_\alpha^* = \frac{\partial G}{\partial h_\alpha} = \sum_{i=1}^{nc} \frac{\partial G}{\partial n_{iv}} \frac{\partial n_{iv}}{\partial h_\alpha}; \quad \alpha = 1, M+1 \quad (3.75)$$

Or

$$\mathbf{g}^* = \mathbf{T} \mathbf{g} \quad (3.76)$$

where \mathbf{T} is an $(M+1) \times nc$ matrix with elements:

$$T_{\alpha i} = \frac{\partial n_{iv}}{\partial h_\alpha} = \sum_{i=1}^{nc} \frac{\partial n_{iv}}{\partial \ln K_i} \frac{\partial \ln K_i}{\partial h_\alpha}; \quad \alpha = 1, M+1, i = 1, nc \quad (3.77)$$

or $\mathbf{T}^T = \mathbf{U}^{-1} \mathbf{C}^T$, or

$$\mathbf{T} = \mathbf{C} \mathbf{U}^{-1} \quad (3.78)$$

The Hessian matrix with respect to \mathbf{h} is

$$\mathbf{H}^* = \mathbf{T} \mathbf{H} \mathbf{T}^T \quad (3.79)$$

and the Newton iteration equation in the pseudo-reduced method is

$$\mathbf{H}^* \Delta \mathbf{h} = -\overline{\mathbf{g}^*} \quad (3.80)$$

Michelsen's procedure consists in calculating a new Hessian and a new gradient starting from those evaluated in the conventional method (taking advantage of symmetry properties), then solving a smaller linear system to update \mathbf{h} , and finally calculating updated $\ln K_i$ from (eq. 3.61). The method can be implemented with minimal modifications in existing codes using conventional methods. The spectral decomposition is performed only once for all at the beginning of calculations. If Newton iterations are used, the convergence path of the pseudo-reduced method follows naturally the one of the conventional method using $\ln \mathbf{K}$ as independent variables.

A similar pseudo-reduced method is proposed by [Michelsen et al. [2013b]] for stability testing. It is shown in Appendix G that Michelsen's (eq. 3.80) can be obtained in a different way directly from the Newton iteration equation with Y_i as independent variables, by using a different transformation matrix and without neglecting any terms. The pseudo-reduced method for phase stability has the same convergence path as the conventional method with $\ln Y_i$ as independent variables.

3.2.2 A new reduction method for the stability analysis problem

3.2.2.1 Introduction

Phase stability testing is an important sub-problem in phase equilibrium calculations; it assesses the state of a system, (decides whether a system of given composition \mathbf{z} is stable at given pressure and temperature conditions or if it splits into two or more stable phases). Phase stability consists in finding either all stationary points or only the global minimum of the Gibbs tangent plane distance (*TPD*) function [Michelsen [1982b]]. The results of phase stability analysis are of high importance for initialization of multiphase flash calculations and for validating the results of a phase split [Baker et al. [1982]].

The *TPD* surface is non-convex and often highly nonlinear, and phase stability calculations at certain conditions are rather difficult [Hoteit and Firoozabadi [2006], Nichita et al. [2007c]], mainly in the vicinity of the stability test limit locus (STLL), also denoted limit of parallel tangent plane [Whitson and Michelsen

[1990]] or shadow curve [Rasmussen et al. [2006]]. Besides, compositions of many effluents in chemical and petroleum engineering are characterized by a (very) large number of components, resulting in time-consuming simulations. In petroleum reservoir compositional simulation, even if the number of components in the mixture is limited (typically less than ten), a very large number of phase equilibrium calculations are performed in full-field scale simulations. Depending on the structure of the simulator, flash calculations may be part of a larger problem, but stability testing must be carried out at any time step in any grid block likely to experience a phase split; even if stability testing may be bypassed according to some simple criteria [Rasmussen et al. [2006]], a very large number of calls for the stability routine is expected during a field-scale simulation run.

Recently, [Nichita and Graciaa [2011]] proposed a new reduction method for two-phase flash calculations; unlike most of the previous reduction methods, the independent variables are not the reduction parameters and the approach corresponds to the minimization of the Gibbs free energy [Michelsen et al. [2013b], Petitfrere and Nichita [2014b]], which facilitates the comparison with conventional methods (as recently pointed out by [Michelsen et al. [2013b]]). Moreover, the number of iterations is the same as in conventional flash calculations using the natural logarithm of equilibrium constants as independent variables, and the iteration path is related to that in the conventional method. The present chapter presents a calculation procedure for phase stability analysis based on the reduction variables proposed in [Nichita and Graciaa [2011]], which exhibits some very interesting features.

3.2.2.2 Proposed reduction method

One can consider the dependence $\ln \phi_i = \ln \phi_i(\mathbf{Q})$, which leads to the reduced stability formulation of [Nichita et al. [2007b]], or to consider $\ln \phi_i = \ln \phi_i(\mathbf{h}(\mathbf{Q})) = \ln \phi_i(\mathbf{h})$, which suggests to use $\mathbf{h} = (h_1, \dots, h_{M+1})^T$ as the vector of independent variables (in a similar way to the recently reduction method for flash calculations [Nichita and Graciaa [2011]]), rather than the reduction parameters. The key equation is (eq. 3.53): the vector of natural logarithms of fugacity coefficients is simply the product between the transposed of the reduction matrix and the vector \mathbf{h} . On the other hand, h_α depend only on reduction parameters, directly and via the compressibility factors, i.e., $h_\alpha = h_\alpha(\mathbf{Q}, Z(\mathbf{Q})) = h_\alpha(\mathbf{Q})$; $\alpha = 1, M+1$ (eq. 3.54, eq. 3.55, eq. 3.56). This suggests the possibility of updating \mathbf{h} according to the following iterative sequence:

Given $\mathbf{h} \rightarrow \ln \phi$ (eq. 3.53) $\rightarrow \mathbf{Y}$ (eq. 2.14) $\rightarrow \mathbf{x}$ (eq. 2.9) $\rightarrow \mathbf{Q}$ (eq. 3.47) \rightarrow New \mathbf{h} (eq. 3.54, eq. 3.55, eq. 3.56)

Note that the new independent variables \mathbf{h} are unbounded, unlike the reduction parameters \mathbf{Q} (taken as independent variables in [Nichita et al. [2007a], Hoteit and Firoozabadi [2006]]) which are bounded. This have a favorable effect on the convergence behavior and finally on the robustness of the method.

The error equations in the proposed method are

$$e_\alpha \equiv h_\alpha - h_\alpha(\mathbf{Q}) = 0; \quad \alpha = 1, M+1 \quad (3.81)$$

and the elements of the Jacobian matrix are

$$J_{\alpha\beta}^R = \frac{\partial e_\alpha}{\partial h_\beta} = \delta_{\alpha\beta} - \frac{\partial h_\alpha(\mathbf{Q})}{\partial h_\beta}; \quad \alpha, \beta = 1, M+1 \quad (3.82)$$

The resulting linear system of equations in the Newton method is

$$\mathbf{J}^{\mathbf{R}^{(\nu)}} \Delta \mathbf{h}^{(\nu+1)} = -\mathbf{e}^{(\nu)} \quad (3.83)$$

where $\Delta \mathbf{h}^{(\nu+1)} = \mathbf{h}^{(\nu+1)} - \mathbf{h}^{(\nu)}$, ν is the iteration level and $\mathbf{e} = (e_1, \dots, e_{M+1})^T$

The partial derivatives in (eq. 3.82) are calculated as [Nichita and Graciaa [2011]]

$$\frac{\partial h_\alpha(\mathbf{Q})}{\partial h_\beta} = \sum_{\gamma=1}^M \frac{\partial h_\alpha}{\partial Q_\gamma} \frac{\partial Q_\gamma}{\partial h_\beta}; \quad \alpha, \beta = 1, M+1 \quad (3.84)$$

where

$$\frac{\partial h_\alpha}{\partial Q_\gamma} = \left(\frac{\partial h_\alpha}{\partial Q_\gamma} \right)_{Z, \mathbf{Q}_{\mu \neq \gamma}} + \left(\frac{\partial h_\alpha}{\partial Z} \right)_{\mathbf{Q}} \frac{\partial Z}{\partial Q_\gamma}; \quad \alpha = 1, M+1; \quad \gamma = 1, M \quad (3.85)$$

and

$$\frac{\partial Q_\gamma}{\partial h_\beta} = \sum_{i=1}^{nc} q_{\gamma i} \frac{\partial x_i}{\partial h_\beta}; \quad \beta = 1, M+1; \quad \gamma = 1, M \quad (3.86)$$

The partial derivatives required in (eq. 3.84, eq. 3.85 and eq. 3.86) are given in Appendix A. The partial derivatives of mole fractions with respect to the primary variables are

$$\frac{\partial x_i}{\partial h_\beta} = \frac{1}{Y_T} \left(\frac{\partial Y_i}{\partial h_\beta} - x_i \sum_{k=1}^{nc} \frac{\partial Y_k}{\partial h_\beta} \right); \quad i = 1, nc; \quad \beta = 1, M+1 \quad (3.87)$$

Where

$$\frac{\partial Y_i}{\partial h_\beta} = -q_{\beta i} Y_i \quad (3.88)$$

giving

$$\frac{\partial x_i}{\partial h_\beta} = (Q_\beta - q_{\beta i}) x_i \quad (3.89)$$

and

$$\frac{\partial Q_\gamma}{\partial h_\beta} = Q_\beta Q_\gamma - \sum_{i=1}^{nc} q_{\beta i} q_{\gamma i} x_i \quad (3.90)$$

Since $q_{M+1, i} = 1$,

$$\frac{\partial x_i}{\partial h_{M+1}} = 0 \quad (3.91)$$

and

$$\frac{\partial h_\alpha}{\partial h_{M+1}} = \delta_{\gamma, M+1}; \quad \alpha = 1, M+1 \quad (3.92)$$

the last column in the Jacobian matrix is $(0, \dots, 0, 1)^T$.

Thus, it is convenient to solve the linear system of equations of dimensionality $M \times M$

$$\sum_{\alpha=1}^M \mathbf{J}_{\alpha\beta}^{\mathbf{R}} \Delta h_\alpha = -e_\alpha; \quad \alpha = 1, M \quad (3.93)$$

and the last variable is updated at each iteration from

$$\Delta h_{M+1} = -e_{M+1} - \sum_{\alpha=1}^M \frac{\partial e_{M+1}}{\partial h_\alpha} \Delta h_\alpha \quad (3.94)$$

with the cost of a simple summation over M .

The proposed method can be also formulated in terms of equilibrium constants for phase stability \overline{K}_i

$$\ln \overline{K}_i = \sum_{\alpha=1}^{M+1} q_{\alpha i} \overline{h}_{\alpha}; \quad i = 1, nc \quad (3.95)$$

where

$$\overline{h}_{\alpha}(\mathbf{Q}) = h_{\alpha}(\mathbf{Q}) - h_{\alpha z}(\mathbf{Q}_z); \quad \alpha = 1, M + 1 \quad (3.96)$$

The second term corresponds to feed composition.

Using $\overline{\mathbf{h}}$ instead of \mathbf{h} as independent variables is the same thing, since $\Delta \overline{\mathbf{h}} = \Delta \mathbf{h}$ and gradients and Hessians are identical.

It is shown in Appendix B that the proposed method for phase stability corresponds to a constrained minimization of the modified TPD function, with Lagrange multipliers $\overline{\mathbf{h}}$. In Appendix C, using some matrix algebra, the link between conventional and reduction methods is established, as well as an equivalence of convergence path in both hyperspaces.

At the solution, the relation between Jacobian matrices taking \mathbf{Q} and \mathbf{h} as independent variables is (see Appendix F)

$$J_{\alpha\beta}^R = \frac{1}{Y_T} \left[J_{\alpha\beta}^{R-Q} \right]^T; \quad \alpha, \beta = 1, M \quad (3.97)$$

It is clear from (eq. 3.97) that the condition numbers of the Jacobian matrix at the solution are the same for both reduction methods. This is not the case during iterations, each of the two reduction methods having its own convergence path.

3.2.2.2.1 Calculation Procedure

The calculation procedure consists in Successive Substitution (SS) iterations, followed by Newton iterations; however, in the proposed reduction method, one can skip the SS sequence and start directly with Newton iterations.

3.2.2.2.1.a Initialization

Initial guesses for K -values are obtained from the Wilson relation [Wilson [1969]]. The two-sided initialization [Michelsen [1982b]] is used: $Y_i^{(0)} = z_i K_i$ (for liquid phase stability) and $Y_i^{(0)} = z_i / K_i$ (for vapor phase stability). Note that in a multiphase context, more initial guesses may be required [Michelsen [1982a]]; several initialization procedures were proposed [Cañas Marín et al. [2007], Li and Firoozabadi [2012]]. [Firoozabadi and Pan [2002]] used a multilinear regression (as suggested by [Kaul and Thrasher [1996]] for flash calculations) to initialize $\overline{\mathbf{h}}$ by projecting $\ln \mathbf{K}$ onto the $\overline{\mathbf{h}}$ space via an equation similar in form to (eq. 3.95) (this requires the resolution of a linear system of $M + 1$ equations and a number of summations over nc). In this work, the reduction parameters are calculated directly from the initial guess at the first iteration.

Algorithm 3.1 Stability testing with the new reduction method

```

Calculate  $d_i$ , for feed composition
Initialize K-values from Wilson's relation;
Initialize  $Y_i^0$ ;
 $k = 0$ 
Newton = false
while  $S > \epsilon$  do
     $k = k + 1$ 
    if  $k > 1$  then
        Given  $\mathbf{h}$ , calculate  $\ln \phi_i$  from (eq. 3.53)
        if Newton then
            Calculate the elements of Jacobian matrix  $\mathbf{J}^R$  (eq. 3.82)
            Solve the linear system of equations (eq. 3.93)
            Update  $h_\alpha^{(\nu+1)} = h_\alpha^{(\nu)} + \Delta h_\alpha^{(\nu+1)}$ ;  $\alpha = 1, M$ 
            Update  $h_{M+1}$  from (eq. 3.94)
            Calculate K-values from (eq. 3.95)
            Calculate  $\mathbf{Y}$  from (eq. 3.12)
        else
            Calculate  $\mathbf{Y}$  from (eq. 2.14)
        end if
    end if
    Calculate mole fractions  $\mathbf{x}$  from (eq. 2.9)
    Calculate reduction parameters (eq. 3.47)
    Calculate compressibility factor from the cubic EoS (eq. 1.71)
    Calculate  $\mathbf{h}(\mathbf{Q})$  from (eq. 3.54, eq. 3.55, eq. 3.56)
    Compute the error vector  $\mathbf{e}$  from (eq. 3.81)
    Calculate the Euclidean norm,  $S = \sqrt{\sum_{\alpha=1} e_\alpha^2}$  from (eq. 3.98)
    if  $\|\Delta \mathbf{Y}\| \leq \epsilon_{SW}$  then
        Start Newton sequence
        Newton = true
    end if
end while

```

3.2.2.2.1.b Stopping/switching criteria

The stopping criterion is based on the Euclidean norm of the error vector (gradient vector in the reduction method)

$$S = \sqrt{\sum_{\alpha=1}^{M+1} e_\alpha^2} < \epsilon \quad (3.98)$$

The Euclidean norm of the gradient vector in the conventional method is also evaluated

$$S_f = \sqrt{\sum_{i=1}^{nc} g_i^2} < \epsilon_f \quad (3.99)$$

A calculation procedure consisting in SS iterations, followed by Newton iterations combines the advantages and avoids disadvantages of each method. Since, as will be seen in the results section, the convergence radius of the proposed method is quite large and Newton iterations can be used directly; in this case, one can skip step (12) in the sequence described above. Both options (with/without switch) are used in the next section. The criterion for switching from SS to Newton iterations is taken from [Hoteit and Firoozabadi [2006]].

$$\|\Delta \mathbf{Y}\| = \left(\sum_{i=1}^{nc} \left(Y_i^{(\nu)} - Y_i^{(\nu-1)} \right)^2 \right)^{\frac{1}{2}} < \epsilon_{SW} \quad (3.100)$$

3.2.2.3 Results

In this section, several conventional and two reduction methods are tested for various test problems. The two reduction methods considered here are the proposed method (denoted **h – red.**) and the reduction stability method of [Nichita et al. [2007b]] (denoted **Q – red.**), as modified by [Hoteit and Firoozabadi [2006]] (that is, if a reduced variable **Q** goes out of its bounds, $q_{\alpha min} \leq Q_{\alpha} \leq q_{\alpha max}$, revert to SS iterations). Both methods have a very interesting particularity: convergence is ensured (except near the STLL) using only Newton iterations. In the conventional methods, three sets of independent variables are used: Y_i , $\ln Y_i$ and α_i . The tolerance used in all calculations is $\epsilon_f = 10^{-12}$ in (eq. 3.99); the norms S (eq. 3.98) and S_f (eq. 3.99) have the same order of magnitude. The tolerance $\epsilon_{SW} = 10^{-2}$ (as suggested in [Hoteit and Firoozabadi [2006]]) is used whenever a switch is performed from SS to Newton iterations. The Peng-Robinson EoS [Peng and Robinson [1976], Robinson and Peng [1978]] is used in all calculations.

3.2.2.3.1 Y8 mixture

The six-component Y8 mixture [Yarborough [1972]] is a synthetic gas condensate mixture containing normal-alkanes. All BIPs are taken equal to zero ($c = 0$, $m = 1$, $\lambda_1 = 6$), thus there are only three equations. The calculated critical point is $T_C = 293.78$ K and $p_C = 210.67$ bar; the phase envelope and the STLL can be seen in fig. 3.1b from [Nichita et al. [2007c]]. At $T = 335$ K (in the retrograde condensation region), the dewpoint pressure is $p_d = 224.39$ bar, and the stability test limit is at $p = 225.26$ bar.

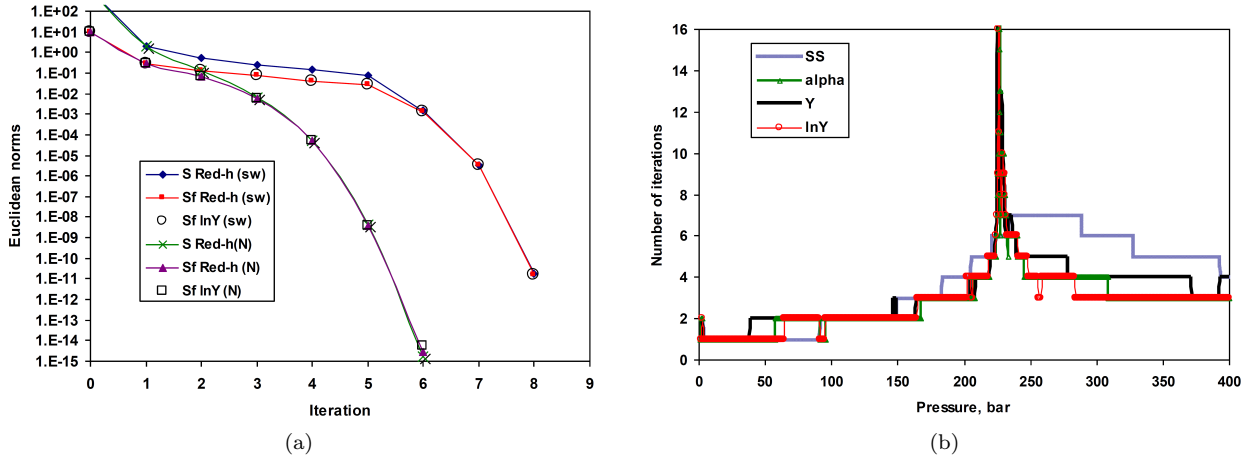


Figure 3.1: a) Euclidean norms vs. iteration level; Y8 mixture at $T=335$ K and $P=200$ bar b) Number of iterations for the Y8 mixture at $T=335$ K in the conventional method

At $P = 200$ bar, the variations of the Euclidean norm S_f vs. iteration number are plotted in fig. 3.1a (in two cases: for switching from SS iterations and for direct Newton iterations) for the proposed stability reduction method and for the conventional method with $\ln Y_i$ as independent variables, as well as the norm S for the proposed method. The norms S_f are almost identical for $\ln Y$ and **h – red.** since the two methods have the same theoretical convergence path. The number of iterations vs. pressure for vapor phase stability testing on the isotherm $T = 335$ K are plotted for various independent variables in the conventional method (with switch from SS iteration) in fig. 3.1b and for the two reduction methods (only Newton iterations) in fig. 3.2a. If one starts directly with Newton iterations for the variables α_i (fig. 3.2b), the method eventually converges, but it requires a quite large number of iterations (more than 20) over a wide pressure interval; in this case, using the variables Y_i , the number of iterations is even higher, and severe convergence problems occur near the STLL, while using $\ln Y_i$ as independent variables, the conventional method converges in less than 8 iterations, except in the vicinity of the STLL. For liquid phase stability testing, the numbers of iterations on the same isotherm are plotted in fig. 3.3a for the two reduction methods. The two peaks

correspond to the intersection of the spinodal with the isotherm, at about 215.5 bar and 73.5 bar (trivial solution saddle points); convergence problems in the vicinity of the spinodal are much less severe than those near the STLL.

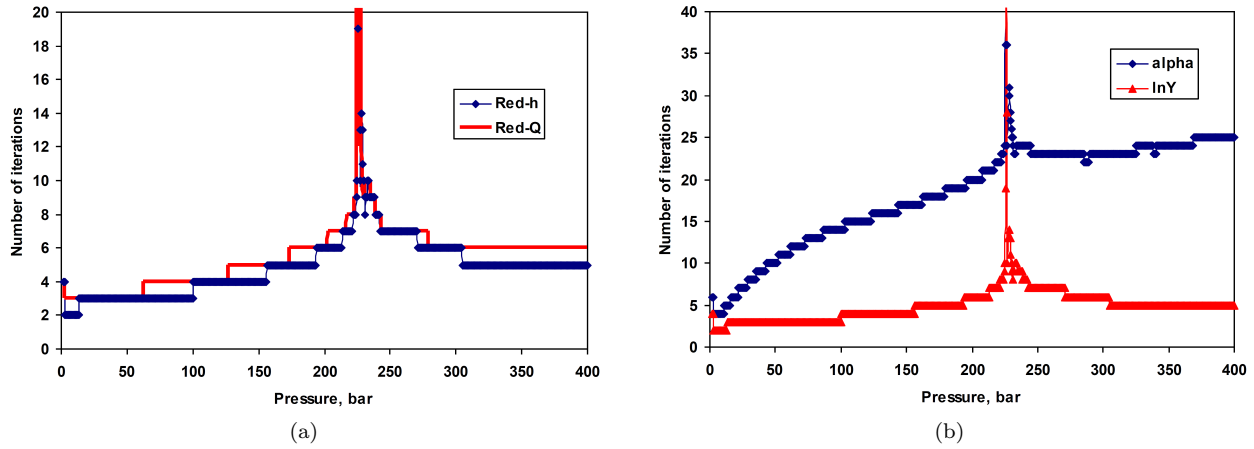


Figure 3.2: a) Number of iterations for the Y8 mixture at $T=335$ K using reduction methods, b) Number of Newton iterations for the Y8 mixture at $T=335$ K in the conventional method

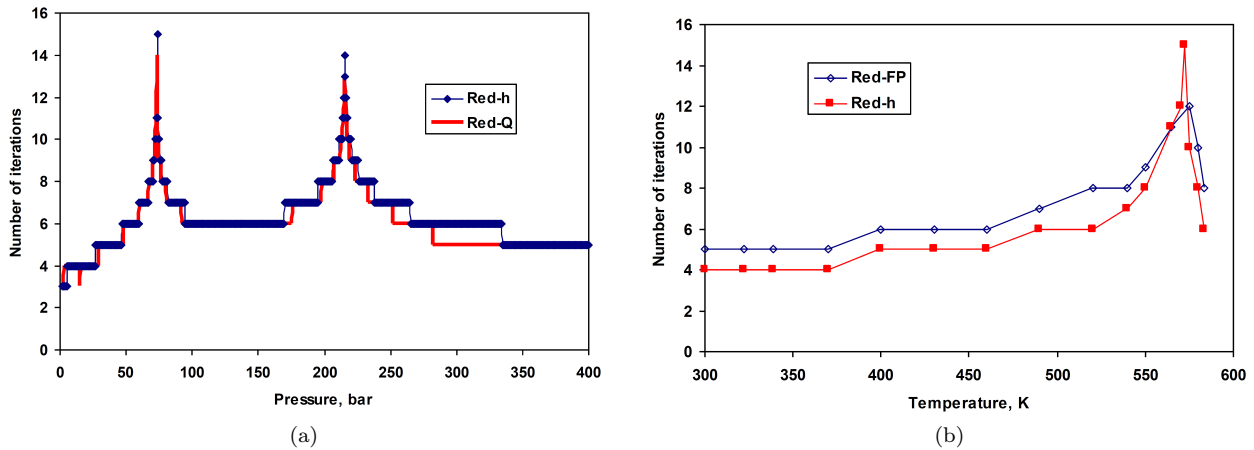


Figure 3.3: a) Number of iterations for the Y8 mixture at $T=335$ K; liquid phase stability, b) Number of iterations for the MY10 mixture at saturation points using reduction methods

3.2.2.3.2 MY10 mixture

The ten-component MY10 (Metcalf and Yarborough [Metcalf and Yarborough [1979]]) mixture is a synthetic oil containing normal alkanes. Composition, component properties and BIPs are taken from [Firoozabadi and Pan [2002]]. Only methane has non-zero BIPs with the other components, $c = 1$, $m = 3$ (non-zero eigenvalues are $\lambda_1 = 9.9574$, $\lambda_2 = 0.0707$, $\lambda_3 = -0.0280$). The calculated critical point is $T_C = 572.2296$ K and $p_C = 79.9367$ bar.

The number of iterations at saturation pressures using the proposed method, as well as those reported [Firoozabadi and Pan [2002]] (denoted here red-FP), are plotted in fig. 3.3b. Extremely close to the critical point our method converges in 15 Newton iterations (note that [Firoozabadi and Pan [2002]] does not report the number of iterations in the temperature window containing the critical point), while at most temperatures 4 to 7 iterations are required for convergence.

3.2.2.3.3 MY10/CO₂ mixture

An amount of 85 % CO₂ is added to the MY10 mixture. In this case, methane and CO₂ have non-zero BIPs ($c = 2$, $m = 5$, and the five nonzero eigenvalues are $\lambda_1 = 10.7487$, $\lambda_2 = 0.2207$, $\lambda_3 = 0.0643$, $\lambda_4 = -0.0328$, $\lambda_5 = -8.64e^{-4}$). At $T = 322$ K, the mixture is at near critical conditions, on the dewpoint side; the dewpoint pressure is $p_d = 119.64$ bar and the STLL is at $p = 120.31$ bar. For this isotherm, the number of iterations required for stability testing in the two reduction methods is plotted vs. pressure in fig. 3.4a, and the results for several independent variables in the conventional method are plotted in fig. 3.4b.

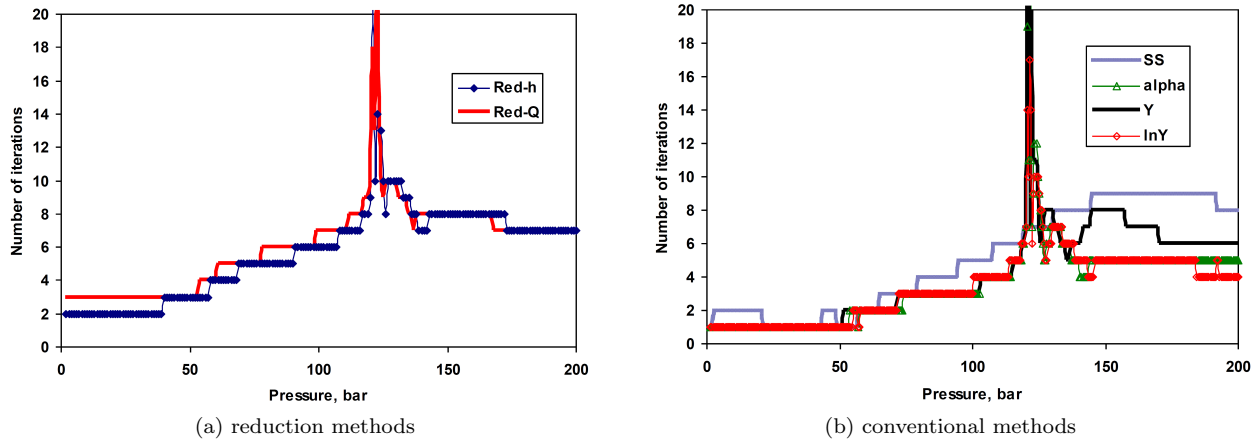


Figure 3.4: Number of iterations for the MY10 - 85% CO₂ mixture at T=322 K

3.2.2.3.4 Billings Crude/Natural Gas Mixture

A Billings crude oil/natural gas mixture from [Roland [1945]] is described by 11 components, with all relevant data taken from [Firoozabadi and Pan [2002]], except the BIP between methane and the heaviest pseudo-component, which is set at $C_{C_1-C_{19}} = 0.04$ (there is a possible misprint in Table 3 from [Firoozabadi and Pan [2002]] where this BIP is zero). Using this characterization, the mixture is a near-critical dewpoint fluid at the temperature $T=366.48$ K (the calculated critical point is $T_C = 355.5929$ K and $p_C = 735.8542$ bar), with a dewpoint pressure $p = 718.698$ bar and the stability test limit at $p = 718.704$ bar (extremely narrow 'shadow' region due to the proximity of the critical point). Only methane has non-zero BIPs with the remaining components ($c = 1$, $m = 3$, and the three non-zero eigenvalues are $\lambda_1 = 10.978282$, $\lambda_2 = 5.008757e^{-2}$, $\lambda_3 = -2.837006e^{-2}$), thus there are five independent variables.

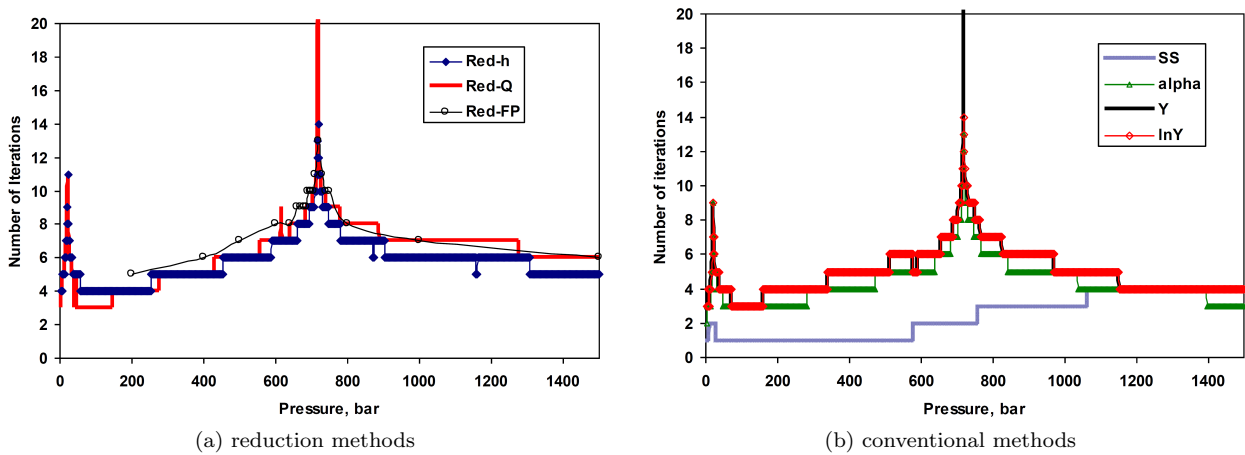


Figure 3.5: Number of iterations for the Billings mixture at T=366.48 K

The results of stability testing on the isotherm $T = 366.48$ K (number of iterations vs. pressure) are given in fig. 3.5a (for the two reduction methods as compared with the results reported in [Firoozabadi and Pan [2002]]) and in fig. 3.5b (for the conventional method). The peak at a low pressure corresponds to the lower branch of the spinodal (at $p = 21.84$ bar).

3.2.2.3.5 Kilgren Reservoir fluid

This is a near-critical reservoir fluid [Kilgren [1966]]. Its characterization by 14 components (with five pseudo-components in the C7+ fraction) is taken from [Firoozabadi and Pan [2002]]. Three components (C1, N2 and CO₂) have non-zero BIPs with the remaining ones ($c = 3$). The six nonzero eigenvalues ($m = 6$) are: $\lambda_1 = 13.561952$, $\lambda_2 = 0.417676$, $\lambda_3 = 0.071585$, $\lambda_4 = -0.043116$, $\lambda_5 = -0.013293$, and $\lambda_6 = 0.005196$. According to this reservoir fluid description, at the reservoir temperature $T = 399.82$ K, the mixture is a high-shrinkage oil (the calculated critical point is $T_C = 476.9271$ K and $p_C = 842.5883$ bar), with a bubble point pressure of $p_b = 1053.144$ bar (the stability test limit is at $p = 1054.545$ bar). Fig. 3.6a depicts the number of iterations required by the two reduction methods (**Q** – red. and **h** – red.) as compared to those reported in [Firoozabadi and Pan [2002]], while the number of iterations for the conventional method using various independent variables are plotted in fig. 3.6b (note that for this mixture the switching criterion is satisfied after just one iteration). As for the previous example, the less pronounced peak at a low pressure corresponds to the lower branch of the spinodal.

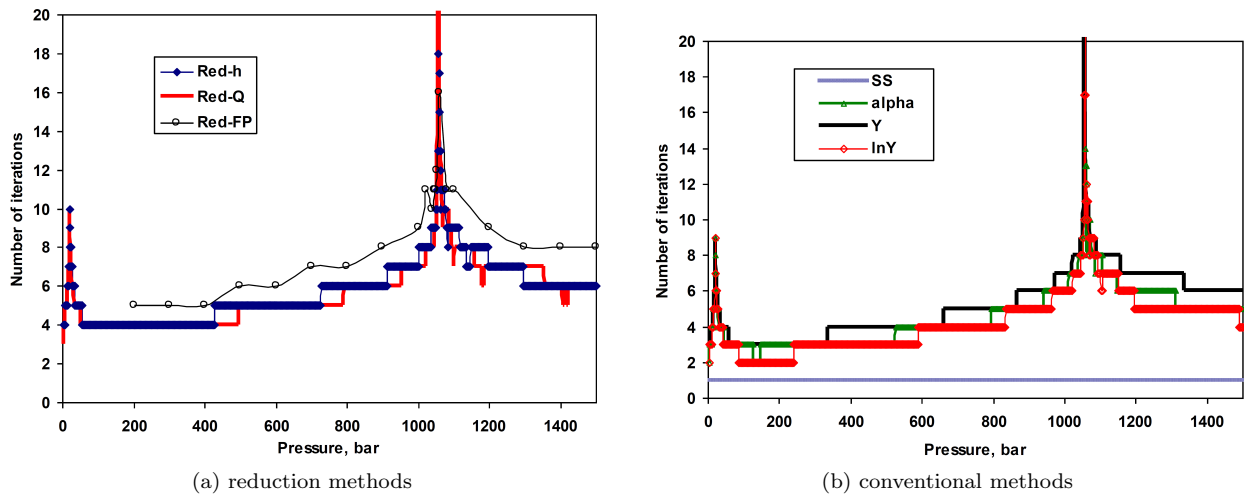


Figure 3.6: Number of iterations for the Kilgren mixture at $T=399.82$ K

3.2.2.4 Discussion

It is very important that the independent variables (the elements of **h**) are unbounded (they are related to $\ln \phi_i$) while the reduction parameters **Q** are bounded (they are related to $x_i \in (0, 1)$); this is, a major advantage of the proposed method for phase stability. If **Q** are independent variables [Pan and Firoozabadi [2003], Hoteit and Firoozabadi [2006]], they may go out of bounds during Newton iterations; in this case either a line search procedure is used [Nichita et al. [2007a]], or a switch back to SS iterations is required [Hoteit and Firoozabadi [2006]] (near the STLL in the single phase region, the line search can be often tedious, and one may need to revert to SS a very large number of times; note that in the latter case, partial derivatives are evaluated but not used for updating the variables). In the proposed approach, the reduction parameters are dependent variables (they are calculated by (eq. 3.47) from mole fractions normalized by (eq. 2.9)), and it is guaranteed that they are always within their bounds ($\min_i q_{\alpha i} < \mathbf{Q} < \max_i q_{\alpha i}$; $\alpha = 1, M$; $i = 1, nc$).

Previous conventional methods for phase stability used $\alpha_i = 2\sqrt{Y_i}$ [Michelsen [1982b]] or the formal

mole numbers Y_i [Hoteit and Firoozabadi [2006]] as independent variables. [Hoteit and Firoozabadi [2006]] reported that using α_i did not add computational efficiency to the Newton method (variables α_i are, however, very important for quasi-Newton methods, which are very sensitive to scaling). In this work we propose the use of $\ln Y_i$ as independent variables for conventional stability testing. Numerical examples show that using Y_i or α_i require almost the same number of Newton iterations (with a small advantage for α_i , which requires one iteration less in some cases); $\ln Y_i$ performs slightly better than α_i when Newton iterations are preceded by a number of SS iterations. However, without SS, the Newton method converges rapidly in all cases for $\ln Y_i$, while for α_i the Newton method eventually converges, but a large number of iterations are required (typically more than 20 for the mixtures studied, see for instance fig. 3.2b). This is due to the fact that for α_i the diagonal terms in the Hessian matrix are $\delta_{ij}(1 + 1/2\bar{g}_i^*)$, as compared to δ_{ij} for $\ln Y_i$. The Hessian for α_i is better conditioned at the solution (where $\bar{g}_i = 0$), but may have a very bad condition far from the solution (in early stages of the process). In the conventional method, we advocate the use of $\ln Y_i$ as independent variables. If it to choose between the variables Y_i and α_i , we recommend α_i .

Superior performances (smaller numbers of iterations and the possibility to apply directly the Newton method without the need of SS in the early stages of calculations) of reduction methods [Firoozabadi and Pan [2002], Nichita et al. [2006a], Hoteit and Firoozabadi [2006]] with respect to the conventional approach have been attributed [Firoozabadi and Pan [2002], Hoteit and Firoozabadi [2006]] to a relative smoothness of the *TPD* surface in the reduced variable hyperplane. In fact, the proposed reduction method have the same convergence path (and hence the same number of iterations) as in the compositional space with a specific change of variables (i.e., $\ln Y_i$), as shown in Appendix C, therefore obtaining the convergence using only Newton iterations is not a feature of the reduced space as claimed in [Firoozabadi and Pan [2002]]. One can similarly show that the reduced flash of [Nichita and Graciaa [2011]] is linked to conventional flash calculations using $\ln K_i$ as independent variables (this was observed but not proved in [Nichita and Graciaa [2011]]; the proof is given in [Petitfrere and Nichita [2014b]]).

A detailed comparative analysis of conventional and reduction methods, including CPU times, is given in [Petitfrere and Nichita [2014b]]. It appears that the fastest method in conventional stability is $\ln K_i$ with Cholesky decomposition of the Jacobian matrix in the resolution of the linear system, and that both reduction methods (**h – red.** and **Q – red.**) are faster than any conventional method if nc is large enough and m is small enough (reduction methods are clearly not efficient for mixtures with small nc and full-ranked BIP matrices). The proposed reduction method is slightly faster than **Q – red.**. For example, if two components have non-zero BIPs with the remaining ones (giving 6 independent variables for **Q – red.** and 7 for **h – red.**), the CPU time for a Newton iteration is proportional to [Petitfrere and Nichita [2014b]]: $t = 0.0175 + 0.0011nc + 0.0002nc^2$ for $\ln Y_i$ (with Cholesky factorization), $t = 0.0237 + 0.0025nc$ for **Q – red.**, and $t = 0.0269 + 0.0014nc$ for **h – red.**. Note that reduction methods exhibit a linear dependence of computation time vs. the number of components, while for $\ln Y_i$ the quadratic dependence exhibits a small curvature (as recently reported also by [Michelsen et al. [2013b]]). The CPU time per iteration (for the same number of iterations required for convergence) for the proposed reduction method is smaller than the one required by **lnY – chol.** starting from 8 components. The proposed method is twice as fast for 20 components, about three times faster for 30 components; for very many components (hundreds, as encountered in some process simulations), the proposed reduction method can be several orders of magnitude faster than conventional ones. Note that these observations were made for an optimized solver for linear systems for $\ln Y_i$; if the same solver, that is, LU decomposition, is used for both methods, the differences between conventional and reduction methods are much higher [Petitfrere and Nichita [2014b]].

The proposed set of variables can also be used in the framework of conventional method for stability testing (with $\ln \mathbf{Y}$ as primary variables) in order to reduce the dimensionality of the linear system of equations.

More precisely, one may keep the $\ln \mathbf{Y}$ Newton implementation in an existing code and use the reduction concept only to solve a linear system of significantly reduced dimensionality; this can be done with minor modifications in the code [Petitfrere and Nichita [2014b]], and even if it is less efficient than the full reduction, such an approach can lead to savings in computer time for large systems (mixtures with many components), mainly if the number of components with non-zero BIPs is small. While such an approach is not efficient in the typical component range of reservoir compositional simulation, it can be very useful for process or transport simulations.

The cause of convergence problems in phase stability calculations is the topology of the TPD surface (see [Nichita et al. [2007c]] for details, including graphical representations of the TPD surface for a binary mixture). At the STLL ('shadow curve'), the stationary point of the TPD function is a saddle point (the Hessian is indefinite and singular at a non-trivial solution with a positive TPD), and the Hessian matrix is ill-conditioned in the vicinity of the STLL (the most severe problems arise for a pressure interval above the STLL; there are no convergence problems for pressures just below the STLL). Phase stability calculations near the STLL are far more difficult than those in the vicinity of a critical point (where the Hessian matrix is positive semidefinite). Any algorithm will experience difficulties in a pressure domain above the STLL. It is clear that a change of variables does not eliminate this problem, since the TPD function in the new hyperspace inherits certain properties from the original one. Considerable research efforts are currently done to solve this problem and different techniques (other than changing the variables) are being investigated to eliminate or at least alleviate problems near the STLL; forthcoming papers will treat this issue.

3.2.2.5 Conclusion

A new reduction method for phase stability testing has been proposed in this chapter, based on the multi-linear expression of the logarithms of fugacity coefficients as functions of the coefficients \mathbf{h} (which are taken as independent variables and are unbounded). The reduction parameters are dependent variables, which, unlike in previous formulations, are guaranteed to be within their bounds. The dimensionality of the problem depends only on the number of components having non-zero BIPs with the remaining ones, and not on the number of components in the mixture, as in the conventional approach.

The proposed method has a high convergence radius and Newton iterations can be used directly (without the need of using successive substitutions); it has the same convergence path as the conventional approach using the natural logarithm of formal mole numbers as independent variables (which seems to be the best choice in the compositional space). It has been shown that the primary variables in our method are Lagrange multipliers corresponding to a constrained minimization of the modified TPD function with respect to a specific set of variables. Thus, the proposed method facilitates comparison of reduced and conventional approaches for phase stability analysis. An apparently hidden formal link between conventional and reduction methods is revealed.

Several test problems for various synthetic and reservoir mixtures have proved the robustness of the proposed method and show that it requires systematically less iterations than i) conventional methods using other independent variables than $\ln Y_i$, and ii) previous reduction methods.

3.2.3 A comparison of conventional and reduction approaches for phase equilibrium calculations

3.2.3.1 Introduction

Our recent formulations of reduction methods for flash calculations [Nichita and Graciaa [2011]] and phase stability testing [Nichita and Petitfrere [2013]] are related to constrained minimizations of the objective

functions (Gibbs free energy and a modified TPD function) and thus facilitate comparisons with conventional methods [Michelsen et al. [2013b]]. Besides, they require simpler partial derivatives in the Newton method.

Several very recent papers ([Michelsen et al. [2013b]], Haugen and Beckner [2013], Gorucu and Johns [2013]) have questioned on the efficiency of the reduction methods as compared to conventional methods. The common conclusion of these papers is that reduction methods become faster than conventional ones starting from a certain number of components in the mixture (about 20 if few components have non-zero BIPs).

In this subsection, we intend to look, besides CPU times required by various conventional and reduction formulations, at:

- Stability testing and flash calculation separately
- Computation times per iteration and globally
- Condition number of the matrix in the linear system of equations
- The solver for the linear system and the effect of using symmetry
- Formal links between conventional and reduction methods, as well as between different reduction methods

Numerical experiments are carried out for various formulations and sets of independent variables for conventional and reduction flash and stability, followed by a discussion and the conclusions. Four Appendices give: the formulation of the reduction flash as a constrained minimization problem, the equivalence of conventional and reduction flash methods (convergence path identity), a link between two reduction methods for phase stability testing and some features of the pseudo-reduced method for phase stability.

3.2.3.2 Results

Numerical experiments have been performed for three mixtures presented in subsection 3.2.2, using the Peng-Robinson EoS [Peng and Robinson [1976], Robinson and Peng [1978]]:

- Y8 mixture (0 BIP mixture).
- MY10 mixture (1 BIP family mixture).
- MY10+CO₂ mixture (2 BIP families mixture). An amount of 16.67% CO₂ is added to the MY10 mixture. In this case, methane and CO₂ have non-zero BIPs ($c = 2$, thus $m = 5$ and $M + 1 = 7$).

The mixture compositions and the non-zero BIPs are presented in table 3.1. The phase envelopes of the three mixtures (showing the critical points) are plotted in fig. 3.7.

Components	Y8		MY10	
	z_i	z_i	C_{C_1-j}	C_{CO_2-j}
C1	0.8097	0.35	-	0.093
C2	0.0566	0.03	0.00	0.128
C3	0.0306	0.04	0.00	0.123
nC4	-	0.06	0.02	0.136
nC5	0.0457	0.04	0.02	0.125
nC6	-	0.03	0.025	0.131
nC7	0.0330	0.05	0.025	0.120
nC8	-	0.05	0.035	0.120
nC10	0.0244	0.30	0.045	0.120
nC14	-	0.05	0.045	0.120

Table 3.1: Feed composition and BIPs for Y8 and MY10 mixtures

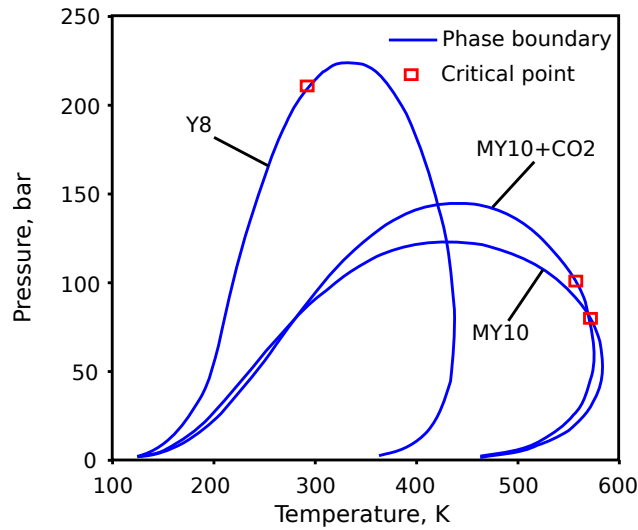


Figure 3.7: Phase envelopes of Y8, MY10 and MY10+CO₂ mixtures

3.2.3.2.1 Solving the linear system

For flash calculation and stability analysis problems, the linear system is generally small. The matrix ($nc \times nc$) is dense for both problems, where nc exceeds 100 for instance in certain process simulations. Direct solvers have been applied in this paper as more efficient compared with iterative ones. Two direct solvers have been used:

- The LU factorization, which is applicable to any system and solves the linear problem in $2/3n^3$ FLOPs (FLoating point Operations).
- The Cholesky factorization, which is applicable to symmetric matrices and takes advantage of the symmetry to solve the linear problem in $n^3/3$ FLOPs. It is important to note that for a symmetric matrix (Hessian and \mathbf{U}^{-1}), only about half of the matrix ($nc \times (nc + 1)/2$ entries) needs to be computed when a Cholesky decomposition is used.

3.2.3.2.2 Independent variables

3.2.3.2.2.a Flash calculations

1. Independent variables leading to symmetric systems solved using Cholesky decomposition

- **nL/nV** the component mole numbers in liquid or vapor phase are taken as independent variables. In this case the Rachford-Rice needs not to be solved.
- **lnK – Chol.** the independent variables are the natural logarithms of the equilibrium constants. The Hessian matrix is decomposed and the linear system is solved as described in a previous section.
- **Michelsen – red.** an $(M + 1) \times (M + 1)$ linear system is solved for the independent variables **h**.

2. Independent variables leading to non-symmetric systems, solved using LU decomposition

- **lnK** for the $\ln K_i$ variables, the Jacobian matrix is non-symmetric and the linear system can be solved directly by means of an LU factorization. Both cases have been tested to see the impact of the Cholesky solver as compared with an LU factorization for the linear system.
- **h – red.** Reduction method with **h** as independent variables.

- **Q – red.** Reduction method with \mathbf{Q} as independent variables (QBL or QBV iterations).

3.2.3.2.2.b Stability testing

1. Independent variables leading to symmetric systems solved using Cholesky decomposition

- **Y** the formal mole numbers Y_i are independent variables.
- **α** the independent variables are Michelsen's .
- **lnY – Chol.** the natural logarithms of Y_i are taken as independent variables; the Hessian matrix is decomposed and the linear system is solved as described in a previous section.
- **Michelsen – red.** an $M \times M$ linear system is solved for the independent variables \mathbf{h} .

2. Independent variables leading to non-symmetric systems, solved using LU decomposition

- **lnY** similar to the **lnK** case, the system is also solved by means of an LU factorization to evaluate the impact of the Cholesky solver as compared to the LU solver.
- **h – red.** Reduction method with \mathbf{h} as independent variables.
- **Q – red.** Reduction method with \mathbf{Q} as independent variables.

3.2.3.2.3 Condition number

The condition number $\kappa(\mathbf{A})$ associated with a linear system of equations $\mathbf{Ax} = \mathbf{b}$ measures the approximation on the solution \mathbf{x} after solving the system. The bigger the condition number is, the bigger the approximation. The condition number associated with a matrix \mathbf{A} is [Golub and van Loan [1996]]

$$\kappa(\mathbf{A}) = \|\mathbf{A}^{-1}\|_2 \cdot \|\mathbf{A}\|_2 \quad (3.101)$$

The norm 2 of a matrix \mathbf{A} corresponds to the square root of the largest eigenvalue (spectral radius) of the matrix obtained by multiplying \mathbf{A} and its adjoint (transpose for real matrices).

$$\|\mathbf{A}\|_2 = \sigma(\mathbf{A}) = \sqrt{\lambda_{\max}(\mathbf{A}^T \mathbf{A})} \quad (3.102)$$

$$\|\mathbf{A}^{-1}\|_2 = \sigma(\mathbf{A}^{-1}) = \sqrt{\lambda_{\max}(\mathbf{A}^{-1} \mathbf{A}^{-T})} = \frac{1}{\sqrt{\lambda_{\min}(\mathbf{A}^T \mathbf{A})}} \quad (3.103)$$

where σ is the spectral radius and λ are the eigenvalues.

From (eq. 3.101, eq. 3.102 and eq. 3.103), the condition number is

$$\kappa(\mathbf{A}) = \sqrt{\frac{\lambda_{\max}(\mathbf{A}^T \mathbf{A})}{\lambda_{\min}(\mathbf{A}^T \mathbf{A})}} \quad (3.104)$$

For \mathbf{A} symmetric, this expression simply becomes the ratio of the largest to the smallest eigenvalue of \mathbf{A} .

$$\kappa(\mathbf{A}) = \frac{\lambda_{\max}(\mathbf{A})}{\lambda_{\min}(\mathbf{A})} \quad (3.105)$$

3.2.3.2.3.a Condition numbers in flash calculations

The condition number for the flash calculation problem has been calculated using various methods and variables along an isotherm for the Y8 mixture at $T=335$ K, up to the phase boundary (dewpoint at $p = 224.39$

bar) (fig. 3.8a). The condition number of the Jacobian/Hessian matrix is calculated at the solution.

The best condition number is given by the $\ln \mathbf{K}$ variables. Close to unity for most of the points, it still gives a good condition number close to the phase boundary. The condition numbers for the mole numbers $\mathbf{n}_L/\mathbf{n}_V$ follow the trend of those for both reduction methods (**Q** – red. and **h** – red.) along the two phase region, with values in the range of 100. However, the system becomes ill-conditioned close to the phase boundary, which explains the difficulties that can be encountered with this variable in the vicinity of phase boundaries. The reduction methods are less altered in this region than $\mathbf{n}_L/\mathbf{n}_V$.

Note that in **lnK** – **Chol.** the linear system in (eq. 3.25) is solved, thus the approximations will be potentially more important than by solving the non-symmetric system in (eq. 3.33) by an LU decomposition. Finally, **Michelsen** – **red.** gives a bad condition number all along the two phase region (around 105). In this method, the decrease of the dimensionality of the space is made at the expense of a severe loss of conditioning.

The singularity for the flash calculation problem is at the critical point (for $T \neq T_c$, the singularities are at the convergence locus for the negative flash [Nichita et al. [2007c], Whitson and Michelsen [1990]]), where two eigenvalues of the Jacobian matrix become zero [Michelsen [1982a]] and the condition number is infinite. Along the phase boundary, the closer to the critical point, the smaller the minimum eigenvalue, the higher the condition number.

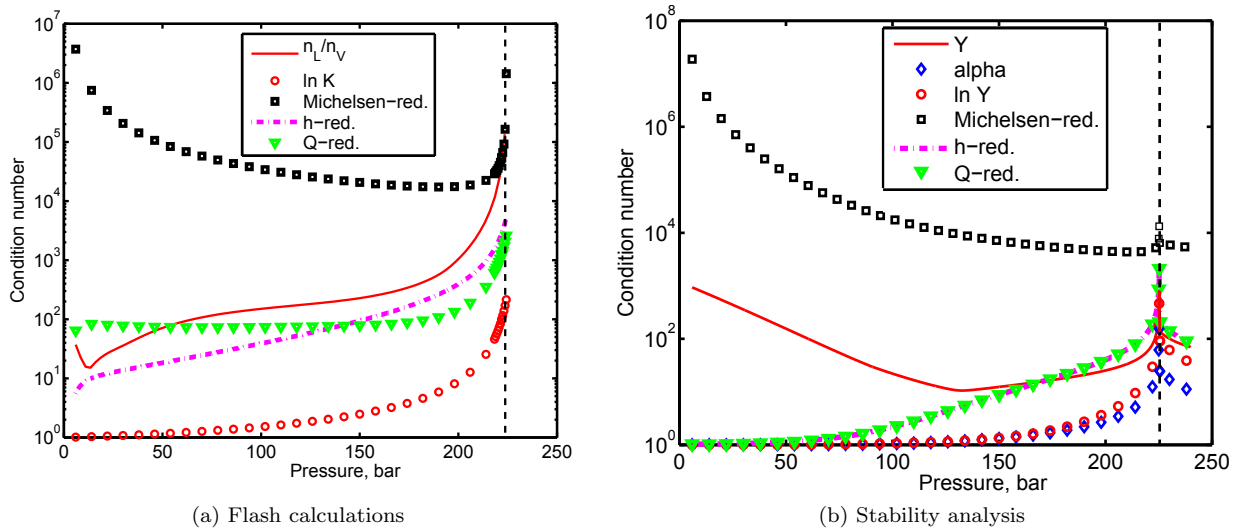


Figure 3.8: Condition number along the isotherm $T=335\text{K}$ for the Y8 mixture

3.2.3.2.3.b Condition numbers in stability testing

The condition number has been calculated using various methods and variables for the stability analysis problem with the same mixture and on the same isotherm at pressures up to 240 bar (fig. 3.8b). The feed is tested as a vapor phase. At $T=335\text{ K}$, the STLL is located at $p = 225.26\text{ bar}$. At the STLL (the singularity for stability testing), one of the Jacobian/Hessian's eigenvalues becomes zero, and the condition number becomes infinite (represented by an asymptotic line). This is the reason why phase stability calculations can be really difficult in the vicinity of the STLL.

As for the flash calculation problem, $\ln \mathbf{Y}$ (an analogy can be made with $\ln \mathbf{K}$ for flash calculations [Nichita and Petitfrere [2013]]) gives a really good condition number all along the isotherm, even when the STLL is approached. The condition number is around unity up to 150 bar and remains below 200 close

to the STLL. Using α gives a better conditioning than $\ln \mathbf{Y}$; however, far from the solution, especially in the early iteration stages, the linear system for $\ln \mathbf{Y}$ is better conditioned, allowing a safe earlier switch from SSI iterations, as discussed in [Nichita and Petitfrere [2013]]. The condition number obtained using \mathbf{Y} is rather high (around 100) all along the isotherm and it increases (around 1000) as the STLL is approached.

Both reduction methods (**Q – red.** and **h – red.**) have the same condition number at the solution (see Appendix F). It remains small up to $p = 150$ bar, but gets larger than for \mathbf{Y} when getting closer to the STLL. As for the flash calculations, the pseudo-reduced method (**Michelsen – red.**) gives a really bad condition number all along the isotherm (around 10^4) which indicates potentially important approximations when solving the linear system.

3.2.3.2.4 Computational time

The computational time is a really sensible parameter. It highly depends on many factors, such as computer features (hardware: processor, memory, etc.), compiler, implementation (cash optimization, vectorization, etc.). In this thesis, for all calculations, an Intel Xeon ES5405@2.00GHz processor has been used.

The choice of the linear solver is also a really important feature. A specific study has been made over different sets of linear solvers [Press et al. [1992], Lapack, homemade LU and Cholesky solvers). Some linear solvers are more adapted for large systems (i.e., Lapack). Some are more adapted for small systems (home-made). It has been noticed that depending on the solver used, the time spent on the linear solver ranged from 10% up to 55% of the total time. The homemade LU and Cholesky solvers were used in this paper.

To obtain the same properties for a mixture using a different number of components, some of the components were split. Given a mixture with nc components, one of them (different from C_1 and CO_2) with a feed composition of z_i , by splitting z_i into z'_i and z''_i , with $z'_i = z''_i = z_i/2$, and assigning to the two new components the same BIPs and critical properties, a new composition can be created leading to the same mixture properties.

Two different comparisons were made to test, separately for flash and stability, the time and the efficiency of each method for various sets of independent variables for mixtures with different number of components: i) time per iteration and ii) time spent for an entire pressure-temperature domain.

3.2.3.2.4.a Flash calculation

Time spent for one iteration

For a given pressure and temperature in the two-phase region, a switch to Newton iterations is carried out after performing a certain number of SSI iterations. Then, the Newton iteration is repeated 10000 times for each set of independent variables. The time spent between the switch to the Newton method and the end of the 10000 iterations is recorded. This operation is repeated for the three mixtures and for different numbers of components. This gives the time spent to perform one Newton iteration for different sets of independent variables, independently of their convergence behavior. This procedure was used for the Y8 mixture (with 0 BIPs) at $P=160$ bar and $T=335$ K. The results are given in fig. 3.9a. For calculations performed at $P=40$ bar and $T=550$ K, the results are reported in fig. 3.9b for the MY10 mixture (with 1 BIP family) and in fig. 3.10 for the MY10+ CO_2 mixture (with 2 BIP families).

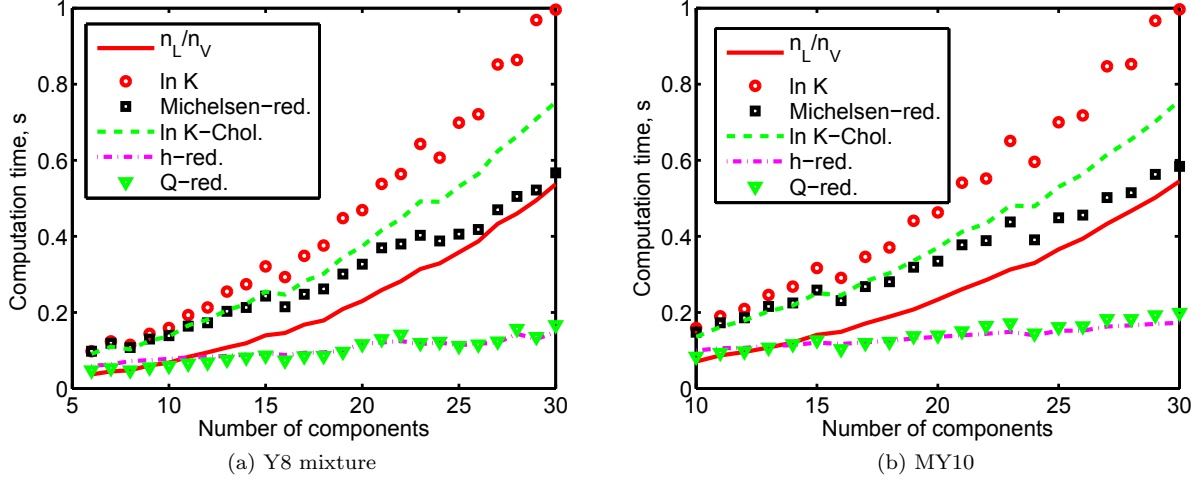


Figure 3.9: Computation time to repeat 10000 times one Newton iteration for the flash problem

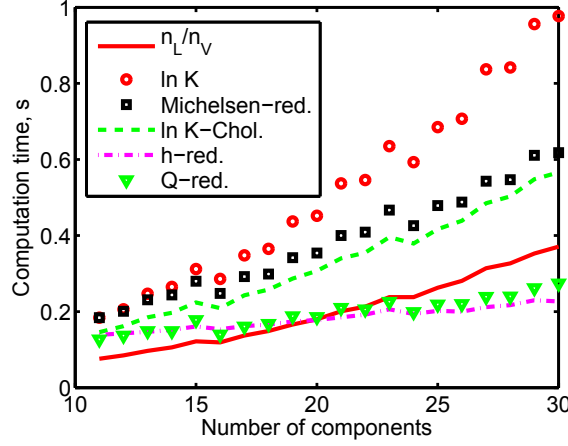


Figure 3.10: Computation time to repeat 10000 times one Newton iteration for the flash problem, MY10+C02

The relative position of the different curves is the same for the different mixtures. As expected, the use of a Cholesky instead of an LU decomposition to solve the linear system for the **lnK** variables decreases the computational time. The differences increase with the size of the system. The time earned per iteration using a Cholesky decomposition varies from a few percent for small systems (see fig. 3.9a for $nc < 10$) up to 50% for large systems (fig. 3.10 at $nc = 30$). This result aspect shows that the resolution of the linear system represents a small amount of the total computation for a small number of components, and as nc is increased, the differences become more and more important. As expected, since the Rachford-Rice equation is not solved, using the mole numbers **nl/nv** as variables leads to the smallest computational time among conventional variables. The computational times required by all conventional methods exhibit a quadratic trend with the number of components. However, for **lnK – Chol.** the quadratic coefficient of the dependence is the smallest, that is, this dependence is the closest to linearity; this observation is consistent with the results of [Michelsen et al. [2013b]], who reported an quasi-linear dependence.

Unlike in conventional methods, the use of reduction or pseudo-reduction methods is affected by the number of BIP families. The full reduction methods (**Q – red.**, **h – red.**) act similarly and exhibit a linear increase of the computational time with the number of components, with **h – red.** faster than **Q – red.** (here the partial derivatives in **Q – red.** were coded as described in [Nichita et al. [2007b]]). A Newton iteration is always faster in reduction methods than in conventional ones for BIPs all equal to zero (Y8) and for 1-BIP

family (MY10). As the number of BIP families increases, the time differences between methods diminish, and for 2-BIP families the time per iteration is almost the same for 11 mixture components. Computational times per iteration were also recorded for 100 and 200 component mixtures for the MY10 and MY10+CO₂ mixtures (see table 3.2). For the MY10+CO₂ mixture the **h – red.** method is almost ten times faster for 100 components, and almost twenty times faster for 200 components as compared to the **lnK – Chol.** method. The **lnK** method using an LU factorization requires almost 50% additional time than **lnK – Chol.** for a large number of mixture components.

Mixture	nl/nv	lnK	lnK-Chol.	Michelsen-red.	h-red.	Q-red.
1 BIP/100 comp.	6.89	11.55	7.82	4.03	0.42	0.56
1 BIP/200 comp.	28.71	46.99	31.36	14.22	0.78	1.09
2 BIP/100 comp.	4.00	11.42	9.81	4.57	0.57	0.75
2 BIP/200 comp.	16.16	46.72	37.43	16.68	1.05	1.45

Table 3.2: Time to perform 10000 times one Newton iteration for MY10 and MY10+CO₂ mixtures for the flash problem

The use of the pseudo-reduced method (**Michelsen – red.**) leads also to a quadratic shape of time vs. number of mixture components, but the quadratic coefficient is smaller than for the conventional variables. For small systems, the computational time is generally higher than for the classical variables. However, as the quadratic coefficient is smaller, the curves cross each other (with **lnK – Chol.** and **nl/nv**) after a certain number of components, depending on the number of BIP families (after 12 for the 0 BIP mixture, and 14 components for the 1 BIP family, and no crossing is seen within the 30 components range for the 2 BIP families mixture). From table 3.2, the pseudo-reduced method is around two times faster than conventional methods with 100 component mixtures.

Time spent for the whole two-phase region

For each mixture, a loop over a pressure and temperature domain is made. After performing a stability analysis, a Wilson initialization is carried out to perform a flash. Starting with SSI iterations, a switch to Newton is performed once the error in the Euclidean norm of the difference of the fugacities becomes smaller than $1e^{-2}$. The Newton method is then used up to convergence ($1e^{-8}$ in the fugacity error evaluated in the Euclidean norm). For each set of variables the time spent between the switch to Newton and the convergence is recorded. In the end, each time is added to get the total time spent to perform Newton iterations within the whole pressure-temperature range. The methodology, which takes into account both the time spent for each Newton iteration and the convergence behavior of each method, is repeated for different numbers of mixture components and for the different mixtures.

For all mixtures, $\Delta p = 1$ bar and $\Delta T = 1$ K. Fig. 3.11a gives the results obtained using the Y8 mixture (with 0 BIPs) for pressures between $p = 6$ bar and $p = 250$ bar and temperatures between $T=150$ K and $T=500$ K. A number of 40850 flashes were performed within this pressure-temperature window. For the MY10 mixture (with 1 BIP family), calculations were performed for pressures between $p = 20$ bar and $p = 140$ bar and temperatures between $T=250$ K and $T=600$ K, and the results are recorded in fig. 3.11b. A number of 28217 flashes were performed within these p/T ranges. The figure fig. 3.12 gives the results obtained with the MY10+CO₂ mixture (with 2 BIP families) for pressures between $p = 20$ bar and $P=300$ bar and temperatures from $T=100$ K up to $T=500$ K. A number of 37852 flashes were performed within these p/T ranges.

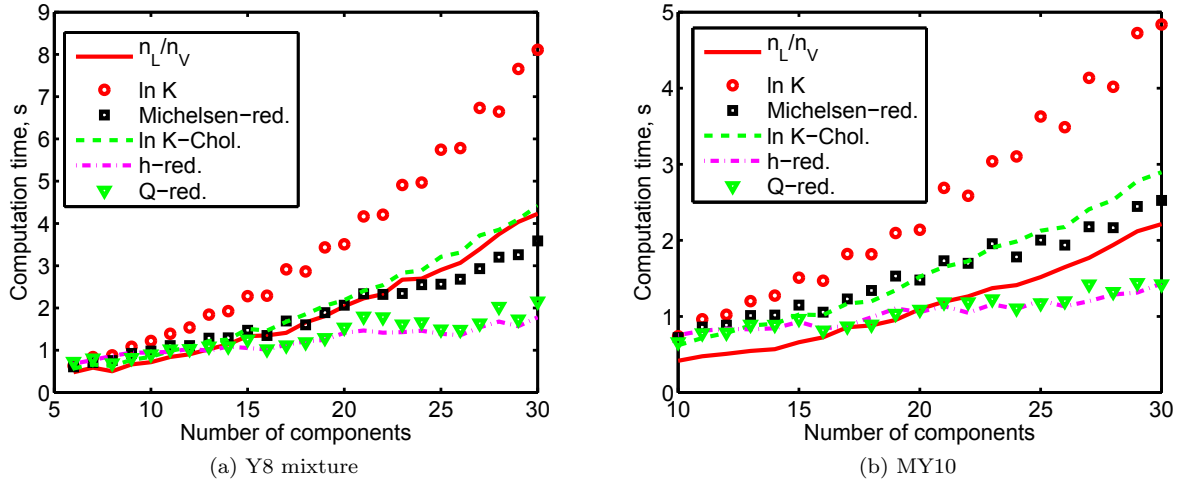


Figure 3.11: Computation time for the whole two-phase region for the flash problem

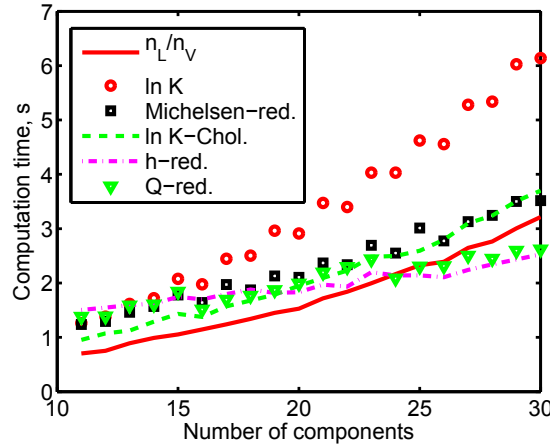


Figure 3.12: Computation time for the whole two-phase region for the flash problem, MY10+CO2

The same trend as observed per iteration can be found here for the different curves. The curves corresponding to conventional and reduction methods cross each other, and starting for a certain number of components the reduction methods become faster. It appears the reduction method **h-red.** is more efficient and stable than the reduction method **Q-red.** The conventional method using mole numbers as variables, **nl/nv**, seems the most efficient but less robust, as will be discussed in the next section.

3.2.3.2.4.b Stability analysis

Time spent for one iteration

The same methodology as for flash calculations is applied for stability analysis. Starting from the [Wilson [1969]] initialization, five SSI iterations are performed before switching to the Newton method. Fig. 3.13a plots the results obtained with the Y8 mixture, fig. 3.13b, those obtained with the MY10 mixture and fig. 3.14 the results for the MY10+CO₂ mixture. The pressure and the temperature chosen are the same as for the flash calculations.

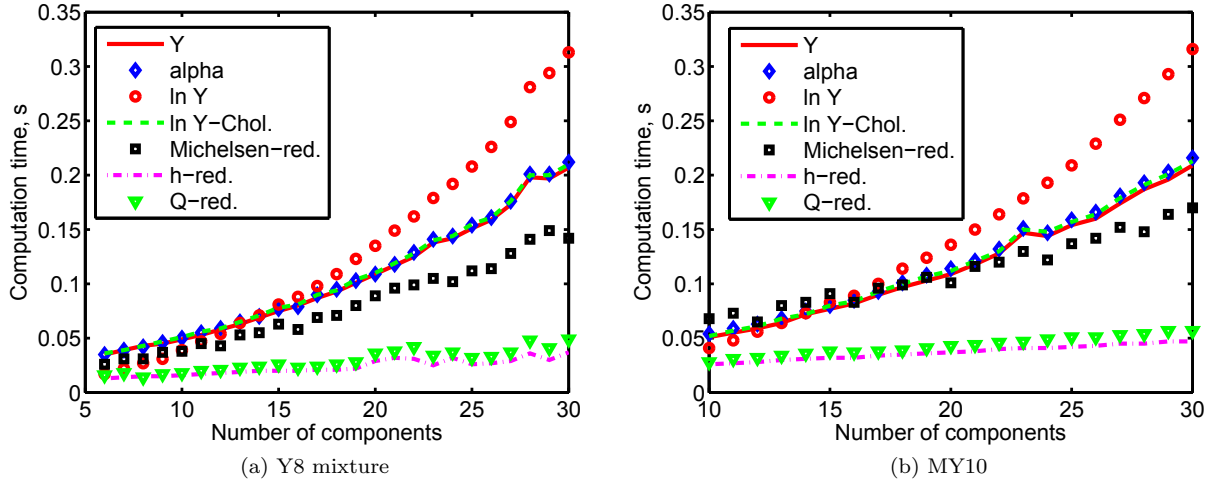


Figure 3.13: Computation time to repeat 10000 times one Newton iteration for the stability problem

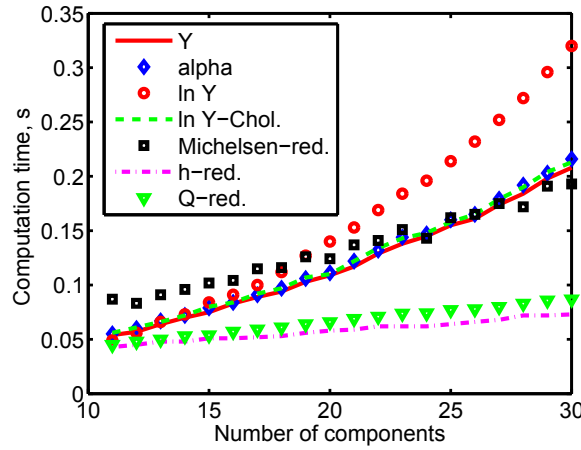


Figure 3.14: Computation time to repeat 10000 times one Newton iteration for the stability problem, MY10+CO₂

As the Rachford-Rice equation is not solved for stability testing, all conventional methods using the Cholesky factorization (Y , $\ln Y$, $\ln Y - \text{Chol.}$) act similarly. A quadratic trend of computational time is observed with the number of components. As expected, using the Cholesky decomposition ($\ln Y - \text{Chol.}$) decreases the computational time as compared with an LU decomposition ($\ln Y$). From fig. 3.13a, fig. 3.13b and fig. 3.14 it can be observed that, unlike in flash calculations, the use of the full reduction methods (Q -red. and h -red.) is always faster than the use of conventional methods for all three mixtures, even for a small number of components, h -red. being more efficient than Q -red. The trend is also linear, which accentuates the differences between full reduced methods and the other methods as the number of components increases.

In table 3.3, the computational times spent for the MY10 mixture and the MY10+CO₂ mixture with 100 and 200 components are reported. The reduction method h -red. is more than 10 times faster than conventional methods using Cholesky with 100 components, and more than 20 times faster with 200 component mixtures. The improvements using the full reduction methods as compared to the conventional methods are much better for the stability analysis problem than for the flash problems.

Mixture	Y	alpha	lnY	lnY-Chol.	Michelsen-red.	h-red.	Q-red.
1 BIP/100 comp.	1.91	1.97	3.83	1.92	1.09	0.13	0.17
1 BIP/200 comp.	7.39	7.69	15.83	7.41	3.94	0.24	0.33
2 BIP/100 comp.	1.89	2.10	3.98	2.51	1.13	0.19	0.26
2 BIP/200 comp.	7.37	8.37	16.52	8.79	3.90	0.34	0.49

Table 3.3: Time to perform 10000 times one Newton iteration for MY10 and MY10+CO₂ mixtures for the stability testing problem

The pseudo-reduced method also acts in a similar way as for flash calculations. Generally more time consuming for a small number of components than methods with classical variables, it becomes more efficient above a certain number of components. The trend is also quadratic, with a quadratic coefficient smaller than for classical variables. From table 3.3, with a 200 component mixture the pseudo-reduced method is more than twice faster than conventional methods, but an order of magnitude slower than full reduction methods.

Time spent for the whole pressure-temperature window

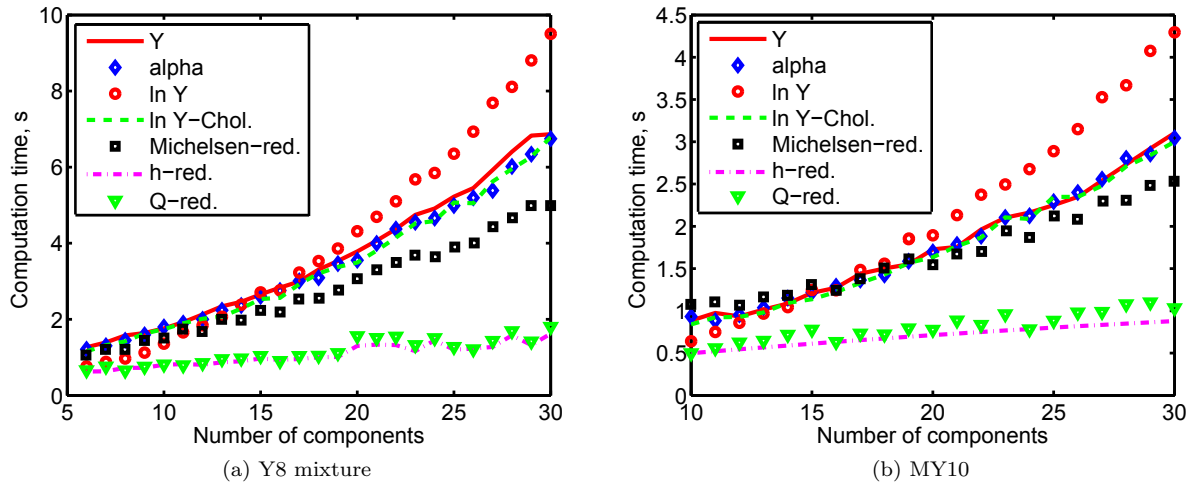


Figure 3.15: Computation time for the whole P-T window for the stability problem

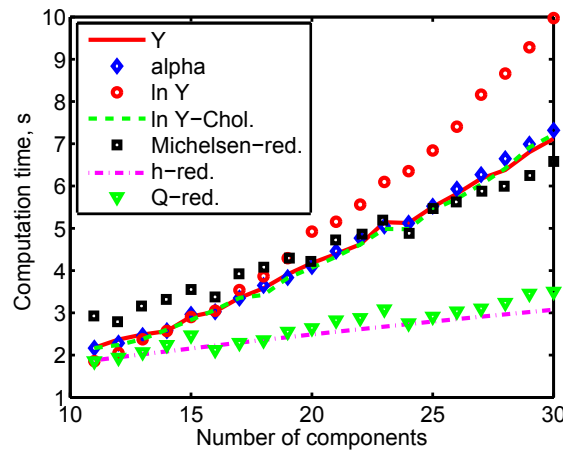


Figure 3.16: Computation time for the whole P-T window for the stability problem, MY10+CO₂

The same methodology as for flash calculations is applied for stability analysis. The switching criterion used is the one given by [Hoteit and Firoozabadi [2006]]. The same pressure-temperature ranges as for flash

calculations are used. This time, the number of stability calculations performed for each mixture is directly given by the number of tested pressure-temperature conditions. The results for the Y8 mixture are given in fig. 3.15a, for the MY10 mixture in fig. 3.15b, and for the MY10+CO₂ mixture in fig. 3.16.

The same conclusions as for the time per iteration can be given. The maximum number of iterations has been set to 200. Moreover, as Δp and ΔT are not very small (1 bar and 1 K), the number of points in the immediate vicinity of the STLL might be small as compared to the whole set of points. This could explain why the convergence behavior does not influence much the shape of the curves; only the time per iterations seems to influence the global result. However, for pressures just above the STLL, $\ln \mathbf{Y}$ and α are the variables giving a more robust formulation as compared to \mathbf{Y} variables (a more detailed discussion can be found in [Nichita and Petitfrere [2013]]).

3.2.3.3 Discussion

For the flash calculation problem, the use of mole numbers as variables (namely \mathbf{nL}/\mathbf{nV}) appears to be the most efficient among the conventional methods, and globally more efficient than reduction methods for a small number of components. However, this choice of variable does lead to less robustness: in the vicinity of the phase boundaries (even far from critical points) the \mathbf{nL}/\mathbf{nV} method has a small convergence ratio and may require a significantly increased number of iterations (starting from the same switch from SSI iterations) to achieve convergence, as compared to other conventional formulations, as well as with reduction methods. The error (Euclidean norm) vs. the iteration level for a two-phase flash of the Y8 mixture at $T=366$ K and $P=214$ bar (very close to the dewpoint pressure in the retrograde condensation region) is plotted for various methods in fig. 3.17a. The $\ln \mathbf{K}$, \mathbf{h} – red. and Michelsen – red. Newton methods require only 3 iterations after the switch. The curves are superposed for $\ln \mathbf{K}$ and \mathbf{h} – red.; only a slight difference can be observed for the pseudo-reduced method. The \mathbf{nL}/\mathbf{nV} method requires more than 60 iterations, with a large number of switches back from Newton to SSI due to an increase in Gibbs free energy between two subsequent iterations (note that in this case the Jacobian is evaluated and the linear system solved, but this information is not used since a SSI step is taken). This kind of behavior is typical for \mathbf{nL}/\mathbf{nV} . Newton iterations near phase boundaries, and the severity of these problems increases as the critical point is approached. For this choice of variables, it is recommended the use of a Trust-region method, which highly improves robustness (see [Petitfrere and Nichita [2014a]] for a detailed discussion on this matter). The \mathbf{Q} – red. method also requires repeated switch-backs, and the total number of iterations is 40; note that after the final switch to Newton iterations, the Euclidean norm first increases before the method converges quadratically in a few iterations.

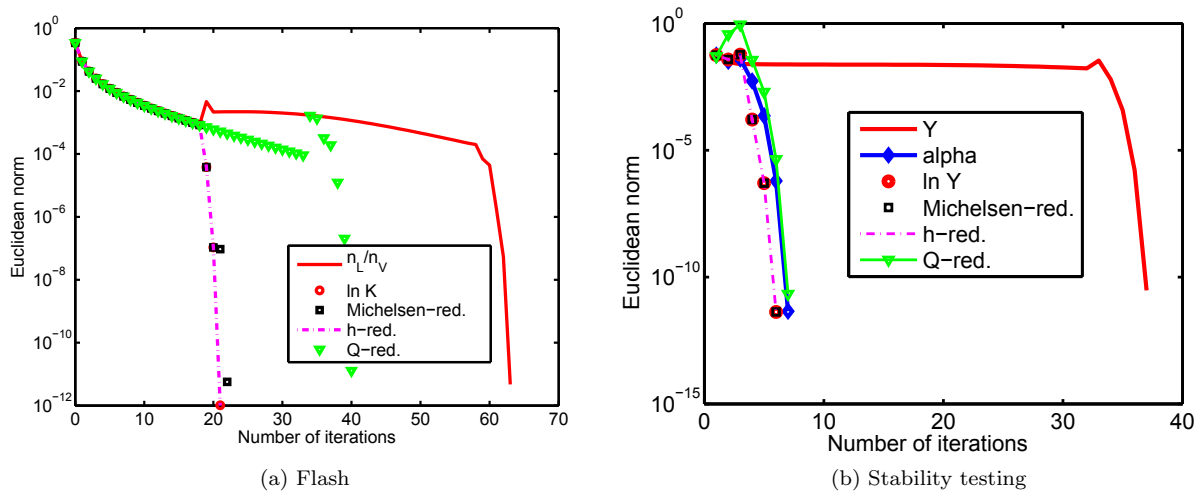


Figure 3.17: Error v.s. iteration number for various methods, for the Y8 mixture at $T=350$ K and $P=230$ bar

The Euclidean norm of the error vector is plotted vs. the iteration level for the stability testing of the

Y8 mixture (with $Y_i^{(0)} = z_i/K_i$) at $T=350$ K and $P=230$ bar (just above the STLL) in fig. 3.17b. The switching criterion [Hoteit and Firoozabadi [2006]] corresponds in this case to an early switch (except for the **Q – red.** method, in which only Newton iterations are performed. Only 6 or 7 iterations are required for convergence by both reduction methods and by all conventional methods (with **lnY**, **h – red.** and **Michelsen – red.** following the same convergence path), except **Y**, which requires almost 40 iterations, with repeated switch-backs to SSI iterations. At even more severe conditions, hundreds of SSI iterations (on switch back from Newton iterations or by using a very restrictive switching criterion) may be necessary before the Newton method converges. It was showed earlier [Nichita and Petitfrere [2013]] that formulations in **lnY** and Michelsen’s α variables are more robust than the one using formal mole numbers as variables and that the use of a Trust-region method highly improves robustness [Petitfrere and Nichita [2014a]] when Y_i are used as independent variables.

For both flash and stability problems, when using classical variables, the use of a Cholesky solver is recommended. For small systems, the differences with a flash using an LU decomposition are not significant (especially for flash calculations), but when nc exceeds 10, the differences increase and become quite important. The reduction based methods do not suffer the use of LU factorization (in the sense that the trend of time with nc remains linear) since the dimensionality of the system is decreased to a small size.

It is commonly said that in conventional methods the CPU time increases at least with the square of the number of components. This is observed in this work when an LU decomposition is used to solve the linear system of equations. However, a weak quadratic trend is revealed if a Cholesky decomposition is used: this observation is consistent with that of [Michelsen et al. [2013b]], who report an almost linear trend of CPU time with nc for classical variables (up to $nc = 25$ components). The choice of the linear solver influence to a great extent the efficiency of conventional methods, and it is recommended to take advantage of symmetry whenever possible.

For the flash calculation problem, the curves of CPU time vs. number of components corresponding to the conventional (**lnK – Chol.**, which is the best conventional method) and reduction methods (**h – red.**, which is the fastest reduction method) cross each other at a certain value of nc , thus, reduction methods are not more efficient than conventional ones for mixtures with few components (the range of compositional simulation). This is a common observation of the three recent publications on this matter [Michelsen et al. [2013b], Haugen and Beckner [2013], Gorucu and Johns [2013]]. Depending on how the comparison was done and on the implementation, the reported crossing point is around 15 in [Haugen and Beckner [2013]], around 20 in [Michelsen et al. [2013b]] (global time stability and flash using a Trust-region approach) and around 20 in [Gorucu and Johns [2013]]. In this work, the two curves cross each other around 22 components.

It is very important to note that if an LU decomposition is used, the reduction methods are faster than **lnK** even for a small number of components; a conventional method is faster than a reduction method only if full advantage of symmetry is taken (in the construction of the Jacobian matrix and the resolution of the linear system).

The use of pseudo-reduced methods was found (in this work, as well as in [Michelsen et al. [2013b]]) to be more efficient than conventional methods for flash and stability only for a relatively large number of components. Here it is also shown that full reduction methods are significantly more efficient than pseudo-reduced methods, having also a better conditioning of the linear system. We however recommend the use of the pseudo-reduced methods for mixtures with very many components; since the code can be easily written with minor modifications to an existing code based on conventional methods.

For stability analysis, the full reduction methods are the most efficient for any number of components. The use of reduction methods, especially **h – red.** (acting better than **Q – red.** in terms of speed and robustness), would be efficient for both reservoir simulation and process design number of components ranges.

Summarizing, conventional methods (when symmetry is used) seems to be more efficient in the range of number of components encountered in compositional reservoir simulation. With nc increasing, the reduction flash is more efficient (per iteration and global) above the crossing point. Thus, for any flash calculation involving mixtures with many components (such as in some process design simulations) both reduction methods give the best results in terms of efficiency, **h – red.** being more stable and efficient.

This work reveals that if we look separately at stability testing, stability reduction methods are more efficient than conventional methods for the entire range of number of components. Therefore, reduction methods for phase stability testing are suitable for compositional reservoir simulators.

Most of the CPU time in compositional reservoir simulations is spent by phase equilibrium calculations. Depending on the structure of the simulator, flash calculations may be part of a larger problem, but stability testing must be carried out in most grid blocks (at any time step and at each iteration level in the solution of the pressure equation) likely to experience a phase split. Phase equilibrium calculations can be either coupled with or decoupled from the equations describing the flow through porous media [Michelsen et al. [2013a]]. For 'coupled' simulators, only the stability testing is performed (except in the multiphase case when a very limited number of flashes may be required for initialization purposes); even though stability testing may be skipped in some single-phase blocks according to some simple criteria (using the 'shadow region' method) [Wong et al. [1990]], a very large number of calls for the stability routine is expected during a field-scale simulation run. In the case of 'decoupled' simulators, both stability testing and phase split are required. To the best of our knowledge, most of commercial and research compositional reservoir simulators are of a 'coupled' type, with no explicit flash calculations required. This is the reason why the observation that reduction methods for phase stability are more efficient than conventional ones for a small number of components is very important.

3.2.3.4 Conclusion

This work presents a comparison between conventional and reduction methods for phase equilibrium calculations, and is complementary with several recent papers [Michelsen et al. [2013b], Haugen and Beckner [2013], Gorucu and Johns [2013]] which questioned on the efficiency of the reduction methods as compared to conventional methods. Here, we have looked separately at the computational time spent by stability testing and flash calculations (per iteration and global). Besides evaluating the computational effort required by various conventional and reduction formulations, we also analyzed the condition number of the linear system of equations, the use of symmetry properties and the influence of the linear solver on efficiency. Some important links are formally established between conventional and reduced or pseudo-reduced methods and between different reduction methods.

Numerical experiments are carried out for several mixtures and for various formulations and sets of independent variables for conventional and reduction methods for flash calculations and stability analysis. The results show that flash calculations using the reduction methods are more efficient than conventional methods only for mixtures with many components (more than 20) and few non-zero binary interaction parameters (this is consistent with observations reported in [Michelsen et al. [2013b], Haugen and Beckner [2013], Gorucu and Johns [2013]]).

This means that reduction methods for flash calculations are suitable for process simulation (in which mixtures may contain hundreds of components), but there are not suitable for compositional simulation, where

the number of components is usually limited. Pseudo-reduced methods are also found to be more efficient than conventional methods only for many components, and may be attractive since they require minor modification in existing codes (pseudo-reduction methods are however far less efficient than full reduction methods).

The main conclusion of this work concerns the range of number of components (less than a dozen) typical for compositional reservoir simulation. While the reduced flash appears less efficient than conventional flash in this range (as also reported in several recent studies), the reduced stability is more efficient than the conventional stability, even for a small number of components. This is extremely important since in most compositional reservoir simulators only the stability testing is performed, since no explicit phase split is required, the flash being part of a bigger problem (coupled with flow equations).

3.2.4 A new reduction method for multiphase equilibrium calculations

3.2.4.1 Introduction

Many authors investigated the multiphase-split calculation problem with conventional variables (in the compositional space). [Risnes et al. [1981]], showed a successive substitution method for multiphase equilibrium problems. [Nghiem and Heidemann [1982]] developed an acceleration procedure for multiphase calculation. The same year, [Mehra et al. [1982]] applied a multiphase flash to compositional simulations. Later on, [Michelsen [1994]] proposed a multiphase equilibrium procedure, and provided initial guesses for multiphase problems. More recently, [Li and Firoozabadi [2012]] extended the number of initial guesses for stability analysis to propose a solution to general systems.

Conventional variables are the most widely used in chemical processing and reservoir simulators. However, recently, reduction variables have been applied and show a good efficiency for those problems. [Michelsen [1986]] is the first to introduce reduction space variables with a three independent variables flash setting all binary interaction parameters (BIPs) to zero. [Jensen and Fredenslund [1987]] used five independent variables in the case of only one component having non-zero BIPs with the remaining ones. [Li and Johns [2006]] proposed another version using a least-square fitting of the binary interaction coefficient matrix leading to a rank two approximated BIP matrix.

[Hendriks [1988]] and [Hendriks and van Bergen [1992]] developed a reduction based on the spectral decomposition of the quadratic form of the BIP matrix. This decomposition was in the heart of new reduction variables such as [Nichita and Minescu [2004]], [Nichita [2006a]]. Finally, [Nichita and Graciaa [2011]] developed a new reduction to deal with two-phase equilibrium calculations. Unlike all previous reduction method, the proposed set of independent variables allows a direct minimization of the Gibbs free energy and leads to an unconstrained problem, with unbounded variables. [Nichita and Petitfrere [2013]] proposed to solve the stability analysis with this new set. [Gorucu and Johns [2014]] extended [Li and Johns [2006]] method by implementing this new set of reduction variables for the two phase-split case.

Comparisons were made between conventional and classical methods. [Michelsen et al. [2013b]] showed computational time for different number of components for the global two phase flash (stability plus phase-split) and compared reduction and conventional variables, in stand-alone calculations. [Haugen and Beckner [2013]] also made comparisons for reservoir simulation test cases and showed the interest of using reduction variables. More recently, [Mohebbinia et al. [2013]] applied [Li and Johns [2006]] method to multiphase flash and showed that the procedure to perform the whole equilibrium flash using the full multiphase reduction variables was almost always time earning as compared with classical variables for any number of components.

Reduction methods deriving from a spectral decomposition allow to decrease the system from $nc \times (np - 1)$

to $m \times (np - 1)$. With an increasing number of phases, [Mohebbinia et al. [2013]] logically showed that the differences between conventional and reduction methods were more and more in favor of the reduction in terms of computational time. They made comparisons for two phase-three phase, four phases.

In this subsection, a direct extension of the two-phase flash reduction method of [Nichita and Graciaa [2011]] is made to deal with multiphase split calculations for any number of phases. Two versions of the algorithm have been developed which only differs from the way the Jacobian matrix is built. The first one is a direct extension of [Nichita and Graciaa [2011]]'s method. The second one makes use of the symmetry of different matrix and allows decreasing the computational time by computing only about half of the elements. Comparisons of the computational times between conventional variables and the new reduction set will be shown to perform one Newton iteration and to compute full three- and four phase regions with different mixtures having different BIP family numbers. A parallel between reduction methods and conventional variables will be made formally, and numerically. Finally, the convergence behavior will be examined for different set of variables.

3.2.4.2 Reduction multiphase flash

Two reduction methods are presented:

- A direct extension of [Nichita and Graciaa [2011]] two-phase flash to multiphase equilibrium (denoted **h – red. – d.**)
- A constrained minimization of Gibbs free energy, which takes advantage of symmetry (denoted **h – red. – s.**).

Both methods lead to the same Newton equation; they differ in the way the Jacobian matrix is constructed.

3.2.4.2.1 Direct extension of Nichita and Graciaa's (2011) reduction method

Using (eq. 3.60) and $\ln K_{ik} = \ln \phi_{iR} - \ln \phi_{ik}$, the natural logarithm of equilibrium constants can be expressed in terms of

$$\bar{h}_{\alpha k} = h_{\alpha R}(\mathbf{Q}_R) - h_{\alpha k}(\mathbf{Q}_k); \quad \alpha = 1, M+1; \quad k = 1, np; k \neq R \quad (3.106)$$

as

$$\ln K_{ik} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \bar{h}_{\alpha k}; \quad i = 1, nc; k = 1, np; k \neq R \quad (3.107)$$

or

$$\ln \mathbf{K}_k = \mathbf{C}^T \bar{\mathbf{h}}_k \quad (3.108)$$

with $\bar{\mathbf{h}}_k = (\bar{h}_{1k}, \bar{h}_{2k}, \dots, \bar{h}_{M+1,k})^T; k = 1, np; k \neq R$ and $\mathbf{Q}_k = (Q_{1k}, Q_{2k}, \dots, Q_{M+1,k})^T; k = 1, np$.

This suggest the following iterative sequence to update the $(np - 1) \times (M + 1)$ independent variables $\bar{\mathbf{h}} = (\bar{\mathbf{h}}_1^T, \dots, \bar{\mathbf{h}}_k^T, \dots, \bar{\mathbf{h}}_{np}^T)^T; k \neq R$:

Given $\bar{\mathbf{h}} \rightarrow \ln \mathbf{K}_k$ (eq. 3.108) $\rightarrow \boldsymbol{\theta}$ (eq. 2.30) $\rightarrow \mathbf{x}_k$ (eq. 2.28) $\rightarrow \mathbf{Q}_k$ (eq. 3.47) $\rightarrow \mathbf{h}$ (eq. 3.54, eq. 3.55, eq. 3.56) \rightarrow New $\bar{\mathbf{h}}$ (eq. 3.106).

The Newton iteration equation is

$$\mathbf{J}^R \Delta \bar{\mathbf{h}} = -\mathbf{e} \quad (3.109)$$

where \mathbf{e} is the error vector $\mathbf{e} = (\mathbf{e}_1^T, \dots, \mathbf{e}_k^T, \dots, \mathbf{e}_{np}^T)^T$; $k \neq R$ obtained by concatenating the vectors $\mathbf{e}_k = (e_{1k}, e_{2k}, \dots, e_{M+1,k})^T$; $k = 1, np$; $k \neq R$ of elements (eq. 3.106)

$$e_{\alpha k} \equiv \bar{h}_{\alpha k} + h_{\alpha k}(\mathbf{Q}_k) - h_{\alpha R}(\mathbf{Q}_R) = 0; \quad \alpha = 1, M+1; k = 1, np; k \neq R \quad (3.110)$$

and \mathbf{J}^R is the Jacobian matrix in the reduction method, of elements

$$J_{\alpha\beta, kp}^R = \frac{\partial e_{\alpha k}}{\partial \bar{h}_{\beta p}} = \delta_{\alpha\beta} \delta_{kp} + \frac{\partial h_{\alpha k}}{\partial \bar{h}_{\beta p}} - \frac{\partial h_{\alpha R}}{\partial \bar{h}_{\beta p}}; \alpha, \beta = 1, M+1; k, p = 1, np; k, p \neq R \quad (3.111)$$

The partial derivatives in (eq. 3.111) are (see Appendix H)

$$\frac{\partial h_{\alpha k}}{\partial \bar{h}_{\beta p}} = \sum_{\gamma=1}^M \frac{\partial h_{\alpha k}}{\partial Q_{\gamma k}} \frac{\partial Q_{\gamma k}}{\partial \bar{h}_{\beta p}}; \quad \alpha, \beta = 1, M+1; k, p = 1, np; p \neq R \quad (3.112)$$

and finally the elements of the Jacobian matrix are

$$J_{\alpha\beta, kp}^R = \delta_{\alpha\beta} \delta_{kp} + \sum_{\gamma=1}^M \frac{\partial h_{\alpha k}}{\partial Q_{\gamma k}} \frac{\partial Q_{\gamma k}}{\partial \bar{h}_{\beta p}} - \sum_{\gamma=1}^M \frac{\partial h_{\alpha R}}{\partial Q_{\gamma R}} \frac{\partial Q_{\gamma R}}{\partial \bar{h}_{\beta p}} \quad \alpha, \beta = 1, M+1; k, p = 1, np; k, p \neq R \quad (3.113)$$

The stopping criteria are $S < \varepsilon$ in term of the Euclidean norm of the gradient vector

$$S = \sqrt{\sum_{\substack{k=1 \\ k \neq R}}^{np} \sum_{\alpha=1}^{M+1} (g_{\alpha k}^R)^2} \quad (3.114)$$

and $S_f < \varepsilon_f$ in term of the Euclidean norm of the errors in fugacities

$$S_f = \sqrt{\sum_{\substack{k=1 \\ k \neq R}}^{np} \sum_{i=1}^{nc} (f_{ik} - f_{iR})^2} \quad (3.115)$$

3.2.4.2.2 Constrained minimization of the Gibbs free energy

The modified reduction parameters are defined as [Kaul and Thrasher [1996]]

$$\bar{Q}_{\alpha k} = \theta_k Q_{\alpha k} = \sum_{i=1}^{nc} q_{\alpha i} n_{ik}; \quad \alpha = 1, M+1; k = 1, np; k \neq R \quad (3.116)$$

with $\bar{Q}_{M+1,k} = \theta_k$ since $q_{M+1,i} = 1$; $i = 1, nc$. In matrix form, (eq 3.116) is

$$\bar{\mathbf{Q}}_k = \mathbf{C} \mathbf{n}_k \quad (3.117)$$

where $\bar{\mathbf{Q}}_k = (\bar{Q}_{1k}, \bar{Q}_{2k}, \dots, \bar{Q}_{M+1,k}, \theta_k)^T$; $k = 1, np$; $k \neq R$.

The dimensionless Gibbs free energy (the objective function) for a multiphase system can be expressed as

$$G(\mathbf{n}, \bar{\mathbf{Q}}) = G_I(\mathbf{n}) + G_E(\bar{\mathbf{Q}}) \quad (3.118)$$

where subscripts I and E denote ideal and excess terms, respectively; G_I depends only on mole numbers and G_E depends only on modified reduction parameters, $\bar{\mathbf{Q}} = (\bar{\mathbf{Q}}_1^T, \dots, \bar{\mathbf{Q}}_k^T, \dots, \bar{\mathbf{Q}}_{np}^T)^T$; $k \neq R$:

$$G_I = \sum_{k=1}^{np} \sum_{i=1}^{nc} n_{ik} \ln x_{ik} = \sum_{k=1}^{np} \sum_{i=1}^{nc} n_{ik} \ln \left(\frac{n_{ik}}{\sum_i n_{ik}} \right) = G_I(\mathbf{n}) \quad (3.119)$$

and

$$\begin{aligned} G_E &= \sum_{k=1}^{np} \sum_{i=1}^{nc} n_{ik} \ln \phi_{ik} = \sum_{k=1}^{np} \theta_k \sum_{i=1}^{nc} x_{ik} \ln \phi_{ik} = \sum_{k=1}^{np} \theta_k g_{E,k}(\mathbf{Q}_k) \\ &= \sum_{k=1}^{np} \bar{Q}_{M+1,k} g_{E,k} \left(\frac{\bar{\mathbf{Q}}_k}{\bar{Q}_{M+1,k}} \right) = G_E(\bar{\mathbf{Q}}_k) \end{aligned} \quad (3.120)$$

where $g_{E,k}$ the molar excess Gibbs free energy of the phase k (see eq. 1.93 and eq. D.4).

If G is minimized subject to the constraints given by (eq 3.116), the Lagrangian function is

$$L(\mathbf{n}, \bar{\mathbf{Q}}, \bar{\mathbf{h}}) = G(\mathbf{n}, \bar{\mathbf{Q}}) - \sum_{\substack{k=1 \\ k \neq R}}^{np} \sum_{\alpha=1}^{M+1} \bar{h}_{\alpha k} \left(\sum_{i=1}^{nc} q_{\alpha i} n_{ik} - \bar{Q}_{\alpha k} \right) \quad (3.121)$$

where $\bar{\mathbf{h}}$ is the vector of Lagrange multipliers.

From

$$\frac{\partial G_I}{\partial n_{ik}} = \ln K_{ik}; \quad i = 1, nc; k = 1, np; k \neq R \quad (3.122)$$

and

$$\frac{\partial L}{\partial n_{ik}} = \ln K_{ik} - \sum_{\alpha=1}^{M+1} q_{\alpha i} \bar{h}_{\alpha k} = 0; \quad i = 1, nc; k = 1, np; k \neq R \quad (3.123)$$

we obtain

$$\ln K_{ik} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \bar{h}_{\alpha k}; \quad i = 1, nc; k = 1, np; k \neq R \quad (3.124)$$

which is exactly (eq. 3.108), that is, the key equation in the direct extension of [Nichita and Graciaa [2011]]'s reduction method.

The elements of the gradient vector $\mathbf{g}^R = [(\mathbf{g}_1^R)^T, \dots, (\mathbf{g}_k^R)^T, \dots, (\mathbf{g}_{np}^R)^T]^T; k \neq R$ with

$$\mathbf{g}_k^R = -\frac{\partial G}{\partial \bar{\mathbf{Q}}_k} \quad (3.125)$$

are (see Appendix I)

$$g_{\alpha k}^R = \bar{h}_{\alpha k} + h_{\alpha k}(\mathbf{Q}_k) - h_{\alpha R}(\mathbf{Q}_R) = 0; \alpha = 1, M+1; k = 1, np; k \neq R \quad (3.126)$$

which are exactly the error equations in the reduction method of [Nichita and Graciaa [2011]], see (eq 3.116).

The Hessian matrix is (its block kp)

$$\mathbf{H}_{kp}^R = \frac{\partial^2 G}{\partial \bar{\mathbf{Q}}_k \partial \bar{\mathbf{Q}}_p} \quad (3.127)$$

and it can be expressed as the sum of two ideal and excess parts

$$\mathbf{H} = \mathbf{H}_I^R + \mathbf{H}_E^R \quad (3.128)$$

The elements of \mathbf{H}_I^R and \mathbf{H}_E^R are given in Appendix J.

The Newton iteration equation is

$$(\mathbf{H}_I^R + \mathbf{H}_E^R) \Delta \bar{\mathbf{Q}} = -\mathbf{g}^R \quad (3.129)$$

The relation between $\Delta \bar{\mathbf{Q}}$ and $\Delta \bar{\mathbf{h}}$ is

$$\Delta \bar{\mathbf{Q}}_k = \sum_p \frac{\partial \bar{\mathbf{Q}}_k}{\partial \bar{\mathbf{h}}_p} \Delta \bar{\mathbf{h}}_p; k = 1, np; k \neq R \quad (3.130)$$

where (eq. J.3)

$$\frac{\partial \bar{\mathbf{Q}}_k}{\partial \bar{\mathbf{h}}_p} = [\mathbf{H}_I^R]_{kp}^{-1} \quad (3.131)$$

Introducing (eq. 3.131) into (eq. 3.130) and the result in (eq. 3.129) gives

$$(\mathbf{I} + \mathbf{H}_E^R [\mathbf{H}_I^R]^{-1}) \Delta \bar{\mathbf{h}} = -\mathbf{g}^R \quad (3.132)$$

or

$$\mathbf{J}^R \Delta \bar{\mathbf{h}} = -\mathbf{g}^R \quad (3.133)$$

where the block structure of the Jacobian matrix is

$$\begin{pmatrix} \mathbf{J}_{11} & \cdots & \mathbf{J}_{1,nc} \\ \vdots & \ddots & \vdots \\ \mathbf{J}_{nc,1} & \cdots & \mathbf{J}_{nc,nc} \end{pmatrix} = \begin{pmatrix} \mathbf{I}_{M+1} & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & \cdots & \mathbf{I}_{M+1} \end{pmatrix} + \begin{pmatrix} \mathbf{H}_{E,1,1}^R & \cdots & \mathbf{H}_{E,1,nc}^R \\ \vdots & \ddots & \vdots \\ \mathbf{H}_{E,nc,1}^R & \cdots & \mathbf{H}_{E,nc,nc}^R \end{pmatrix} \begin{pmatrix} [\mathbf{H}_I^R]_{1,1}^{-1} & \cdots & [\mathbf{H}_I^R]_{1,nc}^{-1} \\ \vdots & \ddots & \vdots \\ [\mathbf{H}_I^R]_{nc,1}^{-1} & \cdots & [\mathbf{H}_I^R]_{nc,nc}^{-1} \end{pmatrix} \quad (3.134)$$

its block kp is

$$\mathbf{J}_{kp}^R = \mathbf{I}_{M+1} + \sum_m^{nc} \mathbf{H}_{E,km}^R [\mathbf{H}_I^R]_{mp}^{-1} \quad (3.135)$$

and its elements are

$$J_{\alpha\beta, kp}^R = \delta_{\alpha\beta} \delta_{kp} + \sum_{\substack{m=1 \\ m \neq R}}^{np} \sum_{\gamma=1}^{M+1} H_{E,\alpha\gamma, km}^R [\mathbf{H}_I^R]_{\gamma\beta, mp}^{-1}; \alpha, \beta = 1, M+1; k, p = 1, np; k, p \neq R \quad (3.136)$$

or

$$J_{\alpha\beta, kp}^R = \delta_{\alpha\beta} \delta_{kp} + \sum_{\substack{m=1 \\ m \neq R}}^{np} \sum_{\gamma=1}^{M+1} \left(\frac{\partial h_{\alpha k}}{\partial \bar{Q}_{\gamma m}} - \frac{\partial h_{\alpha R}}{\partial \bar{Q}_{\gamma m}} \right) \frac{\partial \bar{Q}_{\gamma m}}{\partial \hat{h}_{\beta p}}; \alpha, \beta = 1, M+1; k, p = 1, np; k, p \neq R \quad (3.137)$$

The Jacobian matrix is non-symmetric, but one can take advantage of symmetry of matrices \mathbf{H}_E^R and $[\mathbf{H}_I^R]^{-1}$. In building the Jacobian matrix, the construction of the matrices $[\mathbf{H}_I^R]_{kp}^{-1} = \mathbf{C} \mathbf{U}_{kp}^{-1} \mathbf{C}^T$ (eq. J.3) is the most time consuming part. This makes the dependence between computational time and the number of components to be weakly quadratic (inherited from conventional methods via \mathbf{U}^{-1}) rather than linear as

usually in reduction methods. Instead, we can use the following equations to relate $([H_I^R]^{-1})_{\alpha\beta, kp}$ to the partial derivatives $\frac{\partial Q_{\alpha k}}{\partial h_{\beta p}}$ and $\frac{\partial \theta_k}{\partial h_{\beta p}}$

$$\frac{\partial \bar{Q}_{\alpha k}}{\partial h_{\beta p}} = \theta_k \frac{\partial Q_{\alpha k}}{\partial h_{\beta p}} + Q_{\alpha k} \frac{\partial \theta_k}{\partial h_{\beta p}}; \alpha = 1, M; \beta = 1, M+1; k, p = 1, np; k, p \neq R \quad (3.138)$$

and

$$\frac{\partial \bar{Q}_{M+1, k}}{\partial h_{\beta p}} = \frac{\partial \theta_k}{\partial h_{\beta p}}; \beta = 1, M+1; k, p = 1, np; k, p \neq R \quad (3.139)$$

It had been proven [Nichita and Petitfrere [2013], Petitfrere and Nichita [2014b]] that the Newton iterations in the reduction methods using the Lagrange multipliers as independent variables have the same convergence paths as the conventional method with $\ln \mathbf{K}$ variables.

3.2.4.3 Results

3.2.4.3.1 Mixtures

Three mixtures were tested. The Maljamar reservoir oil, the Maljamar separator oil from [Orr et al. [1981]] and the sour gas mixture from [Robinson et al. [1978]]. All relevant data for these mixtures are taken from [Li and Firoozabadi [2012]]. These three mixtures are mixed in different proportions with carbon dioxide to obtain three- and four-phase regions; fig. 3.18a depicts the p-z phase envelope of the Maljamar separator oil mixed with CO₂ at T=305.35 K and fig. 3.18b the p-z phase envelope of the Maljamar reservoir oil mixed with CO₂ at the same temperature. Phase envelopes of the sourgas-CO₂ (at T=178.8K) is given in fig. 3.19a. Finally, taking the sour gas-CO₂ mixture at a lower temperature, T=123.15 K, a four phase region can be seen whose phase boundary is given in fig. 3.19b. It will be called the sourgas4 test case to differentiate from the three phase sour gas test case at T=305.35 K. Note that all mixtures are exhibiting complicated phase envelopes; some calculations are performed at extremely difficult conditions, such as the vicinity of bi-critical points. The Peng-Robinson equation of state is used in all calculations.

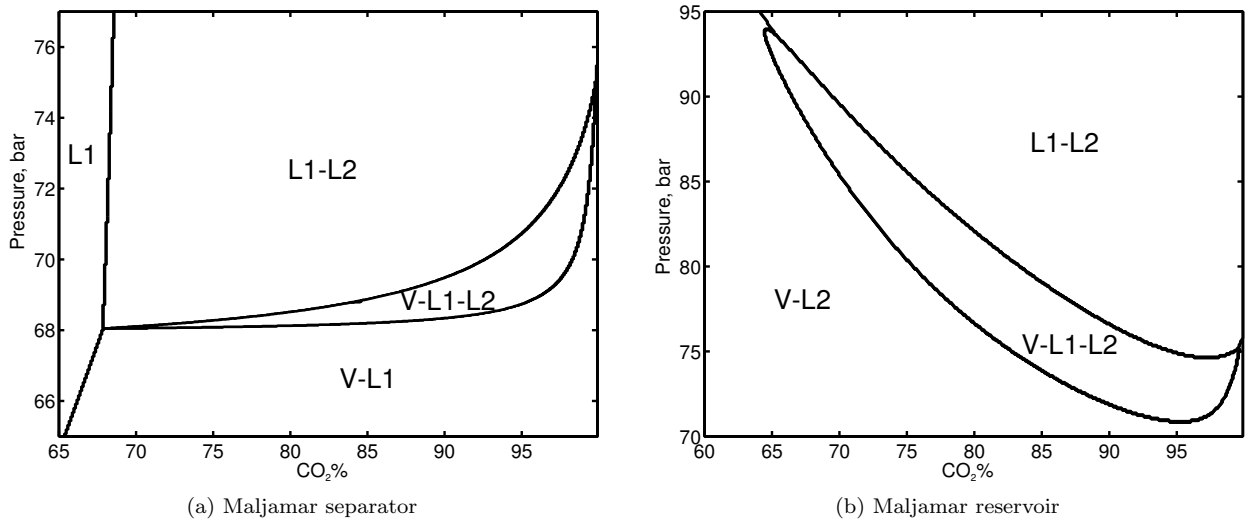


Figure 3.18: Phase envelopes of different mixtures mixed with CO₂ at T=305.35 K

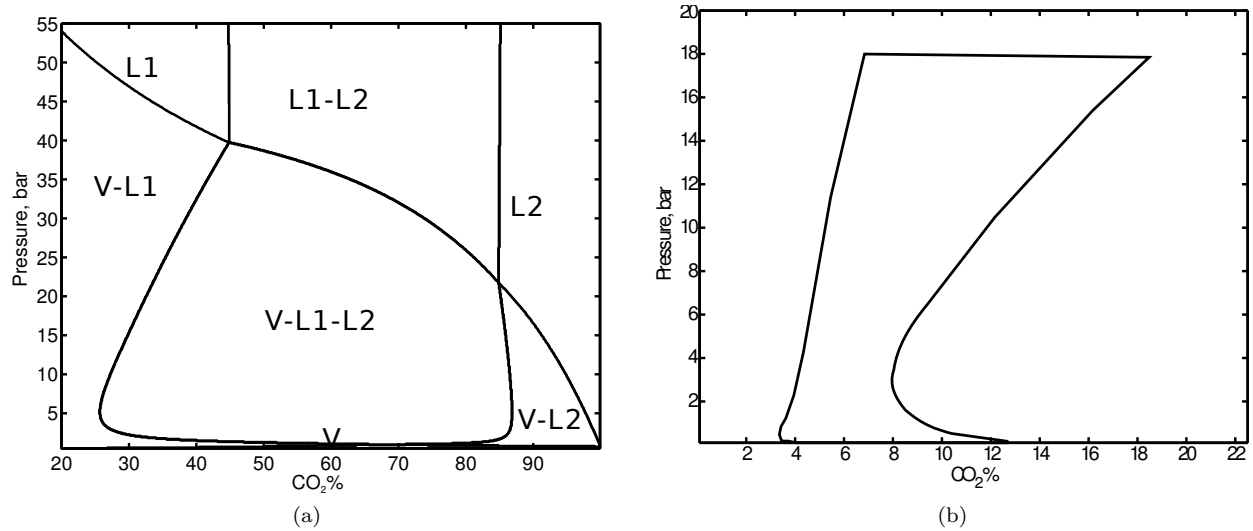


Figure 3.19: b)Phase boundaries of a) the Sour gas mixture at $T=178.8\text{K}$ b) the four-phase region for the sourgas4 mixture at $T=305.35\text{ K}$

3.2.4.3.2 Independent variables

The same solvers as in subsection 3.2.3.2.1 and independent variables as in the subsection 3.2.3.2.2 were used to test the new reduction variables proposed in this section. The direct extension of the two-phase reduction from [Nichita and Graciaa [2011]] (**h-red.**) will be called **h-red.-d.** and the new symmetric based reduction **h-red.-s.** Both reduction variables lead to a non-symmetric Jacobian. The linear system is solved based on a LU factorization.

3.2.4.3.3 Computational time

In this section, the computation time spent by Newton iterations in phase split calculations for np phases ($np = 3$ for the sour gas, the Maljamar separator and the Maljamar reservoir mixtures and $np = 4$ for the sourgas4) are evaluated.

3.2.4.3.3.a Time spent for one iteration

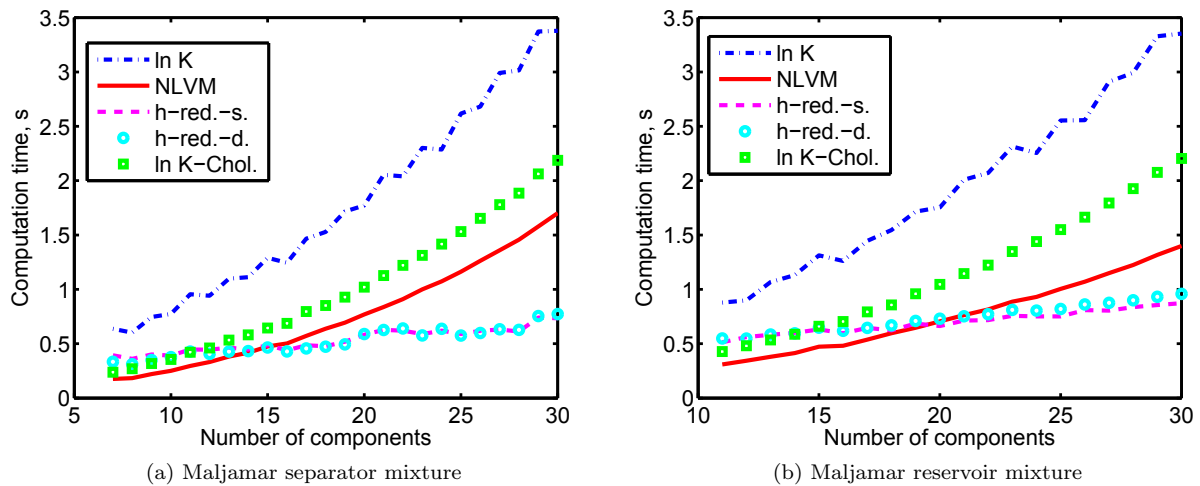


Figure 3.20: Computational time to repeat 10000 times a Newton iteration

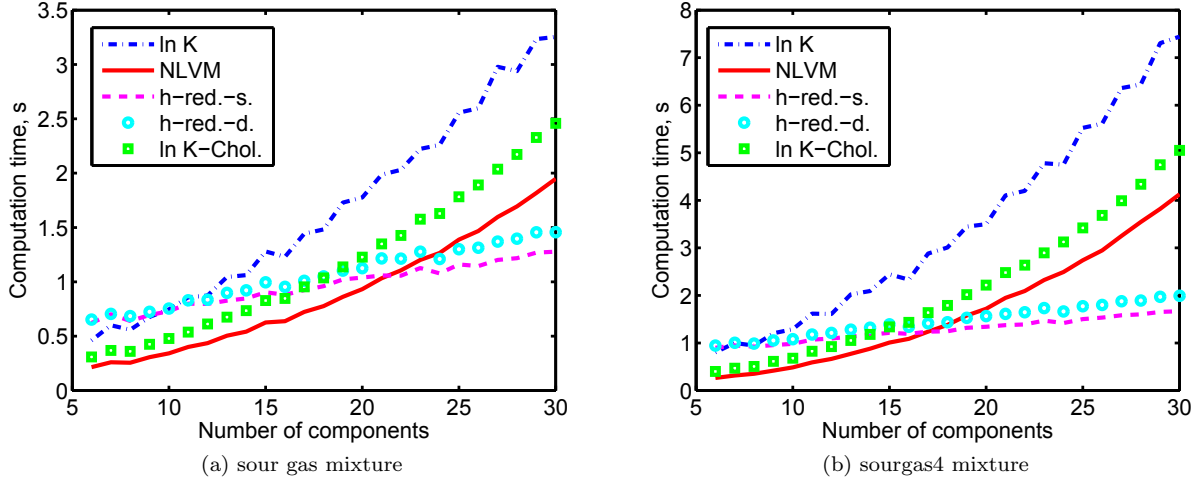
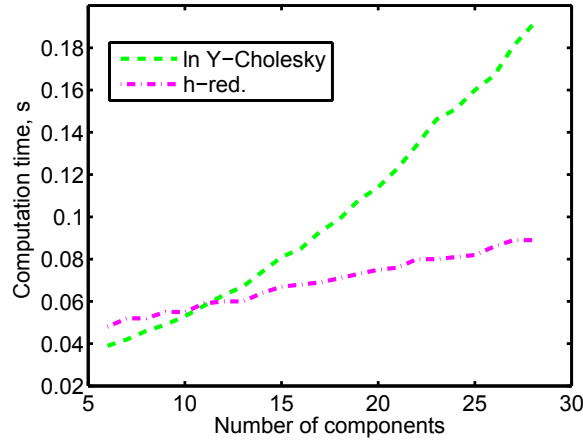


Figure 3.21: Computational time to repeat 10000 times a Newton iteration

The computational time spent to repeat 10000 times the same Newton iteration (the first one after the switch from SSI iterations at $S_f \leq 10^{-2}$) is recorded for each set of variables. This operation is repeated for different numbers of components. This gives the time spent to perform one Newton iteration, independently from the convergence behavior of different methods. In fig. 3.20a the computational time is given for the Maljamar separator mixture at $p = 69.5$ bar and 95% of CO_2 , in fig. 3.20b the time is given for the Maljamar reservoir mixture at $p = 73$ bar and 95% of CO_2 and in fig. 3.21a for the sour gas at $p = 20$ bar and 40% of CO_2 . Finally, in fig. 3.21b the time is given for the Sourgas4 mixture at $p = 15.1$ bar and 8% of CO_2 . The time to repeat 10000 a Newton iteration in the phase stability analysis (for the conventional $\ln \mathbf{Y}$ and the reduction $\mathbf{h} - \text{red.}$ methods) for the sour gas mixture (3 BIPs) at $p = 20$ bar and 40% of CO_2 is given in fig. 3.22.

Figure 3.22: Computational time vs. number of components for the stability analysis of the sour gas mixture (10000 iterations at $T = 178.8$ K and $P = 20$ bar)

3.2.4.3.3.b Time spent to perform flash calculations within the entire three/four phase region

Flash calculations are performed for different P - z conditions covering the whole np phase region for each mixture. For the sour gas mixture, the pressure goes from 5 to 35 bar with $\Delta p = 0.5$ bar, the CO_2 composition ranges from 30 to 80% with $\Delta z = 2\%$. For the Maljamar separator mixture, the pressures lie between 68 and 74 bar with $\Delta p = 0.03$ bar and the CO_2 composition ranges from 85 to 99.9 % with a computation every

0.03%. For the Maljamar reservoir, $70 \text{ bar} \leq p \leq 87 \text{ bar}$ and $\Delta p = 0.3 \text{ bar}$ and for the CO_2 composition, $75 \% \leq z_{\text{CO}_2} \leq 99.9\%$ and $\Delta z = 0.3\%$. Finally, for the sourgas4 mixture, the tested pressures lie between $p = 0.1 \text{ bar}$ and $p = 18 \text{ bar}$ with $\Delta p = 0.5 \text{ bar}$ and the CO_2 composition ranges from 5% to 15% with $\Delta z = 0.2\%$.

For each condition and each set of variables, the computational time spent for the Newton iterations is recorded between the switch from SSI iterations (at $S_f \leq 10^{-2}$) and convergence. The iterations are ended as soon as $S_f \leq 10^{-7}$. Since the gradient is different in conventional and reduction methods, for some conditions, the convergence criterion can be satisfied in the reduced norm ($S \leq 10^{-10}$), without being fulfilled in the conventional norm. In this case, we end up the calculations since the reduction method has converged in its space, and any gradient based optimization method would not decrease the error any further. Whenever a Newton step does not decrease the Gibbs free energy, a switch back to the SSI procedure is performed.

Computational times for each p-z point are added to get the global computational time plotted in fig. 3.23a and fig. 3.23b for the Maljamar mixtures and in fig. 3.24a and fig. 3.24b for the sour gas mixtures. The average number of iterations to perform flash calculations using **lnK**, **lnK – Chol.** and **h – red.** within the whole np phase region is given in table 3.4 for the four mixtures.

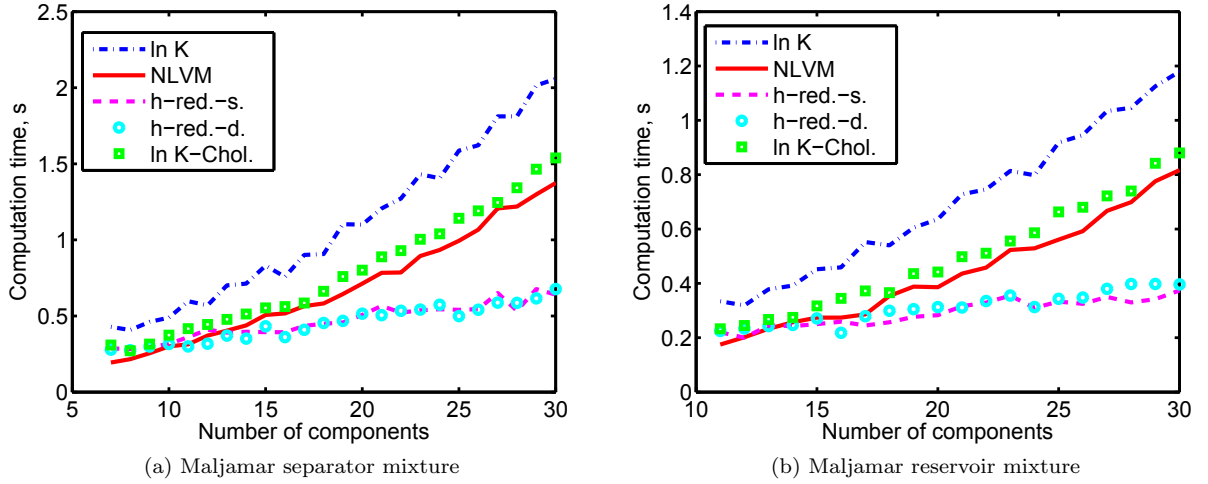


Figure 3.23: Computational time for the entire three phase region

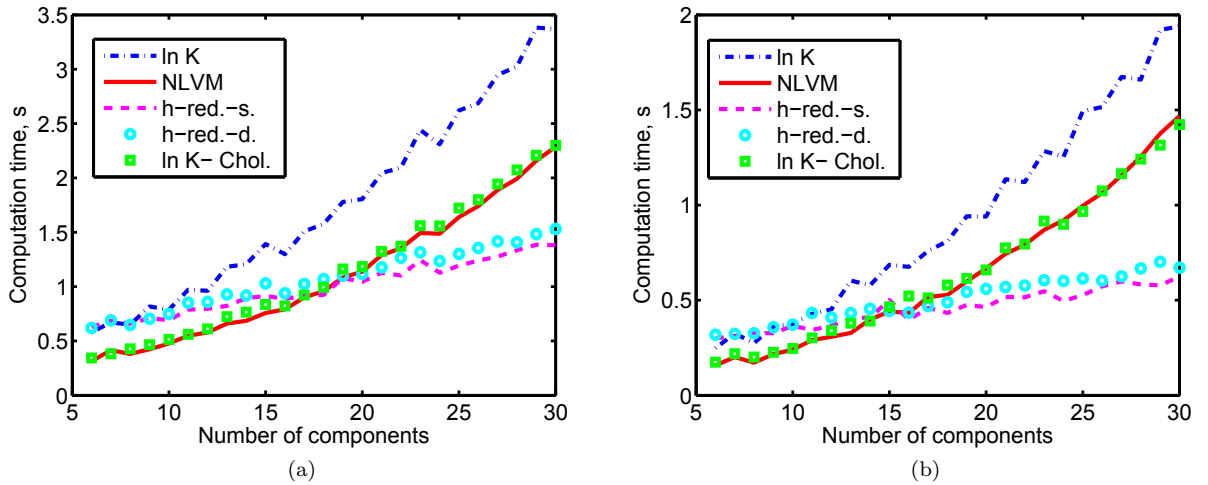


Figure 3.24: Computational time for a) the entire three phase region for the sour gas mixture; b) the entire four phase region for the sourgas4 mixture

3.2.4.3.4 Convergence behavior

In this section, qualitative comparisons of the convergence properties of the different methods are carried out. All the np flashes are initialized based on previous phase splits and stability analysis. SSI iterations are performed as long as $S > 10^{-2}$ and then Newton iterations are used. In fig. 3.25, at $T=305.35$ K, $p = 69$ bar and 95% of CO_2 , the convergence of different three-phase flash methods is analyzed for the Maljamar reservoir mixture. The Euclidean norm S_f is plotted against the iteration number until convergence. Fig. 3.26a shows the number of iterations required for convergence for the three-phase flash region, for the Maljamar separator mixture, for three sets of variables: **h** – red., **nL/nV** and **lnK**. The conditions are taken at 95% of CO_2 and the pressure varies from 68 bar to 70.7 bar. The phase mole fractions along this pressure range are given in fig. 3.26b. Fig. 3.27a shows the number of iterations required to reach convergence for the three-phase flash for the sour gas mixture with the same three variables. The conditions are taken at $p = 20$ bar and the CO_2 composition varies from 10% to 85%. The phase distribution is given fig. 3.27b. Finally, fig. 3.28a shows the number of iterations required for convergence of the four-phase flash for the sourgas4 mixture. The conditions are taken at 7.4% of CO_2 and the pressure varies from 0.1 bar to 18 bar. The phase mole fractions are given in fig. 3.28b.

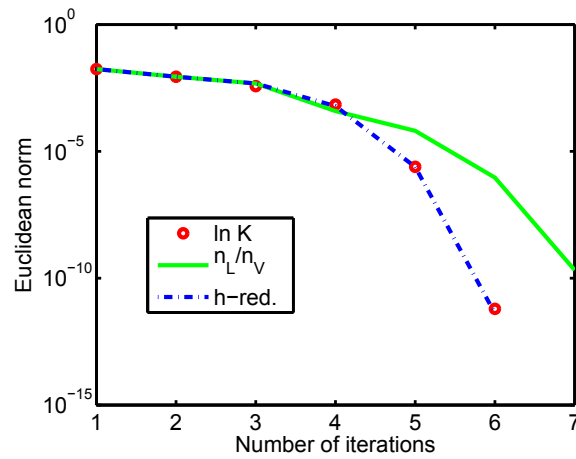


Figure 3.25: Error vs number of iterations at $P=69$ bar and 95% of CO_2 for the Maljamar reservoir mixture

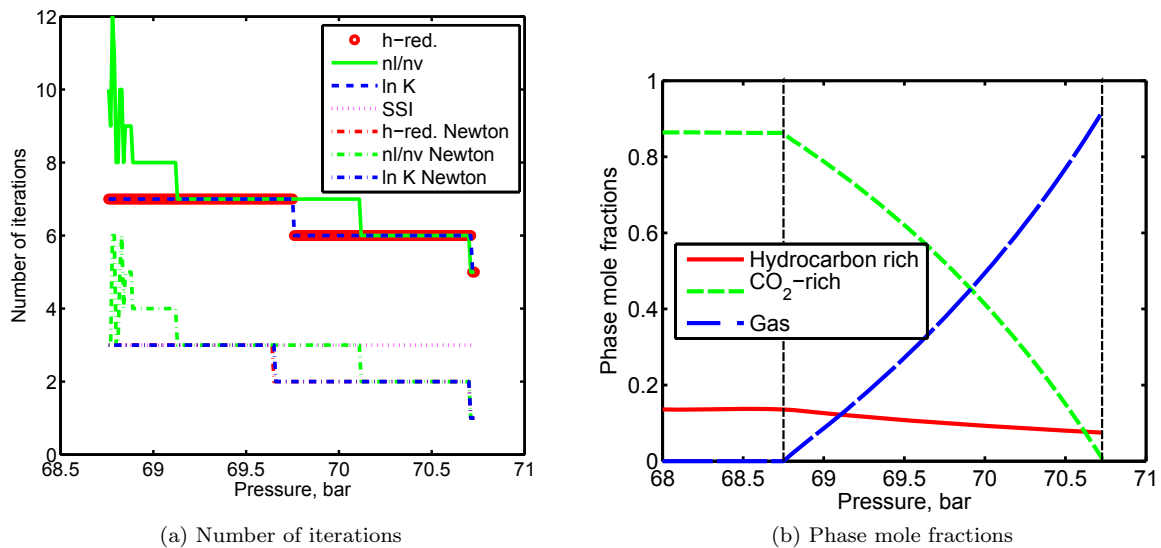
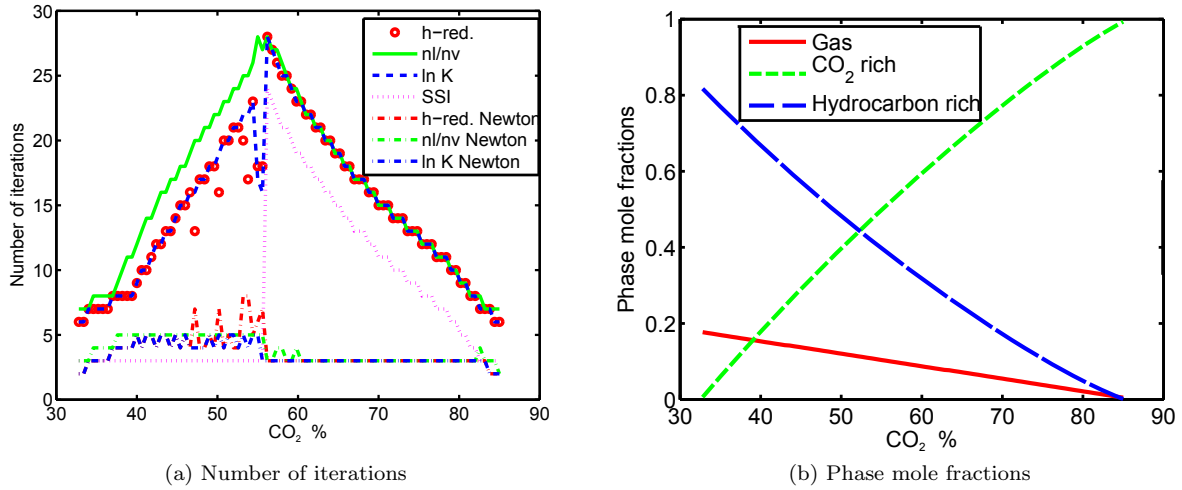
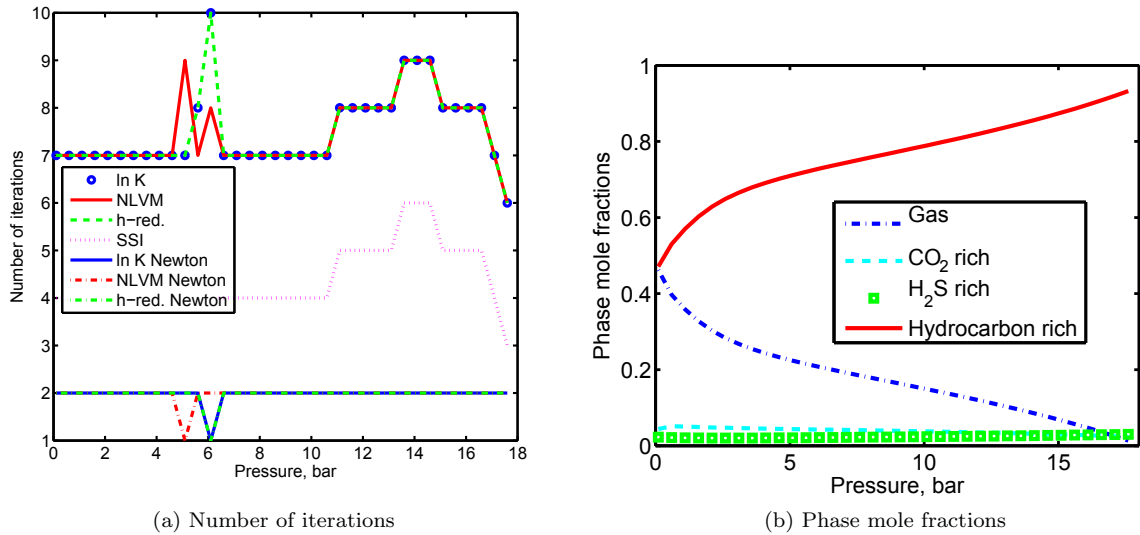


Figure 3.26: Three-phase flash of the Maljamar separator mixture at 95% CO_2

Figure 3.27: Three-phase flash of the sour gas mixture at $P=20$ barFigure 3.28: Four-phase flash of the sour gas mixture at 7.4% CO₂

3.2.4.4 Discussion

From fig. 3.20a to fig. 3.21b and from fig. 3.23a to fig. 3.24b, it is observed that the curves of CPU time vs. nc are crossing each other at slightly different values of nc if we look at the time per Newton iteration and at the time to perform Newton iterations over the entire np phase region. There are several explanations for these differences:

- If a Newton iteration does not decrease the Gibbs free energy, a switch back to SSI is performed (but the time to compute second-order information and to solve the linear system is recorded)
- Theoretically the convergence path of **lnK** and **h-red.** methods is the same (see for instance in fig. 3.25 that the curves are overlapping at each iteration); in practice, this is the case for most of the points in the three-phase region.

However, for difficult conditions, **h-red.** may require more iterations for convergence than the conventional **lnK** method (see certain differences in the average number of iterations in table 3.4. This is due to the approximation errors (the condition number in reduction methods has been revealed by [Petitfrere and

Nichita [2014b]) to be larger than in the conventional **lnK** method); iii) Besides, the norm is not the same for both methods. Sometimes, the convergence is achieved in the reduced space (using the norm S) but is not achieved in the compositional space (using the norm S_f). In these cases, calculations are stopped because the reduction method has converged in its space. This explains why the average number of iterations can be in some cases (i.e., for the Sourgas4 in table 3.4) smaller for **h – red.** than for **lnK**. Fig. 3.26a, fig. 3.27a and fig. 3.28a also show that the reduction method **h – red.** behaves in general similarly to **lnK**, with few exceptions.

Mixture	ln K	ln K-Chol.	h-red.
Maljamar separator	6.3377	6.3377	6.3385
Maljamar reservoir	6.7249	6.7288	6.7256
Sour gas	15.4883	15.4934	15.4927
Sour gas4	6.9423	6.9426	6.9264

Table 3.4: Average number of iterations for the np phase-split calculations for the whole np phase region

The average number of iterations in table 3.4 is also slightly different for **lnK** and **lnK – chol.**. This difference is also due to approximation errors, since the condition number of the Hessian matrix **H** is larger than that of the Jacobian matrix **J**; the differences are significant in the vicinity of a phase boundary. The **lnK – Chol.** method is less robust than **lnK** (using an LU factorization). However, the use of Cholesky is clearly justified since the computational time for **lnK** is much bigger than for **lnK – Chol.** (fig. 3.20a to fig. 3.21b and fig. 3.23a to fig. 3.24b). **lnK – Chol.** still shows a behavior quadratic in time with respect to the number of components, but the polynomial constant of the quadratic term is smaller than for the **lnK** method. Our recommendation regarding conventional methods is to use **lnK – Chol.** (which is faster than **lnK**), and switch to **lnK** whenever a value of θ is very small (a phase boundary is approached), or two phase mole fractions are almost equal (this can indicate the proximity of a critical point), using an ϵ of, say, $1e^{-3}$. Even though the **NLVM** method appears to be the fastest conventional method, it is less robust than **lnK**, requiring systematically more iterations near phase boundaries (its condition number also deteriorates rapidly as a phase boundary is approached); on the other hand, **NLVM** is more robust than **nL/nV** [Michelsen [1982a]]. In fig. 3.27a, with an early switch, globally **nL/nV** requires three more iterations than **lnK** for convergence. For a late switch such as in fig. 3.28a or in fig. 3.27a for CO_2 concentrations greater than 57%, the differences between the methods are small even for the four phase case.

Reduction methods reveal a linear behavior of the time dependence on the number of components in the mixture. They are more costly than conventional methods (**lnK – Chol.** and **NLVM**) for a small number of components, but become time earning with an increasing nc . The crossing of the curves corresponding to the conventional and reduction methods depends on the number of BIP families. For 1 BIP family (Maljamar separator mixture) the computational time per iteration of **h – red.** becomes equivalent to that of **lnK – Chol.** at 11 components (fig. 3.20a) and at 8 components when looking at the global three-phase region (fig. 3.23a). For 2 BIP families (Maljamar reservoir mixture), it occurs at 14 components when looking at the Newton iteration itself (fig. 3.20b) and at 12 components for the entire three-phase region (fig. 3.23b). For 3 BIP families (sour gas mixtures), it happens at 17 components for both cases (fig. 3.21a and fig. 3.24a).

By comparing both proposed reduction methods, the use of the symmetry allows a small decrease in the computational time. Both versions reveal a linear trend (fig. 3.20a to fig. 3.21b and fig. 3.23a to fig. 3.24b), yet the curves are slightly steeper with the direct method. The differences in computational time are not considerable, but are clearly in favor of the **h – red. – s** method.

In table 3.5, an extension of the figures fig. 3.20a to fig. 3.21b and fig. 3.23a to fig. 3.24b is given, giving the results for 200 components. For this number, the reduction method is globally 20 times faster than the

conventional method **lnK** – **Chol.** for three phases. For four-phases, it becomes 35 times faster. As expected, for a system with more phases, conventional and reduction methods become equivalent for a smaller number of components. For the sour gas mixture in fig. 3.24a with three phases, the crossing occurs at 17 components and in fig. 3.24b for four phases at 15 components.

Mixture	h-red.	ln K-Chol.
Maljamar separator	3.6	95.27
Maljamar reservoir	4.69	101.85
Sour gas	5.87	104.6
Sour gas4	7.09	245.03

Table 3.5: Computational time per 10000 iterations for conventional and reduction Newton iterations at 200 components

The reduction method is faster than **lnK** (if symmetry is not used) even for a small number of components. This shows the importance of the linear solver; when a comparison of conventional and reduction methods is reported, it is thus mandatory to report also how the symmetry properties were used in the conventional case.

In all examples, only the np phase-split calculation results are exposed. Results for stability analysis and two phase flash calculations (comparison between conventional and reduction methods) were given in [Petitfrere and Nichita [2014b]]. The global multiphase calculation procedure follows [Michelsen [1982a]], that is, one or two stability analysis are performed for the feed composition for the vapor and/or liquid trial phases. Then alternatively with an $np - 1$ phase-split calculations, at least 7 initial guesses (from [Li and Firoozabadi [2012]]) are tried to test the stability of one equilibrium phase. This means that the global time for a three-phase calculation is given by 7/8 stability analysis, one two-phase flash and one three-phase flash. It has been shown [Petitfrere and Nichita [2014b]] that **h** – **red.** performed the stability analysis faster than the conventional methods or up to 2 nonzero BIP families. For the sour gas mixture (3 nonzero BIP families), the computational time for the stability analysis is given fig. 3.22. In this case, **h** – **red.** becomes equivalent to **lnK** – **Chol.** in terms of computational time for 12 components. For all mixtures, if we were to add the time to compute the stability analysis to the time to compute the two and three phase flashes, the crossing between the curves corresponding to conventional and reduction methods will appear before those exposed in fig. 3.20a to fig. 3.21b and fig. 3.23a to fig. 3.24b. Moreover, the time spent by SSI iterations before the switch was not taken into account. (an SSI iteration is faster in reduction methods, since the evaluation of a quadratic form is avoided). If we took into account all these times, this will tend to bring the crossing of conventional/reduction curves at smaller values of nc than in fig. 3.20a to fig. 3.21b and fig. 3.23a to fig. 3.24b (for example, for the sour gas the crossing is expected to occur somewhere in between 12 and 17).

3.2.4.5 Conclusions

In this work, the reduction methodology of [Nichita and Graciaa [2011]] is extended to multiphase flash calculations with any number of phases. The independent variables are unbounded and derive directly from a constrained minimization of the Gibbs free energy. Moreover, theoretically it has been proven that the convergence path was equivalent than for **lnK**; this was confirmed numerically, except from some difficult conditions (when the system is not too ill-conditioned).

Two versions of the new method are developed. The first one is a direct extension of the two phase reduction method from [Nichita and Graciaa [2011]]. The second one makes use of the symmetry and allows building the Jacobian matrix computing about only half of the elements. The use of the symmetry slightly decreases the computational time, as compared with the direct method. The reduction methods allow a linear growth of computational time with respect to the number of components, while a quadratic trend is observed for conventional formulations. Both versions of the reduction method proved to be more efficient

than conventional methods after a certain number of components.

For reservoir simulation purposes, the new methodology could be competitive for a small number of BIP families (< 3). For a large number of components in the mixture (as encountered in chemical processing), the differences between computational times required by conventional and reduction methods are really important; reduction variables can have a huge impact on computational time for such problems. Finally, it is worth mentioning that a big advantage of the reduction methods is that a detailed composition can be used; no lumping is required (thus no approximations). Lumping may affect to a high extent the location and size of tiny three-phase regions [Nichita et al. [2006b]].

3.3 Phase equilibrium calculations with quasi-Newton methods

3.3.1 Introduction

The second order Newton methods is the most commonly used in phase equilibrium calculations; however, their convergence radius can be small, and a number of first order successive substitutions iterations must be performed before switching to Newton iterations. In this section, we focus on other kinds of minimization methods: the quasi-Newton ones, which represent interesting alternatives. Their convergence radius is larger, there is no need to solve a linear system, and the Hessian matrix is approximated with a very low computational cost. However, quasi-Newton methods are extremely sensitive to scaling; It will be shown that that physical variables (mole number) are not suited for the quasi-Newton method.

The BFGS method [Broyden [1970], Fletcher [1970], Goldfarb [1970], Shanno [1970]] exhibits a supra-linear convergence rate. A rank two update of the Hessian matrix at each iteration guarantees its positive definiteness. The BFGS algorithm has already been applied to both for stability testing and phase-split problems [Ammar and Renon [1987], Garcia-Sanchez et al. [1996], Garcia-Sanchez et al. [2001], Hoteit and Firoozabadi [2006]].

For the stability analysis problem, [Michelsen [1982b]] proposed a set of independent variables which allows a good scaling of the problem and a better conditioned system; using this set, the BFGS method proved to be quite competitive [Hoteit and Firoozabadi [2006]]. In this work, we develop a methodology for finding the appropriate change of variables to obtain the best scaling for phase equilibrium problems; applied to the flash calculation problem, the resulting change of variables lead to a Hessian matrix with a perfect scaling, i.e., $\mathbf{H} = \mathbf{I} + \mathbf{D} + \mathbf{ND}$, where \mathbf{I} is the identity matrix, \mathbf{D} is a diagonal matrix with elements vanishing at the solution, and \mathbf{ND} is an effective low-rank matrix with non-diagonal terms.

3.3.2 The BFGS quasi-Newton method

3.3.2.1 Newton

Let $f(\mathbf{x})$ be the function to minimize and $\mathbf{x} = [x_1, \dots, x_N]$. Let \mathbf{x}_k stands for the vector \mathbf{x} at the k^{th} iteration. The minimum is obtained for

$$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}) = \mathbf{0} \quad (3.140)$$

The Newton method enables a quadratic convergence by means of the update:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{H}_k^{-1} \mathbf{g}_k \quad (3.141)$$

with \mathbf{g}_k is the gradient vector and \mathbf{H}_k is the Hessian matrix:

$$\mathbf{g}_k = \nabla f(\mathbf{x}_k) \quad \mathbf{H}_k = \nabla^2 f(\mathbf{x}_k) \quad (3.142)$$

3.3.2.2 The BFGS update

The BFGS algorithm approximates the Hessian matrix \mathbf{H} by means of a matrix \mathbf{B} , depending on the function f and the gradient \mathbf{g} . The approximation of the Hessian matrix \mathbf{B}_k is updated at each iteration by:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{p}_k} - \frac{\mathbf{B}_k \mathbf{p}_k \mathbf{p}_k^T \mathbf{B}_k}{\mathbf{p}_k^T \mathbf{B}_k \mathbf{p}_k} \quad (3.143)$$

with

$$\mathbf{y}_k = \nabla f(\mathbf{x}_k) - \nabla f(\mathbf{x}_{k-1}) \quad (3.144)$$

and

$$\mathbf{p}_k = \mathbf{x}_k - \mathbf{x}_{k-1} \quad (3.145)$$

As it is a rank-two update matrix, the Sherman-Morrison formula can be used twice to update directly the inverse of the matrix:

$$\mathbf{B}_{k+1}^{-1} = \mathbf{B}_k^{-1} + \frac{(\mathbf{p}_k^T \mathbf{y}_k + \mathbf{y}_k^T \mathbf{B}_k^{-1} \mathbf{y}_k) (\mathbf{p}_k \mathbf{p}_k^T)}{(\mathbf{y}_k^T \mathbf{p}_k)^2} - \frac{\mathbf{B}_k^{-1} \mathbf{y}_k \mathbf{p}_k^T + \mathbf{p}_k \mathbf{y}_k^T \mathbf{B}_k^{-1}}{\mathbf{p}_k^T \mathbf{y}_k} \quad (3.146)$$

As the Hessian matrix is approximated, the order of the method is decreased as compared to the Newton algorithm. The BFGS methods lead to a supra-linear convergence. The update of the variables is given by:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \mathbf{B}_k^{-1} \mathbf{g}_k \quad (3.147)$$

If $\mathbf{B}_k \approx \mathbf{H}_k$, the convergence is nearly quadratic. The first step is carried out choosing $\mathbf{B}_1 = \mathbf{I}$, which corresponds to a gradient descent step. This approximation only requires the evaluation of the gradient (first derivatives of the Gibbs energy, which corresponds here to calculate the fugacities). Besides, as the inverse of the Hessian is updated directly, no linear system needs to be solved. This can be really time saving when dealing with multiphase systems with a high number of components. An important feature is that the construction of BFGS formula guarantees the positive definiteness of the matrix, hence it guarantees a descent direction (unlike the Newton method, in which the Hessian matrix is not guaranteed to be positive definite). In practice, a line search procedure is used with quasi-Newton methods:

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \lambda \mathbf{B}_k^{-1} \mathbf{g}_k \quad (3.148)$$

with λ a real computed based on a line search procedure.

3.3.2.3 The Line search procedure

We introduce the descent direction vector:

$$\mathbf{s}_k = -\mathbf{B}_k^{-1} \mathbf{g}_k \quad (3.149)$$

The line search algorithm solves λ which will sufficiently decrease the Gibbs energy f within the Wolfe norm. It is a one dimension minimization, the objective function being here considered as a function only of λ :

$$f(\lambda) = f(\mathbf{x}_k + \lambda \mathbf{s}_k) \quad (3.150)$$

3.3.2.3.1 The Wolfe conditions

The methods to solve for λ are generally iterative. Let λ_i be the i^{th} iterate on the variable λ , and $\lambda_0 = 0$. The vector solution for the line search procedure is

$$\mathbf{u}_l = \mathbf{x}_k + \lambda_l \mathbf{s}_k \quad (3.151)$$

and let us introduce the notations

$$f_l = f(\lambda_l) \quad \mathbf{g}_l(\mathbf{x}) = \nabla \mathbf{f}(\mathbf{x}) = \left[\frac{\partial f_l}{\partial x_1}, \dots, \frac{\partial f_l}{\partial x_{nc}} \right] \quad g'_l = \mathbf{g}_l^T \mathbf{s}_k \quad (3.152)$$

The iterate \mathbf{u}_l must satisfy the (weak) Wolfe criteria

$$\begin{cases} f_l \leq f_0 + c_1 \lambda_l g'_0 \\ g'_l \leq c_2 |g'_0| \end{cases}$$

In this work, $c_1 = 1d - 4$ and $c_2 = 0.9$.

Starting from $\lambda_1 = 1$ (equivalent to procedures without line search), if \mathbf{u}_1 does not satisfy the Wolfe conditions, λ_{l+1} is calculated. The iteration procedure to update λ is based on a polynomial interpolation of the function f [Nocedal and Wright [2006]].

3.3.2.3.2 Second order polynomial interpolation

If $\lambda_1 = 1$ does not satisfy the Wolfe conditions, in a first step, the function f is approximated with a second order polynomial function

$$f(\lambda) = a\lambda^2 + b\lambda + c \quad (3.153)$$

Using (eq. 3.153) for λ_l and $\lambda_0 = 0$, the difference $f_l - f_0$ is

$$f_l - f_0 = a\lambda_l^2 + b\lambda_l \quad (3.154)$$

Differentiating (eq. 3.153) with respect to λ gives:

$$\frac{\partial f(\lambda)}{\partial \lambda} = 2a\lambda + b \quad (3.155)$$

or,

$$\frac{\partial f(\lambda)}{\partial \lambda} = \mathbf{g}_k^T \mathbf{s}_k = g'_k \quad (3.156)$$

Combining (eq. 3.155) and (eq. 3.156), for $\lambda_0 = 0$ leads to:

$$g'_0 = b \quad (3.157)$$

We are looking for the minimum of f with respect to λ , which is obtained for $\frac{\partial f(\lambda)}{\partial \lambda} = 0$. Combining this equation with (eq. 3.155) for λ_{l+1} , one obtains

$$2a\lambda_{l+1} + b = 0 \quad (3.158)$$

Solving the system based on (eq. 3.154, eq. 3.157 and eq. 3.158) leads to

$$\lambda_{l+1} = \frac{-g'_0 \lambda_l^2}{2(f_l - f_0 - g'_0 \lambda_l)} \quad (3.159)$$

3.3.2.3.3 Third order polynomial interpolation

The function f is now approximated based on a cubic polynomial function:

$$f(\lambda) = a\lambda^3 + b\lambda^2 + c\lambda + d \quad (3.160)$$

Using (eq. 3.160) for λ_l and $\lambda_0=0$, the difference $f_l - f_0$ is

$$f_l - f_0 - g'_0\lambda_l = a\lambda_l^3 + b\lambda_l^2 \quad (3.161)$$

and using (eq. 3.160) for λ_{l-1} and $\lambda_0 = 0$, performing the same operation, one obtains

$$f_{l-1} - f_0 - g'_0\lambda_{l-1} = a\lambda_{l-1}^3 + b\lambda_{l-1}^2 \quad (3.162)$$

We look for λ_{l+1} which satisfies

$$\frac{\partial f(\lambda)}{\partial \lambda} = 3a\lambda^2 + 2b\lambda + c = 0 \quad (3.163)$$

Combining (eq. 3.163) with (eq. 3.156) for $\lambda_0 = 0$, we get

$$g'_0 = c \quad (3.164)$$

Besides, we are looking for the minimum of f with respect to λ , which is obtained for $\frac{\partial f(\lambda)}{\partial \lambda} = 0$. Combining this equation with (eq. 3.163) for λ_{l+1} , one obtains

$$3a\lambda_{l+1}^2 + 2b\lambda_{l+1} + c = 0 \quad (3.165)$$

Combining (eq. 3.161 and eq. 3.162) leads to the following linear system:

$$\begin{pmatrix} \lambda_l^3 & \lambda_l^2 \\ \lambda_{l-1}^3 & \lambda_{l-1}^2 \end{pmatrix} \begin{pmatrix} a \\ b \end{pmatrix} = \begin{pmatrix} f_l - f_0 - g'_0 \\ f_{l-1} - f_0 - g'_0 \end{pmatrix} \quad (3.166)$$

The system is solved to obtain a and b and replacing a , b and c in (eq. 3.165), a second order polynomial equation is solved to obtain λ_{l+1} . At each step, it is important to check if λ_{l+1} remains in a correct range. That is to say $\lambda_{l+1} < 0.5\lambda_l$ and $\lambda_{l+1} > 0.1\lambda_l$.

In order to develop a reliable algorithm, in this work, a switch back procedure to SSI is performed (see the BFGS algorithm). As soon as the BFGS update does not decrease the Gibbs free energy, an SSI step is used instead to guarantee the convergence to the solution. In some situations, it could be more advantageous to accept steps leading to an increase of the Gibbs energy. However, those methodologies are not totally reliable for difficult conditions (near the phase boundaries for instance), and could lead to convergence problems.

3.3.2.4 Ammar and Renon BFGS implementation

[Ammar and Renon [1987]] proposed a BFGS implementation based on the logarithm of the equilibrium constants $\ln \mathbf{K}$. From (eq. 3.35) and (eq. 3.33), the Newton iteration equation can be written as

$$\mathbf{J} \Delta \ln \mathbf{K} = \mathbf{H} \mathbf{U}^{-1} \Delta \ln \mathbf{K} = -\mathbf{g} \quad (3.167)$$

which is equivalent to:

$$\mathbf{U}^{-1} \Delta \ln \mathbf{K} = -\mathbf{H}^{-1} \mathbf{g} \quad (3.168)$$

In the BFGS method, since the Hessian matrix \mathbf{H} is approximated by the matrix \mathbf{B} , the above equation become

Algorithm 3.2 Line search algorithm

```

Given  $\mathbf{x}_k, f_0, \mathbf{s}_k$ 
Compute  $g'_0$  (eq. 3.156)
 $end = 0$ 
 $\lambda_1 = 1$  and  $l = 1$ 
while  $end = 0$  do
   $\mathbf{u}_1 = \mathbf{x}_k + \lambda_l \mathbf{s}_k$ 
  Compute  $f_l$  and  $g'_l$ 
  if Wolfe conditions are satisfied then
     $end = 1$ 
  else
    if  $l = 1$  then
      Use second order interpolation to compute  $\lambda_{l+1}$ 
    else
      Use third order interpolation to compute  $\lambda_{l+1}$ 
    end if
  end if
  if  $\lambda_{l+1} > 0.5\lambda_l$  then
     $\lambda_{l+1} = 0.5\lambda_l$ 
  end if
  if  $\lambda_{l+1} < 0.1\lambda_l$  then
     $\lambda_{l+1} = 0.1\lambda_l$ 
  end if
   $l = l + 1$ 
end while

```

Algorithm 3.3 BFGS algorithm

```

Given  $\mathbf{x}_0, B_0 = I$ 
 $k = 0$ 
Compute  $f_0, \mathbf{g}_0$ 
while  $\|\mathbf{g}_k\| \leq \epsilon$  do
  if  $k = 0$  then
     $\mathbf{s}_k = -\mathbf{g}_k$ 
  else
     $\mathbf{p}_k = \mathbf{x}_k - \mathbf{x}_{k-1}$ 
     $\mathbf{y}_k = \mathbf{g}_k - \mathbf{g}_{k-1}$ 
    Update  $\mathbf{B}_k^{-1}$  using (eq. 3.146)
     $\mathbf{s}_k = -\mathbf{B}_k^{-1} \mathbf{g}_k$ 
  end if
  Solve  $\lambda, \mathbf{x}_{k+1} = \mathbf{x}_k + \lambda \mathbf{s}_k$  (algorithm 3.2)
  Compute  $f_{k+1}$ 
  if  $f_{k+1} > f_k$  then
    Update  $x_{k+1}$  based on SS iteration
    Compute  $f_{k+1}$ 
  end if
  Compute  $\mathbf{g}_{k+1}$ 
   $k = k + 1$ 
end while

```

$$\mathbf{U}^{-1} \Delta \ln \mathbf{K} = -\mathbf{B}^{-1} \mathbf{g} \quad (3.169)$$

The vector $\ln \mathbf{K}$ is updated at each iteration with two sequential updates. First

$$\mathbf{y} = -\mathbf{B}^{-1} \mathbf{g} \quad (3.170)$$

and then a line search procedure is used to update $\ln \mathbf{K}$:

$$\ln \mathbf{K}_{k+1} = \ln \mathbf{K}_k + \lambda \mathbf{U} \mathbf{y} \quad (3.171)$$

with the matrix \mathbf{U} calculated from (eq. 3.30).

After each update, the Rachford-Rice equation is solved for the vapor mole fraction V , then composition and mole numbers for each phase are calculated.

3.3.3 Proposed method

3.3.3.1 Two-phase split calculations

3.3.3.1.1 Alpha variables

Given a change of variables $\alpha_i = \alpha_i(n_L)$, by using the chain rule, the gradient vector is

$$g_i = \frac{\partial G}{\partial \alpha_i} = \frac{\partial G}{\partial n_{iL}} \frac{\partial n_{iL}}{\partial \alpha_i} \quad (3.172)$$

and the Hessian matrix is

$$H_{ij} = \frac{\partial^2 G}{\partial \alpha_i \partial \alpha_j} = \frac{\partial^2 G}{\partial n_{iL} \partial n_{jL}} \frac{\partial n_{iL}}{\partial \alpha_i} \frac{\partial n_{jL}}{\partial \alpha_j} + \frac{\partial G}{\partial n_{iL}} \frac{\partial^2 n_{iL}}{\partial \alpha_i \partial \alpha_j} \quad (3.173)$$

In order to obtain a well-scaled Hessian, that is, to have the identity matrix \mathbf{I} as the first term in the Hessian matrix expression in (eq. 3.27), the term δ_{ij}/LVu_i (with u_i given in (eq. 3.31)) from $\frac{\partial^2 G}{\partial n_{iL} \partial n_{jL}}$ must be multiplied by: $\frac{\partial n_{iL}}{\partial \alpha_i} \frac{\partial n_{jL}}{\partial \alpha_j}$, to obtain:

$$\frac{\delta_{ij}}{LVu_i} \frac{\partial n_{iL}}{\partial \alpha_i} \frac{\partial n_{jL}}{\partial \alpha_j} = \delta_{ij} \quad (3.174)$$

with

$$LV\sqrt{u_i}\sqrt{u_j} = \sqrt{\frac{n_{iL}n_{iV}}{z_i}} \sqrt{\frac{n_{jL}n_{jV}}{z_j}} \quad (3.175)$$

By identification:

$$\frac{\partial n_{iL}}{\partial \alpha_i} = \sqrt{\frac{n_{iL}n_{iV}}{z_i}} \quad (3.176)$$

and from (eq. 3.176)

$$\alpha_i = \int \frac{\sqrt{z_i}}{\sqrt{n_{iL}(z_i - n_{iL})}} dn_{iL} \quad (3.177)$$

Integration gives

$$\alpha_i = 2\sqrt{z_i} \arcsin \left(\sqrt{\frac{n_{iL}}{z_i}} \right) + C_i \quad (3.178)$$

Taking $C_i = 0$,

$$\alpha_i = 2\sqrt{z_i} \arcsin \left(\sqrt{\frac{n_{iL}}{z_i}} \right) \quad (3.179)$$

Since $0 \leq n_{iL} \leq z_i$

$$0 \leq \sqrt{\frac{n_{iL}}{z_i}} \leq 1 \quad (3.180)$$

The liquid mole fractions can be expressed as a function of α_i :

$$\sqrt{n_{iL}} = \sqrt{z_i} \sin \left(\frac{\alpha_i}{2\sqrt{z_i}} \right) \quad (3.181)$$

To ensure a bijective restriction one needs to bound α_i :

$$\alpha_i \in [0, \pi\sqrt{z_i}] \quad (3.182)$$

Deriving the Gibbs free energy with respect to this variable leads to the gradient vector:

$$g_i = \sqrt{\frac{n_{iL}n_{iV}}{z_i}} [(\ln n_{iL} - \ln L + \ln \phi_{iL}) - (\ln n_{iV} - \ln V + \ln \phi_{iV})] \quad (3.183)$$

And the elements of the Hessian matrix are

$$H_{ij} = \delta_{ij} - \sqrt{u_i u_j} (1 + LV \Phi_{ij}) + 1/2 \delta_{ij} g_i \left(\frac{n_{iV} - n_{iL}}{z_i} \right) \quad (3.184)$$

3.3.3.1.2 Beta variables

A similar procedure can be used by deriving with respect of n_{iV} , to obtain:

$$\beta_i = 2\sqrt{z_i} \arcsin \left(\sqrt{\frac{n_{iV}}{z_i}} \right) \quad (3.185)$$

Which leads to the vapor mole numbers:

$$\sqrt{n_{iV}} = \sqrt{z_i} \sin \left(\frac{\beta_i}{2\sqrt{z_i}} \right) \quad (3.186)$$

Let us now establish the relation between α_i and β_i . Using (eq. 3.181),

$$n_{iL} = z_i \sin^2 \left(\frac{\alpha_i}{2\sqrt{z_i}} \right) \quad (3.187)$$

The mass balance equation $z_i = n_{iL} + n_{iV}$ enables to compute n_{iV}

$$n_{iV} = z_i \cos^2 \left(\frac{\alpha_i}{2\sqrt{z_i}} \right) \quad (3.188)$$

Similarly, from (eq. 3.186)

$$n_{iV} = z_i \sin^2 \left(\frac{\beta_i}{2\sqrt{z_i}} \right) \quad (3.189)$$

$$n_{iL} = z_i \cos^2 \left(\frac{\beta_i}{2\sqrt{z_i}} \right) \quad (3.190)$$

From (eq. 3.187) and (eq. 3.190), the following equation is obtained:

$$\cos^2 \left(\frac{\beta_i}{2\sqrt{z_i}} \right) = \sin^2 \left(\frac{\alpha_i}{2\sqrt{z_i}} \right) \quad (3.191)$$

Since, $\alpha_i \in [0, \pi\sqrt{z_i}]$ and $\beta_i \in [0, \pi\sqrt{z_i}]$, $\cos \left(\frac{\beta_i}{2\sqrt{z_i}} \right) \geq 0$ and $\sin \left(\frac{\alpha_i}{2\sqrt{z_i}} \right) \geq 0$

$$\cos \left(\frac{\beta_i}{2\sqrt{z_i}} \right) = \sin \left(\frac{\alpha_i}{2\sqrt{z_i}} \right) \quad (3.192)$$

Which leads to

$$\alpha_i = 2\sqrt{z_i} \arcsin \left(\cos \left(\frac{\beta_i}{2\sqrt{z_i}} \right) \right) = 2\sqrt{z_i} \left(\frac{\pi}{2} - \frac{\beta_i}{2\sqrt{z_i}} \right) \quad (3.193)$$

Hence,

$$\alpha_i = \pi\sqrt{z_i} - \beta_i \quad (3.194)$$

The gradient and the Hessian matrix using β_i as independent variables are directly related to the gradient and the Hessian matrix using α_i as independent variables since

$$\frac{\partial \beta_i}{\partial \alpha_j} = -\delta_{ij} \quad (3.195)$$

and

$$\frac{\partial^2 \beta_i}{\partial \alpha_j \partial \alpha_k} = \delta_{ij} \delta_{ik} \quad (3.196)$$

Hence, the Hessian is the same using both sets of independent variables, and the gradients are related by:

$$g_i^\alpha = -g_i^\beta \quad (3.197)$$

3.3.3.2 Stability analysis

The methodology proposed above can be applied to the phase stability problem. Given a change of variables $\alpha_i = \alpha_i(\mathbf{Y})$, using the chain rule, the gradient vector is

$$g_i = \frac{\partial G}{\partial \alpha_i} = \frac{\partial G}{\partial Y_i} \frac{\partial Y_i}{\partial \alpha_i} \quad (3.198)$$

and the Hessian matrix is

$$H_{ij} = \frac{\partial^2 G}{\partial \alpha_i \partial \alpha_j} = \frac{\partial^2 G}{\partial Y_i \partial Y_j} \frac{\partial Y_i}{\partial \alpha_i} \frac{\partial Y_j}{\partial \alpha_j} + \frac{\partial G}{\partial Y_i} \frac{\partial^2 Y_i}{\partial \alpha_i \partial \alpha_j} \quad (3.199)$$

The term δ_{ij}/Y_i from $\frac{\partial^2 G}{\partial Y_i \partial Y_j}$ must be multiplied by: $\frac{\partial Y_i}{\partial \alpha_i} \frac{\partial Y_j}{\partial \alpha_j}$, to obtain:

$$\frac{\delta_{ij}}{Y_i} \frac{\partial Y_i}{\partial \alpha_i} \frac{\partial Y_j}{\partial \alpha_j} = \delta_{ij} \quad (3.200)$$

Which gives by identification:

$$\frac{\partial Y_i}{\partial \alpha_i} = \sqrt{Y_i} \quad (3.201)$$

From (eq. 3.200),

$$\alpha_i = \int \frac{1}{\sqrt{Y_i}} dY_i \quad (3.202)$$

Integration gives

$$\alpha_i = 2\sqrt{Y_i} + C_i \quad (3.203)$$

Taking $C_i = 0$, the procedure leads to the α_i variable introduced by [Michelsen [1982b]]. [Hoteit and Firoozabadi [2006]] developed an efficient BFGS procedure based on these variables to solve stability analysis problems.

3.3.4 Results

The BFGS algorithm (developed in a previous section) based on the proposed α_i and β_i variables has been implemented. The robustness and the efficiency of the method are tested on two mixtures, providing also comparisons with various methods from the literature [Ammar and Renon [1987], Liu and Nocedal [1989] and Nocedal [1980]]. In this study, the Peng-Robinson equation of state was used, but it is important to mention that the proposed method is independent of the EOS model and any equation of state could be used instead.

3.3.4.1 Tests on different BFGS methods

The BFGS algorithm has been implemented with different variables: α (called **alpha**), β (called **beta**) as well as the mole numbers \mathbf{n}^L or \mathbf{n}^V (denoted **nl/nv**). Both variables α and β are linked and lead to the same Hessian, and $g_i^\alpha = -g_i^\beta$. Independent variables n_{iV} and β_i are used if the mixture is predominantly liquid ($L > V$); otherwise, component mole numbers in the liquid phase, n_{iL} , and α_i are taken as independent variables [Nghiem et al. [1983]]. The procedure has been tested with and without the line search algorithm. BFGS methods with a line search procedure have been attributed the suffix L (i.e., **alpha-L**, **beta-L**). The procedure from [Ammar and Renon [1987]] is also implemented (it is denoted as **lnK-L**). We also compared the results obtained with an open source L-BFGS-B code from Nocedal [Nocedal [1980], Liu and Nocedal [1989]] against our own implementation of the BFGS method.

To simplify the notations, **alpha** and **beta** will be gathered into the same name (α).

3.3.4.2 Mixtures used in this study

Two mixtures were used in this study, the Y8 and the MY10 mixtures presented in subsection 3.2.2. For the Y8 mixture, calculations on two different isotherms have been performed in this section: at $T=285\text{K}$ ($T < T_c$ and close to the critical point) and $T=335\text{K}$ ($T > T_c$). For the MY10 mixture, calculations were performed along an isotherm at $T=565\text{K}$ located close to the critical point.

3.3.4.3 Error and stopping criteria

The Euclidean norm of the of difference of the logarithm of the fugacities is used to evaluate the error for each iteration:

$$S_{f2} = \sqrt{\sum_{i=1}^{nc} (\ln f_{iL} - \ln f_{iV})^2} \quad (3.204)$$

Convergence is assumed when $S_{f2} < 1e^{-8}$.

Moreover, the BFGS matrix is updated based on the differences between the iteration k and $k+1$. When the differences of the Gibbs free energy between two iterations become close to the machine error ($f_k \approx f_{k-1}$), convergence can be considered in this case. The same stopping criteria as [Nocedal [1980], Liu and Nocedal [1989]] is used in this work:

$$S_2 = \frac{f_{k-1} - f_k}{\max(|f_k|, |f_{k-1}|, 1)} < \epsilon_{\text{machine}} \quad (3.205)$$

Here, $\epsilon = 1d + 3$. The machine error is $\epsilon_{\text{machine}} = 1d - 16$.

3.3.4.4 Results

Two configurations were tested with different switching criteria. First, the BFGS has been used starting directly from the Wilson initialization [Wilson [1969]]. Then, SSI steps were first performed before switching to the BFGS method when $S_{f2} < 1e^{-2}$ (which corresponds to a late switch).

3.3.4.4.1 Starting directly without any SSI

In this section, the Wilson relation [Wilson [1969]] is used to initialize the equilibrium constants from which the initial phase distribution is inferred, then the BFGS algorithm is applied without any previous SSI.

For the Y8 mixture, along the isotherm $T=285\text{K}$ (respectively $T=335\text{K}$), the number of iterations required to reach convergence is plotted in fig. 3.29a (respectively in fig. 3.30a), with the corresponding

number of function evaluations given in fig. 3.29b (respectively in fig. 3.30b). For the MY10 mixture along the isotherm $T=565\text{K}$, the number of iterations needed to reach the convergence are given in fig. 3.31a with the corresponding number of function evaluations in fig. 3.31b. The results show a good regularity concerning the number of iterations obtained with $\alpha - L/\alpha$. This indicates that the method is not really sensitive to small perturbations, unlike for $\ln K - L$, in which various small oscillations appear (fig. 3.30a). Regarding the number of function evaluations, some oscillations can be seen for $\alpha - L$, especially close to the phase boundary (see fig. 3.30b). Besides, the use of the line search procedure ($\alpha - L$) slightly increases the number of function evaluations (particularly in fig. 3.29b). The use of the line search procedure is important near the phase boundaries. We noted that without any line search, α diverged for pressures close to the saturation pressure (under 0.3 bar). With a line search procedure ($\alpha - L$ and $\ln K - L$) convergence was obtained for all the conditions of the isotherms. These examples reveal the robustness of the proposed algorithm since no switch is necessary to reach convergence, and the capability of the algorithm to handle difficult conditions.

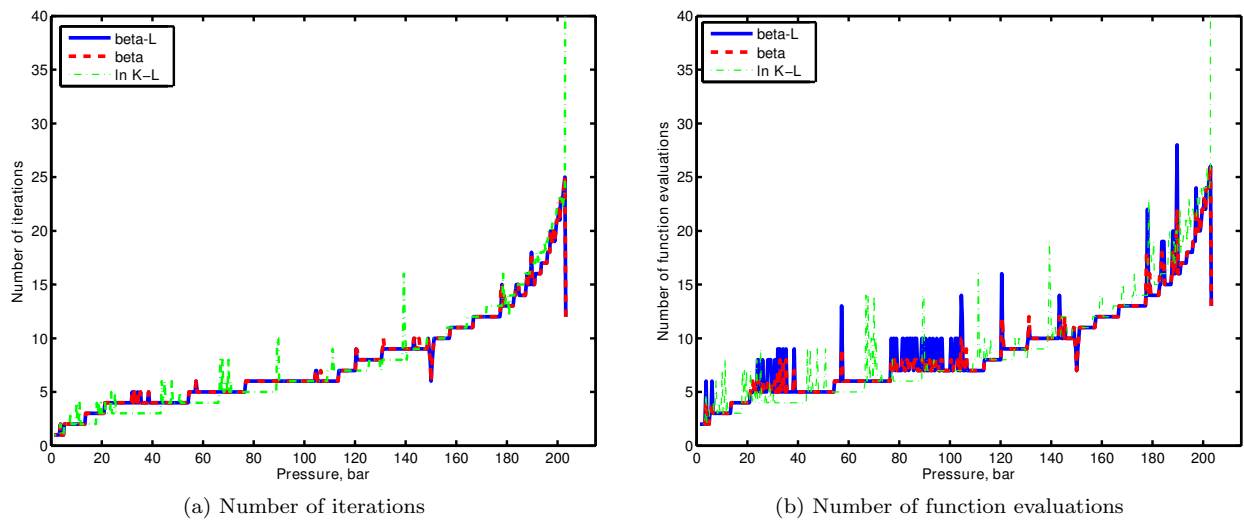


Figure 3.29: Iterations and function evaluations starting directly with the BFGS method; Y8 mixture at $T=285\text{ K}$

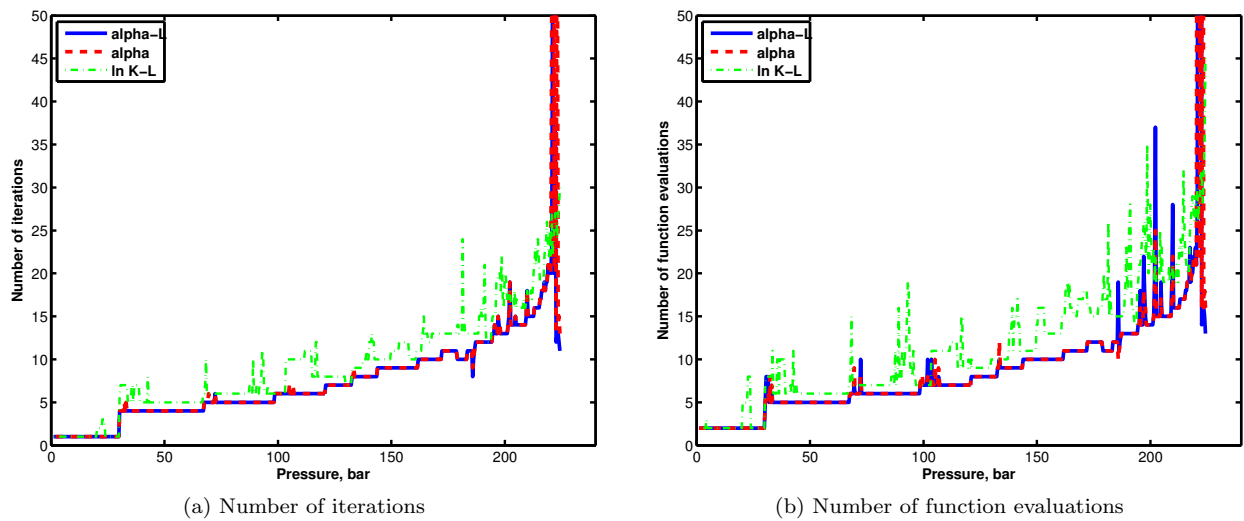


Figure 3.30: Iterations and function evaluations starting directly with the BFGS method; Y8 mixture at $T=335\text{ K}$

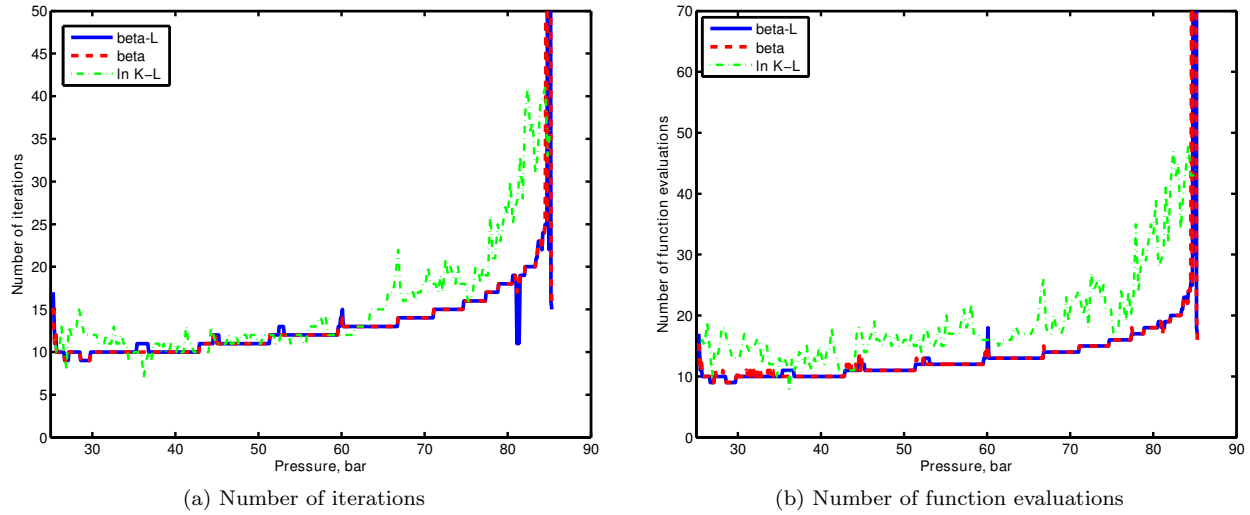


Figure 3.31: Iterations and function evaluations starting directly with the BFGS method; MY10 mixture at T=565 K

Except for some low pressures, along the isotherm T=285 K for the Y8 mixture, the number of iterations are generally smaller for **alpha** as compared with the $\ln K - L$ procedure. For the number of function evaluations, the results are almost always in favor of the proposed method whatever the conditions, including those close to the phase boundaries.

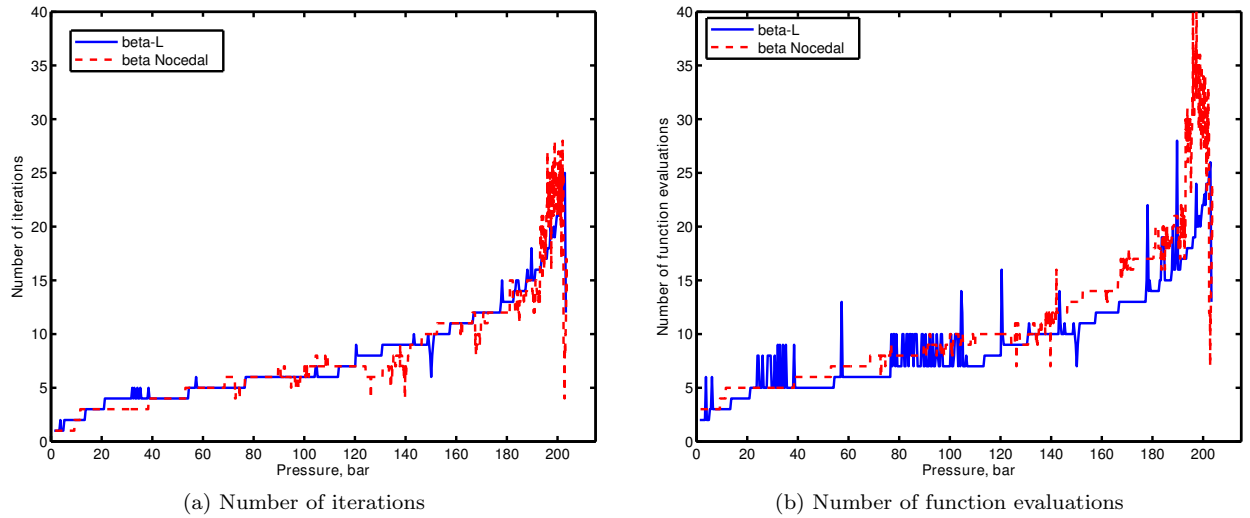


Figure 3.32: Comparison between the proposed BFGS algorithm and Nocedal L-BFGS-B for the Y8 mixture at T=285 K

Taking α or (β) as independent variables, a comparison of the L-BFGS-B code based on [Nocedal [1980], Liu and Nocedal [1989]] and the proposed implementation of the BFGS method is presented for the Y8 mixture. Fig. 3.32a gives the number of iterations obtained for the Y8 mixture along the isotherm T=285K, and fig. 3.33a for the isotherm T=335K. The corresponding number of function evaluations are given in fig. 3.32b along T=285K and in fig. 3.33b along T=335K. In terms of number of iterations, Nocedal's method and **alpha** - **L** gives close results for pressures located away from the phase boundaries. Close to the phase boundaries, oscillations can be noticed with Nocedal's algorithm unlike for the proposed algorithm. Besides, **alpha** - **L** requires less iterations to reach convergence and Nocedal's algorithm generally requires at least two more function evaluations to converge than the code based on the proposed method. Close to the phase boundaries, Nocedal's algorithm starts oscillating and the differences with the provided algorithm

increase. It is important to mention that the L-BFGS-B program has been designed to be general, but suited to large optimization problems. Our approach has been designed specifically for equilibrium calculation problems (adding a switch back procedure to SSI), which explains the differences, especially close to the phase boundaries.

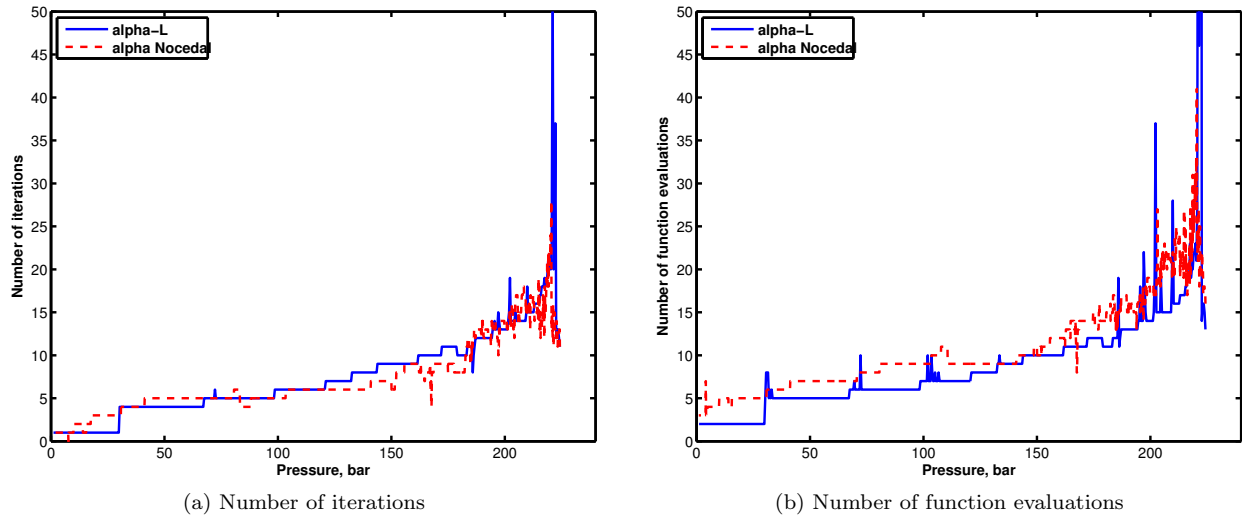


Figure 3.33: Comparison between the proposed BFGS algorithm and Nocedal L-BFGS-B for the Y8 mixture at T=335 K

3.3.4.4.2 Starting after SSI iterations

Using an initial guess from the Wilson relation, SSI steps are performed as long as $S_{f2} < 1e^{-2}$, then a switch to the BFGS method is carried out. For the Y8 mixture, the number of iterations required for convergence along the isotherm T=285 K (respectively T=335K) is given in fig. 3.34a (respectively in fig. 3.35a). The corresponding number of function evaluations is given in fig. 3.34b for T=285K, and in fig. 3.35b for the isotherm T=335K. For the MY10 mixture along the isotherm T=565K, the number of iterations is given in fig. 3.36a as well as the corresponding number of function evaluations in fig. 3.36b. In these figures, the number of SSI iterations before the switch is also given.

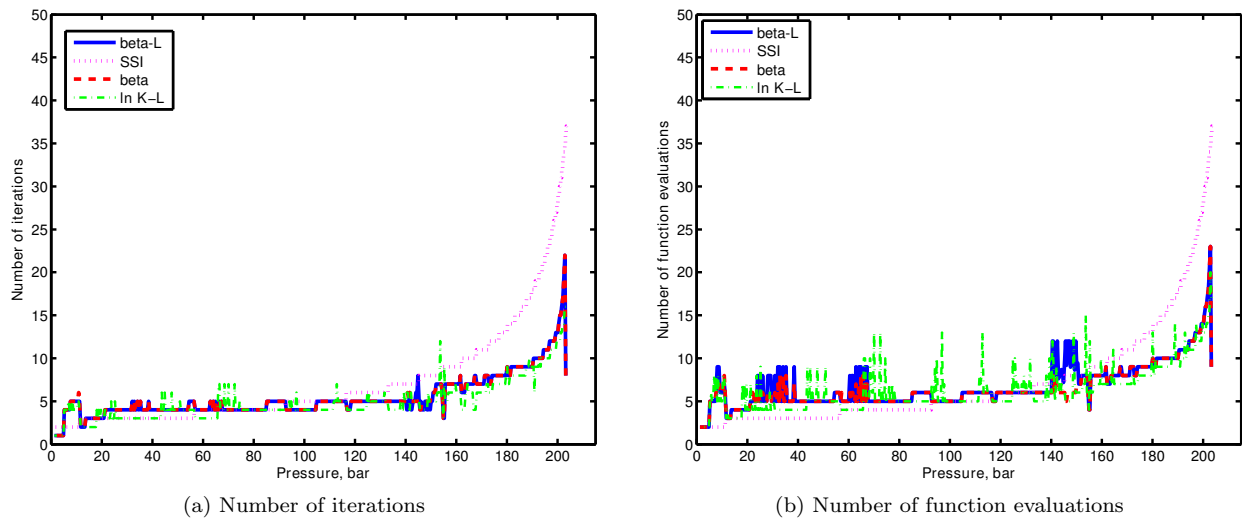


Figure 3.34: Iterations and function evaluations after a switch from SSI, Y8 mixture at T=285 K

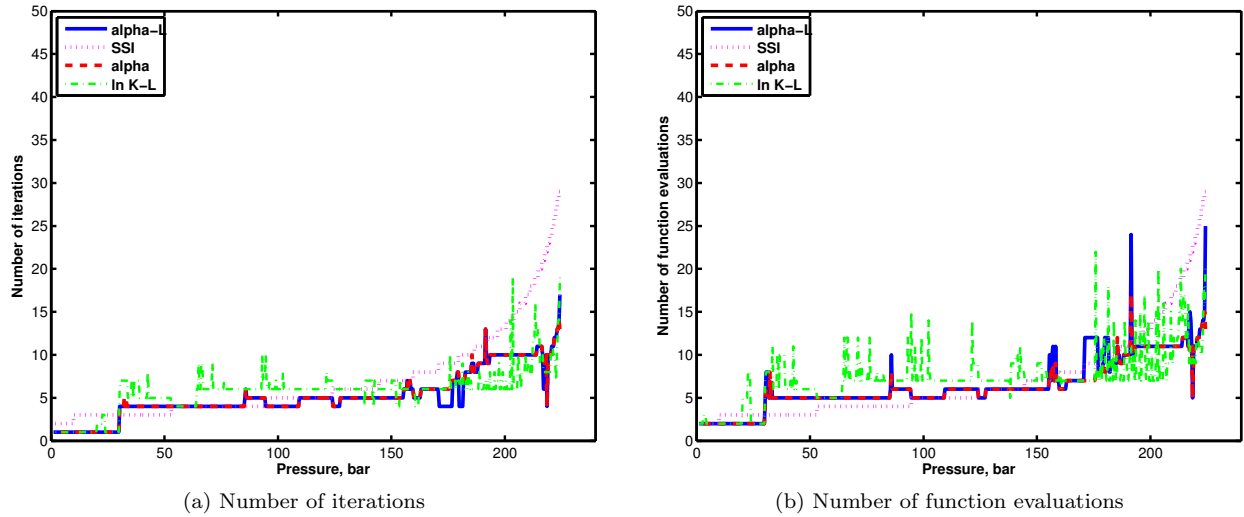


Figure 3.35: Iterations and function evaluations after a switch from SSI, Y8 mixture at T=335 K

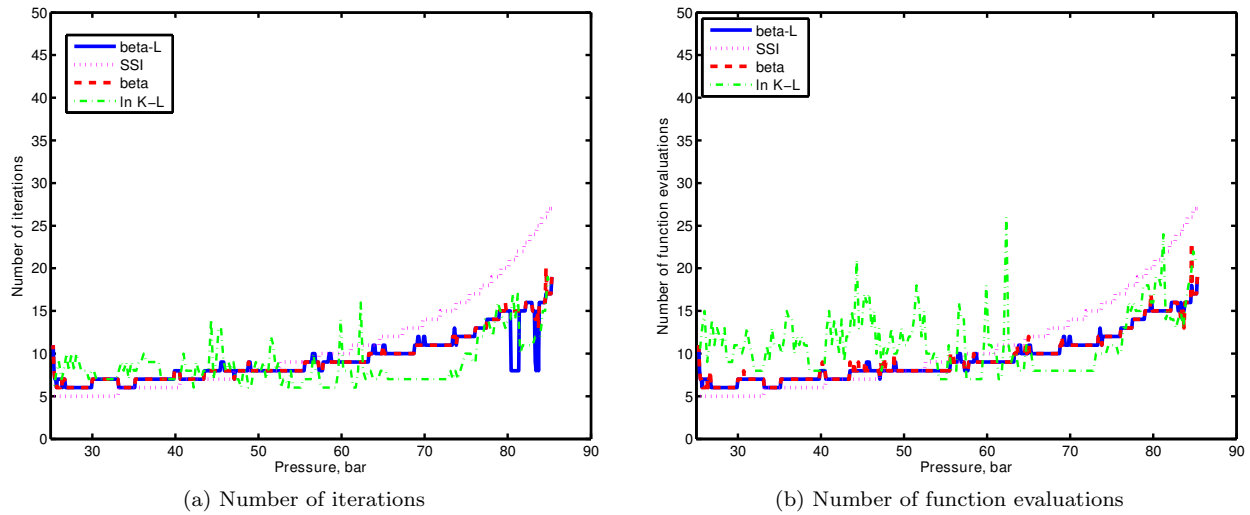


Figure 3.36: Iterations and function evaluations after a switch from SSI, MY10 mixture at T=565 K

Smaller oscillations can be seen with a late switch as compared with a direct start with the BFGS method (see fig. 3.30b against fig. 3.35b for the Y8 mixture). A late switch enables to initialize the BFGS method with a composition located closer to the solution (where the Hessian matrix is better scaled and the system is better conditioned than in early iteration stages), and the number of BFGS iterations after the switch is smaller than starting directly with the BFGS from the Wilson relation, (see for instance fig. 3.29a and fig. 3.29b vs fig. 3.34a and fig. 3.34b). This time, for pressures away from the saturation pressures, the proposed method requires less iterations and function evaluations to reach convergence as compared to $\ln K - L$. However, for conditions close to the phase boundaries, $\ln K - L$ becomes more efficient (see fig. 3.34a to fig. 3.36b). Globally, the same number of iterations is obtained for α as compared with $\alpha - L$. However, this time, the convergence is obtained for all the tested conditions even without any line search procedure (α). Since SSI are performed before the switch, the BFGS is started with a better initialization closer to the solution which makes the convergence easier.

Finally, a comparison is presented between the proposed BFGS method with $\alpha/\alpha - L$ variables and the same algorithm using the mole numbers as independent variables. The mole numbers lead to bad scaling and thus to an ill-conditioned Hessian matrix. This is shown for the Y8 mixture at T=335K in

fig. 3.37, in which \mathbf{nv}/\mathbf{nL} requires significantly more function evaluations to reach convergence. Besides, even with a line search procedure, divergence occurs for many conditions as the phase boundary is approached.

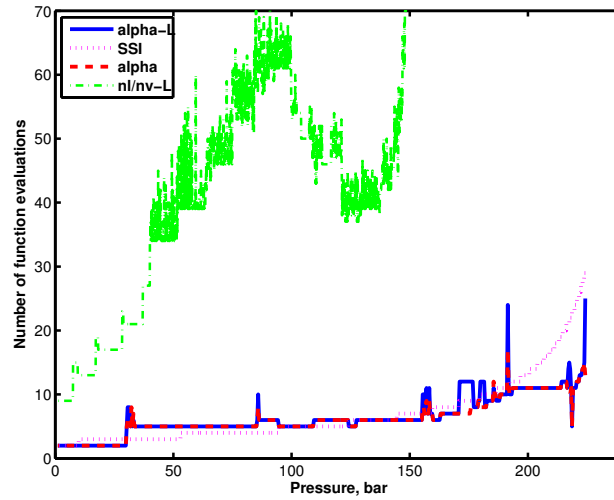


Figure 3.37: $\mathbf{nl}/\mathbf{nv-L}$ compared with $\mathbf{alpha}/\mathbf{alpha-L}$; number of function evaluations for the Y8 mixture at $T=335$ K (after a switch from SSI)

3.3.4.4.3 Convergence analysis

In this section, a comparison of the convergence behavior between various BFGS algorithms ($\mathbf{lnK} - \mathbf{L}$ and \mathbf{alpha}) and the Newton method using \mathbf{lnK} as independent variables is presented. The BFGS methods are started directly from the Wilson initialization. For the Newton algorithm, a switch from SSI is performed when $S_{f2} < 1e^{-1}$. The tests were performed using the Y8 mixture at $T=335$ K, and $P=50$ bar (fig. 3.38a), $P=200$ bar (fig. 3.38b), $P=220$ bar (fig. 3.39a) and $P=224.2$ bar (fig. 3.39b) (the last condition is located very close to the saturation pressure).

The Newton method (as a second order method) requires fewer iterations to converge to the solution as compared to the BFGS based algorithms (in which supra-linear steps are performed). Supra-linear behavior can be seen for instance in fig. 3.39a between the 4th and 9th iterations. In the last iterations for the conditions $T=220$ K and $T=224.2$ K (fig. 3.39a and fig. 3.39b), one can notice that the BFGS method enables to reach nearly-quadratic steps.

For conditions far from the phase boundary (see fig. 3.38a), \mathbf{alpha} requires only two more iterations than the Newton algorithm to reach convergence. Getting closer to the phase boundary, the differences with the Newton method become larger (see fig. 3.38b, fig. 3.39a). However, very close to the phase boundary, three more iterations are necessary for \mathbf{alpha} than for the Newton. This demonstrates the efficiency of the proposed method even for difficult conditions. $\mathbf{alpha} - \mathbf{L}$ always require less iterations than $\mathbf{lnK} - \mathbf{L}$ for the four conditions considered. These differences are increasing when getting closer to the phase envelope (see fig. 3.39b where $\mathbf{alpha} - \mathbf{L}$ only requires 16 iterations to reach convergence whereas $\mathbf{lnK} - \mathbf{L}$ converges in 36 iterations). This confirms the results obtained for the isotherms starting directly with the BFGS method (fig. 3.29a, fig. 3.30a, fig. 3.31a).

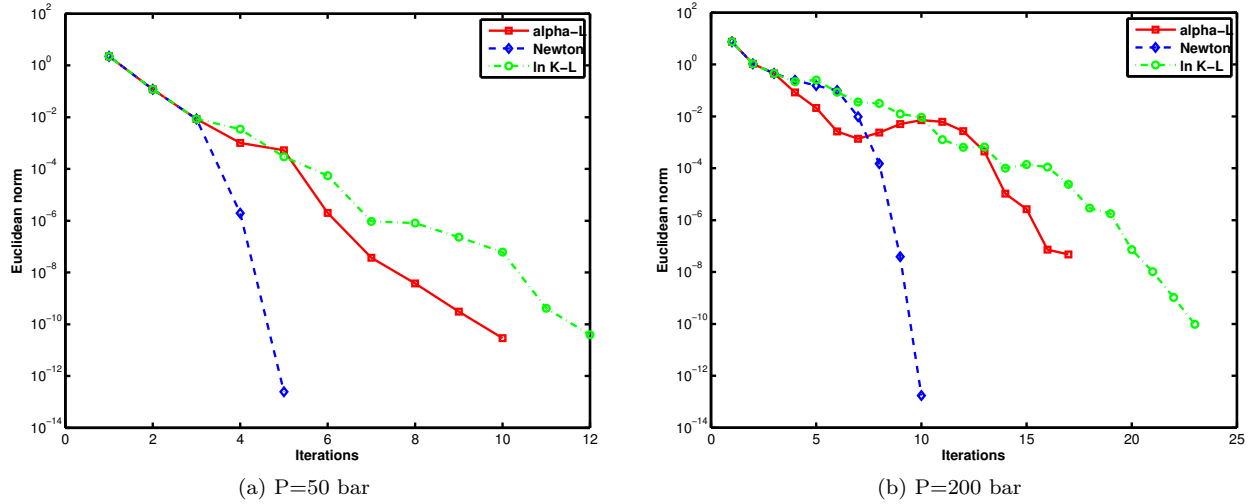


Figure 3.38: Error versus number of iterations for the Y8 mixture at T=335K

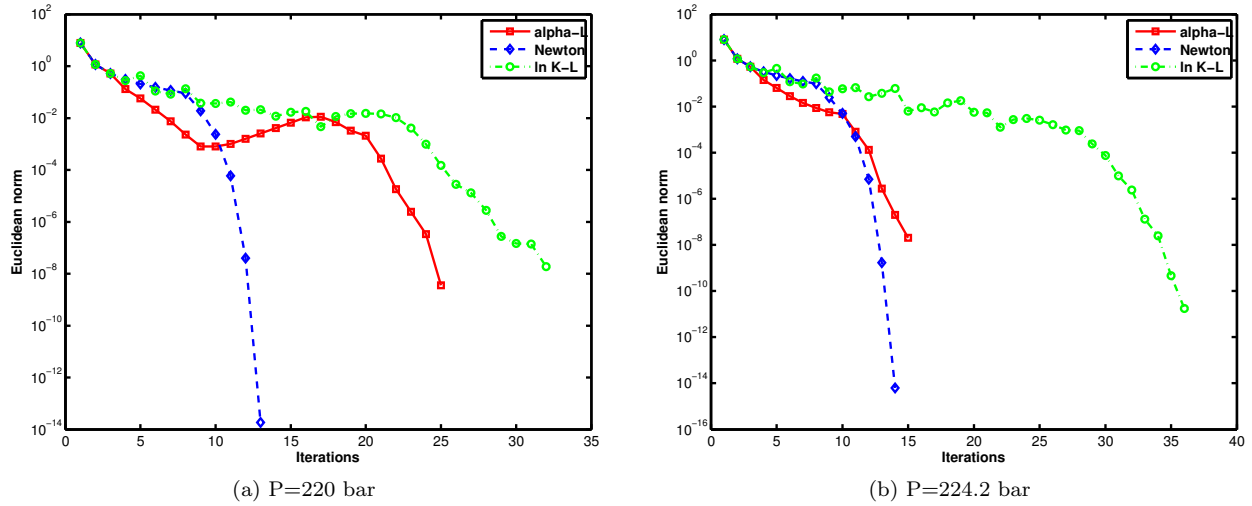


Figure 3.39: Error versus number of iterations for the Y8 mixture at T=335K

3.3.4.5 Discussion

With the use of a line search method, convergence was obtained with the proposed BFGS method for all the presented conditions; without line search, divergence occurs only near the phase boundaries when starting directly from Wilson initialization. Globally, the proposed method proved to converge within fewer iterations and function evaluations than the $\ln \mathbf{K} - \mathbf{L}$ procedure from [Ammar and Renon [1987]].

Oscillations have been noticed with $\ln \mathbf{K} - \mathbf{L}$. Most of the line search methods interpolate a function based on polynomial approximations to compute a step which satisfies the Wolfe conditions. In Ammar and Renon's method, the gradient of the Gibbs free energy is calculated with the mole numbers as independent variables. Therefore the line search procedure is the same as for a BFGS method taking the mole numbers as independent variables. It has been shown that the BFGS methods based on the mole numbers did not perform well; $\ln \mathbf{K} - \mathbf{L}$ inherits this behavior. With the proposed new set of independent variables, the gradient and the Hessian derive directly from the Gibbs energy. The line search procedure is more efficient and convergence to the solution was achieved for all the tested conditions.

For most of the conditions the proposed BFGS based on variables proved to be more efficient (in terms of number of iterations and function evaluations) as compared to $\ln \mathbf{K} - \mathbf{L}$. Yet, with a late switch, for

conditions near the phase boundaries, $\ln \mathbf{K} - \mathbf{L}$ was shown to reach convergence within fewer iterations. The $\ln \mathbf{K}$ variables lead to a better conditioned systems. When performing a late switch, the norm of the fugacity derivatives decreases and the off-diagonal elements become smaller, thus improving the scaling and the condition number of the Hessian. Close to the phase boundary, starting with a good initial guess (after a switch) $\ln \mathbf{K} - \mathbf{L}$ is based on the most efficient variables and leads to a faster convergence to the solution than the other methods. However, starting directly from the Wilson relation, a more robust method is necessary; in this case, $\ln \mathbf{K} - \mathbf{L}$ has been shown to be less stable than the proposed method.

Comparisons with the Newton algorithm were also presented. Generally $\alpha - \mathbf{L}$ requires a few more iterations to converge than the second order method. Each BFGS iteration is faster to compute than a Newton iteration since no fugacity derivatives are required and no linear systems need to be solved. Therefore, even if the BFGS algorithm requires more iterations to converge, the computation time is not necessarily increased. Even though the BFGS method may not compete with the Newton method for a small number of components with a cubic EOS, with a high number of component (in chemical process for instance), the proposed method could become more efficient in terms of computation time. Moreover, it offers an interesting alternative for difficult EOS models; when the fugacity derivatives are complex and difficult to calculate analytically, or there are costly, the BFGS method is really attractive as compared to the Newton method.

3.3.4.6 Conclusions/Perspectives

Since quasi-Newton methods are known to be very sensitive to scaling, a new methodology for finding a set of variables leading to a good scaling of phase equilibrium problems is proposed. A new set of independent variables is proposed for the flash calculation problem; for the phase stability problem, the variables introduced by [Michelsen [1982a]] are found using the proposed methodology. The BFGS algorithm based on this new set of independent variables for the two-phase-split problem was developed and tested for robustness and efficiency. The proposed method was shown to reach convergence for all the testing conditions and to converge generally within fewer iterations and function evaluations than other BFGS methods. Comparisons with the second-order Newton method showed that the proposed algorithm requires only few more iterations for convergence (but the cost per iteration is lower, since no derivatives or resolution of a linear system are required). The BFGS method is attractive for more complex EOS models in which partial derivatives of fugacity can be costly and/or difficult to compute analytically (as for instance the SAFT family of EOS). The proposed variables are also particularly suited for trust-region algorithms [Petitfrere and Nichita [2014a]].

3.4 Robust and efficient Trust-Region based stability analysis and multiphase flash calculations

3.4.1 Introduction

The resolution of the multiphase flash problem requires the minimization of the Gibbs free energy [Michelsen [1982a]] and of the tangent-plane distance (*TPD*) function [Michelsen [1982b]] for phase stability testing. Traditionally, first-order successive substitution (SSI) iterations are performed before switching to the second-order Newton method. In most of the cases, the Newton method works fine (convergence to the solution is achieved in a few iterations) provided the Hessian is positive definite. The most difficult regions in mixture phase envelopes are in the vicinity of singularities: critical points for multiphase flash calculations, convergence locus for negative flashes [Whitson and Michelsen [1990]], the stability test limit locus [Whitson and Michelsen [1990], Hoteit and Firoozabadi [2006], Nichita et al. [2007c]] (or the 'shadow curve' [Rasmussen et al. [2006]]) for stability analysis. Near these loci, algorithms either become extremely slow or have difficulties to converge. In the case of a too early switch, the Newton step is rejected and the calculations are switched back to SSI iterations. Sometimes a very large number of SSI iterations are necessary before the final switch to Newton iterations very close to singularities. This convergence behavior suggested us to investigate how Trust-Region

methods behave in such situations.

Restricted-step methods or Trust-Region methods were first suggested by [Levenberg [1944] and Marquardt [1963]] to solve nonlinear least-squares problems. They are often put in comparison with line search techniques, and they have seen considerable improvements in the past thirty years. The first algorithms were designed to small systems [Hebden [1973], Gill and Murray [1974], Gill and Murray [1978], Gay [1981], Moré and Sorensen [1983], Moré and Sorensen [1993]]. Then, a second branch of Trust-Region methods started to see developments since the improvements in computer science enabled to deal with bigger problems: the method based on high dimension problems, like the conjugate-gradients or generalized Lanczos-Trust-Region algorithms [Steihaug [1983], Shultz et al. [1985], Byrd et al. [1988]]. Except when dealing with a high number of components, this category is not suited for phase equilibrium calculations.

The first to introduce a Trust-Region method for equilibrium flash calculations were [Nghiem et al. [1983]]. They applied Powell's Dogleg method to the two-phase flash calculations. The same method was then used by [Lucia and Liu [1998]]. The dogleg method is a linear combination between a Cauchy step and Newton method. [Nghiem et al. [1983]] adapted this method as a linear combination between the SSI and Newton iterations; the method fails when the Hessian is not positive definite. [Mehra et al. [1982]] extended this method to multiphase equilibrium calculations.

[Michelsen [1992], Michelsen and Mollerup [2007]] addressed the problem of non-positive definite Hessians (revealed during decomposition) in phase equilibrium calculations and suggested the use of Trust-Region methods. If the Hessian matrix \mathbf{H} is not positive definite, it is corrected by adding a diagonal element, that is, $\mathbf{H} + \lambda \mathbf{I}$. Then some other authors have been investigating the Trust-Region methods in phase equilibrium calculations: [Trangenstein [1987]] reported minimization with linear constraints for phase stability and multiphase flash. [Kaul and Thrasher [1996]] used a Trust-Region procedure similar to [Hebden [1973]] for two-phase equilibrium flash calculations using a set of reduced variables. [Pan and Firoozabadi [2003]] also used a Trust-Region method; however, there are certain difficulties in combining the Trust-Region approach with reduction methods.

[Lucia et al. [1993]] developed a Trust-Region method extending the Powell dogleg strategy to the complex domain; this method was used by [Gow et al. [1996]] to model the VLE of binary refrigerant mixtures. [Lucia and Yang [2003]] used a Trust-Region method to calculate downhill directions in terrain global methods; these methods were used to model the complex phase behavior of normal alkane systems [Lucia et al. [2012]], or within a multi-scale framework for multiphase flash calculations [Lucia et al. [2005]]. [Alsaifi and Englezos [2011]] used a Trust-Region Gauss-Newton method for simultaneous stability and flash calculations with the PC-SAFT equation of state. Recently, [Michelsen et al. [2013b]] reported the use of a Trust-Region method for phase equilibrium calculations.

In this chapter, a Trust-Region method derived from [Moré and Sorensen [1983], Moré and Sorensen [1993], and Conn et al. [2000]] is adapted and tested successfully for various phase stability analysis and multiphase flash calculation problems. The section is structured as follows: the proposed Trust-Region method is presented in detail. The results for phase stability and multiphase flash calculations of various mixtures with complex phase envelopes and comparisons between the proposed method with Newton methods using various independent variables are presented before drawing the conclusions.

3.4.2 Trust-Region methods

Trust-Region methods define a region around the current point within which they trust the model to be an appropriate representation of the objective function. Then, a step is chosen which will minimize the model

within this region.

The Taylor development of the objective function f (which in our case can be either G or D_M) around \mathbf{x}_k is

$$f(\mathbf{x}_k + \mathbf{s}) = f_k + \mathbf{g}_k^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{B}_k \mathbf{s} + O(\|\mathbf{s}\|^3) \quad (3.206)$$

with $f_k = f(\mathbf{x}_k)$, $\mathbf{g}_k = \nabla f(\mathbf{x}_k)$ and $\mathbf{B}_k = \nabla^2 f(\mathbf{x}_k)$.

Taylor's development of the gradient gives

$$\mathbf{g}(\mathbf{x}_k + \mathbf{s}) = \mathbf{g}(\mathbf{x}_k) + \mathbf{B}_k \mathbf{s} + O(\|\mathbf{s}\|^2) \quad (3.207)$$

Minimizing the function f means getting the gradient $\mathbf{g} = \nabla f$ to be zero. Therefore, trying to put $\mathbf{g}(\mathbf{x}_k + \mathbf{s}) = \mathbf{0}$ we obtain the Newton method, which is quadratic and consists in solving

$$\mathbf{B}_k \mathbf{s} = -\mathbf{g}(\mathbf{x}_k) \quad (3.208)$$

3.4.2.1 The Trust-Region subproblem

For solving the system by Newton iterations, the Hessian must be positive definite, otherwise the method cannot be applied. In the Trust-Region method, the Hessian is corrected to ensure positive definiteness. If \mathbf{B}_k is not positive definite, by adding a diagonal element to the Hessian, $\mathbf{H}_k = \mathbf{B}_k + \lambda \mathbf{I}$, it becomes definite positive, for a value $\lambda > \max(0, -\lambda_1 + \varepsilon)$, where λ_1 is the smallest eigenvalue of the Hessian matrix.

Using (eq. 3.206), a second order approximation leads to:

$$f(\mathbf{x}_k + \mathbf{s}) \approx f_k + \mathbf{g}_k^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{B}_k \mathbf{s} \quad (3.209)$$

Solving the Trust-Region subproblem means finding the minimum

$$\min_{\|\mathbf{s}\| \leq \Delta_k} m_k(\mathbf{s}) = f_k + \mathbf{g}_k^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{B}_k \mathbf{s} \quad (3.210)$$

Looking more in details to the real step,

$$f(\mathbf{x}_k + \mathbf{s}) = f_k + \mathbf{g}_k^T \mathbf{s} + \frac{1}{2} \mathbf{s}^T \mathbf{H}_k \mathbf{s} - \frac{\lambda}{2} \mathbf{s}^T \mathbf{s} + O(\|\mathbf{s}\|^3) \quad (3.211)$$

The Trust Region carries out steps between the gradient descent (first order) and the Newton (second order) steps which are generally supralinear.

If the Hessian is positive definite, and if $\|-\mathbf{B}_k^{-1} \mathbf{g}_k\| \leq \Delta$, a Newton-step is carried out.

3.4.2.2 Solving the Trust-Region subproblem

The main difficulty relies on solving (eq. 3.210). Two methods exist: near exact methods (the one used in this thesis), and approximated methods (ideal for systems with high dimensionality). It has been proven [Nocedal and Wright [2006]] than solving the Trust-Region subproblem (eq. 3.210) is equivalent to solving the problem for:

$$(\mathbf{B}_k + \lambda \mathbf{I}) \mathbf{s} = -\mathbf{g} \quad (3.212)$$

$$\lambda (\Delta - \|\mathbf{s}\|) = 0 \quad (3.213)$$

$$(\mathbf{B}_k + \lambda \mathbf{I}) \text{ is positive semidefinite} \quad (3.214)$$

For $\lambda \geq 0$ and \mathbf{s} feasible.

Three cases can be distinguished.

3.4.2.2.1 Case 1: $\lambda = 0$ and $\|\mathbf{s}\| \leq \Delta_k$

If $\lambda = 0$ satisfies (eq. 3.213 and eq. 3.214), with $\|\mathbf{s}\| \leq \Delta$, the solution is found (Newton step). Otherwise, the system (eq. 3.213, eq. 3.214, eq. 3.214) is equivalent to solving

$$\|(\mathbf{B}_k + \lambda \mathbf{I})^{-1} \mathbf{g}_k\| = \Delta_k \quad (3.215)$$

With $(\mathbf{B}_k + \lambda \mathbf{I})$ definite positive.

[Conn et al. [2000]] proposed different examples to illustrate the possible roots of the equation $\|s(\lambda)\|_2^2 \leq \Delta$

Let's suppose the problem defined by

$$\mathbf{g} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}$$

\mathbf{B} is positive definite. In fig. 3.40 $\|s(\lambda)\|_2^2$ is plotted against λ .

The case 1 corresponds to the case where $\Delta > 1.5$ (dashed red line in the fig. 3.40), the solution $\lambda = 0$, lies inside the Trust-Region radius. And as it corresponds to the Newton step, this is the optimal one.

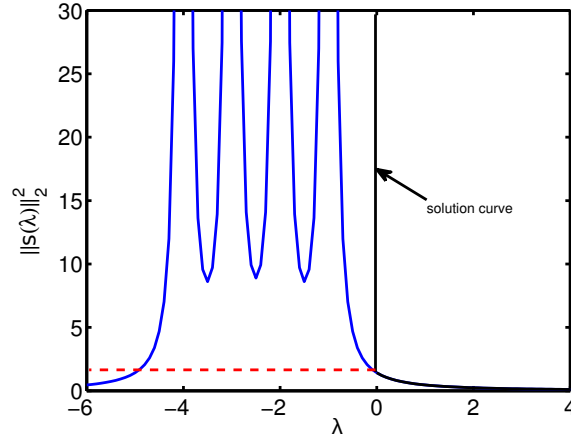


Figure 3.40: Convex example [Conn et al. [2000]]

3.4.2.2.2 Case 2: $\mathbf{q}_1^T \mathbf{g} \neq 0$ and $\lambda \in (-\lambda_1, +\infty)$

By noting that \mathbf{s} is a function of λ , and

$$\mathbf{s}(\lambda) = -(\mathbf{B}_k + \lambda \mathbf{I})^{-1} \mathbf{g}_k \quad (3.216)$$

(eq. 3.215) becomes

$$\|\mathbf{s}(\lambda)\| = \Delta_k \quad (3.217)$$

If \mathbf{B}_k is positive definite, since \mathbf{B}_k symmetric, there is an orthogonal matrix \mathbf{Q} and a diagonal matrix $\mathbf{\Lambda}$ such that $\mathbf{B}_k = \mathbf{Q} \mathbf{\Lambda} \mathbf{Q}^T$, where $\mathbf{\Lambda} = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_N)$ with $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$ are the eigenvalues of \mathbf{B}_k and the matrix \mathbf{Q} contains the orthonormal eigenvectors of \mathbf{B}_k

$$\mathbf{H}_k = \mathbf{B}_k + \lambda \mathbf{I} = \mathbf{Q}(\mathbf{\Lambda} + \lambda \mathbf{I}) \mathbf{Q}^T; \quad \forall \lambda \neq \lambda_j \quad (3.218)$$

then,

$$\mathbf{s}(\lambda) = -\mathbf{Q}(\mathbf{\Lambda} + \lambda \mathbf{I})^{-1} \mathbf{Q}^T \mathbf{g} = -\sum_{j=1}^N \frac{\mathbf{q}_j^T \mathbf{g}}{\lambda_j + \lambda} \mathbf{q}_j \quad (3.219)$$

Where \mathbf{q}_j denotes the j th column of \mathbf{Q} . By orthonormality

$$\|\mathbf{s}(\lambda)\|_2^2 = \sum_{j=1}^N \frac{(\mathbf{q}_j^T \mathbf{g})^2}{(\lambda_j + \lambda)^2} \quad (3.220)$$

therefore, as long as $\mathbf{q}_1^T \mathbf{g} \neq 0$, a unique solution $0 \leq -\lambda_1 < \lambda$ can be found to solve (eq. 3.215).

Solving (eq. 3.217) is equivalent to solve [Hebden [1973]]

$$\varphi(\lambda) = \frac{1}{\|\mathbf{s}(\lambda)\|_2} - \frac{1}{\Delta_k} = 0 \quad (3.221)$$

(eq. 3.221) can be solved with a Newton procedure in one dimension. The derivative of $\varphi(\lambda)$ is

$$\varphi'(\lambda) = -\frac{\langle \mathbf{s}(\lambda), \nabla_{\lambda} \mathbf{s}(\lambda) \rangle}{\|\mathbf{s}(\lambda)\|_2^3} \quad (3.222)$$

If \mathbf{H} is definite positive, a Cholesky decomposition can be performed

$$\mathbf{H} = \mathbf{L} \mathbf{L}^T \quad (3.223)$$

and using $\mathbf{H}(\lambda) = \mathbf{B}_k + \lambda \mathbf{I}$ the scalar product in (eq. 3.222) can be expressed as

$$\langle \mathbf{s}(\lambda), \mathbf{H}^{-1}(\lambda) \mathbf{s}(\lambda) \rangle = \langle \mathbf{s}(\lambda), \mathbf{L}^{-T} \mathbf{L}^{-1} \mathbf{s}(\lambda) \rangle = \langle \mathbf{L}^{-1} \mathbf{s}(\lambda), \mathbf{L}^{-1} \mathbf{s}(\lambda) \rangle = \|\mathbf{w}\|_2^2 \quad (3.224)$$

The update is then given by

$$\lambda^{k+1} = \lambda^k - \frac{\varphi(\lambda)}{\varphi'(\lambda)} = \lambda^k + \left(\frac{\|\mathbf{s}(\lambda)\|_2 - \Delta_k}{\Delta_k} \right) \left(\frac{\|\mathbf{s}(\lambda)\|_2^2}{\|\mathbf{w}\|_2^2} \right) \quad (3.225)$$

[Conn et al. [2000]] proposed a case where \mathbf{B} is non-positive definite:

$$\mathbf{g} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In fig. 3.41 $\|\mathbf{s}(\lambda)\|_2^2$ is plotted against λ for the non-convex example.

The case 2 corresponds to two possible problems:

- In the convex example, if $\Delta^2 \leq 1.5$ (which corresponds to the dashed red line in fig. 3.40), (eq. 3.221) needs to be solved to reach the optimum and find the correct λ .
- In the non-convex example, as $\mathbf{H}(\lambda)$ must be positive definite, the only values of interest are those where $\lambda > \lambda_1$. With λ_1 the minus leftmost eigenvalue. For any values of Δ , a λ can be found to solve the trust-region subproblem.

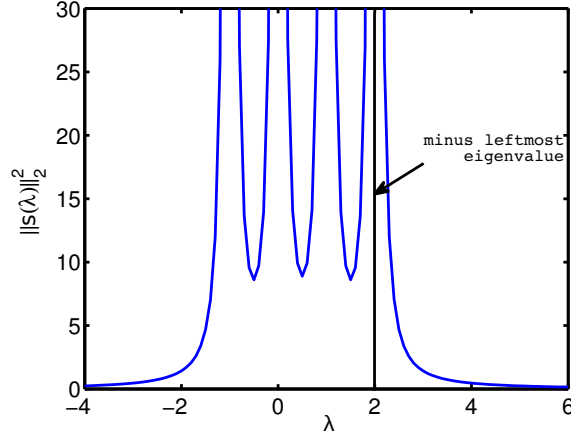


Figure 3.41: Non convex example [Conn et al. [2000]]

3.4.2.2.3 Case 3: $\mathbf{q}_1^T \mathbf{g} = 0$ and $\lambda \in (-\lambda_1, +\infty)$

This problem has been well studied in the literature [Conn et al. [2000], Nocedal and Wright [2006]] and is referred to as the hard case. It occurs when $\mathbf{q}_1^T \mathbf{g} = 0$, that is to say, the subspace made with the eigenvectors corresponding to the most negative eigenvalue of \mathbf{B}_k is orthogonal to the gradient.

In this case, if $\|\mathbf{s}(-\lambda_1)\| < \Delta_k$, the problem cannot be solved.

We have $(\mathbf{B}_k - \lambda_1 \mathbf{I})\mathbf{q}_1 = 0$. By calling

$$\mathbf{s}(\tau) = -(\mathbf{B}_k - \lambda_1 \mathbf{I})^{-1} \mathbf{g}_k + \tau \mathbf{q}_1 \quad (3.226)$$

By orthogonality $\mathbf{q}_j^T \mathbf{q}_1 = 0$, for $j \neq 1$, we get :

$$\mathbf{s} = \sum_{\substack{j=1 \\ \lambda_j \neq \lambda_1}}^N \frac{\mathbf{q}_j^T \mathbf{g}}{\lambda_j - \lambda_1} \mathbf{q}_j + \tau \mathbf{q}_1 \quad (3.227)$$

For any τ , and for $\|\mathbf{q}_1\|_2 = 1$, we get :

$$\|\mathbf{s}\|_2^2 = \sum_{\substack{j=1 \\ \lambda_j \neq \lambda_1}}^N \frac{(\mathbf{q}_j^T \mathbf{g})^2}{(\lambda_j - \lambda_1)^2} + \tau^2 \quad (3.228)$$

By calculating the correct τ , it is always possible to get (eq. 3.217).

Once again, [Conn et al. [2000]] proposed an example to illustrate the hard case:

$$\mathbf{g} = \begin{pmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} -2 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

In fig. 3.42 $\|s(\lambda)\|_2^2$ is plotted against λ for the hard case example.

As $\mathbf{H}(\lambda)$ must be positive definite, $\lambda < 2$ is forbidden. However, unlike for the non-convex case, there is no direct solution for the problem for any values of $\Delta > \Delta_{critic}$ (in fig. 3.42, $\Delta_{critic} = 1.2$).

One needs to use the procedure explained earlier to solve the problem

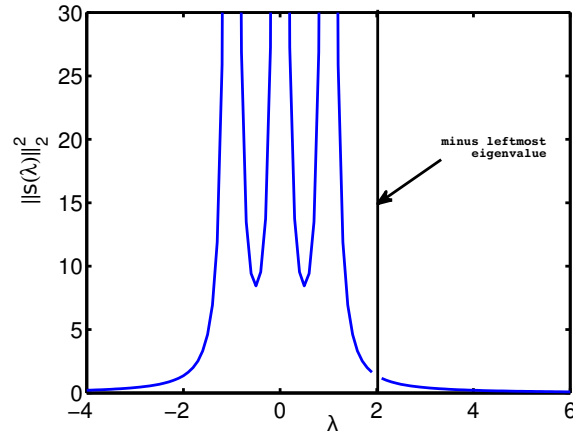


Figure 3.42: Hard case example [Conn et al. [2000]]

3.4.2.3 Algorithm for the Trust-Region subproblem

The Newton method to obtain the correct λ is not used directly. Safe-guarding techniques must be applied to get a robust algorithm. The general algorithm is given below. It follows the algorithms given in chapter 7 of [Conn et al. [2000]], which are directly derived from [Moré and Sorensen [1983], Moré and Sorensen [1993]].

Algorithm 3.4 realistic exact trust-Region subproblem [Conn et al. [2000]]

```

 $\lambda = 0$ 
Try to factorize  $\mathbf{H}(\lambda) = \mathbf{L}\mathbf{L}^T$ 
if Failure then
    Compute  $\lambda_1$ 
     $\lambda = -\lambda_1 + \epsilon$ 
    Factorize  $\mathbf{H}(\lambda) = \mathbf{L}\mathbf{L}^T$ 
end if
Solve  $\mathbf{L}^T \mathbf{s} = -\mathbf{g}$ 
if  $\|\mathbf{s}\|_2 < \Delta$  then
    if  $\lambda = 0$  or  $\|\mathbf{s}\|_2 = \Delta$  then
        STOP : the solution has been found
    else {Hard case}
        Compute an eigenvector  $\mathbf{u}_1$  corresponding to  $\lambda_1$  (e.g., Lanczos or QR)
        Find the root  $\alpha$  of the equation  $\|\mathbf{s} + \alpha \mathbf{u}_1\|_2 = \Delta$  which minimizes (eq. 3.210)
        Replace  $\mathbf{s}$  by  $\mathbf{s} + \alpha \mathbf{u}_1$ 
        STOP : the solution has been found
    end if
end if
while  $|\|\mathbf{s}\|_2 - \Delta| > \kappa_{easy} \Delta$  do
    Solve  $\mathbf{L}\mathbf{w} = \mathbf{s}$ 
    Replace  $\lambda^{k+1} = \lambda^k + \left( \frac{\|\mathbf{s}(\lambda)\|_2 - \Delta}{\Delta} \right) \left( \frac{\|\mathbf{s}(\lambda)\|_2^2}{\|\mathbf{w}\|_2^2} \right)$ 
    Factorize  $\mathbf{H}(\lambda) = \mathbf{L}\mathbf{L}^T$ 
    Solve  $\mathbf{L}^T \mathbf{s} = -\mathbf{g}$ 
end while
    
```

In the algorithm, it has been chosen to calculate the eigenvector and eigenvalues explicitly. [Conn et al. [2000]] proposed another algorithm in which the eigenvector can be approximated. The two methods have been implemented. As the system is generally small ($nc < 12$) in the context of reservoir simulators, it is faster to compute them explicitly. For higher number of components (i.e., chemical processes), the second method will be more advantageous.

3.4.2.4 The Trust-Region size

The size of the Trust-Region Δ_k is critical to the effectiveness of each step. If the region is too small, the algorithm misses an opportunity to take a step which will get closer to the solution. If it is too large, steps which should not be acceptable could be accepted.

3.4.2.4.1 Dynamic update of the Trust-Region size

The choice of the Trust-Region size is based on the agreement between the model function m_k (see eq. 3.210) and the objective function f at the previous iterations:

$$\rho_k = \frac{f(\mathbf{x}_k) - f(\mathbf{x}_k + \mathbf{s}_k)}{m_k(\mathbf{0}) - m_k(\mathbf{s}_k)} \quad (3.229)$$

- If $\rho_k < 0$, this means that $f(\mathbf{x}_k) < f(\mathbf{x}_k + \mathbf{s}_k)$, the step is rejected
- If ρ_k is small, this means that the Trust Region should be decreased (the model is relatively different to the reality (complex region)).
- If $\rho_k \approx 1$, the Trust Region should be extended, since the model matches the reality and better steps could be carried out.

3.4.2.4.2 General Algorithm for the Trust Region method

The general algorithm for the Trust Region method is adapted from [Nocedal and Wright [2006]]. Note that for difficult conditions, mainly in the early iteration stages, the Hessian is so ill-conditioned that the element added to the diagonal of the Hessian is really high. The Trust-Region radius decreases to a really small value, and once the Hessian is positive definite, the Trust-Region radius limits the step, and it can take long to increase it. When λ is too high, the step will be a gradient descent which is smaller than that of an SSI iteration. Therefore, the algorithm is modified in the sense that if $\lambda \geq \kappa$ and $\rho_k \leq \gamma$ an SSI iteration is performed. The use of only one SSI iteration is purely empirical. Based on numerical experiments on the mixtures considered in the results section, it was found that performing two or more SSI iterations at this stage does not decrease the computation time.

The main steps of the Trust-Region (**TR**) method are summarized below:

Trust-region methods contain tunable parameters (such as the initial trust-region size, the parameter controlling the increase/decrease of the trust-region size, etc.). Changing these parameters (which are typically problem-dependent) might affect to a high extent the convergence properties; their choice is empirical and has been based on many numerical experiments. The values giving the better results on average were adopted.

In this thesis, the initial trust-region radius is calculated by $\Delta_0 = \|\mathbf{x}\|/10$ for flash and by $\Delta_0 = \min(\|\mathbf{x}\|/10, 1/10)$ This is the most sensible parameter. It should be chosen not too large (a step not taking into account the irregularities of the function could be accepted) and not too small (the descent step would become too small and close to the one of SSI). This choice generally gives good results for the problems addressed here (multiphase flash and phase stability testing), but it might not be suited for other minimization problems. The parameter γ is set to zero in the present algorithm; this avoids switching back to SSI too often when the model function m_k is significantly different from the objective function f .

For phase stability testing, the problem becomes singular and any iterative method experience convergence problems. The Newton method is quadratically convergent only in a vicinity of the solution, which shrinks as a singularity is approached. In this case, Newton steps are often rejected and a switch back to SSI is performed (sometimes for a very large number of times). It will be shown later on in this chapter on several

Algorithm 3.5 Trust-Region algorithm: [Nocedal and Wright [2006]]

```

Given  $\Delta > 0$ ,  $\Delta_0 \in (0, \Delta)$  and  $\gamma \in [0, \frac{\Delta_k}{4})$ 
for  $k = 0 \rightarrow k_{max}$  do
  Calculate  $f_k, \mathbf{g}_k$ 
  if  $Error > \epsilon$  then
    Calculate  $\mathbf{B}_k$ 
    Calculate  $\rho_k$  solving (eq. 3.210) (TR subproblem)
    if  $\rho_k < \frac{1}{4}$  then
       $\Delta_{k+1} = \frac{\Delta_k}{4}$ 
    else
      if  $\rho_k > \frac{3}{4}$  and  $\|\mathbf{s}_k\| = \Delta_k$  then
         $\Delta_{k+1} = \min(2\Delta_k, \Delta)$ 
      end if
    end if
    if  $\rho_k > \gamma$  and  $\lambda < \kappa$  then
       $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$ 
    else
      SSI update
    end if
  end if
end for

```

examples how the Trust-Region method improves convergence (in some cases spectacularly) near the STLL. For flash calculations, convergence problems arise near phase boundaries (and particularly near critical points), and the Trust-Region method also performs better in this region than Newton's method (as will be shown on an example). High quality initial guesses for flash calculations from phase stability analysis are making these convergence problems less important near phase boundaries.

Close to phase boundaries, the trust-region can partially lie outside the feasible composition space if it is a radial trust-region. If an $Y_i < 0$ (for phase stability), or an $n_{im} \notin (0, z_i)$ (for flash calculations) after performing a trust-region step, the SSI method is used and ρ is set to 0. Indeed, if a variable goes out of its bounds it means that the trust-region radius should be decreased to get all the irregularities of the function and to be able to get closer to the phase boundary without crossing it. This is why ρ is set to 0 (making $\rho < 1/4$ and Δ is decreased).

The presence of trace components in the mixture introduces significant difficulties for phase equilibrium calculation methods. If the composition of a trace component is well below machine precision the Hessian matrix becomes very ill-conditioned, leading to highly severe convergence problems. In compositional reservoir simulators, it is really important to be able to deal with this issue, because trace components can occur in certain applications. In this thesis this problem is overcome by choosing the dependent variable differently for each component (as proposed by [Michelsen [1982a]], see section 2.2, **NLVM** method). This highly improves the condition number of the system, and it allows dealing with trace components.

It is important to note that, as for Newton and SSI methods, the Trust-Region method does not guarantee to give the global minimum of the objective function (TPD function or Gibbs free energy). This is a gradient based local method, which converges to a stationary point of the objective function. To guarantee a global minimum, either a global optimization method, or a global stability analysis mixed with a local minimization for the flash calculations should be used. However, these methods are quite slow and cannot be used within reservoir or process simulators. Here the use of multiple initial guesses for a local minimization of the TPD function in the stability testing allows to get the global minimum with a high degree of certainty (even if the solution is not theoretically guaranteed), providing in the same time a sculptured initialization for multiphase flash calculations.

3.4.2.4.3 Hybrid Newton-Trust Region method

In most cases, phase equilibrium problems (flash calculation and stability analysis) have to be solved for temperatures, pressures and feed compositions located away from the difficult regions of the phase envelope. From early iteration stages, the partial derivatives of the fugacity coefficients with respect to any classical variables are small enough to make the Hessian positive definite. Therefore, a convex problem has to be solved (there is no need to correct the Hessian) and a Newton method is very efficient. When the Trust-Region method is used in those cases, a Newton step is applied as seen previously, but it requires the computation of the norm of the descent direction vector step, as well as an evaluation of ρ at each iteration.

In order to avoid some unnecessary (and time consuming) calculations, a hybrid (Newton-Trust Region) method is proposed, in which a Newton procedure is tried first after the switch from SSI. Whenever the objective function (modified *TPD* function or Gibbs free energy) increases after the update of the variable, a switch is made to use the Trust-Region method. If there is no increase in the objective function, Newton iterations are performed until convergence. In the next section, this hybrid approach will be denoted **Y – TR** for stability testing and **NLVM – TR** for flash calculations.

3.4.3 Results

In this thesis, the numerical procedure used in multiphase equilibrium flash calculations is a sequential one (as first suggested by [Michelsen [1982a]]), and the initialization roughly follows the one proposed by [Li and Firoozabadi [2012]]. For phase stability testing, the switch criterion proposed by [Hoteit and Firoozabadi [2006]] is adopted. For flash calculations, the switching procedure is similar to that proposed by [Nghiem et al. [1983]]. For Newton iterations, a switch back to SSI iterations is carried out whenever, i) the Gibbs free energy (for flash calculations) or the modified *TPD* function (for phase stability) increases between two consecutive iterations, or ii) an independent variables goes out of its bounds (i.e., the bounds are: for mole numbers, $0 \leq n_{ij} \leq z_i$ in flash calculations; for formal mole numbers, $(Y_i \geq 0$ for stability testing); a switch back to SSI iterations is required generally for difficult conditions in the early stages of the iterative process (and indicates a too early switch). The stopping criterion is that the Euclidean norm of the error vector is less than a tolerance $\epsilon = 10^{-10}$.

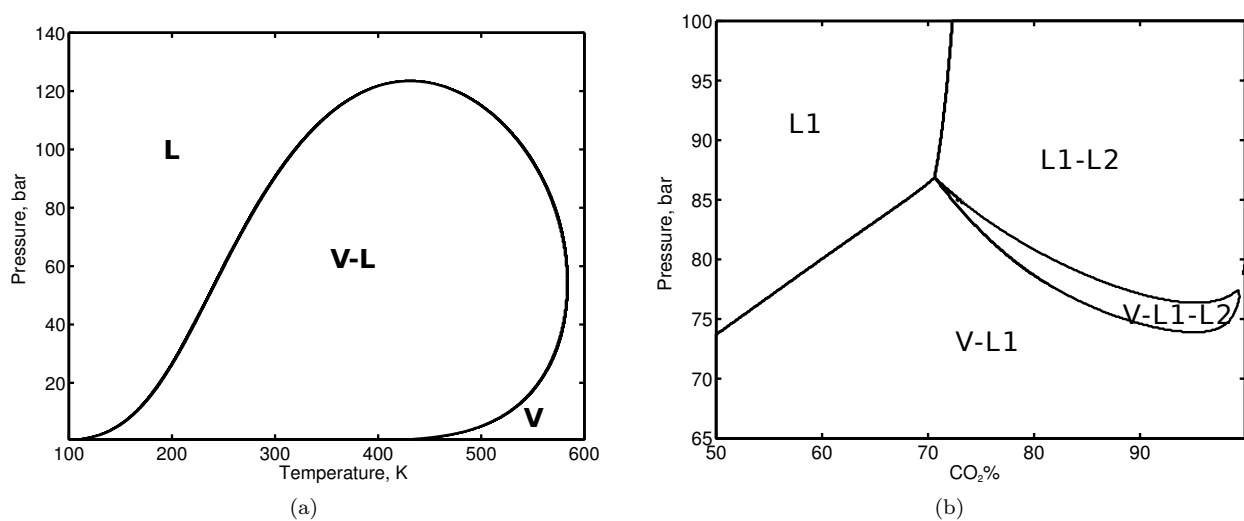


Figure 3.43: Phase envelope a) of the MY10 mixture b) of the Oil B mixture mixed with CO₂ at T=307.6K

In this section, the Trust-Region procedure described above is applied to phase stability analysis and two- and three-phase flash calculations for five synthetic and natural hydrocarbon mixtures, with different amount of

classical contaminants (CO_2 , H_2S) susceptible to drive the formation of a third phase for four of these mixtures.

The first mixture is the ten-component MY10 [Metcalf and Yarborough [1979]] (phase envelope in fig. 3.43a) mixture (already presented). The next four mixtures are: the Maljamar reservoir oil and Maljamar separator oil from [Orr et al. [1981]], the Oil B mixture from [Shelton and Yarborough [1977]] and the sour gas mixture from [Robinson et al. [1978]]. All relevant data for these mixtures are taken from [Li and Firoozabadi [2012]]. These four mixtures are mixed in different proportions with carbon dioxide. Fig. 3.18a depicts the p-z phase envelope of the Maljamar separator oil mixed with CO_2 at $T = 305.35$ K and fig. 3.18b the P-z phase envelope of the Maljamar reservoir oil mixed with CO_2 at the same temperature. Phase envelopes of the Oil B CO_2 (at $T = 307.6$ K) and of the sour gas CO_2 (at $T = 178.8$ K) are given in fig. 3.43b and fig. 3.19a, respectively. Note that the four mixtures are exhibiting complicated phase envelopes; some calculations are performed at really difficult conditions, such as the vicinity of bi-critical points. The Peng-Robinson equation of state [Peng and Robinson [1976], Robinson and Peng [1978]] has been used in all calculations.

3.4.3.1 Phase stability testing

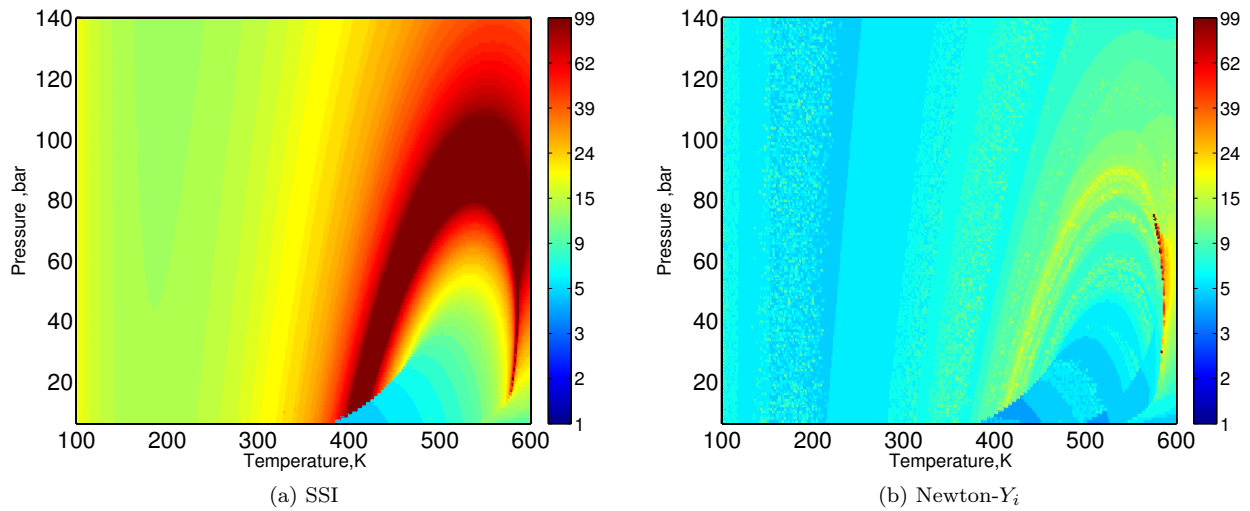


Figure 3.44: Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i/K_i$)

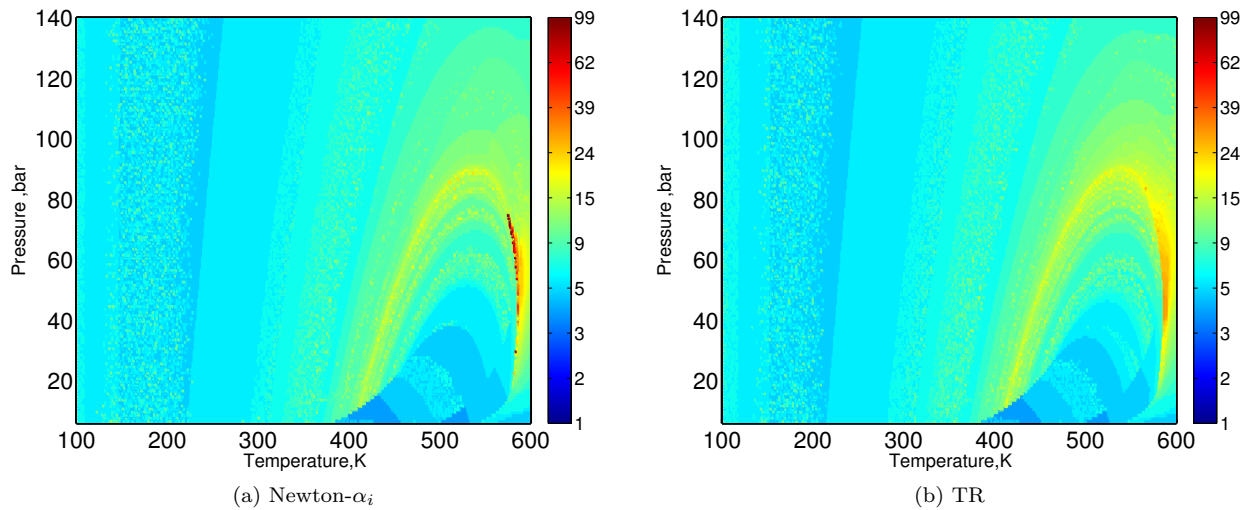


Figure 3.45: Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i/K_i$)

For phase stability analysis, five methods have been compared: the SSI iteration, the SSI-Newton method with Y_i and α_i as independent variables, the Trust-Region (**TR**) method performed over the Y_i variables, and

the hybrid **Newton** – **TR** method (denoted **Y** – **TR**). The number of iterations required by phase stability testing over the whole P-T diagram of the MY10 mixture is given for different methods, with a liquid trial phase ($Y_i^{(0)} = z_i/K_i$) in fig. 3.44a, fig. 3.44b, fig. 3.45a, fig. 3.45b and with a vapor trial phase ($Y_i^{(0)} = z_i K_i$) in fig. 3.46a to fig. 3.47b. The steps between two $P - T$ points on the phase diagrams are $\Delta P = 0.1$ bar and $\Delta T = 0.1$ K. Less than ten iterations are required for convergence by Newton and **TR** methods on most regions of the phase envelope. An increase in the number of iterations can be clearly seen for all methods around the STLL (in fig. 3.44a to fig. 3.45b for $T > T_C$ and in fig. 3.46a to fig. 3.47b for $T < T_C$) and the spinodal (in fig. 3.44a, to fig. 3.45b for $T < T_C$ and in fig. 3.46a to fig. 3.47b for $T > T_C$), which are the singularities for phase stability. The deterioration of convergence properties as the STLL (non-trivial solution at a positive TPD function) is approached, is far more severe for Newton and **TR** methods than near the spinodal (trivial solution for TPD function equal to zero). It also appears from fig. 3.44a to fig. 3.47b that near the STLL **Newton** – α_i performs better than **Newton** – Y_i , and that **TR** is faster than both Newton versions.

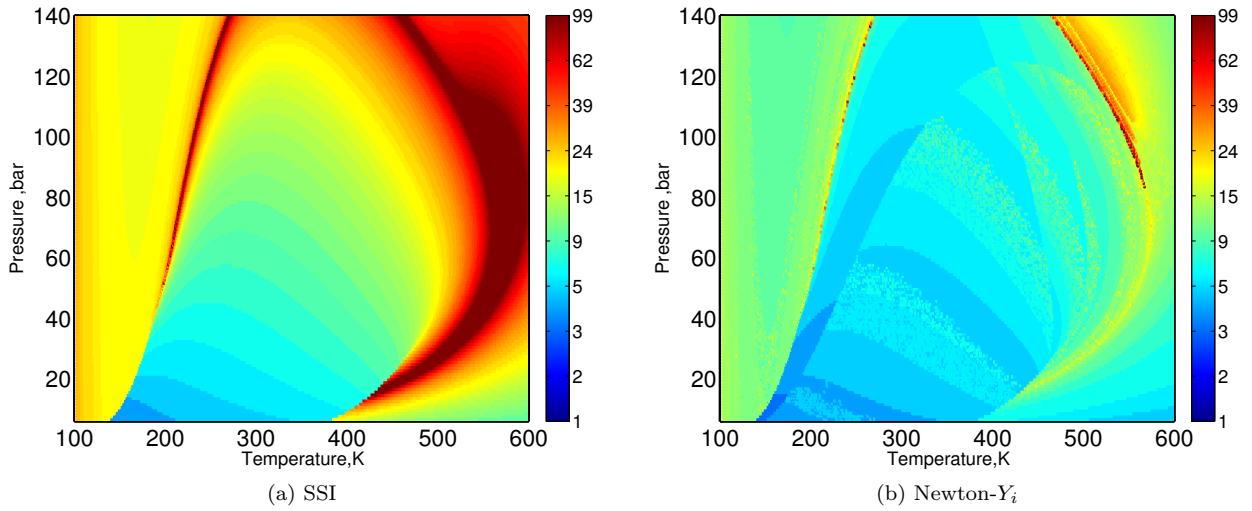


Figure 3.46: Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i K_i$)

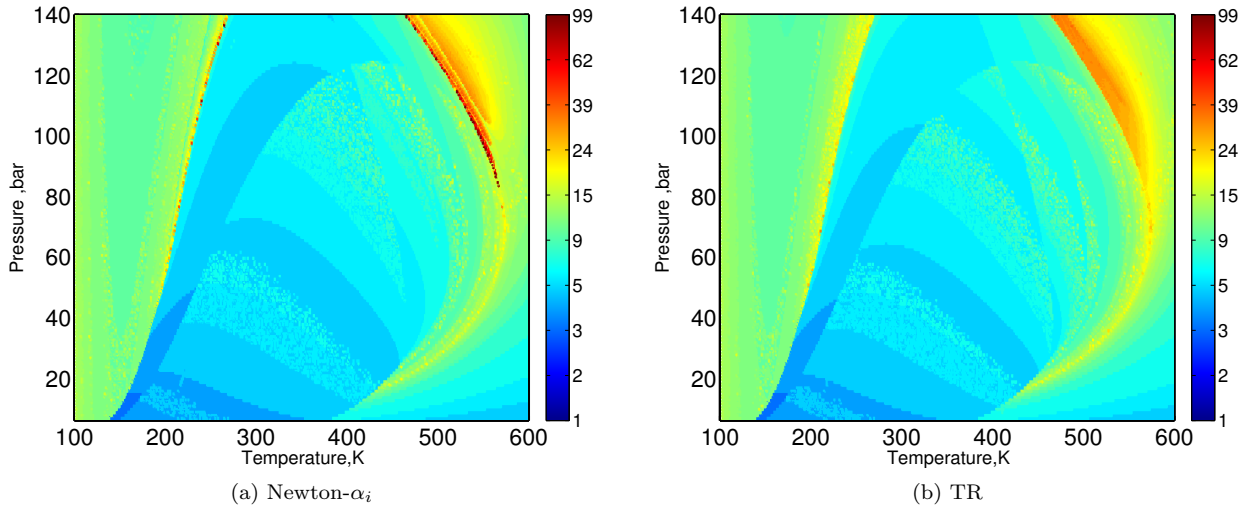


Figure 3.47: Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i K_i$)

The Euclidean norm of the gradient vector is plotted vs. the iteration level for stability testing ($Y_i^{(0)} = z_i K_i$) at $T = 540.5$ K in fig. 3.48a and fig. 3.48b at two pressures above the STLL (103.46 bar). For SSI iterations, the error curve decreases but exhibits an inflection point; **TR** performs better than Newton methods (at $p = 112$ bar, **TR** requires about half of the iterations required by Newton methods). Convergence properties deteriorate as one gets closer to the STLL: at $P = 103.5$ bar (very close to the STLL) the gradient norm

starts to increase during iterations (see fig. 3.49a) at a certain point (the objective function decreases very slowly, with about 10^{-6} per iteration; a descent direction is guaranteed with SSI iterations, since in the iteration equation of SSI, an always positive semi-definite matrix multiplies the gradient vector [Heidemann and Michelsen [1995], Ammar and Renon [1987]]). Newton methods fail to converge (the Hessian matrix is not positive definite anymore) at the given switching conditions, and a switch back is repeatedly required; eventually quadratic convergence is obtained, but after more than one hundred SSI iterations (most of them costly, since second-order information is calculated but finally not used). The **TR** region method converges in less than 20 iterations. At even more severe conditions, hundreds of SSI iterations (on switch back from Newton or using a very restrictive switch) may be necessary before the Newton method converges (a plot is given for $T = 530.5$ K and $P = 108.99$ bar in fig. 3.49b), while the **TR** method manages to handle these extremely difficult conditions within some twenty iterations, with an almost quadratic convergence behavior. For phase stability, the critical point is a global minimum, while the STLL is a saddle point; its vicinity is the most difficult region of the phase envelope for phase stability testing calculations.

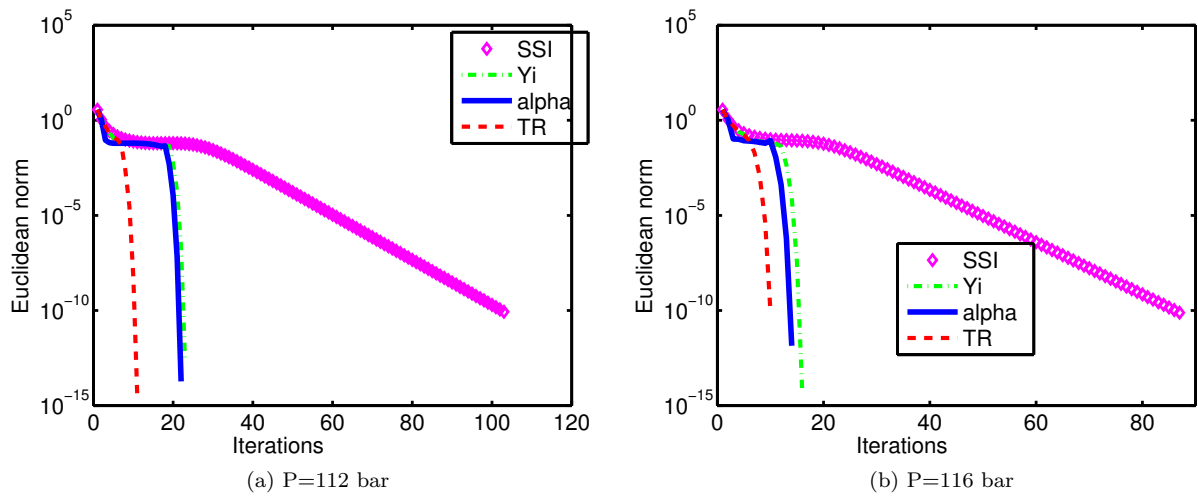


Figure 3.48: Error vs. iteration level for the stability analysis (liquid trial phase, $Y_i = z_i K_i$) of the MY10 mixture at $T=540.5$ K

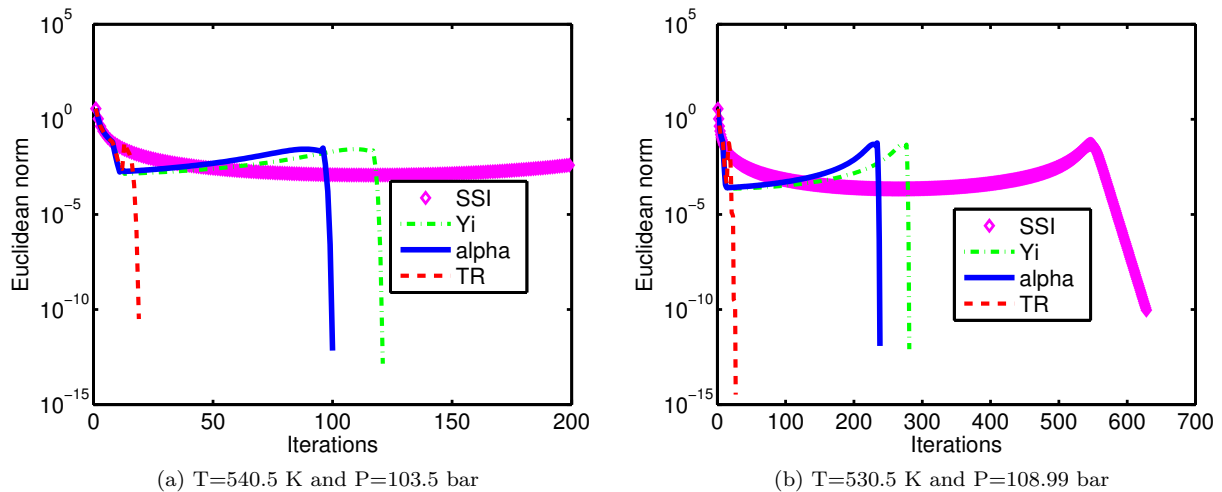


Figure 3.49: Error vs. iteration level for the stability analysis (liquid trial phase, $Y_i = z_i K_i$) of the MY10 mixture

For vapor phase stability, table 3.6 lists the average number of iterations for various methods, and table 3.7 gives the required CPU time (in seconds, for an Intel Xeon CPU Processor X5680, 3.33 GHz). For liquid phase stability, the average number of iterations is given in table 3.8 and the computer time in table 3.9. The

examined points are those used in constructing fig. 3.18a to fig. 3.19a.

Method	Mixture			
	Maljamar Oil (Res)	Maljamar Oil (Sep)	Oil B	Sour gas
Y	8.82	8.54	10.09	5.91
α	7.96	7.59	9.70	5.67
TR	8.79	8.39	9.91	6.13
Y-TR	7.95	7.69	9.13	5.75

Table 3.6: Average number of iterations for four mixtures and various stability testing methods over the P-z diagram (initial guess, $Y_i = z_i K_i$)

Method	Mixture			
	Maljamar Oil (Res)	Maljamar Oil (Sep)	Oil B	Sour gas
Y	1.679	4.77	5.248	2.645
α	1.565	4.27	5.135	2.735
TR	1.841	5.50	7.835	3.481
Y-TR	1.504	4.56	6.762	2.937

Table 3.7: Computation time (in seconds) for four mixtures and various stability testing methods over the P-z diagram (initial guess $Y_i = z_i K_i$)

Method	Mixture			
	Maljamar Oil (Res)	Maljamar Oil (Sep)	Oil B	Sour gas
Y	6.73	8.59	9.17	10.33
α	6.74	8.72	8.82	10.18
TR	6.73	8.59	9.16	9.71
Y-TR	6.74	8.61	8.82	9.61

Table 3.8: Average number of iterations for four mixtures and various stability testing methods over the P-z diagram (initial guess, $Y_i = z_i / K_i$)

Method	Mixture			
	Maljamar Oil (Res)	Maljamar Oil (Sep)	Oil B	Sour gas
Y	1.133	3.965	4.051	4.275
α	1.147	4.122	3.823	4.361
TR	1.159	4.181	4.385	4.497
Y-TR	1.108	4.109	3.893	4.191

Table 3.9: Computation time (in seconds) for four mixtures and various stability testing methods over the P-z diagram (initial guess $Y_i = z_i / K_i$)

In terms of numbers of iterations, **Newton** – α_i performs systematically better than **Newton** – Y_i . **TR** requires more iterations than **Newton** – α_i for liquid phase stability ((table 3.6), while the trend is inversed for vapor phase stability (table 3.7). The hybrid approach **Y** – **TR** performs globally better than both **TR** and **Newton** – α_i .

In terms of computation time, **TR** is faster than **Newton** – Y_i , and slower than **Newton** – α_i , (the cost of an iteration is higher for **TR** due to additional linear algebra operations), but the hybrid method **Y** – **TR** is globally the fastest method.

3.4.3.2 Multiphase flash calculations

For the equilibrium flash calculations, five methods have been compared: the SSI iteration, the **SSI** – **Newton** method with $\ln K_i$ as independent variables (abbreviated below as **lnK**) and with the improved mole number

variables (**NLVM**), the Trust-Region method performed over the improved mole number variables, and the hybrid Newton **NLVM** – **TR** method. It is observed that the Newton method using mole numbers as independent variables performs really bad for difficult multiphase conditions as compared to the other methods, thus Newton **NLV** is not included in the comparison.

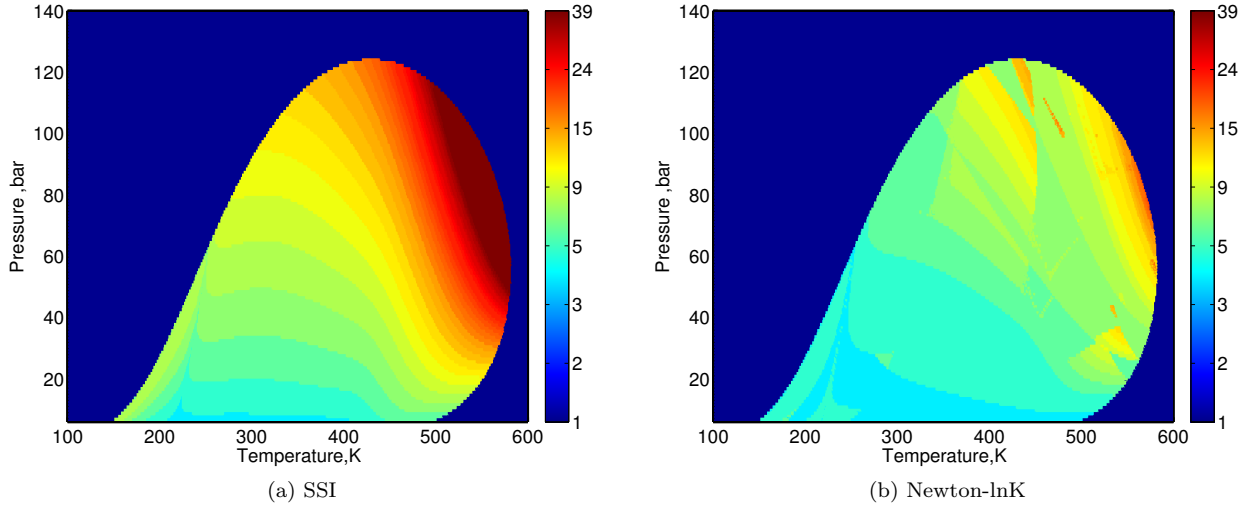


Figure 3.50: Number of iterations for the MY10 mixture (two-phase flash)

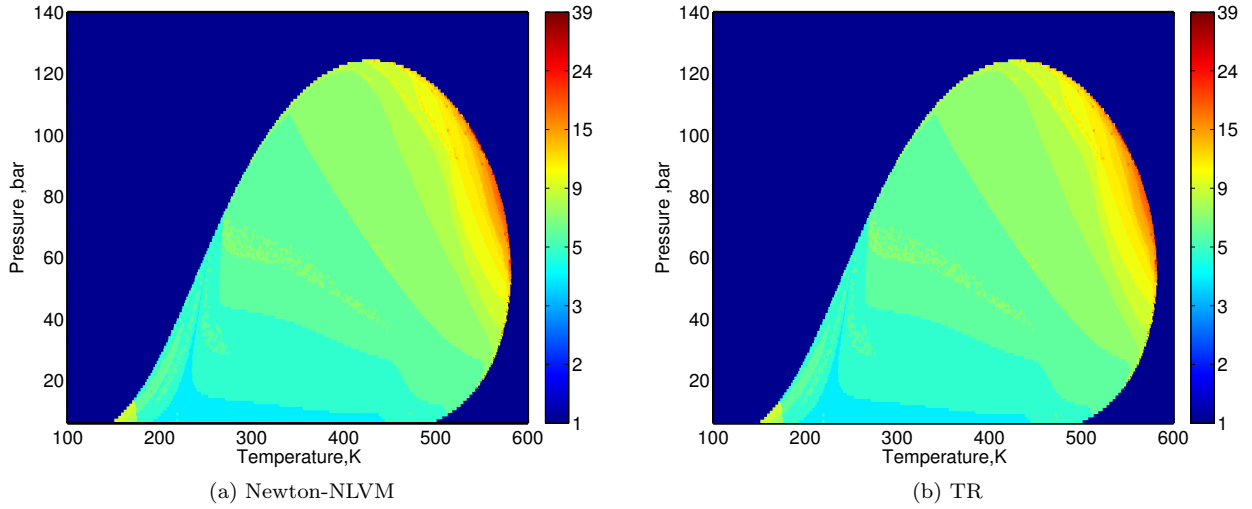


Figure 3.51: Number of iterations for the MY10 mixture (two-phase flash)

The number of iterations to reach the convergence starting from Wilson initialization with various flash calculations methods in the two-phase region is represented in the P-T plane for the MY10 mixture in fig. 3.50a to fig. 3.51b. The steps between two P-T points on the diagram are $\Delta p = 0.1$ bar and $\Delta T = 0.1$ K. Newton and **TR** methods converge rapidly except in the vicinity of the critical point (the singularity for flash calculations); in this region, the **TR** method performs better than Newton methods (**lnK** perform better than **NLVM**).

A comparison of the convergence behavior (Euclidean norm versus the of iteration number) between various flash calculation algorithms (initialization by K-Wilson) for the Oil B mixture combined with 95% moles CO₂ at T=307.6 K is shown in fig. 3.52a (at $p = 73.6$ bar) and fig. 3.52b (at $p = 76.9$ bar). This kind of behavior is systematic for difficult conditions.

The number of iterations in flash calculations using various methods (a-SSI; b-**Newton** – **lnK**; c-**Newton** – **NLVM**; d- **TR**; e- **NLVM** – **TR**.) is represented in P-z diagrams in fig. 3.53a to fig. 3.55 (Maljamar reservoir mixture/CO₂ at $T = 305.35$ K), fig. 3.56a to fig. 3.56e (Maljamar separator mixture/CO₂

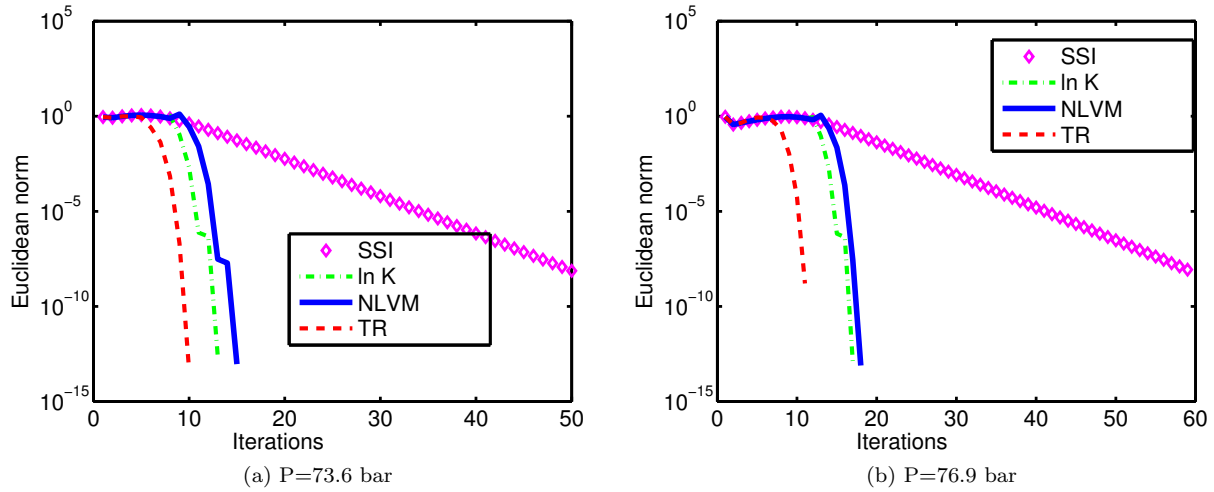


Figure 3.52: Comparison between various flash calculation algorithms (initialization by K-Wilson) for the Oil B mixture/95% CO_2 at $T=307.6$ K

at $T = 305.35$ K), fig. 3.57a to fig. 3.57e (Oil B/ CO_2 at $T = 307.6$ K) and fig. 3.58a to fig. 3.58e (sour gas/ CO_2 at $T = 178.8$ K). The steps between two points on the diagrams are $\Delta p = 0.1$ bar and $\Delta (\%\text{CO}_2)=0.1$. In the three-phase region, the number of iterations given in fig. 3.53a to fig. 3.58e is the sum of the iterations for the two-phase and three phase flash calculations. In all the graphs, the Trust-Region and hybrid algorithms globally give smaller iteration numbers (for both two-phase and three-phase regions of the phase envelopes) than all the other methods tested.

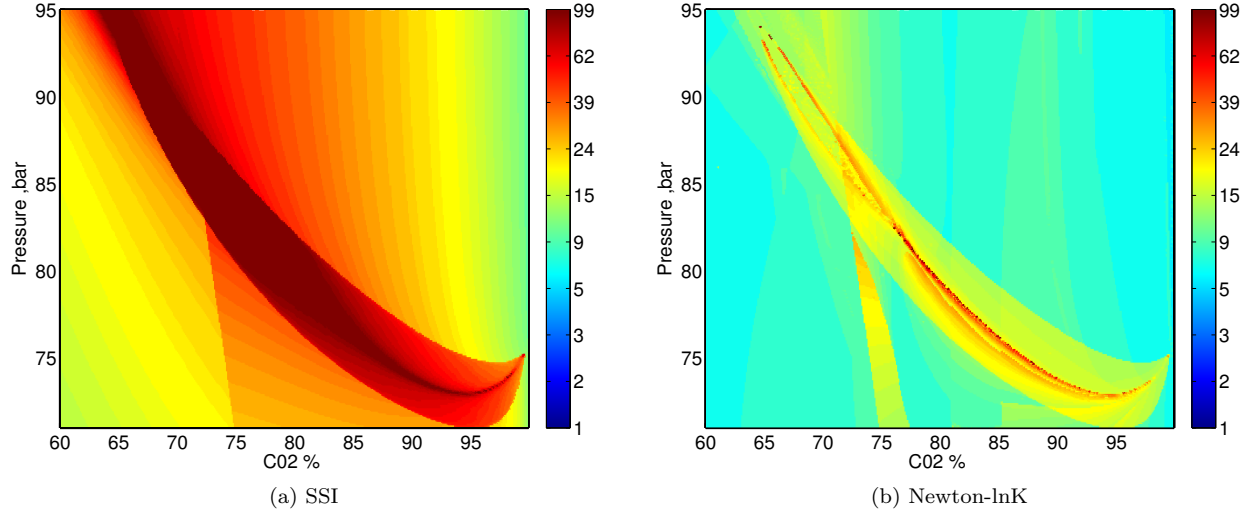


Figure 3.53: Number of iterations (two-and three-phase flashes) for Maljamar reservoir mixture over the Pz-diagram at $T=305.35$ K

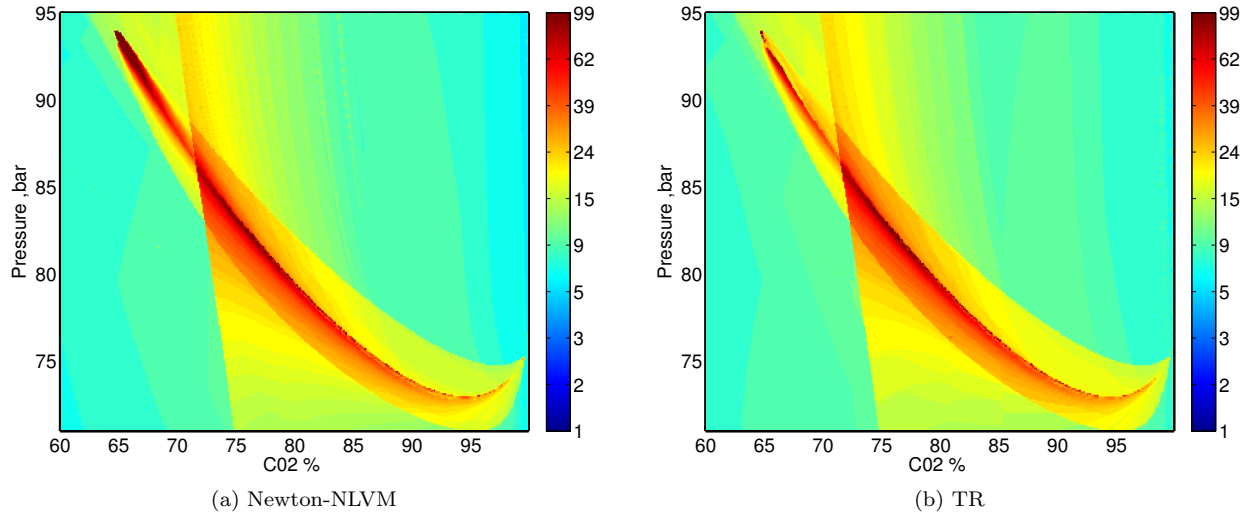


Figure 3.54: Number of iterations (two- and three-phase flashes) for Maljamar reservoir mixture over the Pz-diagram at $T=305.35$ K

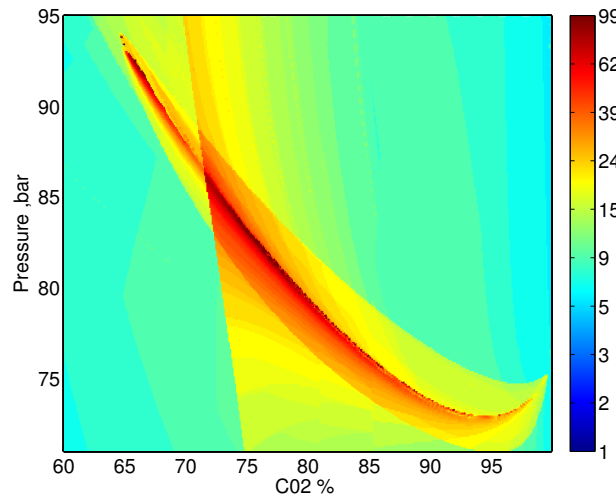


Figure 3.55: Number of iterations (two- and three-phase flashes) for Maljamar reservoir mixture over the Pz-diagram at $T=305.35$ K: NLVM-TR

For the three-phase region of the sour gas CO_2 mixtures at $T = 178.8$ K and $p = 20$ bar, the global minimum of the modified TPD function vs. the molar amounts of CO_2 is plotted in fig. 3.59a, and the number of iterations of Newton and **TR** methods are plotted in fig. 3.59b (the difference between the total number of iterations and the sum of SSI iterations before switch and Newton iterations represents the number of SSI iterations performed after a switch back from unsuccessful Newton iterations). The peak in the total number of iterations corresponds to the angular point of the minimum TPD (giving the worst initial three-phase distribution from phase stability). For less than about 56 % moles CO_2 , the **TR** and hybrid methods perform remarkably well as compared to the Newton ones. For more than 56 % moles CO_2 , all methods perform equally well. Note that if a late switch is adopted (at 10^{-5} as in [Li and Firoozabadi [2012]]), the peak in the total number of iterations would be at more than 100 iterations. The error (Euclidean norm) vs. the iteration level for three-phase flash of the sour gas mixture at $T = 178.8$ K, $P=20$ bar and 51% moles CO_2 . is plotted for various methods in fig. 3.59b: the Newton methods require 18 iterations (**lnK**) and 21 iterations (**NLVM**), while the **TR** method requires only 8 iterations. It should be also noted that when switching back from Newton to SSI, the Hessian is calculated, but second order information is finally not used; the **TR** uses this information.

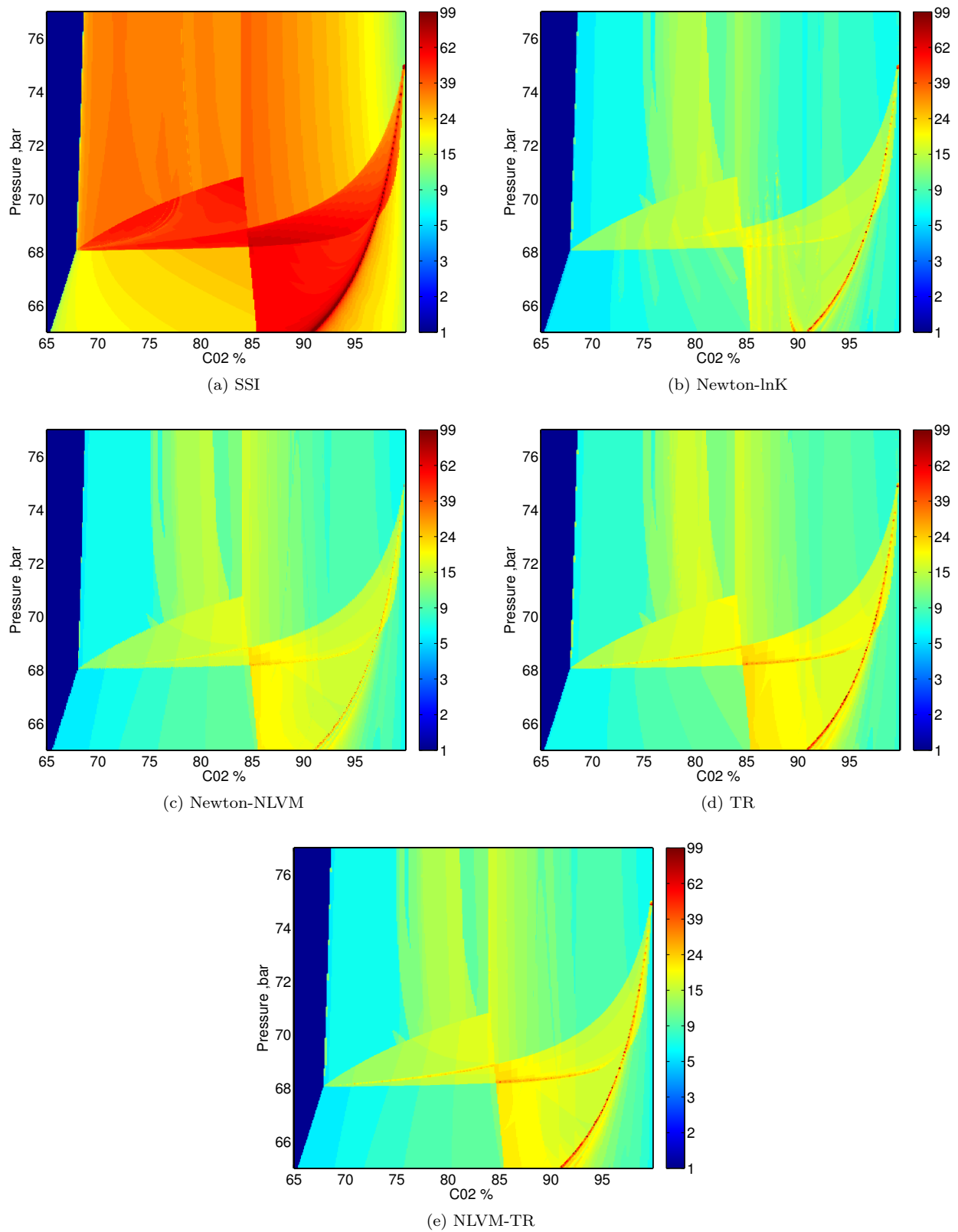


Figure 3.56: Number of iterations (two- and three-phase flashes) for Maljamar separator mixture over the P-z diagram at T=305.35 K

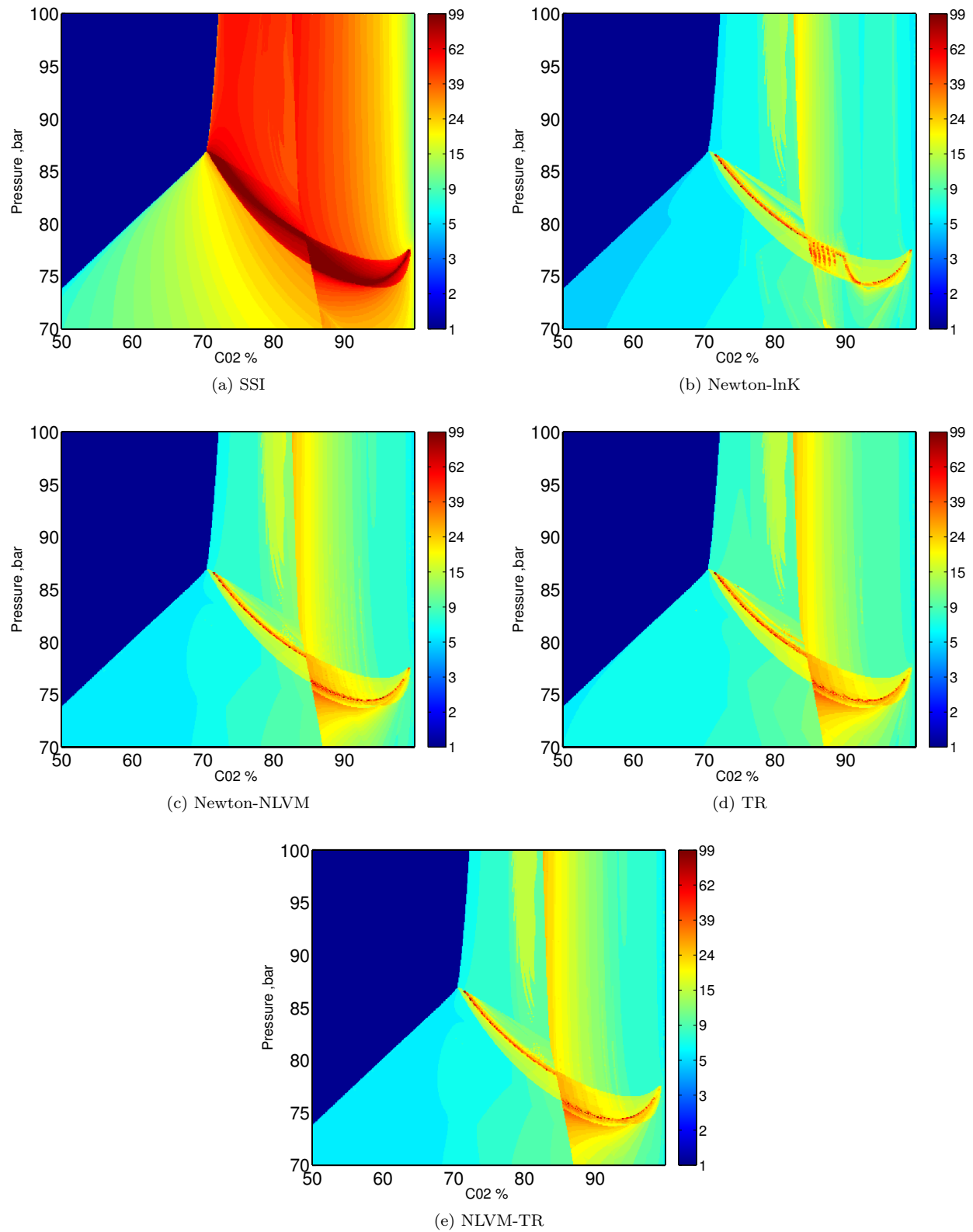


Figure 3.57: Number of iterations (two-and three-phase flashes) for the Oil B over the P-z diagram at T=307.6 K

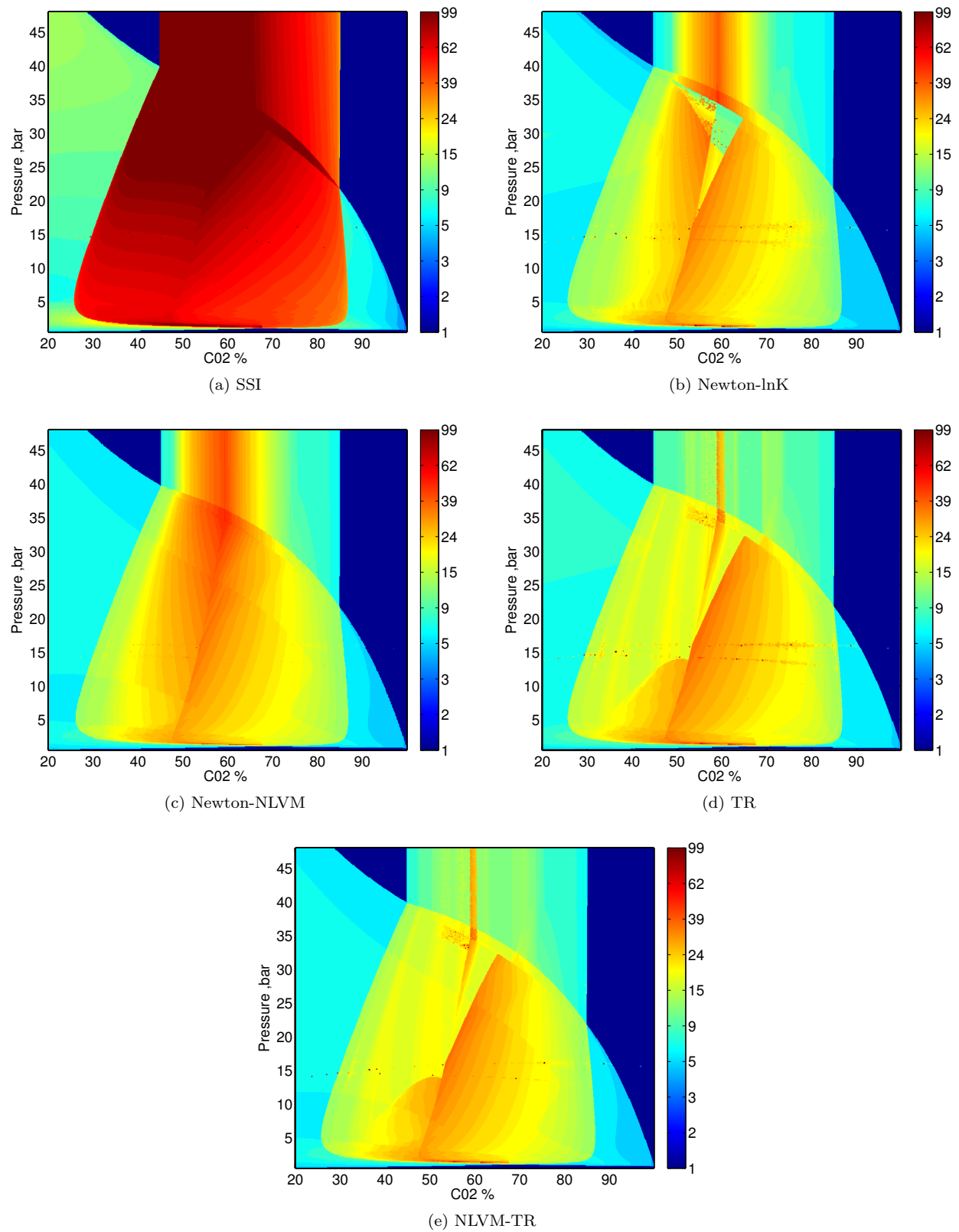


Figure 3.58: Number of iterations (two- and three-phase flashes) for the sour gas mixture over the P-z diagram at $T = 178.8$ K

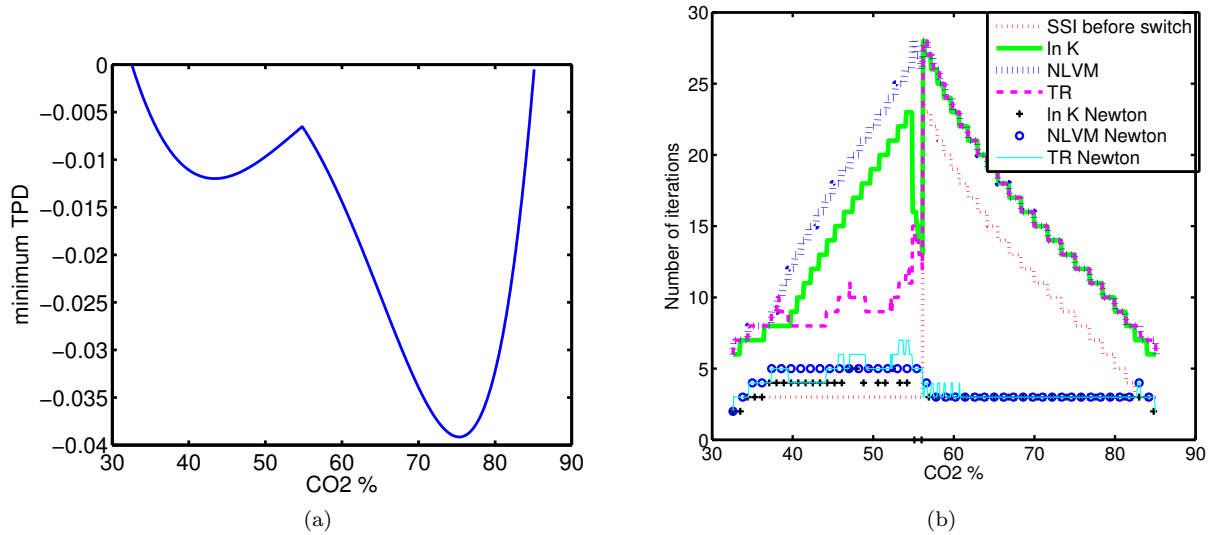


Figure 3.59: a) Global minimum of the modified TPD function b) Number of iterations of Newton and TR methods; for sour gas- CO_2 mixtures at $T=178.8$ K and $P=20$ bar

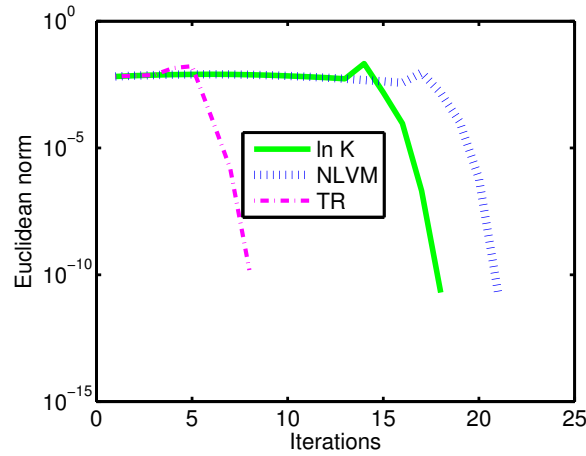


Figure 3.60: Error vs. iteration level for three-phase flash of the sour gas mixture at $T=178.8$ K, $P=20$ bar and 51% moles CO_2

The average number of iterations for two-phase liquid-vapor flash calculations with several calculation methods (SSI, **lnK**, **NLVM**, **TR**, and hybrid **NLVM – TR**) is listed in table 3.10 for four mixtures combined with CO_2 . The examined points are those used in constructing the P - z phase diagrams in fig. 3.53a to fig. 3.58e. Table 3.11 gives the average number of iterations for three-phase flash calculations (including the liquid-liquid phase split case). The CPU time (in seconds) required by various methods for flash calculations (cumulated for two- and three-phase flash flashes) is given in table 3.12.

Method	Mixture			
	Maljamar Oil (Res)	Maljamar Oil (Sep)	Oil B	Sour gas
SSI	34.017	33.859	39.169	37.143
ln K	8.477	8.787	9.280	8.686
NLVM	9.961	9.854	12.151	9.865
TR	9.863	9.854	12.149	7.641
NLVM-TR	9.843	9.850	12.147	7.771

Table 3.10: Average number of iterations for four mixtures and various flash calculation methods over the P - z diagram (two-phase flashes)

Method	Mixture			
	Maljamar Oil (Res)	Maljamar Oil (Sep)	Oil B	Sour gas
SSI	51.031	26.284	82.191	76.102
ln K	6.847	5.518	9.814	13.760
NLVM	7.345	6.874	11.770	15.902
TR	7.305	6.874	8.585	12.914
NLVM-TR	7.307	6.841	8.462	13.113

Table 3.11: Average number of iterations for four mixtures and various flash calculation methods over the P-z diagram (three phase flashes)

Method	Mixture			
	Maljamar Oil (Res)	Maljamar Oil (Sep)	Oil B	Sour gas
SSI	21.070	48.834	20.146	87.025
ln K	13.314	29.280	9.832	52.187
NLVM	11.671	18.518	9.552	30.237
TR	14.126	19.897	11.103	23.523
NLVM-TR	14.135	19.882	11.014	24.530

Table 3.12: Computation time (in seconds) for four mixtures and various flash calculation methods over the P-z diagram, flash (two- and three-phase flash calculations)

In terms of the average numbers of iterations, **Newton** – **lnK** is the fastest for two-phase flashes, and for two mixtures out of four for three-phase flashes (for the remaining two mixtures, **TR** and **NLVM** – **TR** performs better). **TR** is systematically faster than **NLVM**, while **NLVM** – **TR** is faster than **NLVM** in most cases.

In terms of computational time, **NLVM** is the fastest method (globally speaking, because the reverse is true for difficult conditions, and it is less robust than the other methods) except for the sour gas mixture. The other methods require additional workload per iteration as compared to **NLVM**: **TR** for solving the Trust-Region subproblem and **lnK** for the resolution of the Rachford-Rice equations (minimization of the function Q given by (eq. 2.42) in the multiphase case). Both **TR** and **NLVM** – **TR** methods perform remarkably well for the sour gas, and computational times for the other three mixtures are only slightly greater than those required by the other methods. Small differences are observed between **TR** and **NLVM** – **TR**, without revealing a trend.

It should be noted that comparisons of CPU times is always a delicate matter; the performance of a method depends, on many factors, and not only on the algorithm itself. Using vectorization, extensive use of cache memory, optimization flags can greatly improve the performances. Also, the solver can have a huge impact (Cholesky decomposition / LU factorization). The reported times are given indicatively, reproducing the global behavior of a method. However, a further optimized code could make the **TR** method even faster.

One can notice (for all examined mixtures, see fig. 3.53a to fig. 3.58e) that, even though computationally more difficult (bad condition numbers), at conditions near the phase boundaries (VLL-VL or VLL-LL) the number of iterations is smaller than within the three-phase region. The reason is that high quality initial guesses are obtained in the vicinity of phase boundaries from stability testing (for the mole fractions in the incipient phase) and two-phase flash results; the quality of initial guesses from stability deteriorates as one moves away from a phase boundary. This feature was reported for two-phase flash calculations [Pan and Firoozabadi [2003], Nichita and Graciaa [2011]], for which Wilson K-values give better initialization than stability testing away from the phase boundaries.

3.4.4 Conclusions

In this section, a Trust-Region method has been developed and successfully tested for phase equilibrium problems: stability analysis and multiphase flash calculations. A hybrid approach is also proposed: instead of performing **TR** iterations immediately after the switch from the first-order SSI method, Newton iterations are first performed, and the **TR** method is applied only if the objective function value is increasing between two consecutive iterations.

The proposed Trust-Region and hybrid **Newton** – **TR** algorithms have been tested for on a variety of mixtures involving hydrocarbon components, carbon dioxide and hydrogen sulfide, and exhibiting complicated phase envelopes. The proposed methods compare favorably to the widely used SSI-Newton methods with various independent variables.

The **TR** algorithm is mainly decreasing the number of iterations in some parts of the $P - T$ or $P - z$ diagrams where the Newton methods do not perform well (some regions close to STLL for stability analysis, close to critical points, or for conditions in which a poor initial guess is obtained from stability for flash calculations). In these cases, repeated switch back from Newton methods to the SSI method may be required; using the **TR** approach, the second-order information is used to progress towards the solution at a supra-linear rate. The proposed hybrid **Newton** – **TR** algorithm performs generally even better than the **TR**. The more difficult a test point is, the more spectacular the algorithm acts from both efficiency and reliability perspectives.

The proposed algorithm for phase equilibrium calculations is applicable to an arbitrary number of equilibrium phases and it is not dependent on the thermodynamic model; for more complex EoS and/or mixing rules, if the expression of the analytical Hessian is not available, the Trust-Region method can be used with a numerically evaluated Hessian.

In this work, the **TR** method was implemented for phase equilibrium calculations for conventional variables (in the compositional space). The next step is the extension to reduction methods [Hendriks [1988]]. A recently proposed reduction method [Nichita and Graciaa [2011]], allows formulation of flash in the reduced variables space as the minimization of Gibbs free energy, thus it is suitable for a Trust-Region approach, unlike previous reduction methods [Michelsen et al. [2013b]].

3.5 Multiphase flash, using $\ln K$ and phase mole fractions as independent variables

3.5.1 Introduction

[Haugen et al. [2011]] proposed a multiphase flash approach (advocated also by [Li and Firoozabadi [2012]]) which updates both the equilibrium constants and the phase mole fractions at each iteration level. Unlike the **lnK** approach, there is no explicit resolution of the Rachford-Rice equations, which are solved implicitly during a Newton update. The resulting linear system is non-symmetric and is solved based on Gaussian elimination. In a previous chapter, we have shown how symmetry can be efficiently used for multiphase flashes using $\ln K$ as independent variables by using a Cholesky factorization. In this section, it is shown how symmetry can be taken into account in Haugen’s approach, a new algorithm is proposed in which algebraic transformations are used to recast Haugen’s in a way that enables the use of a Cholesky factorization.

3.5.2 New proposed method

If $\ln K$ and θ are considered as independent variables [Haugen et al. [2011]], the error equations are given by:

$$g_{ik} = \ln K_{ik} + \ln \varphi_{ik} - \ln \varphi_{iR} = 0; i = 1, nc; k = 1, np-1 \quad (3.230)$$

and

$$R_k = \sum_{i=1}^{nc} \frac{z_i(K_{ik} - 1)}{1 - \sum_{l=1}^{np-1} \theta_l(K_{il} - 1)} = 0; \quad k = 1, np-1 \quad (3.231)$$

The vector of independent variables is $\xi = (\ln \mathbf{K}_1^T, \dots, \ln \mathbf{K}_{np-1}^T, \boldsymbol{\theta}^T)^T$, with $\ln \mathbf{K}_i = (\ln K_{1,i}, \dots, \ln K_{nc,i})^T$. Note that we use here a different, more natural, ordering of variables than in [Haugen et al. [2011]].

The Newton iteration equation is

$$\mathbf{J} \Delta \xi = -\mathbf{f} \quad (3.232)$$

where \mathbf{J} is the Jacobian matrix and $\mathbf{f} = (\mathbf{g}_1^T, \dots, \mathbf{g}_{np-1}^T, R_1, \dots, R_{np-1})^T$, or

$$\begin{pmatrix} \frac{\partial \mathbf{g}_1}{\partial \ln \mathbf{K}_1} & \dots & \frac{\partial \mathbf{g}_1}{\partial \ln \mathbf{K}_{np-1}} & \bigg| & \frac{\partial \mathbf{g}_1}{\partial \theta_1} & \dots & \frac{\partial \mathbf{g}_1}{\partial \theta_{np-1}} \\ \vdots & \ddots & \vdots & \bigg| & \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{g}_{np-1}}{\partial \ln \mathbf{K}_1} & \dots & \frac{\partial \mathbf{g}_{np-1}}{\partial \ln \mathbf{K}_{np-1}} & \bigg| & \frac{\partial \mathbf{g}_{np-1}}{\partial \theta_1} & \dots & \frac{\partial \mathbf{g}_{np-1}}{\partial \theta_{np-1}} \\ \frac{\partial R_1}{\partial \ln \mathbf{K}_1} & \dots & \frac{\partial R_1}{\partial \ln \mathbf{K}_{np-1}} & \bigg| & \frac{\partial R_1}{\partial \theta_1} & \dots & \frac{\partial R_1}{\partial \theta_{np-1}} \\ \vdots & \ddots & \vdots & \bigg| & \vdots & \ddots & \vdots \\ \frac{\partial R_{np-1}}{\partial \ln \mathbf{K}_1} & \dots & \frac{\partial R_{np-1}}{\partial \ln \mathbf{K}_{np-1}} & \bigg| & \frac{\partial R_{np-1}}{\partial \theta_1} & \dots & \frac{\partial R_{np-1}}{\partial \theta_{np-1}} \end{pmatrix} \begin{pmatrix} \Delta \ln \mathbf{K}_1 \\ \vdots \\ \Delta \ln \mathbf{K}_{np-1} \\ \Delta \theta_1 \\ \vdots \\ \Delta \theta_{np-1} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{np-1} \\ R_1 \\ \vdots \\ R_{np-1} \end{pmatrix} \quad (3.233)$$

where

- $\frac{\partial \mathbf{g}_k}{\partial \ln \mathbf{K}_p}$ are $(nc \times nc)$ matrices.
- $\frac{\partial \mathbf{g}_k}{\partial \theta_p}$ are $(nc \times 1)$ vectors.
- $\frac{\partial R_k}{\partial \ln \mathbf{K}_p}$ are $(1 \times nc)$ vectors.
- $\frac{\partial R_k}{\partial \theta_p}$ are scalars.

The elements of the Jacobian matrix are:

$$\left(\frac{\partial g_{ik}}{\partial \ln K_{jp}} \right)_{\boldsymbol{\theta}} = \sum_{s=1}^{nc} \left(\frac{\partial g_{ik}}{\partial n_{sk}} \right)_{\boldsymbol{\theta}} \left(\frac{\partial n_{sk}}{\partial \ln K_{jp}} \right)_{\boldsymbol{\theta}} \quad (3.234)$$

$$\left(\frac{\partial g_{ik}}{\partial \theta_p} \right)_{\ln \mathbf{K}} = \sum_{s=1}^{nc} \left(\frac{\partial g_{ik}}{\partial n_{sk}} \right)_{\ln \mathbf{K}} \left(\frac{\partial n_{sk}}{\partial \theta_p} \right)_{\ln \mathbf{K}} \quad (3.235)$$

$\left(\frac{\partial R_k}{\partial \ln K_{jp}} \right)_{\boldsymbol{\theta}}$ and $\left(\frac{\partial R_k}{\partial \theta_p} \right)_{\ln \mathbf{K}}$, which are given by (eq. 3.46) and (eq. 3.43) respectively.

The Jacobian matrix is non-symmetric. Using the block structure of \mathbf{J} , the linear system can be written as:

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{C} & \mathbf{D} \end{pmatrix} \begin{pmatrix} \Delta \ln \mathbf{K} \\ \Delta \boldsymbol{\theta} \end{pmatrix} = - \begin{pmatrix} \mathbf{g} \\ \mathbf{R} \end{pmatrix} \quad (3.236)$$

or

$$\mathbf{A} \Delta \ln \mathbf{K} + \mathbf{B} \Delta \boldsymbol{\theta} = -\mathbf{g} \quad (3.237)$$

$$\mathbf{C}\Delta \ln \mathbf{K} + \mathbf{D}\Delta \boldsymbol{\theta} = -\mathbf{R} \quad (3.238)$$

where the dimensions of the matrices are

- $(np-1)nc \times (np-1)nc$ for $\mathbf{A} = \frac{\partial \mathbf{g}}{\partial \ln \mathbf{K}}$
- $(np-1)nc \times (np-1)$ for $\mathbf{B} = \frac{\partial \mathbf{g}}{\partial \boldsymbol{\theta}}$
- $(np-1) \times nc(np-1)$ for $\mathbf{C} = \frac{\partial \mathbf{R}}{\partial \ln \mathbf{K}}$
- $(np-1) \times (np-1)$ for $\mathbf{D} = \frac{\partial \mathbf{R}}{\partial \boldsymbol{\theta}}$

From (eq. 3.238)

$$\Delta \boldsymbol{\theta} = \mathbf{D}^{-1}(-\mathbf{R} - \mathbf{C}\Delta \ln \mathbf{K}) \quad (3.239)$$

introducing (eq. 3.239) into (eq. 3.238) gives

$$(\mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C})\Delta \ln \mathbf{K} = -\mathbf{g} + \mathbf{B}\mathbf{D}^{-1}\mathbf{R} \quad (3.240)$$

or

$$\mathbf{J}^* \Delta \ln \mathbf{K} = -\mathbf{g}^* \quad (3.241)$$

where $\mathbf{J}^* = \mathbf{A} + \mathbf{B}\mathbf{D}^{-1}\mathbf{C}$ is formally identical to \mathbf{J} , but has a different (lower) implicitness level.

Finally, note that one can take advantage of symmetry by expressing \mathbf{A} and \mathbf{B} matrices as

$$\mathbf{A} = \mathbf{I} + \boldsymbol{\Phi}\mathbf{M}^{(\theta)} \quad (3.242)$$

and

$$\mathbf{B} = \boldsymbol{\Phi}\mathbf{M}^{(K)} \quad (3.243)$$

with

$$\left(M_{ij}^{(\theta)}\right)_{kp} = \left(\frac{\partial n_{ik}}{\partial \ln K_{jp}}\right)_{\boldsymbol{\theta}} \quad (3.244)$$

and

$$\left(M_i^{(K)}\right)_{kp} = \left(\frac{\partial n_{ik}}{\partial \theta_p}\right)_{\mathbf{K}} \quad (3.245)$$

In which partial derivatives are given by (eq. 3.41) and (eq. 3.40).

In the proposed formulation of the $\ln K - \boldsymbol{\theta}$ multiphase flash, the Jacobian matrix is calculated exactly as in the $\ln K$ method (using symmetry in its construction), then a Cholesky decomposition is used for solving the linear system (eq. 3.237 and eq. 3.238) as described previously, and finally phase mole fractions $\boldsymbol{\theta}$ are updated using (eq. 3.239). In our formulation, the dimensionality of the linear system is not increased and full advantage of symmetry is taken, leading to a potentially faster Newton $\ln K - \boldsymbol{\theta}$ iteration.

3.5.3 Results

Three mixtures were used in this study: the Maljamar reservoir oil, the Maljamar separator oil and the sour gas mixture exposed in section 3.2.4.3.1. In this section, two different kinds of methods have been tested:

- **lnK**: only updates **lnK** at each iteration, θ are updated solving the Rachford-Rice equations.
- **lnK – θ** : updates both **lnK** and θ at each iteration. The Rachford-Rice equations need not to be solved.

For symmetric matrices, the linear system is solved based on the Cholesky factorization. The methods using symmetry in the resolution of the linear system are; **nl/nv**, **NMVM**, **lnk – Chol.** (see subsection 3.2.3.2.2) and the proposed method (denoted **lnK – Chol.**).

For non-symmetric matrices, the linear system is solved based on an LU factorization. The methods concerned are **lnK**, [Haugen et al. [2011]] (denoted here **lnK – θ**), and finally, the new method which makes the same update of **lnK** and θ , but takes into account the symmetry only in the construction of the Jacobian (denoted here **lnK – θ – sym – LU**).

3.5.3.1 Computational time

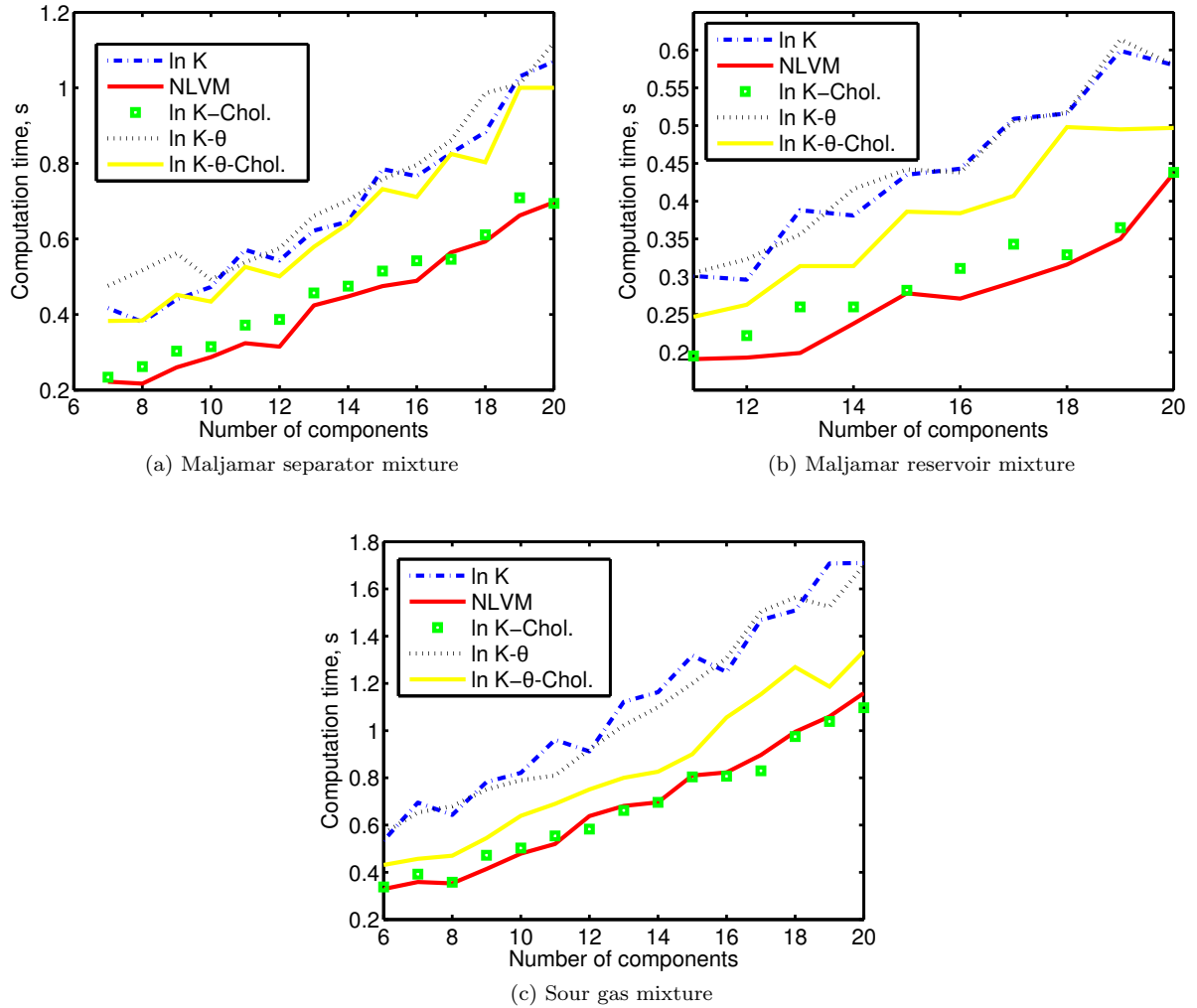


Figure 3.61: Computation time to perform Newton iterations through the whole three phase region

In this subsection, the exact same calculation procedure as in the subsection 3.2.4.3.1 in which flash calculations are performed for different P-z conditions covering the entire three phase region for each mixture. For each mixture, the same conditions as in the subsection 3.2.4.3.3.b were used to perform the calculations. Computing times are recorded between the switch to the Newton method and the convergence.

Computational times for each P-z condition are added to get the global computational time plotted in fig. 3.61a and fig. 3.61b for the Maljamar mixtures and in fig. 3.61c for the sour gas mixtures. In table 3.13,

the average number of iterations to perform calculations within the entire three phase region is given for $\ln K$, $\ln K - \text{Chol.}$, $\ln K - \theta$, $\ln K - \theta - \text{Chol.}$ and NLVM for the three mixtures.

3.5.3.2 Convergence behavior

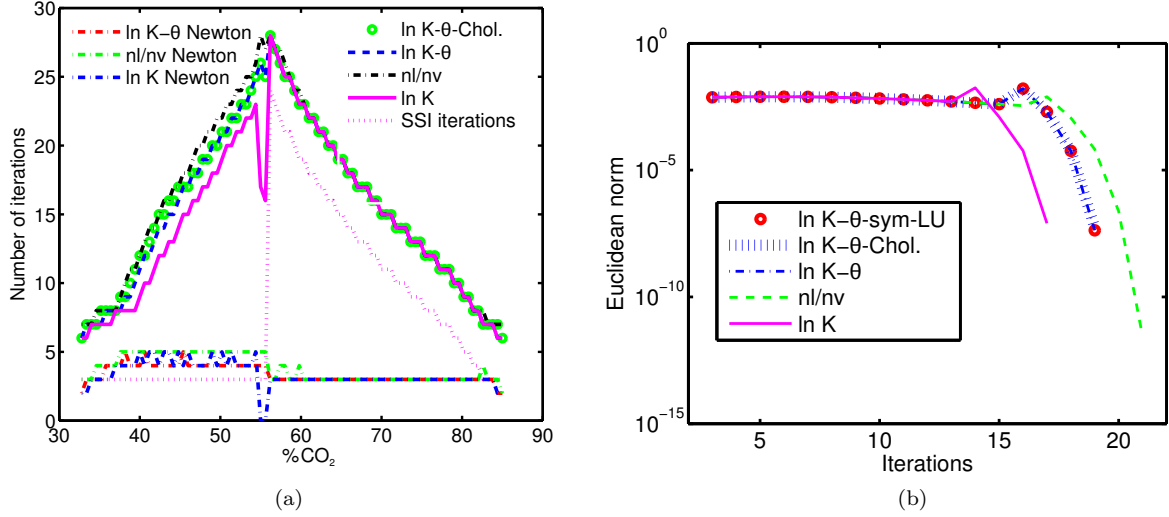


Figure 3.62: a) Convergence properties at 49.6% of CO_2 b) Number of iterations; for the sour gas mixture on the isobar $P=20$ bar

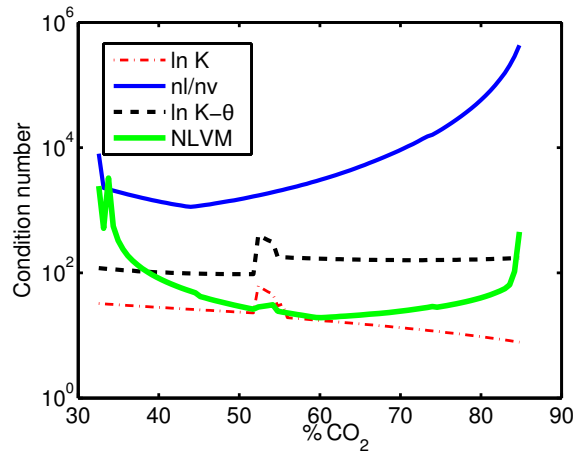


Figure 3.63: Condition number for various methods for the sour gas mixture on the isobar $P=20$ bar

The convergence properties of various Newton methods are compared. The three-phase flash calculations are initialized based on previous phase-split and stability analysis, than SSI iterations are performed as long as $S > 1e^{-2}$ before switching to Newton iterations. For the sour gas mixture at $T=305.35$ K, $P=20$ bar and 49.6% moles CO_2 , the Euclidean norm S (from eq. 3.98) is plotted in fig. 3.62a against the iteration number for various three-phase flash methods. fig. 3.62b shows the number of iterations required to reach convergence for the three-phase flash of the sour gas mixture at $P=20$ bar and compositions varying from 0.10 to 0.85, using different procedures (nl/nv , $\ln K$, $\ln K - \theta$, $\ln K - \theta - \text{sym} - \text{LU}$ and $\ln K - \theta - \text{Chol.}$). The condition numbers of the Jacobian/Hessian matrix for $\ln K$, $\ln K - \theta$, nl/nv and NLVM are plotted vs. composition at the same pressure and temperature conditions in fig. 3.63.

3.5.3.3 Discussion

The condition numbers for $\ln K$ and $\ln K - \theta$ (and to a less extent or $NLVM$) in fig. 3.63 show an anomaly between 53% and 57% moles of CO_2 . (note that at about 57% moles of CO_2 the global minimum of the tangent plane distance function exhibits an angular point as shown in [Petitfrere and Nichita [2014a]]). The reason is that for this composition window, the trial phase composition coming from stability analysis is different, inferring the choice of a different reference phase. The best conditioning of the linear system is for $\ln K$ and the worst, as expected, for nl/nv . With respect to nl/nv , the use of $NLVM$ greatly improves the conditioning, which is comparable to that of $\ln K$ in the middle of the three-phase region; however, the condition number for $NLVM$ deteriorates near phase boundaries. The condition number for $\ln K - \theta$ is greater than the one for $\ln K$ and $NLVM$ (for almost the entire interval), but, as compared to $NLVM$, $\ln K - \theta$ is far better conditioned near phase boundaries.

Taking advantage of the symmetry in solving the linear system leads systematically to a decrease of the computational time (from fig. 3.61a to fig. 3.61c), $\ln K - Chol.$ vs. $\ln K$ and $\ln K - \theta - Chol.$, $\ln K - \theta$; the effect is more important for $\ln K$ than for $\ln K - \theta$, and the differences seems to be more accentuated as the number of components in the mixture is increasing). $\ln K - Chol.$ is almost as fast (but more robust) as $NLVM$ and faster than the other methods. There is little or no gain in computational time if $\ln K - \theta$ is used instead of $\ln K$, and the condition number (fig. 3.63) is also in favor of $\ln K$;

In fig. 3.62a and fig. 3.62b the convergence behavior shows that $\ln K$ is the method converging within the smallest number of iterations. $\ln K - \theta$ and $\ln K - \theta - sym - LU$ have exactly the same convergence path. This has been shown theoretically and is confirmed numerically in fig. 3.62a and fig. 3.62b. Theoretically, $\ln K - \theta - Chol.$ should also have the exact same convergence path. However, fig. 3.62a and fig. 3.62b clearly reveal that sometimes one additional iteration is necessary for convergence; this comes from the approximation errors. In methods using a Cholesky decomposition, the matrix to inverse is the Hessian matrix, which has the worse condition number, see the nl/nv curve in fig. 3.63.

$NLVM$ is the fastest method, globally and per iteration (fig. 3.61a, fig. 3.61b and fig. 3.61c). However, more iterations are needed to reach convergence (fig. 3.62b) since the problem is really ill-conditioned close to phase boundaries, and needs much more iterations than $\ln K$ or $\ln K - \theta$ to converge. (it is less robust than any version based on $\ln K$).

(table 3.13) shows that the average number of iterations for $\ln K - \theta - Chol.$ is quite similar to the one for $\ln K$ for two mixtures. However, for the Maljamar separator mixture, the average number of iterations is much higher for $\ln K - \theta - Chol.$ than for the other variables (table 3.13). This indicates approximation errors. The time earned per Newton iteration is lost by the fact that more iterations are necessary to reach convergence. This explains the small improvement in fig. 3.61a when using $\ln K - \theta - Chol.$ as compared to $\ln K$ or $\ln K - \theta$.

Mixture	$\ln K$	$\ln K - Chol.$	$\ln K - V.$	$\ln K - V - Chol.$	$NLVM$
Sour gas	15.3883	15.4985	16.1958	16.3915	16.8825
Maljamar separator	6.3377	6.3377	6.4893	6.8177	6.7664
Maljamar reservoir	6.7249	6.7288	6.7537	6.9883	7.1972

Table 3.13: Average number of iterations within the whole three phase region

CHAPTER 4

Improvements in the characterization of heavy oils : Semi-continuous thermodynamics

Contents

4.1	Introduction	136
4.2	A new distribution function	138
4.3	Gaussian quadrature	140
4.4	Semi-continuous description	141
4.5	Calculation procedure	144
4.5.1	Discrete equilibrium flash calculations	144
4.5.2	Reconstruction of the initial solution	144
4.5.3	Algorithm	145
4.6	Results	145
4.6.1	Two-phase equilibrium	147
4.6.2	Three-phase equilibrium hydrocarbon mixture-water	149
4.6.3	Three-phase equilibrium hydrocarbon mixture-carbon dioxide	151
4.6.4	Influence of the number of quadrature points on accuracy	152
4.7	Conclusion	154

4.1 Introduction

Several mixtures for interest in industry, such as naturally occurring hydrocarbon mixtures (crude oils and gas condensates), feeds encountered in chemical processing, polymer solutions, etc., contain a very large number of components. Because: (i) a very large number of components can make prohibitive phase equilibrium calculations (as for instance in petroleum reservoir compositional simulation) and (ii) it is impossible to identify all components by standard chemical analysis (the heavier fractions are difficult to characterize), pseudo-components (grouping several individual components or fractions) are used to decrease the dimensionality of the phase equilibrium problem.

Usually, a mixture is lumped into pseudo-components using certain proximity criteria to select which components belong to a given pseudo-component [Montel and Gouel [1984], Newley and Merrill [1991], Lin et al. [2008]], then pseudo-component properties (critical pressure and temperature, acentric factor) and binary interaction parameters (BIPs) are assigned using appropriate averaging techniques. This approximation in fluid composition characterization may lead to significant errors in phase equilibrium calculations.

An elegant alternative to classical lumping procedures is the so-called continuous thermodynamics, which is based on the description of the composition of multicomponent mixtures by approximation with a continuous distribution function. In the semi-continuous thermodynamics, some of the individual components are treated discretely (usually light hydrocarbon components and non-hydrocarbon components: CO_2 , N_2 , H_2S , H_2O , etc.), while the remaining components (usually the heavy components) are included in the continuous part of the mixture. Continuous and semi-continuous thermodynamics were first developed by [Ratzsch and Kehlen [1983]] and by Cotterman and Prausnitz [1985], respectively. Soon thereafter in the 80's, several authors worked on developing continuous or semi-continuous thermodynamic phase equilibrium calculation procedures: [Cotterman et al. [1985], Behrens and Sandler [1986], Shibata et al. [1986], Willman and A.S. [1986], Willman and A.S. [1987a], Willman and A.S. [1987b], Ratzsch et al. [1988]]. Later on, semi-continuous thermodynamics has been applied to a variety of phase equilibrium problems: flash calculations at different specifications [Chou and J.M. [1986]], phase stability analysis using the Gibbs tangent plane distance [Browarzik et al. [1998], Monteagudo et al. [2001b]], critical point calculations [Rochocz et al. [1997]], compositional gradients [Lira-Galeana et al. [1994], Esposito et al. [2000]], liquid-solid equilibrium [Labadie and Luks [2003]], asphaltene precipitation [Monteagudo et al. [2001a]], vapor-liquid equilibrium using equations of state with group contributions [Baer et al. [1997]], etc.

The most encountered methods use the generalized Gauss-Laguerre quadrature to convert the continuous distribution of the mole (mass) concentration into a discrete one. More recently presented approaches are using specific techniques, such as a general Gauss-Stieltjes quadrature method able to calculate nodes and weights for any distribution function [Nichita et al. [2001]], orthonormal polynomials [Liu and Wong [1997]] or the quadrature method of moments [Lage [2007]].

However, most of the approaches presented in the literature are based on standard distributions. If the feed composition cannot be modeled by a standard distribution functions, or if it is simply highly irregular, most methods would not work properly. The semi-continuous approach based on the quadrature method of moments developed by [Lage [2007]] uses an optimal quadrature rule by taking the feed composition distribution as the weight function and it works with arbitrary compositions. However, in the original formulation of the QMoM, the quadrature is solved using Gordon's product-difference algorithm (PDA) [Gordon [1968]], which works well only for a small number of quadrature points. [John and Thein [2012]] compared the performance of the QMoM using PDA with the long quotient-modified difference algorithm (LQMDA) [Sack and Donovan [1972]] and the Golub-Welsch algorithm [Golub and Welsch [1969]]. They found that in certain situations the PDA failed (starting from eight quadrature nodes in their considered examples) whereas the two other examined methods were successful. [Gautschi [2004]] showed that the problem is ill conditioned, and the condition number grows exponentially with the number of quadrature points.

In this work, the QMoM is applied to multiphase equilibrium calculations for actual oil mixtures using a cubic equation of state with non-zero BIPs. The calculation of the quadrature is based on the procedure proposed by [Gautschi [1994]] (ORTHOPOL), which avoids problems due to the ill-conditioned nature of the problem and is suitable for an undetermined number of quadrature points (unlike the QMoM with the PDA). In some applications, it is important to use a number of pseudo-components larger than seven (which seems to be the limit of applicability of the PDA).

The chapter is structured as follows: first, a new distribution function is introduced; after a brief recall on Gaussian quadratures the semi-continuous description is presented. The general algorithm is given, then results for two- and three-phase flash calculations on a heavy oil mixed with carbon dioxide and water are presented before concluding. The calculation of the quadrature is detailed in two appendices.

4.2 A new distribution function

Usually, in semi-continuous thermodynamics, classical distribution functions are used to model the feed composition, such as the gamma [Cotterman and Prausnitz [1985], Shibata et al. [1986], Willman and A.S. [1986], Nichita et al. [2001], Whitson [1983]], beta [Park and Kim [1993]], exponential [Behrens and Sandler [1986], Nichita et al. [2001]] distributions, etc., or even multivariate distributions [Willman and A.S. [1987a], Willman and A.S. [1987b]]. In this work, a different way to define the distribution function and to use it within the Gaussian quadrature calculation is presented.

Let us consider the ratio of liquid mole numbers to feed composition

$$D_i = \frac{n_i^L}{z_i} = \frac{x_i L}{x_i L + y_i V} = \frac{1}{1 + \frac{V}{L} K_i} = \frac{1}{1 + \frac{V}{L} \exp(\ln K_i)} \quad (4.1)$$

Where n_i^L are the component mole numbers in the liquid phase, L and V are the liquid and vapor phase mole fractions, respectively, and $K_i = y_i/x_i$ are the equilibrium constants. From (eq. 4.2), D_i is a logistic function in terms of $\ln K_i$. For two-parameter cubic equations of state (EoS) and van der Waals mixing rules, the equilibrium constants can be written [Nichita and Minescu [2004]] as

$$\ln K_i = c_0 + c_b b_i + c_a \sqrt{a_i} + \sum_{k=1}^m c_k \gamma_{k,i} \quad (4.2)$$

Where m is the number of components having non-zero binary interaction parameters (BIPs) with the remaining components in the mixture, and the terms $\gamma_{k,i} = (1 - k_{ik})\sqrt{a_i}$; $i = 1, n$; $k = 1, m$ contain the non-zero BIPs. In (eq. 4.2), $\sqrt{a_i}$, b_i and $\gamma_{k,i}$ are the elements of the reduction matrix [Hendriks [1988]] and c_k are functions of the reduction parameters [Hendriks [1988]].

Equilibrium constants can also be expressed as a function of a characterization variable I (which can be taken as carbon number, molecular weight, true boiling point, or specific gravity; in this work the carbon number is used as characterization variable):

$$\ln K(I) = c_0 + c_b b(I) + c_a \sqrt{a(I)} + \sum_{k=1}^m c_k \gamma_k(I) \quad (4.3)$$

Where $\gamma_k(I) = (1 - k_{kI})\sqrt{a(I)}$

The function $K(I)$ is a strictly decreasing function for the C7+ fraction of a same family; the critical properties and the acentric factor are monotonic: $I > J \implies T_C(I) > T_C(J)$, $p_C(I) < p_C(J)$ and $\omega(I) < \omega(J)$. It is recommended also to use monotonic BIPs (that is, $k_{kI} < k_{kJ}$). Therefore, component' I will be more present in the liquid phase than the component' J .

In terms of I , (eq. 4.2), rewritten as

$$D(I) = \frac{n^L(I)}{z(I)} = \frac{x(I)L}{x(I)L + y(I)V} = \frac{1}{1 + \frac{V}{L} \exp(\ln K(I))} \quad (4.4)$$

defines a distribution function.

Let us call the distribution $D(I)$ (or (n^L/z) the liquid to feed mole ratio (LFMR)). The feed composition of a Canadian heavy oil (denoted Oil S) is plotted against the carbon number in fig. 4.1. Extended analysis of the C7+ fraction was available, and mixture composition is described by 100 components. To illustrate

the shape of the LFMR distribution function, the n_i^L/z_i ratio is plotted vs. carbon number in fig. 4.2a and fig. 4.2b for various pressures in the two-phase region within a large interval of temperatures (from 300 K to 800 K). In fig. 4.3a, the initial composition of the oil S has been manually altered to give a highly irregular feed composition; The LFMR function is plotted for the altered composition in fig. 4.3b, showing that the proposed function is smooth even though the feed composition is far from classical distributions or it could not even be approximated by a smooth function.

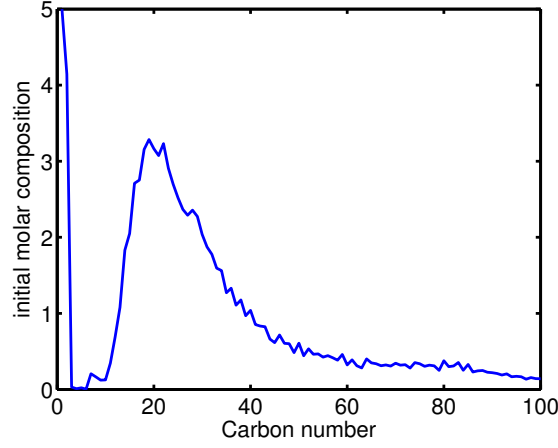


Figure 4.1: Feed composition vs. carbon numbers for the S. oil

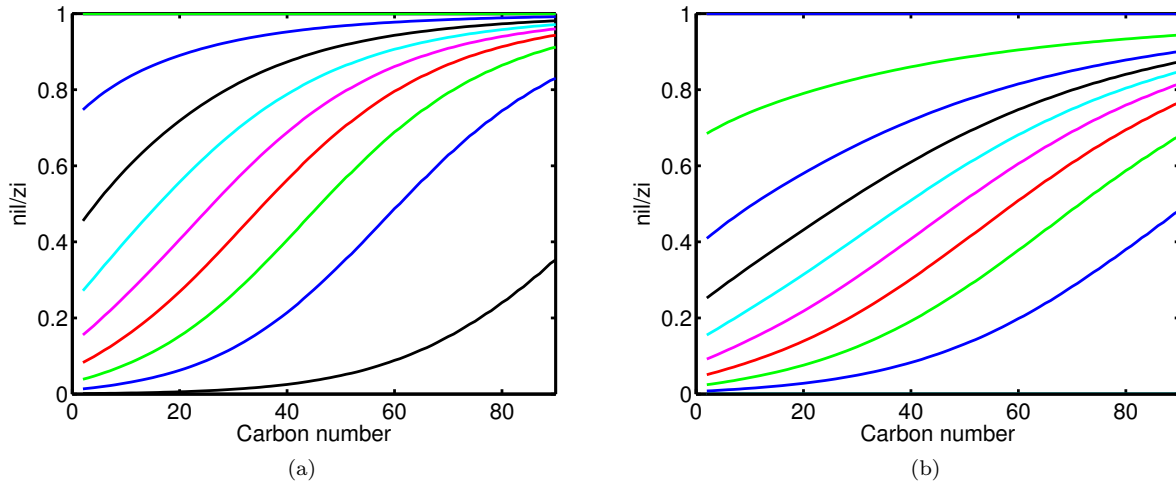


Figure 4.2: LFMR for different pressures and temperatures for the S. oil

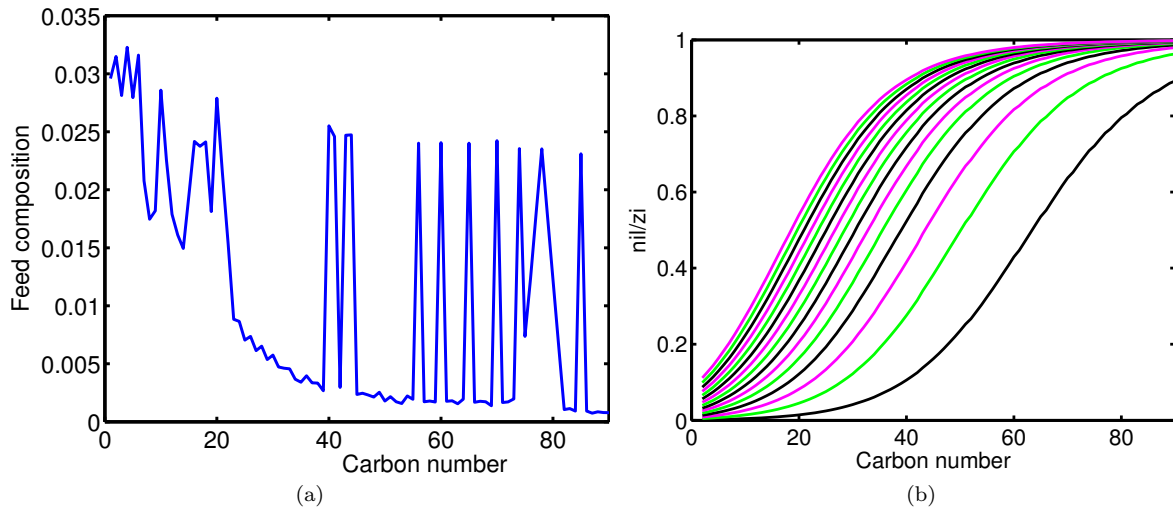


Figure 4.3: a) Altered feed composition (oil S) b) LFMR; vs carbon number for the altered oil S

4.3 Gaussian quadrature

A Gaussian quadrature approximates a definite integral by a discrete sum

$$\int_a^b W(x)f(x)dx \approx \sum_{i=0}^n w_i f(x_i) \quad (4.5)$$

Where n is the number of quadrature points, $f \in L_2([ab])$, $W \in L_2([ab])$, $x_i \in \mathbb{R}$, $\forall i = 1, n, w_i \in \mathbb{R}$, $\forall i = 1, n$.

The construction of the Gaussian quadrature requires the computation of quadrature nodes and weights. Let p_n be a nontrivial polynomial of degree n such that

$$\int_a^b W(x)x^k p_n(x)dx = 0 \quad (\forall) k = 0, 1, \dots, n-1 \quad (4.6)$$

If $x_i \in]a, b[$ are the n roots of the polynomial $p_n(x)$, then there exist n weights w_i which make exact the integral calculated by the Gaussian quadrature for all polynomials $q(x)$ of degree $2n-1$ or less. The polynomial $p_n(x)$ is said to be an orthogonal polynomial of degree n associated to the weight function $W(x)$. This polynomial is unique and its coefficients must be calculated to solve the quadrature.

The ordinary moments for a given weight function are given by

$$\mu_r = \int_a^b x^r W(x)dx, r = 0, 1, \dots, 2n-1 \quad (4.7)$$

and the modified moments [Wheeler [1974]] are

$$v_n = \int_a^b p_n(x)W(x)dx \quad (4.8)$$

In the quadrature method of moments, all the moments of the weight function are calculated (ordinary moments for Gordon's PDA [Gordon [1968] and Golub and Welsch [1969]] algorithm, and modified moments for the LQMDA [Sack and Donovan [1972]] and Wheeler's algorithm [Wheeler [1974]]). Once all the moments are computed, the computation of the inner products (see Appendix K) becomes straightforward.

[John and Thein [2012]] showed that the PDA is well suited only up to a certain number of nodes (up to seven in their examples), and is less robust than the Golub-Welsch algorithm and the LQMDA. Besides the fact that the QMoM with PDA [Gordon [1968]] leads to an ill-conditioned problem when increasing the number of quadrature points ([Gautschi [2004]] proved that the condition number grows exponentially with the number of quadrature points), they showed that in certain cases failures are due to the vanishing of a moment. [John and Thein [2012]] recommended the use of LQMDA or one of its variants (for instance Wheeler's algorithm [Wheeler [1974]]) with the QMoM.

In this work, the calculation of the quadrature is based on the procedure proposed by Gautschi (ORTHOPOL) [Gautschi [1994]], which avoids problems due to the ill-conditioned nature of the problem. The calculation of the unique orthogonal polynomial associated to the weight function is presented in Appendix K and the calculation of quadrature nodes and weights is given in Appendix L. This represents the core routine of the QMoM. The complexity, as well as the difference between different methods consists in computing the inner products for α_n and β_n (the coefficients of the recursive formula for orthogonal polynomials), in

equations (eq. K.2) and (eq. K.3).

4.4 Semi-continuous description

If the composition is discrete, the complementarity relation of feed mole fractions z_i can be written by separating the components which remain discrete in the semi-continuous description from the components belonging to the continuous part of the mixture

$$\sum_{i=1}^{n_D} z_i^D + \sum_{i=1}^{n_c} z_i^C = 1 \quad (4.9)$$

For the semi-continuous description, (eq. 4.9) writes as

$$\sum_{i=1}^{n_D} z_i^D + \int_{I_{\min}}^{I_{\max}} z^C(I) dI = 1 \quad (4.10)$$

with a continuous integral replacing the second summation, where I_{\min} and I_{\max} are specified and are related to the minimum and maximum carbon number in the continuous part of the mixture.

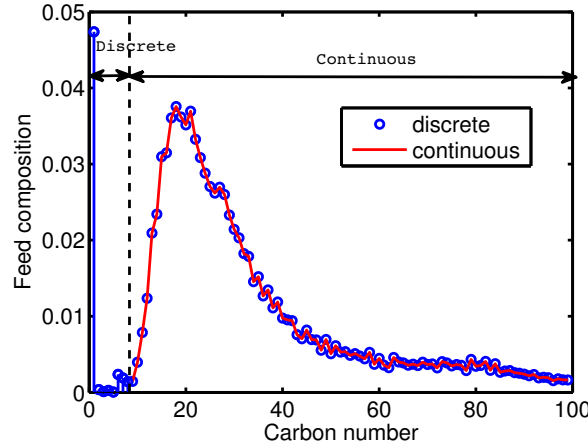


Figure 4.4: Feed composition of the semi-continuous mixture (Oil S) after cubic spline interpolation on the continuous part

From a discrete feed composition, a cubic spline interpolation can be used to create the continuous fractions. For the Oil S, there are 6 discrete components (from methane to hexanes fraction; note that normal and iso intermediary components were grouped together) and 94 components in the continuous part of the mixture. Feed composition of the semi-continuous mixture (Oil S) is represented in fig. 4.4, showing the discrete components (the light ones) and the continuous portion of the mixture as obtained after a cubic spline interpolation. The generation of the continuous part creates a shift in the feed composition (their sum will not be equal to one anymore), which has to be rescaled as

$$z^C(I) = \frac{z^C(I)}{X} \quad (4.11)$$

with

$$X = \frac{\int_{I_{\min}}^{I_{\max}} z^C(I) dI}{1 - \sum_{i=1}^{n_D} z_i^D} \quad (4.12)$$

Let us consider that the liquid phase composition is described by an arbitrary distribution (not necessarily the same as for the feed composition). Similar to (eq. 4.10), one can write for the liquid phase

$$\sum_{i=1}^{n_D} x_i^D + \int_{I_{\min}}^{I_{\max}} x^C(I) dI = 1 \quad (4.13)$$

or, in terms of liquid mole numbers

$$\sum_{i=1}^{n_D} n_{iL} + \int_{I_{\min}}^{I_{\max}} n_L(I) dI = L \quad (4.14)$$

By multiplying and dividing the integrand with $z^C(I)$, (eq. 4.14) writes

$$\sum_{i=1}^{n_D} n_{iL} + \int_{I_{\min}}^{I_{\max}} z^C(I) \frac{n_L(I)}{z^C(I)} dI = L \quad (4.15)$$

and one can recognize in the above equation the distribution LFMR, $D(I) = n_L(I)/z^C(I)$, introduced in the section 2 of this chapter (and plotted for various temperature-pressure conditions in fig. 4.2a and fig. 4.2b for the Oil S).

The Gaussian quadrature will be quite accurate as long as the function $f(x)$ can be well approximated by a polynomial function. It has been seen that n_L/z has always the same shape, it is smooth, and thus can be well approximated by polynomial functions. Polynomial regression tests showed that the error is smaller than 10^{-3} for polynomials with order greater than six. In addition, unlike in a classical Gaussian quadrature (in which the weight function is given), the procedure of [Gautschi [1994]] allows arbitrary weight functions $W(x)$. Therefore, the feed composition can be used as the weight function.

The integral in (eq. 4.5) can be approximated using

$$W(I) = z^C(I) \quad (4.16)$$

And

$$f(I) \equiv D(I) = \frac{n_L(I)}{z^C(I)} \quad (4.17)$$

giving

$$\int_{I_{\min}}^{I_{\max}} z^C(I) \frac{n_L(I)}{z^C(I)} dI = \sum_{j=1}^{n_C} z_j \frac{n_{jL}(x_j)}{z_j} \quad (4.18)$$

Or

$$\int_{I_{\min}}^{I_{\max}} W(I) f(I) dI = \sum_{j=1}^{n_C} w_j f(x_j) \quad (4.19)$$

It is very important to note that f is not used within the Gaussian quadrature calculation. The distribution $n_L(I)/z^C(I)$ is not known explicitly before performing a flash calculation; however, it is known that the shape of this distribution function is always the one of a logistic function and it is well approximated by a polynomial; this is the only requirement on the function f , in order to get an accurate approximation. Moreover, one can take $f = 1$; in this case (eq. 4.19) simply reads

$$\int_{I_{\min}}^{I_{\max}} z^C(I) dI = \sum_{j=1}^{n_C} z_j \quad (4.20)$$

The key in Lage's QMoM is to take the feed composition as a weight function. In fact, Lage's methodology might be called the feed-weight quadrature method (FWQM) rather than QMoM (since the quadrature can be solved also by methods not involving moments directly, see the previous section). In this work, we have arrived at the same choice in a somewhat different way, by justifying the choice of $w_i = z_i$ based on some physics, that is, on the newly introduced LFMR distribution. This choice of the weight function has been made precisely to obtain the LFMR function (which is smooth for any temperatures and pressure conditions for a given feed). It is shown above that the given weight function allows to satisfy the mass balance for the feed, provided $f = 1$ (the quadrature in (eq. 4.19) is exact for any polynomial function of order less than $2n - 1$). This ensures to compute pseudo-components which will give a really good representation of the whole continuous fluid and give accurate results for an equilibrium flash calculation.

By solving the above quadrature problem (eq. 4.19), the abscissas (nodes) and the discrete weights that approximate the continuous integral are obtained. The nodes correspond to the carbon numbers (or any other characterization variables, as for instance true boiling points) where the quadrature is applied, and weights give the mole fractions of the pseudo-components. It means that instead of performing the flash on the semi-continuous mixture, the quadrature calculates N carbon numbers, corresponding to N pseudo-components, and phase equilibrium calculations can be carried out as for a discrete mixture for accurately reproduce the fluid behavior.

The procedure presented above can be readily extended if different distributions are used for each component family or different portions of the continuous part are modeled by different distribution functions, (eq. 4.9) in the discrete case can be written as

$$\sum_{i=1}^{n_D} z_i^D + \sum_{k=1}^{n_{distr}} \sum_{j=1}^{n_C(k)} z_{kj}^C = 1 \quad (4.21)$$

and (eq. 4.10) in the semi-continuous case becomes

$$\sum_{i=1}^{n_D} z_i^D + \sum_{k=1}^{n_{distr}} \int_{I_{\min}(k)}^{I_{\max}(k)} z_k^C(I) dI = 1 \quad (4.22)$$

Where n_{distr} is the number of distributions used.

The method the most commonly found in the literature is to approximate the feed composition with a standard distribution and to solve a classical Gaussian quadrature. Generally, the parameters of the distribution giving the best fit of the feed composition have to be calculated, and then the quadrature can be applied to the approximated distribution. However, the feed composition is not always regular and can have a shape far away from any standard distribution. In such cases, this methodology cannot be applied accurately. In the QMoM, the weight function is arbitrary, and for each calculated pseudo-component, the discrete weight function corresponds to its feed composition. Moreover, the function to which the quadrature is applied is always regular and smooth, guaranteeing an efficient and adapted computation of the pseudo-components for every feed composition.

In our implementation, the code was designed to incorporate two options:

- Discrete weights (feed composition) are directly used in the quadrature, as in Lage's QMoM;
- Interpolated weight functions are used (for instance by cubic splines as mentioned above).

As it will be seen later in the results section, the first option gives more accurate results; the second option may be useful if the problem is reformulated in terms of a different characterization variable (say true

boiling points from a distillation curve).

4.5 Calculation procedure

4.5.1 Discrete equilibrium flash calculations

After carrying out the quadrature procedure with the arbitrary weights z_k^C , the abscissas x_j (the nodes of the quadrature) are found. These nodes give the carbon numbers corresponding to the N 'pseudo-components' in the semi-continuous description of the mixture. The properties required in traditional flash calculations are then calculated for these carbon numbers, and any flash calculation method can be used as if the system is discrete.

Pseudo-component properties (critical properties and acentric factor, as well as BIPs between discrete components and pseudo-components) can be calculated based on any correlation from the literature (in the present paper, an in-house correlation has been used). The only requirement is to respect the monotonicity of the critical properties, which is the key point for getting the logistic function. Cubic splines are used for critical properties, acentric factors and BIPs to create continuous functions of carbon numbers. Then, the properties are evaluated at each carbon number I to obtain the properties and BIPs for the 'lumped' system.

A classical equilibrium flash calculation is performed on the discrete composition in which the pseudo-components corresponding to the continuous part are added. Since pseudo-component properties can be written as a function of the carbon number, any thermodynamic model (even complex equations of state with various mixing rules) can be used, as long as the monotonic behavior of $K(I)$ is observed (as in (eq. 4.3) for cubic equations of state).

In two-phase flash calculations, successive substitution iterations followed by second-order Newton iterations are used, starting from Wilson's K-values for initialization. In three-phase flash calculations, the results of phase stability testing and two-phase flash calculations are used for initialization.

4.5.2 Reconstruction of the initial solution

At the solution of flash calculations on the lumped mixture, it may be needed to calculate the composition of the full mixture, and not only on the lumped one.

Two methodologies were tested to recover the full composition from the lumped composition:

- The cubic spline interpolation over the $\ln K$ vs. carbon number distribution. The distribution is a really smooth distribution as well, and could be very well interpolated by polynomial functions.
- The delumping (inverse lumping) procedure, as developed by [Nichita and Leibovici [2006] and Nichita et al. [2006b]] for multiphase systems.

It was observed that when all BIPs are set to zero, the first approach acts better. Conversely, when non-zero BIPs are used in the cubic EoS, the second method is clearly superior. Thus, the delumping method is the one used in the paper, and the one advised.

Let us suppose that $m < n$ components in the mixture have non-zero BIPs with all the other components, that is, $k_{kj} = k_{jk} \neq 0$ for $I, j = k + 1, n$ and $k_{ji} = k_{ij} = 0$ for $i, j = m + 1, n$. Usually $m \ll n$ (BIPs are non-zero for non-hydrocarbon - hydrocarbon pairs, and between light and heavy hydrocarbon components; it is considered that there are no interactions between components belonging to the continuous portion of the

mixture, thus BIPs between heavy components are all set to zero).

The key equation for delumping is (eq. 4.2), relating through a multilinear expression the equilibrium ratios of the detailed mixture to some bulk properties of the lumped mixture (via the coefficients c_k) and to some constant properties of the components in the detailed mixture. The delumping procedure consists in the following steps:

- An equilibrium flash calculation is performed on the semi-continuous ('lumped') mixture
- The c_k constants (from eq. 4.2) for the lumped mixture are calculated at the solution.
- $\sqrt{a_i}$, b_i , γ_{ki} are calculated for the full discrete mixture
- $\ln K_i$ are then calculated using (eq. 4.2)
- Calculation of phase distribution (Rachford-Rice equation) and compositions

In the multiphase case, the delumping procedure of [Nichita et al. [2006b]] extends the above procedure, the phase distribution is calculated by the minimization of a convex function rather than by solving a system of non-linear equations [Michelsen [1994]].

Using a delumping procedure, besides recovering some information on the detailed mixture, allows a better comparison between discrete and semi-continuous descriptions, not only on phase distribution, but also on phase compositions.

4.5.3 Algorithm

Summarizing, the calculation algorithm for flash calculations using the semi-continuous description of hydrocarbon mixtures is as follows:

1. Assign weights as feed mole fractions (either discrete or interpolated by using cubic splines)
2. Solve the quadrature
 - (a) Calculate the coefficients of the orthogonal polynomial
 - (b) Find the roots of the orthogonal polynomial (nodes of the quadrature)
 - (c) Calculate the weights of the quadrature
3. Assign carbon numbers and composition to the pseudo-components
4. Calculate the pseudo-component properties and the BIPs
5. Perform a flash calculation on the lumped mixture
6. Optional, apply the delumping procedure to recover phase composition of the detailed feed and improved phase mole fractions.

4.6 Results

The accuracy of flash calculations using the semi-continuous description is tested for three different mixtures:

- The recombined S oil (real mixture), denoted hereby mixture A (feed composition in fig. 4.1);
- Two alterations of the mixture A : mixtures B and C (feed compositions vs. carbon number are given in fig. 4.5a and fig. 4.5b, respectively).

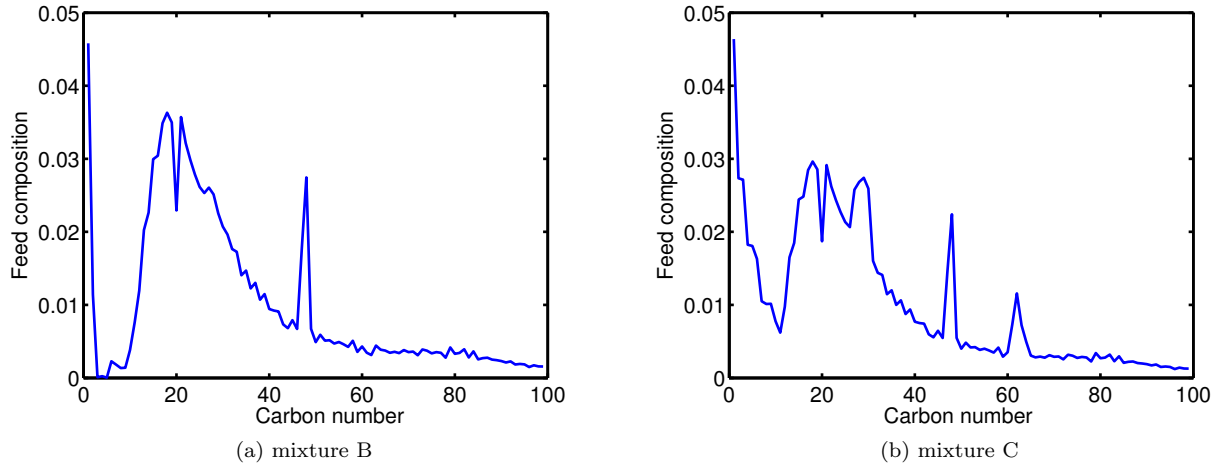


Figure 4.5: Altered of the S oil (mixture A)

The shape of the composition of Mixture A vs. carbon number suggests that it could be approximated by a gamma distribution function. The approximation would possibly give a small error. However, altered mixtures B and C have been created to show that the algorithm can work even with an initial composition far from known distributions.

Three tests were performed: (i) Two-phase equilibrium flash calculations at $T=450$ K; (ii) Three-phase equilibrium flash calculations with a small amount of water (5.4% moles) at $T=310$ K; and (iii) Three-phase equilibrium flash calculations with a high amount of CO_2 (90% moles) at $T=249$ K. The Peng-Robinson EoS [Peng and Robinson [1976]] is used in all calculations. Non-zero BIPs between C_1 and hydrocarbon components are assigned according to [Firoozabadi et al. [1978]]. BIPs for CO_2 -hydrocarbon pairs are set to 0.12, and for the H_2O - hydrocarbon pairs all BIPs are set to 0.5. It should be noted that the QMoM was tested previously using cubic EoS with zero BIPs [Rodrigues et al. [2012]]. In this work, the QMoM is tested using large non-zero values of the BIPs; using non-zero BIPs is important in most practical applications when equations of state are used.

The full discrete flash has been performed with 100 components (in the results, we call the solution discrete). The semi-continuous flash was performed with 5 quadrature points, and 6 discrete components, giving a total of 11 components for two-phase flash calculations and for three-phase flashes for H_2O -hydrocarbon mixtures. For CO_2 -hydrocarbon mixtures, 6 quadrature points were used, giving a total of 14 components.

The errors between different representations of the mixture feed composition are given as root mean square error (RMSE) and relative root mean square error (RRMSE). For the phase mole fractions:

$$RMSE = \sqrt{\frac{1}{n_P} \sum_{i=1}^{n_P} \left(\beta_i^{(sc)} - \beta_i^{(df)} \right)^2} \quad (4.23)$$

And

$$RRMSE = \sqrt{\frac{1}{n_P} \sum_{i=1}^{n_P} \left(\frac{\beta_i^{(sc)} - \beta_i^{(df)}}{\beta_i^{(df)}} \right)^2} \quad (4.24)$$

and for the phase compositions

$$RMSE_i = \sqrt{\frac{1}{n_P} \sum_{j=1}^{n_P} \left(x_{ij}^{(del)} - x_{ij}^{(df)} \right)^2}; i = 1, n \quad (4.25)$$

and

$$RRMSE_i = \sqrt{\frac{1}{n_P} \sum_{j=1}^{n_P} \left(\frac{x_{ij}^{(del)} - x_{ij}^{(df)}}{x_{ij}^{(df)}} \right)^2}; i = 1, n \quad (4.26)$$

Where n_p is the number of phases in equilibrium, β_j is the mole fraction of the phase j , x_{ij} is the mole fraction of the component i in the phase j , and superscripts (sc), (df) and (del) denote the semi-continuous, discrete and delumped feeds, respectively. Calculations were performed using the two options for weight functions (using the discrete feed compositions as weight function or using an interpolated weight function). The results are presented only for the first option in fig. 4.6 to fig. 4.17b, because, as can be seen from fig. 4.18a to fig. 4.20b, it gives more accurate results (usually errors are at least one order of magnitude lower).

4.6.1 Two-phase equilibrium

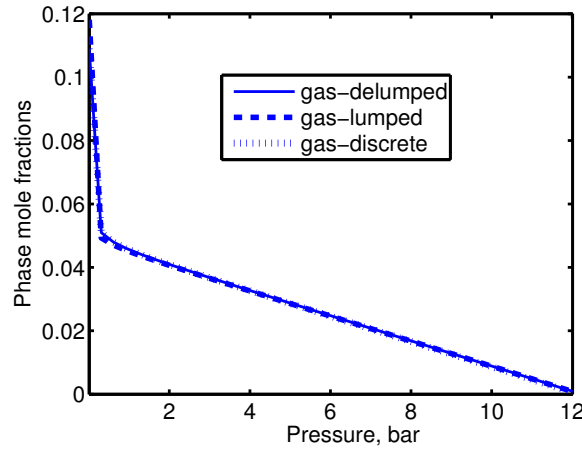


Figure 4.6: Mole fraction in the vapor phase for mixture A at T=450 K

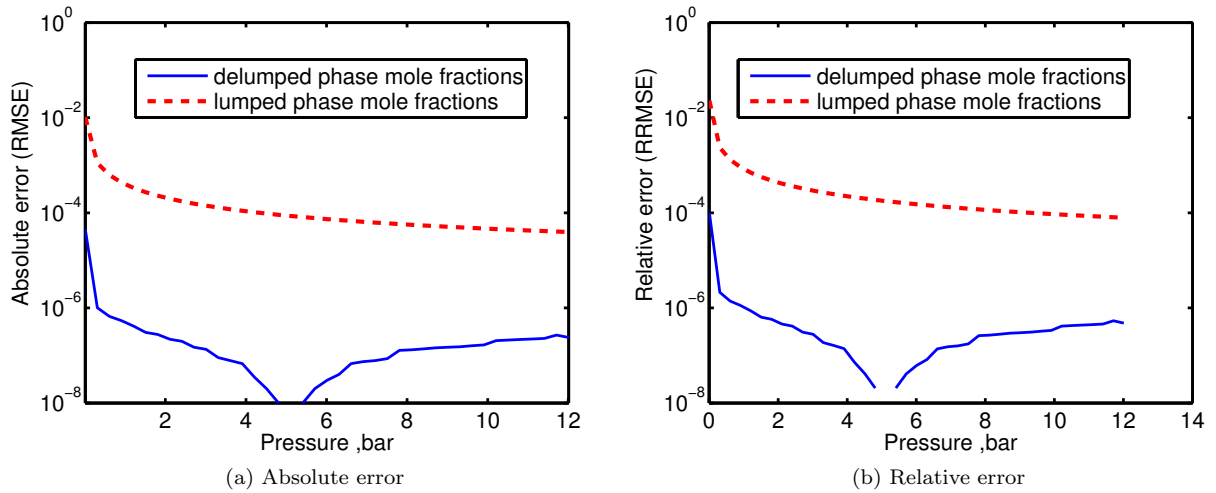


Figure 4.7: Error in phase mole fractions for mixture A at T=450 K

Two-phase equilibrium flash calculations were performed at $T=450$ K for the three mixtures. Mixture A (oil S) is a heavy oil, thus the bubble point pressure is low (12 bar at $T=450$ K). The mole fractions in the vapor phase for mixture A are plotted against pressure at $T=450$ K in fig. 4.6, for the full discrete, semi-continuous (denoted here 'lumped') and delumped representations of the feed composition. An excellent agreement is observed, the absolute (eq. 4.23, see fig. 4.7a) and relative (eq. 4.24, see fig. 4.7b) errors in phase mole fractions of lumped and delumped results with respect to the discrete results being of the order of 10^{-4} or less on the entire pressure interval, except at low pressures.

The absolute errors (eq. 4.25) and relative errors (eq. 4.26) in component mole fractions (between detailed and delumped feeds) at $T=450$ K and $P=8$ bar are plotted against the carbon number in fig. 4.8a and fig. 4.8b, respectively. The errors are very small for the continuous part of the mixture (RMSE of the order of 10^{-6} or less, RRMSE of the order of 10^{-4}). Higher deviations are observed for components in the discrete part of the semi-continuous mixture. Plots of errors against the carbon number at $T=450$ K and $P=8$ bar are given for mixture B in fig. 4.9a (RMSE) and fig. 4.9b (RRMSE), and for mixture C in fig. 4.10a (RMSE) and fig. 4.10b (RRMSE). It can be seen from fig. 4.8a to fig. 4.10b that the initial composition affects the accuracy of the results. The errors in composition are increasing with the irregularities (from mixture A to mixture C), which is normal because the number of pseudo-components (quadrature points) is too small to model all the irregularities. However, absolute errors are still very small and relative errors reasonably small for the highly irregular feed composition of mixture C.

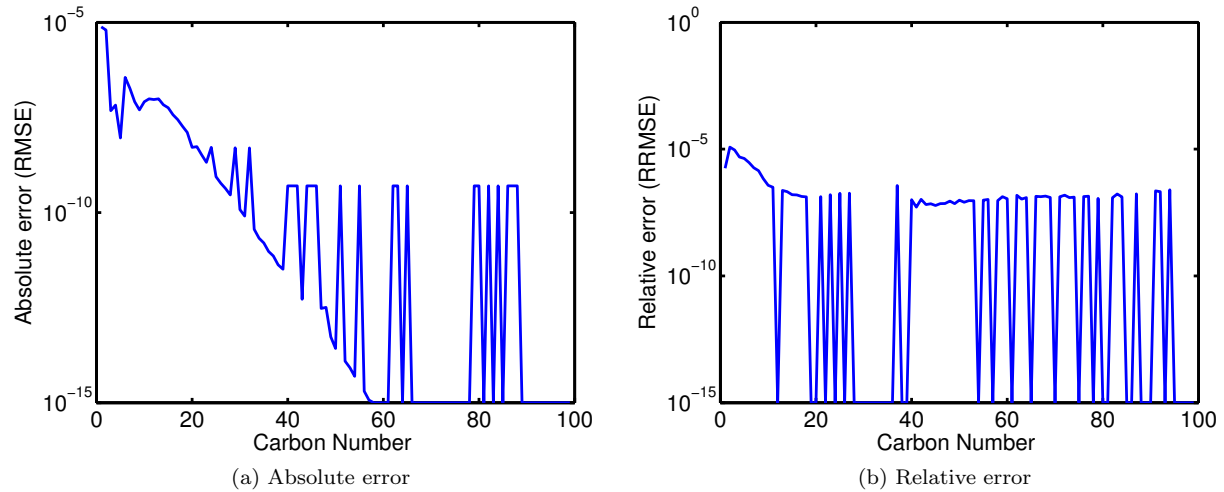


Figure 4.8: Error in phase compositions for mixture A at $T=450$ K and $P=8$ bar

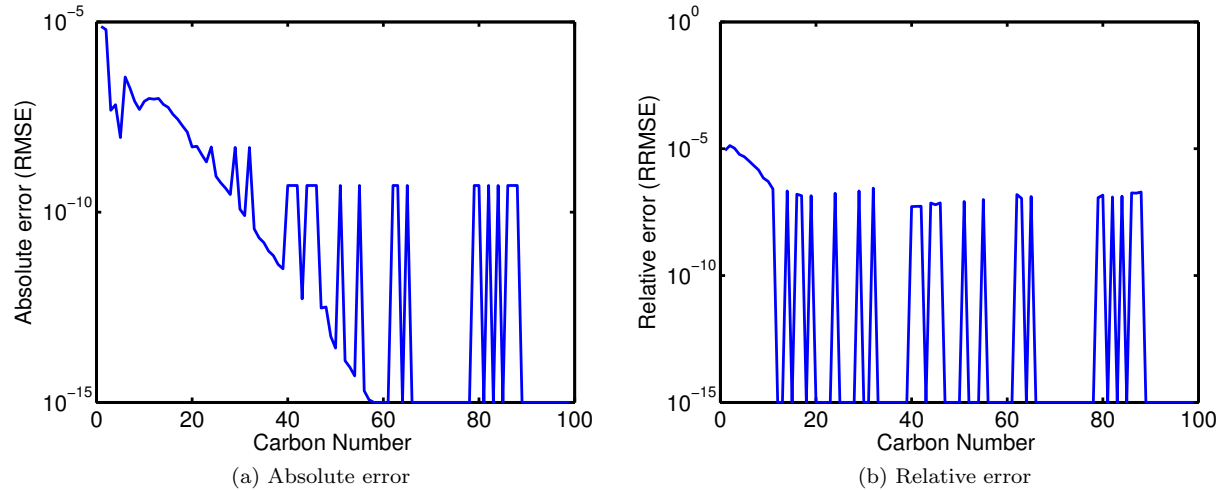


Figure 4.9: Error in phase compositions for mixture B at $T=450$ K and $P=8$ bar

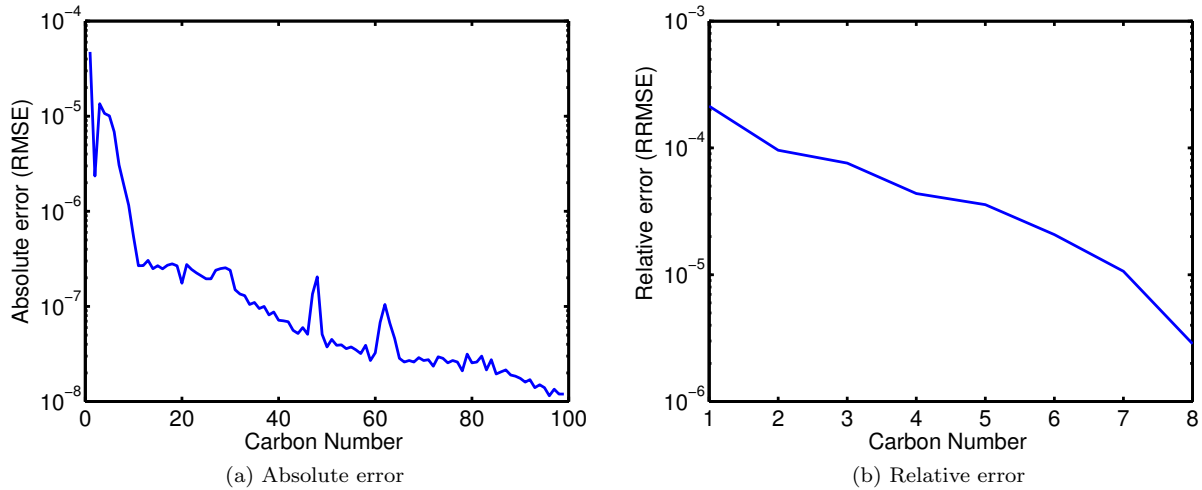


Figure 4.10: Error in phase compositions for mixture C at $T=450$ K and $P=8$ bar

4.6.2 Three-phase equilibrium hydrocarbon mixture-water

Three-phase equilibrium flash calculations were performed at $T=310$ K for mixture B mixed with water. The phase distribution of mixture B+ H_2O is plotted in fig. 4.11a for the pressure range in the three-phase region.

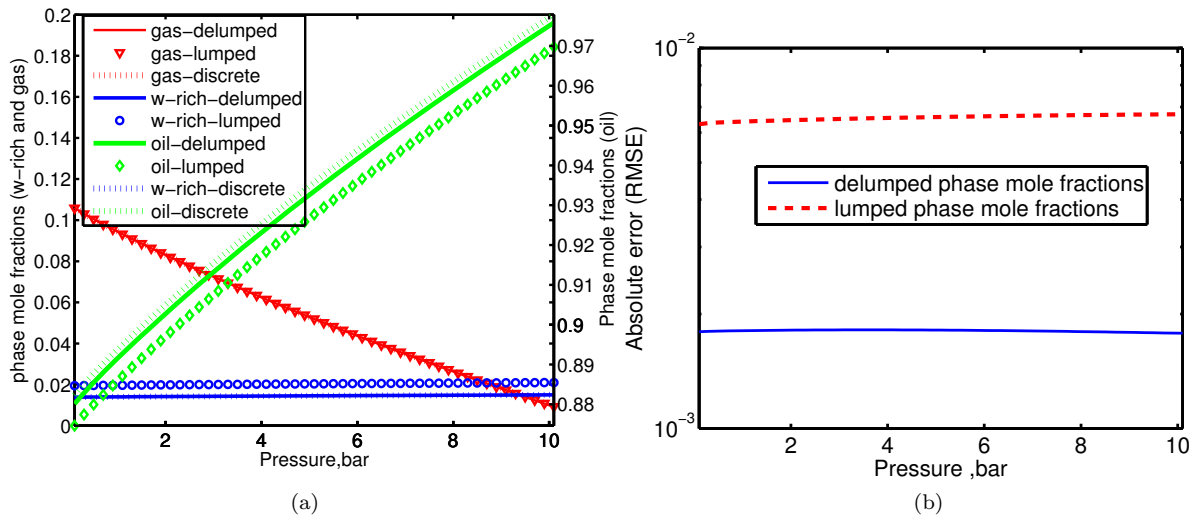


Figure 4.11: a) Phase mole fractions for mixture B + H_2O at $T = 310$ K and b) Errors in phase mole fractions for mixture B + H_2O at $T=310$ K

Absolute errors (RMSE) in phase mole fractions of delumped and lumped representations with respect to the detailed representation are depicted vs. pressure in fig. 4.11b. In fig. 4.12a, the absolute errors (in an Euclidean norm) are presented separately for each equilibrium phase (gas, oil and water-rich liquid); relative errors are given in fig. 4.12b. The absolute errors are all reasonably small; the highest relative errors are those for the water-rich liquid phase and for the gas phase at pressures just below the bubble point.

It can be noticed from fig. 4.11b that the delumped phase mole fractions are located between the detailed and lumped ones, thus the errors are smaller for the delumped than for the lumped representations (see fig. 4.12a and fig. 4.12b). This confirms earlier observations by [Nichita and Leibovici [2006]] (for two-phase flashes) and [Nichita et al. [2006b]] (for multiphase flashes). Results of the delumping reveal the quality of the lumping procedure (in this case of the semi-continuous description of mixture composition). However,

using any procedure for lumping into pseudo-components leads to certain level of approximation and some information may be lost on phase compositions and thus on the global behavior of the mixture.

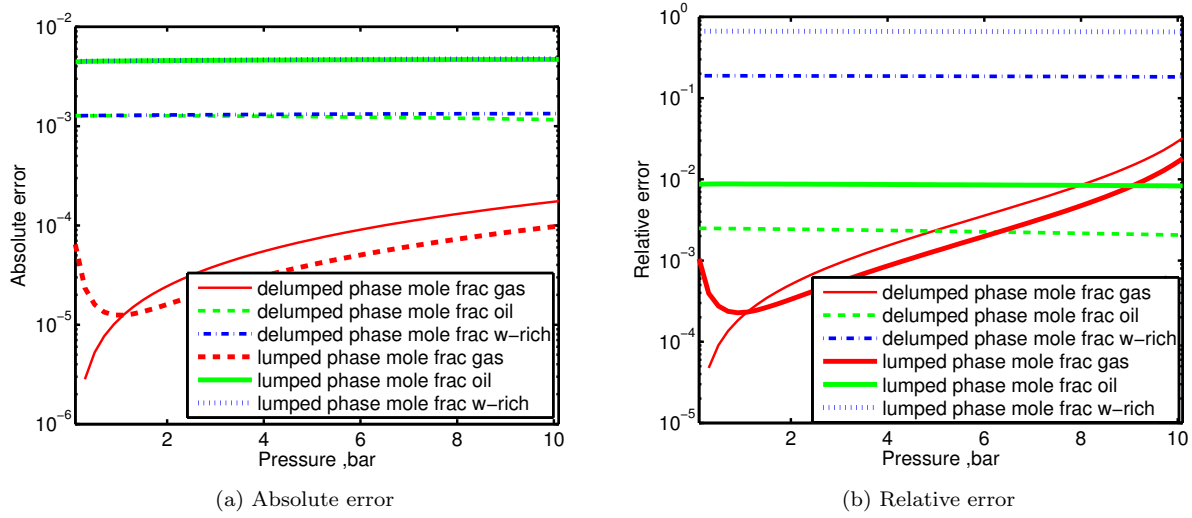


Figure 4.12: Phase mole fractions for mixture B + H₂O at T = 310 K

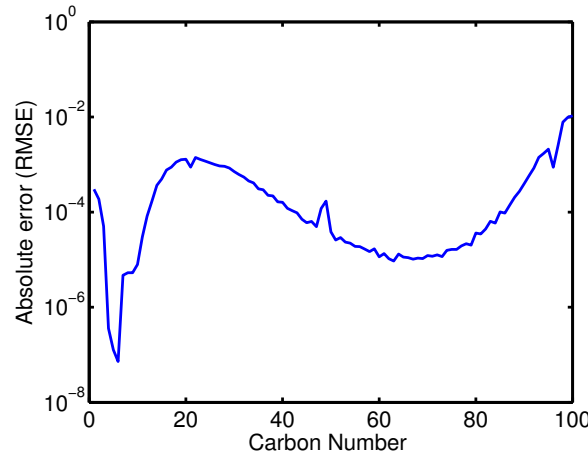


Figure 4.13: Errors in phase compositions for mixture B+H₂O at T=310 K and P=6 bar

The absolute error (RMSE) in component mole fractions is plotted against carbon number at T=310 K and P=6 bar in fig. 4.13. The absolute (fig. 4.14a) and relative (fig. 4.14b) errors (in Euclidean norms) are detailed for the composition of each equilibrium phase.

It must be added that the three-phase equilibrium of this hydrocarbon-water system represents an extreme case for testing the semi-continuous description, because of the extremely low solubility of hydrocarbon components in water and of very small water mole fractions in the gas phase. This explains why the absolute errors are small, while the relative errors may be high.

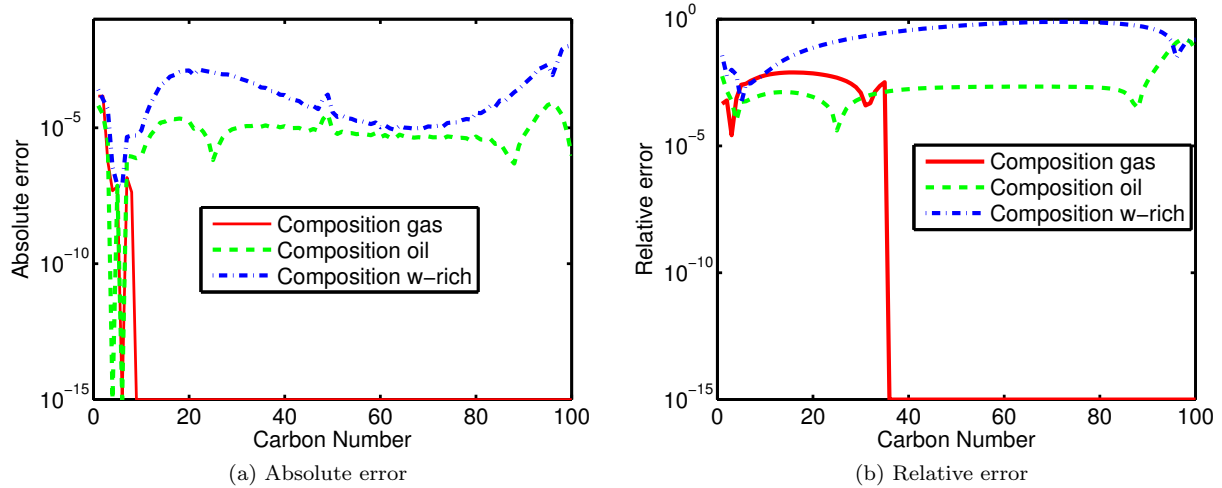


Figure 4.14: Errors in phase compositions for mixture B+H₂O at T=310 K and P=6 bar

4.6.3 Three-phase equilibrium hydrocarbon mixture-carbon dioxide

The phase distribution of mixture A+ CO₂ at T=249 K is plotted in (fig. 4.15) for the pressure range in the three-phase region. When CO₂ is added to the mixture A, the three phase region is very narrow (less than 1 bar). The semi-continuous description can accurately identify it.

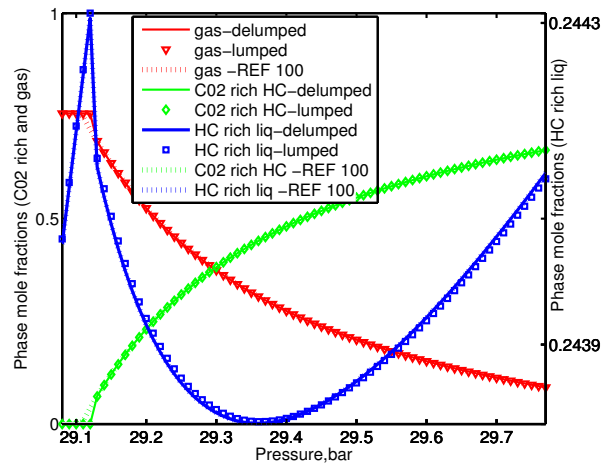
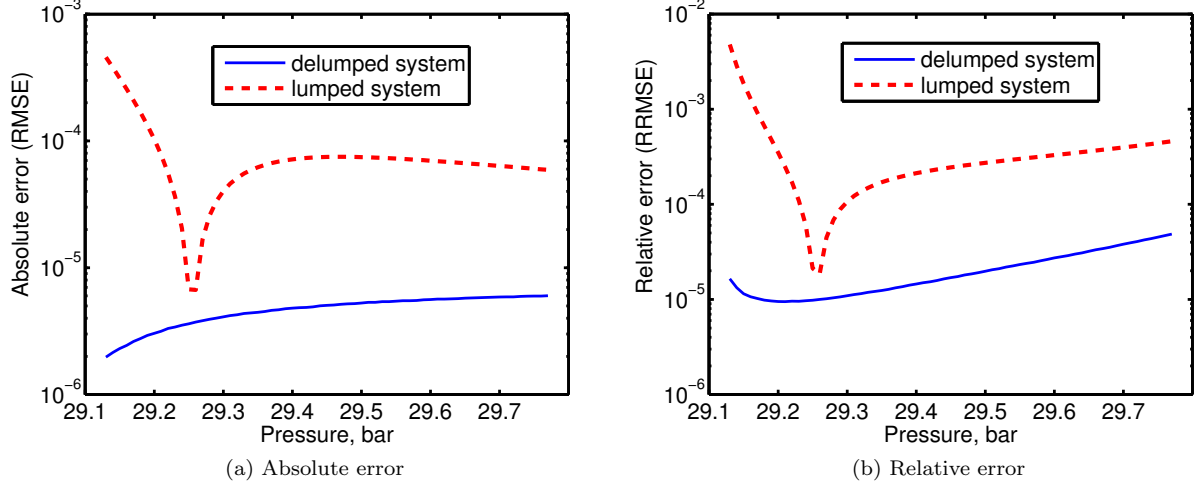
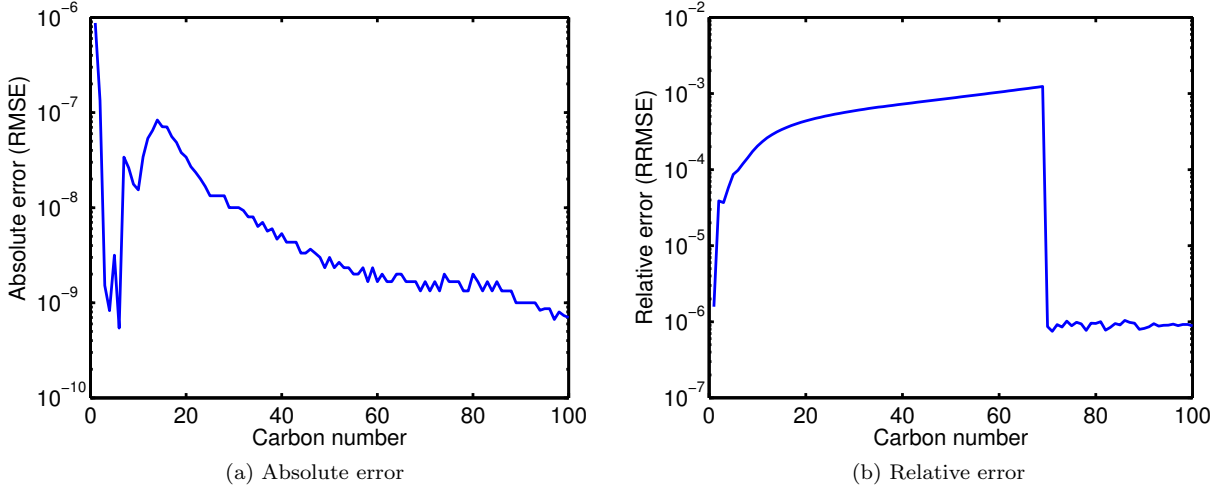


Figure 4.15: Phase mole fractions mixture A+CO₂ at T=249 K

The absolute error (RMSE) and relative error (RRMSE) in phase mole fractions of delumped and lumped representations with respect to the detailed representation are plotted vs. pressure in fig. 4.16a. and fig. 4.16b, respectively.

The absolute errors in component mole fractions (between detailed and delumped feeds) at T=249 K and P=29.5 bar are plotted vs. carbon number in fig. 4.17a; the relative errors are presented in fig. 4.17b. The errors are very small for the continuous part of the mixture (RMSE is of the order of 10⁻⁶ or less, while RRMSE is of the order of 10⁻³ or less), but larger errors can be observed for components in the discrete part of the semi-continuous mixture.

Figure 4.16: Errors in phase mole fraction, mixture A+CO₂ at T=249 KFigure 4.17: Errors in phase compositions for mixture A+CO₂ at T=249 K and P=29.5 bar

4.6.4 Influence of the number of quadrature points on accuracy

The influence of the number of quadrature points (pseudo-components) on the accuracy of two- and three-phase flash calculations is evaluated. First, at a given temperature, errors on phase mole fractions are evaluated over an entire pressure interval. The root mean square error is given in this case by

$$RMSE = \sqrt{\frac{1}{n_{pts}n_P} \sum_{j=1}^{n_{pts}} \sum_{i=1}^{n_P} \left(\beta_{i,j}^{(sc)} - \beta_{i,j}^{(df)} \right)^2} \quad (4.27)$$

where n_{pts} is the number of pressure points.

Then, errors on phase compositions are evaluated at a given pressure, with an RMSE given by

$$RMSE = \sqrt{\frac{1}{n_P n} \sum_{i=1}^n \sum_{j=1}^{n_P} \left(x_{ij}^{(sc)} - x_{ij}^{(del)} \right)^2} \quad (4.28)$$

The absolute errors (RMSE given by eq. 4.27) on phase mole fractions (over the entire two-phase region) vs. the number of quadrature points are plotted for mixture A at T=450 K in fig. 4.18a. In fig. 4.18b,

absolute errors (RMSE given by eq. 4.28) on phase compositions are represented for mixture A as a function of the number of quadrature points at $P=8$ bar. The absolute errors on phase mole fractions (over the entire three-phase region) are plotted vs. the number of quadrature points in fig. 4.19a for mixture A + H_2O at $T=310$ K and in fig. 4.20a for mixture A + CO_2 at $T=249$ K. The absolute errors on phase compositions vs. the number of quadrature points are plotted in fig. 4.19b for mixture A + H_2O at $T=310$ K and $P=6$ bar and for mixture A + CO_2 at $T=249$ K and $P=29.5$ bar.

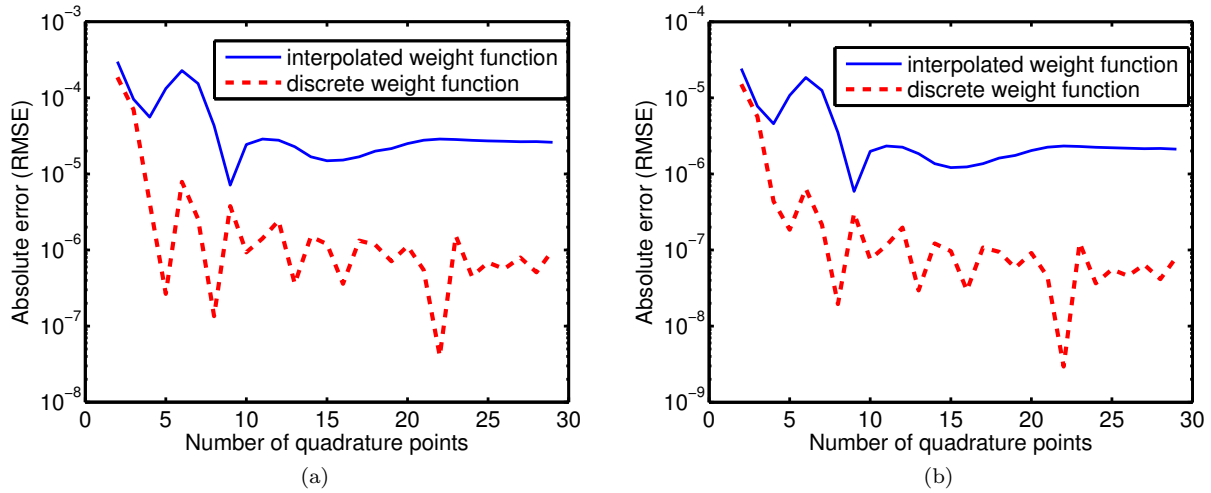


Figure 4.18: Errors vs. the number of quadrature points, mixture A at $T=450$ K. a) In phase mole fractions in the two-phase region; b) in compositions at $P=8$ bar

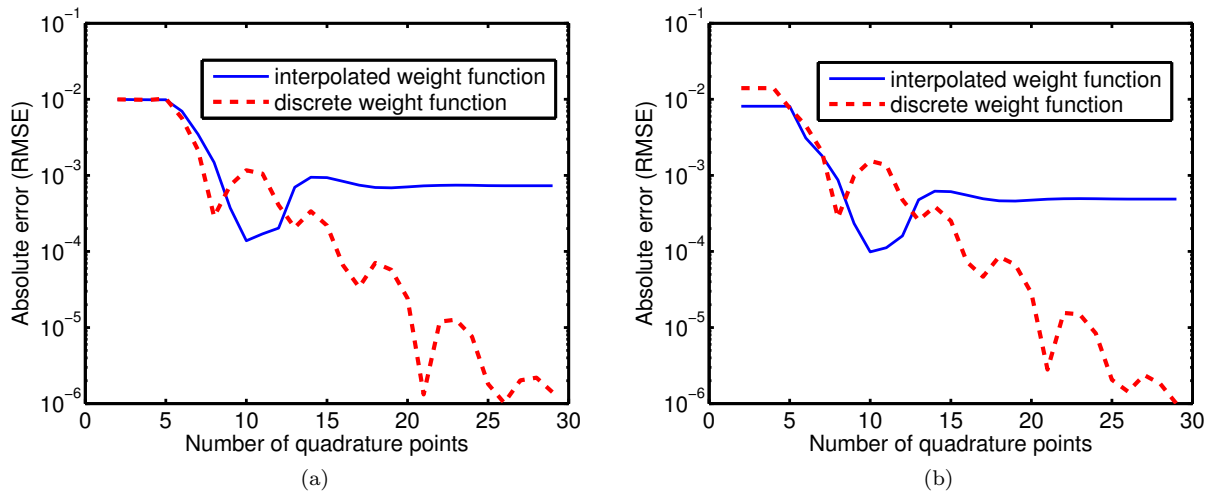


Figure 4.19: Absolute errors vs. the number of quadrature points, mixture A+ H_2O at $T=310$ K; a) In phase mole fractions in the three-phase region; b) in compositions at $P=6$ bar

In Figures fig. 4.18a to fig. 4.20b, errors are plotted for both discrete and interpolated weight functions. In all cases, the errors are at least one order of magnitude lower for the discrete weight function. It can be observed from these figures that there is practically no improvement in accuracy for N greater than 6-8 for the interpolated weight function. However, for the discrete weight function, the decreasing trend is observed for higher numbers of quadrature points, at least for multiphase flashes.

Theoretically, the error plotted against the number of quadrature points should give a monotonically decreasing function with a horizontal asymptote (at $N \rightarrow +\infty$). However, this is not the case, as can be seen from figures fig. 4.18a to fig. 4.20b (a trend line was added to each of these figures). The discrete mixture has been transformed into a semi-continuous mixture by means of a cubic spline; this approximation changes

the initial mole fractions. This is why, even for a large number of calculated pseudo-components, the error between discrete and semi-continuous descriptions never reaches zero; this also explains the non-monotonic behavior for the interpolated weight functions. In the case of the discrete weight functions, the non-monotonic behavior can be explained by the irregularity of the feed composition and by the use of non-zero BIPs.

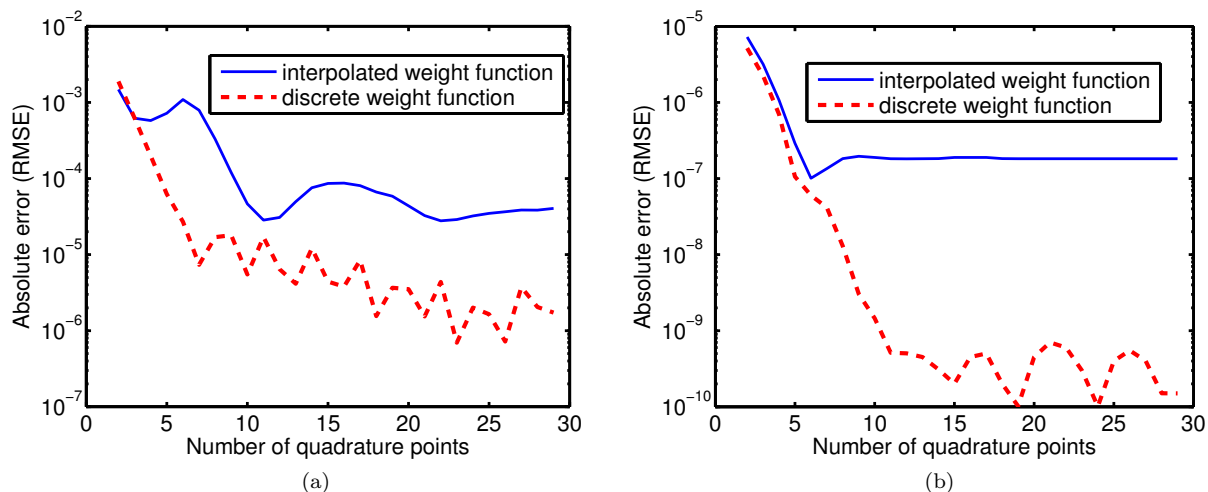


Figure 4.20: Absolute errors vs. the number of quadrature points, mixture A+CO₂ at T=249 K; a) In phase mole fractions in the three-phase region; b) in compositions at P=29.5 bar

4.7 Conclusion

In this work, the quadrature method of moments (QMoM) is applied to multiphase equilibrium calculations for actual oil mixtures (combined with non-hydrocarbon components, such as water and carbon dioxide) using a cubic equation of state with non-zero BIPs. As discussed, Lage's methodology might be rather called the feed-weight quadrature method (FWQM) than QMoM (since the quadrature can be solved also by methods not involving moments directly). The quadrature is solved using a procedure (ORTHOPOL) which avoids problems due to the ill-conditioned nature of the problem and is suitable for an undetermined number of quadrature points (unlike the original formulation of the QMoM).

The key in Lage's QMoM is to take the feed composition as a weight function. In fact, Lage's methodology might be called the feed-weight quadrature method (FWQM) rather than QMoM (since the quadrature can be also solved by methods not involving moments directly). A hidden feature of the QMoM is revealed, by identifying a distribution function used to approximate a newly considered dependence, denoted the liquid to feed mole ratio distribution. It is shown that this dependence is extremely smooth, and, the most important, the shape of the function remains identical for any pressure and temperature conditions for a given feed composition. It must be noted that the LFMR intervenes in a key point of the derivation of the QMoM, and its smoothness acts favorably on its performance. Here the choice of the feed composition as the weight function is based on some physics, that is, on the newly introduced LFMR distribution.

Any calculation procedure for discrete flash calculations can be used for the semi-continuous description (including discrete components and pseudo-components obtained by solving the quadrature). At the solution of the flash calculation on the semi-continuous mixture, an analytical and consistent delumping (inverse lumping) procedure is carried out (with an extremely small computational effort) to recover the detailed composition in each equilibrium phase and to obtain an improved phase distribution. The delumping procedure allows a better comparison between discrete and semi-continuous descriptions, not only on phase

distribution, but also on phase compositions.

The QMoM in our implementation has been tested for different initial compositions of a hydrocarbon mixture for two-phase equilibrium calculations and in the presence of water and carbon dioxide, for three-phase flash calculations. In all cases, the location of the phase boundaries, the phase distributions and the phase compositions are well reproduced by the semi-continuous description.

Since the QMoM relates the discrete initial detailed composition having a large number of components to a small number of components which approximates the continuous portion of the mixture, it can be viewed in a larger sense as a lumping - delumping procedure. It is generally applicable, even in cases when no standard distribution function can model the feed composition, or several distribution functions are needed to model different portions of the mixture or different homologous series. It is proven in this work that [Lage [2007]]'s methodology coupled with a delumping procedure and using a cubic equation of state with non-zero BIPs works for actual oil mixtures with good accuracy.

CHAPTER 5

Reservoir Simulation

Contents

5.1	Introduction	156
5.2	Flow of fluids through porous media	158
5.2.1	Mass conservation	159
5.2.2	Quantity of movement	160
5.2.3	Equation of energy conservation	160
5.3	Fluid properties	162
5.3.1	Phase viscosities	162
5.3.2	Mass densities	163
5.3.3	Relative permeability	163
5.3.4	Enthalpy	164
5.4	Three-phase bypass	166
5.4.1	Introduction	166
5.4.2	Mathematical background	166
5.4.3	Parameterization	168
5.4.4	Status identification	171
5.4.5	Pressure and Temperature parameterization and interpolation	171
5.5	Simulations	172
5.5.1	Two phase case: match with Eclipse (commercial reservoir simulator from Schlumberger)	172
5.5.2	Full isothermal three phase compositional simulations: CO ₂ gas injection	173
5.5.3	Full thermal three-phase compositional simulations for steam injection	179
5.6	Comparisons between different compositional acceleration procedures	189
5.7	Conclusion	190

5.1 Introduction

One of the objectives of the thesis is to perform fully compositional simulations of the steam flooding process. The first reservoir simulators dealing with this process were designed in the late sixties (Spillette and Nielsen [1968], Shutler [1969], Shutler [1970]). The first models were based on a three-phase linear model [Shutler [1969]], then extended to two dimensions [Shutler [1970]]. Later on, [Vinsome [1974]] introduced an IMPES method to simulate the steam drive process. [Coats [1976]] proposed the first simulator coupling thermal and compositional modelling. [Coats [1978]] presented an extension of [Coats [1976]] on the base of a three-dimensional, compositional model to simulate steam injection. Later on, [Ishimoto [1985]] proposed to compute the hydrocarbon phase properties by means of an equation of state and to use Raoult's law to compute the solubility of water in the oil. Then, [Chein et al. [1989]] proposed a general compositional

simulator which could deal with thermal options using both K-value and EOS-based formulations. Once again, the water phase was assumed to be an ideal solution.

More recently, [Cicek and Ertekin [1996], and later Cicek [2005]] developed a fully implicit compositional multiphase simulator to simulate steam injection problems. In their formulation, water is treated within the equilibrium calculations. The simulator is based on K-value models which can be tuned to match phase-diagrams. However, the tables are generally functions of one oil component, the pressure and the temperature. The K-values approach generally does not gives correct solubilities of hydrocarbon components in the water phase and solubilities of water in the hydrocarbon liquid phase.

It has been observed (fig. 6) however, that for high temperatures, the solubility of the water component in the hydrocarbon rich phase was not negligible. The current commercial softwares make some approximations to simulate steam injection problems. Those approximations are even more important in case of heavy oils. ECLIPSE 500 and INTERSECT for instance, are thermal compositional simulators working with K-Values, or (in case of INTERSECT) which treat water separately from the equilibrium flash calculation; this can lead to important inaccuracies.

For this reason, some authors started to simulate steam injection with EoS-based full compositional reservoir simulators. Using the fact that the solubility of hydrocarbons within the water phase is negligible for some ranges of pressures-temperatures, for steam flooding problems some authors have developed a free water flash based on this assumption [Luo and Barrufet [2005]] and more recently applied to heavy oil SAGD problems [Heidari [2014]]. The same methodology has been applied to in-situ upgrading [Lapene [2010]].

For cold CO₂ injection problems, no assumption can be made and some authors have extended simulator models with full four-phase multiphase flash problems. [Varavei and Sepehrnoori [2009] and Okuno [2009]] developed a four-phase EOS-based simulator. As the solubility of water in hydrocarbons is negligible for small temperatures, they did not include water within the equilibrium calculations.

However, the reservoir water is generally closer to brine than to pure water and the free-water hypothesis is not valid anymore. [Brantferger [1991]] developed a simulator using an equation of state to calculate the thermodynamic properties of each phase (even for the water phase). They proposed an isenthalpic flash, taking enthalpy as primary variable instead of the temperature. [Voskov et al. [2009]] proposed a general purpose reservoir simulator. They developed an efficient way to calculate the phase equilibrium based on the tie-line parameterization of the flash. [Varavei and Sepehrnoori [2009]] and more recently [Zaydullin et al. [2014]] developed full three phase flash simulations for steam injection problems.

Finally, [Feizabadi [2013]] extended compositional simulations to simulate isothermal solvent injection problems with a full four-phase EOS-based compositional simulator.

However, to the best of our knowledge, no case of fully EoS based compositional simulations of steam injection (like the SAGD process) with extra-heavy oil has been proposed in the literature yet, treating the water phase within the equilibrium calculations (and without any limiting assumption). This kind of simulations seem to be important and will offer improved tools for predictive studies for the heavy oil fields.

In this chapter, a description of the different equations used in reservoir simulations will be given, as well as the different models to represent the physical properties of the fluids.

In the previous chapter of this work, new algorithms for the equilibrium calculation problems were presented. Based on this work, a program dealing with multiphase calculations has been developed integrating all the presented features (see chapters 2 and 3), called MFlash in this thesis.

The program will be first tested in standalone calculations (i.e. without any reservoir simulator) for various mixtures, and the results will be compared with literature and experimental data. Then some tests with different reservoir simulators will be performed to check the program efficiency and robustness.

MFlash has been integrated into two reservoir simulators the T-PP platform (internal reservoir simulator from Total S.A.) and AD-GPRS from the University of Stanford. The simulator is based on automatic differentiations [Younis [2009] and Zhou [2012]], which allows a treatment of nonlinear physics [Voskov et al. [2009d], Voskov and Zhou [2012]]. It is based on flexible discretizations such as Multi-Point Flux Approximation and on general Adaptive Implicit Method [Zhou et al. [2011]], high performance computing [Zhou et al. [2011] and Tchelepi and Zhou [2013]], thermal compositional simulations [Iranshahr et al. [2010a], Zaydullin et al. [2013]]. It also can be coupled with geomechanics [Garipov et al. [2012]] and include a joint-based optimization [Kourounis et al. [2010], Volkov and Voskov [2013]].

Different processes will be simulated. First, a two-phase simulation in presence of water treated separately from the equilibrium calculations, will be tested. A comparison with the ECLIPSE commercial simulator will also be presented.

Then, three and four-phase fully compositional simulations of CO₂ injection problems will be described. In different simulations, it has been noticed that when co-injecting solvent with the steam, sometimes a solvent-rich phase could appear. We did not investigate the importance of modelling this phase for the oil production forecast, but in an attempt to develop a robust thermodynamic package to simulate the SAGD process, a four-phase simulation has been successfully tested and will be presented for isothermal CO₂ injection problems.

Finally, for the thermal processes, fully compositional simulations of the steam-flooding process in heterogeneous reservoir as well as fully compositional simulations of the SAGD process will be performed with an Athabasca bitumen.

In this last part, different methodologies improving computational times in the reservoir simulations will be compared for all the presented simulations:

- The reduced variables for the stability analysis against conventional variables
- The bypass approach proposed by [Rasmussen et al. [2006]], which avoids stability computations (called B-R) in this thesis
- The procedure from [Voskov and Tchelepi [2008], Voskov and Tchelepi [2009a], Voskov and Tchelepi [2009b], Iranshahr et al. [2010a]] and more recently [Zaydullin et al. [2013]] based on the tie-simplex parameterization of the space will be used together with MFlash.

5.2 Flow of fluids through porous media

At initial conditions, a petroleum reservoir is said to be in hydrostatic equilibrium. All forces (such as capillarity pressure, gradient pressure, gravity, thermodynamic) are balanced and are in equilibrium. However, as soon as fluid is injected and/or produced, the forces are not in equilibrium any longer and fluid advection and diffusion appears. Theoretically, the fluid is not in thermodynamic equilibrium either. However, a generally accepted assumption in the reservoir simulation framework is to consider the fluid in local thermodynamic equilibrium. The continuum model can be seen as a large number of differentially small subsystems [Islam et al. [2010]].

5.2.1 Mass conservation

5.2.1.1 Accumulation term

For a multicomponent system, the total mole number of a component i in control volume dV is given by:

$$\phi \sum_{j=1}^{np} x_{ij} \rho_j S_j dV \quad (5.1)$$

with

- ρ_j , the molar density of the phase j (mol.m^{-3}) (all the units will be given in S.I.)

$$\rho_j = \frac{n_j}{V_j} \quad \forall j = 1, np \quad (5.2)$$

- n_j , the mole number of the phase j (mol)
- S_j , the saturation of the phase j :

$$S_j = \frac{V_j}{\sum_{l=1}^{np} V_l} \quad \forall j = 1, np \quad (5.3)$$

- V_j , the volume of phase j (m^3).
- ϕ , the porosity of the volume dV

Hence, the rate of change of the mole number of component i in dV is given by:

$$\frac{\partial}{\partial t} \left(\phi \sum_{j=1}^{np} x_{ij} \rho_j S_j \right) dV \quad (5.4)$$

5.2.1.2 Advective term

A part of the mole accumulation in dV comes from the transport of components within the phases that flows in and out of the control volume. Let dS be the differential element of the control volume. The rate of inflow and outflow of the component i across the boundary of the control volume dV is given by

$$\sum_{j=1}^{np} -\mathbf{n} \cdot \mathbf{v}_j x_{ij} \rho_j dS = -\nabla \cdot \left(\sum_{j=1}^{np} \mathbf{v}_j x_{ij} \rho_j \right) dV \quad (5.5)$$

with \mathbf{v}_j the velocity of the phase j (m.s^{-1}) and \mathbf{n} the vector normal to dS .

5.2.1.3 Diffusive and dispersive term

Diffusion is mathematically treated with a diffusion tensor \mathbf{D}_{ij} of component i in phase j . The flux of diffusion is taken to be

$$-\mathbf{D}_{ij} \cdot \nabla \rho_j x_{ij} \quad \forall i = 1, nc \quad \forall j = 1, np \quad (5.6)$$

where \mathbf{D}_{ij} is in ($\text{m}^2 \cdot \text{s}^{-1}$)

The total inflow and outflow of component i which goes out of the control volume dV is then given by:

$$-\mathbf{n} \cdot \phi \sum_{j=1}^{np} \mathbf{D}_{ij} \cdot \nabla \rho_j x_{ij} dS = \nabla \cdot \left(\phi \sum_{j=1}^{np} \mathbf{D}_{ij} \cdot \nabla \rho_j x_{ij} \right) dV \quad (5.7)$$

When a fluid flows through a porous medium, the non-homogeneous velocity encountered at the pore-scale (different velocity within each pore) increases the molecular mixing. This is called the dispersion [Larry [1989]]. The dispersion coefficient is proportional to the local phase velocity. This term is generally comparable or greater than molecular diffusion.

Generally high numerical diffusion is observed at the field scale simulation. Therefore, the physical diffusion and dispersion are generally not taken into account.

5.2.1.4 Mass conservation equation

Neglecting the diffusive and dispersive terms at field scale, the mass conservation equation of the component i is given by:

$$\frac{\partial}{\partial t} \left(\phi \sum_{j=1}^{np} x_{ij} \rho_j S_j \right) + \nabla \cdot \left(\sum_{j=1}^{np} \mathbf{v}_j x_{ij} \rho_j \right) + \sum_{j=1}^{np} \rho_j q_j = 0 \quad \forall i = 1, nc \quad (5.8)$$

where a source term $\sum_{j=1}^{np} \rho_j q_j$ is added in the unit volume, q_j is the reduced source of phase j (reciprocal of time constant (s^{-1})).

5.2.2 Quantity of movement

At pore scale where the approximation of continuous media exists, the Navier-Stokes equations can be written for the quantity of movement conservation. At macroscopic (Darcy) scale, for a 3D flow system with the gravitational force, the Navier-Stokes equation can be simplified to give the generalized Darcy's law:

$$\mathbf{v}_j = -\frac{k_{rj}}{\mu_j} \mathbf{K} (\nabla p_j - \rho_{m,j} \wp \nabla z) \quad j = 1, np \quad (5.9)$$

with

- p_j phase pressure (Pa), and $p_{ij} = p_i - p_j$, capillary pressure between phases i and j .
- k_{rj} the relative permeability of the phase j .
- $\rho_{m,j}$ the mass density of the phase j ($kg.m^{-3}$).
- $\wp \nabla z$ gravity acceleration ($m.s^{-2}$).
- \mathbf{K} is the permeability tensor (m^2).
- μ_j is the viscosity of the phase j ($Pa.s$).

5.2.3 Equation of energy conservation

5.2.3.1 Accumulation

The total energy of a unit volume of the fluid system is given by the sum of the internal energy for each phase.

$$\sum_{j=1}^{np} \rho_j S_j U_j \quad (5.10)$$

where U_j is the specific internal energy of the phase j (in $J.mol^{-1}$). For a porous media, the energy coming from the rock should be added:

$$\phi \sum_{j=1}^{np} \rho_j S_j U_j + (1 - \phi) \rho_r U_r \quad (5.11)$$

where U_r is the rock specific internal energy (here per unit of mass), ρ_r is rock mass density.

5.2.3.2 Heat conduction

From the Fick's law, the heat conduction through the reservoir fluid and rock are:

$$Q_f = -\lambda_f \nabla T_f \quad Q_r = -\lambda_r \nabla T_r \quad (5.12)$$

where λ_f , λ_r are respectively the coefficient of thermal conductivity of the fluid and the rock (in $W.m^{-1}.K^{-1}$); T_f and T_r are respectively, the reservoir fluid temperature and the reservoir rock temperature (in K).

In many cases, the rock and the reservoir fluid temperature can be assumed identical, i.e. $T_f = T_r = T$, as a consequence, the difference between inflow and outflow for the heat flux is then given by:

$$-\nabla \cdot (\lambda \nabla T) \quad (5.13)$$

where λ is bulk thermal conductivity.

5.2.3.3 Convective term in energy transfer

The enthalpy can be viewed as the sum of the internal energy of a fluid and the work performed on the system.

$$H = U + PV \quad (5.14)$$

Using the specific enthalpy of a phase ($h_j = H_j/n_j$ (in $J.mol^{-1}$)), the convective term of the energy equation is given by:

$$\nabla \cdot \left(\sum_{j=1}^{np} \mathbf{v}_j \rho_j h_j \right) dV \quad (5.15)$$

5.2.3.4 Energy conservation equation

The energy balance may be given in the form:

$$\begin{aligned} 0 = & \frac{\partial}{\partial t} \left[\phi \sum_j \rho_j S_j U_j + (1 - \phi) \rho_r U_r \right] \\ & + \left[\nabla \cdot \left(\sum_{j=1}^{np} \rho_j h_j \mathbf{v}_j \right) \right] - \nabla \cdot (\lambda \nabla T) + \sum_{j=1}^{np} h_j \rho_j q_j \end{aligned} \quad (5.16)$$

where $\sum_{j=1}^{np} h_j \rho_j q_j$ accounts for the source term, and q_j is the same as for (eq. 5.8).

5.3 Fluid properties

5.3.1 Phase viscosities

5.3.1.1 Light-Medium fluid model

The LBC (Lohrenz-Bray-Clark) viscosity model [Lohrenz et al. [1964]] (model incorporated in ECLISPE and INTERSECT) is given by:

$$\mu = \mu^* + \frac{1}{\zeta} \left[\left(\sum_{k=0}^4 a_k \rho_r^k \right)^4 \exp \left(\frac{\rho_r}{\rho_{max,r} - \rho_r} \right) - 10^{-4} \right] \quad (5.17)$$

$$\mu_i^* = \begin{cases} 3.4 \cdot 10^{-4} \frac{T_{ri}^{0.94}}{\zeta_i} & \text{if } T_{ri} < 1.5 \\ 1.778 \cdot 10^{-4} \frac{(4.58 T_{ri} - 1.67)^{5/8}}{\zeta_i} & \text{if } T_{ri} \geq 1.5 \end{cases}$$

with

$$T_{ri} = \frac{T}{T_{ci}} \quad \zeta_i = \alpha \cdot T_{ci}^{1/6} M_{wi}^{-1/2} P_{ci}^{-2/3} \quad (5.18)$$

In the original LBC correlation [Lohrenz et al. [1964]],

$$\mu^* = \frac{\sum_{i=1}^{nc} x_i \mu_i^* \sqrt{M_{wi}}}{x_i \sqrt{M_{wi}}} \quad (5.19)$$

with x_i the phase mole fraction of the phase

$$\rho_r = \rho \sum_{i=1}^{nc} x_i V_{ci} \quad \rho_{max,r} = \infty \quad (5.20)$$

$$\zeta = \alpha \cdot \left[\sum_{i=1}^{nc} x_i T_{ci} \right]^{1/6} \left[\sum_{i=1}^{nc} x_i M_{wi} \right]^{-1/2} \left[\sum_{i=1}^{nc} x_i P_{ci} \right]^{-2/3} \quad (5.21)$$

This model is not suited for heavy oils. In other cases, this model can be used to model the viscosity of any phase.

5.3.1.2 Heavy oil

For heavy oil applications, other models have been integrated. For the oil phase, an exponential law has been used for each component:

$$\mu_i = a_i \exp \left(\frac{b_i}{T} \right) \quad (5.22)$$

The parameters a_i and b_i are computed solving a least-square problem to fit the experimental data.

For the water phase, the viscosity model is computed using the STARS formulation. Finally for the gas phase,

$$\mu_i = c_i T^{d_i} \quad (5.23)$$

The parameters c_i and d_i are computed by solving a least-square problem with the experimental data.

In the end, the viscosity of the fluid can be computed for each phase base on the relation

$$\mu = \sum_{i=1}^{nc} x_i \mu_i \quad (5.24)$$

5.3.2 Mass densities

5.3.2.1 STARS liquid densities

For the extra heavy oil case (SAGD), TOTAL S.A. provided the input parameters to the STARS model to fit the experimental values, so the liquid densities were computed using the STARS formulation for this test case.

The pure component mass density is given by

$$\rho_{m,i} = \rho_i^0 \exp [ct_{1k}(T - T_{ref}) + ct_{2k}(T^2 - T_{ref}^2)/2 - cp_k(p - p_{ref}) - cpt_k(p - p_{ref})(T - T_{ref})] \quad (5.25)$$

With ρ_i^0 is the pure component mass density at referent temperature and pressure, ($kg.m^{-3}$), ct_{1k} (in K^{-1}), ct_{2k} (in K^{-2}), cp_k (in Pa^{-1}) and cpt_k (in $Pa^{-1}K^{-1}$) are coefficients which can be tuned to match the correct mass density of the fluid. The phase density is given by:

$$\rho = \sum_{i=1}^{nc} x_i \rho_{m,i} \quad (5.26)$$

5.3.2.2 Volume translated Peng-Robinson

For all the other cases, the densities were computed based on the EoS model. The Peng-Robinson equation of states generally does not predict the liquid densities accurately. Corrections of volume are necessary to fit the densities observed experimentally. These corrections do not change the equilibrium results and depend on the temperature and the compositions:

$$V = V^{PR} + V^{CORR} \quad (5.27)$$

with

$$V^{CORR} = \sum_{i=1}^{nc} C_i n_i \quad (5.28)$$

$$C_i = B_i (\alpha_i + \gamma_i T) \quad (5.29)$$

Here, α_i (no unit) and γ_i (K^{-1}) can be tuned to match the experimental volumes and B_i is the EOS parameter

$$B_i = \frac{\Omega_b RT_{ci}}{P_{ci}} \quad (5.30)$$

5.3.3 Relative permeability

5.3.3.1 Two- and three-phase relative permeabilities

For two phase systems (i.e. water/oil), [Brooks and Corey [1964]], proposed a relative permeability model based on the saturations:

$$k_{rw} = (S^*)_w^n \quad (5.31)$$

$$k_{ro} = (1 - S^*)_o^n \quad (5.32)$$

where

$$S^* = \frac{S_w - S_{wc}}{1 - S_{wc} - S_{or}} \quad (5.33)$$

here

- S_{wc} the irreducible critical saturation; the value for which the water start flowing.
- S_{or} the residual saturation of the oil.
- S_w the saturation of the water phase

For three phase systems, [Stone [1973]] proposed a model which interpolates between the two phase relative permeabilities.

$$S_o^* = \frac{S_o - S_{om}}{1 - S_{wc} - S_{om}} \quad S_o \geq S_{om} \quad (5.34)$$

$$S_w^* = \frac{S_w - S_{wc}}{1 - S_{wc} - S_{om}} \quad S_w \geq S_{wc} \quad (5.35)$$

$$S_g^* = \frac{S_g}{1 - S_{wc} - S_{om}} \quad S_o \geq S_{or} \quad (5.36)$$

S_{om} being the residual saturation of the oil in the three phase system.

[Fayers and Matthews [1984]] proposed to evaluate S_{om} with the following formulas:

$$S_{om} = \alpha S_{orw} + (1 - \alpha) S_{org}; \quad \alpha = 1 - \frac{S_g}{1 - S_{wc} - S_{org}} \quad (5.37)$$

where S_{orw} is the oil residual saturation in the oil/water system and S_{org} is the oil residual saturation in the oil/gas system.

5.3.3.2 Four-phase relative permeabilities

[Yuan and Pope [2012]] developed new continuous permeability models, which could be used to four phases. More recently, [Varavei [2009]], [Feizabadi [2013]], choose a [Brooks and Corey [1964]] model to model four phase relative permeabilities. This model is not physical because each phase moves independently from each other, which should not be the case. This is the model chosen in this work for the four phase simulation test cases.

$$\begin{aligned} S_{we} &= \frac{S_w - S_{wr}}{1.0 - S_{wr} - S_{gr}} & k_{rw} &= k_{rw}^0 (S_{we})^{ew} \\ S_{ge} &= \frac{S_g - S_{gr}}{1.0 - S_{wr} - S_{gr}} & k_{rg} &= k_{rg}^0 (S_{ge})^2 [1 - (1 - S_{ge})^{eg}] \\ S_{oe} &= \frac{S_o - S_{or}}{1.0 - S_{wr} - S_{or} - S_{gr}} & k_{ro} &= k_{ro}^0 (S_{oe})^2 [1 - (1 - S_{oe})^{eo}] \\ S_{le} &= \frac{S_l - S_{lr}}{1.0 - S_{wr} - S_{lr} - S_{gr}} & k_{rl} &= k_{rl}^0 (S_{le})^2 [1 - (1 - S_{le})^{el}] \end{aligned} \quad (5.38)$$

5.3.4 Enthalpy

5.3.4.1 EoS based enthalpy of the oil and gas phases

Let x_i be the composition of the component i for a given phase. The enthalpy H of a given phase at a given pressure and temperature, can be expressed by

$$H(p, T, \mathbf{n}) = \sum_{i=1}^{nc} x_i H_i^*(T) + H^R(p, T, \mathbf{n}) \quad (5.39)$$

5.3.4.1.1 Ideal heat capacity and ideal enthalpy

The heat capacity of an ideal gas can be approximated with a polynomial function of the temperature [Reid et al. [1987]]. For an ideal gas, the molar ideal gas heat capacity is given by:

$$Cp_i^*(T) = Cp_{A,i} + Cp_{B,i}T + Cp_{C,i}T^2 + Cp_{D,i}T^3 \quad (5.40)$$

[Passut and Danner [1972]] provide data for values of $Cp_{A,i}$ to $Cp_{D,i}$ for the most common components in Petroleum Engineering. By definition the molar ideal gas enthalpy of the component i can be calculated from

$$H_i^*(T) = \int_{T_{ref}}^T Cp_i^*(x) dx \quad (5.41)$$

which leads to

$$H_i^*(T) = Cp_{A,i}(T - T_{ref}) + \frac{Cp_{B,i}}{2}(T^2 - T_{ref}^2) + \frac{Cp_{C,i}}{3}(T^3 - T_{ref}^3) + \frac{Cp_{D,i}}{4}(T^4 - T_{ref}^4) \quad (5.42)$$

5.3.4.1.2 Residual enthalpy

The residual enthalpy can be computed from the EoS. From [Michelsen and Mollerup [2007]],

$$H^R(T, p, \mathbf{n}) = -RT^2 \sum_{i=1}^{nc} \left(\frac{\partial \ln \phi_i}{\partial T} \right)_{p, \mathbf{n}} \quad (5.43)$$

$$H^R(T, p, \mathbf{n}) = (Z - 1)RT + \frac{T \frac{\partial A}{\partial T} - A}{(\delta_2 - \delta_1)B} \ln \left(\frac{Z + \delta_2 B}{Z + \delta_1 B} \right) \quad (5.44)$$

5.3.4.2 Enthalpy of the water phase

The water properties are not really accurate, when computed with a cubic equation of state. Therefore, for the water phase, another model is used to compute the enthalpy (which is similar to STARS's model). In STARS simulator, the molar heat capacity is computed from

$$Cp_{g,i}(T) = Cp_{E,i} + Cp_{F,i}T + Cp_{G,i}T^2 + Cp_{H,i}T^3 \quad (5.45)$$

By integrating with respect to the temperature, the molar gas enthalpy can be obtained:

$$H_{g,i}(T) = \int_{T_R}^T Cp_{g,i}(x) dx \quad (5.46)$$

which gives integrating (5.45) into (eq. 5.46):

$$H_{g,i}(T) = Cp_{E,i}(T - T_R) + \frac{Cp_{F,i}}{2}(T^2 - T_R^2) + \frac{Cp_{G,i}}{3}(T^3 - T_R^3) + \frac{Cp_{H,i}}{4}(T^4 - T_R^4) \quad (5.47)$$

And the gas enthalpy can be obtained from:

$$H_g(T, \mathbf{n}) = \sum_{i=1}^{nc} x_i H_{g,i}(T) \quad (5.48)$$

with x_i the composition of the component i in the water phase.

$$H_{g,i}(T) = Cp_{E,i}(T - T_R) + \frac{Cp_{F,i}}{2}(T^2 - T_R^2) + \frac{Cp_{G,i}}{3}(T^3 - T_R^3) + \frac{Cp_{H,i}}{4}(T^4 - T_R^4) \quad (5.49)$$

For liquids, the enthalpy of vaporization is subtracted from the mixture enthalpy. The vaporization enthalpy is computed from:

$$Hvap_i = \begin{cases} 0, & \text{if } T < Tc_i \\ Hvr_i(Tc_i - T)^{EV_i}, & \text{else} \end{cases} \quad (5.50)$$

with Hvr_i and EV_i input data for the reservoir simulator. The enthalpy of the water phase becomes:

$$H(T, \mathbf{n}) = H_g(T, n) - \sum_{i=1}^{nc} x_i Hvp_i \quad (5.51)$$

It should be noticed that it is independent from the pressure.

5.4 Three-phase bypass

5.4.1 Introduction

For given pressure and temperature conditions, if for two different feed compositions \mathbf{z}_1 and \mathbf{z}_2 , the same phase compositions are obtained at the minimum of the Gibbs energy, \mathbf{z}_1 and \mathbf{z}_2 belongs to the same tie-simplex.

[Voskov and Tchelepi [2009a]] noticed that for gas injection problems, the solution path involves a limited number of tie-simplex in thermodynamics. Thus, based on a parameterization procedure, they developed an algorithm (CSAT: Compositional Space Adaptive Tabulation) which re-uses the information from previous equilibrium calculation results to bypass the stability for redundant compositions. The procedure has been developed [Voskov and Tchelepi [2009b]], [Voskov and Tchelepi [2009c]], [Iranshahr [2012]]. More recently, [Zaydullin et al. [2013]], [Zaydullin et al. [2014]] extended the procedure to make a more robust algorithm: the three-phase bypass method (called 3PB afterwards). They reported great improvements in computational times. To fulfill the objective of the thesis to decrease the CPU time of the equilibrium calculation part, a collaboration with the University of Stanford was launched to test the 3PB method and compare it with a classical procedure without 3PB. A presentation of the method is given in the following.

5.4.2 Mathematical background

5.4.2.1 Working space

The system of equations to solve when doing multiphase flash calculations is given by (see chapter 2):

$$f_{ij} - f_{i,np} = 0 \quad i = 1 \cdots nc \quad j = 1 \cdots np - 1 \quad (5.52)$$

$$z_i - \sum_{j=1,np} x_{ij} \theta_j = 0 \quad i = 1 \cdots nc \quad (5.53)$$

$$\sum_{i=0}^{nc-1} (x_{i,np} - x_{ij}) = 0 \quad j = 1, np - 1 \quad (5.54)$$

$$1 - \sum_{j=0}^{np-1} \theta_j = 0 \quad (5.55)$$

The overall mole fractions $\mathbf{z} \in R^{nc}$ are usually used as independent variables in reservoir simulations. One of them can be expressed as a linear combination of the others:

$$z_{nc} = 1 - \sum_{i=1}^{nc-1} z_i \quad (5.56)$$

Therefore, the working compositional space is given by:

$$\Delta^{nc-1} = \left[\mathbf{z} \in R^{nc}, z_i > 0, \sum z_i = 1 \right] \quad (5.57)$$

The compositional space can be expressed as a linear combination of the mole fractions \mathbf{x} (eq. 5.53)

$$z_i = \sum_{j=1}^{np} \theta_j x_{ij} \quad i = 1 \cdots nc - 1 \quad (5.58)$$

This means that for a given tie-simplex Δ defined by its composition at the equilibrium (\mathbf{x}), each feed composition which belongs to Δ is only a function of the phase mole fractions. Therefore, for each tie-simplex, each feed composition (vector of $nc - 1$ dimensions) can be parameterized with only np dimensions. The (np) dimensional tie-simplex is defined by:

$$\Delta^{np} = \left[\boldsymbol{\theta} \in R^{np}, \theta_i > 0, \sum \theta_i = 1 \right] \quad (5.59)$$

When the dimension np of the tie-simplex is $np = 2$, the tie-simplex is called a tie-line, when $np = 3$, it is called a tie-triangle.

Each tie-simplex Δ^{np} represents a convex geometrical form in the compositional space (such as line, triangle) where each vertex corresponds to the composition of one of the phases. A ternary diagram (for a given pressure and temperature) is represented in fig. 5.1. A tie-triangle (corresponding to the three-phase region of composition $[\mathbf{x}_i, \mathbf{y}_i, \mathbf{w}_i]$) is represented in grey, and different sets of tie-lines are represented in red, blue and green (which correspond to different two-phase regions).

[Voskov and Tchelepi [2009c]] showed that for a strictly np -phase system, i.e. for each multiphase region, the region is convex and there is a unique tie-simplex which intersects a given feed composition.

They also proved [Voskov and Tchelepi [2008]] that a tie-line parameterization of the space is possible for two phases. Latter on, [Voskov and Tchelepi [2009a]] extended the idea for any number of phases. As long as the tie-simplex space is not degenerated (i.e. the tie-simplex of dimension N which parameterizes the space does not becomes a tie-simplex of dimension $N - 1$), and the phase mole fractions θ can be greater or smaller than 0, the tie-simplex space can be used to parameterize the equilibrium thermodynamics problem.

Based on [Voskov and Tchelepi [2009a]] observation in which few tie-simplexes are accessed during a reservoir simulation, this parameterization was proposed in order to skip most of the stability analysis tests during a simulation.

To illustrate this idea, an example is given in fig. 5.1. To simplify, let's consider the temperature and pressure as fixed. Suppose the tie-simplexes represented in fig. 5.1 have been stored and suppose a stability analysis is required for a composition z_i . As z_i lies within the parameterized tie-triangle, it can be known directly that at equilibrium, three phases are present for this composition and stability analysis can be avoided. In practice the method is more complex and a description of the method is proposed in the next subsections.

5.4.2.2 Continuity of a simplex

Tie-simplexes can be used to parameterize the whole system because [Iranshahr et al. [2013]] proved that the tie-simplex parameterization in the compositional space was a continuous function of pressure, temperature and composition. The continuity property makes a discretization possible.

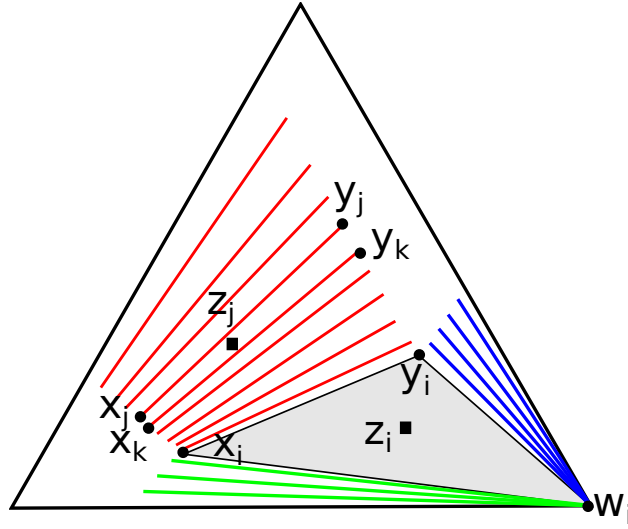


Figure 5.1: Example of parameterization of the tie-simplex space

5.4.2.3 Gibbs energy analysis

Not only is the tie-simplex continuous, but a tie-simplex which parameterizes the thermodynamics in the tie-simplex space has also been shown to minimize the Gibbs energy [Iranshahr et al. [2012]].

The surface Gibbs energy of a mixture is given by:

$$F(\mathbf{z}) = \sum_{i=1}^{nc} z_i \mu_i \quad (5.60)$$

[Iranshahr et al. [2012]] showed that the Gibbs free energy G is convex:

$$G(\mathbf{z}) = \begin{cases} F(\mathbf{z}) & \text{if single phase} \\ \sum_{j=1}^{NP} \theta_j F(\mathbf{x}_j) & \text{if inside a tie-simplex } j \end{cases} \quad (5.61)$$

5.4.3 Parameterization

In this section, the parameterization of the tie-simplex space is presented, first keeping the pressure and temperature constant and then with the variation of both parameters.

5.4.3.1 Parameterization of the compositional space

5.4.3.1.1 Parameterization of tie-simplex giving the maximum number of phases (Δ^{np})

Starting from a composition $\mathbf{z} = \sum_{j=1}^{nc} \frac{\mathbf{x}_j}{np}$, in the middle of Δ^{np} (a negative flash is performed to locate the first tie-simplex leading to the maximum number of phases. An example is given here for three components and three phases (fig. 5.4a). In this figure, the tie-triangle is represented in red.

The composition \mathbf{x} gives the np vertices of the tie-simplex. This composition is stored.

5.4.3.1.2 Parameterization of tie-simplex plane

Starting from the edge of the tie-simplex, in the same plane, a mesh is adaptively created in the space $[\mathbf{x}, \theta]$ (see fig. 5.4a). This mesh can be more or less accurate depending on its size.

The status (number of phases obtained with the equilibrium program) is saved for each vertex of the mesh.

The 3PB algorithm simplifies the problem approximating continuous phase envelopes by means of discrete envelopes (due to the construction of the mesh). In fig. 5.2, a comparison from [Iranshahr et al. [2012]] between the phase envelope computed with CSAT (previous version of 3PB) is carried out with the real one (the same applies for 3PB).

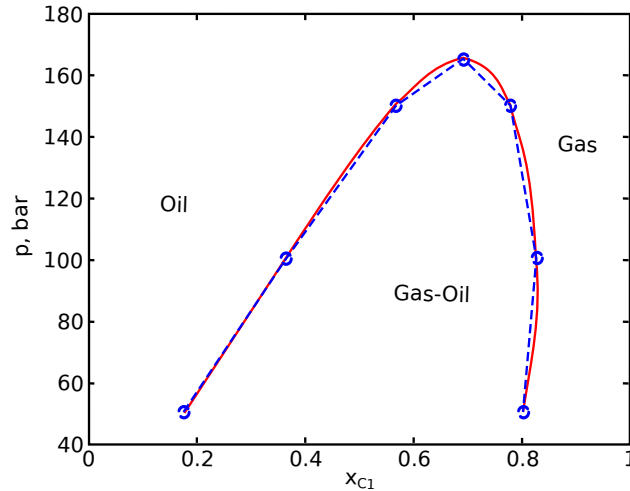


Figure 5.2: Comparisons between the exact phase envelope and the one obtained with CSAT after [Iranshahr et al. [2012]]

Once the plane of the tie-simplex has been meshed, a new composition (taken to be at a distance d from the previous plane) is computed. As long as the new composition lies within the tie-simplex in np dimensions, the whole procedure is repeated (see fig. 5.3)

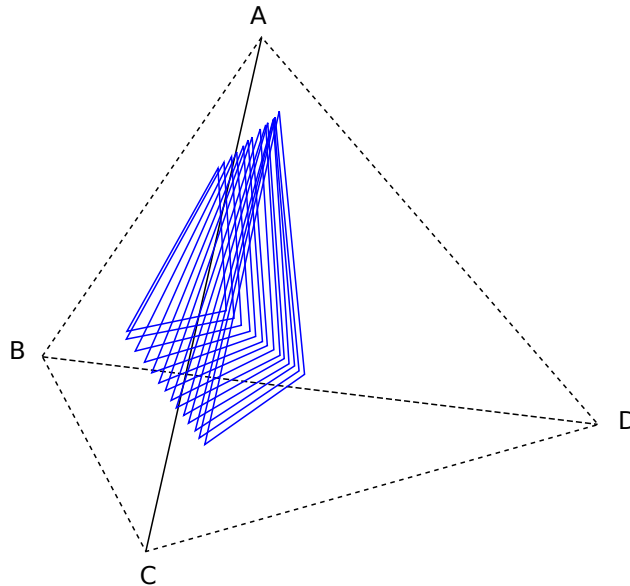


Figure 5.3: Parameterization of different tie-triangle planes

Only a small number of tie-triangles are stored to keep an efficient procedure. The general space is meshed and interpolations are made.

In most of the cases, the composition does not lie exactly within the plane of a stored tie-triangle. This is also the case for pressure and temperature.

5.4.3.2 Parameterization in temperatures and pressures

A good estimation of the range of pressures and temperatures accessed during the injection process in the reservoir simulator can be known before the beginning of the simulation.

[Iranshahr et al. [2010a]] developed a parameterization procedure for the pressure and temperature for CSAT (the same procedure is used in 3PB). Here is a description of the methodology.

For each variables, a discrete grid is created giving regular values of T and p between the a-priori T_{min} and T_{max} (p_{min} and p_{max} respectively) that will be accessed during the simulation.

For a given number of parameterized temperatures (N_T) and pressures (NP), a grid is computed with a step :

$$\Delta p = \frac{p_{max} - p_{min}}{np - 1} \quad \Delta T = \frac{T_{max} - T_{min}}{N_T - 1} \quad (5.62)$$

During the process, interpolation is performed between two adjacent grid cells: $p_i < p < p_{i+1}$ and $T_j < T < T_{j+1}$.

However, just below the MCP (Minimal Critical Pressure), the parameterization is more difficult. The MCP is the minimal pressure for which the composition is intersected by a critical tie-line (a critical tie-line is given by $x_i = y_i$; $\forall i = 1, nc$). Two nearby discrete temperature values may have really distinct MCPs. This is why a refinement must be applied to avoid bad interpolations. A refinement procedure is used close to the MCP is used [Iranshahr et al. [2010a]].

If some pressures and temperatures appear outside of their bounds, the parameterization is extended to include the new equilibrium conditions.

5.4.3.3 Projection in a tie-simplex space

In the parameterization, only few planes of the tie-simplex space are parameterized for memory and CPU time purposes. Most of the time the compositions \mathbf{z} in the tie-simplex space do not belong to any parameterized plane. To obtain the status for a given composition \mathbf{z} , a projection of \mathbf{z} is made to the closest parameterized tie-simplex. The procedure is now presented.

Any point that belongs to the subspace of the tie-simplex is defined by:

$$Z_i(\theta) = \sum_{j=1}^{np} x_{ij} \theta_j \quad (5.63)$$

From a given composition \mathbf{z} , a projection is carried out in each tie-simplex to find the closest one (in Δ^{np}). The solution of this projection gives the phase mole fractions θ_i . The distance from a feed composition to a tie-simplex is given by:

$$F(\theta) = \sum_{i=1}^{nc} (Z_i(\theta) - z_i)^2 = \sum_{i=1}^{nc} \left(\sum_{j=1}^{np} x_{ij} \theta_j - z_i \right)^2 \quad (5.64)$$

From the tie-simplex giving the smallest distance, if the phase mole fractions θ_i , $i = 1, np$ are all within $[0, 1]$, the solution is found. Otherwise, depending on the positive θ_i , it is easy to locate the face of the tie-simplex (and then the $np - 1$ region in which the projection belongs to).

Within this framework,

- if $F > \epsilon$, that means that new tie-simplexes have to be computed and stored;
- else, the solution is found.

5.4.4 Status identification

Once the closer tie-simplex from a given composition has been localized, the projection lies within one element of the parameterized mesh presented earlier. Giving the status of each vertex, the stability analysis can be passed by or not:

- if all the vertices give the same status, the status is assumed to be the same and the global Newton method can be performed directly (fig. 5.5b).
- if one status is different from the others, the equilibrium flash calculation is computed to determine the real status (fig. 5.4b).

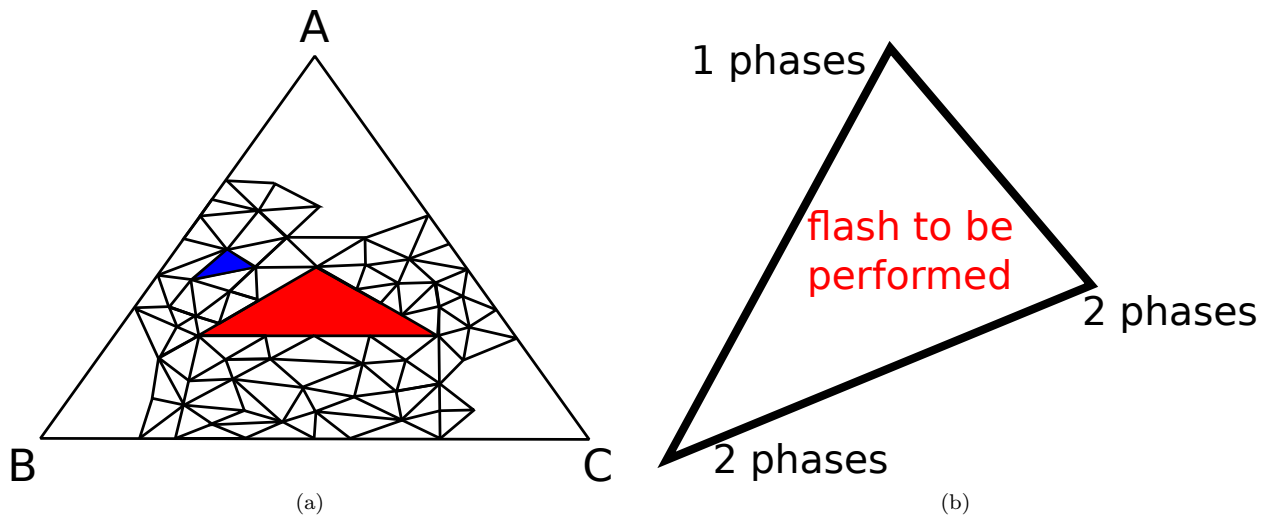


Figure 5.4: a) Parametrization of the space b) Status identification, stability required

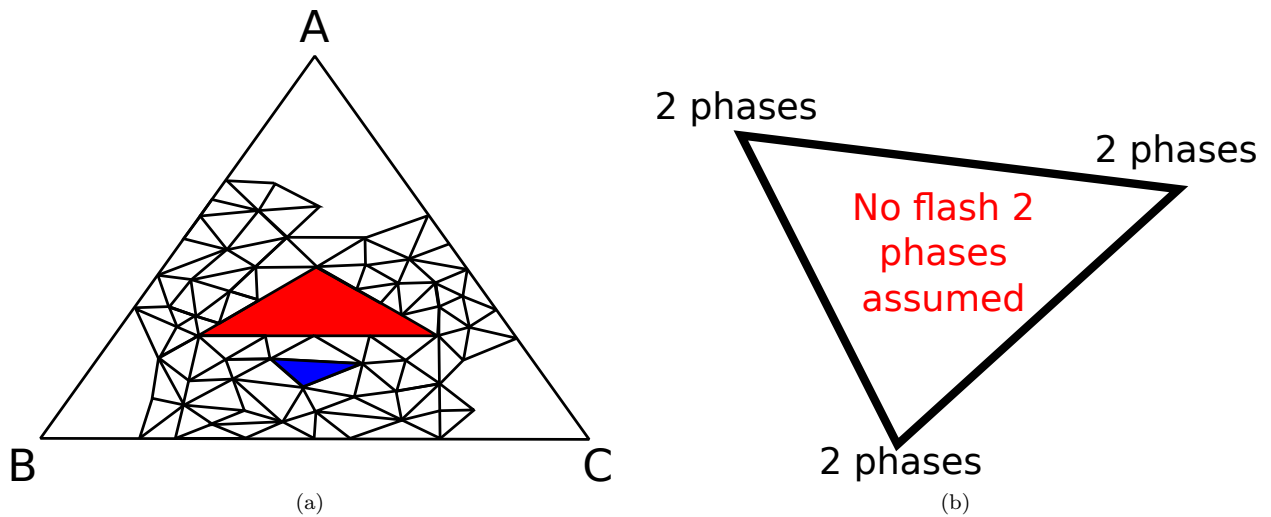


Figure 5.5: a) Parametrization of the space b) Status identification, stability skipped

5.4.5 Pressure and Temperature parameterization and interpolation

The interpolation is first carried out in pressure:

$$\psi_{p,T_k} = \psi_{p_i,T_k} + \delta p (\psi_{p_{i+1},T_k} - \psi_{p_i,T_k}) \quad k = j, j + 1 \quad (5.65)$$

Then in temperature:

$$\psi_{p,T} = \psi_{p,T_j} + \delta T (\psi_{p,T_{j+1}} - \psi_{p,T_j}) \quad (5.66)$$

where ψ represents tie-simplex equilibrium compositions and:

$$\delta p = \frac{p - p_i}{p_{i+1} - p_i} \quad \delta T = \frac{T - T_j}{T_{j+1} - T_j} \quad (5.67)$$

The interpolated tie-simplex is not identical to the actual tie-simplex through the composition and is acceptable only if they are close to each other within a tolerance (5.64). If the composition is far from the interpolated tie-line, a new table of tie-lines in the p-T grid is generated.

In each point interpolated point, one needs to pre-compute an estimate of the MCP to be sure that a parameterization is possible. Moreover, one problem which can occur is when for a given pressure, the two adjacent interpolations give different number of phases. In this case, the flash is performed to secure the computation.

5.5 Simulations

5.5.1 Two phase case: match with Eclipse (commercial reservoir simulator from Schlumberger)

5.5.1.1 Mixture/Reservoir properties

To test MFlash in the reservoir simulation framework, a two-phase case has first been tested. It corresponds to a gas injection problem which is the third comparative SPE test problem [Kenyon and Behie [1987]].

The simulation domain measures 4022m \times 1609m \times 48.768m, the porosity is 13%, Kv=10Kz, and the medium is heterogeneous. The grid is shown in fig. 5.7.

The Peng-Robinson equation of state is used; the component properties are given in table 5.1 and the BIPs in table 5.2.

In this case, MFlash has been integrated within TPP (reservoir simulator from Total S.A.). A comparison is done between MFlash+TPP and the ECLIPSE simulator by Schlumberger.

Comp	Tc, K	Pc, bar	w	Mw, g/mol
CO ₂	310.25	73.87	0.22500	44.01000
N ₂	126.2	33.94	0.04000	28.01300
C ₁	190.6	46.04	0.01300	16.04300
C ₂	305.43	48.84	0.09860	30.07000
C ₃	669.8	42.66	0.15240	44.09700
C ₄₋₆	448.078	35.5	0.21575	66.86942
L ₁	465.618	28.32	0.31230	107.77943
L ₂	587.799	17.07	0.55670	198.56203
L ₃	717.717	11.06	0.91692	335.19790

Table 5.1: SPE3, feed properties

	CO ₂	N ₂	C ₁	C ₂	C ₃	C ₄₋₆	L ₁	L ₂
N ₂	-.0200							
C ₁	.1000	.0360						
C ₂	.1300	.0500	.000000					
C ₃	.1350	.0800	.000000	.000				
C ₄₋₆	.1277	.1002	.092810	.000	.000			
L ₁	.1000	.1000	.130663	.006	.006	.0		
L ₂	.1000	.1000	.130663	.006	.006	.0	.0	
L ₃	.1000	.1000	.130663	.006	.006	.0	.0	.0

Table 5.2: SPE3, BIP

The injection wells are shown in yellow in fig. 5.7. The production wells are shown in blue in the same figure. Water is injected at 100 STB/days, gas is produced at a control gas rate of 500 STB/days.

5.5.1.2 Simulation/Results

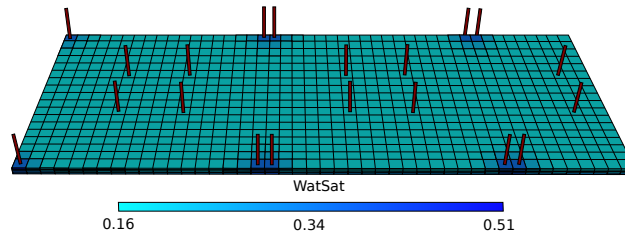


Figure 5.6: Water saturation at the end of the simulation, SPE3 case

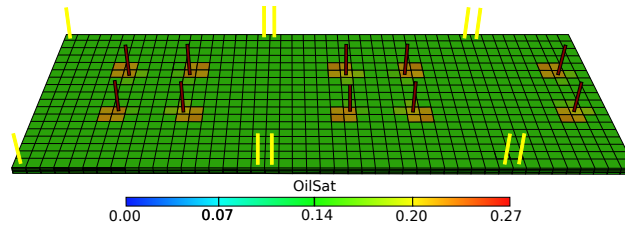


Figure 5.7: Oil saturation at the end of the simulation, SPE3 case

In fig. 5.6 and fig. 5.7, the water saturation (respectively the oil saturation) at the end of the simulations for MFlash+TPP and ECLIPSE is represented (both simulations lead to the same results). In fig. 5.8 the total production rates of water, gas and oil are plotted, for the solution generated with ECLIPSE and the solution generated with MFlash+TPP. The total production rates are identical in the two simulations, validating the two-phase flash developed in this thesis (see chapter 2 and 3).

The remaining simulations were performed by coupling AD-GPRS and MFlash in the framework of a collaboration with Stanford University. The simulator was coupled with the developed flash code, used as an external library. The simulations presented in the following subsections come from this collaboration.

5.5.2 Full isothermal three phase compositional simulations: CO₂ gas injection

The phase equilibrium code has been designed to work with $np > 1$ phases. Multiphase simulations were carried out, first for CO₂-rich/gas/hydrocarbure-rich systems. In this section, three and four phase simulations of CO₂ injection will be presented.

In our simulations, it has been noticed that a solvent-rich phase could appear when co-injecting solvent with the steam in the reservoir. To develop a robust package, the presence of a fourth phase was necessary to

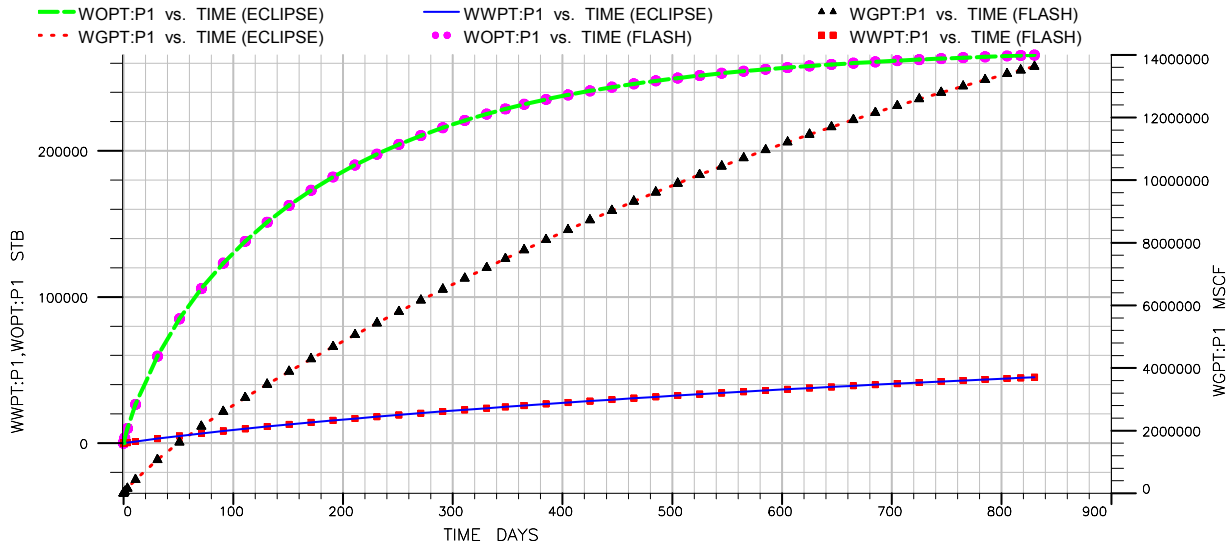


Figure 5.8: Total production vs time for the SPE3 case, comparisons between ECLIPSE and the flash program with TPP

avoid discontinuities which could lead to possible crash of the simulation. So, we investigated the presence of a fourth phase, and developed a four phase CO_2 injection case.

5.5.2.1 Three-phases, CO_2 injection

5.5.2.1.1 Stand-alone simulation

The mixture is the Bob Slaughter Block (BSB) West Texas oil. The characterization of the mixture is taken from [Khan et al. [1992]]. From the experimental data, they provided BIP and critical properties to fit the results. These parameters are given in table 5.3.

At the reservoir conditions, at 313.71 K, the P-z phase diagram of the BSB oil is represented in fig. 5.9 for pressures varying between 35 and 135 bar and compositions of CO_2 between 10 mol% and 99.9 mol%. In red the experimental values of the liquid-vapor/liquid transition and the three phase envelope, obtained by [Khan et al. [1992]] are represented. The results obtained with MFlash are very close to the experimental data.

Components	feed	Molecular weight	Tc	p _C	w	BIP δ_{i,CO_2}
CO_2	0.0337	44.01	304.200	73.765	0.225	-
C_1	0.0861	16.04	160.000	46.002	0.008	0.055
C_{2-3}	0.1503	37.20	344.148	44.992	0.131	0.055
C_{4-6}	0.1671	69.50	463.222	33.996	0.240	0.055
C_{7-15}	0.3304	140.96	605.694	21.749	0.618	0.105
C_{16-27}	0.1611	280.99	751.017	16.541	0.957	0.105
C_{28+}	0.0713	519.62	942.478	16.418	1.268	0.105

Table 5.3: BSB fluid properties

5.5.2.1.2 Reservoir simulations

[Okuno [2009]] developed a test case for this mixture, based on three layers. He developed a four phase simulation (in presence of water), in which the water is treated separately from the flash. In this thesis, the same simulation has been carried out without the water. Further, a four phase case in presence of water will be described.

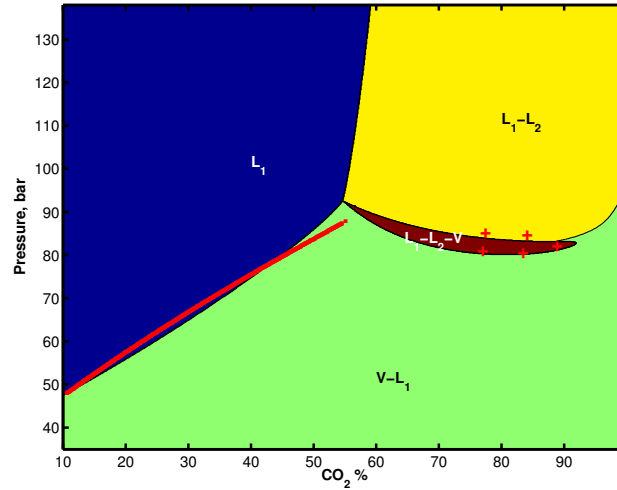


Figure 5.9: Phase envelope BSB comparisons with experimental values

The BSB test case parameters (as given in [Okuno [2009]]) are shown in table 5.4. The injection well is located in the left-hand side of the reservoir and the producer is located in the right hand side (see fig. 5.10). 5% of C_1 and 95% of CO_2 are injected in the injection well at a BHP of 89.63 bar. The producer produces at a BHP of 62.05 bar. The initial reservoir pressure is 75.84 bar.

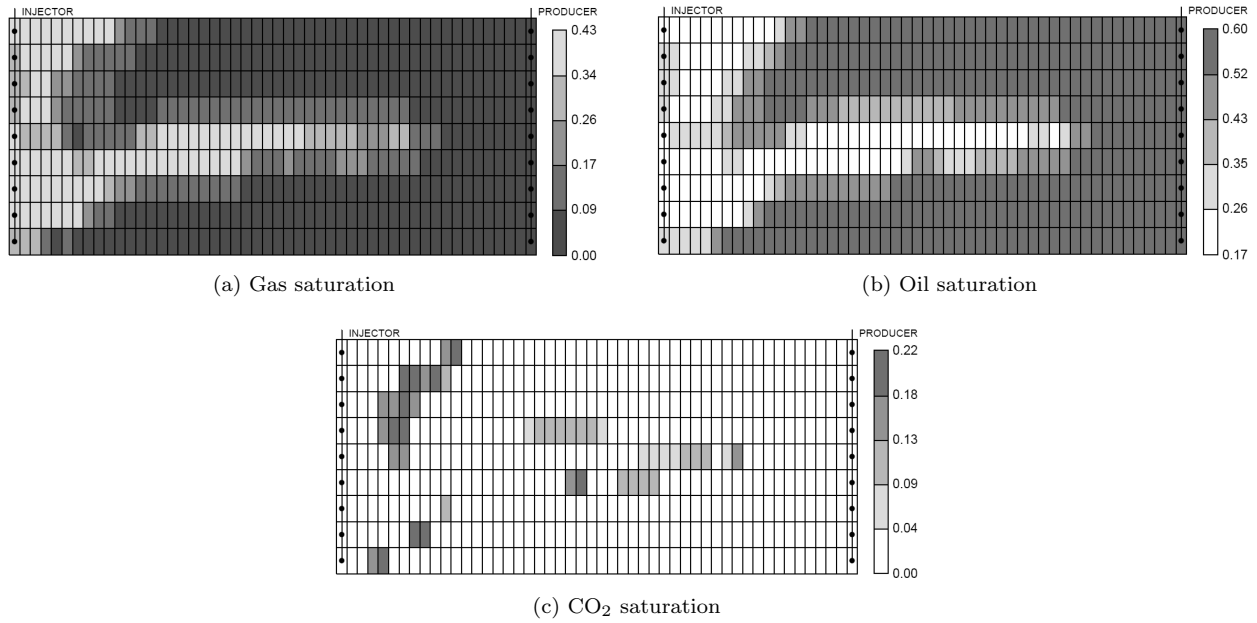
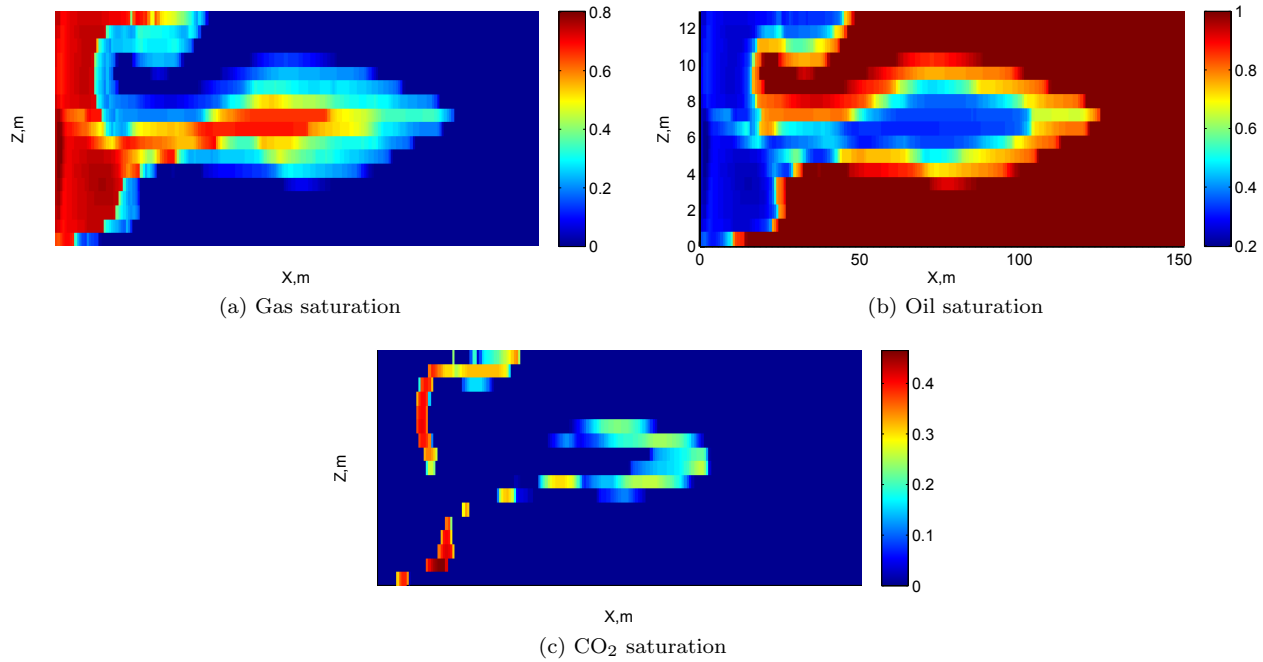
Dimensions of a cell	0.762m \times 3.048m \times 0.762m
Number of grid cells	200 \times 1 \times 18
	Top layer / Middle layer / Bottom layer
Thickness (m)	6.096 / 3.048 / 4.572
Permeability (mD)	7.0 / 11.2 / 9.8
Porosity	0.08 / 0.10 / 0.09
	W / O / G / CO_2 -rich
Residual saturations	0.40 / 0.20 / 0.05 / 0.05
Endpoint relative permeability model	0.35 / 0.50 / 0.65 / 0.65
Exponent	3.0 / 3.0 / 3.0 / 3.0

Table 5.4: BSB reservoir properties

The results obtained by [Okuno [2009]] are given in fig. 5.10. The simulations performed in this thesis were different; [Okuno [2009]] has not provided any volume shift for the density model of the water and the viscosity model. The number of cells has been increased in our simulations to improve the accuracy. Finally, there was no water in the simulation performed for this case. In fig. 5.11a, fig. 5.11b, fig. 5.11c, the results we obtained with MFlash are represented. Furthermore, except for a more important diffusion effect that can be noticed with the MFlash, the saturation profiles are similar to those obtained by [Okuno [2009]].

5.5.2.1.3 Results

By injecting gas into the reservoir, close to the injection well, the gas sweeps the oil in place. When getting closer to the front of saturations, the oil phase has not been totally swept yet and both phases are present. At this stage, the oil is not well displaced by the gas, because at these conditions both phases are not really miscible.

Figure 5.10: CO₂ injection process by [Okuno [2009]]Figure 5.11: CO₂ injection process, saturations obtained with AD-GPRS + MFLASH

The reservoir pressure is progressively increasing during the injection. Gas and Oil become miscible at these conditions and a CO₂-rich phase appears. The saturation of the oil phase becomes really low, while the CO₂ rich phase saturation is significantly increasing. This indicates an extraction of the oil components by the CO₂ rich phase. Some authors showed that the CO₂ injection increase the production due to the extraction of medium and heavy oil components by the CO₂ rich phase [Creek and Sheffield [1993]]. The density of the CO₂ rich phase is close to the density of the oil phase, which increases the miscibility of both phases and generally helps the extraction of the heavy components. In the same time, the viscosity of the CO₂-rich phase remains generally lower than the viscosity of the oil phase.

5.5.2.2 Four-phases, CO₂ injection

5.5.2.2.1 Stand-alone simulations

The phase equilibrium code developed in this thesis has been designed to work with any number of phases. In this subsection, a four-phase simulations of CO₂ injection is presented. By adding water into the BOB fluid, a four-phase system is obtained.

In this simulation, water is given a composition of $z_{water} = 0.1$, and the composition of the BOB oil is scaled to obtain $\sum_{i=1}^{nc} = 1$. The BIP $\delta_{water,i}$, $\forall i \neq w$ are set to 0.5. The water properties are given in table 5.5.

The p-z phase diagram of the BOB oil in presence of water, at T=313.71 K is given in fig. 5.12.

Feed	Molecular weight	Tc	p _C	w
0.1	18.00	647.35	221.0	0.3434

Table 5.5: BSB water properties

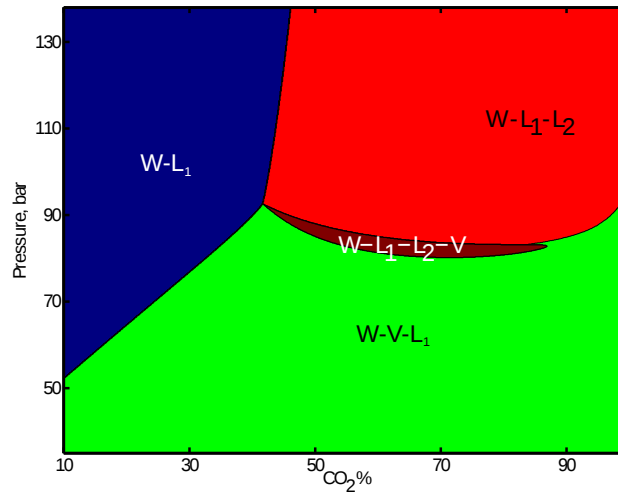
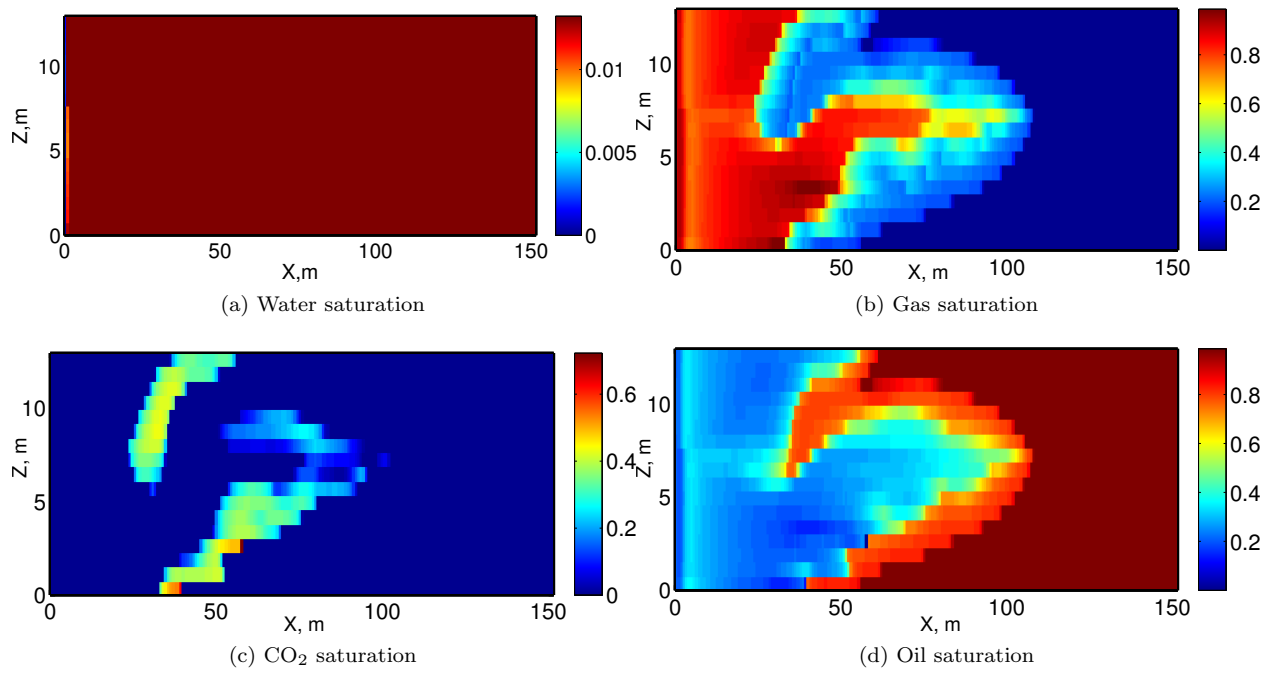


Figure 5.12: 4 phase case phase p-z diagram

5.5.2.2.2 Reservoir simulations

The same simulation test case as for the three-phase CO₂ injection problem is used here in presence of water.

In fig. 5.13a, fig. 5.13b, fig. 5.13c and fig. 5.13d are represented the saturations of water, gas, CO₂ and oil, respectively after 90 days of simulation. In fig. 5.14, the pressure is given for the same time.

Figure 5.13: Four-phase simulation of CO₂ injection

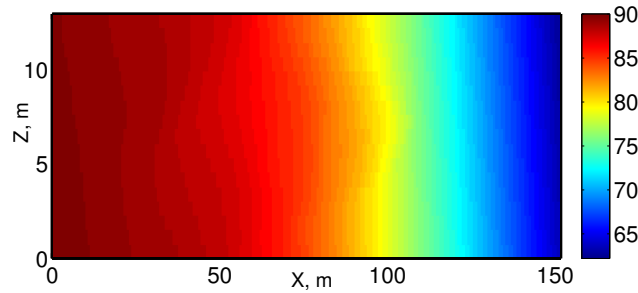


Figure 5.14: Pressure, bar; for the four phase CO₂ test case, T=50 days

The saturations are comparable to those obtained with the previous three-phase CO₂ injection case, yet, stronger diffusion effects take place this time. One of the possible explanations comes from the relative permeability model difference for three and four phases (see the relative permeability section). In this case, at low temperature, the water does not really mix with the other phases as can be seen in fig. 5.13a.

These results show that a four-phase simulation is possible with the developed equilibrium code. The flash successfully handles the four-phase case, providing a good regularity in the saturations (see fig. 5.13b, fig. 5.13c and fig. 5.13d)

5.5.3 Full thermal three-phase compositional simulations for steam injection

5.5.3.1 Light oil cases

5.5.3.1.1 Stand-alone simulations

In stand-alone, the code is tested for water-hydrocarbon mixtures, at different conditions. The first mixture contains three components: C₁, H₂O, nC₁₀, whose properties are given in table 5.6. fig. 5.15a and fig. 5.15b show ternary diagrams. Each system is represented with a different color. The phase transitions obtained by [Iranshahr et al. [2010b]] are represented in black squares. Good agreements are found with MFlash results. The phase transitions are also well modeled.

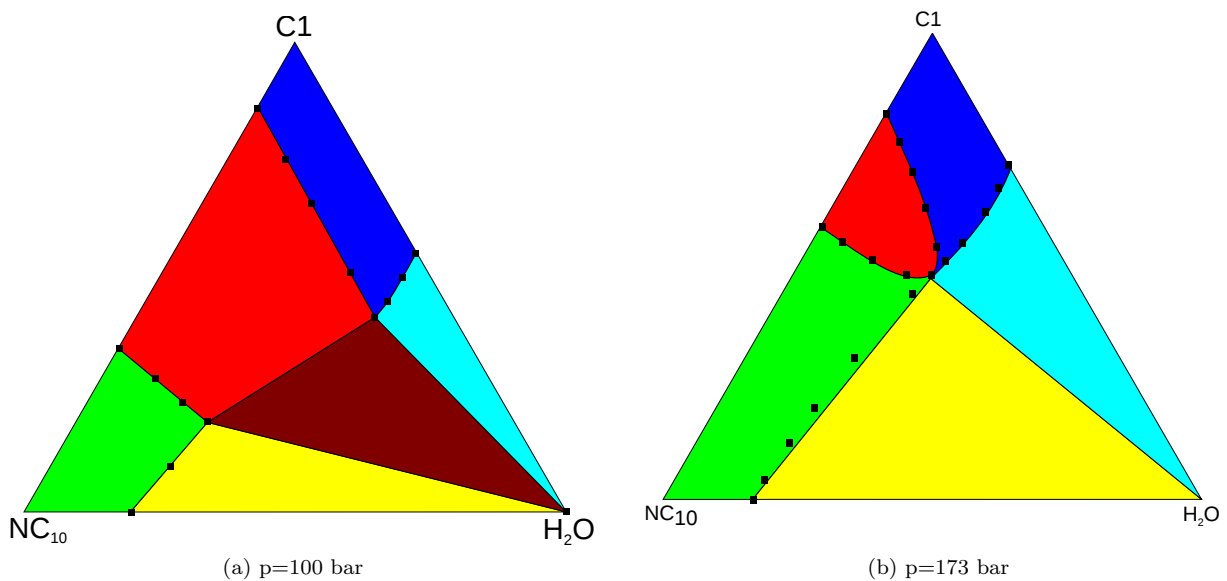


Figure 5.15: Three component mixture, at T=520 K

Comp	C ₁	C ₁₀	H ₂ O
T _c	190.6	617.6	647.3
P _c	45.4	21.08	220.48
w	0.008	0.490	0.344
M _w	16.043	142.29	18.015
δ_{i,H_2O}	0.4907	0.45	
$\delta_{i,C_{10}}$	0.0522		

Table 5.6: Fluid properties, light oil

5.5.3.1.2 Reservoir simulations

The composition of the light mixture is given in table 5.7.

Comp	CO ₂	C ₁₀	H ₂ O
z	0.01	0.44	0.3
T _c	304.20	618.50	647.37
P _c	73.80	21.23	221.20
w	0.224	0.484	0.345
M _w	44.010	142.29	18.015

Table 5.7: Fluid properties, light steam injection case

The reservoir consists of only one layer. In this example, a two dimensional simulation (xy) without gravity effects is considered. The reservoir is heterogeneous and the permeabilities are given in fig. 5.16. The remaining properties of the reservoir are given in table in 5.8.

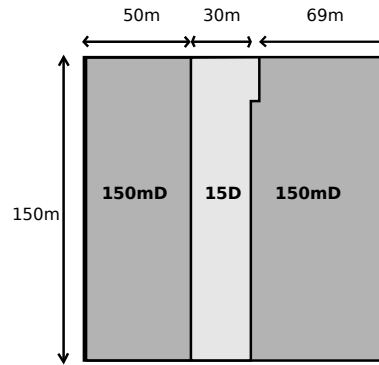


Figure 5.16: relative permeabilities for the light oil steam injection test case

Dimensions of a cell	5m × 5m × 1m
Number of grid cells	30 × 30 × 1
	W / G / O
Residual saturations	0.05 / 0.0 / 0.0
Endpoint relative permeability model	1 / 1 / 1
Exponent	2.0 / 3.0 / 2.0

Table 5.8: Light oil reservoir properties

The initial parameters are:

- reservoir pressure: 30 bars at a depth of 2750 m, then the pressure is initialized based on a hydrostatic equilibrium procedure.
- reservoir temperature: 290 K

Steam is injected at the bottom left-hand corner, with a production well located at the top right-hand corner. After 100 years of simulation, the saturations are given in fig. 5.18. The steam displaces the oil to the production well located at the top right hand corner. The steam chamber does not have the shape of a circle since the medium is heterogeneous. In the end, the breakthrough can be seen in fig. 5.18.

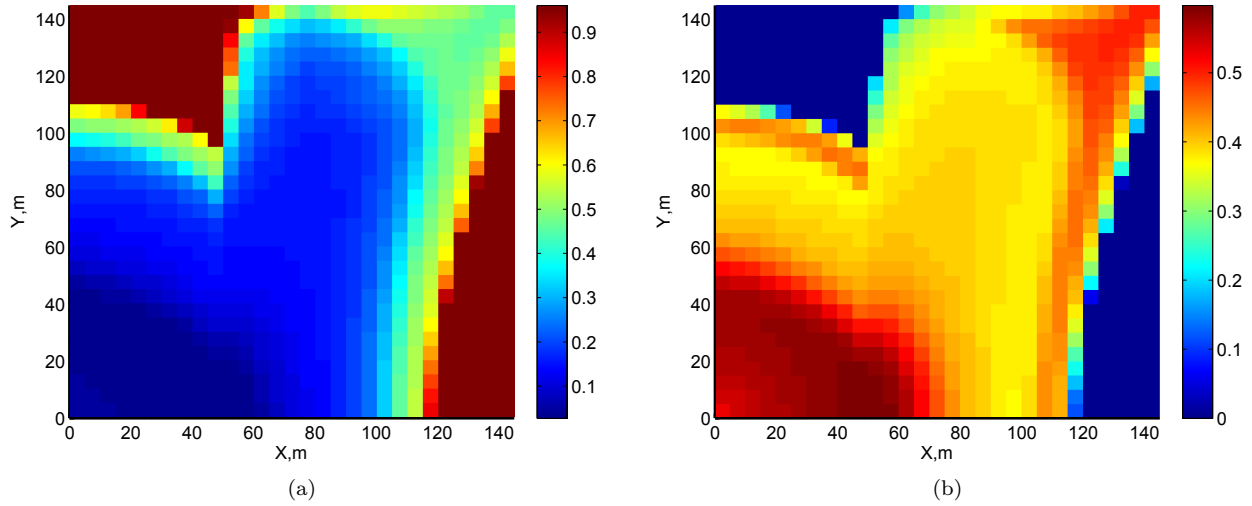


Figure 5.17: a)Oil Saturation, b)Gas Saturation for the light oil steam injection case

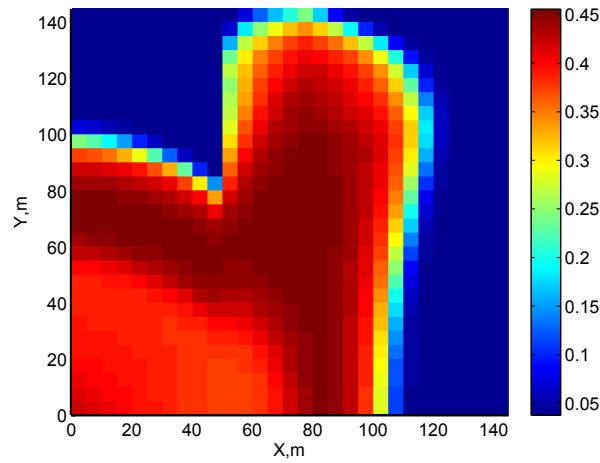


Figure 5.18: Water saturation

In this case, a comparison with [Lucia et al. [2000]]'s algorithm was done. The results are almost identical. The computation time (or CPU time) is around 5 times higher for this case with Lucia's algorithm as compared to the simulation with MFlash.

The fluid moves along the high permeability area (15D compare with 150mD for the other two pars) (fig. 5.17 and fig. 5.18). The steam sweeps the oil to the oil producer. After 100 days, the water breakthrough occurs. Some oil remains in the regions with small permeabilities.

5.5.3.2 Medium oil in a highly heterogeneous reservoir

A simulation with a more complex steam injection problem (the highly heterogeneous SPE10 reservoir test case) is performed. The mixture is a medium oil whose properties are defined in table 5.9. The reservoir

parameters are given in table 5.10.

Comp	CO2	C10	C16	H2O
z	0.01	0.44	0.25	0.3
Tc	304.20	618.50	722.59	647.37
Pc	73.80	21.23	14.04	221.20
w	0.224	0.484	0.717	0.345
Mw	44.010	142.29	226.432	18.015

Table 5.9: Fluid properties, medium steam injection case

The initial reservoir parameters are given below:

- reservoir pressure: 30 bars at a depth of (2750 m), then the remaining pressures are initialized based on hydro-static equilibrium of the reservoir.
- reservoir temperature: 290 K

Dimensions of a cell	6.096m \times 3.048m \times 0.6096m
Number of grid cells	60 \times 220 \times 3
	W / G / O
Residual saturations	0.0 / 0.0 / 0.0
Endpoint relative permeability model	1 / 1 / 1
Exponent	2.0 / 2.0 / 2.0

Table 5.10: SPE10 reservoir properties

All the wells are vertical. The production wells are located in the four corners of the domain and the injection well is located in the center of the reservoir.

Injection parameters:

- Injection of steam and CO₂: 90 % of water and 10 % of CO₂.
- Temperature of injection: 500 K.
- Pressure of injection: 50 bars.

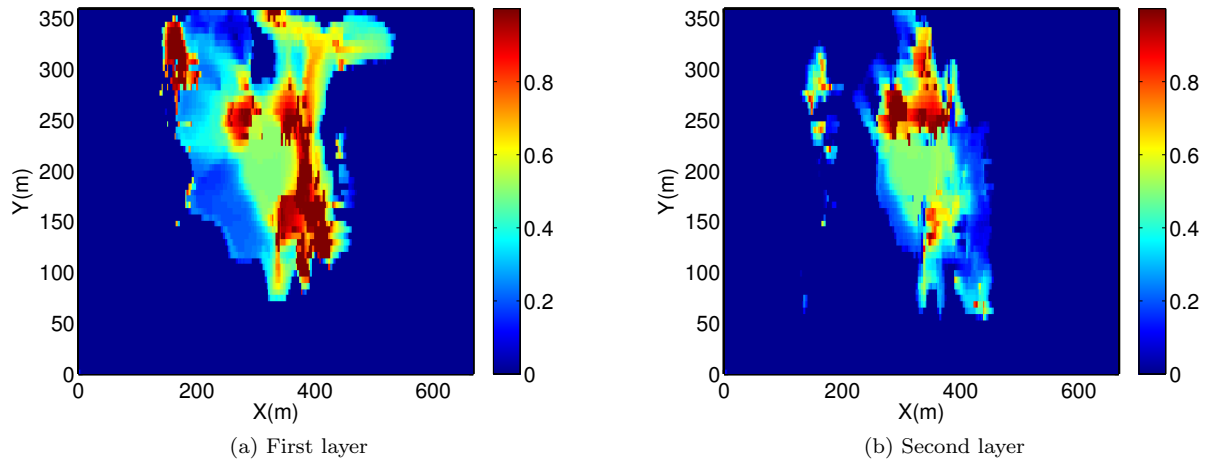
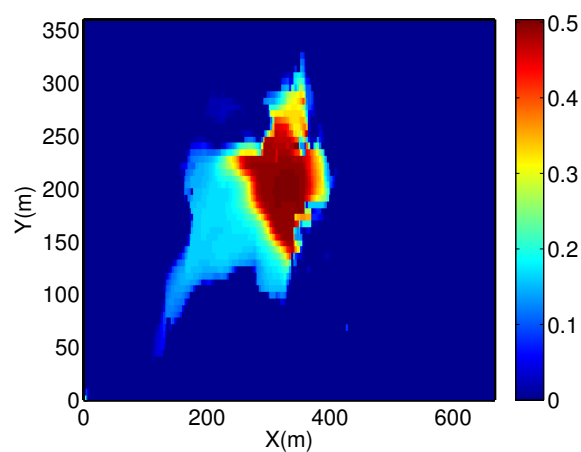
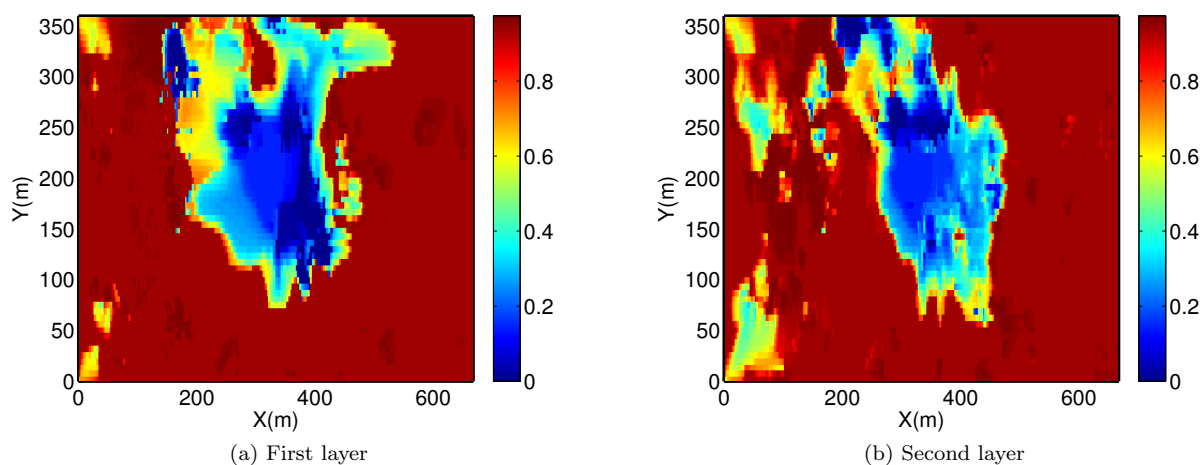
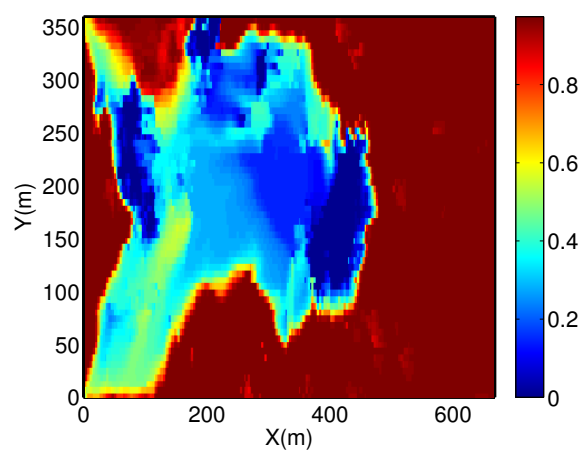


Figure 5.19: Gas saturation for the SPE10 case, at T=600 days

Figure 5.20: Gas saturation for the SPE10 case, third layer, $T=600$ daysFigure 5.21: Oil saturation for the SPE10 case, at $T=600$ daysFigure 5.22: Oil saturation for the SPE10 case, third layer, $T=600$ days

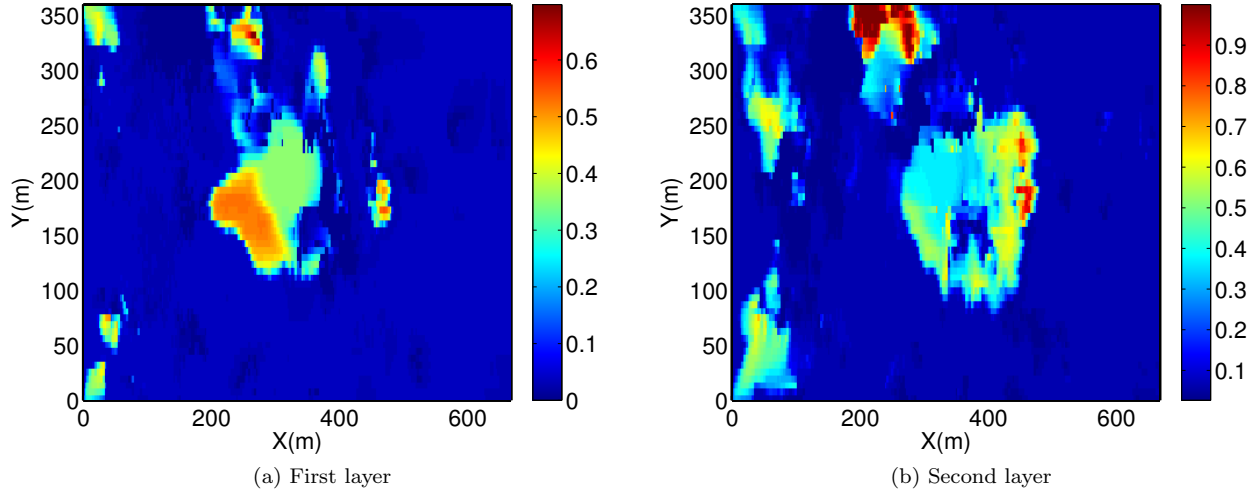


Figure 5.23: Water saturation for the SPE10 case at T=600 days

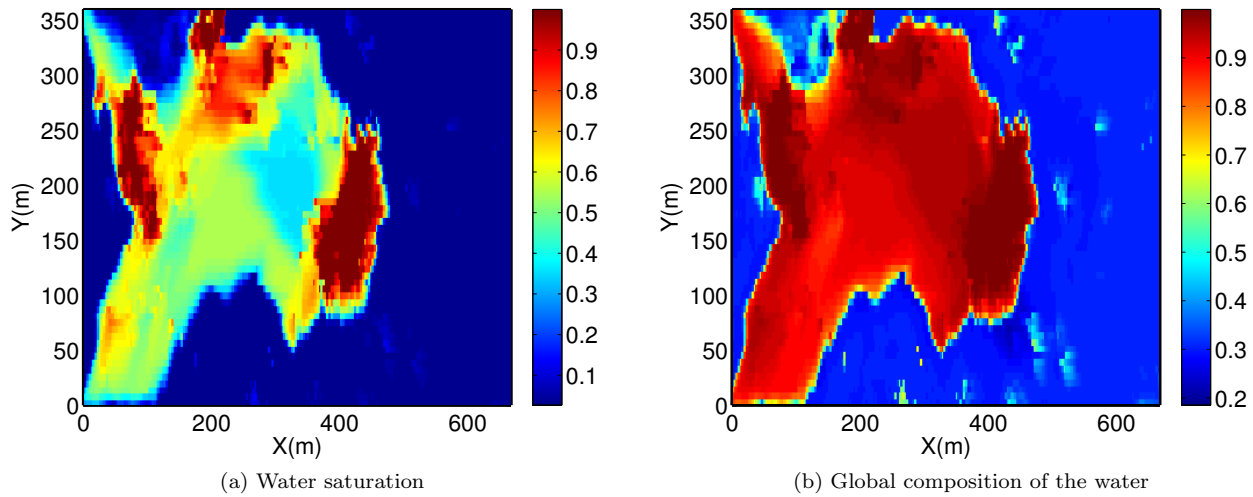


Figure 5.24: Plot of different properties in the third layer of the SPE10 case after T=600 days

Only one time step cut occurred during the whole simulation, revealing a robust flash, with a good phase identification. In fig. 5.19a, fig. 5.19b and fig. 5.20 are plotted the gas saturation for the three layers of the SPE10 cases, after 600 days of simulation. The saturations reflect the heterogeneities of the porous media. The water saturations are plotted in fig. 5.23a, fig. 5.23b and fig. 5.24a. Finally the oil saturations are shown in fig. 5.21a, fig. 5.21b and fig. 5.22.

The high permeability zone on the right hand side of the field act as a barrier and prevent the expansion of the steam in the right hand side. The steam and water flushes the oil to the two producers located in the left-hand side. The simulation is stopped with the water breakthrough. In fig. 5.24b, the global composition of the water component is plotted, which reveals the water expansion in the reservoir up to the two producers in the left-hand side of the reservoir.

During the steam injection in the reservoir, the high injection temperature decreases the viscosity of the oil and increases its mobility. The oil is swept more easily leading to the enhanced oil recovery. fig. 5.20 shows the high gas saturation around the injection well. Then, away from the injection well, the water and light components condense at the contact with the cold region. fig. 5.24a shows the higher saturation of water when the gas disappears.

Finally, the major volume of oil has been produced in the left part of the reservoir.

5.5.3.3 Extra-heavy oil

The full compositional simulations of the SAGD case became possible, with the recent developments in AD-GPRS (by R.Zaydullin and D.Voskov (Stanford University)) and by coupling the simulator with the MFlash program.

5.5.3.3.1 The SAGD process

Two horizontal wells are located one under the other (the injection well above). As steam is injected, a steam chamber is formed around the injection well. At the chamber edge, the water and the light oil components condense. Besides, the elevated temperature decreases the oil viscosity and the oil becomes mobile. The condensed water and the extra-heavy oil flows to the production well under the gravity.

Two simulations of the SAGD process have been performed. The first consists of a synthetic heavy oil, initially at $T=285$ K. The oil is not heated before the steam injection. In the second case, a real Athabasca bitumen has been modelled. In this case, the oil is pre-heated before the steam injection.

5.5.3.3.2 Synthetic heavy oil

The oil composition is given in table 5.11.

Comp	CO2	M0	M2	H2O
z	0.02	0.57	0.2	0.21
Tc	304.18	444.19	1073.15	647.35
Pc	73.80	36.01	8.10	221.0
w	0.23	0.18	0.90	0.3434
Mw	44.01	65.56	900.00	18.00

Table 5.11: Fluid properties, synthetic heavy oil

The initial parameters are:

- reservoir pressure: 8.11 bars at the depth 180 m, then the remaining pressure are initialized based on hydrostatic equilibrium of the reservoir.
- reservoir temperature: 285 K
- the initial oil viscosity oil viscosity at $T=285$ K is $8.86e^{+7}$ cP

The injection well is located at the depth 117.5m and the production well, 5 meters below.

Injection parameters are:

- fraction of steam (99.4 mol%) and $\text{CO}_2\text{-C}_1$ (0.6 mol%).
- injection temperature: 500 K
- injection pressure: 25 bars.

Only half of the steam chamber (SC) cross section is shown in fig. 5.25a. Each cell is a cube of length 0.5 m, the grid dimension is made by $101 \times 1 \times 52$.

fig. 5.25a, fig. 5.25b and fig. 5.26a represent the saturations of water, gas and oil respectively after 150 days of simulation. The temperature is given in fig. 5.26b for the same time. The SC develops with the characteristic triangular shape of the SAGD process. In fig. 5.25a, an accumulation of the water phase can be seen along the SC. Along the SC, the high temperature decreases the liquid viscosities, enabling the oil to flow to the production well under gravity drainage.

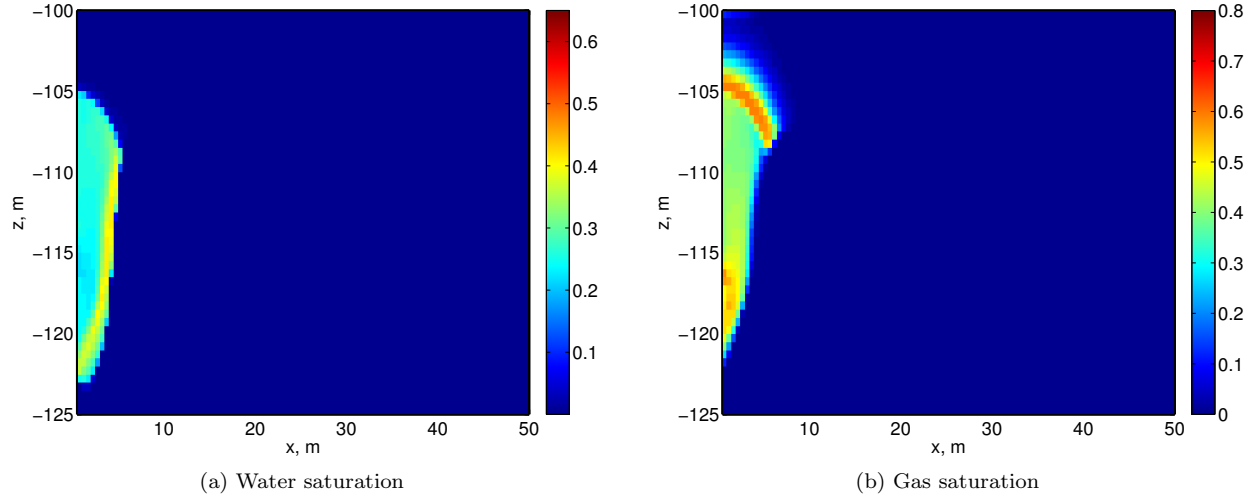


Figure 5.25: Saturations for the synthetic SAGD at $t=150$ days

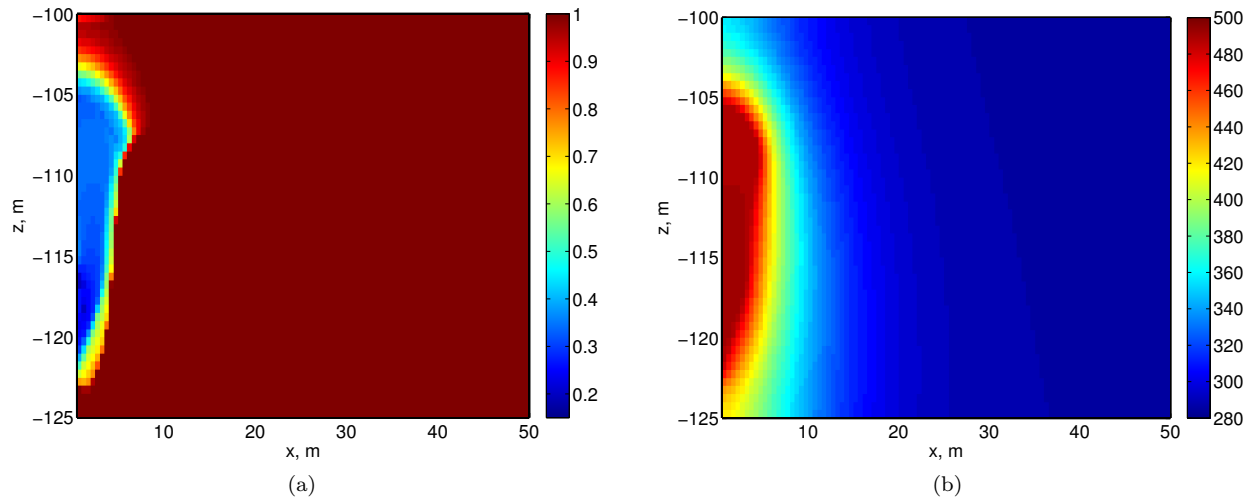


Figure 5.26: a) Oil saturation b) Temperature; for the synthetic SAGD at $t=150$ days

5.5.3.3.3 Athabasca bitumen

Comp	CO ₂	C ₁	M ₀	M ₁	M ₂	Asph	H ₂ O
z	0.001	0.03	0.007	0.52472	0.27769	0.10959	0.05
T _c	304.18	190.6	444.19	798.42	1073.15	1203.15	647.35
P _c	73.80	46.10	36.01	17.00	8.10	12.40	221.0
w	0.23	0.011	0.18	0.6941	0.90	0.95	0.3434
M _w	44.01	16.04	65.56	275.00	900.00	900.00	18.00

Table 5.12: Fluid properties, extra heavy oil

The oil used for SAGD modelling, is an Athabasca oil containing extra-heavy components. The fluid properties are given in table 5.12. The initial parameters and the grid are the same as for the synthetic heavy oil. At $T = 283K$, the initial oil viscosity is $\mu = 6.64e^{+11}$ cP. Steam is injected (99.5 mol%) with solvent (0.5 mol%). The solvent is the M0 pseudo-component made of C3-C10 components.

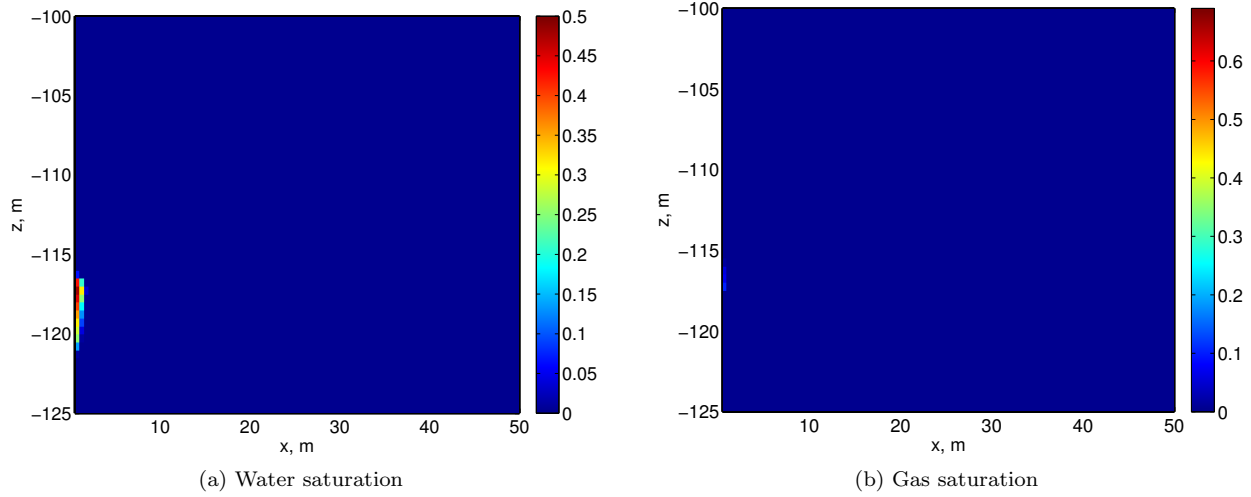


Figure 5.27: Saturations for the Athabasca oil SAGD at $t=20$ days

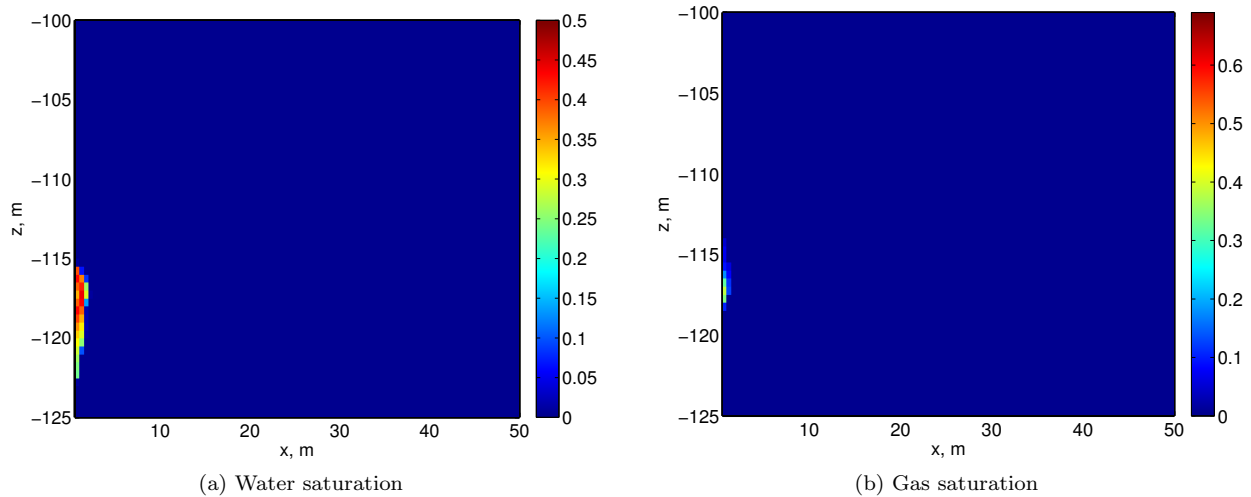


Figure 5.28: Saturations for the Athabasca oil SAGD at $t=30$ days

fig. 5.27a and fig. 5.27b show the saturation of the water and gas phase respectively at $t=20$ days. At the beginning, a connection between the two wells is made. No steam chamber is seen. At $t=30$ days, the same saturation fields are shown in fig. 5.28a and fig. 5.28b. This time corresponds to the established connection between the wells. The injected hot water reaches the producer. Now, the steam chamber is progressively expanding above the injection well.

fig. 5.29a, fig. 5.29b and fig. 5.30a show the saturation fields (of water/gas and oil respectively) at $t=360$ days. At this time, the SC has been developed, growing mainly vertically. Water and oil accumulation can be seen along the steam chamber edge. Both heated liquids are flowing to the production well by gravity drainage. Moreover, high concentration of gas can be seen at the top of the SC. The light components with a lower density move to the top of the SC and encounter the liquid phases (oil and water) at the edge of the chamber. fig. 5.30b shows the temperature profile in the reservoir for the same time.

Due to the gravity effects, the gas mixture moves to the top of the reservoir. Once the steam chamber

reaches the top, it starts growing horizontally.

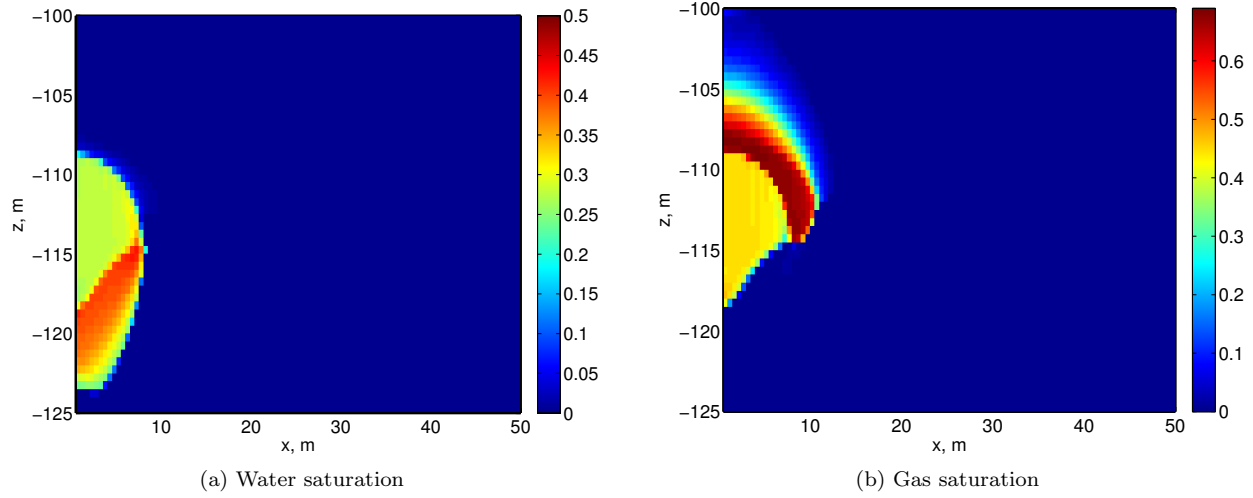


Figure 5.29: Saturations for the Athabasca oil SAGD at $t=360$ days

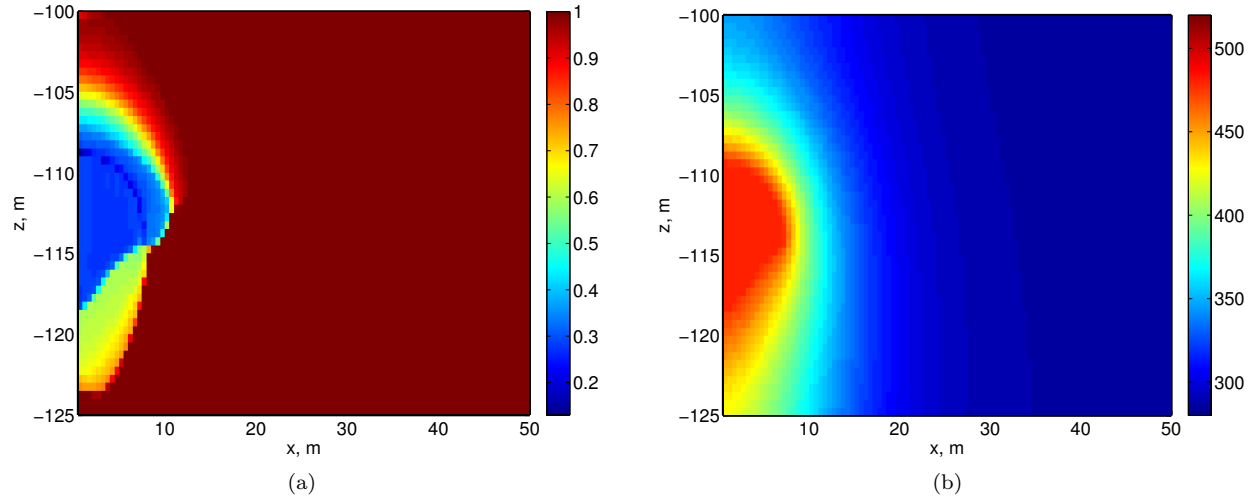


Figure 5.30: a) Oil saturation b) Temperature for the Athabasca oil SAGD at $t=360$ days

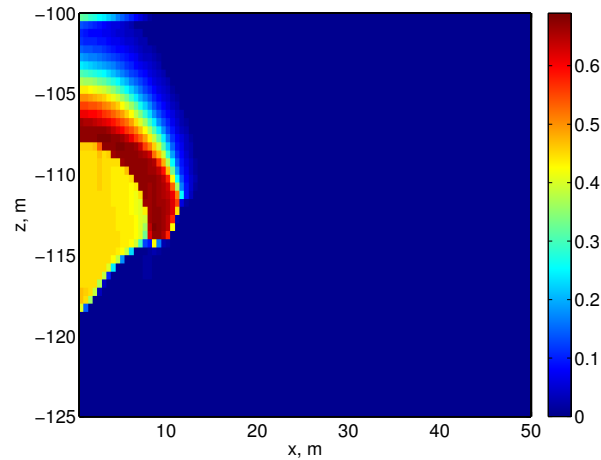


Figure 5.31: Gas saturation for the Athabasca oil SAGD at $t=410$ days

Finally, at $t=410$ days, the steam chamber reaches the top of the reservoir; fig. 5.31 shows the saturation of the gas phase at this moment.

5.6 Comparisons between different compositional acceleration procedures

Different acceleration techniques described in chapter 2, 3 and 5 were also tested with MFlash:

- the reduction variables (RV) for the stability analysis. We noticed in a previous chapter that the reduction was decreasing the computation time of the stability analysis. We propose here to test the variables in the reservoir simulation framework.
- the bypass method from [Rasmussen et al. [2006]] (labelled B-R) as presented in chapter 2.
- The three-phase bypass method (3PB) presented in section 5.4.

Those algorithms were compared with simulations using MFlash with conventional variables (labelled Conventional).

Computation times were tested for three different cases: the three-phase CO₂ injection case, the light oil steam injection case and the SPE10 steam injection case. Table 5.13 show the total time of simulation for the different algorithms. Table 5.14 gives the percentage of the equilibrium calculations in the total time of the simulation.

Test case	Conventional	RV	B-R	3PB
CO ₂ injection	1311.4085	1298	1297	-
light oil	25.0129	26.2679	26.0675	16.2
SPE10	11865	8166	8270	5185

Table 5.13: Total time of the simulation for the different cases (in seconds)

Test case	Conventional	RV	B-R	3PB
CO ₂ injection	71%	71 %	70%	-
light oil	45%	46%	46%	6%
SPE10	33%	41%	39%	8%

Table 5.14: Percentage of the equilibrium flash in the simulation CPU time for the different cases

Tables 5.14 and 5.13 reveal a different time for the CO₂ injection case and the steam injection cases. When dealing with CO₂ and liquid/liquid phase behavior, more initial guesses are needed than for steam injection problems. This is why the equilibrium computation represents 70 % of the total simulation for the CO₂ injection case, whereas it is around 40% for the steam injection test cases.

Tables 5.14 and 5.13 show that the use of the reduction method generally decreases the total time of simulation. It is particularly true for the SPE10 test case (8166s vs 11865s for conventional variables).

However, in this study, the use of the B-R method as developed in [Rasmussen et al. [2006]], in presence of water does not improve the computational time for the tested cases. Indeed, the water initial guess for the stability analysis often converges to a minimum which is different from the trivial solution and no phase equilibrium can be bypassed.

The use of the 3PB coupled with MFlash is the option leading to the smallest computation times for all the cases. The 3PB method uses negative flash procedure to compute supporting tie-simplex and uses multiphase calculations to adaptively parameterize the tie-simplex space. The results obtained with this method were identical to MFlash without any acceleration technique, and demonstrated a good robustness.

In terms of performances, 3PB re-uses the information from previous equilibrium calculations to pass by the computation of further stability analysis. Many stability tests can be avoided by using this procedure. Here, the use of 3PB improves the time of computation from 8 up to 12 times as compared with MFlash. For the whole simulation, with 3PB the averaged computation time is 1.6 times faster than with MFlash alone. However, using a flash based on global optimization, or with higher initial guesses, or with a more accurate EoS model, the differences may increase significantly.

To check the robustness of the 3PB, a comparison was performed for the synthetic oil SAGD test case. Table 5.15 gives the computational results obtained with MFlash with reduction variables and 3PB for 90 days of simulation. Taking the flash in stand-alone as the reference, the relative error on the oil production rate for the 3PB simulation remained always under 6%, taking large steps (system ill-conditioned where convergence problems occur). By taking smaller steps, the error becomes negligible. In this case, the total computation time with 3PB was much smaller than for MFlash: (2529s versus 3914s), decreasing the fraction of the equilibrium calculations from 45% to 10% of the total simulation. 3PB seems quite robust and capable of handling simulations of difficult processes quite efficiently.

	RV	3PB
Total simulation time	3914 s	2529 s
Equilibrium % of the total simulation	45	10

Table 5.15: Synthetic oil, SAGD computation time for 90 days of simulation

Note that, all the tests were carried out using only one core. In each cell, the equilibrium calculation does not depend on properties from other cells (local computations), therefore equilibrium calculations can be efficiently parallelized. Using a parallel procedure, the percentage of the total time spent in the equilibrium calculations would become much smaller, and the CPU time required for equilibrium calculations would represent a smaller proportion of the total simulation time.

Finally, analyzing all the simulations (except for CO₂ injection where more initial guesses are required), one can see that the time of the equilibrium calculations with or without the use of 3PB always remains under 50% of the total time, even for the SAGD test case.

5.7 Conclusion

In this chapter, a rigorous validation of the developed phase equilibrium calculations program has been performed.

Different phase diagrams obtained with the developed equilibrium program showed good agreements with the experimental and literature data. The good reproduction of various phase diagrams (LLV-LLLV) illustrates the capability of the program to predict the true phase behavior of such systems. As stand-alone comparisons can be made for some specific conditions, in order to guarantee the good functionality of the equilibrium algorithm for a wide variety of conditions, different reservoir simulations were performed.

Reservoir simulations cover a great variation of p-T-z conditions. Phase transitions involve passing across phase boundaries and/or those supercritical conditions. Any failure in the convergence to the correct minimum can lead to the divergence of a simulation. The capability to carry out correctly a whole simulation demonstrates the robustness of the code.

The developed program was first plugged with TPP, to simulate an isothermal water injection process. The same simulation was also performed with the ECLIPSE simulator by Schlumberger and the same production rates were obtained for both simulations. The ECLIPSE simulator has been tested against a variety of

experimental cases, and is currently used in production division in Total. The good agreement in the results indicates that the developed algorithm converged to the correct minimum of the Gibbs energy during the whole simulation.

Another test was performed with a full three phase CO₂ injection problem. Comparisons were made with [Okuno [2009]] which also showed similar saturation profiles. The simulation ran up to the CO₂ breakthrough (450 days). After adding water to the model, the same simulation was carried out and led to four-phase systems at some conditions. With a high number of phases, the difference in the Gibbs free energy (ΔG) between a system with np phase becomes really small as compared to the system with $np + 1$ phases. Robust stability analysis codes have to be designed to handle the high number of phases. In this work, a four-phase simulation has been performed during 100 days. This indicates the capability of the algorithm to deal with small ΔG , and illustrates the robustness of the stability analysis algorithm.

Simulations of steam injection cases were also performed. The highly heterogeneous SPE10 reservoir was used to perform a 3D steam flooding simulation up to the water breakthrough. With three dimensional heterogeneous reservoirs, more thermodynamic paths are accessed during a simulation. Once again, this example reveals the robustness of the developed algorithm.

Moreover, a fully compositional reservoir simulation of the SAGD process was performed with a real extra heavy oil mixture. To the best of our knowledge, this is the first simulation of the kind in the literature. Thermal compositional simulations are quite complex because the important change in the conditions is observed (composition, pressure and temperature). Indeed, for the steam injection process, the steam distillation and the condensation of the water and the light oil components can be observed. With heavy oils, the gravity creates concentration gradients which imply many changes in the oil composition.

When injecting steam, the steam flushes progressively all the other components. The phase envelope of the resulting composition becomes narrow and the multiphase problem approaches ill-conditioning where a small variation in the conditions can create an important change in the equilibrium state. A simulation of the SAGD process has been performed up to 410 days revealing the capability of the algorithm to converge even for difficult (nearly ill-conditioned) problems. The presented simulations show that the developed code can overcome these difficulties.

The proposed thermodynamic code was also used with different acceleration procedures, namely, based on the bypass method by [Rasmussen et al. [2006]] and the three-phase bypass by [Zaydullin et al. [2013]]. Without any acceleration technique, the program has taken around 45% of the total CPU time for steam injection problems. In presence of water for multiphase systems, the bypass method did not perform really well. However, with 3PB, the CPU time was decreased and reached around 10% of the total time, keeping the same solution as the thermodynamic program alone.

Last but not least, one of the main objective of the thesis was to perform simulations of steam injection problems within a CPU time for equilibrium calculations under 50 % of the total simulation time. This objective has been achieved.

CHAPTER 6

Conclusions and Perspectives

1 Conclusions

In this thesis, improvements in equilibrium algorithms made possible the realization of fully compositional simulations of steam injection with heavy oils.

Most of the phase equilibrium calculations are performed based on the Newton-Raphson method and use conventional independent variables. These algorithms were investigated and different extensions were proposed. The logarithm of the equilibrium constants ($\ln \mathbf{K}$) is the best choice of conventional independent variables for multiphase equilibrium calculations. This set of variables lead to the best condition number and Newton methods based on $\ln \mathbf{K}$ converge in fewer iterations than the other Newton algorithms. In this work, extending [Michelsen [1982a]]’s method to multiphase-split problem, we proposed to compute the Jacobian matrix as a product of two symmetric matrices, and to solve the linear system using a Cholesky factorization, which proved to reduce the computation time by about a third in most of the calculations.

Initiated by [Michelsen [1986] and Hendriks [1988]], reduction methods have been investigated in the literature and proved to be highly efficient. In this work, a new reduction method for stability testing and phase-split problems was presented. It is based on the multi-linear expression of the logarithms of fugacity coefficients as functions of the coefficients \mathbf{h} (which are taken as independent variables and are unbounded). The reduction parameters are dependent variables, which, unlike in previous formulations, are guaranteed to be within their bounds. The dimensionality of the problem depends only on the number of components having non-zero BIPs with the remaining ones, and not on the number of components in the mixture, as in the conventional approach. The proposed method has a large convergence radius and Newton iterations can be used directly for the stability analysis problem (without the need of using successive substitutions), it has the same convergence path as the conventional approach using the natural logarithm of formal mole numbers (for stability analysis) and of the equilibrium constant (for phase-split calculations) as independent variables. This method was proved to be the best Newton method for the stability analysis problem (in terms of efficiency and condition number).

Comparisons between different conventional and reduction methods were performed for the stability testing and the multiphase-split calculation problems. Until now, the comparisons from the literature have focused on the global CPU times to compute the whole multiphase equilibrium problem. In this work, comparisons were performed independently for each problem (stability analysis, two phase-split and multiphase-split calculations). Besides, the condition number and convergence path were also investigated, leading to more detailed and extensive comparisons and new conclusions:

- For the stability testing problem, in reservoir simulation purposes, the proposed reduction method is the most efficient, whereas for the multiphase-split problem, the conventional methods require less CPU times.
- For chemical processes and any problems involving a large number of components, the use of the proposed reduction method is extremely time earning as compared to conventional Newton methods.

Furthermore, we proposed a Trust-Region method to handle both stability and phase-split problems. The Trust-Region (TR) can be viewed as an 'extension' of the Newton method. TR can reach supra-linear steps even when the Hessian is non-positive definite, and quadratic steps otherwise by adding a diagonal element to the matrix. An hybrid algorithm was presented. Instead of performing TR iterations immediately after the switch from the first-order SSI method, Newton iterations are first tested, and the TR method is applied only if the objective function value is increasing between two consecutive iterations. This hybrid methodology remarkably decreased the computation time of the method keeping the the high robustness of the TR method. The tests were performed covering whole multiphase-regions and some calculations were successfully performed at really difficult conditions, such as the immediate vicinity of bi-critical points.

Moreover, a new methodology has been developed to find variables leading to a well-scaled Hessian in the BFGS methods. This procedure leads to the α variable introduced by [Michelsen [1982b]] for the stability analysis and to a new variable for phase-split problems. A quasi-Newton strategy with a safeguarding switch back procedure (when the Gibbs energy increases between two iterations) was developed based on this variable and proved to reach convergence even close to phase boundaries where algorithms often encounter convergence problems. The algorithm compared favorably (in terms of iterations, function evaluations and robustness) with Nocedal's LBFGS-B code and existing BFGS implementations, for different mixtures. Only few additional function evaluations are necessary as compared with a Newton method, which makes it an attractive method. This procedure is extremely suited for more complex EOS models.

Combining all the developed algorithms, a general equilibrium calculation program was developed. It has been designed to handle any number of phases. Tests were performed on various systems and comparisons with numerical examples from other authors, and experimental data, have validated the proposed approach.

In addition to improve the robustness and efficiency of the equilibrium algorithms, a new methodology to generate pseudo-components giving an accurate representation of a mixture has been developed. The methodology is based on the quadrature method of moments (QMoM) which is here applied to multiphase equilibrium calculations for actual oil mixtures (combined with non-hydrocarbon components, such as water and carbon dioxide) using a cubic equations of state with non-zero BIPs. The quadrature is solved using a procedure (ORTHOPOLE) which avoids problems due to the ill-conditioned nature of the problem and is suitable for an undetermined number of quadrature points (unlike the original formulation of the QMoM).

The QMoM relates the discrete initial detailed composition having a large number of components to a small number of components which approximates the continuous portion of the mixture. It can be viewed in a larger sense as a lumping procedure. It is generally applicable, even in cases when no standard distribution function can model the feed composition, or several distribution functions are needed to model different portions of the mixture or different homologous series. It is proven in this work that this methodology coupled with a delumping procedure and using a cubic equation of state with non-zero BIPs works for actual oil mixtures with good accuracy.

The QMoM in our implementation has been tested for different initial compositions of a hydrocarbon mixture for two-phase equilibrium calculations and in the presence of water and carbon dioxide, for three-phase flash calculations. In all cases, the location of the phase boundaries, the phase distributions and the phase compositions were well reproduced by the semi-continuous description (however only 10 pseudo-components in average were necessary). In the reservoir simulation framework this tool appear essential since the computer time per time step is proportional to the power three of the number of equations to be solved simultaneously [Burger et al. [1985]] (and each component leads to one more equation).

Subsequently, the developed equilibrium calculation program has been implemented within two different reservoir simulators: TPP a reservoir simulator developed in Total S.A. and AD-GPRS simulator from the

University of Stanford.

For three-phase water injection problems, the exact same production and injection rate curves for the oil, gas and water phases were obtained with the ECLIPSE reservoir simulator and the developed equilibrium calculation program plugged in TPP. This reveals the capability of the equilibrium program to converge to the correct equilibrium solution. Also, simulations of CO₂ injection cases were presented (known to be difficult). Three- and four-phase isothermal compositional reservoir simulations were performed up to the CO₂ breakthrough. The simulations showed the capability of the code to handle more than three phases, which can be highly difficult due to the small differences of the Gibbs energy between three-phase and four-phase systems.

Simulations of steam injection cases were also performed. The highly heterogeneous SPE10 reservoir was used to test a 3D steam flooding simulation up to the water breakthrough. With three dimensional heterogeneous reservoirs, more thermodynamic paths are accessed during a simulation. Once again, this example reveals the robustness of the developed algorithm.

Finally, a fully compositional reservoir simulation of a SAGD process was realized with an extra heavy oil mixture. To the best of our knowledge, this is the first simulation of the kind in the literature (with a complete modeling of the water phase within equilibrium calculations). Thermal compositional simulations are quite complex because important changes in the conditions (composition, pressure and temperature) are observed. Moreover, the gravity creates concentration gradients which imply many compositional changes.

The developed code was also used in parallel with different acceleration methods such as the three-phase bypass algorithm (by [Zaydullin et al. [2013]]). Fully thermal compositional three-phase reservoir simulations of the steam flooding process could then be carried out with the equilibrium calculation time representing less than 10% of the total CPU time. Last but not least, one of the main objective of the thesis was to reduce under 50% the computational time for equilibrium flash calculations within simulations of steam injection problems. This objective has been achieved.

Through this thesis, three papers have been already published [Nichita and Petitfrere [2013], Petitfrere and Nichita [2014a], Petitfrere and Nichita [2014b]]; a fourth paper was submitted (under review) [Petitfrere et al. [2014]] and three other papers are in preperation and will be submitted shortly (corresponding to the subsections 3.2.4, 3.3 and 3.5).

2 Perspectives

Through the proposed work, new perspectives appear. In this thesis, new minimization algorithms were successfully implemented such as a Trust-Region and a quasi-Newton method. These algorithms are independent of the EoS model. Procedures leading to a smaller number of iterations become more and more efficient as the complexity increases (because computing the fugacities and fugacity derivatives becomes more and more time-consuming). Therefore, both quasi-Newton and Trust-Region methods would perform more efficiently when using complex model as compared with Newton methods. The Trust-region because the method converges in less iterations and the BFGS because no derivatives are needed in the Hessian approximation.

Furthermore, the BFGS procedure has been developed for the two-phase split problem and an extension to the multiphase-split problem is intended. With more phases, larger systems have to be solved at each Newton iterations and the BFGS method (which does not solve explicitly linear systems) would perform more and more efficiently as compared with Newton methods.

New perspective also appear for the compositional reservoir simulation. The developed equilibrium

program was shown to be robust and efficient and could be used for other processes. Adding a source term in the mass balance equations and in the equation of energy, irreversible chemical reactions (based on activation energies) could also be modeled. Simulations of the in-situ upgrading process were performed using this methodology [Lapene [2010]].

Different acceleration techniques were presented and tested. One of them, the three-phase bypass [Zaydullin et al. [2013]] presents interesting features such as the parameterization of the space. The method could be used to help phase identifications. When two liquid phases are present, it is sometimes difficult to identify one another. The parameterization enables a clear identification in the whole compositional space, thus avoiding discontinuities.

[Gaganis and Varotsis [2012]] proposed another acceleration method that seems to be efficient in the framework of compositional reservoir simulation of complex processes (such as CO₂ injection). Using machine-learning methods, they train the system to generate discriminating functions capable of providing the sign of the minimum tangent plane distance within only few operations. It would be interesting to compare the method against the three-phase bypass method.

Last but not least, the presented simulations could be improved using more accurate models. It is well known that volumetric parameters and most of the derivatives properties are not modeled correctly based on cubic equations of state. More suited equations of state could be used to perform simulations of steam injection processes.

APPENDIX

Reduction variables

A Elements of the Jacobian matrix/partial derivatives for the reduced stability analysis

The partial derivatives required in (eq. 3.84) are (for $\alpha, \gamma = 1, m$)

$$\left(\frac{\partial h_\alpha}{\partial Q_\gamma}\right)_{Z, Q_{\mu \neq \gamma}} = \delta_{\alpha\gamma} \frac{h_\alpha}{Q_\alpha} \quad (\text{A.1})$$

$$\left(\frac{\partial h_\alpha}{\partial Q_M}\right)_{Z, Q_{\mu \neq M}} = -\frac{1}{B} \left(h_\alpha + 2\lambda_\alpha Q_\alpha \frac{Z}{\pi}\right) \quad (\text{A.2})$$

$$\left(\frac{\partial h_M}{\partial Q_\gamma}\right)_{Z, Q_{\mu \neq \gamma}} = \frac{h_\gamma}{Q_M} \quad (\text{A.3})$$

(eq. A.1 and eq. A.3) correct misprints in [Nichita and Graciaa [2011]].

$$\left(\frac{\partial h_M}{\partial Q_M}\right)_{Z, Q_{\mu \neq M}} = \frac{1}{B^2} \left[-2h_M B - 1 + Z \left(1 + \frac{A}{\pi}\right)\right] \quad (\text{A.4})$$

$$\left(\frac{\partial h_{M+1}}{\partial Q_\gamma}\right)_{Z, Q_{\mu \neq \gamma}} = 0 \quad (\text{A.5})$$

$$\left(\frac{\partial h_{M+1}}{\partial Q_M}\right)_{Z, Q_{\mu \neq M}} = \frac{1}{Z - B} \quad (\text{A.6})$$

and

$$\left(\frac{\partial h_\alpha}{\partial Z}\right)_Q = \frac{2\lambda_\alpha Q_\alpha}{\pi} \quad (\text{A.7})$$

$$\left(\frac{\partial h_M}{\partial Z}\right)_Q = \frac{1}{B} \left(1 - \frac{A}{\pi}\right) \quad (\text{A.8})$$

$$\left(\frac{\partial h_{M+1}}{\partial Z}\right)_Q = -\frac{1}{Z - B} \quad (\text{A.9})$$

where $\pi = (Z + \delta_1 B)(Z + \delta_2 B)$. One can further substitute $\frac{A}{\pi} = \frac{1}{Z-B} - 1$ in some of the above equations.

The partial derivatives of the compressibility factor (the phase index is dropped for simplicity) with respect to the reduced variables are:

$$\frac{\partial Z}{\partial Q_\beta} = \frac{\partial Z}{\partial A} \frac{\partial A}{\partial Q_\beta}; \beta = 1, m \quad (\text{A.10})$$

with

$$\frac{\partial A}{\partial Q_\beta} = 2\lambda_\beta Q_\beta; \beta = 1, m \quad (\text{A.11})$$

$$\frac{\partial Z}{\partial A} = - \left(\frac{\partial F}{\partial A} \right)_{Z,B} / \left(\frac{\partial F}{\partial Z} \right)_{A,B} \quad (\text{A.12})$$

$$\frac{\partial Z}{\partial Q_M} \equiv \frac{\partial Z}{\partial B} = - \left(\frac{\partial F}{\partial B} \right)_{Z,A} / \left(\frac{\partial F}{\partial Z} \right)_{A,B} \quad (\text{A.13})$$

where F is the implicit expression of the EoS (left-hand side of (eq. 1.71)) and

$$\left(\frac{\partial F}{\partial A} \right)_{Z,B} = Z - B \quad (\text{A.14})$$

$$\left(\frac{\partial F}{\partial B} \right)_{Z,A} = (\delta_1 + \delta_2 - 1) Z^2 + [2B\delta_1\delta_2 - (\delta_1 + \delta_2)(2B + 1)] Z - A - \delta_1\delta_2(3B + 2)B \quad (\text{A.15})$$

and

$$\left(\frac{\partial F}{\partial Z} \right)_Q = 3Z^2 + 2[(\delta_1 + \delta_2 - 1)B - 1]Z + A + \delta_1\delta_2B^2 - (\delta_1 + \delta_2)B(B + 1) \quad (\text{A.16})$$

B Minimization of the modified TPD function in the reduction method

Michelsens modified TPD function

$$D^*(\mathbf{Y}) = 1 - Y_T + \sum_{i=1}^{nc} Y_i (\ln Y_i - \ln z_i) + \sum_{i=1}^{nc} Y_i [\ln \phi_i(\mathbf{Y}) - \ln \phi_i(\mathbf{z})] \quad (\text{B.1})$$

can be written as

$$D^* = D_I^* + D_E^* \quad (\text{B.2})$$

where subscripts I and E denotes the ideal and excess terms, respectively.

Let us define the vector $\overline{\mathbf{Q}} = (\overline{Q}_1, \dots, \overline{Q}_M, Y_T)^T$ of modified reduction parameters, with elements

$$\overline{Q}_\alpha = \sum_{i=1}^{N_C} q_{\alpha i} Y_i; \quad \alpha = 1, M \quad (\text{B.3})$$

with $\overline{Q}_\alpha = Y_T Q_\alpha$, since $Y_i = Y_T x_i$, and

$$\overline{Q}_{M+1} = Y_T = \sum_{i=1}^{N_C} q_{M+1,i} Y_i \quad (\text{B.4})$$

The objective function D^* can also be written as

$$D^*(\mathbf{Y}, \overline{\mathbf{Q}}) = D_I^*(\mathbf{Y}) + D_E^*(\overline{\mathbf{Q}}) \quad (\text{B.5})$$

that is, D_I^* depends only on \mathbf{Y}

$$D_I^*(\mathbf{Y}) = 1 - \sum_{i=1}^{nc} Y_i + \sum_{i=1}^{nc} Y_i (\ln Y_i - \ln z_i) \quad (\text{B.6})$$

and D_E^* depends only on $\overline{\mathbf{Q}}$

$$D_E^*(\bar{\mathbf{Q}}) = Y_T \sum_{i=1}^{nc} x_i [\ln \phi_i(\mathbf{Y}) - \ln \phi_i(\mathbf{z})] = \bar{Q}_{M+1} [g_E(\bar{\mathbf{Q}}) - g_{Ez}(\bar{\mathbf{Q}})] \quad (\text{B.7})$$

where the molar excess Gibbs free energy is a function of reduction parameters, that is, $g_E = g_E(\mathbf{Q}) = g_E(\bar{\mathbf{Q}}/\bar{Q}_{M+1}) = g_E(\bar{\mathbf{Q}})$.

The objective function $D^*(\mathbf{Y}, \bar{\mathbf{Q}})$ is minimized subject to the constraints

$$\sum_{i=1}^{nc} q_{\alpha i} Y_i - \bar{Q}_\alpha = 0; \alpha = 1, M \quad (\text{B.8})$$

and

$$\sum_{i=1}^{nc} Y_i - \bar{Q}_{M+1} = 0 \quad (\text{B.9})$$

The Lagrangian function is defined as

$$L(\mathbf{Y}, \bar{\mathbf{Q}}, \mathbf{h}) = D^*(\mathbf{Y}, \bar{\mathbf{Q}}) - \sum_{\alpha=1}^M \bar{h}_\alpha \left(\sum_{i=1}^{nc} q_{\alpha i} Y_i - \bar{Q}_\alpha \right) - \bar{h}_{M+1} \left(\sum_{i=1}^{nc} Y_i - Y_T \right) \quad (\text{B.10})$$

where $\bar{\mathbf{h}} = (\bar{h}_1, \dots, \bar{h}_M, \bar{h}_{M+1})^T$ is the vector of Lagrange multipliers.

From the theory of constrained optimization it follows that

$$\left(\frac{\partial L}{\partial Y_i} \right)_{\mathbf{Q}, Y_{j \neq i}} = \left(\frac{\partial D_I^*}{\partial Y_i} \right)_{\mathbf{Q}, Y_{j \neq i}} - \sum_{\alpha=1}^M \bar{h}_\alpha q_{\alpha i} - \bar{h}_{M+1} = \ln \bar{K}_i - \sum_{\alpha=1}^{M+1} \bar{h}_\alpha q_{\alpha i} = 0; i = 1, nc \quad (\text{B.11})$$

since $\left(\frac{\partial D_I^*}{\partial Y_i} \right)_{\mathbf{Q}, Y_{j \neq i}} = \ln Y_i - \ln z_i = \ln \bar{K}_i$, and

$$\left(\frac{\partial L}{\partial \bar{Q}_\alpha} \right)_{\mathbf{Y}, \bar{\mathbf{Q}}_{\beta \neq \alpha}} = \bar{h}_\alpha + \frac{\partial D_E^*}{\partial \bar{Q}_\alpha} = 0; \alpha = 1, M+1 \quad (\text{B.12})$$

On the other hand

$$\frac{\partial D_I^*}{\partial Y_i} = \sum_{\alpha=1}^{M+1} \frac{\partial D_I^*}{\partial \bar{Q}_\alpha} \frac{\partial \bar{Q}_\alpha}{\partial Y_i} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \frac{\partial D_I^*}{\partial \bar{Q}_\alpha}; i = 1, nc \quad (\text{B.13})$$

where we have used

$$\frac{\partial \bar{Q}_\alpha}{\partial Y_i} = q_{\alpha i}; \alpha = M+1, i = 1, nc \quad (\text{B.14})$$

from (eq. B.3, eq. B.4).

By identification, we obtain from (eq. B.11 and eq. B.13)

$$\frac{\partial D_I^*}{\partial \bar{Q}_\alpha} = \bar{h}_\alpha; \alpha = 1, M+1 \quad (\text{B.15})$$

The partial derivative of the excess part of D^* with respect to modified reduction parameters is

$$\frac{\partial D_E^*}{\partial \bar{Q}_\alpha} = \frac{1}{Y_T} \frac{\partial D_E^*}{\partial Q_\alpha} = \frac{\partial g_E}{\partial Q_\alpha} - \frac{\partial g_{Ez}}{\partial Q_\alpha} = h_\alpha(\mathbf{Q}) - h_{\alpha z} = \bar{h}_\alpha(\mathbf{Q}); \alpha = 1, M \quad (\text{B.16})$$

from (eq. 3.59), and

$$\frac{\partial D_E^*}{\partial \bar{Q}_{M+1}} = g_E - g_{Ez} - \sum_{\alpha=1}^M Q_\alpha \frac{\partial g_E}{\partial Q_\alpha} + \sum_{\alpha=1}^M Q_\alpha \frac{\partial g_{Ez}}{\partial Q_\alpha} = h_{M+1}(\mathbf{Q}) - h_{M+1,z} = \bar{h}_{M+1}(\mathbf{Q}) \quad (\text{B.17})$$

$$\text{because } g_E - \sum_{\alpha=1}^M Q_\alpha \frac{\partial g_E}{\partial Q_\alpha} = \sum_{\alpha=1}^{M+1} Q_\alpha h_\alpha - \sum_{\alpha=1}^M Q_\alpha h_\alpha = h_{M+1}(\mathbf{Q}).$$

The gradient in the reduction method is (from eq. B.15 and eq. B.16, eq. B.17)

$$g_\alpha^R = \frac{\partial D^*}{\partial \bar{Q}_\alpha} = \frac{\partial D_I^*}{\partial \bar{Q}_\alpha} + \frac{\partial D_E^*}{\partial \bar{Q}_\alpha} = \bar{h}_\alpha + \bar{h}_\alpha(\mathbf{Q}); \alpha = 1, M+1 \quad (\text{B.18})$$

and the Newton iteration equation is

$$\mathbf{H}^R \Delta \bar{\mathbf{Q}} = -\mathbf{g}^R \quad (\text{B.19})$$

The Hessian matrix in the reduction method is

$$H_{\alpha\beta}^R = \frac{\partial g_\alpha^R}{\partial \bar{Q}_\beta} = \frac{\partial^2 D^*}{\partial \bar{Q}_\alpha \partial \bar{Q}_\beta}; \alpha, \beta = 1, M+1 \quad (\text{B.20})$$

or

$$\mathbf{H}^R = \mathbf{H}_I^R + \mathbf{H}_E^R \quad (\text{B.21})$$

The ideal part of the Hessian is

$$[\mathbf{H}_I^R]_{\alpha\beta} = \frac{\partial^2 D_I^*}{\partial \bar{Q}_\alpha \partial \bar{Q}_\beta} = \frac{\partial}{\partial \bar{Q}_\beta} \left(\frac{\partial D_I^*}{\partial \bar{Q}_\alpha} \right) = \frac{\partial \bar{h}_\alpha}{\partial \bar{Q}_\beta}; \alpha, \beta = 1, M+1 \quad (\text{B.22})$$

and its excess part is

$$[\mathbf{H}_E^R]_{\alpha\beta} = \frac{\partial^2 D_E^*}{\partial \bar{Q}_\alpha \partial \bar{Q}_\beta}; \alpha, \beta = 1, M+1 \quad (\text{B.23})$$

From (eq. B.22)

$$\Delta \bar{\mathbf{h}} = \mathbf{H}_I^R \Delta \bar{\mathbf{Q}} \quad \text{and} \quad \Delta \bar{\mathbf{Q}} = [\mathbf{H}_I^R]^{-1} \Delta \bar{\mathbf{h}} \quad (\text{B.24})$$

and (eq. B.19) can be written as

$$(\mathbf{I} + \mathbf{H}_E^R [\mathbf{H}_I^R]^{-1}) \Delta \bar{\mathbf{h}} = -\frac{\partial D^*}{\partial \bar{\mathbf{Q}}} \quad (\text{B.25})$$

Using the chain rule, the inverse of \mathbf{H}^R is

$$[\mathbf{H}_I^R]^{-1} = \frac{\partial \bar{\mathbf{Q}}}{\partial \bar{\mathbf{h}}} = \frac{\partial \bar{\mathbf{Q}}}{\partial \mathbf{Y}} \frac{\partial \mathbf{Y}}{\partial \ln \bar{\mathbf{K}}} \frac{\partial \ln \bar{\mathbf{K}}}{\partial \bar{\mathbf{h}}} = \mathbf{C} \mathbf{U}^{-1} \mathbf{C}^T \quad (\text{B.26})$$

where $\frac{\partial \bar{\mathbf{Q}}}{\partial \mathbf{Y}}$ is given by (eq. B.14), $\frac{\partial \ln \bar{K}_i}{\partial Y_j} = U_{ij}$ (from eq. 3.9 and eq. 3.12), and $\frac{\partial \ln \bar{K}_i}{\partial h_\alpha} = q_{i\alpha}; i = 1, nc; \alpha = M+1$, from (eq. 3.95); finally $[\mathbf{H}_I^R]^{-1}$ is

$$[\mathbf{H}_I^R]_{\alpha\beta}^{-1} = \sum_{i=1}^{nc} q_{\alpha i} \sum_{j=1}^{nc} q_{\beta j} Y_j; \alpha, \beta = 1, M+1 \quad (\text{B.27})$$

Independent variables \mathbf{h} can be used instead of $\bar{\mathbf{h}}$ for solving the system of equations (eq. B.25); the gradients are the same, From (eq. 3.96) and $\bar{h}_\alpha = h_{\alpha z} - h_\alpha; \alpha = 1, M+1$

$$g_\alpha^R = h_\alpha(\mathbf{Q}) - h_\alpha; \alpha = 1, M+1 \quad (\text{B.28})$$

where $h_\alpha(\mathbf{Q})$ are given by (eq. 3.54, eq. 3.55, eq. 3.56).

The results are identical with both variables, since $\Delta \bar{\mathbf{h}} = \Delta \mathbf{h}$ and the Hessian matrices with respect to $\bar{\mathbf{h}}$ and \mathbf{h} are also identical.

(eq. B.28) are exactly the error equations (eq. 3.81) in the proposed method. Thus, the calculation procedure is equivalent to a constrained minimization of the modified TPD function.

The formalism presented in this appendix is related to those presented by [Kaul and Thrasher [1996]] for two-phase flash calculations and by [Firoozabadi and Pan [2002]]. In this work the modified TPD function D^* is minimized with respect to \mathbf{Y} and $\bar{\mathbf{Q}} = (\bar{Q}_1, \dots, \bar{Q}_M, Y_T)^T$ subject to constraints given by (eq. B.3, eq. B.4), while in [Firoozabadi and Pan [2002]], the TPD function D is minimized with respect to $N_C - 1$ independent mole fractions, subject to the constraints given by (eq. 3.47).

In the present formulation \bar{h}_{M+1} is not dependent only of \bar{h}_α ; $\alpha = 1, M$ (as in Refs. [Nichita and Graciaa [2011], Firoozabadi and Pan [2002]]) but also on Y_T

$$\bar{h}_{M+1} = \ln Y_T - \ln \left[\sum_{i=1}^{nc} z_i \exp \left(\sum_{\alpha=1}^M q_{\alpha i} \bar{h}_\alpha \right) \right] \quad (\text{B.29})$$

thus it is treated as an independent variable.

C Equivalence of conventional and reduction stability methods

The gradient in the conventional method can be expressed as

$$g_i = \frac{\partial D^*}{\partial Y_i} = \sum_{\alpha=1}^{M+1} \frac{\partial D^*}{\partial \bar{Q}_\alpha} \frac{\partial \bar{Q}_\alpha}{\partial Y_i}; i = 1, nc \quad (\text{C.1})$$

or, from (eq. B.14)

$$g_i = \sum_{\alpha=1}^{M+1} q_{\alpha i} \frac{\partial D^*}{\partial \bar{Q}_\alpha} = \sum_{\alpha=1}^{M+1} q_{\alpha i} g_\alpha^R; i = 1, nc \quad (\text{C.2})$$

or further, in matrix form

$$\mathbf{g} = \mathbf{C}^T \mathbf{g}^R \quad (\text{C.3})$$

The Hessian matrix in the conventional method can be expressed as

$$H_{ij} = \frac{\partial^2 D^*}{\partial Y_i \partial Y_j} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \sum_{\beta=1}^{M+1} q_{\beta j} \frac{\partial^2 D^*}{\partial \bar{Q}_\alpha \partial \bar{Q}_\beta} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \sum_{\beta=1}^{M+1} q_{\beta j} H_{\alpha\beta}^R; i, j = 1, nc \quad (\text{C.4})$$

which in matrix form reads

$$\mathbf{H} = \mathbf{C}^T \mathbf{H}^R \mathbf{C} \quad (\text{C.5})$$

Equations (eq. C.3) and (eq. C.5) relate gradients and Hessian matrices in conventional and reduction methods using the reduction matrix \mathbf{C} .

Using (eq. C.3 and eq. C.5), the Newton iteration equation in the conventional method (eq. 3.3) can be written as

$$\mathbf{C}^T \mathbf{H}^R \mathbf{C} \Delta \mathbf{Y} = -\mathbf{C}^T \mathbf{g}^R \quad (\text{C.6})$$

or, using $\Delta \mathbf{Y} = \mathbf{U}^{-1} \Delta \ln \mathbf{Y}$ (eq. 3.8) and $\Delta \ln \mathbf{Y} = \mathbf{C}^T \Delta \mathbf{h}$

$$\mathbf{C}^T \mathbf{H}^R \mathbf{C} \mathbf{U}^{-1} \mathbf{C}^T \Delta \mathbf{h} = -\mathbf{C}^T \mathbf{g}^R \quad (\text{C.7})$$

On the other hand, the Newton iteration equation in the reduction method (eq. B.19 and eq. B.25) is

$$\mathbf{H}^R \Delta \bar{\mathbf{Q}} = \mathbf{J}^R \Delta \bar{\mathbf{h}} = -\mathbf{g}^R \quad (\text{C.8})$$

where the Jacobian matrix in the reduction method is

$$\mathbf{J}^R = \frac{\partial \mathbf{g}^R}{\partial \bar{\mathbf{h}}} = \mathbf{H}^R [\mathbf{H}_I^R]^{-1} \quad (\text{C.9})$$

Finally, combining (eq. C.7) with (eq. C.8, (eq. C.9 and eq. B.26) and putting $\Delta \bar{\mathbf{h}} = \Delta \mathbf{h}$, one obtains

$$\mathbf{H}^R (\mathbf{C} \mathbf{U}^{-1} \mathbf{C}^T) \Delta \mathbf{h} = \mathbf{H}^R [\mathbf{H}_I^R]^{-1} \Delta \mathbf{h} = \mathbf{J}^R \Delta \mathbf{h} = -\mathbf{g}^R \quad (\text{C.10})$$

that is, the Newton iteration equation in the reduction method is obtained directly from the Newton iteration equation in the conventional method by some matrix algebra operations. This means that the convergence path in the compositional space is related to the convergence path in the reduced space by $\Delta \ln \mathbf{Y} = \mathbf{C}^T \Delta \mathbf{h}$; the number of iterations and Euclidean norms at each iteration level are the same in the proposed reduction method and in the conventional method with $\ln \mathbf{Y}$ as independent variables.

D Reduction flash as a constrained minimization problem

The formalism for a two-stage minimization of the Gibbs free energy was first established for zero BIPs by [Kaul and Thrasher [1996]], and extended later by [Pan and Firoozabadi [2003]] to reduced flash with non-zero BIPs using spectral decomposition. This Appendix follows the lines in [Pan and Firoozabadi [2003]], showing the equivalence, in terms of independent variables, gradients and Hessians of the reduction method of Nichita and Graciaa [Nichita and Graciaa [2011]] with a constrained minimization of the Gibbs free energy.

The dimensionless Gibbs free energy (the objective function) for a two-phase vapor-liquid system can be expressed as

$$G = G_I + G_E \quad (\text{D.1})$$

where subscripts I and E denote ideal and excess terms, respectively. The two terms in (eq. D.1) are

$$G_I = \sum_{i=1}^{nc} n_{iV} \ln y_i + \sum_{i=1}^{nc} n_{iL} \ln x_i \quad (\text{D.2})$$

and

$$G_E = V g_{E,V}(\mathbf{Q}_V) + L g_{E,L}(\mathbf{Q}_L) \quad (\text{D.3})$$

where the molar excess Gibbs free energy of the phase k is $g_k^E(\mathbf{x}_k) = \sum_{i=1}^{nc} x_{ik} \ln \phi_{ik}(\mathbf{x}_k)$, or, using [Michelsen et al. [2013b], Rasmussen et al. [2006]]

$$g_k^E(\mathbf{Q}_k) = \sum_{\alpha=1}^{M+1} Q_{k\alpha} h_{k\alpha}(\mathbf{Q}_k) \quad (\text{D.4})$$

which depends only on the reduction parameters.

It is more convenient to use the modified reduction parameters $\overline{\mathbf{Q}}_{\mathbf{V}} = (\overline{Q}_{V1}, \dots, \overline{Q}_{VM}, V)^T$, with

$$\overline{Q}_{V\alpha} = VQ_{V\alpha} = \sum_{i=1}^{nc} q_{\alpha i} n_{iV}; \alpha = 1, M \quad (\text{D.5})$$

$$\overline{Q}_{V,M+1} = V = \sum_{i=1}^{nc} q_{M+1,i} n_{iV} \quad (\text{D.6})$$

or

$$\overline{\mathbf{Q}}_{\mathbf{V}} = \mathbf{C}\mathbf{n}_{\mathbf{V}} \quad (\text{D.7})$$

Taking into account that $n_{iL} = z_i - n_{iV}$, $x_{ik} = n_{ik} / \sum_i n_{ik}$ and $Q_{L\alpha} = (Q_{F\alpha} - VQ_{V\alpha}) / L$ (with $Q_{F\alpha} = \sum_{i=1}^{nc} q_{\alpha i} z_i$), the objective function (eq. D.1) can be written as

$$G(\mathbf{n}_{\mathbf{V}}, \overline{\mathbf{Q}}_{\mathbf{V}}) = G_I(\mathbf{n}_{\mathbf{V}}) + G_E(\overline{\mathbf{Q}}_{\mathbf{V}}) \quad (\text{D.8})$$

that is, G_I depends only on mole numbers and G_E only on modified reduction parameters.

If G is minimized subject to the constraints given by (eq. D.5, eq. D.6), the Lagrangian function is

$$L(\mathbf{n}_{\mathbf{V}}, \overline{\mathbf{Q}}_{\mathbf{V}}, \mathbf{h}) = G(\mathbf{n}_{\mathbf{V}}, \overline{\mathbf{Q}}_{\mathbf{V}}) - \sum_{\alpha=1}^{M+1} h_{\alpha} \left(\sum_{i=1}^{nc} q_{\alpha i} n_{iV} - \overline{Q}_{V\alpha} \right) \quad (\text{D.9})$$

where $\mathbf{h} = (h_1, \dots, h_{M+1})^T$ is the vector of Lagrange multipliers.

$$\frac{\partial L}{\partial n_{Vi}} = 0 \Rightarrow \frac{\partial G_I}{\partial n_{Vi}} = \sum_{\alpha=1}^{M+1} h_{\alpha} q_{\alpha i} \quad (\text{D.10})$$

$$\frac{\partial L}{\partial \overline{Q}_{V\alpha}} = 0 \Rightarrow \frac{\partial G_E}{\partial \overline{Q}_{V\alpha}} + h_{\alpha} = 0 \quad (\text{D.11})$$

$$\frac{\partial L}{\partial h_{\alpha}} = 0 \Rightarrow \overline{Q}_{V\alpha} = \sum_{i=1}^{M+1} q_{\alpha i} n_{Vi} \quad (\text{D.12})$$

Or,

$$\frac{\partial G_I}{\partial n_{Vi}} = \ln K_i \quad \forall i = 1, nc \quad (\text{D.13})$$

$$\ln K_i = \sum_{\alpha=1}^{M+1} q_{\alpha i} h_{\alpha}; i = 1, nc \quad (\text{D.14})$$

which is exactly (eq. 3.61), which is the key equation in the reduction method of Nichita and Graciaa [16]. The Lagrange multipliers can be expressed as [Kaul and Thrasher [1996], Pan and Firoozabadi [2003]]. Taking into account that $\mathbf{n}_{\mathbf{V}}(\overline{\mathbf{Q}}_{\mathbf{V}})$, using a chain rule,

$$\frac{\partial G_I}{\partial n_{Vi}} = \sum_{\alpha=1}^{M+1} \frac{\partial G_I}{\partial \overline{Q}_{V\alpha}} \frac{\partial \overline{Q}_{V\alpha}}{\partial n_{Vi}} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \frac{\partial G_I}{\partial \overline{Q}_{V\alpha}} = \ln K_i \quad (\text{D.15})$$

By identification with (eq. D.14),

$$h_{\alpha} = \frac{\partial G_I}{\partial \overline{Q}_{V\alpha}}; \alpha = 1, M+1 \quad (\text{D.16})$$

The partial derivatives of G_E with respect to the modified reduction parameters are

$$\frac{\partial G_E}{\partial \bar{Q}_{V\alpha}} = \frac{\partial g_{E,V}}{\partial Q_{V\alpha}} - \frac{\partial g_{E,L}}{\partial Q_{L\alpha}} = h_{V\alpha}(\mathbf{Q}_V) - h_{L\alpha}(\mathbf{Q}_L); \alpha = 1, M \quad (\text{D.17})$$

and

$$\frac{\partial G_E}{\partial V} = g_{E,V} - g_{E,L} - \sum_{\alpha=1}^M Q_{V\alpha} \frac{\partial g_{E,V}}{\partial Q_{V\alpha}} + \sum_{\alpha=1}^M Q_{L\alpha} \frac{\partial g_{E,L}}{\partial Q_{L\alpha}} = h_{V,M+1}(\mathbf{Q}_V) - h_{L,M+1}(\mathbf{Q}_L) \quad (\text{D.18})$$

The equality of molar excess Gibbs free energy of a phase and its h-counterpart comes from

$$g_{Ek} - \sum_{\alpha=1}^M Q_{k\alpha} \frac{\partial g_{Ek}}{\partial Q_{k\alpha}} = \sum_{\alpha=1}^{M+1} Q_{k\alpha} h_{k\alpha} - \sum_{\alpha=1}^M Q_{k\alpha} h_{k\alpha} = h_{k,M+1}(\mathbf{Q}_k) \quad (\text{D.19})$$

Partial derivatives calculated from (eq. D.17 and eq. D.18) are simpler than those calculated in [Pan and Firoozabadi [2003]].

The elements of the gradient vector in the reduction method are

$$g_\alpha^R = \frac{\partial G}{\partial \bar{Q}_{V\alpha}} = \frac{\partial G_I}{\partial \bar{Q}_{V\alpha}} + \frac{\partial G_E}{\partial \bar{Q}_{V\alpha}}; \alpha = 1, M + 1 \quad (\text{D.20})$$

or, from (eq. D.16, eq. D.17 and eq. D.18)

$$g_\alpha^R = h_\alpha + h_{V\alpha}(\mathbf{Q}_V) - h_{L\alpha}(\mathbf{Q}_L); \alpha = 1, M + 1 \quad (\text{D.21})$$

which are exactly the error equations in the reduction method of [Nichita and Graciaa [2011]], see (eq. 3.68).

The Newton iteration equation in the reduction method is

$$\mathbf{H}^R \Delta \bar{\mathbf{Q}}_V = -\mathbf{g}^R \quad (\text{D.22})$$

with the elements of the Hessian matrix

$$H_{\alpha\beta}^R = \frac{\partial^2 G}{\partial \bar{Q}_{V\alpha} \partial \bar{Q}_{V\beta}} = \frac{\partial g_\alpha^R}{\partial \bar{Q}_{V\beta}}; \alpha, \beta = 1, M + 1 \quad (\text{D.23})$$

Or

$$(\mathbf{H}_I^R + \mathbf{H}_E^R) \Delta \bar{\mathbf{Q}}_V = -\mathbf{g}^R \quad (\text{D.24})$$

The ideal part of the Hessian is

$$[\mathbf{H}_I^R]_{\alpha\beta} = \frac{\partial^2 G_I}{\partial \bar{Q}_{V\alpha} \partial \bar{Q}_{V\beta}} = \frac{\partial}{\partial \bar{Q}_{V\beta}} \left(\frac{\partial G_I}{\partial \bar{Q}_{V\alpha}} \right) = \frac{\partial h_\alpha}{\partial \bar{Q}_{V\beta}}; \alpha, \beta = 1, M + 1 \quad (\text{D.25})$$

and its excess part is

$$[\mathbf{H}_E^R]_{\alpha\beta} = \frac{\partial^2 G_E}{\partial \bar{Q}_{V\alpha} \partial \bar{Q}_{V\beta}} = \frac{\partial}{\partial \bar{Q}_{V\beta}} \left(\frac{\partial G_E}{\partial \bar{Q}_{V\alpha}} \right) = \frac{\partial h_{V\alpha}(Q_V)}{\partial \bar{Q}_{V\beta}} - \frac{\partial h_{L\alpha}(Q_L)}{\partial \bar{Q}_{V\beta}}; \alpha, \beta = 1, M + 1 \quad (\text{D.26})$$

From (eq. D.25)

$$\Delta h = H_I^R \Delta \bar{\mathbf{Q}}_V \quad (\text{D.27})$$

thus

$$\Delta \overline{\mathbf{Q}}_{\mathbf{V}} = [\mathbf{H}_{\mathbf{I}}^{\mathbf{R}}]^{-1} \Delta \overline{\mathbf{h}} \quad (\text{D.28})$$

Using (eq. D.28), (eq. D.24) becomes

$$(\mathbf{I} + \mathbf{H}_{\mathbf{E}}^{\mathbf{R}}[\mathbf{H}_{\mathbf{I}}^{\mathbf{R}}]^{-1})\Delta \mathbf{h} = -\mathbf{g}^{\mathbf{R}} \quad (\text{D.29})$$

where the Jacobian matrix is

$$\mathbf{J}^{\mathbf{R}} = \mathbf{I} + \mathbf{H}_{\mathbf{E}}^{\mathbf{R}}[\mathbf{H}_{\mathbf{I}}^{\mathbf{R}}]^{-1} \quad (\text{D.30})$$

The elements of the inverse matrix $[\mathbf{H}_{\mathbf{I}}^{\mathbf{R}}]^{-1}$ are (from eq. D.25)

$$[H_{\mathbf{I}}^{\mathbf{R}}]_{\alpha\beta}^{-1} = \frac{\partial \overline{Q}_{V\alpha}}{\partial h_{\beta}}; \alpha, \beta = 1, M+1 \quad (\text{D.31})$$

and introducing (eq. D.26, eq. D.31) in (eq. D.30), the elements of the Jacobian matrix are

$$J_{\alpha\beta}^{\mathbf{R}} = \delta_{\alpha\beta} + \sum_{\gamma=1}^{M+1} \left(\frac{\partial h_{V\alpha}(\mathbf{Q}_{\mathbf{V}})}{\partial \overline{Q}_{V\gamma}} - \frac{\partial h_{L\alpha}(\mathbf{Q}_{\mathbf{L}})}{\partial \overline{Q}_{V\gamma}} \right) \frac{\partial \overline{Q}_{V\gamma}}{\partial h_{\beta}} = \delta_{\alpha\beta} + \frac{\partial h_{V\alpha}(\mathbf{Q}_{\mathbf{V}})}{\partial h_{\beta}} - \frac{\partial h_{L\alpha}(\mathbf{Q}_{\mathbf{L}})}{\partial h_{\beta}} \quad (\text{D.32})$$

which are exactly those from the reduction method of [Nichita and Graciaa [2011]], see (eq. 3.69).

Therefore, from (eq. D.21 and eq. D.32), the reduction method of [Nichita and Graciaa [2011]] corresponds to the constrained minimization of G with respect to $\overline{\mathbf{Q}}$ and subject to the equality constraints given by (eq. D.5, eq. D.6).

E Equivalence of conventional and reduction flash calculation methods

The equivalence of conventional and reduction flash calculation methods can be obtained in a similar manner to that for stability analysis in [Nichita and Petitfrere [2013]].

The gradient in the conventional method can be expressed as

$$g_i = \frac{\partial G}{\partial n_{iV}} = \sum_{\alpha=1}^{M+1} \frac{\partial G}{\partial \overline{Q}_{V\alpha}} \frac{\partial \overline{Q}_{V\alpha}}{\partial n_{iv}}; \quad i = 1, nc \quad (\text{E.1})$$

or, since $\partial \overline{Q}_{V\alpha} / \partial n_{iv} = q_{\alpha i}$ (from (eq. D.5 and eq. D.6))

$$g_i = \sum_{\alpha=1}^{M+1} q_{\alpha i} \frac{\partial G}{\partial \overline{Q}_{V\alpha}} = \sum_{\alpha=1}^{M+1} q_{\alpha i} g_{\alpha}^{\mathbf{R}}; i = 1, nc \quad (\text{E.2})$$

or further, in matrix form

$$\mathbf{g} = \mathbf{C}^T \mathbf{g}^{\mathbf{R}} \quad (\text{E.3})$$

The Hessian matrix in the conventional method can be expressed as

$$H_{ij} = \frac{\partial^2 G}{\partial n_{iV} \partial n_{jV}} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \sum_{\beta=1}^{M+1} q_{\beta j} \frac{\partial^2 G}{\partial \overline{Q}_{V\alpha} \partial \overline{Q}_{V\beta}} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \sum_{\beta=1}^{M+1} q_{\beta j} H_{\alpha\beta}^{\mathbf{R}}; i, j = 1, nc \quad (\text{E.4})$$

which in matrix form reads

$$\mathbf{H} = \mathbf{C}^T \mathbf{H}^R \mathbf{C} \quad (\text{E.5})$$

Equations (eq. E.3 and E.5) relate gradients and Hessian matrices in conventional and reduction methods using the reduction matrix \mathbf{C} . Introducing these two equations in the Newton iteration equation in the conventional method (eq. 3.25) one obtains

$$\mathbf{C}^T \mathbf{H}^R \mathbf{C} \Delta \mathbf{n}_V = -\mathbf{C}^T \mathbf{g}^R \quad (\text{E.6})$$

(eq. E.6) with $\mathbf{C} \Delta \mathbf{n}_V = \Delta \overline{\mathbf{Q}}_V$ (from eq. D.7) can be obtained from the Newton iteration equation (eq. D.22) by a simple transformation; therefore, if $\overline{\mathbf{Q}}_V$ are used as independent variables in the reduction method, the convergence path is related to that of the conventional method using the variables \mathbf{n}_V .

Now using $\Delta \mathbf{n}_V = \mathbf{U}^{-1} \Delta \ln \mathbf{K}$ (from Eq. 5) and $\Delta \ln \mathbf{K} = \mathbf{C}^T \Delta \mathbf{h}$ (from eq. 3.62), (eq. E.6) becomes

$$\mathbf{C}^T \mathbf{H}^R \mathbf{C} \mathbf{U}^{-1} \mathbf{C}^T \Delta \mathbf{h} = -\mathbf{C}^T \mathbf{g}^R \quad (\text{E.7})$$

On the other hand, the Newton iteration equation in the reduction method is

$$\mathbf{H}^R \Delta \overline{\mathbf{Q}}_V = \mathbf{J}^R \Delta \mathbf{h} = -\mathbf{g}^R \quad (\text{E.8})$$

where the Jacobian matrix in the reduction method is (from eq. D.30)

$$\mathbf{J}^R = \frac{\partial \mathbf{g}^R}{\partial \mathbf{h}} = \mathbf{H}^R [\mathbf{H}_I^R]^{-1} \quad (\text{E.9})$$

The inverse of \mathbf{H}_I^R is

$$[\mathbf{H}_I^R]^{-1} = \frac{\partial \overline{\mathbf{Q}}_V}{\partial \mathbf{h}} = \frac{\partial \overline{\mathbf{Q}}_V}{\partial \mathbf{n}_V} \frac{\partial \mathbf{n}_V}{\partial \ln \mathbf{K}} \frac{\partial \ln \mathbf{K}}{\partial \mathbf{h}} = \mathbf{C} \mathbf{U}^{-1} \mathbf{C}^T \quad (\text{E.10})$$

where $\partial \overline{\mathbf{Q}}_V / \partial \mathbf{n}_V = \mathbf{C}$ and $\partial \ln K / \partial \mathbf{h} = \mathbf{C}^T$ are obtained by derivation in (eq. D.7) and (eq. 3.62), respectively.

Finally, combining (eq. E.7, eq. E.8, eq. E.9, eq. E.10), one obtains

$$\mathbf{H}^R (\mathbf{C} \mathbf{U}^{-1} \mathbf{C}^T) \Delta \mathbf{h} = \mathbf{H}^R [\mathbf{H}_I^R]^{-1} \Delta \mathbf{h} = \mathbf{J}^R \Delta \mathbf{h} = -\mathbf{g}^R \quad (\text{E.11})$$

that is, the Newton iteration equation in the reduction method is obtained directly from the Newton iteration equation in the conventional method by some matrix algebra operations. This means that the convergence path in the compositional space is related to the convergence path in the reduced space by $\Delta \ln \mathbf{K} = \mathbf{C}^T \Delta \mathbf{h}$, that is, at each iteration level in the reduction method, $\ln K_i^{(\nu)}$ corresponding to $h_\alpha^{(\nu)}$ are the same as those in the conventional $\ln K$ method. Thus, the number of iterations and Euclidean norms at each iteration level are the same in the reduction method and in the conventional $\ln K$ method.

F Link between reduction methods Q-red. and h-red. for phase stability testing

The partial derivatives in (eq. 3.84) can also be expressed as

$$\frac{\partial h_\alpha(\mathbf{Q})}{\partial h_\beta} = \sum_{i=1}^{nc} \frac{\partial h_\alpha[\mathbf{Q}(\mathbf{x})]}{\partial x_i} \frac{\partial x_i}{\partial h_\beta}; \alpha, \beta = 1, M+1 \quad (\text{F.1})$$

Taking into account that [Nichita and Petitfrere [2013]]

$$\frac{\partial x_i}{\partial h_\beta} = x_i (Q_\beta - q_{\beta i}) \quad (\text{F.2})$$

the partial derivatives in (eq. F.1) reads

$$\frac{\partial h_\alpha}{\partial h_\beta} = Q_\beta \sum_{i=1}^{nc} x_i \frac{\partial h_\alpha}{\partial x_i} - \sum_{i=1}^{nc} q_{\beta i} x_i \frac{\partial h_\alpha}{\partial x_i} \quad (\text{F.3})$$

The following equality holds

$$\frac{\partial \ln \phi_i}{\partial Q_\alpha} = \frac{\partial h_\alpha}{\partial x_i}; \quad \alpha = 1, M; i = 1, nc \quad (\text{F.4})$$

Since

$$\frac{\partial \ln \phi_i}{\partial Q_\alpha} = \sum_{\gamma=1}^M \frac{\partial \ln \phi_i}{\partial h_\gamma} \frac{\partial h_\gamma}{\partial Q_\alpha} = \sum_{\gamma=1}^M q_{\gamma i} \frac{\partial h_\gamma}{\partial Q_\alpha}; \alpha = 1, M; i = 1, nc \quad (\text{F.5})$$

And

$$\frac{\partial h_\alpha}{\partial x_i} = \sum_{\gamma=1}^M \frac{\partial h_\alpha}{\partial Q_\gamma} \frac{\partial Q_\gamma}{\partial x_i} = \sum_{\gamma=1}^M q_{\gamma i} \frac{\partial h_\alpha}{\partial Q_\gamma}; \quad \alpha = 1, M; i = 1, nc \quad (\text{F.6})$$

Combining (eq. F.3) with (eq. F.4) and (eq. 3.73) and putting $x_i = Y_i/Y_T$, one obtains

$$\frac{\partial h_\alpha}{\partial h_\beta} = -\frac{1}{Y_T} \left[Q_\beta \sum_{i=1}^{nc} \frac{\partial Y_i}{\partial Q_\alpha} - \sum_{i=1}^{nc} q_{\beta i} \frac{\partial Y_i}{\partial Q_\alpha} \right] \quad (\text{F.7})$$

and the elements of the Jacobian matrix $\mathbf{J}^{\mathbf{R}}$ are

$$J_{\alpha\beta}^{\mathbf{R}} = \delta_{\alpha\beta} - \frac{\partial h_\alpha}{\partial h_\beta} = \frac{1}{Y_T} \left[\delta_{\alpha\beta} \sum_{i=1}^{nc} Y_i + Q_\beta \sum_{i=1}^{nc} \frac{\partial Y_i}{\partial Q_\alpha} - \sum_{i=1}^{nc} q_{\beta i} \frac{\partial Y_i}{\partial Q_\alpha} \right]; \alpha, \beta = 1, M+1 \quad (\text{F.8})$$

By comparing (eq.) with (eq. 3.74), the Jacobian matrices $\mathbf{J}^{\mathbf{R}}$ (its elements for $\alpha, \beta = 1, M$) and $\mathbf{J}^{\mathbf{R-Q}}$ are related by

$$J_{\alpha\beta}^{\mathbf{R}} = \frac{1}{Y_T} \left[J_{\alpha\beta}^{\mathbf{R-Q}} \right]^T; \quad \alpha, \beta = 1, M \quad (\text{F.9})$$

G Pseudo-reduction methods for phase stability testing

The gradient vector in the pseudo-reduced method is [Michelsen et al. [2013b]]

$$\bar{g}_\alpha^* = \frac{\partial D}{\partial h_\alpha} = \sum_{i=1}^{nc} \frac{\partial D}{\partial \alpha_i} \frac{\partial \alpha_i}{\partial h_\alpha}; \quad \alpha = 1, M+1 \quad (\text{G.1})$$

Or

$$\bar{\mathbf{g}}^* = \mathbf{T} \bar{\mathbf{g}} \quad (\text{G.2})$$

Where

$$T_{\alpha i} = \frac{\partial \alpha_i}{\partial h_\alpha} = \sum_{i=1}^{nc} \frac{\partial \alpha_i}{\partial \ln Y_i} \frac{\partial \ln Y_i}{\partial h_\alpha}; \quad \alpha = 1, M+1, i = 1, nc \quad (\text{G.3})$$

or $\mathbf{T}^{\mathbf{T}} = \mathbf{U}^{-1/2} \mathbf{C}^{\mathbf{T}}$, or

$$\mathbf{T} = \mathbf{C}\mathbf{U}^{-1/2} \quad (\text{G.4})$$

The Hessian matrix is

$$\mathbf{H}^* = \mathbf{T}\bar{\mathbf{H}}\mathbf{T}^T \quad (\text{G.5})$$

and the Newton iteration $\mathbf{H}^*\Delta\mathbf{h} = -\bar{\mathbf{g}}^*$ becomes

$$\mathbf{T}\bar{\mathbf{H}}\mathbf{T}^T\Delta\mathbf{h} = -\mathbf{T}\bar{\mathbf{g}} \quad (\text{G.6})$$

[Michelsen et al. [2013b]] used an approximation of the Hessian by neglecting \mathbf{g} in (eq. G.5). Putting $\mathbf{g} = \mathbf{0}$ in (eq. 3.17) relating Hessians $\bar{\mathbf{H}}$ and \mathbf{H} , from (eq. G.5) we have

$$\mathbf{H}^* = \mathbf{T}\mathbf{U}^{-1/2}\mathbf{H}\mathbf{U}^{-1/2}\mathbf{T}^T \quad (\text{G.7})$$

Or

$$\mathbf{H}^* = \mathbf{V}\mathbf{H}\mathbf{V}^T \quad (\text{G.8})$$

Where

$$\mathbf{V} = \mathbf{C}\mathbf{U}^{-1} = \mathbf{T}\mathbf{U}^{-1/2} \quad (\text{G.9})$$

Using (eq. 3.16) relating gradients $\bar{\mathbf{g}}$ and \mathbf{g} , (eq. G.6) becomes

$$\mathbf{V}\mathbf{H}\mathbf{V}^T\Delta\mathbf{h} = -\mathbf{V}\mathbf{g} \quad (\text{G.10})$$

Let us start now with the Newton iteration equation (eq. 3.3) for Y_i as independent variables. Since $\ln \mathbf{Y} = \mathbf{C}^T\Delta\mathbf{h}$

$$\mathbf{H}\mathbf{U}^{-1}\mathbf{C}^T\Delta\mathbf{h} = -\mathbf{g} \quad (\text{G.11})$$

By left-multiplying both members of (eq. G.11) with $\mathbf{V} = \mathbf{C}\mathbf{U}^{-1}$, and taking into account that $\mathbf{U}^{-1}\mathbf{C}^T = \mathbf{V}^T$, we get exactly (eq. G.10). Thus, Michelsens equation is the same as directly obtained from the Newton equation with \mathbf{Y} as independent variables. Note that for evaluating \mathbf{V} no square roots are required. Thus, Michelsens pseudo-reduced method for stability testing is approximate with respect to $\bar{\mathbf{H}}$ and exact with respect to \mathbf{H} .

Finally, one can note that by left-multiplying (eq. 3.83) with $[\mathbf{H}_I^R]^{-1}$ the resulting linear system is identical to that of Michelsen (starting from the conventional method), (eq. G.6), but obtained starting from the reduced method. The coefficient matrix is symmetric, but the procedure is not efficient, since a matrix product followed by a Cholesky factorization is slower than an LU factorization.

H Partial derivatives in the direct extension of Nichita and Gracia[2011]'s reduction method

The partial derivatives

$$\frac{\partial h_{\alpha k}}{\partial h_{\beta p}} = \sum_{\gamma=1}^M \frac{\partial h_{\alpha k}}{\partial Q_{\gamma k}} \frac{\partial Q_{\gamma k}}{\partial h_{\beta p}}; \alpha, \beta = 1, M+1; k, p = 1, np; p \neq R \quad (\text{H.1})$$

are required in the expression of the Jacobian matrix.

$$\frac{\partial h_{\alpha k}}{\partial Q_{\gamma k}} = \left(\frac{\partial h_{\alpha k}}{\partial Q_{\gamma k}} \right)_{Z_k, Q_{\mu \neq \gamma, k}} + \left(\frac{\partial h_{\alpha k}}{\partial Z_k} \right)_{\mathbf{Q}_k} \frac{\partial Z_k}{\partial Q_{\gamma k}}; \alpha = 1, M+1; \gamma = 1, M; k = 1, np \quad (\text{H.2})$$

are calculated as in [Nichita and Graciaa [2011]] and

$$\frac{\partial Q_{\gamma k}}{\partial h_{\beta p}} = \sum_{i=1}^{nc} q_{\gamma i} \frac{\partial x_{ik}}{\partial h_{\beta p}}; \beta = 1, M+1; \gamma = 1, M \quad (\text{H.3})$$

where

$$\frac{\partial x_{ik}}{\partial h_{\beta p}} = \left(\frac{\partial x_{ik}}{\partial K_{ip}} \right)_{\theta} \frac{\partial K_{ip}}{\partial h_{\beta p}} + \sum_{\substack{m=1 \\ m \neq R}}^{np} \left(\frac{\partial x_{ik}}{\partial \theta_m} \right)_{\mathbf{K}} \frac{\partial \theta_m}{\partial h_{\beta p}}; \beta = 1, M+1; i = 1, nc \quad (\text{H.4})$$

and

$$\frac{\partial x_{ik}}{\partial K_{ip}} = d_i (\delta_{kp} - K_{ik} \theta_p); k, p = 1, np, k \neq R \quad (\text{H.5})$$

$$\frac{\partial x_{iR}}{\partial K_{ip}} = -d_i \theta_p \quad (\text{H.6})$$

$$\frac{\partial x_{ik}}{\partial \theta_p} = -d_i K_{ik} (K_{ip} - 1); k = 1, np, k \neq R \quad (\text{H.7})$$

$$\frac{\partial x_{iR}}{\partial \theta_p} = d_i (K_{ip} - 1); k = 1, np, k \neq R \quad (\text{H.8})$$

$$\frac{\partial K_{ip}}{\partial h_{\beta p}} = q_{\beta i}; \beta = 1, M+1; i = 1, nc \quad (\text{H.9})$$

$$d_i = \frac{z_i}{E_i^2}; i = 1, nc \quad (\text{H.10})$$

From the Rachford-Rice equations

$$\sum_{\substack{m=1 \\ m \neq R}}^{np} \left(\frac{\partial R_k}{\partial \theta_m} \right)_{\mathbf{K}, \theta_{s \neq m}} \frac{\partial \theta_m}{\partial h_{\beta p}} = - \left(\frac{\partial R_k}{\partial h_{\beta p}} \right)_{\theta}; \beta = 1, M+1; k, p = 1, np; k, p \neq R \quad (\text{H.11})$$

Or

$$\mathbf{S} \left(\frac{\partial \theta}{\partial \mathbf{h}} \right)_{\beta p} = (\mathbf{f}^R)_{\beta p}; \beta = 1, M+1; p = 1, np; p \neq R \quad (\text{H.12})$$

This implies the resolution of $(M+1) \times (np-1)$ linear systems of dimensionality $(np-1) \times (np-1)$. The matrix \mathbf{S} is-evaluated and decomposed once, than only back substitutions are required for the RHS vectors

$$(f_k^R)_{\beta p} = \left(\frac{\partial R_k}{\partial h_{\beta p}} \right)_{\theta} = \sum_{i=1}^{nc} q_{\alpha i} \left(\frac{\partial R_k}{\partial \ln K_{ip}} \right)_{\theta} = \delta_{kp} Q_{\beta R} - \theta_p \sum_{i=1}^{nc} q_{\alpha i} x_{iR} w_{ik} \quad (\text{H.13})$$

I Gradient vector in multiphase reduction

The elements of the gradient vector in the reduction method are

$$g_{\alpha k}^R = \frac{\partial G}{\partial Q_{\alpha k}} = \frac{\partial G_I}{\partial Q_{\alpha k}} + \frac{\partial G_E}{\partial Q_{\alpha k}}; \alpha = 1, M+1 \quad (\text{I.1})$$

Differentiating $G_I(\mathbf{n}) = G_I(\bar{Q}(\mathbf{n}))$ with respect to \mathbf{n}_k gives

$$\frac{\partial G_I}{\partial n_{ik}} = \sum_{\alpha=1}^{M+1} \frac{\partial G_I}{\partial \bar{Q}_{\alpha k}} \frac{\partial \bar{Q}_{\alpha k}}{\partial n_{ik}} = \sum_{\alpha=1}^{M+1} \frac{\partial G_I}{\partial \bar{Q}_{\alpha k}} q_{\alpha i} \quad (\text{I.2})$$

On the other hand, combining (eq. 3.122) and (eq. 3.124) gives

$$\frac{\partial G_I}{\partial n_{ik}} = \sum_{\alpha=1}^{M+1} q_{\alpha i} \bar{h}_{\alpha k}; i = 1, nc; k = 1, np; k \neq R \quad (\text{I.3})$$

From (eq. I.2) and (eq. I.3) the relation between Lagrange multipliers and the partial derivative of with respect to the modified reduction parameters follows

$$\bar{h}_{\alpha k} = \frac{\partial G_I}{\partial \bar{Q}_{\alpha k}}; \alpha = 1, M+1 \quad (\text{I.4})$$

The partial derivatives of G_E with respect to the modified reduction parameters are

$$\frac{\partial G_E}{\partial \bar{Q}_{\alpha k}} = \frac{\partial g_{E,k}}{\partial Q_{\alpha k}} - \frac{\partial g_{E,R}}{\partial Q_{\alpha R}}; \alpha = 1, M \quad (\text{I.5})$$

From (eq. D.4)

$$\frac{\partial g_{E,k}}{\partial Q_{\alpha k}} = h_{\alpha k}; \alpha = 1, M; k = 1, np \quad (\text{I.6})$$

Introducing (eq. I.6) into (eq. I.5) gives

$$\frac{\partial G_E}{\partial \bar{Q}_{\alpha k}} = h_{\alpha k} - h_{\alpha R}; \alpha = 1, M \quad (\text{I.7})$$

For $\alpha = M+1$ the derivatives of G_E are

$$\frac{\partial G_E}{\partial \bar{Q}_{M+1,k}} = \frac{\partial G_E}{\partial \theta_k} = g_{E,k} - g_{E,R} - \sum_{\alpha=1}^M Q_{\alpha k} \frac{\partial g_{E,k}}{\partial Q_{\alpha k}} + \sum_{\alpha=1}^M Q_{\alpha R} \frac{\partial g_{E,R}}{\partial Q_{\alpha R}} \quad (\text{I.8})$$

Since, using (eq. D.4) and (eq. I.6)

$$g_{E,k} - \sum_{\alpha=1}^M Q_{\alpha k} \frac{\partial g_{E,k}}{\partial Q_{\alpha k}} = \sum_{\alpha=1}^{M+1} Q_{\alpha k} h_{\alpha k} - \sum_{\alpha=1}^M Q_{\alpha k} h_{\alpha k} = h_{M+1,k} \quad (\text{I.9})$$

and from (eq. I.8) and (eq. I.9)

$$\frac{\partial G_E}{\partial \bar{Q}_{M+1,k}} = h_{M+1,k} - h_{M+1,R} \quad (\text{I.10})$$

Introducing (eq. I.4, eq. I.7 and eq. I.10) in (eq. I.1), the expression of the gradient is

$$g_{\alpha k}^R = \frac{\partial G}{\partial \bar{Q}_{\alpha k}} = \bar{h}_{\alpha k} + h_{\alpha k} - h_{\alpha R} = 0; \alpha = 1, M+1; k = 1, np; k \neq R \quad (\text{I.11})$$

J Elements of the Hessian matrix in multiphase reduction

The ideal part of the Hessian is (using eq. I.4)

$$(H_I^R)_{\alpha\beta, kp} = \frac{\partial^2 G_I}{\partial \bar{Q}_{\alpha k} \partial \bar{Q}_{\beta p}} = \frac{\partial}{\partial \bar{Q}_{\beta p}} \left(\frac{\partial G_I}{\partial \bar{Q}_{\alpha k}} \right) = \frac{\partial \bar{h}_{\alpha k}}{\partial \bar{Q}_{\beta p}}; \alpha, \beta = 1, M+1 \quad (\text{J.1})$$

and its excess part is

$$(H_E^R)_{\alpha\beta, kp} = \frac{\partial^2 G_E}{\partial \bar{Q}_{\alpha k} \partial \bar{Q}_{\beta p}} = \frac{\partial}{\partial \bar{Q}_{\beta p}} \left(\frac{\partial G_E}{\partial \bar{Q}_{\alpha k}} \right) = \frac{\partial h_{\alpha k}(\mathbf{Q}_k)}{\partial \bar{Q}_{\beta p}} - \frac{\partial h_{\alpha R}(\mathbf{Q}_R)}{\partial \bar{Q}_{\beta p}}; \alpha, \beta = 1, M+1 \quad (\text{J.2})$$

The inverse of the ideal part of the Hessian matrix needs to be calculated The block kp is

$$[\mathbf{H}_I^R]_{kp}^{-1} = \frac{\partial \bar{\mathbf{Q}}_k}{\partial \bar{\mathbf{h}}_p} = \frac{\partial \bar{\mathbf{Q}}_k}{\partial \mathbf{n}_k} \frac{\partial \mathbf{n}_k}{\partial \ln \mathbf{K}_p} \frac{\partial \ln \mathbf{K}_p}{\partial \bar{\mathbf{h}}_p} = \mathbf{C} \mathbf{U}_{kp}^{-1} \mathbf{C}^T \quad (\text{J.3})$$

where from (eq. 3.117)

$$\frac{\partial \bar{\mathbf{Q}}_k}{\partial \mathbf{n}_k} = \mathbf{C}; \quad [(M+1) \times nc] \quad (\text{J.4})$$

from (eq. 3.108)

$$\frac{\partial \ln \mathbf{K}_p}{\partial \bar{\mathbf{h}}_p} = \mathbf{C}^T; \quad [nc \times (M+1)] \quad (\text{J.5})$$

and from (eq. 3.37)

$$\frac{\partial \mathbf{n}_k}{\partial \ln \mathbf{K}_p} = \mathbf{U}_{kp}^{-1}; \quad [nc \times nc] \quad (\text{J.6})$$

and finally the elements of $[\mathbf{H}_I^R]_{kp}^{-1}$ (for the block kp) are

$$([\mathbf{H}_I^R]^{-1})_{\alpha\varphi, kp} = \sum_{i=1}^{nc} q_{\alpha i} \sum_{j=1}^{nc} q_{\beta j} (U_{ij}^{-1})_{kp}; \alpha, \beta = 1, M+1 \quad (\text{J.7})$$

The matrix $[\mathbf{H}_I^R]^{-1}$ has the block structure

$$\begin{pmatrix} [\mathbf{H}_I^R]_{11}^{-1} & \cdots & [\mathbf{H}_I^R]_{1,nc}^{-1} \\ \vdots & \ddots & \vdots \\ [\mathbf{H}_I^R]_{nc,1}^{-1} & \cdots & [\mathbf{H}_I^R]_{nc,nc}^{-1} \end{pmatrix} = \begin{pmatrix} \mathbf{C} & \mathbf{0} \\ & \ddots \\ \mathbf{0} & \mathbf{C} \end{pmatrix} \begin{pmatrix} (\mathbf{U}^{-1})_{11} & \cdots & (\mathbf{U}^{-1})_{1,nc} \\ \vdots & \ddots & \vdots \\ (\mathbf{U}^{-1})_{nc,1} & \cdots & (\mathbf{U}^{-1})_{nc,nc} \end{pmatrix} \begin{pmatrix} \mathbf{C}^T & & \mathbf{0} \\ & \ddots & \\ \mathbf{0} & & \mathbf{C}^T \end{pmatrix} \quad (\text{J.8})$$

The off diagonal blocks are zero because

$$\frac{\partial Q_{\alpha k}}{\partial n_{jp}} = \delta_{kp} q_{\alpha j}; \alpha = 1, M+1; j = 1, nc; k, p = 1, np; k, p \neq R \quad (\text{J.9})$$

$$\frac{\partial \ln K_{ik}}{\partial \bar{h}_{\beta p}} = \delta_{kp} q_{\beta i}; \beta = 1, M+1; i = 1, nc; k, p = 1, np; k, p \neq R \quad (\text{J.10})$$

For the particular case of two-phase equilibrium

$$[H_I^R]_{\alpha\beta}^{-1} = \frac{\theta_V \theta_L}{s} \left(s \sum_{i=1}^{nc} q_{\alpha i} q_{\beta i} u_i + S_\alpha S_\beta \right); \alpha, \beta = 1, M+1 \quad (\text{J.11})$$

Where

$$s = 1 - \sum_{i=1}^{nc} u_i; \quad S_\alpha = \sum_{i=1}^{nc} q_{\alpha i} u_i; \quad u_i = \frac{x_i y_i}{z_i} \quad (\text{J.12})$$

The elements of the excess part of the Hessian are

$$(H_E^R)_{\alpha\beta, kp} = \frac{\partial h_{\alpha k}}{\partial \bar{Q}_{\beta p}} - \frac{\partial h_{\alpha R}}{\partial \bar{Q}_{\beta p}} = \frac{\delta_{kp}}{\theta_k} \frac{\partial h_{\alpha k}}{\partial Q_{\beta k}} + \frac{1}{\theta_R} \frac{\partial h_{\alpha R}}{\partial Q_{\beta R}}; \alpha, \beta = 1, M \quad (\text{J.13})$$

$$(H_E^R)_{\alpha, M+1, kp} = \frac{\partial h_{\alpha k}}{\partial \theta_p} - \frac{\partial h_{\alpha R}}{\partial \theta_p} = -\frac{\delta_{kp}}{\theta_k} \sum_{\gamma=1}^M Q_{\gamma p} \frac{\partial h_{\alpha k}}{\partial Q_{\gamma k}} - \frac{1}{\theta_R} \sum_{\gamma=1}^M Q_{\gamma R} \frac{\partial h_{\alpha R}}{\partial Q_{\gamma R}}; \alpha = 1, M \quad (\text{J.14})$$

$$(H_E^R)_{M+1, \beta, kp} = \frac{\partial h_{M+1, k}}{\partial Q_{\beta p}} - \frac{\partial h_{M+1, R}}{\partial Q_{\beta p}} = \frac{\delta_{kp}}{\theta_k} \frac{\partial h_{M+1, k}}{\partial Q_{\beta k}} + \frac{1}{\theta_R} \frac{\partial h_{M+1, R}}{\partial Q_{\beta R}}; \beta = 1, M \quad (\text{J.15})$$

$$(H_E^R)_{M+1, M+1, kp} = \frac{\partial h_{M+1, k}}{\partial \theta_p} - \frac{\partial h_{M+1, R}}{\partial \theta_p} = -\frac{\delta_{kp}}{\theta_k} \sum_{\gamma=1}^M Q_{\gamma p} \frac{\partial h_{M+1, k}}{\partial Q_{\gamma k}} - \frac{1}{\theta_R} \sum_{\gamma=1}^M Q_{\gamma R} \frac{\partial h_{M+1, R}}{\partial Q_{\gamma R}} \quad (\text{J.16})$$

Because

$$(H_E^R)_{\alpha, M+1, kp} = (H_E^R)_{M+1, \beta kp}; \alpha, \beta = 1, M \quad (\text{J.17})$$

(eq. J.15) is used since there are no summations involved.

APPENDIX

Semi-continuous description of mixture composition

K Computation of the α_k and β_k

The orthogonal polynomial obeys a three-term recurrence relation given by the following formula:

$$p_{n+1}(x) = (x - \alpha_n)p_n(x) - \beta_n p_{n-1}(x), n = 0, 1, \dots \quad (\text{K.1})$$

With $p_{-1}(x) = 0$, $p_0(x) = 1$ and

$$\alpha_n = \frac{\langle p_n, xp_n \rangle}{\langle p_n, p_n \rangle} \quad (\text{K.2})$$

$$\beta_n = \frac{\langle p_n, p_n \rangle}{\langle p_{n-1}, p_{n-1} \rangle} \quad (\text{K.3})$$

Where $\langle \cdot, \cdot \rangle$ is the inner product. For a standard Gaussian quadrature, the coefficients α_n , β_n are given. However, if arbitrary weight functions are used, one needs to calculate these coefficients; the calculation procedure follows the method of [Gautschi [1994]]. Inner products have to be calculated as

$$\langle u, v \rangle = \int_a^b u(x)v(x)W(x)dx \quad (\text{K.4})$$

First, the interval $[a, b]$ (corresponding to $[I_{\min}, I_{\max}]$) is decomposed in a finite number of subintervals: $\text{supp } w \subset \bigcup_{i=1}^m [a_i, b_i]$, $m \geq 1$, giving

$$\int_a^b u(t)v(t)W(t)dt = \sum_{i=1}^m \int_{a_i}^{b_i} u(t)v(t)W(t)dt \quad (\text{K.5})$$

The integral in the right hand side of K.5 is approximated by a classical quadrature rule (the Fejr quadrature is used, [Gautschi [1994]]).

$$\int_{a_i}^{b_i} u(t)v(t)W(t)dt \approx \sum_{r=1}^{N_i} w_{r,i} u(x_{r,i})v(x_{r,i}) \quad (\text{K.6})$$

where N_i is the number of quadrature points used to discrete the inner product, and K.5 becomes

$$\int_a^b u(t)v(t)W(t)dt \approx \sum_{i=1}^m \sum_{r=1}^{N_i} w_{r,i} u(x_{r,i})v(x_{r,i}) \quad (\text{K.7})$$

$\alpha_0^{N_i}$ and $\beta_0^{N_i}$ can then be calculated from

$$\alpha_0^{N_i} = \frac{\int_a^b tW(t)dt}{\int_a^b W(t)dt} = \frac{\sum_{i=1}^m \sum_{r=1}^{N_i} w_{r,i} x_{r,i}}{\beta_0^{N_i}} \quad (\text{K.8})$$

$$\beta_0^{N_i} = \int_a^b W(t)dt = \sum_{i=1}^m \sum_{r=1}^{N_i} w_{r,i} \quad (\text{K.9})$$

From here, two different methods have been implemented: The first one is the the Stieljes procedure, based on the direct computation of $\alpha_k^{N_i}$ and $\beta_k^{N_i}$ from the recurrence (A1); it can give bad condition numbers for large n (greater than 12 [Press et al. [1992]]). The second method is based on the Lanczos algorithm [Lanczos [1950]]. It ensures a good condition number for any value of n . For a symmetric matrix \mathbf{A} , there exists a similarity transformation $\mathbf{Q}^T \mathbf{A} \mathbf{Q} = \mathbf{T}_2$. Given the matrix \mathbf{A} , the Lanczos algorithm allows to compute the matrices \mathbf{Q} and \mathbf{T}_2 which satisfy this relation. For the computed nodes $x_{r,i}$ and weights $w_{r,i}$ coming from the Fejer quadrature, [Gautschi [1994]] showed that there exists a matrix \mathbf{Q}_1^T which satisfies

$$\begin{bmatrix} \mathbf{1} & \mathbf{0}^T \\ \mathbf{0} & \mathbf{Q}_1^T \end{bmatrix} \begin{bmatrix} \mathbf{1} & \sqrt{\mathbf{w}}^T \\ \sqrt{\mathbf{w}} & \Delta \end{bmatrix} \begin{bmatrix} \mathbf{1} & \mathbf{0} \\ \mathbf{0}^T & \mathbf{Q}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{1} & \sqrt{\beta_{0,N}} \mathbf{e}_1^T \\ \sqrt{\beta_{0,N}} \mathbf{e}_1 & \mathbf{T} \end{bmatrix} \quad (\text{K.10})$$

Where $\mathbf{w} = (\sqrt{w_{11}}, \sqrt{w_{12}}, \dots, \sqrt{w_{21}}, \dots, \sqrt{w_{N_i,m}})^T$, $\text{diag}(\Delta) = (x_{11}, x_{12}, \dots, x_{21}, \dots, x_{N_i,m})$ and \mathbf{T} is a tri-diagonal matrix [Golub and Welsch [1969]] which comes from the recursive relation (eq. K.1) (see Appendix E).

Calling the symmetric matrix

$$A = \begin{bmatrix} \mathbf{1} & \sqrt{\mathbf{w}}^T \\ \sqrt{\mathbf{w}} & \Delta \end{bmatrix} \quad (\text{K.11})$$

the Lanczos algorithm will compute the matrices \mathbf{Q} and \mathbf{T}_2 . Therefore, it will compute the desired $\alpha_k^{N_i}$ and $\beta_k^{N_i}$.

[Golub and Welsch [1969]] showed that $\lim_{N_i \rightarrow +\infty} \alpha_k^{N_i} = \alpha_k$ and $\lim_{N_i \rightarrow +\infty} \beta_k^{N_i} = \beta_k$. The procedures start with a small N_i and N_i is increased until the convergence over α_k and β_k is obtained (that is, $|\beta_k^{(N_i)} - \beta_k^{(N_i-1)}| \leq \varepsilon \beta_k^{(N_i)}$, $k = 0, n-1$); usually convergence is achieved in few iterations.

L Calculation of quadrature nodes and weights

Once the coefficients of the polynomials are found, the roots of the polynomial given by K.1 need to be calculated. [Golub and Welsch [1969]] replace the problem of computing the zeros of orthogonal polynomial with the equivalent eigenvalue problem. The recursive relation can be written in matrix form as

$$x\mathbf{P} = \mathbf{T}\mathbf{P} + p_n \mathbf{e}_{n-1} \quad (\text{K.1})$$

where, $\mathbf{P}(x) = (p_0(x), \dots, p_{n-1}(x))^T$, $\mathbf{e}_{n-1} = (0, \dots, 0, 1)^T$, and \mathbf{T} is a tri-diagonal matrix with elements $T_{ii} = \alpha_i$; $i = 0, n-1$, $T_{i,i+1} = 1$, $i = 0, n-2$ and $T_{i-1,i} = \beta_i$; $i = 2, n-1$.

By using a diagonal matrix \mathbf{D} with $D_{00} = 1$ and $D_{ii} = \sqrt{\beta_i}$; $i = 1, n-1$, the problem can be transformed into solving the eigenvalues and eigenvectors of the Jacobi matrix $\mathbf{J} = \mathbf{D}^{-1} \mathbf{T} \mathbf{D}$. The elements of the matrix \mathbf{J} are $J_{ii} = \alpha_i$; $i = 0, n-1$, and $J_{i,i+1} = \sqrt{\beta_i}$, $i = 0, n-2$. The matrices \mathbf{T} and \mathbf{J} have the same spectrum and eigenvectors, since they are related by a similarity transformation. The matrix \mathbf{J} is

symmetric tri-diagonal, and finding all its eigenvalues and eigenvectors is relatively efficient [Press et al. [1992]].

Moreover, [Wilf [1962]] showed that if \mathbf{v}_i is an eigenvector corresponding to the eigenvalue x_i , normalized so that $\mathbf{v}^T \mathbf{v} = 1$, then the corresponding weight is

$$w_j = \mu_0 v_{j,1}^2 \quad (\text{K.2})$$

where $v_{j,1}$ is the first component of \mathbf{v}_j and

$$\mu_0 = \int_a^b W(x) dx \quad (\text{K.3})$$

The calculation procedure can be further speeded, since only the first element of each eigenvector is required in calculations. It is very important to notice that the function f does not appear in the calculation of the discrete weights and abscissas.

LIST OF FIGURES

1	API gravity	15
2	Enhanced oil Recovery [Larry [1989]]	16
3	Viscosity versus Temperature	17
4	a) Steam injection with vertical wells and b) Steam injection analysis [Baker [1969] and Mandl and Volek [1969]]	17
5	a) SAGD process and b) SAGD analysis	18
6	Water solubility in gas phase for the C1/nC4/H2O system, after [McKetta and Katz [1948]]	18
2.1	Multiphase flash, input/output	32
2.2	Multiphase equilibrium calculation procedure	33
2.3	Sign of the tangent plane distance for the Y8 mixture	35
2.4	Smallest eigenvalue of (eq. 2.16), for the Y8 mixture	37
2.5	Rachford-Rice function	39
2.6	Multiphase flash calculation procedure	43
3.1	a) Euclidean norms vs. iteration level; Y8 mixture at T=335 K and P=200 bar b) Number of iterations for the Y8 mixture at T=335 K in the conventional method	61
3.2	a) Number of iterations for the Y8 mixture at T=335 K using reduction methods, b) Number of Newton iterations for the Y8 mixture at T=335 K in the conventional method	62
3.3	a) Number of iterations for the Y8 mixture at T=335 K; liquid phase stability, b) Number of iterations for the MY10 mixture at saturation points using reduction methods	62
3.4	Number of iterations for the MY10 - 85% CO ₂ mixture at T=322 K	63
3.5	Number of iterations for the Billings mixture at T=366.48 K	63
3.6	Number of iterations for the Kilgren mixture at T=399.82 K	64
3.7	Phase envelopes of Y8, MY10 and MY10+CO ₂ mixtures	68
3.8	Condition number along the isotherm T=335K for the Y8 mixture	70
3.9	Computation time to repeat 10000 times one Newton iteration for the flash problem	72
3.10	Computation time to repeat 10000 times one Newton iteration for the flash problem, MY10+C02	72
3.11	Computation time for the whole two-phase region for the flash problem	74
3.12	Computation time for the whole two-phase region for the flash problem, MY10+C02	74
3.13	Computation time to repeat 10000 times one Newton iteration for the stability problem	75
3.14	Computation time to repeat 10000 times one Newton iteration for the stability problem, MY10+C02	75
3.15	Computation time for the whole P-T window for the stability problem	76
3.16	Computation time for the whole P-T window for the stability problem, MY10+C02	76
3.17	Error v.s. iteration number for various methods, for the Y8 mixture at T=350 K and P=230 bar	77
3.18	Phase envelopes of different mixtures mixed with CO ₂ at T=305.35 K	85
3.19	b)Phase boundaries of a) the Sour gas mixture at T=178.8K b) the four-phase region for the sourgas4 mixture at T=305.35 K	86
3.20	Computational time to repeat 10000 times a Newton iteration	86
3.21	Computational time to repeat 10000 times a Newton iteration	87

3.22 Computational time vs. number of components for the stability analysis of the sour gas mixture (10000 iterations at $T = 178.8$ K and $P = 20$ bar)	87
3.23 Computational time for the entire three phase region	88
3.24 Computational time for a) the entire three phase region for the sour gas mixture; b) the entire four phase region for the sourgas4 mixture	88
3.25 Error vs number of iterations at $P = 69$ bar and 95% of CO_2 for the Maljamar reservoir mixture	89
3.26 Three-phase flash of the Maljamar separator mixture at 95% CO_2	89
3.27 Three-phase flash of the sour gas mixture at $P = 20$ bar	90
3.28 Four-phase flash of the sour gas mixture at 7.4% CO_2	90
3.29 Iterations and function evaluations starting directly with the BFGS method; Y8 mixture at $T = 285$ K	102
3.30 Iterations and function evaluations starting directly with the BFGS method; Y8 mixture at $T = 335$ K	102
3.31 Iterations and function evaluations starting directly with the BFGS method; MY10 mixture at $T = 565$ K	103
3.32 Comparison between the proposed BFGS algorithm and Nocedal L-BFGS-B for the Y8 mixture at $T = 285$ K	103
3.33 Comparison between the proposed BFGS algorithm and Nocedal L-BFGS-B for the Y8 mixture at $T = 335$ K	104
3.34 Iterations and function evaluations after a switch from SSI, Y8 mixture at $T = 285$ K	104
3.35 Iterations and function evaluations after a switch from SSI, Y8 mixture at $T = 335$ K	105
3.36 Iterations and function evaluations after a switch from SSI, MY10 mixture at $T = 565$ K	105
3.37 nl/nv-L compared with alpha/alpha-L; number of function evaluations for the Y8 mixture at $T = 335$ K (after a switch from SSI)	106
3.38 Error versus number of iterations for the Y8 mixture at $T = 335$ K	107
3.39 Error versus number of iterations for the Y8 mixture at $T = 335$ K	107
3.40 Convex example [Conn et al. [2000]]	111
3.41 Non convex example [Conn et al. [2000]]	113
3.42 Hard case example [Conn et al. [2000]]	114
3.43 Phase envelope a) of the MY10 mixture b) of the Oil B mixture mixed with CO_2 at $T = 307.6$ K	117
3.44 Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i/K_i$)	118
3.45 Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i/K_i$)	118
3.46 Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i K_i$)	119
3.47 Number of iterations for the MY10 mixture, stability analysis ($Y_i = z_i K_i$)	119
3.48 Error vs. iteration level for the stability analysis (liquid trial phase, $Y_i = z_i K_i$) of the MY10 mixture at $T = 540.5$ K	120
3.49 Error vs. iteration level for the stability analysis (liquid trial phase, $Y_i = z_i K_i$) of the MY10 mixture	120
3.50 Number of iterations for the MY10 mixture (two-phase flash)	122
3.51 Number of iterations for the MY10 mixture (two-phase flash)	122
3.52 Comparison between various flash calculation algorithms (initialization by K-Wilson) for the Oil B mixture/95% CO_2 at $T = 307.6$ K	123
3.53 Number of iterations (two-and three-phase flashes) for Maljamar reservoir mixture over the Pz-diagram at $T = 305.35$ K	123
3.54 Number of iterations (two-and three-phase flashes) for Maljamar reservoir mixture over the Pz-diagram at $T = 305.35$ K	124
3.55 Number of iterations (two-and three-phase flashes) for Maljamar reservoir mixture over the Pz-diagram at $T = 305.35$ K: NLVM-TR	124

3.56	Number of iterations (two- and three-phase flashes) for Maljamar separator mixture over the P-z diagram at T=305.35 K	125
3.57	Number of iterations (two- and three-phase flashes) for the Oil B over the P-z diagram at T=307.6 K	126
3.58	Number of iterations (two- and three-phase flashes) for the sour gas mixture over the P-z diagram at T= 178.8 K	127
3.59	a) Global minimum of the modified <i>TPD</i> function b) Number of iterations of Newton and TR methods; for sour gas-CO ₂ mixtures at T=178.8 K and P=20 bar	128
3.60	Error vs. iteration level for three-phase flash of the sour gas mixture at T=178.8 K, P=20 bar and 51% moles CO ₂	128
3.61	Computation time to perform Newton iterations through the whole three phase region	133
3.62	a) Convergence properties at 49.6% of CO ₂ b) Number of iterations; for the sour gas mixture on the isobar P=20 bar	134
3.63	Condition number for various methods for the sour gas mixture on the isobar P=20 bar . . .	134
4.1	Feed composition vs. carbon numbers for the S. oil	139
4.2	LFMR for different pressures and temperatures for the S. oil	139
4.3	a) Altered feed composition (oil S) b) LFMR; vs carbon number for the altered oil S	139
4.4	Feed composition of the semi-continuous mixture (Oil S) after cubic spline interpolation on the continuous part	141
4.5	Altered of the S oil (mixture A)	146
4.6	Mole fraction in the vapor phase for mixture A at T=450 K	147
4.7	Error in phase mole fractions for mixture A at T=450 K	147
4.8	Error in phase compositions for mixture A at T=450 K and P=8 bar	148
4.9	Error in phase compositions for mixture B at T=450 K and P=8 bar	148
4.10	Error in phase compositions for mixture C at T=450 K and P=8 bar	149
4.11	a) Phase mole fractions for mixture B + H ₂ O at T = 310 K and b) Errors in phase mole fractions for mixture B + H ₂ O at T=310 K	149
4.12	Phase mole fractions for mixture B + H ₂ O at T = 310 K	150
4.13	Errors in phase compositions for mixture B+H ₂ O at T=310 K and P=6 bar	150
4.14	Errors in phase compositions for mixture B+H ₂ O at T=310 K and P=6 bar	151
4.15	Phase mole fractions mixture A+CO ₂ at T=249 K	151
4.16	Errors in phase mole fraction, mixture A+CO ₂ at T=249 K	152
4.17	Errors in phase compositions for mixture A+CO ₂ at T=249 K and P=29.5 bar	152
4.18	Errors vs. the number of quadrature points, mixture A at T=450 K. a) In phase mole fractions in the two-phase region; b) in compositions at P=8 bar	153
4.19	Absolute errors vs. the number of quadrature points, mixture A+H ₂ O at T=310 K; a) In phase mole fractions in the three-phase region; b) in compositions at P=6 bar	153
4.20	Absolute errors vs. the number of quadrature points, mixture A+CO ₂ at T=249 K; a) In phase mole fractions in the three-phase region; b) in compositions at P=29.5 bar	154
5.1	Example of parameterization of the tie-simplex space	168
5.2	Comparisons between the exact phase envelope and the one obtained with CSAT after [Iranshahr et al. [2012]]	169
5.3	Parameterization of different tie-triangle planes	169
5.4	a) Parametrization of the space b) Status identification, stability required	171
5.5	a) Parametrization of the space b) Status identification, stability skipped	171
5.6	Water saturation at the end of the simulation, SPE3 case	173
5.7	Oil saturation at the end of the simulation, SPE3 case	173

5.8	Total production vs time for the SPE3 case, comparisons between ECLIPSE and the flash program with TPP	174
5.9	Phase envelope BSB comparisons with experimental values	175
5.10	CO ₂ injection process by[Okuno [2009]]	176
5.11	CO ₂ injection process, saturations obtained with AD-GPRS + MFLASH	176
5.12	4 phase case phase p-z diagram	177
5.13	Four-phase simulation of CO ₂ injection	178
5.14	Pressure, bar; for the four phase CO ₂ test case, T=50 days	179
5.15	Three component mixture, at T=520 K	179
5.16	relative permeabilities for the light oil steam injection test case	180
5.17	a)Oil Saturation, b)Gas Saturation for the light oil steam injection case	181
5.18	Water saturation	181
5.19	Gas saturation for the SPE10 case, at T=600 days	182
5.20	Gas saturation for the SPE10 case, third layer, T=600 days	183
5.21	Oil saturation for the SPE10 case, at T=600 days	183
5.22	Oil saturation for the SPE10 case, third layer, T=600 days	183
5.23	Water saturation for the SPE10 case at T=600 days	184
5.24	Plot of different properties in the third layer of the SPE10 case after T=600 days	184
5.25	Saturations for the synthetic SAGD at t=150 days	186
5.26	a) Oil saturation b) Temperature; for the synthetic SAGD at t=150 days	186
5.27	Saturations for the Athabasca oil SAGD at t=20 days	187
5.28	Saturations for the Athabasca oil SAGD at t=30 days	187
5.29	Saturations for the Athabasca oil SAGD at t=360 days	188
5.30	a) Oil saturation b)Temperature for the Athabasca oil SAGD at t=360 days	188
5.31	Gas saturation for the Athabasca oil SAGD at t=410 days	188

LIST OF TABLES

3.1	Feed composition and BIPs for Y8 and MY10 mixtures	67
3.2	Time to perform 10000 times one Newton iteration for MY10 and MY10+CO ₂ mixtures for the flash problem	73
3.3	Time to perform 10000 times one Newton iteration for MY10 and MY10+CO ₂ mixtures for the stability testing problem	76
3.4	Average number of iterations for the np phase-split calculations for the whole np phase region	91
3.5	Computational time per 10000 iterations for conventional and reduction Newton iterations at 200 components	92
3.6	Average number of iterations for four mixtures and various stability testing methods over the P-z diagram (initial guess, $Y_i = z_i K_i$)	121
3.7	Computation time (in seconds) for four mixtures and various stability testing methods over the P-z diagram (initial guess $Y_i = z_i K_i$)	121
3.8	Average number of iterations for four mixtures and various stability testing methods over the P-z diagram (initial guess, $Y_i = z_i / K_i$)	121
3.9	Computation time (in seconds) for four mixtures and various stability testing methods over the P-z diagram (initial guess $Y_i = z_i / K_i$)	121
3.10	Average number of iterations for four mixtures and various flash calculation methods over the P-z diagram (two-phase flashes)	128
3.11	Average number of iterations for four mixtures and various flash calculation methods over the P-z diagram (three phase flashes)	129
3.12	Computation time (in seconds) for four mixtures and various flash calculation methods over the P-z diagram, flash (two- and three-phase flash calculations)	129
3.13	Average number of iterations within the whole three phase region	135
5.1	SPE3, feed properties	172
5.2	SPE3, BIP	173
5.3	BSB fluid properties	174
5.4	BSB reservoir properties	175
5.5	BSB water properties	177
5.6	Fluid properties, light oil	180
5.7	Fluid properties, light steam injection case	180
5.8	Light oil reservoir properties	180
5.9	Fluid properties, medium steam injection case	182
5.10	SPE10 reservoir properties	182
5.11	Fluid properties, synthetic heavy oil	185
5.12	Fluid properties, extra heavy oil	186
5.13	Total time of the simulation for the different cases (in seconds)	189
5.14	Percentage of the equilibrium flash in the simulation CPU time for the different cases	189
5.15	Synthetic oil, SAGD computation time for 90 days of simulation	190

REFERENCES

- N.M. Alsaifi and P. Englezos. Prediction of multiphase equilibrium using the pc-saft equation of state and simultaneous testing of phase stability. *Fluid Phase Equilibria*, 302:169–178, 2011.
- M.N. Ammar and H. Renon. The isothermal flash problem: New methods for phase split calculations. *AIChE J.*, 33:926–939, 1987.
- U. Baer, D. Browarzik, and H. Kehlen. Thermodynamics of semicontinuous mixtures using equations of state with group contributions. *Fluid Phase Equilibria*, pages 9–44, 1997.
- L.E Baker, A.C. Pierce, and K.D. Luks. Gibbs energy analysis of phase equilibria. *SPE Journal*, 22:731–742, 1982.
- P.E Baker. An experimental study of heat flow in steam flooding. *Soc. Petroleum Eng. J.*, 9:89–99, 1969.
- R.A. Behrens and S.I. Sandler. The use of semicontinuous description to model the c7+ fraction in equation of state calculation. *SPE Res. Eng.*, 3:1041–1047, 1986.
- K.M. Brantferger. *Development of a Thermodynamically Consistent, Fully Implicit, Compositional, Equation-Of-State, Steamflood Simulator*. PhD thesis, University of Texas at Austin, 1991.
- R.H Brooks and A.T. Corey. *Hydraulic properties of porous media*. Hydrology Papers, 1964.
- D. Browarzik, M. Kowalewski, and H. Kehlen. Stability calculations of semicontinuous mixtures based on equations of state. *Fluid Phase Equilibria*, 142:149–162, 1998.
- C.G. Broyden. The convergence of a class of double-rank minimization algorithms. *Journal of the Institute of Mathematics and Its Applications*, pages 76–90, 1970.
- J. Burger, P. Sourieau, and M. Combarnous. *Thermal methods of oil recovery*. Technip Editions, 1985.
- Richard H. Byrd, Robert B. Schabel, and Gerald A. Shultz. Approximate solution of the trust region problem by minimization over two-dimensional subspaces. *Mathematical programming*, 40:247–263, 1988.
- W.A. Cañas Marín, J.D. Ortiz-Arango, U.E. Guerrero-Aconcha, and S. Hernandez-Báez. Improved two-sided tangent plane initialization and two-phase-split calculations. *Ind. Eng. Chem. Res.*, 46:5429–5436, 2007.
- M.C.H. Chein, Yardumian H.E., E.Y. Chung, and W.W. Todd. The formulation of a thermal simulation model in a vectorized, general purpose reservoir simulator. Tenth SPE Symposium on Reservoir Simulation, Houston, Texas, 1989.
- G.F Chou and Prausnitz J.M. Adiabatic flash calculations for continuous or semi-continuous mixtures using an equation of state. *Fluid Phase Equilibria*, 30:75–82, 1986.
- O. Cicek. Numerical simulation of steam displacement of oil in naturally fractured reservoirs using fully implicit compositional formulation: A comparative analysis of the effects of capillary and gravitational forces in matrix/fracture exchange term. SPE Annual Technical Conference and Exhibition, Dallas, Texas, 2005.
- O. Cicek and T. Ertekin. Development and testing of a new 3d field scale fully implicit multi-phase compositional steam injection simulator. SPE European 3D Reservoir Modeling Conference, 1996.

- K.H. Coats. Simulation of steamflooding with distillation and solution gas. *Soc. Petroleum Eng. J.*, 16: 235–247, 1976.
- K.H. Coats. A highly implicit steamflood model. *Soc. Pet. Eng. J.*, 18:369–383, 1978.
- A.R. Conn, N.I.M. Gould, and Ph. L. Toint. *Trust-Region Methods*. SIAM/MPS Series on Optimization, 2000.
- R.L. Cotterman and J.M. Prausnitz. Flash calculations for continuous or semicontinuous mixtures using an equation of state. *Ind. Eng. Chem. Proc. Des. Dev.*, 24:434–443, 1985.
- R.L. Cotterman, R. Bender, and J.M. Prausnitz. Phase equilibria for mixtures containing very many components. development and application of continuous thermodynamics for chemical process design. *Ind. Eng. Chem. Proc. Des. Dev.*, 24:194–203, 1985.
- J.L. Creek and J.M. Sheffield. Phase behavior, fluid properties, and displacement characteristics of permian basin reservoir fluid/co₂ systems. *SPE Res. Eng.*, 8:34–42, 1993.
- R.O. Esposito, M. Castier, and F.W. Tavares. Phase equilibrium calculations for semi-continuous mixtures subject to gravitational fields. *Ind. Eng. Chem. Res.*, 39:4415–4421, 2000.
- F.J. Fayers and J.D. Matthews. Evaluation of normalized stone’s methods for estimating three-phase relative permeabilities. *SPE Journal*, 24:224–232, 1984.
- S.A. Feizabadi. *An equation-of-State Based Mathematical Modeling of Four-Phase Flow in Porous Media*. PhD thesis, University of Calgary, 2013.
- A. Firoozabadi and H. Pan. Fast and robust algorithm for compositional modeling: Part i - stability analysis. *SPE Journal*, 7:78–89, 2002.
- A. Firoozabadi, Y. Herim, and D. Katz. Reservoir depletion calculations for gas condensates using extended analyses in the pengrobinson equation of state. *Can. J. Chem. Eng.*, 56:610–615, 1978.
- R. Fletcher. A new approach to variable metric algorithms. *Computer Journal*, 13:317–322, 1970.
- V. Gaganis and N. Varotsis. Non-iterative phase stability calculations for process simulation using discriminating functions. *Fluid Phase Equilibria*, 314:69–77, 2012.
- V. Gaganis and N. Varotsis. An improved bip matrix decomposition method for reduced flash calculations. *Fluid Phase Equilibria*, 340:63–76, 2013.
- V. Gaganis and N. Varotsis. An integrated approach for rapid phase behavior calculations in compositional modeling. *J. Petrol. Sci. Eng.*, 118:74–87, 2014.
- J. Garcia-Sanchez, M.N. Schwartzentrube, M.N. Ammar, and H. Renon. Modeling of multiphase liquid equilibria for multicomponent mixtures. *Fluid Phase Equilib.*, 121:207–225, 1996.
- J. Garcia-Sanchez, G. Eliosa-Jimenez, A. Salas-Padron, O. Hernandez-Garduza, and D. Apam-Martinez. Modeling of microemulsion phase diagrams from excess gibbs energy models. *Chem. Eng. J.*, 84:257–274, 2001.
- T.T. Garipov, M. Karimi-Fard, and H.A. Tchelepi. Coupled geomechanics and flow in fractured media. ECMOR Conference paper, 2012.
- W. Gautschi. Algorithm 726: Orthopol-a package of routines for generating orthogonal polynomials and gauss-type quadrature rules. *ACM Trans. Math. Software*, 20:21–62, 1994.
- W. Gautschi. *Orthogonal Polynomials: Computation and Approximation*. Oxford University Press, 2004.

- D.M. Gay. Computing optimal locally constrained steps. *SIAM J. Sci. and Stat. Comput.*, 2:186–197, 1981.
- P.E. Gill and W. Murray. Newton type methods for unconstrained and linearly constrained optimization. *Math. Program.*, 7:311–350, 1974.
- P.E. Gill and W. Murray. Algorithms for the solution of the nonlinear least-squares problem. *SIAM J. Numer. Anal.*, 15:977–992, 1978.
- D. Goldfarb. A family of variable metric updates derived by variational means. *Mathematics of Computation*, 24:23–26, 1970.
- G.H. Golub and C.F. van Loan. *Matrix Computations*. John Hopkins Univerisy Press, third edition, 1996.
- G.H. Golub and J.H. Welsch. Calculation of gauss quadrature rules. *Math. Comp.*, 2:221–230, 1969.
- R. Gordon. Error bounds in equilibrium statistical mechanics. *J. Math. Phys.*, 9:655–663, 1968.
- S.E. Gorucu and R.T. Johns. Comparison of reduced and conventional phase equilibrium calculations. SPE Reservoir Simulation Symposium, Feb. 18-20, The Woodlands, TX, USA, 2013.
- S.E. Gorucu and R.T. Johns. New reduced parameters for flash calculations based on two-parameter bip formula. *J. Petrol. Sci. Eng.*, 116:50–58, 2014.
- A.S Gow, F.P. Stein, X. Guo, and A. Lucia. Simulation of binary refrigerant mixture vapor-liquid equilibria using a quasi-regular nonrandom fluid theory and local composition mixing rules. *Fluid Phase Equilibria*, 116:60–67, 1996.
- K.B. Haugen and B.L. Beckner. A critical comparison of reduced and conventional eos algorithms. *SPE J.*, 18:378–388, 2013.
- K.B. Haugen, A. Firoozabadi, and L. Sun. Efficient and robust three-phase split computations. *AIChE J.*, 57:2555–2565, 2011.
- M.D. Hebden. An algorithm for minimization using exact second derivatives. Tech. Report TP515. Harwell, England: Atomic Energy Research Establishment, 1973.
- M. Heidari. *Equation of State Based Thermal Compositional Reservoir Simulator for Hybrid Solvent/Thermal Processes*. PhD thesis, Department of Chemical and Petroleum Engineering, Calgary University, 2014.
- R.A. Heidemann and M.L. Michelsen. Instability of successive substitution. *Ind. Eng. Chem. Res.*, 34:958–966, 1995.
- E.M. Hendriks. Reduction theorem for phase equilibria problems. *Ind. Eng. Chem. Res.*, 27:1728–1732, 1988.
- E.M. Hendriks and A.R.D. van Bergen. Application of a reduction method to phase equilibrium calculations. *Fluid Phase Equilibria*, 74:17–34, 1992.
- H. Hoteit and A. Firoozabadi. Simple phase stability-testing algorithm in the reduction method. *AIChE J.*, 52:2909–2920, 2006.
- A. Iranshahr. *Tie-Simplex Method For Thermal-Compositional Simulation*. PhD thesis, Department of Energy Resources Enginnering, Stanford University, 2012.
- A. Iranshahr, D. Voskov, and H. Tchelepi. Tie-simplex parameterization for eos-based thermal compositional simulation. *Soc. Pet. Eng. J.*, 2(15):545–556, 2010a.
- A. Iranshahr, D. Voskov, and H.A. Tchelepi. Generalized negative-flash method for multiphase multicomponent systems. *Fluid Phase Equilibria*, 299:272–284, 2010b.

- A. Iranshahr, D. Voskov, and H. Tchelepi. Gibbs energy analysis: Compositional tie-simplex space. *Fluid Phase Equilib.*, 321:49–58, 2012.
- A. Iranshahr, D. Voskov, and H. Tchelepi. Tie-simplex based compositional space parameterization: Continuity and generalization to multiphase systems. *AIChE J.*, 5(59):1684–1701, 2013.
- K. Ishimoto. One-dimensional fully implicit compositional model for steam flooding. Master’s thesis, University of Texas, Austin, 1985.
- M.R. Islam, S.H. Moussavizadegan, S. Mustafiz, and J.H. Abou-Kassem. *Advanced Petroleum Reservoir Simulation*. Wiley, 2010.
- B.H. Jensen and A. Fredenslund. A simplified flash procedure for multicomponent mixtures containing hydrocarbons and one non-hydrocarbon using two-parameter cubic equations of state. *Ind. Eng. Chem. Res.*, 26:2129–2134, 1987.
- V. John and F. Thein. On the efficiency and robustness of the core routine of the quadrature method of moments (qmom). *Chem. Eng. Sci.*, 75:327–333, 2012.
- P. Kaul and R.L. Thrasher. A parameter-based approach for two-phase-equilibrium prediction with cubic equations of state. *SPE Reservoir Eng.*, 11:273–279, 1996.
- D. Kenyon and G.A. Behie. Third spe comparative solution project: Gas cycling of retrograde condensate reservoirs. *SPE Engineers*, 39:981–997, 1987.
- S.A. Khan, G.A. Pope, and K. Sepehrnoori. Fluid characterization of three-phase CO_2 /oil mixtures. Symposium at the SPE/DOE Enhanced Oil Recovery, Tulsa, Oklahoma, 1992.
- K.H. Kilgren. Phase behavior of a high-pressure condensate reservoir fluid. *J. Petrol. Technol.*, 18:1001–1005, 1966.
- D. Kourounis, D. Voskov, and K. Aziz. Adjoint methods for multicomponent flow simulation. 12th European Conference on the Mathematics of Oil Recovery, Oxford, 2010.
- J.A. Labadie and K.D. Luks. Solidfluid phase equilibria of compositionally complex mixtures: contrast of equilibrium and process treatments. *Fluid Phase Equilibria*, 205:215–232, 2003.
- P.L.C. Lage. The quadrature method of moments for continuous thermodynamics. *Comput. Chem. Eng.*, 31:782–799, 2007.
- C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *J. Res. Bur. Std.*, 45:255–282, 1950.
- A. Lapene. *Etude Experimentale et Numerique de la Combustion In-Situ d’Huiles Lourdes*. PhD thesis, Universite de Toulouse, Institut de Mecanique des Fluides de Toulouse, 2010.
- W.L. Larry. *Enhanced oil recovery*. Technology and Engineering, 1989.
- C.F. Leibovici and J. Neoschil. A new look at the rachford-rice equation. *Fluid Phase Equilibria*, 74:303–308, 1992.
- C.F. Leibovici and D.V. Nichita. A new look at multiphase rachford-rice equations for negative flashes. *Fluid Phase Equilibria*, 267:127–132, 2008.
- K. Levenberg. A method for the solution of certain problems in least squares. *Quart. Appl. Math.*, 2:164–168, 1944.

- G.N. Lewis and M. Randall. *Thermodynamics and the Free Energy of Chemical Substances*. McGraw-Hill Book Co, 1923.
- Y.K. Li and R.T. Johns. Rapid flash calculations for compositional simulation. *SPE Reservoir Evaluation and Engineering*, 8:521–529, 2006.
- Z. Li and A. Firoozabadi. General strategy for stability testing and phase-split calculations in two and three phases. *SPE Journal*, 17:1096–1107, 2012.
- B. Lin, C.F. Leibovici, and S.B. Jørgensen. Optimal component lumping: Problem formulation and solution techniques. *Comput. CHem. Eng.*, 32:1167–1172, 2008.
- C. Lira-Galeana, A. Firoozabadi, and J.M. Prausnitz. Computation of compositional grading in hydrocarbon reservoirs. application of continuous thermodynamics. *Fluid Phase Equilibria*, 102:143–158, 1994.
- D.C. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming*, 45(3):503–528, 1989.
- J.L. Liu and D.S.H. Wong. Rigorous implementation of continuous thermodynamics using orthonormal polynomials. *Fluid Phase Equilibria*, 129:113–127, 1997.
- J. Lohrenz, B.G. Bray, and C.R. Clark. Calculating viscosities of reservoir fluids from their compositions. *Journal of Petroleum Technology*, 915:1171–1176, 1964.
- A. Lucia and D. Liu. An acceleration method for dogleg methods in simple singular regions. *Ind. Eng. Chem. Res.*, 37:1358–1363, 1998.
- A. Lucia and F. Yang. Multivariable terrain methods. *AIChE J.*, 49:2553–2563, 2003.
- A. Lucia, X. Guo, and X. Wang. Process simulation in the complex domain. *AIChE J.*, 39:461–470, 1993.
- A. Lucia, L. Padmanabhan, and S. Venkataraman. Multiphase equilibrium flash calculations. *Comp. and Chem. Eng.*, 24:2557–2569, 2000.
- A. Lucia, P.A. DiMaggio, M.L. Bellows, and L.M. Octavio. The phase behavior of n-alkane systems. *Comput. Chem. Eng.*, 29:2363–2379, 2005.
- A. Lucia, B.M. Bonk, R.R. Waterman, and A. Roy. A multi-scale framework for multi-phase equilibrium flash. *Comput. Chem. Eng.*, 36:79–98, 2012.
- Sh. Luo and M.A. Barrufet. The effect of water-in-oil solubility on oil reservoir in the steam-injection process. *SPE Reservoir Evaluation and Engineering*, 8:528–533, 2005.
- G. Mandl and C.W. Volek. Heat and mass transport in steam-drive processes. *Soc. Petroleum Eng. J.*, 9: 59–79, 1969.
- D. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *SIAM J. Appl. Math.*, 11: 431–441, 1963.
- J.L.Jr McKetta and D.L. Katz. Methane-n-butane-water system in two and three-phase regions. *Ind. Eng. Chem.*, 40:835, 1948.
- R.K. Mehra, R.A. Heidemann, and K. Aziz. Computation of multiphase equilibrium for compositional simulation. *SPE Journal*, 22:61–68, 1982.
- R.K. Mehra, R.A. Heidemann, and K. Aziz. An accelerated successive substitution algorithm. *Can. J. Chem. Eng.*, 61:590–596, 1983.

- R.S. Metcalfe and L. Yarborough. The effect of phase equilibria on the co₂ displacement mechanism. *SPE Journal*, 19:242–252, 1979.
- M.L. Michelsen. The isothermal flash problem. part ii. phase-split calculation. *Fluid Phase Equilibria*, 9: 21–40, 1982a.
- M.L. Michelsen. The isothermal flash problem. part i. stability. *Elsevier Scientific Publishing Company*, 9: 1–20, 1982b.
- M.L. Michelsen. Simplified flash calculations for cubic equation of state. *Ind. Eng. Chem. Process Des. Dev.*, 25:184–188, 1986.
- M.L. Michelsen. Phase equilibrium calculations. what is easy and what is difficult ? *Comput. Chem. Eng.*, 16:19–29, 1992.
- M.L. Michelsen. Calculation of multiphase equilibrium. *Comput. Chem. Eng.*, 18:545–550, 1994.
- M.L. Michelsen and J.M. Mollerup. *Thermodynamic Models: Fundamentals and Computational Aspects*. Tie-Line Publications, second edition, 2007.
- M.L. Michelsen, E.H. Stenby, I. Aavatsmark, A. Belkadi, E. Vignati, W. Yan, E. Moggia, and A. Cominelli. Speeding up compositional reservoir simulation through an efficient implementation of phase equilibrium calculation. *SPE 163598*, 2013a. SPE Reservoir Simulation Symposium, 18-20 February, The Woodlands, Texas, USA.
- M.L. Michelsen, W. Yan, and E.H. Stenby. A comparative study of reduced variables based flash and conventional flash. *SPE Journal*, 18:952–959, 2013b.
- S. Mohebbinia, K. Sepehrnoori, and R.T. Johns. Four-phase equilibrium calculations of carbon dioxide/hydrocarbon/water systems with a reduced method. *SPE J.*, 18:943–951, 2013.
- R. Monroy-Loperena. A note on the analytical solution of cubic equations of state in process simulation. *Ind. Eng. Chem. Res.*, 51:6972–6976, 2012.
- J.E.P. Monteagudo, P.L.C. Lage, and K. Rajagopal. Towards a polydisperse molecular thermodynamic model for asphaltene precipitation in live-oil. *Fluid Phase Equilibria*, 187-188:443–471, 2001a.
- J.E.P. Monteagudo, G.I. Rochocz, P.L.C. Lage, and K. Rajagopal. Extending the gibbs tangent plane semicontinuous mixtures. *Fluid Phase Equilibria*, 190:1–13, 2001b.
- F. Montel and P. Gouel. A new lumping lumping scheme of analytical data for compositional studies. SPE 13119, the 59th ATCE held in Houston, TX, September 16-19, 1984.
- J.J. Moré and D.C. Sorensen. Computing a trust region step. *SIAM J. Sci. Statist. Comput.*, 4:553–572, 1983.
- J.J. Moré and D.C. Sorensen. Generalizations of the trust region problem. *Optim. Method. Softw.*, 2:189–209, 1993.
- T.M.J. Newley and Jr. Merrill. Pseudocomponent selection for compositional simulation. *SPE Res. Eng.*, 6: 490–496, 1991.
- L.X. Nghiem and R.A. Heidemann. *General acceleration procedure for multiphase flash calculation with application to oil-gas-water systems*. Computer Modelling Group, 1982, 1982.
- L.X. Nghiem, K. Aziz, and Y.K. Li. A robust iterative method for flash calculations using the soave-redlich-kwong or the peng-robinson equation of state. *SPE Journal*, 23:521–530, 1983.

- D.V. Nichita. Critical points calculation from cubic eos using the reduction method. *Fluid Phase Equilibria*, pages 223–231, 2005.
- D.V. Nichita. A reduction method for phase equilibrium calculation with cubic equations of state. *Braz. J. Chem. Eng.*, 23:427–434, 2006a.
- D.V. Nichita. A new method for critical points calculation from cubic eos. *AIChE J.*, 52:1220–1227, 2006b.
- D.V. Nichita. Phase envelope construction for mixtures with many components. *Energy and Fuels*, 22:488–495, 2008.
- D.V. Nichita and A. Graciaa. A new reduction method for phase equilibrium calculations. *Fluid Phase Equilibria*, 302:226–233, 2011.
- D.V. Nichita and C.F. Leibovici. An analytical consistent pseudo-component delumping procedure for equations of state with non-zero binary interaction parameters. *Fluid Phase Equilibria*, 245:71–82, 2006.
- D.V. Nichita and C.F. Leibovici. A rapid and robust method for solving the rachford-rice equation using convex transformations. *Fluid Phase Equilibria*, 353:38–49, 2013.
- D.V. Nichita and F. Minescu. Efficient phase equilibrium calculation in a reduced flash context. *Can. J. Chem. Eng.*, 82:1225–1238, 2004.
- D.V. Nichita and M. Petitfrere. A reduction method for phase stability analysis. *Fluid Phase Equilibria*, 358: 27–39, 2013.
- D.V. Nichita, F. Minescu, and I. Cretu. Regression analysis and c7+ description for accurate pvt data calculation with equations of state. *Petrol. Geosci.*, 7:181–189, 2001.
- D.V. Nichita, S. Gomez, and E. Luna. Phase stability analysis with cubic equations of state using a global optimization method. *Fluid Phase Equilibria*, 194-197:411–437, 2002.
- D.V. Nichita, D. Broseta, and de Hemptinne. Multiphase equilibrium calculation using reduced variables. *Fluid Phase Equilibria*, 246:15–27, 2006a.
- D.V. Nichita, D. Broseta, and C.F. Leibovici. Consistent delumping of multiphase flash results. *Comput. Chem. Eng.*, 30:1026–1037, 2006b.
- D.V. Nichita, D. Broseta, J.C de Hemptinne, and V. Lachet. Efficient phase equilibrium calculation for compositional simulation : The direct reduced flash. *Taylor and Francis*, pages 315–342, 2007a.
- D.V. Nichita, D. Broseta, J.C de Hemptinne, and V. Lachet. Efficient phase equilibrium calculation for compositional simulation: the direct reduced flash. *Petrol. Sci. Technol.*, 25:315–342, 2007b.
- D.V. Nichita, D. Broseta, and F. Montel. Calculation of convergence pressure/temperature and stability test limit loci of mixtures with cubic equations of state. *Fluid Phase Equilibria*, 261:176–184, 2007c.
- J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of Computation*, 35:773–782, 1980.
- J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer, second edition, 2006.
- R. Okuno. *Modeling of Multiphase Behavior for Gas Flooding Simulation*. PhD thesis, The University of Texas at Austin, 2009.
- R. Okuno, R.T. Johns, and K. Sepehrnoori. Three-phase flash in compositional simulation using a reduced method. *SPE Journal*, 15:689–703, 2010a.

- R. Okuno, R.T. Johns, and K. Sepehrnoori. Application of a reduced method in compositional simulation. *SPE Journal*, 15:39–49, 2010b.
- R. Okuno, R.T. Johns, and K. Sepehrnoori. A new algorithm for rachford-rice for multiphase compositional simulation. *SPE Journal*, 15:313–325, 2010c.
- F.M. Orr, A.D. Yu, and C.L. Lien. Phase behavior of co₂ and crude oil in low-temperature reservoirs. *SPE Journal*, 21:480–492, 1981.
- H. Pan and A. Firoozabadi. Fast and robust algorithm for compositional modeling: Part ii-two-phase flash calculations. *SPE Journal*, 8:380–391, 2003.
- H. Pan and H. Tchelepi. Reduced variable method for general-purpose compositional reservoir simulation. SPE-131737-MS, International Oil and Gas Conference and Exhibition in China, 8-10 June, Beijing, 2011a.
- H. Pan and H. Tchelepi. Compositional flow simulation using reduced-variables and stability-analysis bypassing. SPE 142189, SPE Reservoir Simulation Symposium, 21-23 February, The Woodlands TX, 2011b.
- J. Park and H. Kim. Continuous thermodynamics of phase equilibria using the beta distribution function and an equation of state. *Korean J. Chem. Eng.*, 10:71–77, 1993.
- C.A. Passut and R.P. Danner. Correlation of ideal gas enthalpy, heat capacity, and entropy. *Ind. Eng. Chem. Process Des. Develop.*, 11(4):543–546, 1972.
- D.Y. Peng and D.B. Robinson. A new two-constant equation of state. *Industrial and Engineering Chemistry*, page 5964, 1976.
- M. Petitfrere and D.V. Nichita. Robust and efficient trust-region based stability analysis and multiphase flash calculations. *Fluid Phase Equilib.*, 362:5168, 2014a.
- M. Petitfrere and D.V. Nichita. A comparison of conventional and reduction approaches for phase equilibrium calculations. submitted to *Fluid Phase Equilibria*, 2014b.
- M. Petitfrere, D.V. Nichita, and F. Montel. Multiphase equilibrium calculations using the semi-continuous thermodynamics of hydrocarbon mixtures. *Fluid Phase Equilib.*, 362:365–378, 2014.
- W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes*. Cambridge University Press, second edition, 1992.
- C.P. Rasmussen, K. Krejbjerg, M. Michelsen, and K.E. Bjurstrøm. Increasing the computational speed of flash calculations with applications of compositional, transient simulations. *SPE Reservoir Evaluation and Engineering*, 9:32–38, 2006.
- M. Ratzsch and H. Kehlen. Continuous thermodynamics of complex mixtures. *Fluid Phase Equilibria*, 14: 225–234, 1983.
- M. Ratzsch, H. Kehlen, and J. Schumann. Flash calculations for a crude oil by continuous thermodynamics. *Chem. Eng. Commun.*, 71:113–125, 1988.
- Otto Redlich and J.N.S. Kwong. On the thermodynamics of solutions. *Chem. Rev.*, pages 233–244, 1949.
- R.C. Reid, J.M. Prausnitz, and B.E. Poling. *Properties of Gas and Liquids*. Mc-Graw Hill, 1987.
- R. Risnes, V. Dalen, and J.I. Jensen. Phase equilibrium calculations in the near-critical region. European Symposium on Enhanced Oil Recovery, Bournemouth, Great Britain, 1981.

- D.B. Robinson and D.Y. Peng. The characterization of the heptanes and heavier fractions for the gpa peng-robinson programs. Gas Processors Association, Research Report RR-28, 1978.
- D.B. Robinson, H. Kalra, and H. Rempis. The equilibrium phase properties of a synthetic sour gas mixture and a simulated natural gas mixture. Tulsa: GPA Report, 1978.
- G.L. Rochocz, M. Castier, and S.I. Sandler. Critical point calculations for semi-continuous mixtures. *Fluid Phase Equilibria*, 139:137–153, 1997.
- R.C. Rodrigues, V.R.R. Ahon, and P.L.C. Lage. An adaptive characterization scheme for the simulation of multistage separation of continuous mixtures using the quadrature method of moments. *Fluid Phase Equilibria*, 318:1–12, 2012.
- C.H. Roland. Vapor-liquid equilibria for natural gas - crude oil mixtures. *Ind. Eng. Chem.*, 37:930–936, 1945.
- R.A. Sack and A.F. Donovan. An algorithm for gaussian quadrature given modified moments. *Numer. Math.*, 18:465–478, 1972.
- D. F. Shanno. Conditioning of quasi-newton methods for function minimization. *Mathematics of Computation*, 24:647–656, 1970.
- J.L. Shelton and L. Yarborough. Multiple phase behavior in porous media during co2 or rich-gas flooding. *J. Petrol. Technol.*, 29:1171–1178, 1977.
- S.K. Shibata, R.A. Behrens, and S.I. Sandler. Phase equilibrium calculations for continuous and semicontinuous mixtures. *Chem. Eng. Sci.*, 42:1977–1987, 1986.
- G.A. Shultz, R.B. Schabel, and R.H. Byrd. A family of trust-region-based algorithms for unconstrained minimization with strong global convergence properties. *SIAM J. Numer. Anal.*, 22:47–67, 1985.
- N.D. Shutler. Numerical three-phase model of the linear steamflood process. *Soc. Petroleum Eng.*, 9:232–246, 1969.
- N.D. Shutler. Numerical three-phase model of the two-dimensional steamflood process. *Soc. Petroleum Eng.*, 10:405–417, 1970.
- G Soave. Equilibrium constants from a modified redlichkwong equation of state. *Chem. Eng. Sci.*, 27:1197–1203, 1972.
- A.G. Spillette and R.L. Nielsen. Two-dimensional method for predicting hot waterflood recovery behavior. *J. Petroleum Technol.*, 20:627–538, 1968.
- M.A. Stadtherr, C.A. Schnepper, and J.F. Brennecke. Robust phase stability analysis using interval methods. *AIChE Symp. Ser.*, 91:356–359, 1995.
- T. Steihaug. The conjugate gradient method and trust regions in large scale optimization. *SIAM J. Numer. Anal.*, 20:626–637, 1983.
- H.L. Stone. Estimation of three-phase relative permeability and residual oil data. *Journal of Canadian Petroleum Technology*, 12:53–61, 1973.
- A.C. Sun and W.D. Seider. Homotopy-continuation method for stability analysis in the global minimization of the gibbs free energy. *Fluid Phase Equil.*, 103:213–249, 1995.
- H. Tchelepi and Y. Zhou. Multi-gpu parallelization of nested factorization for solving large linear systems. Society of Petroleum Engineers, 2013.

- A. Trangenstein. Customized minimization techniques for phase equilibrium computations in reservoir simulation. *Chem. Eng. Sci.*, 42:2847–2863, 1987.
- A. Varavei. *Development of an Equation-of-State Thermal Flooding Simulator*. PhD thesis, The University of Texas at Austin, 2009.
- A. Varavei and K. Sepehrnoori. An eos-based compositional thermal reservoir simulator. SPE 119154, Reservoir Simulation Symposium, The Woodlands, Texas, 2009.
- J. Vidal. *Thermodynamics applications in chemical engineering and the petroleum industry*. Institut Français du Pétrole Publication, 1997.
- P.K.W. Vinsome. A numerical description of hot-water and steam drives by the finite-difference method. Annu. Fall Meeting of Soc. Petroleum Eng., SPE Paper No 5248, 1974.
- O. Volkov and D. Voskov. Advanced strategies of forward simulation for adjoint-based optimization. SPE 163592, RSS, Woodland, TX, 2013.
- D. Voskov and H. Tchelepi. Compositional space parameterization for miscible displacement simulation. *Transport in Porous Media*, 1(75):111–128, 2008.
- D. Voskov and H. Tchelepi. Compositional space parameterization: Multicontact miscible displacements and extension to multiple phases. *Soc. Pet. Eng. J.*, 3(14):441–449, 2009a.
- D. Voskov and H. Tchelepi. Compositional space parameterization: Theory and application for immiscible displacements. *Soc. Pet. Eng. J.*, 3(14):431–440, 2009b.
- D. Voskov and H. Tchelepi. Tie-simplex based mathematical framework for thermodynamical equilibrium computation of mixtures with an arbitrary number of phases. *Fluid Phase Equilib.*, 1-2(283):111, 2009c.
- D. Voskov and Y. Zhou. Technical description of ad-gprs. energy resources engineering. Stanford University, 2012.
- D. Voskov, R. Younis, and H. Tchelepi. Comparison of nonlinear formulations for isothermal compositional flow simulation. SPE 118996. Presented at SPE reservoir simulation symposium, Woodlands, TX, 2009.
- D. Voskov, H. Tchelepi, and R. Younis. General nonlinear solution strategies for multi-phase multi-component eos based simulation. SPE 118996, RSS, Woodland, TX, February, 2009d.
- K.C. Wang, Q. and Chao. Vapor-liquid and liquid-liquid equilibria and critical states of water + n-decane mixtures. *Fluid Phase Equilibria*, 59:207, 1990.
- J.C. Wheeler. Modified moments and gaussian quadratures. *Rocky Mt. J. Math.*, 4:287–296, 1974.
- C.H. Whitson. Characterizing hydrocarbon plus fractions. *SPE Journal*, 23:683–694, 1983.
- C.H. Whitson and M.L. Michelsen. The negative flash. *Fluid Phase Equilibria*, 53:51–72, 1990.
- H.S. Wilf. *Mathematics for the Physical Sciences*. Wiley, 1962.
- B. Willman and Teja A.S. Continuous thermodynamics of phase equilibria using a multivariate distribution function and an equation of state. *AIChE J.*, 32:2067–2076, 1986.
- B. Willman and Teja A.S. Prediction of dew points of semicontinuous natural gas and petroleum mixtures. 1. characterization by use of an effective carbon number and ideal solution prediction. *Ind. Eng. Chem. Res.*, 26:948–952, 1987a.

- B. Willman and Teja A.S. Prediction of dew points of semicontinuous natural gas and petroleum mixtures. 2. nonideal solution calculations. *Ind. Eng. Chem. Res.*, 26:953–957, 1987b.
- G. Wilson. A modified redlich-kwong equation of state, application to general physical data calculations. paper no. 15C presented at the AIChE 65th National Meeting, Cleveland, Ohio, May 4-7, 1969.
- T.W. Wong, A. Firoozabadi, and K. Aziz. Relationship of the volume-balance method of compositional simulation to the newton-raphson method. *SPE J.*, 5:415–422, 1990.
- W. Yan and E.H. Stenby. On multiphase negative flash ideal solutions. *Fluid Phase Equilibria*, pages 41–47, 2012.
- L. Yarborough. Vapor-liquid equilibrium data for multicomponent mixtures containing hydrocarbon and non-hydrocarbon components. *J. CHem. Eng. Data*, 19:129–133, 1972.
- R.M. Younis. *Advances in Modern Computational Methods for Nonlinear Problems; A Generic Efficient Automatic Differentiation Framework, and Nonlinear Solver that Converge All The Time*. PhD thesis, Stanford University, 2009.
- R.M. Younis and K. Aziz. Parallel automatically differentiable data-types for next-generation simulator development. Proceedings of Reservoir Simulation Symposium, Houston, TX, February, 2007.
- C. Yuan and G. Pope. A new method to model relative permeability in compositional simulators to avoid discontinuous changes caused by phase-identification problems. *SPE Journal*, 17:1221–1230, 2012.
- R. Zaydullin, D. Voskov, and H.A. Tchelepi. Nonlinear formulation based on an equation-of-state free method for compositional flow simulation. *SPE Journal*, 2(18):264–273, 2013.
- R. Zaydullin, D. Voskov, S.C. James, H. Henley, and A. Lucia. Fully compositional and thermal reservoir simulation. *Comp. and Chem. Eng.*, 63:51–65, 2014.
- Y. Zhou. *Parallel general-purpose reservoir simulation with coupled reservoir models and multisegment wells*. PhD thesis, Stanford University, 2012.
- Y. Zhou, H.A. Tchelepi, and B.T. Mallison. Automatic differentiation framework for compositional simulation on unstructured grids with multi-point discretization schemes. SPE Reservoir Simulation Symposium, 21-23 February, The Woodlands, Texas, 2011.