

A New Preconditioner for Iterative Phase Splitting Routines

Cand. Scient. Thesis in Computational Science

Jan Nordbotten

Department of Mathematics
University of Bergen



4th September 2002

Preface

The purpose of this work is to propose a new, faster, non-iterative phase split estimate, and compare it with the previously published method of Wang and Stenby [9], in terms of their suitability as possible preconditioners for an iterative scheme.

This has been done as a Masters Degree project at the Department of Mathematics, University of Bergen.

I thank all the people who have been supportive during my studies, especially my supervisors Magne Espedal and Helge Dahle. My thanks also go to all the people whom have distracted me from my studies, so that my mind had a chance to take a break.

Jan Nordbotten
Bergen, August 2002

Contents

Preface	II
Objective	1
1 Introduction	3
2 Reservoir simulation	7
2.1 The SOM	8
2.2 Weakness of sequential approaches	9
3 Non-iterative flash calculation	11
3.1 Derivation	12
3.2 Discussion	13
4 A new approximate flash calculation	15
4.1 Derivation	16
4.2 Evaluation of $\frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha}$	17
4.3 New estimate for $\frac{\partial N_i^\alpha}{\partial N_i}$ and $\frac{\partial N_i^\alpha}{\partial p}$	18
4.4 Complexity of the new procedure	20
4.5 Discussion	21
5 Numerical experiments	23
5.1 Relevant data	24
5.2 Experiments	24
5.3 Discussion of results	29
5.3.1 Composition changes	29
5.3.2 Pressure changes	30
5.3.3 Several steps	30
6 Conclusion and further work	33
6.1 Conclusion	34
6.2 Further work	34

Appendix A	37
Nomenclature	38
Appendix B	41
Overview of test code	42
Bibliography	43

Objective

In a reservoir simulator, equations relating physical variables in time and space are formulated, and to the best of our abilities solved. In a full compositional model, information concerning the balance of components among phases is of great importance when computing fluid properties and flow. Therefore it is common to update the phase split often. In a reservoir simulator, we might have millions of data points, and since the equations governing the phase split must be solved by iterative routines, phase split calculations represent a major computational cost. This is the motivation for developing and investigating preconditioners for iterative phase split routines.

In this work, our main objective is to construct a new preconditioner for iterative phase splitting routines. We first investigate an existing preconditioner, [9], and recognize its weakness in its poor order of execution time. The work required is proportional to $O(n_c^3)$, where n_c is the number of components in the mixture. The new preconditioners developed have a complexity of $O(n_c^2)$.

Chapter 1

Introduction

The physical state in a given point in space and time can be described by variables. In the case of an oil reservoir, the main variables are:

The number of moles of the chemical component i in phase α , N_i^α

Temperature, T

Pressure of phase α , p^α

Porosity, ϕ

Saturation of phase α , S^α

Permeability, K

The Greek letter α (and later β) denotes any of the phases gas (g), oil (o) or water (w) and $i = 1 \dots n_c$. The upper bound n_c is the number of components in the mixture.

Given the above variables, we have knowledge of the amount of oil and gas present. In the process of extracting oil from a reservoir, the variables above will vary with time, for all points in the reservoir, and knowing how they vary is of great value if our objective is to optimize oil production.

The characteristics of the components, and the above variables enable the calculation of the properties:

Mass density of phase α , ρ^α

Concentration of component i in phase α , c_i^α

Viscosity of phase α , μ^α

Relative permeability of phase α , k_{r_α}

Capillary pressures between phases α and β , $p_{c_{\alpha,\beta}}$

Darcy velocity of phase α , \vec{u}^α

The equations needed to calculate all the relations between the above variables can be divided into algebraic and differential. It is common to divide the variables into two groups, primary and secondary, where the primary variables are calculated from the differential equations and the secondary from the algebraic. The primary variables are most often selected from the first of the above lists. We refer to [1, 8, 10, 11] for details. The main differential equations will be given in the following.

We know that mass must be conserved, so for any volume CV with interface CS :

$$\frac{\partial}{\partial t} \int_{CV} N_i dV = - \int_{CS} F_i \cdot d\vec{S} + \int_{CV} q_i dV \quad (1.1)$$

Where $F_i = \sum_{\alpha} c_i^{\alpha} \rho^{\alpha} u^{\alpha}$ is the molar flux of component i , \vec{S} is the outward normal vector to the interface CS , and q_i is a source/sink term. If the volume CV is to vary with time, the evaluation of the surface integral on the right hand side must reflect this.

Temperature is governed in time by the heat flow equation:

$$\frac{\partial}{\partial t} \int_{CV} (\rho u) dV = \int_{CS} (k \nabla T) \cdot d\vec{S} - \int_{CS} (h \rho \vec{u}) \cdot d\vec{S} + \int_{CV} q dV \quad (1.2)$$

$$\rho u = \sum_{\alpha=g,o,w} \phi S^{\alpha} u^{\alpha} \rho^{\alpha} + u^r \rho^r (1 - \phi) \quad (1.3)$$

$$h \rho \vec{u} = \sum_{\alpha=g,o,w} h^{\alpha} \rho^{\alpha} \vec{u}^{\alpha}$$

where ρ and u are the mass density and internal energy, and the superscript r represents the rock phase. Further, k is heat conductivity and h is enthalpy. Enthalpy is defined as the sum of the internal energy and the work potential (pressure):

$$h = u + p \quad (1.4)$$

The first term on the right hand side of Equation (1.2) represents heat flow (and is a variant of Fourier's law), while the second term represents the energy carried by the fluid flow. In addition to the above equations, we have the usual expression for the Darcy phase velocity \vec{u}^{α} :

$$\vec{u}^{\alpha} = -K \frac{k_{r\alpha}}{\mu^{\alpha}} (\nabla p^{\alpha} - \rho^{\alpha} g) \quad \forall \alpha \quad (1.5)$$

Where g is the gravitational constant. The total Darcy velocity is the sum of the phase velocities:

$$\vec{u} = \vec{u}^o + \vec{u}^g + \vec{u}^w \quad (1.6)$$

These equations are often manipulated to provide equations directly for the pressures, when one of these these is used as a primary variable [2]. To calculate the molar masses N_i^{α} , fugacity balance may be used:

$$f_i^{\alpha} = f_i^{\beta} \quad \forall i, \alpha, \beta \quad (1.7)$$

where $f_i^{\alpha} = f_i^{\alpha}(p, T, N_1^{\alpha}, N_2^{\alpha}, \dots, N_{n_c}^{\alpha})$ where n_c is the number of components in the system. The fugacity is defined more precisely in Section 4.2, and more details on thermodynamics

can be found in [4]. Equation (1.7) will be the basis for most of this work, since iterative routines often take an initial estimate of the distribution of components between phases, and by applying an iterative scheme based on some form of equation (1.7) produce a new phase split.

In the next chapter the basin simulator SOM is introduced, and the preconditioners developed in this work are envisioned to be implementable in SOM. In Chapter 3, a flash calculation preconditioner from literature is presented, followed by the development of two new preconditioners in Chapter 4. Numerical experiments on these methods is conducted in Chapter 5, and Chapter 6 contains conclusions and suggestions for further work. Appendices contain nomenclature and an overview of the test code.

Chapter 2

Reservoir simulation

2.1 The SOM

The model described in Chapter 1 has been implemented with various simplifications in different simulators. This section will briefly describe one of them, the “Secondary Oil Migration” simulator SOM, see [3].

The SOM is a basin simulator, built on the equations in Chapter 1. The full details of the simulator are not of interest for us as it is only introduced to provide a general setting for the further work. Built in C++ SOM is module based, and object oriented, so that modules may easily be modified or changed, as done in [8]. We see that this implies that the SOM does not have just one numerical discretization of the equations of Chapter 1, but may have rather different discretizations in different modules. However, the general framework of the simulator is not altered.

In [8], the main routine is described as initializing five classes:

- TimeStepping
- Basin
- Geometry
- InputData
- InitValues

We will pause a moment to review the functions of these classes. We start from the bottom of the list.

InitValues and InputData

These initiation classes perform the ordinary tasks of reading simulation data from file and/or keyboard.

Geometry

The Geometry class contains grid information, as well as boundary conditions. The boundary conditions are stated as pressure and temperature restrictions.

Basin

Basin contains two main modules, a Rock module and a Fluid module. Together these contain information as to how the equations of state, viscosity, compressibility, relative permeability, capillary pressures, etc. are related.

TimeStepping

When reading the description of the class TimeStepping we see the following procedure is used:

Direct quote:

- update the top depth (UpdateTopDepth),
- calculate the transmissibilities (CalcGeomDepCoup),
- calculate the temperature (DoTemperature),
- calculate the secondary variables (CalcPhaseSplit),
- calculate the water pressure and the molar masses (DoPressMolMass),
- calculate the secondary variables (CalcPhaseSplit),
- calculate the new time step.

End quote.

Note that CalcPhaseSplit is called twice in each time-step. We also see as a consequence that not all variables that appear in the fugacity equations are altered between each call to CalcPhaseSplit. This is reflected in the numerical experiments implemented later.

With the knowledge that the phase split must be updated in every data point twice each time-step, as well as knowledge that iterative routines are used to calculate fugacities, we recognize the value of accurate initial guesses for the iterative routines.

Commonly, one has used either experimentally derived relationships, as in [6], or the previous time-step as initial guesses. In this work we will attempt to find fast and reliable estimates for the updated phase split.

2.2 Weakness of sequential approaches

Before we continue with the task outlined in the last paragraph above, we will pause a moment to review the drawback of a sequential procedure as described above.

Phase composition, temperature, pressure, and other physical variables are continuous variables, and the reason for the complexity of the problem is indeed the coupling of all these variables. With sequential solution techniques, we essentially assume that if we only take small enough steps, we may ignore these couplings. In fact, we are so confident in this assumption, that no back substitution or iterations are made when calculating new time-steps.

Of special interest for us is the fact that the primary variables, when updated in the classes `DoTemperature` and `DoPressMolMass`, are as a consequence inaccurate and that this implies that even though we may make a phase split that is correct according to the primary variables, it will not correspond to physical reality.

Just as important as accuracy is the concepts of stability and convergence. Due to the inaccuracies outlined above, we immediately understand that without stability, our methods are not reliable. Also, if we know that our methods converge, we may adjust our step size to control the error.

One realizes that accurate direct schemes for calculating the phase splits would enable us to avoid some of the sequentialization from Section 2.1. This is because one could insert the phase split solution into the mass transport, pressure, and heat equations so that the solution to these equations took into consideration the thermodynamics of the system.

Chapter 3

Non-iterative flash calculation

3.1 Derivation

We will in the following exclusively concern ourselves with the phase split. First we present a non-iterative flash calculation (NIFC), as presented by Wang and Stenby in [9].

We do the linear approximation:

$$dN_i^\alpha = \left(\frac{\partial N_i^\alpha}{\partial p}\right)_{N,T} dp + \left(\frac{\partial N_i^\alpha}{\partial T}\right)_{N,p} dT + \sum_{j=1} \left(\frac{\partial N_i^\alpha}{\partial N_j}\right)_{p,T,N_{k \neq j}} dN_j \quad \forall i \quad (3.1)$$

Neglecting temperature variation in the following, we further use:

$$N_{i,new}^\alpha = N_{i,old}^\alpha + dN_i^\alpha \quad (3.2)$$

$$\ln f_j^o = \ln f_j^g \quad (3.3)$$

$$N_i^o + N_i^g = N_i$$

By differentiating the last two equations, we obtain for the variables $\frac{\partial N_k^o}{\partial p}$ and $\frac{\partial N_k^g}{\partial N_j}$:

$$\sum_{k=1}^{n_c} \left(\frac{\partial \ln f_i^o}{\partial N_k^o} + \frac{\partial \ln f_i^g}{\partial N_k^g} \right) \frac{\partial N_k^o}{\partial p} = \frac{\partial \ln f_i^g}{\partial p} - \frac{\partial \ln f_i^o}{\partial p} \quad \forall i \quad (3.4)$$

and

$$\sum_{k=1}^{n_c} \left(\frac{\partial \ln f_i^o}{\partial N_k^o} + \frac{\partial \ln f_i^g}{\partial N_k^g} \right) \frac{\partial N_k^o}{\partial N_j} = \frac{\partial \ln f_i^g}{\partial N_j^g} \quad \forall i, j \quad (3.5)$$

The linear system defined by Equation (3.4) is solvable in $O(n_c^3)$ operations, and the decomposition may be used again in Equation (3.5) since the left hand side is equal. This allows for a total computational time of $O(n_c^3)$. Note that calculating the compressibility factor Z for a phase (see Equation (4.9) below) requires $O(n_c^2)$ operations, and that the further calculations needed for fugacity can be done in order $O(1)$. When calculating the derivatives numerically, the variables A, B (Equation (4.10)) can be updated for the perturbed compositions in $O(n_c)$ operations, and hence the compressibility for the $n_c + 1$ compositions needed can be computed in $O(n_c^2)$. Finally, we need to calculate n_c^2 derivatives, again these take $O(1)$ time. In total, estimating the Jacobian $J^{n_c \times n_c}$ is done in $O(n_c^2)$.

3.2 Discussion

We observe that the vectors appearing on the left hand side of equations (3.4) and (3.5) represent the true gradients of the implicit functions $N_i^\alpha = N_i^\alpha(P, N_1 \dots N_{n_c})$. This means that if we let $dN_i^\alpha \rightarrow 0$ in Equation (3.2), the equations (3.2) - (3.5) converge to the actual solution.

The error occurring using equations (3.2) - (3.5) is therefore exclusively due to the non-linearity of $N_i^\alpha(P, N_1 \dots N_{n_c})$ (which is caused by the nonlinear part of f_i^α), Figure 3.1. If one attempts to use equations (3.2) - (3.5) several times in succession, without letting the solution converge with an iterative routine, one would get an additional error due to the fact that Equation (3.3) is no longer exact. This is further discussed in Section 5.3.3.

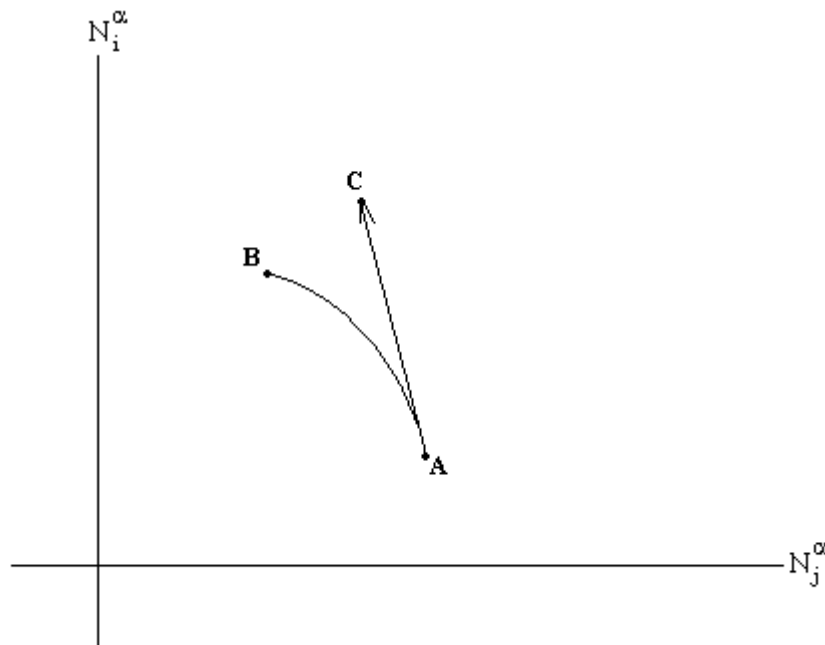


Figure 3.1: If we let the figure represent the variation of any two components $i \neq j$ during a time-step, the path AB would represent the true movement of the composition, while the line segment AC represents the changes estimated by NIFC.

We will from now on refer to the procedure described in equations (3.2) - (3.5) as NIFC (Non-Iterative Flash Calculations).

One may also note another weakness with NIFC is that when a component i disappears in a phase α , $\ln f_i^\alpha = -\infty$, and that the partial derivatives occurring in equations (3.4) and (3.5) are not defined. This is a problem when using this method in simulations with many components, as heavier components tend to drop out of the gas phase at lower temperatures and pressures. Components that only appear in one phase will from now

on be called non-split components. To avoid the problem of undefined derivatives of non-split components, we introduce the method NIFC-NS (Non-Iterative Flash Calculation - Non-Split components). This method is identical to NIFC, except that we ignore the components that appear in only one phase. Mathematically, this is equivalent to setting $\frac{\partial \ln(f_i^\alpha)}{\partial N_j^\alpha} = 0$ when i or j denotes a non split component.

Chapter 4

A new approximate flash calculation

4.1 Derivation

Bearing in mind the same ideas in [9], we can proceed using equations (3.2) and (3.3) with:

$$f_i^o = f_i^g \quad (4.1)$$

We do this to avoid double logarithms later. This leads to a variant of equations (3.4) and (3.5):

$$\sum_k \left(\frac{\partial f_i^o}{\partial N_k^o} + \frac{\partial f_i^g}{\partial N_k^g} \right) \frac{\partial N_k^o}{\partial p} = \frac{\partial f_i^g}{\partial p} - \frac{\partial f_i^o}{\partial p} \quad \forall i \quad (4.2)$$

$$\sum_k \left(\frac{\partial f_i^o}{\partial N_k^o} + \frac{\partial f_i^g}{\partial N_k^g} \right) \frac{\partial N_k^o}{\partial N_j} = \frac{\partial f_i^g}{\partial N_j} \quad \forall i, j \quad (4.3)$$

The temperature can be dealt with in an analogue fashion to pressure, and for simplicity will not be considered in the following.

Inserting $f_i^\alpha = N_i^\alpha \tilde{f}_i^\alpha$ above, we may arrive at the form:

$$\sum_k \left(\delta_{ik} + \frac{N_i^o N_i^g}{N_i} \left(\frac{\partial \ln(\tilde{f}_i^o)}{\partial N_k^o} + \frac{\partial \ln(\tilde{f}_i^g)}{\partial N_k^g} \right) \right) \frac{\partial N_k^o}{\partial p} = \frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g / f_i^o)}{\partial p} \quad (4.4)$$

$$\sum_k \left(\delta_{ik} + \frac{N_i^o N_i^g}{N_i} \left(\frac{\partial \ln(\tilde{f}_i^o)}{\partial N_k^o} + \frac{\partial \ln(\tilde{f}_i^g)}{\partial N_k^g} \right) \right) \frac{\partial N_k^o}{\partial N_j} = \frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g)}{\partial N_j} \quad (4.5)$$

Where δ_{ik} is the Kronecker delta, and

$$\tilde{f}_i^\alpha = \frac{f_i^\alpha}{N_i^\alpha} \quad (4.6)$$

Equation (4.5) can be solved simply by evaluating the derivatives (either numerically or analytically), and hence solve the linear system of equations that arise. This would most likely result in an equivalent estimate to NIFC. With multicomponent oil where the number of components easily rise above 100, the above system (and also that of equations (3.4) and (3.5)) may take some time to invert and solve. To be precise: We have n_c^2 equations in the system (4.5), which in Wang's scheme is solvable by inverting (e.g. PLU decomposing) a single $n_c \times n_c$ matrix, and subsequently applying this to the n_c systems. Both these operations take $O(n_c^3)$. We will show that with the system (4.5), the Jacobi $\frac{\partial N_i^o}{\partial N_j}$ is possible to estimate in $O(n_c^2)$, which is identical to the length of the vector, and hence must be optimal.

4.2 Evaluation of $\frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha}$

When treating the term $\frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha}$, we will use the fugacity based on the Peng-Robinson equation [7]:

$$f_i^\alpha = c_i^\alpha \psi_i^\alpha$$

$$\psi_i^\alpha = \frac{p^\alpha}{Z^\alpha - B^\alpha} \exp \left[\frac{b_i}{b^\alpha} (Z^\alpha - 1) \right] \left[\frac{Z^\alpha + (\sqrt{2} + 1)B^\alpha}{Z^\alpha - (\sqrt{2} - 1)B^\alpha} \right]^{-n_i^\alpha} \quad (4.7)$$

$$n_i^\alpha = \frac{A^\alpha}{2\sqrt{2}B^\alpha} \left[\frac{2}{a^\alpha} \sum_j (a_i a_j)^{1/2} c_j^\alpha (1 - \kappa_{ij}) - \frac{b_i}{b^\alpha} \right] \quad (4.8)$$

Where Z^α is the positive real solution of the Peng-Robinson equation of state:

$$Z^3 - (1 - B^\alpha)Z^2 + (A^\alpha - 3(B^\alpha)^2 - 2B^\alpha)Z - (A^\alpha B^\alpha - (B^\alpha)^2 - (B^\alpha)^3) = 0 \quad (4.9)$$

When the equation has several positive roots, the largest is that of the gas phase and the smallest is that of the oil phase. The variables A^α and B^α are determined from:

$$A^\alpha = \frac{a^\alpha p^\alpha}{R^2 T^2}, \quad B^\alpha = \frac{b^\alpha p^\alpha}{RT} \quad (4.10)$$

where the constants a^α and b^α are determined from critical conditions (note that a^α is in fact a function of temperature):

$$a^\alpha = \sum_{i,j=1}^{n_c} \sqrt{(a_i a_j)} c_i^\alpha c_j^\alpha (1 - \kappa_{ij}), \quad b^\alpha = \sum_{i=1}^{n_c} c_i^\alpha b_i$$

$$a_i = 0.45724 \frac{R^2 T_i^2}{P_i} [1 + m_i (1 - \sqrt{T_{ri}})]^2, \quad b_i = 0.07780 \frac{RT_i}{P_i} \quad (4.11)$$

$$m_i = 0.37464 + 1.54226 \omega_i - 0.26992 \omega_i^2, \quad T_{ri} = \frac{T}{T_i}, \quad P_{ri} = \frac{P}{P_i}$$

With the definitions above, and Equation (4.6), we have:

$$\frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha} = \frac{\partial}{\partial N_j^\alpha} \ln\left(\frac{c_i^\alpha}{N_i^\alpha}\right) + \frac{\partial \ln(\psi_i^\alpha)}{\partial N_j^\alpha} \quad (4.12)$$

At this point we will propose the simplification:

$$\frac{\partial \ln(\psi_i^\alpha)}{\partial N_j^\alpha} \ll \frac{\partial}{\partial N_j^\alpha} \ln\left(\frac{c_i^\alpha}{N_i^\alpha}\right) \quad (4.13)$$

Deeper investigation into the fugacity coefficient, indicates that the above simplification in many cases is not valid. However, the results obtained in the numerical experiments presented later, along with the speed at which they may be obtained, justifies some further investigation.

We then calculate:

$$\begin{aligned} \frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha} &= \frac{\partial}{\partial N_j^\alpha} \ln\left(\frac{c_i^\alpha}{N_i^\alpha}\right) \\ &= \frac{\frac{\partial}{\partial N_j^\alpha} \left(\frac{c_i^\alpha}{N_i^\alpha}\right)}{\frac{c_i^\alpha}{N_i^\alpha}} = -\frac{\frac{1}{(\sum_k N_k^\alpha)^2}}{\frac{1}{\sum_k N_k^\alpha}} \\ &= -\frac{1}{\sum_k N_k^\alpha} = -\frac{1}{N^\alpha} = -\frac{c_j^\alpha}{N_j^\alpha} \end{aligned} \quad (4.14)$$

4.3 New estimate for $\frac{\partial N_i^\alpha}{\partial N_i}$ and $\frac{\partial N_i^\alpha}{\partial p}$

Inserting here the results from Section 4.2 into the left hand sides of equations (4.4) and (4.5), we obtain:

$$\sum_k (\delta_{ik} - \frac{N_i^o N_i^g}{N_i} (\frac{1}{N^o} + \frac{1}{N^g})) \frac{\partial N_k^o}{\partial p} = \frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g/f_i^o)}{\partial p} \quad (4.15)$$

$$\sum_k (\delta_{ik} - \frac{N_i^o N_i^g}{N_i} (\frac{1}{N^o} + \frac{1}{N^g})) \frac{\partial N_k^o}{\partial N_j} = \frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g)}{\partial N_j^g} \quad (4.16)$$

We further arrive at:

$$\sum_k (\delta_{ik} - \frac{N_i^o N_i^g}{N_i} \frac{N}{N^o N^g}) \frac{\partial N_k^o}{\partial p} = \frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g / f_i^o)}{\partial p} \quad (4.17)$$

$$\sum_k (\delta_{ik} - \frac{N_i^o N_i^g}{N_i} \frac{N}{N^o N^g}) \frac{\partial N_k^o}{\partial N_j} = \frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g)}{\partial N_j^g} \quad (4.18)$$

These are in the form:

$$J' W = B$$

Where J' is a matrix, W a vector with the partial derivatives, and B a vector.

We see from Equation (4.17) and Equation (4.18) that J' is independent of the index j , and hence we have n_c systems of n_c linear equations which can be solved simultaneously with the same inversion of J' . Also, note that $J^* = J' - I$ defines a matrix J^* where all columns are identical. We use this information to construct the inverse of J' . Let \vec{j} be any column of J^* , and $\vec{i}_n = [11\dots 1]$ such that $J^* = \vec{j} \cdot \vec{i}_n$. Let us for simplicity of notation introduce $(J')^{-1} = K$, and let K , as J' , be $K = I + K^* = I + \vec{k} \cdot \vec{i}_n$. This enables:

$$\begin{aligned} I &= (J')^{-1} \cdot J = (I + \vec{k} \cdot \vec{i}_n) \cdot (I + \vec{j} \cdot \vec{i}_n) \\ \implies (\vec{k} + \vec{j} + \vec{k} \cdot (\vec{i}_n \cdot \vec{j})) \cdot \vec{i}_n &= 0 \\ \implies \vec{k} &= \frac{-\vec{j}}{1 + \text{sum}(\vec{j})} \\ \implies (J')^{-1} = K = I + \vec{k} \cdot \vec{i}_n &= I + \frac{-\vec{j} \cdot \vec{i}_n}{1 + \text{sum}(\vec{j})} \end{aligned} \quad (4.19)$$

We now have a simple expression for $(J')^{-1}$, and inserting Equation (4.19) into Equation (4.18), we get:

$$\begin{aligned} \frac{\partial N_k^o}{\partial N_j} &= \sum_i (\delta_{ik} + \frac{N_k^o N_k^g}{N_k} \frac{N}{N^o N^g}) (\frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g)}{\partial N_j^g}) \quad \forall j, k \\ &= S_{2,j} \frac{\frac{N_k^o N_k^g}{N_k} \frac{N}{N^o N^g}}{1 + S} + (\frac{N_k^o N_k^g}{N_k} \frac{\partial \ln(f_k^g)}{\partial N_j^g}) \\ S &= \sum_i -\frac{N_i^o N_i^g}{N_i} \frac{N}{N^o N^g} \\ S_{2,j} &= \sum_i \frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g)}{\partial N_j^g} \end{aligned} \quad (4.20)$$

and likewise when inserting Equation (4.19) into Equation (4.17):

$$\begin{aligned}
\frac{\partial N_k^o}{\partial p} &= \sum_i \left(\delta_{ik} + \frac{\frac{N_k^o N_k^g}{N_k} \frac{N}{N^o N^g}}{1 + S} \right) \left(\frac{N_i^o N_i^g}{N_i} \frac{\partial \ln(f_i^g / f_i^o)}{\partial p} \right) \quad \forall k \\
&= S_3 \frac{\frac{N_k^o N_k^g}{N_k} \frac{N}{N^o N^g}}{1 + S} + \left(\frac{N_k^o N_k^g}{N_k} \frac{\partial \ln(f_k^g / f_k^o)}{\partial p} \right) \\
S &= \sum_i -\frac{N_i^o N_i^g}{N_i} \frac{N}{N^o N^g} \\
S_3 &= \sum_i \frac{N_i^o N_i^g}{N_i} \frac{\ln(f_i^g / f_i^o)}{p}
\end{aligned} \tag{4.21}$$

The system (4.20) may also be presented with (4.14) inserted everywhere, leading to:

$$\begin{aligned}
\frac{\partial N_k^o}{\partial N_j} &= \sum_i \left(\delta_{ik} + \frac{\frac{N_k^o N_k^g}{N_k} \frac{N}{N^o N^g}}{1 + S} \right) \left(\frac{N_i^o}{N_j} \delta_{ij} - \frac{N_i^o N_i^g}{N_i N^g} \right) \quad \forall j, k \\
&= \left(\frac{N_j^o}{N_j} + S_4 \right) \frac{\frac{N_k^o N_k^g}{N_k} \frac{N}{N^o N^g}}{1 + S} + \left(\frac{N_k^o}{N_j} \delta_{kj} - \frac{N_k^o N_i^g}{N_k N^g} \right) \\
S &= \sum_i -\frac{N_i^o N_i^g}{N_i} \frac{N}{N^o N^g} \\
S_4 &= \sum_i -\frac{N_i^o N_i^g}{N_i N^g}
\end{aligned} \tag{4.22}$$

Observe that this final system does not involve any fugacity calculations at all, and therefore avoids many time consuming operations.

4.4 Complexity of the new procedure

We will refer to the procedure proposed in equations (4.20) and (4.21) as FFC1 (Faster non-iterative Flash Calculations) and the procedure proposed in Equation (4.22) as FFC2. Pressure variation in FFC2 will be ignored for the time being.

After calculating the fugacities in $O(n_c^2)$ operations (as discussed in the end of Section 3.1), we observe that $S_{2,j}$, S_3 and S_4 are calculated in $O(n_c)$ operations, so all the $S_{2,j}$'s hence are computed in $O(n_c^2)$. Equations (4.20), (4.21) and (4.22) are all calculated in $O(1) \forall j, k$.

Since we then have $n_c^2 + n_c$ equations it then follows that solving the whole system takes $O(n_c^2)$ operations, as promised.

We note that although both FFC1 and FFC2 have the same order of execution time, many more calculations are required in FFC1 than FFC2 due to the complexity of calculating fugacities and their derivatives. Therefore FFC2 is by far the faster alternative.

4.5 Discussion

We see that we have managed to reduce the computational cost of NIFC, $O(n_c^3)$, to $O(n_c^2)$ at the expense of one rather drastic assumption; we have neglected the partial derivatives of the fugacity coefficient with respect to component molar mass on the left (and right in the case of Equation (4.22)) hand side of equations (4.15) and (4.16).

As a consequence the vector $\frac{\partial N_k^\alpha}{\partial N_j^\alpha}$ computed is no longer the true gradient of $N_i^\alpha(P, N_1 \dots N_{n_c})$, and hence that letting $dN_i^\alpha \rightarrow 0$ in Equation (3.2) does not make the system converge. This is unfortunate, but not necessarily all that bad. We note that in addition to the errors described for NIFC in Section 3.2, we now have the additional error due to the assumptions made in Equation (4.14), see Figure 4.1.

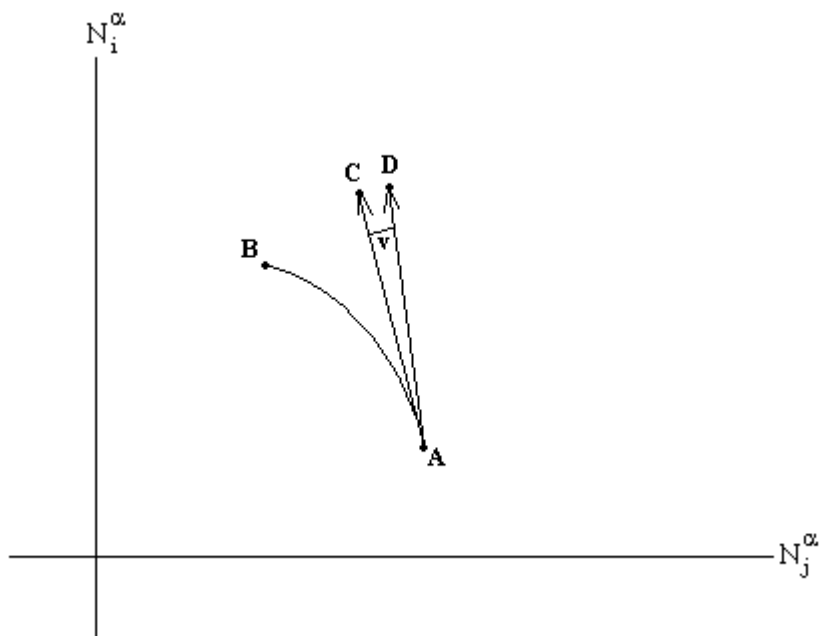


Figure 4.1: This figure (analogue to Figure 3.1), again shows the true compositional change AB, the estimate of NIFC, C, and the additional error introduced in the new methods, described by the angle v .

It is also worth noting that the error occurring in NIFC (described in Section 3.2) due to the non-split in phases is eliminated in FFC2 due to Equation (4.14).

We introduce FFC1-NS in the same way as we did with NIFC and NIFC-NS in Section 3.2: FFC1-NS estimates the new composition with the assumption that $\frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha} = 0$ when i or j denotes a non-split component.

Chapter 5

Numerical experiments

5.1 Relevant data

When evaluating the validity of the new preconditioners, it is essential to test them in an environment where we can avoid all other potential sources of error. For this purpose we have used the following setting as a testing scheme:

Using conventional means (a phase-splitting program called PVTsim, available courtesy of Norsk Hydro), we obtained phase splits for several compositions, at various thermodynamic states. Basing our calculations on the split of one composition, at a given pressure and temperature, we may then be able to attempt to estimate other splits. This is a simulation of the real-life scenario where you have computed the change of composition, and need to update the phase-split. As we saw in Section 2.1 the pressure and composition is updated separately, so we conduct the tests without attempting simultaneous update of composition and pressure. Temperature is still assumed constant.

The input data is therefore: Total phase composition, pressure and temperature, along with the correct phase split, and the desired variation of composition or pressure. Output is the phase split at the new composition, which can be compared with known data. The component characteristics used in calculating the fugacities (equations (4.7)-(4.11)) are available to our program as well.

We will compare NIFC with the two preconditioners introduced in this work, FFC1 and FFC2. We will also investigate the effect of using several steps.

5.2 Experiments

Although several compositions have been tested, with different components, we will for brevity only present one experiment here. The other experiments presented us with much the same results.

We are looking at a 22 component system, where two components are non-hydrocarbon, 11 were hydrocarbons, and the remaining nine were pseudo-components of C_{10}^+ fractions. The components in this experiment are the same as found in many reservoirs, indeed, the composition chosen here closely matches one of the sample cases in PVTsim. Experiments are also conducted with this system when the non-hydrocarbons were removed.

First we tested with two closely related ($\Delta N_i < 0.1 * N_i$) compositions, while holding the pressure constant. Then we tested related pressures ($\Delta p = 1$) while keeping the composition constant. To assess the effect on the derivatives of non-existent components in phases (described in Section 3.2), we conducted experiments on the procedures NIFC and FFC1 (where the undefined derivatives appearing were replaced by estimates from FFC2), and also the special procedures NIFC-NS and FFC1-NS. The results are presented in tables 5.1 and 5.2, along with the results for FFC2.

The inaccuracy referred to in the tables is calculated by the formula:

$$e = \frac{\|(\overline{N}_i^o)_{ex} - (\overline{N}_i^o)_{est}\|}{\|(\overline{N}_i^o)_{ex} - (\overline{N}_i^o)_{init}\|} \quad (5.1)$$

Where $(\overline{N}_i^o)_{ex}$, $(\overline{N}_i^o)_{est}$, $(\overline{N}_i^o)_{init}$ are the exact, estimated, and initial composition vectors, respectively. $\|\cdot\|$ is the 2-norm of the vectors. This is motivated by the fact that $\|(\overline{N}_i^o)_{ex} - (\overline{N}_i^o)_{est}\|$ is the error fed to the subsequent iterative routine, while $\|(\overline{N}_i^o)_{ex} - (\overline{N}_i^o)_{init}\|$ is the error that would be present if no preconditioning was done. Therefore estimating that the composition will not change during the time-step leads to an error of 1.00, as appears in Table 5.2 for FFC2. In the experiments, we naturally also looked at the error in the gas vectors, as well as the oil/gas ratios, and although variations occur, the selected error is typical.

	State		Inaccuracy in procedure					
	Pres. (bar)	Temp. (C)	NIFC	NIFC-NS	FFC1	FFC1-NS	FFC2	Non-SC
Case 1	100	50	0.77	0.65	0.61	0.61	0.11	6
Case 2	150	50	0.43	0.37	0.26	0.17	0.22	4
Case 3	100	100	0.43	0.43	0.15	0.10	0.08	4
Case 4	200	100	0.42	0.38	0.36	0.32	0.29	2
Case 5	300	100	0.64	0.64	0.81	0.81	0.63	0
Case 6*	100	0	0.38	0.45	0.39	0.33	0.20	7
Case 7*	280	100	0.50	0.50	1.30	1.30	0.75	0
Case 8*	330	250	0.84	0.84	1.00	1.00	0.24	0
Case 9*	200	400	0.22	0.22	0.61	0.61	0.26	0
Case 10*	100	250	0.37	0.39	0.34	0.33	0.10	1

Table 5.1: In this table we have constant temperature and pressure, while the composition is varying. The '*' denotes cases where the composition is of pure hydrocarbon components.

To see the results in connection to the properties of the mixtures, we include the phase envelopes of Cases 1-5, Figure 5.1 and Cases 6-10, Figure 5.2.

Finally we include experiments with multiple steps. These are done in full scale only for FFC2, since the time required for the runs of FFC1 and NIFC was not available. We also did a smaller scale test of NIFC, assuming that the results for FFC1 would be somewhere in between. In the Cases presented in tables 5.1 and 5.2 multiple steps have little or no effect on the precision of FFC2, so only the results for NIFC will be presented here in tables 5.3 and 5.4, with the understanding that they may be compared with the results given for FFC2 in tables 5.1 and 5.2.

	State		Inaccuracy in procedure					
	Pres. (bar)	Temp. (C)	NIFC	NIFC-NS	FFC1	FFC1-NS	FFC2	Non-SC
Case 1	100	50	0.68	0.18	0.18	0.15	1.00	6
Case 2	150	50	0.49	0.11	0.26	0.28	1.00	4
Case 3	100	100	0.54	0.11	0.15	0.14	1.00	4
Case 4	200	100	0.56	0.04	0.35	0.37	1.00	2
Case 5	300	100	0.03	0.03	0.69	0.69	1.00	0
Case 6*	100	0	0.69	0.27	0.21	0.16	1.00	7
Case 7*	280	100	0.02	0.02	0.60	0.60	1.00	0
Case 8*	330	250	0.52	0.52	0.89	0.89	1.00	0
Case 9*	200	400	0.15	0.15	0.19	0.19	1.00	0
Case 10*	100	250	0.26	0.05	0.29	0.30	1.00	1

Table 5.2: In this table we have constant temperature and composition, while the pressure is varying. The '*' denotes cases where the composition is of pure hydrocarbon components.

	State		Inaccuracy in procedure				Non-SC
	Pres. (bar)	Temp. (C)	1 step	2 steps	5 steps	10 steps	
Case 1	100	50	0.77	0.62	0.48	0.44	6
Case 2	150	50	0.43	0.39	0.42	0.41	4
Case 3	100	100	0.43	0.44	0.44	0.44	4
Case 4	200	100	0.42	0.38	0.37	0.36	2
Case 5	300	100	0.64	0.99	1.04	0.97	0
Case 6*	100	0	0.38	0.38	0.43	0.45	7
Case 7*	280	100	0.50	0.40	0.77	1.00	0
Case 8*	330	250	0.84	1.02	1.08	1.16	0
Case 9*	200	400	0.22	0.24	0.24	0.24	0
Case 10*	100	250	0.37	0.38	0.38	0.38	1

Table 5.3: Tests for multiple steps with NIFC, composition variation. All tests without eliminating non-split components.

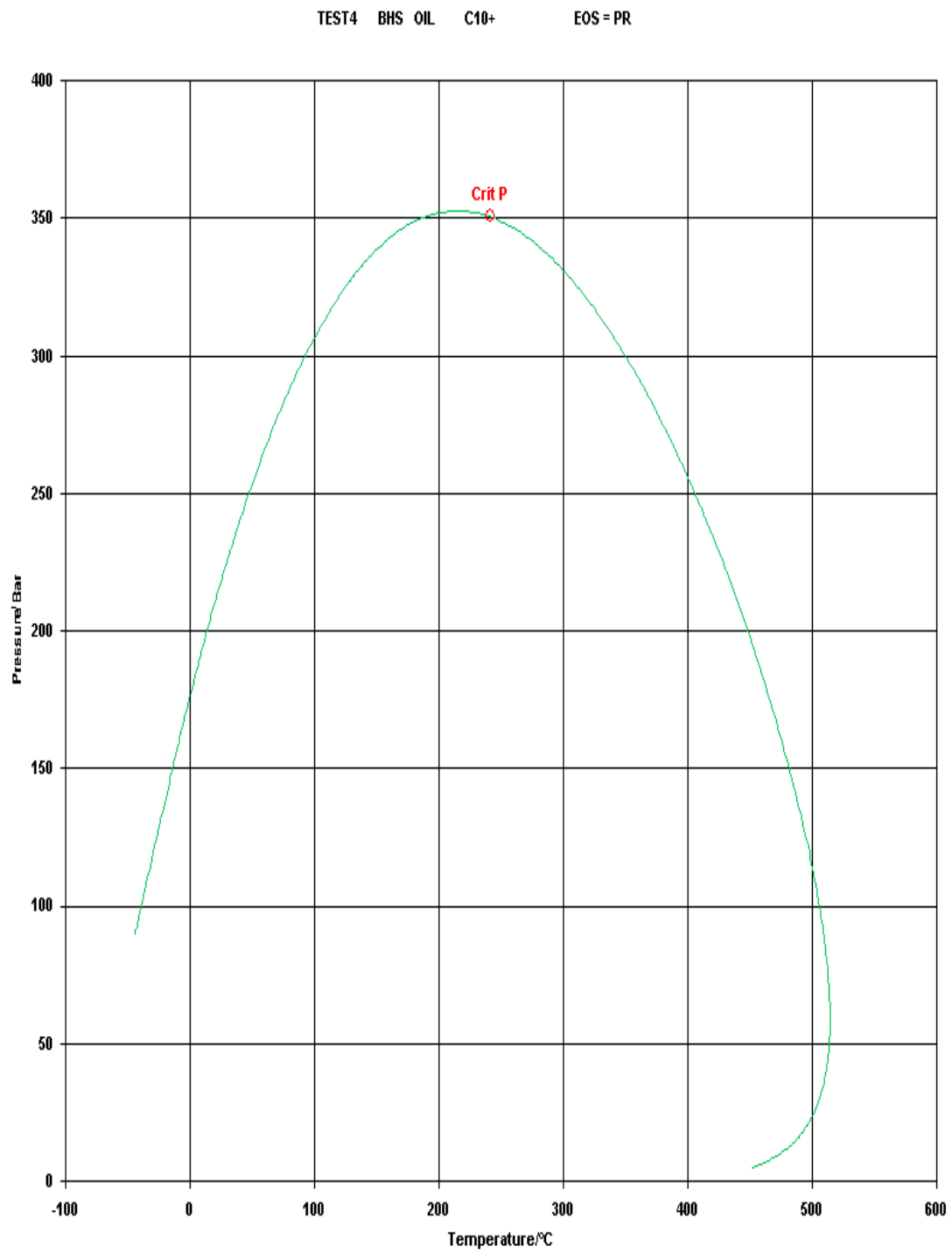


Figure 5.1: Initial phase envelope and critical point for cases 1-5.

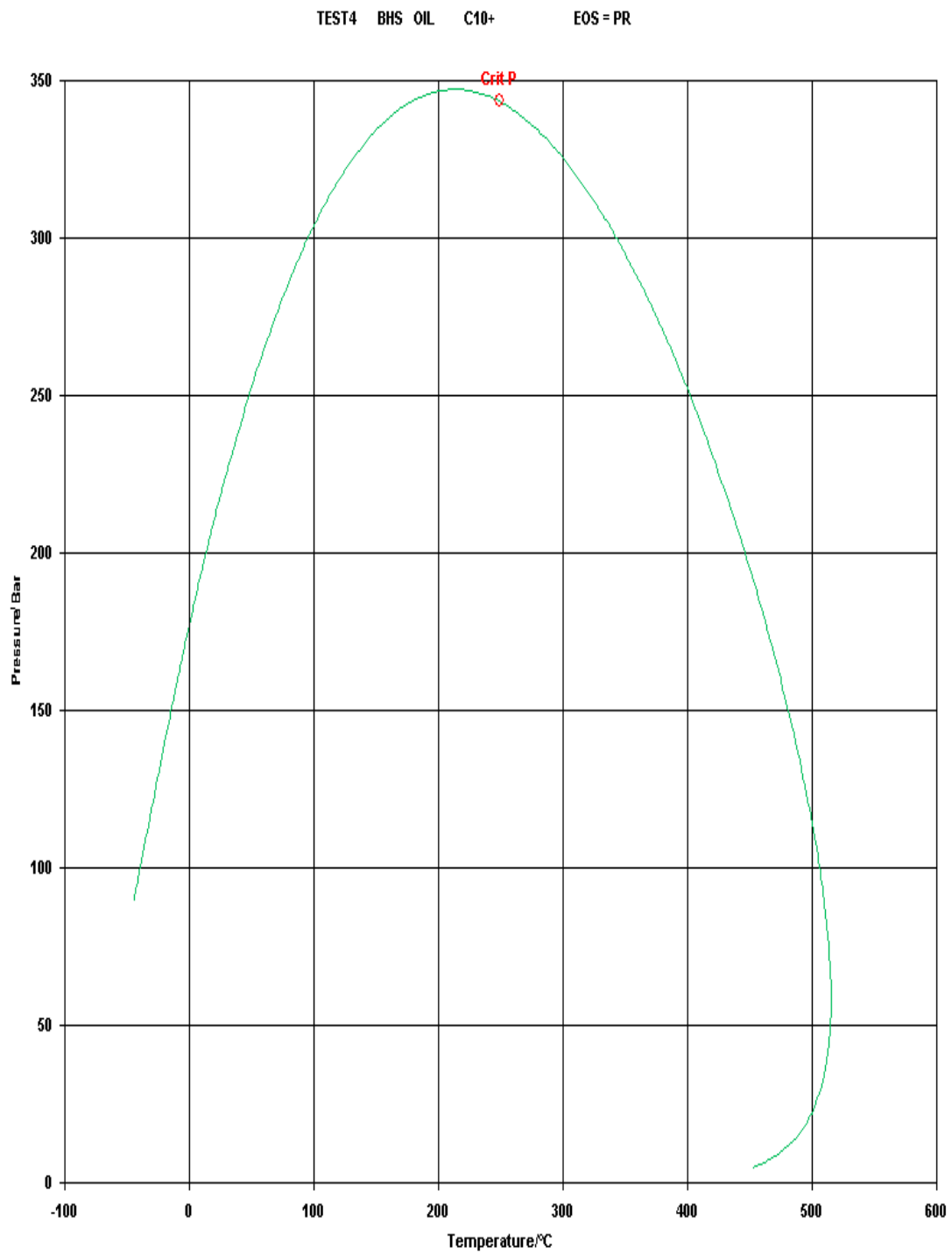


Figure 5.2: Initial phase envelope and critical point for cases 6-10.

	State		Inaccuracy in procedure				
	Pres. (bar)	Temp. (C)	1 step	2 steps	5 steps	10 steps	Non-SC
Case 1	100	50	0.68	0.43	0.17	0.09	6
Case 2	150	50	0.49	0.62	0.13	0.05	4
Case 3	100	100	0.54	0.28	0.11	0.06	4
Case 4	200	100	0.56	0.23	0.10	0.07	2
Case 5	300	100	0.03	0.04	0.05	0.05	0
Case 6*	100	0	0.69	0.38	0.25	0.10	7
Case 7*	280	100	0.02	0.38	0.09	0.06	0
Case 8*	330	250	0.52	0.59	0.63	0.64	0
Case 9*	200	400	0.15	0.15	0.15	0.15	0
Case 10*	100	250	0.26	0.15	0.08	0.07	1

Table 5.4: Tests for multiple steps with NIFC, pressure variation. All tests without eliminating non-split components.

5.3 Discussion of results

The results presented above give us ample possibility to draw some conclusions. This will be done in the following subsections.

5.3.1 Composition changes

When viewing the test results presented in Table 5.1 as well as the phase envelopes in figures 5.1 and 5.2, we are led to the conclusions that follow.

The method NIFC suffers by the fact that it does not handle components that are only in one phase. Eliminating these components as done in NIFC-NS generally helps, although this introduces a new error. If we focus on NIFC-NS we observe that the performance of this method is closely dependent on the distance from the phase envelope, the critical point and the number of non-split components.

The first proposed methods, FFC1 and FFC1-NS, lead to better results with non-split components than NIFC-NS, but suffer when we approach the phase envelope and the critical point. This may be due to the fact that we use two different estimates for the derivative of the logarithm of the fugacity (the numerical derivative used in NIFC and Equation (4.14)) and that these may vary greatly in critical regions. In Case 7, FFC1 even manages to propose a worse split than just using the old phase split.

The second proposed method, FFC2, performs in all cases except 2, 7, and 9, better than both FFC1 and NIFC. We know from the derivation of the method that we may expect poor results in regions with strong dependency on the factor ψ_i^α (from Equation (4.7)).

The poor result in Case 7 was therefore expected, and viewing similar cases to Case 8 from different tests, we may say that this very good split was a bit lucky.

Since the proximity to the phase envelope seems to be so important for the accuracy of all methods, one might wish to have knowledge of the phase envelope before any calculations are initiated. A description of an efficient way of calculating the phase envelope can be found in [5]. This idea will not be pursued further here.

The author would also like to comment that while testing, FFC2 was observed to be faster by far than the two other methods, although no time testing has been conducted.

5.3.2 Pressure changes

There is little need to dwell too much at the results from pressure variation. Table 5.2 clearly shows the great advantage of NIFC-NS over NIFC, and gives good reference values when evaluating the proposed methods.

FFC1 does not seem to benefit from the elimination of non-split components, and produces reasonably good results. As above, we see that when approaching the phase envelope the results are substantially poorer than those of NIFC.

FFC2 has no method of handling pressure variation.

5.3.3 Several steps

It is obvious that when approximating a new composition in v steps, we use v times as much time. Since the main feature of the above procedure is speed, we must assess the benefits of several steps.

Given the inaccuracy of the solution in some of the more testing conditions (near the critical point and the phase envelope), we understand that this solution procedure can not be used alone in finding the next phase split. This is no surprise, indeed we consistently called our solution procedure a preconditioner, as we understand that the assumption $\frac{\partial \ln(\psi_i^\alpha)}{\partial N_j^\alpha} = 0$ is not always justifiable.

We propose to do as in [9], that is, to use our estimate as an initial estimate in a standard iteration procedure. This has not been implemented, as state of the art phase-splitting codes are not available to us.

Using v steps to approximate a new split can be seen as applying the iteration only every v steps. But we must then be aware of the nature of our solution procedure. We have based our system of equations on the fact that $\frac{\partial f_i^o}{\partial N_j} - \frac{\partial f_i^g}{\partial N_j} = 0 \forall i, j$, and differentiated this equation. This implies that $\frac{\partial f_i^o}{\partial N_j} - \frac{\partial f_i^g}{\partial N_j} = \text{constant} \forall i$. Which is indeed true when the

constant is 0 and the system is in balance. But after a step leading to an only approximate solution, we will have $\frac{\partial f_i^o}{\partial N_j} - \frac{\partial f_i^g}{\partial N_j} = \text{constant} \neq 0$. This is shown in Figure 5.3.

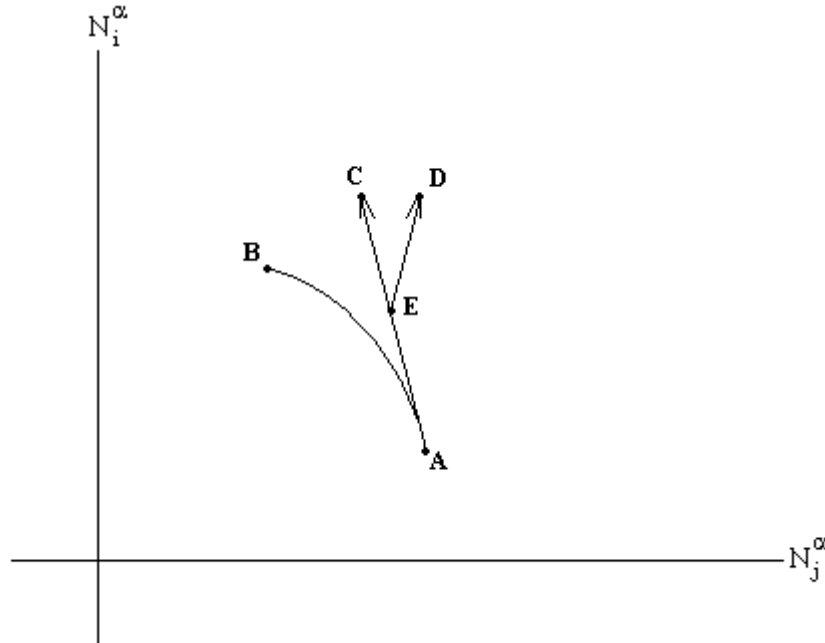


Figure 5.3: This is the same figure as in Section 3.2, but with a possible two step method introduced. The first step leads to point E, the second step leads to a point D beyond our control.

When viewing the test results from Table 5.3 (cases 3, 5, 6, (7), 8, 9, and 10), we see that using the direct methods reviewed here several times per application of the iterative cannot be recommended. Especially when keeping in mind the extra computational time involved. The same conclusion can be drawn from Table 5.4, although the results here are slightly better.

Chapter 6

Conclusion and further work

6.1 Conclusion

We have presented two new preconditioners for phase splitting algorithms in Chapter 4, abbreviated FFC1 and FFC2. The preconditioner NIFC from [9] was presented in Chapter 3. Experiments on these were conducted and evaluated in Chapter 5.

The experiments show that the method FFC2 performs at least as well as the previously published NIFC when concerned with composition variation. We further note that since the test results indicate that the preconditioners are not accurate enough to be used alone, and therefore must be used along with an iterative scheme, the new method FFC2 has a marked advantage over NIFC, due to its simplicity and speed.

In the case of pressure variation, we see that the results of NIFC and FFC1-NS are comparable. Again, we suggest that the new method FFC1-NS has the advantage, when considering computational costs.

Generally, we suggest to use FFC2 to estimate changes due to variation in mixture compositions, and FFC1-NS when estimating changes due to pressure changes. This is seen as the third and first terms on the right hand side of Equation (3.1).

We note that when the mixture is close to the phase envelope, all methods produce generally more unpredictable estimates. We have no data from experiments indicate errors $e > 1$ for NIFC or FFC2 and FFC1-NS when used in conjunction as mentioned in the two previous paragraphs.

Multiple steps between iterations is not recommended, as this may result in overall worse results than doing just one step, as described in Section 5.3.3.

In sum, we see that preconditioning with the conjunction of FFC2 and FFC1-NS provides results at least as accurate as the previously published method used as reference, while having a substantially lower computational cost. There is therefore reason to be motivated into further research and implementation of these methods.

6.2 Further work

The natural continuation of the investigation is to implement the preconditioner in a simulator like SOM, so that more rigorous experiments may be performed with respect to the accuracy of the new methods, as well as time testing the preconditioner against the previous pure iterative method.

In the case of non-split components, one might try to implement non square matrices, that is: Instead of setting $\frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha} = 0$ when i or j denotes a non-split component, one might set $\frac{\partial \ln(\tilde{f}_i^\alpha)}{\partial N_j^\alpha} = 0$ only when i denotes a non-split component. This might produce better results,

although it affects the speed of the method.

Temperature variation should be included.

In an implementation, one might wish to not use the current node as the basis preconditioner, it might turn out that one of the neighboring nodes might be more related to the new composition, especially in advection dominated processes. By backtracking characteristics, one might find a candidate node. A comparison to find which composition was closest related can be done by choosing the composition at node n_i , where the expression $g(n_i, n_j)$:

$$g(n_i, n_j) = \|(\overline{N}_i)_{old, node\ n_i} - (\overline{N}_i)_{new, node\ n_j}\| \quad (6.1)$$

attains a minimum. Here (\overline{N}_i) is the vector of total molar mass of components, n_i is chosen from some set of candidate nodes, and n_j is the node where we want to do the phase split. The subscripts *old* and *new* indicate previous and new time-step.

This method may be used not only when updating a time-step. Grid cells close to each other should have related phase splits, this would make an impact on the first calculation of the phase split on the grid.

The ideas introduced in the final paragraph of Section 2.2 should be followed up. This might involve inserting FFC2 into mass transport equations and see if this would lead to a more accurate calculation of multicomponent flow. Work in this direction has been done at this institute, but with much simpler approximations to phase split dynamics than FFC2.

The proposed method is a first order approximation. Higher order approximations should be investigated.

A more rigorous analytic evaluation of the methods would be desirable, although difficult (no mention of any such is made in Wang's report).

Appendix A

Nomenclature

Symbols

- a constant, see Equation (4.11)
- A variable, see Equation (4.10)
- b constant, see Equation (4.11)
- B variable, see Equation (4.10)
- c_i^α concentration of component i in phase α
- CS surface of control volume, CV
- CV control volume
- d depth
- e inaccuracy in preconditioner
- δ Kronecker delta ($\delta_{i,j} = 1$ if $i = j$, 0 otherwise)
- f fugacity
- F flux
- g gravitational constant
- h enthalpy
- I identity matrix
- J a Jacobian matrix
- κ permeability tensor of the medium
- m constant, see Equation (4.11)
- n_c number of components
- n_i^α exponent in Equation (4.7)
- N molar mass
- \bar{N} molar mass seen as a vector
- ϕ porosity
- p pressure
- p_c capillary pressure
- p_i critical pressure of component i

- p_{ri} relative pressure, see Equation (4.11)
- ψ fugacity coefficient
- ρ mass density
- R universal gas constant
- q source/sink term
- S saturation
- t time
- T temperature
- T_i critical temperature of component i
- T_{ri} relative temperature, see Equation (4.11)
- \vec{u} Darcy velocity
- u internal energy
- μ viscosity
- ω eccentric factor
- Z compressibility

Superscripts

- α index for the phases gas, oil, water
- β index for the phases gas, oil, water
- g gas
- o oil
- r rock phase
- w water

Subscripts

- i index for components
- j index for components
- k index for components
- T temperature

Abbreviations

- *Non-SC* Non-Split Component
- *NIFC* Non-Iterative Flash Calculation
- *FFC* Fast non-iterative Flash Calculation
- *-NS* - Non-Split components
- *SOM* Secondary Oil Migration
- *PLU* Permutation Lower Upper (matrices)

Appendix B

Overview of test code

The code is constructed with the concept of clarity in mind, instead of quickness of execution or elegance, since its purpose is merely to test the validity of our preconditioner, as described in sections 3.1, 4.1 and 5.1.

We are concerned with three main files (*script6.m*, *Estimat2.m*, *Estimat4.m*), three utility files (*c_to_N.m*, *N_to_c.m*, *fugacity.m*), and one datafile, all coded in Matlab.

script6.m is our main file, it functions as following:

- Input from user (which method to use).
- Loads data in two datasets (primary and secondary).
- Estimates the split for the secondary dataset based on the primary, by calling *Estimat2.m* or *Estimat4.m*.
- Compares estimate with the true values from the secondary dataset.
- Saves to file the relative error of the estimate to the change in composition, oil/gas ratio etc.

Estimat2.m estimates then new composition using FFC2.

Estimat4.m calculates the Jacobian needed using *fugacity.m*, and then estimates using NIFC or FFC1.

c_to_N.m and *N_to_c.m* have no function except converting between composition (as the data is originally given in) and molar mass (as is used in the calculations), and back.

fugacity.m is the top method in a hierarchy of functions that calculate the fugacity as described in Section 4.1.

There are several additional scrips functions that serve the purpose of checking the validity of the above mentioned. These are of little interest now.

Bibliography

- [1] G. Acs, S. Doleschall, and É. Farkas. General purpose compositional model. *Society of Petrol Engineers*, 1985.
- [2] Z. Chen, G. Qin, and R. Ewing. Analysis of a compositional model for fluid in porous media. *Journal of Applied Mathematics*, 2000.
- [3] E. F. M. Elewaut. Secondary oil migration project, an integrated geochemical and quantitative modeling approach for understanding and predicting secondary oil migration. *National Geological Survey*, 1996.
- [4] A. Firoozabadi. *Thermodynamics of Hydrocarbon Reservoirs*. McGraw-Hill, 1999.
- [5] M. Michelsen. Calculation of phase envelopes and critical points for multicomponent mixtures. *Fluid Phase Equilibria*, 1980.
- [6] M. Michelsen. The isothermal flash problem. part ii. phase-split calculation. *Fluid Phase Equilibria*, 1982.
- [7] D.-Y. Peng and D. B. Robinson. A new two-constant equation of state. *Ind. Eng. Chem., Fundam.*, 1976.
- [8] H. Reme. *The Preconditioning and Mult-Point Flux Approximation Methods for solving Secondary Oil Migration and Upscaling Problems*. PhD thesis, Department of Mathematics, University of Bergen, 1999.
- [9] P. Wang and E. H. Stenby. Non-iterative flash calculation algorithm in compositional reservoir simulation. *Fluid Phase Equilibria*, 1994.
- [10] J. W. Watts. A compositional formulation of the pressure and saturation equations. *SPE Reservoir Engineering*, 1986.
- [11] Y.-S. Wu and P. A. Forsyth. On the selection of primary variables in numerical formulation for modeling multiphase flow in porous media. *Journal of Contaminant Hydrology*, 2001.