Progress Report

Grant #DE-FC26-04NT15531

Name of the Recipient: University of Utah

Project Title: Parallel, Multigrid Finite Element Simulator for Fractured/Faulted and Other Complex Reservoirs based on Common Component Architecture (CCA)

Principal Investigator:
Milind D. Deo
Professor
Department of Chemical and Fuels Engineering
University of Utah, 50 South Central Campus Drive
Salt Lake City, Utah 84112
Phone: (801)581-7629
FAX: (801)585-9291
e-mail: mddeo@eng.utah.edu

Date Submitted: January 31, 2006

Period Covered by the Report: September 9, 2004 – December 31, 2005
Disclaimer:
This report was prepared an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
Executive Summary

The objectives of this project are to develop an efficient, general-purpose finite-element simulator with unstructured grids to model complex reservoirs. Most reservoirs are fractured to a certain degree and now there are better tools to characterize and map faults and fractures, field-wide. Generation of reservoir domains with faults and fractures using relevant geological information is one of the tasks. This project would make it possible to incorporate this enhanced characterization directly into a reservoir simulation framework. One key idea is to develop multi-scale computing schemes so that phenomena at different scales are represented and solved efficiently. This feature will be very important in solving a data set with a number of faults/fractures expeditiously. The goal was also to create a fast simulator with all the modern computational tools. Parallelization of large unstructured grid systems and standardization of files and various components employed are the activities designed to provide the outcome of a parallel, modular simulator of complex systems that is accessible and user friendly.

Significant accomplishments in the current year are listed below.

1. A methodology to grid complex, three-dimensional objects with faults and fractures was identified. The approach is to use CUBIT, the general gridding program created at Sandia National Laboratory. Using this tool and the scripts written at the University of Utah, it is now possible to take complicated faulted/fractured domains and create grids to honor the complicated geometry.

2. Three-dimensional, three-phase simulations were performed on grids generated using CUBIT. Methods to perform these computations with or without the matrix (fracture-only) simulations were developed.
3. Multiple ways of creating field-wide fracture networks were identified. Simulations on domain with a complicated set, generated using Fracman (software developed by Golder Inc.), are underway.

4. A thermodynamics simulator (obtaining compositions of vapor and liquid phases) using the successive substitution method was completed. This is the first step toward the development of a compositional simulator for fractured/faulted systems.

5. Formulation of thermal simulation equations was completed.

6. Intuitive well models of complicated wells for the mixed finite-element simulator were developed.

7. The simulator structure was modified to use not only the conventional reservoir simulation boundary conditions (bottom hole pressure and rate constraints) but also constant flux boundary conditions and Dirichlet (constant pressure) specifications.

8. In order to facilitate the modularization and standardization of simulator components, decoupling of the discretization method and the numerical model was performed. The input file was standardized to use the universal xml format.

9. Paralleization has been built into the structure of the simulator.

10. Seamless coupling with PetSc was accomplished leading to the use of the most advanced linear solvers such as GMRES and TFQMR.

Integration of geologic information and reservoir engineering evidenced in this project is achieved with a team of geoscientists and engineers.
Background

Reservoir simulation is a mature science. Significant advances have been made in the last few years. Simulation is increasingly being used for reservoir development planning, forecasting and management. Most of the commercial and state of the art reservoir simulators use structured grids utilizing the finite-difference method. At the University of Utah, a series of finite-element simulators for fractured reservoirs have been developed. The feasibility of using finite-element reservoir simulators was demonstrated by showing an excellent match between the performances of these simulators and Eclipse the most commonly used finite-difference simulator. These simulators, utilizing the unstructured grid system, are more suitable to represent and model complex geometries and domains.

The unstructured grid systems would also be more suitable in modeling complex well systems. The advantages of using completely unstructured grids have been demonstrated in medical applications, where the grids body-fit the geometrical domain. This project will develop the infrastructure for the use of fully unstructured grids in reservoir simulation. The finite-difference simulators have had a 50-year history and are well-tuned to handle reservoir problems. The finite-element simulators need to be optimized for speed. The use of finite-element simulators is common in aerospace and in CFD (computational fluid mechanics). Lessons from these fields need to be brought into the area of reservoir simulation. The simulators at the University of Utah use PETSC (Portable Extensible Toolkit for Scientific Computation). The most modern linear and nonlinear solvers are used in this package, developed at the Argonne National Laboratory. Enhancements in the form of multigridding methods, generalized parallel computing for domains with complex set of fractures and wells are needed.
An oil reservoir may undergo several recovery processes (water flooding, gas flooding, thermal recovery, etc) over its lifetime. The current simulator structure requires the use of different simulators to study each phase in the life of the simulation. Constructing a reservoir simulator using different modules will allow examination of different recovery processes using the same geologic data set and grid. Compositional and thermal simulation modules will be built in this project to allow the study of compositional and thermal processes with a fully unstructured grid and with discrete fractures and faults (if, necessary). A structure to allow this type of development, much more easily than in the past, is also necessary and will be built first.

Well geometries are also becoming increasingly complex. Drilling maximum reservoir contact wells (MRC) is common worldwide. It will be difficult, if not impossible to represent and model these wells using the current practice of structured grids. When wells with complex geometries are placed in structured grids, complicated well models are required. New well models for unstructured grid geometries are needed and are proposed in this project.

Geometrical construction of the reservoir domain is performed manually in most cases, making the task cumbersome, particularly for complex systems. When faults and fractures are an integral part of the reservoir domain, uncertainty in creating a reliable geologic data set is higher than when the reservoir is not fractured. A rule–based algorithm to create and map fractures and simulate the resulting set of realizations quickly (“on-the-fly”) to match field observations is necessary to reduce the uncertainty in the geologic model. This will also be accomplished in this project.

The growth in the reservoir simulation technology has been explosive, but also haphazard to a certain degree. Since there are so many different protocols, it is difficult to utilize and transfer the advances in some components to other simulation features. Some level of
component standardization is necessary to overcome this problem and provide a plug and play type of environment for integrated reservoir simulation. At a recent Reservoir Simulation Forum sponsored by the SPE (Forum Notes, 2003), this capability was one of the most desired by the industrial participants. The U.S. Department of Energy (the SciDAC program) has taken the lead in establishing Common Component Architecture (CCA) in a number of high-performance computing applications.

Some of the common components in reservoir simulation are

- Physical models,
- Numerical algorithms
- Well models
- Petrophysical models
- Gridding
- Parallel computing

A CCA platform would integrate these applications so that advances or modifications in one would have an immediate impact on the efficacy of the complete application.

Parallel computing is not widely used in reservoir simulation. This is because the technology is still out of reach of most users. Use of parallel computing would allow simulation of extremely complex domains with millions of unstructured grid blocks. Parallel computing is built into the University of Utah discrete-fracture, finite-element simulators. It is important to generalize the parallel computing concept for unstructured grids. From the user perspective, this development needs to be seamless. The problem definition and proposed solution is outlined in a schematic in Figure 1.
Figure 1: The problems in being able to simulated complex (realistic) reservoir domains and the solutions proposed

Problems:
- Complex domains with fractures, faults
- Faster, efficient finite-element simulators
- Different recovery mechanisms over reservoir life
- Complicated wells (multilaterals, MRC wells)
- Rapid validation of generated fracture networks
- No standards for components in a simulator
- Difficulty of parallel computing

Solutions:
- Use of unstructured grids (finite-element) body-fitting gridding
- Multigrid, multiscale methods
- Jacobian calculations
- Separation of reservoir model and the numerical method
- Addition of modular thermal and compositional modules for use with finite-element grids
- Innovative well models for unstructured grid systems
- Rule-based algorithms to generate data, grid and simulate rapidly and validate using available field data
- Use of Common Component Architecture, a scheme pioneered by the U.S. DOE
- Data distribution and parallel computing at the touch of a button for heterogeneous systems
Objectives

The goals of this project are based on the prior discussion of problems, proposed solutions and the state of the art, and are enumerated below.

1. This project will focus on developing a fast finite-element simulator capable of performing compositional and thermal simulations on complex fractured, faulted systems. Such a simulator is currently not available. The simulator will provide the means of studying all possible physical processes using a grid that is created to fit the reservoir (and well) geometries.

2. This project will develop new multiscale methods to model large fractured systems. Most reservoirs are fractured to a certain degree. Hence, this development will have a number of industrial applications.

3. Intuitive well models for the finite-element simulators will be developed. These will streamline the representation of simple (vertical, horizontal or slanted) and complicated (multilateral, maximum reservoir contact) wells. The need to develop special well models for specific well types and grid geometries will be eliminated.

4. The project will develop a process to map and verify faults and fractures. Integration of data into rule-based algorithms, quick gridding and simulation using the generated domain and verification with existing production data, will be the components of this process.

5. New parallel schemes for dealing with data sets with a large number of fractures will be developed in this project. These schemes will make fractured reservoir simulation with a large number of cells (elements) practical.

6. A platform to standardize the various components of reservoir simulation will be created. Common Component Architecture (CCA) will be used. This platform will be a
demonstration of how completely different components can be linked in a plug and play mode. It will be possible to create a repository of interconnected software modules that is accessible and usable by the entire community.
Results and Discussion

Simulator Development
This task was concerned with improving the performance of the existing black-oil simulator and adding compositional and thermal modules to the simulator. The simulators can use the most modern linear and nonlinear solvers by linking with PetSc. Most of the simulators use the GMRES linear solver in PetSc. The use of multigrid preconditioners to further optimize the performance of the simulators is being explored.

This task also required the decoupling of the reservoir model (called the physical model in this report) and the discretization method. This was achieved by creating two separate modules for the physical model (PM) and the discretization method (DM). The idea was to be able to combine different physical models (black oil, compositional, thermal, etc.) with the discretization method of choice (finite difference, finite element, etc.). This decoupling was suitable for any problems which were described, in general by a conservation law of the type shown below.

\[ \sum_{i} u_i + \frac{\partial \rho}{\partial t} = 0 \]

In this equation, \( u \) represents any type of flux (material or heat) and \( \rho \) represents the conserved property (saturation or energy). The DM defines the number of control volumes and transmits that information to PM. The PM has knowledge of the properties of the control volume. Subsequently, DM will query the required pressure (or temperature) information from PM and compute flux according to the geometric information of elements. This gradient information is passed back to PM upon which PM attaches the model specific properties (relative
permeabilities, specific heats, etc.) to form the model prescribed flux. This is combined with the accumulation term to create residual functions.

For example, the residual function for each phase in a black oil model is described by:

$$R_i = \frac{k_i}{B\mu} (f_p + \rho g f_z) + \frac{\Delta V}{\Delta t} \left( \phi S \right)_{t+1} - \left( \phi S \right)_t$$

where:

- $f_p = -k \nabla P$
- $f_z = -k \nabla z$

Based on the concepts described, $f_p$, $f_z$ and all direct derivatives are calculated by DM while the property-based terms and derivatives are calculated by PM.

Work on the development of a standard finite-element compositional simulator was begun. As first part of this, a thermodynamic simulator that would provide vapor and liquid compositions and amounts was written and verified. This simulator was written using the successive substitution method. A Newton’s algorithm is now being applied to solve the same problem. The compositional flow equations were written. The degree of uncoupling described earlier would require that only the physical compositional equation module be written. The numerical method and meshing will already be in place and will be linked to the new reservoir module.

In a similar fashion, formulation of the thermal reservoir model with the oil, steam and gas phases was completed. The code for both the compositional and thermal modules is almost complete.
Development of a Compositional Module

Introduction

Fluids in petroleum reservoirs contain thousands of chemical components that influence the physical properties and phase behavior during oil production. It is impractical to describe petroleum fluids in terms of all individual components.

Pseudo components (groups of molecules) with average physical properties are used to describe the reservoir fluid. Reservoir fluids are classified into various types based on the formation value factor and API gravity as black oils, volatile oils, retrograde gas, wet gas etc.

Black oil models are used to study the conventional recovery techniques in reservoirs for which fluid properties can be expressed as functions of pressure and bubble point pressure.

Compositional models are used when either in-place or injected fluid causes fluid properties to be dependent on the composition also. Examples of problems generally requiring compositional models are (1) primary production or injection process (such as nitrogen injection) into gas condensate and volatile oil reservoirs, (2) enhanced recovery from reservoirs by CO2 or enriched gas injection. The black oil model is a special case of compositional model, where the phase equilibrium relations can be reduced to linear relations between pressure and component mole fraction.

Sometimes it is inappropriate to use a black oil model, and oil and gas must be represented by more than two pseudo components. The phase behavior, in this case is represented by an Equation of States (EOS) and phase equilibrium relations.
The number of equations in a compositional model increases from that of a simple black oil model and the choice of equations and variables gets more complex. The performance of the simulator (convergence of non linear and linear iterations) can be strongly influenced by these factors.

**Formulation of the Equations**

Compositional models can be classified as mass balance type and volume balance type depending on the equations used to solve. Mass balance type is most commonly used and is discussed here. The mass balance equation for each hydrocarbon component and water are the equations for this type of model.

\[
F_i = \frac{\partial}{\partial t} [V \phi (S_{\omega} \rho_\omega x_i + S_{g} \rho_g y_i)] - \sum_l [T(\lambda_{\omega} \rho_\omega x_i \Delta \Phi_\omega + \lambda_g \rho_g y_i \Delta \Phi_g)] \\
+ \sum_w \left( \rho_\omega x_i q^w_\omega + \rho_g y_i q^w_g \right) = 0
\]

\[
F_w = \frac{\partial}{\partial t} [V \phi (S_w \rho_w)] - \sum_l [T(\lambda_w \rho_w \Delta \Phi_w)] + \sum_w (q^w_w) = 0
\]

Phase equilibrium relations for each hydrocarbon component are needed in addition to the mass balance equations. Water and hydrocarbon components are completely separated and hence only two-phase equilibrium between the oil and gas phases is considered. The condition for vapor liquid equilibrium from thermodynamics is the equality of the fugacity in the liquid and vapor phase. Hence, we have

\[
F_{e,i} = f^l_i - f^v_i = 0, \quad i = 1, \ldots, n_e
\]
There are some linear constraints in addition to the mass balance and phase equilibrium equations that complete the system of equations. The constraints to be satisfied are:

- Mole balance constraints

\[
\sum_{i=1}^{n_c} x_i - 1 = 0
\]
\[
\sum_{i=1}^{n_c} y_i - 1 = 0
\]

- Saturation or volume constraint

\[
\sum_{i=1}^{n_p} S_p - 1 = 0
\]
\[
or V_\phi - V_T = 0
\]

- Capillary pressure constraint

\[
P_{cow} - (P_o - P_w) = 0
\]
\[
P_{cog} - (P_g - P_o) = 0
\]

Hence, the \(2n_c + 1\) non linear equations and five linear constraint equations make the \(2n_c + 6\) equations that represent the compositional model.

The \(2n_c + 1\) non linear equations are the mass balance equations and the phase equilibrium relations, of which the \(n_c + 1\) independent mass balance equations constitute the primary equations. The primary equations are those needed to be solved to fix both the intensive and extensive state of the system. They are also the equations solved simultaneously in a fully implicit model. The mass balance equations are a natural choice since their number equals the
number of primary equations. The remaining nonlinear equations form the secondary equations are solved after the primary equations.

Different researchers have put forth various models and they differ in the variables and equations are chosen. Coats model is used in this work.

The \( n_c + 1 \) mass balance equations for the \( n_c \) components and that of water constitute the primary equations and the equality of fugacity constitutes the secondary equations. The variables solved for in this model are:

- \( P, S_o, S_g, x_i, y_i, i = 2, \ldots, n_c \) (when both oil and gas phases are present)
- \( P, S_o, S_g, x_i, i = 2, \ldots, n_c \) (when only oil phases is present)
- \( P, S_o, S_g, y_i, i = 2, \ldots, n_c \) (when only gas phases is present)

The variables for different control volume may differ and we need to switch variables when a hydrocarbon phase disappears or reappears. The variable selection in this model is called the natural variable selection, since the equations can be directly expressed in terms of these variables and evaluation of the derivatives in the Jacobian is easy. The model was proposed as a fully implicit model, but can be reduced to IMPES by variable elimination.

**Flash Calculations**

Compositional models are more complicated than black oil models in their formulation. Moreover, they require additional flash calculations. The phase equilibrium relations can be reduced to a set of linear constraints between component mole fractions and pressure in black oil models. Hence, black oil model is a special case of compositional model.
Compositional simulation require flash calculations to be performed at every Newton iteration of each time step.

At a given temperature and pressure, flash calculations yield the number of moles of component $i$ in the liquid phase, and in the vapor phase, given the total number of moles of each component in the feed. If the mole fraction of each component in the fluid is known, we solve for the liquid and vapor mole fractions. The condition for equilibrium is given by,

$$F_{e_i} = f^l_i - f^v_i = 0,$$  $i = 1, \ldots, N_c,$

which are systems of nonlinear equations. This system can be solved by a direct successive substitution method or Newton's method.

In practice equilibrium ratios $K_i$ are introduced.

$$K_i = \frac{y_i}{x_i}$$

The overall mole fraction is related to the liquid and vapor mole fraction as

$$z_i = x_i (1 - f_i) + y_i f_v = 0$$

Where, $f_v$ is the hydrocarbon vapor mole fraction in the fluid.

Mole fractions $x_i$ and $y_i$ can further be expressed in terms of $f_v, z_i,$ and $K_i;$ and in practice, equilibrium ratios $K_i$ are introduced.
The above equations are combined to nonlinear equation that is solved iteratively.

\[
\begin{align*}
x_i &= \frac{z_i}{1 + f_v(1 - K_i)} \\
y_i &= \frac{z_i(K_i - i - 1)}{1 + f_v(1 - K_i)}
\end{align*}
\]

It can be shown that \( K_i \) is related to the fugacity coefficients \( \phi_i \) and \( \phi_v \) as follows.

\[
K_i = \frac{\phi_i}{\phi_v} = \frac{f_d / (x_i P)}{f_v / (y_i P)} = \frac{f_d}{f_v} \frac{x_i}{y_i}
\]

1. Solve for \( f_v \)
2. Computed \( x_i, y_i \) from equations shown above
3. Calculate compressibility factor for each phase from an Equation of State (EOS)
4. Calculate the fugacity of each component in liquid and vapor phases.
5. Check for convergence.
6. If not converged, update equilibrium ratios from equation for fugacities.
The Newton method involves computing derivatives of the equilibrium equation in the Jacobian matrix. The equations are solved for one of the following sets of variables:

- \( f_i, x_i, i = 2, \ldots, n_c \)
- \( f_v, y_i, i = 2, \ldots, n_c \)

The successive substitution method converges slowly but convergence is usually achieved, whereas the Newton method converges rapidly near the true solution. So, a typical approach is to use successive substitution method first, after the error is reduced to a certain level, the Newton's method is invoked for rapid convergence.

The flash calculation procedure is also used to compute the phase properties. The molecular weight of a phase is given by

\[
MW_o = \sum_{i=1}^{n_c} MW_i \cdot x_i MW_g = \sum_{i=1}^{n_c} MW_i \cdot y_i
\]

The density of a phase \( p \) is computed by the following relation

\[
\rho_p = \frac{P_p}{Z_p RT}
\]
where \( Z_p \) is one of three roots of the cubic EOS. The smallest root is used for the oil phase and the largest for the gas phase, the intermediate root representing an unstable phase.

**Equation of State Used**

The hydrocarbon liquid and gas phase densities and component fugacities are computed from a cubic equation of state such as Redlich–Kwong, Peng–Robinson equation of state. The Peng–Robinson equation of state used in this case is given by

\[
P = \frac{RT}{(V - b)} - \frac{a}{V(V + b) + b(V - b)}
\]

where

\[
a = \sum_{i=1}^{n_c} \sum_{j=1}^{n_c} x_i x_j a_{ij},
\]

\[
a_{ij} = (1 - k_{ij}) \sqrt{a_i a_j},
\]

\[
a_i = \frac{0.45724 R^2 T_c^2}{P_c} \cdot \alpha,
\]

\[
b_i = \frac{0.07780 R T_c}{P_c},
\]

\[
\alpha = \left[ 1 + (0.37464 + 1.5422 \omega - 0.26992 \omega^2)(1 - T_r^{0.5}) \right]^2,
\]

\[
A = \frac{a P}{R^2 T^2},
\]

\[
A = \frac{b P}{RT},
\]

The equation can also be written in a cubic form as:
\[ Z^3 - (1 - B)Z^2 + (A - 3B^2 - 2B)Z - (AB - B^2 - B^3) = 0 \]

The compressibilities of the liquid and the gas phases are obtained by solving the cubic equation of state with their respective compositions. The largest root is the compressibility for the gas phase and the smallest root that of the oil phase.

The fugacity of a component \( i \) is given by

\[ f_i = x_i e^{\phi_i} P \]

and the exponential term \( \phi_i \) in the fugacity coefficient term is given by

\[
\phi_i = \frac{b_i}{b} (Z - 1) - \ln(Z - B), \\
+ \frac{A}{2\sqrt{2}B} \left( \frac{b_i}{b} - \frac{1}{a \partial x_i} \right) \ln \left[ \frac{Z + (1 + \sqrt{2})B}{Z + (1 - \sqrt{2})B} \right] \\
\frac{\partial a}{\partial x_i} = \sum_{j=1}^{n_c} x_j a_{ij}
\]

**Verification of Flash Calculation**

The flash algorithm discussed above with a Peng--Robinson equation of state was tested for a 3 component and a 10 component system. The results of the flash were compared with the results of an isothermal flash from a commercial simulator Winprop of CMG. A three component system consisting methane, butane and decane was chosen as a test system. An isothermal flash was performed at 1600 psi and 168\(^0\) F. The results of the flash calculation are presented in the Table 1.
Ten component system

A ten component system comprising of C1, C2, C3, C4, C5, C6, C7, C8, C9 and C10 was chosen as a test system. An isothermal flash was performed at 2750 psi and 220°F. The results of the flash calculation are presented in the Table 2.

<table>
<thead>
<tr>
<th>Component</th>
<th>Initial Composition</th>
<th>Winprop</th>
<th>Ufes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z</td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>CO2</td>
<td>0.88</td>
<td>0.799144</td>
<td>0.945281</td>
</tr>
<tr>
<td>C5</td>
<td>0.06</td>
<td>0.081984</td>
<td>0.042251</td>
</tr>
<tr>
<td>C16</td>
<td>0.06</td>
<td>0.118872</td>
<td>0.012468</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Component</th>
<th>Initial Composition</th>
<th>Winprop</th>
<th>Ufes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>z</td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>C1</td>
<td>0.6436</td>
<td>0.535754</td>
<td>0.784518</td>
</tr>
<tr>
<td>C2</td>
<td>0.0752</td>
<td>0.074902</td>
<td>0.075589</td>
</tr>
<tr>
<td>C3</td>
<td>0.0474</td>
<td>0.052732</td>
<td>0.040433</td>
</tr>
<tr>
<td>C4</td>
<td>0.0412</td>
<td>0.050253</td>
<td>0.029371</td>
</tr>
<tr>
<td>C5</td>
<td>0.0297</td>
<td>0.038925</td>
<td>0.017646</td>
</tr>
<tr>
<td>C6</td>
<td>0.0138</td>
<td>0.019155</td>
<td>0.006803</td>
</tr>
<tr>
<td>C7</td>
<td>0.0303</td>
<td>0.044059</td>
<td>0.012321</td>
</tr>
<tr>
<td>C8</td>
<td>0.0371</td>
<td>0.055902</td>
<td>0.012533</td>
</tr>
<tr>
<td>C9</td>
<td>0.0415</td>
<td>0.064423</td>
<td>0.011548</td>
</tr>
<tr>
<td>C10</td>
<td>0.0402</td>
<td>0.063896</td>
<td>0.009238</td>
</tr>
</tbody>
</table>

In a control volume, there may be up to two hydrocarbon phases (oil and gas). The pressure, temperature and overall composition determine whether two hydrocarbon phases exist or not.
The existence of hydrocarbon phases needs to be established during simulation. If only a single hydrocarbon phase exists, equations and variables for some models may have to be changed.

When a control volume has two hydrocarbon phases, we need to monitor its condition, to find when it will change to a single hydrocarbon phase. For each Newton iteration, we solve the flow equations, and get the saturation solutions. If either \( S_o \) or \( S_g \) is negative, it is set to zero before initializing for the next iteration. Then the overall mole fractions is set to the oil or gas saturation.

Negative flash could also be used to ascertain the phase. A flash calculation is performed. If the liquid mole fraction, \( f_l < 0 \), then the oil phase does not exist anymore, else if \( f_l > 1 \), then the gas phase does not exist anymore.

- If gas phase disappears, \( S_g \leq 0 \), set \( S_g = 0 \) and \( y_e = 0 \)
- If oil phase disappears \( S_o \leq 0 \), set \( S_o = 0 \) and \( x_e = 0 \)

When a control volume has only one hydrocarbon phase, we need to monitor its condition for the reappearance of the oil or gas phase. When the the control volume changes to two hydrocarbon phases we may need to change the equations and variables.

When a control volume has only one hydrocarbon phase at the end of a Newton iteration (i.e. \( S_g \) or \( S_o \) is zero), a saturation pressure is computed for the single phase control volume. If the calculated saturation pressure is less than the control volume pressure, the control volume
remains in the single hydrocarbon phase, otherwise both hydrocarbon phases exist and the absent
phase saturation is set to 0.001.

When a control volume has only one hydrocarbon phase at the end of a Newton iteration, we can
also perform a negative flash as mentioned above with pressure and overall composition as the
inputs. If \( f_i \) is between 0 and 1, then both hydrocarbon phases exist and phase saturations are
initialized once again.

Saturation pressure is calculated in the manner proposed by Fussel and Yanosik. They proposed
the solution of \( n_c + 1 \) equations.

\[
F_{e,i} = f_i^l - f_i^v = 0, \quad i = 1, \ldots, n_c,
\]

\[
Ps - \sum_{j=1}^{n_c} \frac{f_j^l}{\phi_j^v} = 0
\]

for the \( n_c + 1 \) unknowns using the Newton--Raphson method to obtain the saturation pressure of
a liquid phase. Improved convergence has been found by replacing the above equation with:

\[
\sum_{j=1}^{n_c} y_j - 1 = 0
\]

For the liquid-phase compositions and saturation pressure, the following equations are used.

\[
Ps - \sum_{j=1}^{n_c} \frac{f_j^v}{\phi_j^v} = 0
\]

\[
\sum_{j=1}^{n_c} x_j - 1 = 0
\]

This saturation pressure computation is performed for each control volume where
or $S_g$ or $S_c$ are zero. The recalculated saturation pressure and absent phase composition are stored and used as starting values for the next iteration's Newton Raphson solution of equations.

The saturation pressure computation algorithm discussed was tested for a 3 component, and 10 component. The results of the saturation pressure computation were compared with the results of a commercial simulator Winprop of CMG.

A three component system with methane, butane and decane was chosen as a test system. The initial composition was 0.5, 0.3 and 0.2 respectively. The saturation pressure and vapor phase composition was computed at 168°F. The results of the flash calculation are presented in the Table 3.

<table>
<thead>
<tr>
<th>Component</th>
<th>Winprop</th>
<th>Ufes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
<td>y</td>
</tr>
<tr>
<td>CO₂</td>
<td>0.5</td>
<td>0.945281</td>
</tr>
<tr>
<td>C₅</td>
<td>0.3</td>
<td>0.042251</td>
</tr>
<tr>
<td>C₁₆</td>
<td>0.2</td>
<td>0.012468</td>
</tr>
</tbody>
</table>

A ten component example is shown in Table 4.
Development of a Thermal Module

As a first step toward development of a thermal simulator, a black oil model which accounts for a water phase and an oil phase was formulated. Energy balance is added to see the temperature distribution in the oil reservoir. The non-flow heat transfer due to heat conduction of the solid matrix is also evaluated in the energy balance equation by Fourier’s law. Fluid fluxes are calculated from Darcy’s equation.

The structure of simulators being developed at University of Utah allows coupling of different physical models (PM) to various discretization models (DM). The thermal model being developed will first be linked to the two-dimensional, block-centered finite element DM. Once results are found satisfactory, it will be coupled to a three-dimensional DM. The ultimate goal of thermal model is to add the energy balance equation to the compositional model (under development) to observe the distillation effect.

Governing equations of the thermal model are shown below.

<table>
<thead>
<tr>
<th>Component</th>
<th>Winprop x</th>
<th>Winprop y</th>
<th>Ps(atm)</th>
<th>Ufes x</th>
<th>Ufes y</th>
<th>Ps(atm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₁</td>
<td>0.6436</td>
<td>0.739661</td>
<td>219.16</td>
<td>0.6436</td>
<td>0.740497</td>
<td></td>
</tr>
<tr>
<td>C₂</td>
<td>0.0752</td>
<td>0.074794</td>
<td></td>
<td>0.0752</td>
<td>0.074771</td>
<td></td>
</tr>
<tr>
<td>C₃</td>
<td>0.0474</td>
<td>0.042396</td>
<td></td>
<td>0.0474</td>
<td>0.042333</td>
<td></td>
</tr>
</tbody>
</table>
Mass balance:
\[
\frac{\partial S_i \rho_i}{\partial t} = -\nabla \cdot \rho_i \vec{f}_i + \Phi_i^M
\]
i: Phase (1=Oil, 2=Water)
\(\vec{f}_i\): Volumetric flow flux of phase i (ft/day)
\(\rho_i\): Molar density of phase i (lbmol/cu-ft)
\(\Phi_i^M\): Source term of phase i (lbmol/cu-ft day)

Energy balance:
\[
\frac{\partial (\phi \sum_{i=1}^{\text{phase}} S_i \rho_i \tilde{U}_i + (1-\phi) \rho_R \tilde{U}_R)}{\partial t} = -\sum_{i=1}^{\text{phase}} \nabla \cdot \rho_i \vec{f}_i \tilde{H}_i + \nabla \cdot k_c \nabla T + \Phi^E
\]
\(\tilde{U}_i\): Molar internal energy of phase i (Btu/lbmol)
\(R\): Rock
\(\tilde{H}_i\): Molar enthalpy of phase i (Btu/lbmol)
k_c: Heat conductivity of rock (Btu/ft day R)
\(\Phi^E\): Source term (Btu/cu-ft day)

For the calculation of the volumetric flow flux, Darcy’s equation is used.
\[
\vec{f}_i = -\frac{kk_{rij}}{\mu_i} \nabla \cdot \Psi_i
\]
k: Absolute permeability (sq-ft)
k_{rij}: Relative permeability
\(\mu_i\): Viscosity (lbf day/sq-ft)

\(\Psi_i\) [= 1/(144) \(P_i + \rho_i \frac{g}{g_c} Z\)]: Potential of phase i (lb/sq-ft)
\(P_i\): Pressure of phase i (lb/sq-ft)
\(\frac{g}{g_c}\) [= 1]: Gravity const (lb/sq-ft)
\(Z\): Deep (ft)

In solving the above equations, dependent variables must be chosen. The following dependent variables were found to be appropriate.

P_o: Oil phase pressure
S_w: Water saturation
T: Temperature
Equations are solved using these additional constraints.

Saturation: \( S_w + S_o = 1 \)

Capillary pressure: \( P_{cw} = P_o - P_w \)

\[ P_{cw} = f(S_o) \]

It is necessary to choose fluid property relationships. The fluid property relationships used along with appropriate derivatives (necessary for Jacobian calculations) are shown below.

Fluid properties:

**Viscosity:**

\[ \mu_i = f(T) \]

\[
\begin{align*}
\mu_o &= A_o + e^{\frac{B_o}{T}} \\
\mu_w &= \frac{A_w}{B_w(T-T_o) + C_w(T-T_o)^2 - 1.0}
\end{align*}
\]

\[
\frac{\partial \mu_o}{\partial S_w} = 0 \\
\frac{\partial \mu_w}{\partial S_w} = 0
\]

\[
\frac{\partial \mu_o}{\partial T} = -B_o e^{\frac{B_o}{T}} T^{-2}
\]

\[
\frac{\partial \mu_w}{\partial T} = -\frac{A_w (B_w + 2(T-T_o))}{(B_w(T-T_o) + C_w(T-T_o)^2 - 1.0)^2}
\]

\[
\frac{\partial \mu_o}{\partial P_o} = 0 \\
\frac{\partial \mu_w}{\partial P_o} = 0
\]

**Density:**
\[ \rho_i = f(T, P_i) \]

\[
\begin{align*}
\rho_o &= A_o (B_o + C_o T) e^{D_o (P_o - E_o)} \\
\rho_w &= A_w (B_w + C_w T) e^{D_w (P_w - E_w)}
\end{align*}
\]

\[
\frac{\partial \rho_o}{\partial S_w} = 0 \quad \frac{\partial \rho_w}{\partial S_w} = 0
\]

\[
\frac{\partial \rho_o}{\partial T} = A_o C_o e^{D_o (P_o - E_o)} \quad \frac{\partial \rho_w}{\partial T} = A_w C_w e^{D_w (P_w - E_w)}
\]

\[
\frac{\partial \rho_o}{\partial P_o} = A_o (B_o + C_o T) D_o e^{D_o (P_o - E_o)} \quad \frac{\partial \rho_w}{\partial P_w} = A_w (B_w + C_w T) D_w e^{D_w (P_w - E_w)}
\]
\textbf{Enthalpy:}
\[ H_i = f(T) \]
\[ \{ \begin{split} H_O &= H^0_O + C_{P_{AO}}(T - T_0) + \frac{C_{P_{BO}}}{2}(T^2 - T_0^2) + \frac{C_{P_{CO}}}{3}(T^3 - T_0^3) + \frac{C_{P_{DO}}}{4}(T^4 - T_0^4) \\ H_W &= H^0_W + C_{P_{AW}}(T - T_0) + \frac{C_{P_{BW}}}{2}(T^2 - T_0^2) + \frac{C_{P_{CW}}}{3}(T^3 - T_0^3) + \frac{C_{P_{DW}}}{4}(T^4 - T_0^4) \end{split} \} \]
\[ \begin{aligned} \frac{\partial H_i}{\partial S_w} &= 0 \\ \frac{\partial H_w}{\partial S_w} &= 0 \end{aligned} \]
\[ \frac{\partial H_i}{\partial T} = \begin{cases} \frac{\partial H_O}{\partial T} &= C_{P_{AO}} + C_{P_{BO}}T + C_{P_{CO}}T^2 + C_{P_{DO}}T^3 \\ \frac{\partial H_W}{\partial T} &= C_{P_{AW}} + C_{P_{BW}}T + C_{P_{CW}}T^2 + C_{P_{DW}}T^3 \end{cases} \]
\[ \begin{aligned} \frac{\partial H_i}{\partial P_o} &= 0 \\ \frac{\partial H_w}{\partial P_o} &= 0 \end{aligned} \]

\textbf{Internal energy:}
\[ \vec{U}_i = f(T, P_i) \]
\[
\begin{align*}
\vec{U}_o &= \vec{H}_o - \frac{P_o}{\rho_o} \\
\vec{U}_w &= \vec{H}_w - \frac{P_w}{\rho_w}
\end{align*}
\]
\[
\frac{\partial \vec{U}_i}{\partial S_w} = \begin{cases}
\frac{\partial \vec{U}_o}{\partial S_w} = 0 \\
\frac{\partial \vec{U}_w}{\partial S_w} = 0
\end{cases}
\]
\[
\frac{\partial \vec{U}_i}{\partial T} = \begin{cases}
\frac{\partial \vec{U}_o}{\partial T} = \frac{\partial \vec{H}_o}{\partial T} + \frac{P_o}{\rho_o} \frac{\partial \rho_o}{\partial T} \\
\frac{\partial \vec{U}_w}{\partial T} = \frac{\partial \vec{H}_w}{\partial T} + \frac{P_w}{\rho_w} \frac{\partial \rho_w}{\partial T}
\end{cases}
\]
\[
\frac{\partial \vec{U}_i}{\partial P_o} = \begin{cases}
\frac{\partial \vec{U}_o}{\partial P_o} = -\frac{1}{\rho_o} + \frac{1}{\rho_o^2} \frac{\partial \rho_o}{\partial P_o} \\
\frac{\partial \vec{U}_w}{\partial P_o} = -\frac{1}{\rho_w} + \frac{1}{\rho_w^2} \frac{\partial \rho_w}{\partial P_o}
\end{cases}
\]
Rock property correlations are also essential in completing the formulation.

**Rock properties:**

**Porosity:**

\[ \phi = f(P_o) \]

\[ \phi = \phi^0 (1 + C_R (P_o - P^0)) \]

\[ \frac{\partial \phi}{\partial S_w} = 0 \]

\[ \frac{\partial \phi}{\partial T} = 0 \]

\[ \frac{\partial \phi}{\partial P_o} = \phi^0 C_R \]

**Internal energy:**

\[ \dot{U}_R = f(T) \]

\[ \dot{U}_R = \dot{U}_R^0 + C P_R (T - T_{R\text{\_initial}}) \]

\[ \frac{\partial \dot{U}_R}{\partial S_w} = 0 \]

\[ \frac{\partial \dot{U}_R}{\partial T} = C P_R \]

\[ \frac{\partial \dot{U}_R}{\partial P_o} = 0 \]

**Density:**

\[ \rho_R = \text{const} \]

\[ \rho_R = \text{const} \]

\[ \frac{\partial \rho_R}{\partial S_w} = 0 \]

\[ \frac{\partial \rho_R}{\partial T} = 0 \]

\[ \frac{\partial \rho_R}{\partial P_o} = 0 \]

**Thermal conductivity:**

\[ k_c = \text{const} \]

\[ k_c = \text{const} \]

\[ \frac{\partial k_c}{\partial S_w} = 0 \]

\[ \frac{\partial k_c}{\partial T} = 0 \]

\[ \frac{\partial k_c}{\partial P_o} = 0 \]
Absolute permeability:

\[ k = \text{const} \]
\[ \frac{\partial k}{\partial S_w} = 0 \]
\[ \frac{\partial k}{\partial T} = 0 \]
\[ \frac{\partial k}{\partial P_o} = 0 \]

Relative permeability:

\[ k_{ri} = f(S_i) \]
\[ \frac{\partial k_{ri}}{\partial S_w} = \text{lookup} \]
\[ \frac{\partial k_{ri}}{\partial T} = 0 \]
\[ \frac{\partial k_{ri}}{\partial P_o} = 0 \]

The equations are reduced to the mobility form to help construction of the Jacobian matrix.

Mass balance:

\[ \frac{\partial S_i \phi_i}{\partial t} = \nabla \cdot M_{ijkl} k \nabla \psi_i + \Phi_i^M \]

Oil phase: \( M_{OM} = \frac{k_{ro} \rho_o}{\mu_o} \)

\[ \frac{\partial M_{OM}}{\partial S_w} = \frac{\rho_o}{\mu_o} \frac{\partial k_{ro}}{\partial S_w} \]
\[ \frac{\partial M_{OM}}{\partial T} = \frac{k_{ro} \rho_o}{\mu_o} \frac{\partial \rho_o}{\partial T} - \frac{k_{ro} \rho_o}{\mu_o^2} \frac{\partial \mu_o}{\partial T} \]
\[ \frac{\partial M_{OM}}{\partial P_o} = \frac{k_{ro} \rho_o}{\mu_o} \frac{\partial \rho_o}{\partial P_o} \]

Water phase: \( M_{WM} = \frac{k_{rw} \rho_w}{\mu_w} \)

\[ \frac{\partial M_{WM}}{\partial S_w} = \frac{\rho_w}{\mu_w} \frac{\partial k_{rw}}{\partial S_w} \]
\[ \frac{\partial M_{WM}}{\partial T} = \frac{k_{rw} \rho_w}{\mu_w} \frac{\partial \rho_w}{\partial T} - \frac{k_{rw} \rho_w}{\mu_w^2} \frac{\partial \mu_w}{\partial T} \]
\[ \frac{\partial M_{WM}}{\partial P_o} = \frac{k_{rw} \rho_w}{\mu_w} \frac{\partial \rho_w}{\partial P_o} \]
Energy balance:

\[
\frac{\partial (\phi \sum_{i=1}^{\text{Phase}} S_i \rho U_i + (1-\phi) \rho R \dot{U}_R)}{\partial t} = \sum_{i=1}^{\text{Phase}} \nabla \cdot M_i \varepsilon_i \nabla \Psi_i + \nabla \cdot k_C \nabla T + \Phi^E
\]

Oil phase: \( M_{OE} = \frac{k_{oE} \rho_O}{\mu_O} \dot{H}_O \)

\[
\frac{\partial M_{OE}}{\partial S_W} = \frac{\rho_O \dot{H}_O}{\mu_O} \frac{\partial k_{oE}}{\partial S_W}
\]
\[
\frac{\partial M_{OE}}{\partial \dot{T}} = \frac{k_{oE} \rho_O \dot{H}_O}{\mu_O} \frac{\partial \mu_O}{\partial \dot{T}} + \frac{k_{oE} \rho_O}{\mu_O} \frac{\partial \dot{H}_O}{\partial \dot{T}}
\]
\[
\frac{\partial M_{OE}}{\partial P_O} = \frac{k_{oE} \rho_O}{\mu_O} \frac{\partial P_O}{\partial P_O}
\]

Water phase: \( M_{WE} = \frac{k_{wE} \rho_W}{\mu_W} \dot{H}_W \)

\[
\frac{\partial M_{WE}}{\partial S_W} = \frac{\rho_W \dot{H}_W}{\mu_W} \frac{\partial k_{wE}}{\partial S_W}
\]
\[
\frac{\partial M_{WE}}{\partial \dot{T}} = \frac{k_{wE} \rho_W \dot{H}_W}{\mu_W} \frac{\partial \mu_W}{\partial \dot{T}} + \frac{k_{wE} \rho_W}{\mu_W} \frac{\partial \dot{H}_W}{\partial \dot{T}}
\]
\[
\frac{\partial M_{WE}}{\partial P_O} = \frac{k_{wE} \rho_W}{\mu_W} \frac{\partial P_O}{\partial P_O}
\]

A general discretization procedure for the differential equations is shown.

Mass balance:

\[
R_{Res \text{idual}} = V \frac{S \phi \rho - (S \phi \rho)_{\text{Old}}}{\Delta t} + \sum_{C=1}^{N} \sum_{i=1}^{\text{Phase}} \nabla \cdot A_c \cdot M_{Mi} \varepsilon_i \nabla \Psi_i - \Phi_{\text{Source}}
\]

Energy balance:

\[
R_{E \text{Residual}} = V \frac{(\phi \sum_{i=1}^{\text{Phase}} S_i \rho U_i + (1-\phi) \rho R \dot{U}_R) - (\phi \sum_{i=1}^{\text{Phase}} S_i \rho U_i + (1-\phi) \rho R \dot{U}_R)_{\text{Old}}}{\Delta t} + \sum_{C=1}^{N} \sum_{i=1}^{\text{Phase}} A_c \cdot (M_{Mi} \varepsilon_i \nabla \Psi_i + k_C \nabla T) - \Phi_{\text{Source}}
\]
Finally, the residuals shown above are evaluated in an iterative manner by using the Newton’s method. Jacobian matrix required is calculated analytically.

\[ \vec{R} = \vec{J} \cdot \delta \vec{X} \]

\[
\begin{bmatrix}
\frac{\partial R_{On}}{\partial P_{On}} & \frac{\partial R_{On}}{\partial S_{Wn}} & \frac{\partial R_{On}}{\partial T_{m}} \\
\frac{\partial R_{Wn}}{\partial P_{On}} & \frac{\partial R_{Wn}}{\partial S_{Wn}} & \frac{\partial R_{Wn}}{\partial T_{m}} \\
\frac{\partial R_{Em}}{\partial P_{On}} & \frac{\partial R_{Em}}{\partial S_{Wn}} & \frac{\partial R_{Em}}{\partial T_{m}}
\end{bmatrix}
\begin{bmatrix}
\frac{\partial R_{On}}{\partial P_{On}} & \frac{\partial R_{On}}{\partial S_{Wn}} & \frac{\partial R_{On}}{\partial T_{m}} \\
\frac{\partial R_{Wn}}{\partial P_{On}} & \frac{\partial R_{Wn}}{\partial S_{Wn}} & \frac{\partial R_{Wn}}{\partial T_{m}} \\
\frac{\partial R_{Em}}{\partial P_{On}} & \frac{\partial R_{Em}}{\partial S_{Wn}} & \frac{\partial R_{Em}}{\partial T_{m}}
\end{bmatrix}
\begin{bmatrix}
P_{On} \\
S_{Wn} \\
T_{m}
\end{bmatrix}
\]

**Multiscale Methods**

When physical phenomena at different scales are coupled for simulation, the time step size needs to conform to the smallest one. This approach is computationally demanding. Multiscale simulation is the best technical approach to solve this problem. For example, in reservoir simulation local time-stepping method has been applied to grid refinement around wells. The goal is to adopt the multiscale method for fractured/faulted reservoir simulation. This will allow faster and more accurate fractured reservoir simulation. The simulation grid block volume of subsurface fractures and rock matrices differ usually over two to four orders of magnitude as shown in Figure 2. The fracture width has been enlarged for visibility.
Figure 2: A grid for fracture simulation.

When large volume differences appear in multiphase flow simulation, the time-step size has to be adjusted to accommodate the smallest volume in order to capture the saturation changes within the smallest control volume. In this case, small time-step size is required because of the presence of fractures. Consequently, small time steps have to be used for the entire simulation, leading to excessive computational effort. One solution to this problem is applying the multiscale method. The success of the multiscale method heavily depends on the synchronization of these two time scales. At time equals $t_1$ and $t_2$ these two time scales need to be synchronized to ensure global and local mass balance.

The first part of this activity required the incorporation of fractures/faults in finite element (MFE)-based simulations and examination of performance without multiscale. This task was completed and simulation results for a complicated domain with sloping fractures are shown in Figure 3.
Figure 3: Example of a three-dimensional, three-phase simulation with sloping discrete-fractures in a complex domain

In the development and verification of multiscale algorithms, it is important to be able to simulate a variety of boundary conditions at junctions of fractures or other features. It is now possible to simulate domains with no-flow boundary conditions in the interior of the domain. An example of this simulation in two-phase flow is shown in Figure 4. The faults shown in Figure 3 are now no-flow barriers. The water diversion around the faults to production wells is clearly seen.
Figure 4: The same faults shown in Figure 1 are modeled as no-flow. Water diversion and oil trapping (red) are seen.

**Wells for Unstructured Grids**

Basic well models (vertical, horizontal and slanted) developed for structured grids are not appropriate for simulating unconventional wells (multilaterals, MRCs, etc). The basic idea is to calculate the well bore pressure and well bore inflow within a box shaped domain that contains the entire well through the use of Green’s function. The box domain is then mapped to the real computational domain and the same boundary conditions are applied to solve for block pressures. Based on well bore pressure, well bore inflow rate and block pressure, a well index can be calculated for each well segment. Heterogeneity around such well is accounted for by introducing a skin factor along the well trajectory. Figure 5 shows this idea.
The above approach is complicated and requires significant preprocessing. After all the calculations, negative well indices may result. In this project, *intuitive* well models, fully integrated with the system grid will be developed for unstructured systems. These well models will not require special algorithms. Conventionally wells are placed at the center of a block in reservoir simulation with structured grids. A multiplication factor is necessary if the well is off-center. In the intuitive well model, the well will be placed at the corner. The advantage of this placement is more obvious for unstructured grids. The same non-conventional well in Figure 5 is shown in Figure 6 with a few tetrahedral elements surrounding one segment of the well. One advantage of this intuitive well model is that the unstructured grid used for the simulation completely conforms to the layout of the well. The flux between tetrahedral elements and the well can be calculated based on the pressure at the block center and well bore and the geometry of tetrahedral elements. The intuitive well model requires no preprocessing of any kind and heterogeneity around wells can be fully accounted for.

**Figure 5. Picture showing how unconventional wells are modeled in structured grids.**
Consider a nonconventional well in three dimensions (shown in Figure 7). The control volume is adjacent to the well. With the robust gridding method described above, the well is directly incorporated in the grid that is created. The flux between tetrahedral elements and the well can be calculated based on the pressure at the block center and wellbore, and the geometry of tetrahedral elements. The interpolation function for flux calculation between the element and the well is based on the steady state solution to the cylindrical well flow problem.
\[
q = \frac{\theta h k}{\ln\left(\frac{r_0}{r_w}\right)} \left(\frac{P - P_{bhp}}{B \mu}\right)^{k_r}
\]

In the above, \(q\) is the phase flow rate, \(r_w\) is the well radius, \(P\) is the block pressure and \(P_{bhp}\) is the bottom-hole pressure. The well model shown here does not require special algorithms. The advantage of this intuitive well model is that the unstructured grid used for the simulation completely conforms to the layout of the well. The intuitive well model requires no preprocessing of any kind and heterogeneity around wells can be fully accounted for.

The well model can also be used for representing multilateral and “fish-bone” type wells. In these representations, the gravity head due to different phases needs to be accounted for accurately and special algorithms have been developed to consider these effects.

**Fracture Characterization and Mapping**

This task is concerned with generating faults and fractures using all the geologic data, simulating the resulting domain (or multiple realizations) quickly and verifying and adjusting fracture networks using all the available production data.

Several methods of generating fracture networks were tested. The properties of fracture affected rock can vary widely. Depending on the way in which fractures are formed, they may be open conduits or deformation bands which restrict flow. One conceptual diagram with possible variations in grid block permeabilities due to the presence of fractures is shown in Figure 8.
Figure 8: Conceptual diagram showing different types of fractures or deformation bands contributing to the permeability of the host rock.

The type of porosity and permeability that can be generated quantitatively in these settings was calculated and is shown in Figure 9.

Figure 9: The figure shows enhancements and reductions in porosities and permeabilities from 50 md and 500 md matrix rocks.
Another approach integrates fieldwork with deterministic and stochastic modeling to gain insight into how three-dimensional permeability structure of a fault develops through time. Using general conclusions drawn from the physical model, a three-dimensional stochastic model for the fracture structure can be developed. A tool that can construct stochastic fracture features is Fracman. A fracture network created using this tool (supplied by Golder Inc.) is shown in Figure 10. A grid is being created for this domain and simulations with in the discrete-fracture model and in the upscaled mode are being performed.

![Fracture Network](image)

**Figure 10: Figure showing fracture networks generated using a stochastic approach**

Once the fracture networks (like the one shown in Figure 10) are created, it is important to create grids for these complex three-dimensional fractured systems. Several different tools were examined to accomplish this goal. These included CUBIT (developed at the Sandia National
Laboratory), LaGriT (developed at the Los Alamos National Laboratory) and Tetgen (developed at the Weierstrass Institute for Applied Analysis and Stochastics, Germany). CUBIT has been found appropriate to grid curved features also. A gridded complex domain with curved fractures is shown in Figure 11.

![Figure 11: A domain with complex curved features gridded using CUBIT](image)

Three-phase simulations were performed on this domain, results of which are shown in Figure 12.

![Figure 12: Gas saturations after primary production and water saturations after water injection in a complex system with curved features](image)
**Parallel Computing Schemes**

The emphasis was on parallel codes for highly heterogeneous systems, where fractures/faults and wells intersect and form complicated domains. The most important consideration in performing computations in parallel (on multiple processors) is robust domain decomposition. ParMetis and Zoltan were evaluated for domain decomposition of unstructured systems. ParMetis is a general domain decomposition tool developed at the University of Minnesota. Zoltan is a dynamic parallel data migration program developed at the Sandia National Laboratory. Scalability of a 250,000 element system on 16 processors was shown using this approach. A similar approach is being used for decomposition of fractured/faulted systems.

Effectiveness of parallel computation is determined by the quality of domain decomposition. A nonconventional well in a regular finite-difference grid is shown in Figure 13. If the domain is decomposed into the four regions shown in the figure, blocks that the well passes through (which happen to be in domain 3) must be assigned as ghost blocks to domain 2, since communication between these blocks is required. This dramatically increases the communication time between the processors and reduces the parallel computing effectiveness.

![Figure 13: Difficulty of domain decomposition with regular grids and nonconventional wells](image)
To overcome this difficulty, well location and connectivity of the well with the surrounding elements are automatically taken into account in the approach developed at the University of Utah. The concept is shown in Figure 14.

Figure 14: Domain decomposition concept for unstructured grids with nonconventional well

In Figure 14, a two-dimensional domain with an unconventional well is shown. Connectivity of the well to the neighboring elements is identified and specified. Based on this ParMetis performs the domain decomposition of the type shown in Figure 13.

**Common Component Architecture for Reservoir Simulation**

The basic idea in this task was to create standardized modules that can be exchanged between different simulators and applications. A number of steps to achieve this objective have been completed. Decoupling of the physical model and the discretization method has been demonstrated. Most input files now use the XML format and do not need special parsers. All components are modular and use other well-structured modules such as PeTSc. Parallel tasks have been built into the simulators and use ParMetis, the domain-decomposition module.
The Common Component Architecture (CCA) developed at the Center for Component Technology for Terascale Simulation Software (CCTTSS) is an effort at the creation of a component-based software development model suitable for the needs of high-performance scientific simulation. CCA defines specifications for high-performance scientific components, and frameworks and is designed to glue components together through network. CCA also promotes the development of domain-specific standard interfaces, for example, a standard interface for reservoir simulation. CCA is the answer for creating standards in reservoir simulation.

CCA components will be created from the individual modules using simple header commands described in the CCA protocol. Modules are being created to be CCA compliant.

**Summary**

All reservoirs are fractured to a certain degree. The main goal of this project was to overcome difficulties associated with the representation and simulation of complex fractured/faulted systems with complicated wells. Significant advances were made in achieving this goal with the development of modular simulators that are able to access the most modern computational algorithms. A generic well model to represent complex wells was developed and parallel computing scheme to decompose unstructured-grids with complex wells was devised. A methodology to create three-dimensional unstructured grids of fractured/faulted domains was identified. First steps in the development of compositional and thermal finite-element models were taken. This will lead to the capability of simulating different physical processes (for example, primary production and water flooding followed by CO2 or steam flooding). A number of technology transfer activities were carried out to disseminate the ongoing research
efforts. A close integration of the efforts of geoscientists and engineers made it possible to create simulations of geologically realistic domains.

This project will create tools to better manage fractured/faulted reservoirs and reservoirs with complex wells. Better management of these reservoirs will lead to significant improvement in oil and gas production from these reservoirs.

Milestone and Deliverable Schedule

The milestone that was scheduled to be met at the end of first year of the project was complete decoupling between the physical models and discretization methods. This was accomplished. All the tasks are progressing per schedule.

Briefings, technical presentations

1. A Technical Presentation about the benefits of the project was made to the members of the Energy and Geosciences Institute at the University of Utah in September 2004.

2. A Technical Presentation of the results of some of the models was made at a meeting with BHP, Occidental and El Paso, Houston, October 2004.

3. A Paper on three-dimensional, three-phase finite-element simulation was presented in January/February 2005 at the International Reservoir Simulation Symposium in Houston, Texas.

4. One Day Symposium on Fractured Reservoir Characterization and Simulation was organized on March 1, 2005 in Salt Lake City Utah. The Program and presentations are available upon request.

5. Two Presentations at the Geologic Society of America Meeting in Salt Lake City, October 2005

Budget

The budget for budget periods 2 and 3 is shown below.
### Budget for Years 2 and 3

<table>
<thead>
<tr>
<th></th>
<th>Year 2</th>
<th>Year 3</th>
<th>Totals for Year 2 and 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graduate Students (3)</td>
<td>$ 66,000</td>
<td>$ 72,000</td>
<td>$ 138,000</td>
</tr>
<tr>
<td>Undergraduate Students</td>
<td>$ 5,000</td>
<td>$ 5,000</td>
<td>$ 10,000</td>
</tr>
<tr>
<td>Benefits (9%)</td>
<td>$ 6,390</td>
<td>$ 6,930</td>
<td>$ 13,320</td>
</tr>
<tr>
<td>Post-doc</td>
<td>$ 42,000</td>
<td>$ 45,000</td>
<td>$ 87,000</td>
</tr>
<tr>
<td>Post-doc Benefits (33%)</td>
<td>$ 13,860</td>
<td>$ 14,850</td>
<td>$ 28,710</td>
</tr>
<tr>
<td>Faculty Salaries</td>
<td>$ 42,075</td>
<td>$ 43,350</td>
<td>$ 85,425</td>
</tr>
<tr>
<td>M. Deo, Schamel, Forster</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Faculty Benefits (33%)</td>
<td>$ 13,885</td>
<td>$ 14,306</td>
<td>$ 28,191</td>
</tr>
<tr>
<td>Consultant (Dr. Jim Evans, USU)</td>
<td>$ 10,300</td>
<td>$ 10,600</td>
<td>$ 20,900</td>
</tr>
<tr>
<td>Total Benefits</td>
<td>$ 34,135</td>
<td>$ 36,086</td>
<td>$ 70,221</td>
</tr>
<tr>
<td>Supplies</td>
<td>$ 5,000</td>
<td>$ 5,025</td>
<td>$ 10,025</td>
</tr>
<tr>
<td>Travel</td>
<td>$ 6,000</td>
<td>$ 7,500</td>
<td>$ 13,500</td>
</tr>
<tr>
<td><strong>Total Direct Costs</strong></td>
<td><strong>$ 210,510</strong></td>
<td><strong>$ 224,561</strong></td>
<td><strong>$ 435,071</strong></td>
</tr>
<tr>
<td>Indirect Costs</td>
<td>$ 104,202</td>
<td>$ 111,157</td>
<td>$ 215,359</td>
</tr>
<tr>
<td>(49.5% of all direct costs except equipment)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>U of U Cost Share</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Graduate Student Salaries</td>
<td>$ 30,000 $ 32,050</td>
<td>$ 62,050</td>
<td></td>
</tr>
<tr>
<td>Benefits</td>
<td>$ 2,700</td>
<td>$ 2,885</td>
<td>$ 5,585</td>
</tr>
<tr>
<td>Indirect Costs</td>
<td>$ 16,186</td>
<td>$ 17,293</td>
<td>$ 33,479</td>
</tr>
<tr>
<td>Total Cost Share</td>
<td>$ 48,886</td>
<td>$ 52,228</td>
<td>$ 101,114</td>
</tr>
<tr>
<td><strong>Total Costs</strong></td>
<td><strong>$ 314,712</strong></td>
<td><strong>$ 335,718</strong></td>
<td><strong>$ 650,430</strong></td>
</tr>
<tr>
<td><strong>DOE Share</strong></td>
<td>$ 265,826</td>
<td>$ 283,490</td>
<td>$ 549,316</td>
</tr>
</tbody>
</table>