Chapter 17
Reservoir Simulation


17.1 Introduction—K.H. Coats
The Merriam-Webster Dictionary defines simulate as assuming the appearance of without the reality. Simulation of petroleum reservoir performance refers to the construction and operation of a model whose behavior assumes the appearance of actual reservoir behavior. The model itself is either physical (for example, a laboratory sandpack) or mathematical. A mathematical model is a set of equations that, subject to certain assumptions, describes the physical processes active in the reservoir. Although the model itself obviously lacks the reality of the reservoir, the behavior of a valid model simulates—assumes the appearance of—the actual reservoir.

The purpose of simulation is estimation of field performance (e.g., oil recovery) under one or more producing schemes. Whereas the field can be produced only once, at considerable expense, a model can be produced or run many times at low expense over a short period of time. Observation of model results that represent different producing conditions aids selection of an optimal set of producing conditions for the reservoir.

The tools of reservoir simulation range from the intuition and judgment of the engineer to complex mathematical models requiring use of digital computers. The question is not whether to simulate, but rather which tool or method to use. This chapter concerns the numerical mathematical model requiring a digital computer. The Reservoir Simulation chapter in the 1987 edition of the Petroleum Engineering Handbook included a general description of reservoir simulation models, a discussion related to how and why they are used, choice of different types of models for different-reservoir problems, and reliability of simulation results in the face of model assumptions and uncertainty in reservoir-fluid and rock-description parameters. That material is largely omitted here. Instead, this chapter attempts to summarize current practices and trends related to development and application of reservoir simulation models.
Models have been referred to by type, such as black-oil, compositional, thermal, generalized, or IMPES, Implicit, Sequential, Adaptive Implicit, or single-porosity, dual-porosity, and more. These types provide a confusing basis for discussing models; some refer to the application (e.g., thermal), others to the model formulation (e.g., implicit), and yet others to an attribute of the reservoir formation (e.g., dual-porosity). The historical trend, though irregular, has been and is toward the generalized model, which incorporates all the previously mentioned types and more. The generalized model, which represents most models in use and under development today, will be discussed in this chapter. Current model capabilities, recent developments, and trends will then be discussed in relation to this generalized model.

The 10 SPE Comparative Solution Project problems, SPE1 through SPE10, are used for some examples below. Table 17.1 gives a brief description of those problems.
where $\Delta P$ is $P^{n+1} - P^n$ and the $N \times N$ matrix $A$ represents the Jacobian $\partial F/\partial P$. The element $A_{ij}$ of $A$ is itself an $n \times n$ matrix $\partial F/\partial P_{ij}$, with scalar elements $a_{ij} = \partial f_i/\partial P_{ij}$, and $s$ each $= 1,2,...,n$. Eq. 17.3 is solved by the model’s linear solver. The matrix $A$ is very sparse because $A_{ij}$ is 0 unless block $j$ is a neighbor of block $i$.

The calculations for a timestep consist of a number of Newton (nonlinear or outer) iterations terminated by satisfaction of specified convergence criteria. Each Newton iteration requires:

(a) Linearization of the constraint equations and conservation Eq. 17.1.
(b) Linear algebra to generate the $A$ matrix coefficients.
(c) Iterative solution of Eq. 17.3 (inner or linear iterations).
(d) Use of the new iterate $P^{n+1}$ to obtain from Eq. 17.1 the moles of each component in the gridblock.
(e) A flash to give phase compositions, densities, and saturations which allow generation of the $A$ matrix coefficients for the next Newton iteration.

17.1.2 Model Formulations. A major portion of the model’s total CPU time is often spent in the linear solver solution of Eq. 17.3. This CPU time in turn reflects the many multiply operations required. The model formulation has a large effect on the nature and expense of those multiplies.

**Implicit vs. Explicit.** The interblock flow term in Eq. 17.1,

$$q_{ij} = T_{ij} \sum_{j=1}^{J} \delta \rho_j s_{ij} \Delta \rho - \gamma_j \Delta Z \quad (17.4)$$

uses phase mobilities, densities, and mol fractions evaluated at the upstream blocks. A gridblock is implicit in, say, the variable $S_i$ if the new time level value $S_i^{n+1}$ is used to evaluate interblock flow terms dependent upon it. The block is explicit in $S_i$ if the old time level value $S_i^n$ is used.

**The Implicit Formulation.** The implicit formulation expresses interblock flow terms using implicit (new time level) values of all variables in all gridblocks. As a consequence, all nonzero $A_{ij}$ elements of the $A$ matrix of Eq. 17.3 are full $n \times n$ matrices. The resulting multiplies in the linear solver are then either matrix-matrix or matrix-vector multiplies, requiring work (number of scalar multiplies) of order $n^3$.

**The IMPES Formulation.** Early papers presented the basis of the IMPES (implicit pressure, explicit saturations) formulation for the black-oil case: take all variables in the interblock flow terms explicit, except for pressure, and eliminate all nonpressure variables from the linearizations required. The model formulation has a large effect on the nature and expense of those multiplies from a similar equation where the $A$ matrix elements are 2 x 2 matrices.

**The IMPES Type Formulation.** The IMPES type formulation expresses interblock flow terms explicitly, except for pressure, and eliminates all nonpressure variables from the linearizations required. For each Newton iteration, the model CPU time per gridblock per timestep was 67 (9.7) times larger for Implicit than for IMPES. The percentage of total CPU time spent in the IMPES pressure Eq. 17.7 is much less for the IMPES formulation than for the Implicit formulation.

**The IMPES Type Formulation.** The stability of the IMPES formulation for the two-phase water/oil case was improved by following the IMPES pressure equation solution with solution of a water saturation equation using implicit saturations (mobilities). This concept was extended to the three-phase case and called the sequential formulation. For each Newton iteration, the method requires solution of the IMPES pressure Eq. 17.7, followed by solution for two saturations from a similar equation where the $A_{ij}$ elements of the $A$ matrix are 2 x 2 matrices.

**The Adaptive Implicit Formulation.** The Adaptive Implicit Method (AIM) uses different levels of implicitness in different blocks. In each gridblock, each of the $n$ variables may be chosen explicit or implicit, independent of the choices in other gridblocks. The choices may change from one timestep to the next. This results in the same equation $\Delta P = -F$ as the Implicit formulation except that the elements $A_{ij}$ of the $A$ matrix are rectangular matrices of variable size. The numbers of rows and columns in $A_{ij}$ equal the numbers of implicit variables in blocks $i$ and $j$, respectively; all $A_{ij}$ are square matrices.

**The Adaptive Implicit Method (AIM) Formulation.** The Adaptive Implicit Method (AIM) uses different levels of implicitness in different blocks. In each gridblock, each of the $n$ variables may be chosen explicit or implicit, independent of the choices in other gridblocks. The choices may change from one timestep to the next. This results in the same equation $\Delta P = -F$ as the Implicit formulation except that the elements $A_{ij}$ of the $A$ matrix are rectangular matrices of variable size. The numbers of rows and columns in $A_{ij}$ are square matrices. The CPU expense per Newton iteration of an AIM model lies between that of IMPES and Implicit models, tending toward the former as more blocks are taken implicit in pressure only.

**Choice of Formulation.** For a given problem, the previous four formulations generally give widely different CPU times. Generalizations regarding the best formulation have many exceptions. Arguably, the trend is or should be toward sole use of the AIM formulation. This is discussed in the Stable Step and Switching Criteria sections to follow. Current simulation studies use all of these formulations. The Implicit formulation is generally faster than IMPES for single-well coning studies, and for thermal and naturally fractured reservoir problems. For other problems, IMPES is generally faster than Implicit for moderate or large $n$ (say, $n > 4$). Most participants used IMPES for SPE Comparative Solution Project problems SPE1, SPE3, SPE5, and SPE10. All participants used the Implicit formulation for SPE2, SPE4, SPE6, and SPE9. No participants in SPE1 through SPE10 used a Sequential model, and, with few exceptions, none used AIM.

A frequently stated generalization is that numerical dispersion error is significantly larger for Implicit than for IMPES formulations. Truncation error analysis shows this error to be proportional to $\Delta x \times \Delta t$ for Implicit and $\Delta x \times \Delta t$ for IMPES. Real problem nonlinearities and heterogeneity render the analysis approximate and the generalization of limited merit. For example, Figs. 17.1 through 17.3 show virtually identical Implicit and IMPES results for the blackoil 9,000-block SPE9 and 2,000-block gas/oil SPE10 problems. For SPE9 (SPE10), the average timestep was 67 (9.7) times larger for Implicit than for IMPES. The percentage of total CPU
tion requirement and CPU time should be similar if "equivalent" mass and volume error tolerances are used as convergence criteria.

**Variable Choice.** The linear algebra required to reduce the gridblock's $n$ conservation equations to the IMPES pressure equation is influenced by the choice of variables. The influence is absent for black oil, moderate for "moderate" $n$ and up to a factor of three for large $n$ (say, > 15). The choices of $p$ and mol fractions $z_i$ or mol numbers$^{23,35}$ are better than the choice of $p$, saturations, and phase mol fractions$^{23}$ for large $n$. The effect of this variable choice on total CPU time is often small because the affected work is often a small part of total CPU time. This IMPES reduction is absent in the Implicit formulation and the last of the above variable choices is arguably preferable.$^{22}$

**Adaptive Flash Calculations.$^{13}$** The work of EOS flash calculations, including the generation of fugacities and their derivatives, can significantly affect model efficiency when the linear solver does not dominate total CPU time. There may be little need to perform (most of) that work in a gridblock when $p$ and composition are changing slowly. Use of internal, intelligent criteria dictated when that work is needed can significantly reduce the total-run flash calculation CPU time.$^{11}$ This is similar in principle to the AIM selection of explicit variables for gridblocks which are quiescent in respect to throughput ratio.

17.1.4 Stable Timestep and Switching Criteria. This topic relates to the observation that lower run turnaround time can increase benefits from a reservoir study allotted a budgeted time period. As a corollary, time spent in repeated runs fighting model instabilities or time-stepping is counterproductive. While many factors affect this run time, it always equals the product of the CPU time/step ($n$) (number of timesteps). The first factor is large and the second "small" for the Implicit formulation, and conversely for the IMPES formulation. IMPES is a conditionally stable formulation requiring that $\Delta t < \Delta t^*$ to prevent oscillations and error growth, where $\Delta t^*$ is maximum stable timestep. The conditional stability stems from the explicit treatment of non-pressure variables in the interblock flow terms. Mathematicians performed stability analyses for constant-coefficient difference equations bearing some resemblance to IMPES. Authors in our industry extended and applied their results to derive expressions for $\Delta t^*$, in particular,$^{23,37}$$^{49}$

$$\Delta t^* = \frac{V}{\sum_i \left( \frac{q_i}{\mu_i} + \frac{q_i}{\mu_j} \right) + 2P_{\text{dual}} (T_i + T_j + T_k)}$$  (17.8)

for the black-oil 3D case of gas/oil flow. This shows that stable step $\Delta t^*$ is dependent upon flow rates, phase mobility, and capillary pressure derivatives, which of course vary with time and from one gridblock to another. Thus, at a given timestep, there are block-dependent stable step values $\Delta t_i^*$, where $1 < i < N$ and the IMPES stable step is Min$(i) \Delta t_i^*$. An IMPES model using this internally determined stable step will run stably but may suffer from the weakest-link principle. As an extreme example, consider a 500,000-gridblock problem where, over a 100-day period, the $\Delta t_i^*$ value is 0.01 day for one block and > 30 days for all other blocks. The IMPES model will require 10,000 timesteps over the 100-day period.

In the AIM formulation, the stable step $\Delta t_i^*$ depends upon the number and identities of variables chosen explicit in block $i$. Theoretically, $\Delta t_i^* = \epsilon i$ if all block $i$ variables are chosen implicit. In the previous example, all non-pressure variables could be chosen implicit in the block where $\Delta t_i^* = 0.01$ and explicit in all other blocks. The AIM model would then require CPU time/step essentially no greater than the IMPES model but would require only three timesteps for the 100-day period.

Numerous papers$^{23,37}$ address the problem of determining expressions for the $\Delta t_i^*$ for use internally as switching criteria to select block variables as explicit or implicit in the AIM model. The stability analyses involved are complex and may be impractically complex when allowing the implicit vs. explicit variable choice to include all permutations (in number and identity) of the $n$ variables. The most reliable and efficient AIM models in the future will stem from continuing research leading to the following: (a) $\Delta t_i^*$ estimates which are "accurate," and (b) implicit vs. explicit variable choices, block by block, which are near-optimal$^{24}$ and minimize total CPU time, (CPU time/step) ($n$) (number of steps).

17.1.5 The Linear Solver. Preconditioned Orthomin$^{38}$ is the most widely used method for iterative solution of Eqs. 17.3 or 17.7. Nested Factorization (NF)$^{39}$ and incomplete LU factorization (ILU(n))$^{40}$ are the two most widely used preconditioners. The term "LU factorization" refers to the factoring of the matrix $A$ into the product of a lower triangular matrix $L$ and an upper triangular matrix $U$. That is an expensive operation but is straightforward, involving only Gaussian elimination. The term "ILU(n)" denotes incomplete LU factorization, where only limited fill-in is allowed and $n$ is the "order of fill."$^{41}$ NF performs exceptionally well when transmissibilities associated with a particular direction (in a structured grid) dominate those in other directions uniformly throughout the grid. In general, ILU(n) or red-black ILU(n)$^{42}$ (RBILU(n)) is less sensitive than NF to ordering of the blocks and spatial variation of the direction of dominant transmissibilities. In addition, RBILU(n) or ILU(n) have the parameter $n$ (order of allowed infill) which can be increased as needed to solve problems of any difficulty.

A literature search and discussions with numerous developers and users have failed to establish consensus on whether NF or ILU preconditioning is better. Some are strong advocates of one method and others are just as adamantly supportive of the other. But many find, like this writer, that the better method is problem-dependent and it is difficult to find a reliable $p$ indicator for making an up-front choice. In the writer's experience, (a) when NF works well, it is faster than ILU methods, (b) RBILU(0) with no residual constraint is frequently the best of the ILU variants and a good default choice, and (c) in some cases, global residual constraint with the ILU or RBILU method is beneficial.

17.1.6 Cartesian Grids and Reservoir Definition. For many years, simulation used orthogonal Cartesian grids. In the past 15 years, numerous papers have described local grid refinement and various non-Cartesian grids, as discussed in the Gridding section. These papers show that non-Cartesian grids can reduce grid-orientation effects and provide definition and accuracy near wells, faults, highly heterogeneous areas, and so on more efficiently than Cartesian grids. The premise that Cartesian grids cannot provide required accuracy efficiently in these respects has come to be accepted as a fact. In addition, advances in geophysics have led to geostatistical description of permeability and porosity on a fine scale once unimaginable. Increasingly, our papers include examples using thousands of gridblocks for two- or few-well "patterns," in part to reflect these geostatistical descriptions. The purpose of this section is to show, using a few examples, that Cartesian grids can provide adequate accuracy and reservoir and near-well definition efficiently in some cases, even without local grid refinement. No generalizations from the examples used are intended. For the most part, the examples are taken from the literature.

SPE7 is an $x$-direction horizontal well problem with a $9 \times 9 \times 6$ Cartesian grid representing a $2,700 \times 2,700 \times 160$-ft reservoir section. The specified block $\Delta x$ values decrease from 620 to 60 ft at the well, presumably to increase near-well definition and accuracy of results. The $\Delta x$ are uniformly 300 ft. Fig. 17.4 compares Case 1A results for the SPE7 grid with results using uniform areal spacing $\Delta x = \Delta y = 300$ ft. The near-well $y$-direction refinement of the specified grid has no effect and is not necessary in this problem.

SPE8 is a gas/oil problem with one gas injector and two producers on the corners of a $5,000 \times 5,000 \times 325$-ft square reservoir. A $10 \times 10 \times 4$ Cartesian grid with uniform $\Delta x = \Delta y = 500$ ft is specified. Five participants compared their results for that grid with results from their
(areally) locally refined or unstructured grids. They showed good agreement for grids having approximately four times fewer blocks than the $10 \times 10 \times 4$ grid. Fig. 17.5 shows equally good agreement for a $5 \times 5 \times 4$ ($\Delta x = \Delta y = 1,000$ ft) Cartesian grid with no local refinement.

### Example 17.1
Table 17.2 gives data for Example 17.1, a ¼ five-spot, vertical-well problem. Fig. 17.7 shows two block-centered grids (a) and (b) used for this type of problem. The four-fold smaller well blocks of grid (b) provide finer well definition and presumably increase the accuracy of results. Fig. 17.8 shows the identical results for $10 \times 10$ grid (a) and $11 \times 11$ grid (b). Results are nearly identical for the $5 \times 5$ grid (a) and $6 \times 6$ grid (b), and Fig. 17.9 shows insignificant difference between $3 \times 3$ grid (a) and $4 \times 4$ grid (b) results. The grid (b) doubles the grid (a) IMPES run CPU time but contributes no greater accuracy. Well-index ef-

SPE10 (Model 1) is a 2D cross-sectional gas/oil problem with a geostatistical permeability distribution given on a $100 \times 1 \times 20$ Cartesian fine grid. Coarse-grid submittals included results using upscaling and local grid refinement. A homogeneous $5 \times 1 \times 5$ Cartesian grid with no alteration of relative permeability matched the $100 \times 1 \times 20$ results nearly exactly. SPE10 (Model 2) is a 3D water/oil problem with a 1.122 million-cell geostatistical grid. Some coarse grid submittals included sophisticated upscaling and gridding techniques with no pseudozonation of relative permeability and grids from 4,810 to 70,224 blocks. Others used simple flow-based upscaling to 75- to 2,000-block Cartesian grids with moderate pseudoization of relative permeability and grids from 4,810 to 70,224 blocks. Others used simple flow-based upscaling to 75- to 2,000-block Cartesian grids with moderate changes. In general, the latter submittals showed the best agreement with the fine-grid solution.

Numerous papers show that non-Cartesian grids can significantly reduce the grid-orientation effects of Cartesian grids. However, most of the examples used to study those effects are highly adverse mobility ratio displacements in homogeneous, horizontal reservoirs. In reservoirs with more normal fluid mobilities, areal fluid movement is more strongly affected by heterogeneity and/or gravity forces associated with reservoir structure (variable dip), and grid-orientation effects tend toward a second-order effect. As an example, the SPE10 (Model 2) water/oil problem reservoir is highly heterogeneous. Fig. 17.6 compares five-point and nine-point field results for an upscaled $28 \times 55 \times 85$ Cartesian grid. The close agreement indicates an absence of grid-orientation effects even though the unfavorable oil/water viscosity ratio is 10 and there is no dip.

---

**Fig. 17.4**—Effect of near-well grid refinement, SPE7 Case 1A.

**Fig. 17.5**—Effects of coarser grids on producing GOR.

**Fig. 17.6**—9-point vs. 5-point, SPE10 Model 2.
**TABLE 17.2—EXAMPLE 1 DATA**

40-acre five-spot with no dip and 180-ft pay thickness

Incompressible oil and water properties:

\[ P_o = 43 \text{ lbm/ft}^3 \quad \mu_o = 3 \text{ cp} \quad R_o = 0 \]

\[ P_w = 64 \text{ lbm/ft}^3 \quad \mu_w = 0.3 \text{ cp} \]

Grid: \( N_x = N_y = 15 \), \( \Delta x = \Delta y = 15 \text{ ft} \)

\[ v = 0.2 \quad k_r = 1000 \text{ md} \quad k_i = 100 \text{ md} \]

\[ S_{ow} = S_{orw} = 0.2 \quad k_{orw} = 1.0 \quad k_{cor} = 1.0 \]

\[ k_r = k_m [(S_{orw})/(1.0 - S_{orw})] \quad k_i = k_m [(1.0 - S_{orw})/(0.0 - S_{orw})] \]

Water injection well at \( x = y = 0 \) is completed in all 15 layers and injects 2,550.7 STB/D water.

Production well at \( x = y = 660 \text{ ft} \) is completed in top 5 layers (upper \( 1/3 \) of pay) and flows on pressure constraint against a bottomhole wellbore pressure of 4,000 psia.

Discussion of runs for this problem refers to various grids by their \( N_x \times N_y \) dimensions because \( N_x = 15 \) in all cases.

---

**Fig. 17.7**—Block-centered 3 \( \times \) 3 and 4 \( \times \) 4 Cartesian grids used in Example 17.1.

Fig. 17.10 shows a small effect of grid refinement on Example 17.1 results for grids from 20 \( \times \) 20 to 3 \( \times \) 3. The results indicate little need to enhance near-well definition by unstructured grids or by grid refinement (global or local) for grids finer than 3 \( \times \) 3 for this problem.

**Example 17.2.** Flexible non-Cartesian grids are shown to significantly reduce the required number of grids.\(^29\) An example provided\(^28\) was water/oil coning in a horizontal well in a 600 \( \times \) 300 \( \times \) 230 m homogeneous reservoir. Results were: (a) a 25,823-block 31 \( \times \) 17 \( \times \) 49 Cartesian grid was required to obtain a converged solution, and (b) a 3D 2,066-block CVFE unstructured grid gave correct results. Table 17.3 gives data for Example 17.2, a similar problem. Fig. 17.11 compares Example 17.2 results for 60 \( \times \) 31 \( \times \) 48 and 10 \( \times \) 7 \( \times \) 9 Cartesian grids. The 630-block coarse Cartesian grid results here agree as well with the Cartesian 60 \( \times \) 31 \( \times \) 48 fine-grid results as the reported 2,066-block CVFE results agree with the 31 \( \times \) 17 \( \times \) 49 Cartesian fine-grid results.

Non-Cartesian grids are argued to define irregular reservoir boundaries more efficiently than Cartesian grids. This is not necessarily true. For over 30 years, many models have used active-block coding. While the Cartesian grid extends past boundaries to numerous inactive blocks, those inactive blocks are dropped by the model and require no computer storage or CPU. These numerous inactive blocks pose a problem only for models, if any, that do not use active-block coding.

---

**17.2 Linear Solver—John Wallis and J.W. Watts**

The linear equation solver is an important component in a reservoir simulator. It is used in the Newton step to solve the discretized nonlinear partial differential equations. These equations describe mass balances on the individual components treated in the model. For nonisothermal problems, an energy balance is added to the system. The matrix problem involves solving \( Ax = b \), where \( A \) is typically a large sparse matrix, \( b \) is the right-side vector, and \( x \) is the vector of unknowns. In the IMPES formulation, there is a single unknown per cell pressure. In the fully implicit formulation, there is a fixed number \( n \) of unknowns per cell where \( n > 2 \). In the adaptive implicit formulation, there is a variable number of unknowns per cell. In most formulations, pressure is an unknown for each cell. The matrix \( A \) typically has associated well constraint equations and well variables and may be partitioned in block 2 \( \times \) 2 form as...
As a rule, the robustness of the iterative solution is on the implicitly defined matrix system

\[
\begin{bmatrix}
A_{ww} & A_{wr} \\
A_{rw} & A_{rr}
\end{bmatrix}
\begin{bmatrix}
x_w \\
x_r
\end{bmatrix}
= 
\begin{bmatrix}
h_w \\
h_r
\end{bmatrix}
\]

where \(x_w\) is the well variable-solution vector and \(x_r\) is the reservoir variable-solution vector. The matrix \(A_{ww}\) is often diagonal. In this case, the well variables may be directly eliminated, and the iterative solution is on the implicitly defined matrix system

\[
(A_{RR} - A_{Rw} A_{ww}^{-1} A_{wr}) x_R = b_R - A_{Rw} A_{ww}^{-1} b_w
\]

The well variables are then obtained by back substitution as

\[
x_w = A_{ww}^{-1} (b_w - A_{wr} x_r)
\]

If \(\mathbf{A}\) is large, solution of the matrix equations is impractical using direct methods such as Gaussian elimination because of computer storage or CPU time requirements. Iterative solution based on projection onto Krylov subspaces is typically used. These Krylov subspaces are spaces spanned by vectors of the form \(p(\mathbf{A}) \mathbf{v}\), where \(p\) is a polynomial. Basically, these techniques approximate \(\mathbf{A}^{-1} \mathbf{b}\) by \(p(\mathbf{A}) \mathbf{b}\). The commonly used methods for constructing \(p(\mathbf{A}) \mathbf{b}\) are Orthomin\(^{10}\) and GMRES.\(^{11}\) Both methods minimize the residual norm over all vectors in span \([\mathbf{b}, \mathbf{A} \mathbf{b}, \mathbf{A}^2 \mathbf{b}, \ldots, \mathbf{A}^{m-1} \mathbf{b}]\) at iteration \(m\). They should yield identical results. From a practical standpoint, it does not matter which is used.

A technique known as preconditioning can improve both the efficiency (speed in a typical problem) and robustness (ability to solve a wide range of problems at least reasonably well) of Orthomin or GMRES. Preconditioning involves transforming the original matrix system into one with the same solution that is easier to solve. As a rule, the robustness of the iterative scheme is far more dependent on the preconditioning than on the specific Krylov subspace accelerator used. The preconditioner \(\mathbf{M}\) is a matrix that approximates \(\mathbf{A}\), and has the property that the linear systems of the form \(\mathbf{M} \mathbf{x} = \mathbf{b}\) are easily and inexpensively solved. For most linear solvers the following preconditioned system is solved:

\[
\mathbf{M}^{-1} \mathbf{A} \mathbf{x} = \mathbf{b}, \quad \text{where} \quad \mathbf{x} = \mathbf{M}^{-1} \mathbf{y}
\]
The preconditioned ORTHOMIN(k) algorithm, which retains the last \( k \) orthogonal direction vectors, is given by

1. Compute \( r_0 = b - A x_0 \) Set \( p_0 = M^{-1} r_0 \).
2. For \( j = 0, 1, 2, \ldots \), until convergence Do:
   a. \( x_j = x_j + \alpha J p_j \)
   b. \( r_j + 1 = r_j - \alpha \beta J p_j \)
   c. \( \beta j = \frac{(A M^{-1} r_j + 1, A p_j)}{(A p_j, A p_j)} \), for \( i = j - k, \ldots, j \)
   d. \( p_{j+1} = M^{-1} r_{j+1} + \sum_{i=j-k}^{j} \beta_i p_i \)
3. End Do

Because of the nature of the reservoir simulation equations, only certain preconditioners are effective in solving them. Reservoirs are typically shaped like pancakes, being much broader than they are thick. This geometry leads to strong vertical connectivity. Some preconditioners exploit this property. The most commonly used such preconditioner is NF. The convergence of NF is sensitive to the cell ordering. Best results are usually obtained by ordering the cells first along the direction of highest transmissibility and then successively along directions of decreasing transmissibility. This nearly always means that NF should be ordered first in the vertical direction.

The other commonly used preconditioners are incomplete lower triangular/upper triangular factorizations of the matrix, or ILU(\( n \)), where \( n \) is the level of infill that is retained during the elimination process. Performance of these can be improved by using a red-black checkerboard ordering (also called D4 ordering) of the nodes. Red-black ordering on a five-point (in 2D) or seven-point (in 3D) grid leads to direct elimination of the unknowns at the red cells, leaving a system containing only the unknowns at the black cells. The result is a halving of the number of unknowns. An ILU preconditioner using red-black ordering with zero infill on the reduced system is referred to as RBILU(0) and is the most frequently-used form of ILU.

In IMPES models with either no faults or vertical faults only, RBILU(0) or ILU(1) combined with z-line additive corrections typically converges very rapidly. NF can also be used effectively in cases involving vertical faults and pinchouts because the matrix retains the structure required by NF. Nonvertical faults disrupt the matrix structure that makes red-black orderings attractive. In models containing them, ILU is the method of choice. Nine-point discretizations also cause problems for red-black orderings, but cause no difficulty for NF.

NF and RBILU(0) are commonly used in implicit models. Another very effective approach exploits the fact that pressure is the “stiff” variable. The Combinative or CPR method is a two-step preconditioning that extracts a pressure equation from the implicit matrix. It iterative­ly solves for a pressure correction at each iteration, uses the pressure correction to form a new residual, applies an inexpensive implicit preconditioning such as diagonal scaling or line Gauss-Seidel to the new residual, and then uses the sum of the two steps as the approximate solution. In compositional models, this two-step method can be much faster than one-step methods.

Many models include well-constraint equations that add well pressures to the set of unknowns. A simple but effective way of dealing with these equations is to order them first in the global matrix and then eliminate the well pressures from the set of unknowns. In this approximate elimination, any infill terms are either column-summed or row-summed into the main diagonal of the reservoir matrix, which is then factored using NF. The reservoir matrix problem is then solved iteratively and the well variables are obtained by back substitution.

Parallel iterative solution typically uses a domain decomposition approach in which the grid is partitioned into domains that contain approximately the same number of cells. The partition­­ing should be done such that coupling is strongest between cells within a domain. As a result, domains normally have groups of columns of cells. One way to accelerate the iteration is to color the domains in red-black fashion and apply an NF-type procedure in which the outer level of nesting is the coupling between domains and either NF or an ILU variation is used to factor the individual domains.

The solution of matrices arising from unstructured grids typically involves some variant of ILU with cell orderings such as Reverse Cuthill-McKee (RCM) or Minimum Degree Fill (MDF).

17.3.1 Introduction.

The aim of gridding in reservoir simulation is to turn the geological model of the field into a discrete system on which the fluid-flow equations can be solved.

The basic structure of an oil reservoir is a set of geological horizons representing bedding planes. The reservoir may contain faults, at which the strata are displaced. It is usually possible to identify many more layers in the geological model than it is practical to include in reservoir flow simulation, so some upscaling of rock properties will normally be carried out. Even after this process, the geology to be represented is rarely homogeneous at the scale of the simulation grid.

Two related issues are involved in choosing a grid for reservoir simulation: the accuracy with which the geological description of the reservoir is matched, and the discretization of the flow equations. In a classical finite-difference scheme, the point values of pressures and saturations are used as solution variables, and the differential operators that appear in the fluid-flow equations may be expanded as difference expressions of these point values to some order. An alternative approach is to use an integral finite-difference or finite-volume method in which the fluid-flow equations are integrated over a set of cell volumes. This yields a set of equations in which the mass conservation conditions for the fluid in the simulation cell volumes are relat-
ed to the flows through the interfaces between those cell volumes. Rock properties such as porosity are assumed constant over the cell or controlled volume. This yields a discretization scheme which is conservative (each outflow from one cell is an inflow to another) and for which the fluid in place may be obtained straightforwardly. The mass conservation equations for a timestep from $T$ to $T + \Delta T$ then become:

$$
\begin{align*}
\rho \frac{\partial \text{V}}{\partial t} + \nabla \cdot (\text{V} \rho) &= 0, \\
\rho \frac{\partial \text{m}}{\partial t} + \nabla \cdot (\text{m} \rho) &= \sum_f \Delta T \cdot \sum_c \Delta \Phi_f \cdot \text{F}_{c_x},
\end{align*}
$$

where $\text{V}$, $\rho$ is the density of conserved component $c$ in cell $a$, $\text{m}_{a,c}$ is the mass of conserved component $c$ in cell $a$, $Q_{a,c}$ is the injection or production rate of component $c$ because of wells, and $\text{F}_{c_x}$ is the rate of component $c$ in phase $p$ from cell $a$ to its neighbor $b$. In general, the flows $\text{F}_{c_x}$ may involve the solution values of a number of cells, the number of cells involved defining the stencil of the numerical scheme. The linear pressure dependence of flows given by Darcy’s law leads to an expression of the type:

$$
\text{F}_{c_x} = \sum_c \Delta T \cdot \sum_f \Delta \Phi_{f_x} \cdot \text{F}_{c_x},
$$

where $\Delta \Phi_{f_x}$ is the potential difference of phase $p$ between cell $a$ and cell $x$, which includes pressure, gravity and capillary pressure contributions:

$$
\Delta \Phi_{f_x} = \frac{\partial \phi}{\partial p} \cdot P_x - \frac{\partial \phi}{\partial p} \cdot P_a + \frac{\partial \phi}{\partial \rho} \cdot \rho_x - \frac{\partial \phi}{\partial \rho} \cdot \rho_a + \frac{\partial \phi}{\partial \mu} \cdot \mu_x - \frac{\partial \phi}{\partial \mu} \cdot \mu_a.
$$

The constant coefficients of mobility and potential difference products, $\Delta \Phi_{f_x}$, are commonly termed the transmissibilities. 

When the flows between two cells $a$ and $b$ can be expressed as a function of the solution values in just those two cells, so that the summation over cells includes just $x = b$, the flow expression takes a two-point form. The flow expression then takes a simple form:

$$
\text{F}_{c_x} = \Delta T \cdot \sum_c \Delta \Phi_{f_x} \cdot \text{F}_{c_x}. 
$$

Other options for discretization are available, such as Galerkin finite elements and mixed finite-element. It is sometimes possible to cast a finite-element Galerkin discretization into the upstreamed transmissibility-based form.

17.3.2 Regular Cartesian Grids. A simple 3D grid is the regular Cartesian grid (Fig. 17.12). Cells in such a grid may be simply identified using their $(i,j,k)$ index values.

Each of the grid elements will be assigned a single permeability or porosity value. In this case, it is possible to obtain the transmissibility value as a harmonic average:

$$
\mathbf{T}_{ab} = 1 / (1/\mathbf{T}_{aa} + 1/\mathbf{T}_{bb}) \quad \mathbf{T}_{ab} = K_{ab} \cdot A_{ab} / (d_x^2/2), \quad \mathbf{T}_{bb} = K_{bb} \cdot A_{bb} / (d_y^2/2),
$$

where cell $b$ is the neighbor to cell $a$ in some direction and $K$ is the cell permeability in that direction. $A$ is the area of the cell orthogonal to the direction of flow, and $d$ the dimension of the cell in that direction. Such a two-point transmissibility assumes a permeability tensor with primary axes aligned along the grid axes.

Although regular grids are normally defined in normal Cartesian coordinates, it is also possible to use an $(r, \phi, z)$ radial system. The resulting grid is cylindrical and is important for the special case of near-well studies dominated by radial inflow. For a 3D system, regular grids yield seven-point schemes, in which the flow equations for a cell involve solution values for just the cell and its six neighbors. Not all the elements in the grid need represent active solution variables in the simulation. Some cells may be inactive, representing volumes of the reservoir with zero porosity. Such inactive cells are usually compressed out of reservoir simulation solution arrays prior to the memory and time-intensive flow solution stage, and enable reservoirs with irregular boundaries to be represented within extended simulation grids.

The horizons that delimit rock strata are generally not horizontal, but are dipped, curved, or faulted. Unless extremely fine, a true regular grid that is orthogonal in all three axes will be unable to assign rock properties accurately to cell volumes. Such a layer-cake structure can be used, but will generally misalign property values (Fig. 17.13) in which the orthogonal grid provides a rather poor match to the dipping strata represented by the shaded layers. However, it is possible that improving computer power will bring such rasterized grids to a level of refinement at which a sufficiently good representation may be obtained.

**Dip-Normal Geometry.** A simple variation of a regular grid, in which the regular grid is rotated to bring the layers of cells into alignment with the bedding planes. Such a description would only suit a reservoir with a single, constant angle of dip. As geological descriptions have improved, fewer and fewer model reservoirs are found to fit this simple pattern, and something more flexible is required.

**Block-Center Geometry.** A simple model in which transmissibility between blocks is calculated on the basis of linear interpolation between the center values of the cells. This is a simple way of representing variable dip, but is difficult to represent graphically in a consistent way. Pore volumes are calculated on the basis of a series of flat regular cells with variable depths...
Fig. 17.13—Orthogonal grid used to represent dip.

Fig. 17.14—(a) Obtaining pore volumes; (b) obtaining transmissibilities.

Thus, for the pair of cells illustrated,

\[ T = K \cdot \frac{A}{d}, \quad \text{where} \quad K = \frac{2}{d_1} + \frac{1}{d_2} \]

where \( A \) is the average area over which flow occurs and \( \theta \) is a dip correction given by \( \cos^2 \theta \), where \( \theta \) is the angle of dip of a line joining the cell centers to the horizontal. Such a block-center option is suitable for unfauluted reservoirs and is commonly supplied as a simulator option.

**17.3.3 Hexahedral Grids.** Further improvements in geological modeling threw an emphasis on describing faults, and made it important to distinguish depth displacements due to dip and faulting. This is difficult in block-centred geometry in which the cell is positioned by its centre depth and \( \Delta x, \Delta y, \Delta z \) dimensions. To define faulting more precisely it is useful to define the position of grid cell by its corner point locations. A hexahedral shape with eight corners and bilinear planes as surfaces then describes the cell geometry. Faults, both vertical and inclined, may be described precisely (Fig. 17.15). Such grids are often called corner-point grids.

In both the dipped and general hexahedral grids, the orthogonality of a completely rectangular grid no longer exists, and the result is that the two-point property of the flows between the cells is lost—the flow between cell a and cell b is not just a function of the solution values of cells a and b, but also of an additional coupling and use of a low-order transmissibility scheme. This is normally done for extra couplings introduced by dip angles, which are often small.

Although this corner-point description handles the fault issue, the basic coordinate system remains a regular grid (i.e., the grid is structured). Fitting such a basically regular system to the irregular shapes of a reservoir remains a difficulty that may be solved in two basic ways—either by distorting the grid and fitting the cells into the geometry, or by truncating the grid to the reservoir position.

**17.3.4 Multiple-Domain Hexahedral Grids.** In some cases, a single structured grid system cannot match the overall structure of a reservoir, so a block grid or domain-based grid is used. This consists of a number of subgrids, each with a local regular \((i,j,k)\) structure, but linked together to model the entire reservoir. The block hexahedral system gives rise to multiple \((i,j,k)\) indexing systems—\((i,j,k,l)\), where the \( l \) index specifies the grid system. These comprise a series of regular grids. Such regular gridding systems have advantages for upscaling and downscaling—for example, a natural coarsening of a regular grid may be simply defined by grouping sets of coordinates in each direction.

**17.3.5 Grid Refinement.** A common requirement in reservoir simulation is an increased level of detail around an item of interest such as a well. This is frequently obtained in structured grids by local grid refinement, replacing a set of cells in the original grid by a finer grid (Fig. 17.16). The inserted grid may be Cartesian (center) or radial (upper left). Local refinement may be regarded as a form of multiple domain structured grid, in that it consists of a number of linked structured grids. Flows at the edges of local refinements generally take a multipoint form.

**17.3.6 Unstructured Grids.** The problems involved in using a regular structured grid to represent reservoir geometry can be avoided by using an unstructured grid. This is constructed around a set of solution points that need have no particular indexing scheme. These points may
be triangulated into a mesh of triangles or tetrahedrons. A control volume is constructed around the nodes of the resulting mesh to define the simulator cell volumes. The perpendicular bisector (PEBI) method introduced into reservoir simulation by Heinemann used a technique also known as a Voronoi grid. Starting from any set of solution points, the PEBI cell volumes are defined by the perpendicular bisection planes between these points. The resulting control volume is defined by the perpendicular planes—it is the set of points closer to the node than any other. This is shown in Fig. 17.17, in which the bisectors to the heavy lines joining the solution points enclose the control volume, represented by the shaded area. The grid is locally orthogonal, and the desirable property of two point flows is obtained. The actual cell volumes may have a variety of shapes, depending on the exact placement of the solution points, but are typically hexagonal in two dimensions. Grid refinement occurs naturally in areas where solution points are closely spaced.

The two-point property is not naturally preserved in anisotropic reservoirs, although it can be regained by transforming to a K-orthogonal grid in which the geometry is transformed so that \( K_n \) is parallel to the vector joining the solution nodes, where \( K \) is the permeability tensor and \( n \) is the normal to the cell volume surface. For nonisotropic cases in which the grid is not K-orthogonal, the flows will be functions of the solution values in more than two cells as in the general hexahedral case.

An unstructured grid may be defined in two dimensions, and then applied to each layer of a reservoir model, so that a typical cell is a hexagonal prism. This is sometimes termed a 2½D unstructured grid. Alternatively, a full 3D unstructured grid may be defined. The 3D approach is most effective when applied to model a local structure such as a branched well.

Unstructured grids yield an elegant and flexible grid description. However, the ability to identify cells by a simple set of indices is lost, and items such as wells need to be positioned in true space terms. The systems of linear equations generated by unstructured grids are also commonly regarded as more difficult to solve than those produced by structured grids. However, it may be more true to say that optimal solution schemes are simpler to find for structured grids, where the row and plane order provides a natural ab initio solution variable-ordering scheme.

### 17.3.7 Truncated Regular Grids

The truncated approach fits in well with the rectangular grids used in geological modeling. A simple rectangular grid is always used in the areal direction, but faults may subdivide the rock volume in a given column. The areal grid is not modified to match the faults. Thus the two marked volumes in Fig. 17.18 represent different cells, but may have the same \( i, j \) indices, so this creates a multiple-domain grid. A disadvantage is the more complex shape of cells at the edge of the grid. Transmissibilities for such cells may need to be calculated numerically. Apart from the truncated cells, all the grid cells are hexahedra that are rectangular in plan.

### 17.3.8 Other Gridding Systems

#### Triangular or Tetrahedral Grids

The underlying solution points of a PEBI mesh can be linked together into a Delaunay triangulation. In 2D, this creates triangles, and in 3D it creates tetrahedra. One option would clearly be to use triangular or tetrahedral cells directly and associate cell volumes with these. This technique is rather rarely used in reservoir simulation. Partly this may be historical, but the Delaunay triangulation is rather less stable under grid changes than a Voronoi grid, and triangulation can more often lead to “sliver” cells with a high surface area but a small volume.

#### Curvilinear Grid Systems

In some special cases, a transformed coordinate system may be used, based around an expected flow pattern. Such grids are not well adapted to represent geological data, and have been used less frequently as more detailed reservoir descriptions have become available.

### 17.3.9 Future Directions

Two themes emerge from current trends in reservoir simulation gridding. The increasing sophistication of data preparation and solver technology indicates a move towards unstructured grids as a general method of solving the flow equations for a given reser-
voir simulation problem. On the other hand, reservoir simulation is increasingly seen as part of a decision-making process rather than an isolated activity, so the ability to map easily onto the generally regular data structures used in seismic and geological modeling becomes an important issue. In this role, structured grids may have advantages of simplicity and scalability.

An ideal is to separate the construction of the flow-simulation grid from the description of the reservoir geometry. This ties in with a further ideal, inherent in many discretization schemes, that the scale of the simulation grid should be below the scale of the problem structure.

For more complex shape-dominated problems, the unstructured approach looks general and flexible, providing that the data-handling and cell-identification methods can be moved to true $x,y,z$ space preprocessing software.

### 17.4 Upscaling of Grid Properties—Alan Grindheim

#### 17.4.1 Definition
Upscaling, or homogenization, is substituting a heterogeneous property region consisting of fine grid cells with an equivalent homogeneous region made up of a single coarse-grid cell with an effective property value. It is performed for each of the cells in the coarse grid and for each of the grid properties needed in the flow-simulation model. Therefore, the upscaling process is essentially an averaging procedure in which the static and dynamic characteristics of a fine-scale model are to be approximated by that of a coarse-scale model. A conceptual illustration of the upscaling process is shown in Fig. 17.19.

#### 17.4.2 Can Upscaling Be Avoided?
Typically, 3D geological models contain detailed descriptions of the reservoir that can be hard to capture properly with a significantly coarser model. Therefore, it would be preferable if upscaling could be avoided. Currently, an average-sized flow simulation model consists of approximately 100,000 active grid cells. This is to ensure that the CPU consumption of a simulation run will be reasonable (i.e., within practical limits).

1. Either volume or flux vice, depending on the type of property that is to be upscaled.

Because a typical 3D geological model may consist of approximately 10 million active grid cells, it is obviously infeasible to run fluid-flow simulations directly on the geological model. Hence, upscaling is a required part of current reservoir modeling workflows.

Seen through the eyes of the geologist, the upscaling task may be a painful experience because all the geological details that were put into the model seem to be lost in the process. For a reservoir engineer, on the other hand, effective properties might be all that matter.

For volumetric (additive) properties such as porosity and saturation, the effective flow-cell value is simply given by the bulk and pore volume weighted arithmetic average, respectively, of the geo cells inside it. For the permeability, which is intrinsic (nonadditive) by nature, no such simple averaging method exists. The complexity one needs to take into account when upscaling permeability is considerable; therefore, all current techniques provide only an approximation of the true effective cell permeability. This approximation may range from very good to very poor, depending on the complexity of the fine-scale permeability distribution as well as the upscaling method used.

#### 17.4.3 Upscaling Techniques for Absolute Permeability
Homogenization of absolute permeability does not have an exact analytical solution, except for in a few idealized cases. The challenge of computing an accurate effective permeability has resulted in a large number of upscaling techniques. These techniques range from simple statistical averages to advanced numerical methods.

Tensor methods are the most accurate techniques available for computing the effective cell permeability. These are based on solving a second-order elliptic differential equation describing single-phase, incompressible, steady-state fluid flow in a region without sources and sinks (i.e., wells). Some flow-based methods may provide a full permeability tensor. However, because most multiphase flow simulators can only handle a diagonal permeability tensor because of the use of a seven-point stencil in 3D, diagonal tensor methods are most frequently used whether directly or indirectly (through a diagonalization of a full tensor). For a diagonal tensor, only the effective permeability in the principal directions of flow ($x,y,$ and $z$) will be nonzero.
Local upscaling
Regional upscaling
Global upscaling

Fig. 17.20—Upscaling schemes and the size of the computational region (geo grid in white, flow grid in black).

The flow equation is usually discretized with a finite-difference scheme, although finite-element methods are also applied occasionally. To compute all the directional components of the permeability tensor, the discretization and solution of the flow equation must be performed for each of the principal flow directions (i.e., three separate single-phase simulations need to be performed). Each simulation involves the iterative solution of a linear equation system (typically, the linear solver is a conjugate gradient method, preconditioned by incomplete Cholesky or LU factorization). The unknowns in this equation system are the geo-cell pressures inside the flow cell, whereas known quantities are the geo-cell dimensions and permeabilities, as well as the pressure conditions along the faces of the flow cell. When the numerical solution of the fine-scale pressure distribution has converged, an effective permeability is computed by equating expressions for the flux through the heterogeneous geo cells with the flux through the equivalent homogeneous flow cell using some form of Darcy's law.

The pressure field is usually solved locally—that is, for one flow cell at a time. However, as discussed in the next subsection, the size of the computational region may not necessarily be limited to that of the upscaling region (i.e., the flow cell).

17.4.4 Upscaling Schemes for Absolute Permeability. Based on the size of the computational region, the single-phase upscaling process may either be described as local, regional, or global. With local upscaling techniques, the computational region is identical to the upscaling region (i.e., only geo cells inside the flow cell are considered in the upscaling computations). For regional upscaling, the computational region is expanded beyond that of the flow cell to include a buffer region of neighboring geo cells. In the case of global upscaling, the computational region is that of the entire geo model. Fig. 17.20 provides a schematic drawing of how the computational region varies with the different upscaling schemes. These are further discussed in the subsections that follow.

It should be noted that the different upscaling schemes are only relevant when considering flow-based (tensor) methods. It is also important to realize that even though the computational region may vary according to the scheme used, the upscaling region remains unchanged and is of course defined by the flow cell, as in the case of the simple, analytical upscaling techniques.

17.4.5 Local Upscaling. Because it used to be too time-consuming to compute the fine-scale pressure field for the complete geo grid in a single operation, the flow-based methods have traditionally been restricted to solving the pressure field locally—that is, for a single flow cell at a time. Hence, the effective cell permeability is computed separately and independently of the other flow cells, which may or may not be correct depending on how representative the imposed pressure conditions along the faces of the flow cell are.

Different types of artificial boundary conditions for the flow cell have been suggested over the years, all with the objective of providing as good an approximation of the real boundary conditions as possible. An important design criterion for the artificial boundary conditions is the conservation of flux in and out of the flow cell.

The first type of boundary conditions proposed for the local solution of the pressure equation was published by Warren and Price in 1961. Their approach is to impose a constant pressure gradient in a selected direction of flow by specifying a pressure of 1 on the inflow face and a pressure of 0 on the outflow face. By allowing no flow to pass through the sides of the cell, all fluxes are forced to go in the principal direction of flow. Therefore, this type of boundary conditions is often referred to as the no-flow or sealed-sides boundary conditions.

The sealed-sides boundary conditions are graphically illustrated in Fig. 17.21 for flow in the vertical direction (here in the case of a flow cell containing a barrier).
Implementation is an important factor in the relative efficiencies of different formulations. For a given problem, different models using the same formulation can give widely different CPU times. For example, the IMPES CPU times reported by different participants in the six-component compositional SPE5 problem differed by a factor of over 50.

17.1.3 Advances in Model Formulations. The IMPES formulation was improved by concepts of relaxed volume, better choice of variables, and "adaptive" flash calculations.

**Relaxed Volume.** The relaxed volume concept relates to the timestep calculation Steps (d) and (e) given previously. Step (d) gives the mass of each component in the gridblock, $M_j^{i+1}$, which in turn gives overall composition $\{z_1\}^{i+1}$. The Step (e) flash then gives phase amounts and densities which in turn give new iterate $S_w$, $S_o$, and $S_g$ values. These saturations do not sum to 1.0 because of the nonlinear nature of the conservation Eq. 17.1. If the saturations are altered (e.g., divide each by their sum) to exactly satisfy the volume balance $\Sigma_j S_j = 1$, then an incremental (timestep) mass-balance error occurs. If the saturations are not altered, then mass is conserved but there is a volume-balance error $L_j S_j - 1$. The authors chose to preserve mass and carry forward the volume balance error from iterate to iterate and step to step. The volume balance going into iterate $i + 1$ is $\Sigma_j S_j = 1 - \Sigma_j S_j$. This in effect conserves both mass and volume because there is no permanent or accumulating volume error—only that of the given timestep. Equally important, there is no need to iterate out the volume error to a "tight" tolerance, and Newton iterations and model CPU are reduced. In contrast, the previous or historical IMPES procedure reset saturations to preserve volume and iterated out the mass-balance error. Because the latter error was not carried forward, more Newton iteration (and CPU time) was required to keep the permanent, accumulating mass balance error tolerably low. This use of relaxed volume with carryover also reduces Newton iterations and CPU time in the Implicit formulation.
permeability tensor is derived by setting up the boundary conditions for x, y, and z directions, respectively, in three independent single-phase simulations. As documented in Refs. 71 and 72, a tensor technique based on the sealed-sides boundary conditions tends to bias the estimated effective permeability toward a low value. The physical implication of this is most clearly seen in the case of a bimodal permeability system of sand and shale. This is because the sealed-sides method consistently underestimates the reservoir flow characteristics by thickening shale barriers and narrowing sand channels. The latter effect also has a tendency of disconnecting stacked sand channels.

Take, for example, the flow illustrated by Fig. 17.21. Because the barrier extends across the entire length of the local upscaling region, the resulting effective permeability (in the z-direction) will be zero. For vertical flow, the result, therefore, is a thickening of the shale in the flow model equal to the thickness of the flow cell. Depending on which factors that affect fluid flow in the region of the cell, this may or may not be a representative value for that particular flow cell.

Strictly speaking, the sealed-sides boundary conditions are only valid if no wells are present and the flow cells are symmetric in each direction of the grid as illustrated in 2D by Fig. 17.22. Hence, the sealed-sides boundary conditions assume that the flow cell is surrounded by mirror images of itself.

By the end of the 1980s, 3D geological models had started to appear more regularly on the modeling scene. This resulted in a new demand for advanced upscaling. In this renewed effort, two alternative boundary conditions for solving the local pressure solution in a flow-based method were suggested more or less at the same time. One was based on linear boundary conditions.

The use of linear boundary conditions in flow-based upscaling was suggested by Guerillot et al. in 1989 and Samier in 1990 to enable the computation of a full-permeability tensor. Instead of setting the flow through the sides of the cell to zero, the pressure along the sides is allowed to vary in a linear fashion that matches the constant pressure on the two cell faces perpendicular to the flow. Hence, the imposed pressure gradient is still constant, but the flow is allowed to enter and leave the cell at any point along the sides parallel to the main flow direction. Therefore, this type of boundary conditions is also referred to as the open-sides boundary conditions. The situation is graphically illustrated in Fig. 17.23 for flow in the vertical direction (here in the case of a flow cell containing a barrier).

As with the sealed-sides boundary conditions, three independent single-phase simulations, with the main flow direction in x, y, and z respectively, are needed to yield all of the components of the permeability tensor. With open-sides boundary conditions, however, also the off-diagonal components will generally be nonzero. Hence, unlike the sealed-sides boundary conditions where the effective permeability is limited to that of a diagonal tensor, the open-sides boundary conditions, as previously mentioned, give a full permeability tensor. The resulting full tensor may be either symmetric or nonsymmetric depending on the properties of the method under consideration.

As documented in Refs. 75 and 76, a tensor technique based on the open-sides boundary conditions tends to bias the estimated effective permeability toward a high value. The physical implication of this is most clearly seen in the case of a bimodal permeability system of sand and shale. This is because the open-sides method consistently overestimates the reservoir flow characteristics by narrowing shale barriers and thickening sand channels. The latter effect also has a tendency of connecting isolated sand channels.

Take, for example, the situation illustrated by Fig. 17.23. Even though the barrier extends across the entire length of the local upscaling region, the resulting effective permeability (in the z-direction) will be significantly larger than zero. For vertical flow, the result is therefore a narrowing of the shale in the flow model equal to the horizontal dimensions of the flow cell. Depending on which factors affect fluid flow in the region of the cell, this may or may not be a representative value for that particular flow cell.

The use of linear boundary conditions has its origin in the effective medium theory, which states that any region of permeability behaves as if embedded within the average medi-
Fig. 17.24—The fine-scale properties are uniform and nonzero outside the upscaling region (the geo cells outside the upscaling region are drawn in gray to reflect that the average value need not be known).

Strictly speaking, these boundary conditions are therefore only valid if the neighboring flow cells are of a uniform, nonzero permeability. This is illustrated in 2D by Fig. 17.24.

The use of periodic boundary conditions originates from the volume averaging theory, and its use in flow-based upscaling was first introduced by Durlofsky and Chung in 1990 and by Durlofsky in 1991.\(^{75,76}\) Durlofsky used periodic boundary conditions, together with Darcy’s law and the classic requirement of flux conservation, to derive a full permeability tensor. A somewhat different approach, which also uses a periodic pressure field around the flow cell, was proposed by Øistein Boe et al. in 1994.\(^{77,78}\) This uses a weak form of Darcy’s law to prove that periodic boundary conditions result in a full permeability tensor that is both symmetric and positive definite. The Norsk Hydro tensor method is based on the conservation of dissipation (mechanical energy per unit weight of fluid), although it turns out that fluxes are conserved as well.

A simplistic illustration of the periodic boundary conditions is given in Fig. 17.25 for flow in the vertical direction (here in the case of a flow cell containing a barrier).

Although the periodic boundary conditions generally result in an effective permeability that is higher than that computed with the sealed-sides boundary conditions, the effective vertical permeability for the upscaling region illustrated in Fig. 17.25 will also be zero.

Fig. 17.25—The periodic boundary conditions in z-direction (\(P_{\text{in}} = P_{\text{out}}\), \(P_{\text{in}} = P_{\text{out}}\)).

Please note that if a medium is symmetric on the scale of \(\Delta l\), then it will be periodic on the scale of \(2\Delta l\).

The relative performance of the tensor methods that is caused by the various boundary conditions has proven to be of considerable interest. As it happens, the sealed-sides method provides a lower bound and the open-sides method an upper bound of the effective permeability. The periodic-based method turns out to give an effective permeability estimate that generally lies in between the two previous methods.

With regard to the outer bounds of effective permeability, it is well known that the harmonic and arithmetic means provide the absolute lower and upper limit of the effective permeability, respectively. It is less known that the uncertainty range in the effective permeability may be narrowed using the composite averages. In fact, it may be mathematically proven that the harmonic-arithmetic average provides a closer lower limit than the pure harmonic mean, whereas the arithmetic-harmonic average provides a closer upper limit than the pure arithmetic mean (truly valid only for regular grids). In this context, it is important to realize that the two flow-based methods (sealed and open sides) provide an even narrower uncertainty band for the effective permeability, but at the expense of increased CPU time.

The relative performance of the most important local upscaling techniques is shown in Fig. 17.27. Fig. 17.27 also indicates the “inner” uncertainty range of the true effective permeability for the sake of comparison.

17.4.6 Regional Upscaling. Regional upscaling is applied to reduce the influence of the artificial boundary conditions on the effective permeability estimate by moving the boundary of the computational region away from the flow cell. This implies that the influence of neighboring geo cells is taken into account in addition to the geo cells inside the flow cell. In other words, regional upscaling represents an expansion of the local computational region outside the volume of the flow cell. The size of the so-called buffer or skin around each flow cell is usually
Still, as discussed by Holden and Nielsen,39 the global upscaling approach is not enough to ensure maximum accuracy in the modeling of the effective permeability. Because the value of the effective permeability is influenced by changes in the pressure field, the flow-cell permeabilities should strictly be recomputed by the global method for every timestep taken by the multiphase flow simulator. In practice, though, it might be good enough to update the effective permeability field whenever a significant change occurs as a result of altering the well configuration or production/injection rates and so on. Hence, the ultimate upscaling scheme for the absolute permeability might be the one that is coupled with the multiphase flow simulator and automatically updates the absolute effective permeability field for each timestep. With the current computer power and the lack of proper integration between the geological model and the simulation model, this is hardly achievable yet.

17.4.8 Best Practice Guidelines. As may be understood from the previous sections, there exists no single upscaling method for absolute permeability that is superior to all other methods in all situations, at least not until it has been fully established that the global upscaling scheme represents the ultimate method of choice. Selecting the proper upscaling method from the many available choices can be quite a challenge. The choice of sophistication in the upscaling method generally depends on one or several of the following factors:

- The complexity of the fine-scale permeability distribution (i.e., the geo model).
- The degree of upscaling that needs to be performed (i.e., the coarsening factor).
- The number of permeability realizations that need to be upcaled.
- The time available to the project for performing upcalsing.
- The intended use of the flow model.

Because an exact validation of the upscaling process cannot be performed unless a multiphase flow simulation is carried out on the geo model itself, two alternative upscaling approaches for identifying the proper homogenization method are presented here.

The Absolute Upscaling Approach. This approach assumes that there exists a way to properly validate the absolute performance of an upscaling method without resorting to an extremely time-consuming (if at all feasible) finite-volume simulation of the geo model. As documented in Ref. 81, streamline simulation offers a very efficient way of validating the performance of upscaling methods. The validation process is carried out by first running a streamline simulation on the geo model to compute the reference solution. Then a streamline simulation is run on the flow model for each of the upscaling methods that are to be evaluated. The simulated performance of the various upscaling methods is then compared to that of the geo model. The validation of upscaling methods is best done under single-phase flow conditions to avoid introducing other model parameters (e.g., relative permeabilities and associated rock properties) that also need to be upcaled in one way or the other. A higher confidence may also be obtained for the validation process if the actual well pattern is used in the streamline simulations.

Using the previously described validation scheme, the modeler may choose to evaluate any upscaling method until one with a satisfactory performance is found. Still, a more systematic way of identifying the optimum upscaling method is desirable. With the absolute upscaling approach that is presented here, the modeler is offered a multistep procedure that is to be terminated as soon as a satisfactory upscaling method has been identified. The recommended procedure involves the following steps:

1. Compute the upper and lower bounds of the effective permeability using the arithmetic-harmonic (or pure arithmetic) and harmonic-arithmetic (or pure harmonic) average techniques, respectively. Being of the analytical type, these methods are very fast and will provide a first quantification of the upscaling uncertainty. Validate the performance of the two composite methods against that of the geo model using a single-phase streamline simulator with the actual well pattern.

2. If the performance of any of the two methods in Step 1 is within an acceptable range of the geo model, then terminate the procedure and choose the appropriate method. If, on the other hand, the performance of both methods is unsatisfactory because of the complexity of the geo model, then use the upper (open-sides) and lower (sealed-sides) bound diagonal tensor methods to narrow the uncertainty in the flow-model performance. Validate the performance of the two tensor methods against that of the reference solution.

3. If the performance of any of the methods in Step 2 is within an acceptable range of the reference solution, then terminate the procedure and choose the appropriate method. If, on the other hand, the performance of both methods is unsatisfactory, then the following alternatives may be worth considering:
   a. If time allows, refine or coarsen the flow grid (whatever is best) to achieve a better separation of the length scales. Then repeat the upscaling of the outer bounds (in Step 2) and redo the validation to check if the performance of either method has improved.
   b. Apply a tensor method with periodic or semi-open-boundary conditions [the semi-open boundary conditions alternative is available in some applications using a multiplier between 0 (sealed) and 1 (open) to the side faces of the computational region]. As previously mentioned, this should result in an intermediate estimate of the effective permeability tensor and therefore provide a flow model performance that lies somewhere in between the two methods in Step 2.
   c. Select the best of the two tensor methods in Step 2 and convert the local method to a regional method using a buffer region of modest size. Validate its performance. If necessary, repeat this step using an increasingly larger buffer region until a satisfactory performance of the flow model is obtained.
   d. If for some reason none of the previous alternatives are an option, then one needs to apply the method that best satisfies the wanted flow behavior of a given cell or cells in a given region. In other words, a combination of the outer bound techniques within one and the same model may be a fourth alternative.

As previously mentioned, the method using open-sides boundary conditions is a good estimator of sand continuity and quality, whereas the sealed-sides boundary conditions method is better at detecting the presence and effect of barriers.

Consider a long horizontal oil producer in the Troll West Gas Province that is protected above and below the overlying gas cap by a calcite barrier just above the well. If the local grid resolution in the Troll full-field model was such that one could apply the open-sides boundary conditions technique on the cells containing the well, and the sealed-sides boundary conditions technique on the cells containing the barrier, then this would be the optimum local upscaling approach.

However, if both a segment of the well trajectory and a segment of the calcite are present inside the same flow cell, then the open-sides boundary conditions technique will give a good estimate of the well's PI but result in a much too early gas breakthrough, whereas the sealed-sides boundary conditions method will better capture the effect of the calcite but give a too low estimate of the well’s PI. If this is the case, one needs to consider applying one of the alternatives A or C (alternative B will, in this particular example, give the same result as the sealed-sides method).

The multistep procedure of the absolute upscaling approach is graphically illustrated in Fig. 17.29.

The Relative Upscaling Approach. This approach acknowledges the fact that an exact validation of the upscaling results cannot be achieved in practice. Therefore, instead of trying to validate the absolute performance of an upscaling method, the approach diagnoses the relative performance of outer bound methods using the actual multiphase finite-volume simulator. This implies that a full black-oil simulation is run on the flow model for each of the upscaling methods that are to be evaluated. The deviation in the simulated performance between outer bound
methods will then reflect the part of the model uncertainty that originates from the upscaling process itself. To ensure a high degree of relevance in the diagnostics, it is important that the test simulations contain a representative description of the actual flow model.

If the project is pressed for time, the simulations may be skipped altogether in favor of a faster, although less robust, way of performing the diagnostics. Instead of analyzing simulation profiles, a normalized difference parameter may be computed on a cell by cell basis using the formula

\[ \phi = \frac{K_{\text{upp}} - K_{\text{low}}}{(K_{\text{upp}} + K_{\text{low}})/2} \] .............................. (17.18)

The relative upscaling approach that is presented here utilizes a multistep procedure that applies outer bound methods of increasing accuracy until the best possible upscaling method can be identified. The recommended procedure involves the following steps:

1. Compute the upper and lower bounds of the effective permeability using the arithmetic-harmonic (or pure arithmetic) and harmonic-arithmetic (or pure harmonic) average techniques, respectively. Being of the analytical type, these methods are very fast and will provide a first quantification of the upscaling uncertainty. Run the finite-volume simulator for each of the two composite methods (or compute a grid-based difference parameter) and perform the diagnostics.

2. If the performance gap between the two methods in Step 1 is acceptable (small), then terminate the procedure and choose either of the two. If the deviation in performance is unsatisfactory because of the complexity of the geo model, then use the upper (open-sides) and lower (sealed-sides) bound diagonal tensor methods to narrow the uncertainty in the flow model performance. Run the finite-volume simulator for each of the two tensor methods (or compute a grid-based difference parameter) and perform the diagnostics.

3. If this reduces the upscaling uncertainty to within acceptable limits, then either of the two diagonal tensor methods may be used to provide the final permeability field for the flow model. If the deviation in the performance is unsatisfactory (large), then the following alternatives may be worth considering:

   1. If time allows, refine or coarsen the flow grid (whatever is the best) to achieve a better separation of the length scales. Then repeat the upscaling of the outer bounds (in Step 2) and rerun the simulations (or recompute the difference parameter) to check if the performance gap (upscaling uncertainty) has narrowed.

   2. Apply a tensor method with semi-open or periodic boundary conditions. As previously mentioned, this should result in an intermediate estimate of the effective permeability tensor and hence provide a flow model performance that lies somewhere in between the two methods in Step 2.

   3. Convert the two local tensor methods in Step 2 into regional methods using a buffer region of modest size. Rerun the finite-volume simulator (or recompute the difference parameter) and check the performance gap. If necessary, repeat this step using an increasingly larger buffer region until the upscaling uncertainty reaches acceptable limits, at least as far as practically possible.

   4. If for some reason none of the previous alternatives are an option, then one needs to apply the method that best satisfies the wanted flow behavior of a given cell or cells in a given region. In other words, a combination of the outer bound techniques within one and the same model may be a fourth alternative.

The multistep procedure of the relative upscaling approach is graphically illustrated in Fig. 17.30.
17.5 Streamline Simulation—Rod P. Batycky and Marco R. Thiele

17.5.1 Introduction. Streamline-based flow simulation differentiates itself from cell-based simulation techniques such as finite-differences and finite-elements in that phase saturations and components are transported along a flow-based grid defined by streamlines (or streamtubes) rather than moved from cell-to-cell. This difference allows streamlines to be extremely efficient in solving large, heterogeneous models if key assumptions in the formulation are met by the physical system being simulated (see below). Specifically, large relates to the number of active grid cells.

Streamlines represent a snapshot of the instantaneous flow field and thereby produce data such as drainage/irrigation regions associated with producing/injecting wells and flow rate allocation between injector/producer pairs that are not easily determined by other simulation techniques. The computational speed and novel solution data available have made streamlines an important, complementary approach to traditional simulation approaches to perform sensitivity runs, quantify the impact of upscaling algorithms used to move models from the geoscience scale to the simulation scale, visualize the flow field, perform more reliable full-field simulations where sector models would normally be used, enable the ranking of predicted field behavior of given multiple production scenarios and input parameters, evaluate the efficiency of injectors and producers, reduce turnaround time in history matching, and perform other established reservoir engineering tasks. A comprehensive overview on streamline-based flow simulation has recently been presented by Thiele.82

17.5.2 Applicability of Streamline Simulation. The power of streamline simulation lies in its simplicity. The main objective is to capture how injected reservoir volumes (usually water and/or gas) displace resident reservoir volumes given well locations, well rates, reservoir geometry, and geological description. One of the key underlying assumptions in streamline simulation is that the system be close to incompressibility. This decouples saturations from the underlying pressure field and allows each streamline to be treated as being independent from the streamlines next to it.

Many fields under waterflooding or other pressure maintenance schemes are excellent candidates for streamline modeling and have been successfully modeled in this way.83,84 Forecast simulations under the assumption of voidage replacement are another good example where streamlines can be very effective. Even miscible gas injection schemes have been successfully modeled.85 At high pressures, the displacement of resident oil by gas is primarily an issue of interfacial tension.86 Since then, streamlines and streamtubes have received repeated attention as a way simulating local sweep efficiency and channeling, something streamlines are designed to model without incurring numerical difficulties associated with other formulations.87 An early application to streamline projects was presented by Emanuel.88 Crane and Blunt89 used streamlines to model solute transport. More recently, streamlines have been shown to be very effective in modeling fractured reservoirs using a dual porosity formulation.90

17.5.3 Historical. Streamlines have been in the petroleum literature as early as Muskat and Wyckoff’s 1934 paper.91 In 1937, Muskat presented the governing analytical solutions for the stream function and the potential function for 2D domains using the assumption of incompressible flow.92 Since then, streamlines and streamtubes have received repeated attention as a way to numerically predict the movement of fluids, even after the advent of finite-difference methods in the early 1960s. Important early contributions were made by Fay and Pratts,93 Higgins and Leighton,94 Bommer and Schecter,95 Martin and Wegner,96 Lake et al.,97 and Emanuel et al.98

In the early 1990s, streamlines were revived because advances in geological modeling techniques were producing models that were too large for finite differences to simulate in an acceptable time frame. For streamlines to be applicable to real field cases, important advances

---

Fig. 17.30—Flow chart for the relative upscaling approach.
were made that extended streamlines to 3D using a time-of-flight variable, allowing for streamlines to be periodically updated and included gravity.

17.5.4 Mathematics of the Streamline Method. The streamline method and the underlying mathematics for incompressible multiphase flow are briefly outlined here. For a detailed discussion as well as additional references describing streamline methods, see Batycky et al. and Blunt et al.

**Governing IMPES Equations.** The streamline method is an IMPES-type formulation with the pressure field solved for implicitly and the oil/gas/water saturations solved for explicitly along streamlines. The governing equation for pressure, \( P \), for multiphase incompressible flow without capillary or diffusion effects is given by

\[
\nabla \cdot \left( \sum_{j=1}^{n} \frac{1}{\mu_j} k_j (\nabla \cdot (P + \rho_j g D)) \right) = 0, \quad \text{(17.19)}
\]

where \( D \) is the depth below datum, \( g \) is gravitational acceleration constant, \( k \) is the permeability tensor, \( \mu_j \) is the relative permeability, \( \rho_j \) is viscosity, and \( \rho_l \) is the phase density of phase \( j \).

The total velocity, \( \mathbf{u}_P \), is derived from the 3D solution to the pressure equation and application of Darcy’s law. The explicit material-balance equation for each incompressible phase \( j \) is then given by

\[
\frac{\partial S_j}{\partial t} + \mathbf{u} \cdot \nabla f_j + \nabla \cdot \mathbf{G}_j = 0, \quad \text{(17.20)}
\]

Each phase fractional flow, \( f_j \), is given by

\[
f_j = \frac{k_{ij}}{\mu_{ij}} \frac{1}{\sum_{j=1}^{n} k_{ij} \mu_{ij}}, \quad \text{(17.21)}
\]

and the phase velocity resulting from gravity effects because of phase density differences is given by

\[
\mathbf{G}_j = \mathbf{k} \cdot g f_j \nabla D \sum_{i=1}^{n} k_{ij} (\rho_i - \rho_j) / \rho_j. \quad \text{(17.22)}
\]

The difference between finite-difference simulation and streamline simulation is the way the explicit material balance equations (Eqs. 17.20 through 17.22) are solved. In finite difference, the material balance equations are solved between gridblocks, whereas in streamline simulation the material balance equations are solved along streamlines. How this is done is explained next.

**Solution to the Transport Equation.** In a standard finite-difference method, Eq. 17.20 is discretized and solved on the underlying grid on which the pressure field is computed. The solution to Eq. 17.20 is governed by the grid CFL condition, which can lead to prohibitively small timestep sizes, particularly for models with high permeability contrasts and/or high local flow velocities. With streamlines, this grid CFL limit is avoided completely by solving Eq. 17.20 along each streamline using a time-of-flight (TOF) coordinate transform.

Streamlines are traced from sources to sinks based on the underlying total velocity field. As each streamline is traced, compute the TOF along the streamline, which is defined as

\[
T = \int_0^T \frac{\phi(D)}{\mu(D)} \, d\tau, \quad \text{(17.23)}
\]

and leads to the definition

\[
\mathbf{u}_P = \phi \mathbf{D} = \nabla \phi / \phi \mathbf{D}, \quad \text{(17.24)}
\]

Using Eq. 17.19, rewrite Eq. 17.20 as

\[
\frac{\partial S_{f_j}}{\partial t} + \frac{\partial f_j}{\partial t} + \frac{1}{\phi} \nabla \cdot \mathbf{G}_j = 0, \quad \text{(17.25)}
\]

Because the gravity term is not aligned along a streamline direction, Eq. 17.25 is split into two parts (operator splitting), giving two 1D equations. The convective portion of the material-balance equation along streamlines is given by

\[
\frac{\partial S_{f_j}}{\partial t} = 0, \quad \text{(17.26)}
\]

while the portion resulting from phase-density differences solved along gravity lines is given by

\[
\frac{\partial f_j}{\partial t} + \frac{1}{\phi} \nabla \cdot \mathbf{G}_j = 0. \quad \text{(17.27)}
\]

Both Eqs. 17.26 and 17.27 represent 1D equations that are solved using standard finite-difference numerical techniques. There are still CFL limits that restrict timestep sizes in these equations, but these are local to each streamline or gravity line, rather than at the 3D grid level.

**Timestepping.** In field-scale displacements, the streamline paths change with time because of the changing fluid distributions and the changing well conditions. As a result, the total velocity field is periodically updated, and new streamlines are recomputed to reflect the nonlinear nature of the displacement.

To move the 3D saturation distribution forward in time between successive streamline distributions from time \( T_i \) to \( T_{i+1} = T_i + dT_i \), the algorithm pictured in Fig. 17.31 is used.

The basic algorithm for streamline-based flow simulation is as follows: (1) Given initial conditions (i.e., pressures and saturations for each active cell in the system) and well conditions, the pressure is solved implicitly for each cell, as is done in conventional finite-difference methods (Eq. 17.19). (2) With the pressures known, the total velocity for each cell interface can be determined using Darcy’s Law. The total velocity is then used to trace streamlines using Pollock’s algorithm. (3) 1D mass conservation equations are then solved along each streamline, independently of each other (Eq. 17.26). The initial conditions for the streamlines are obtained by a mapping from the underlying 3D grid onto each streamline. The mass-transport problem is marched forward in time along each streamline for a pre-specified global timestep \( dT_i \), and then the solution is mapped back onto the 3D grid. Gravity is included by...
The fine-scale properties are periodic on the scale of the flow cell (geo grid in white, flow grid in black).

Fig. 17.26—The fine-scale properties are periodic on the scale of the flow cell (geo grid in white, flow grid in black).

The permeability estimate of a regional upscaling method will improve as the size of the buffer region increases, and it will ultimately be equal to the "true" effective permeability when the buffer size has reached the boundaries of the geo model for all three directions. The gain in accuracy is largest in the beginning (i.e., for small buffer values). This is illustrated by Fig. 17.28, showing the behavior of the lower- and upper-bound tensor techniques in the case of increasing buffer size. Please note that in Fig. 17.28, the outer bounds are shown to be symmetric around the "true" effective permeability. Generally, this is not the case.

17.4.7 Global Upscaling. Strictly, the fine-scale pressure field must be determined for the entire geo grid simultaneously to compute "exact" effective permeabilities for the flow cells. In the past, however, this has been too CPU-intensive to be performed in practice. With the introduction of new and promising solution algorithms such as the Output Least Squares (OSL) method, global upscaling schemes can now be realized. In the paper by Holden and Nielsen, the OSL method is used to minimize the difference in pressure, as well as velocity, between the geo and flow grids in an iterative process. Because the CPU consumption of the applied equation solver is proportional to the number of geo cells, a global solution will use approximately the same amount of computational time as the sum of all the local computations. Therefore, the new global upscaling scheme is just as fast as any local method.

An obvious advantage with the global upscaling approach is that one avoids using artificial boundary conditions around the upscaling region (i.e., instead of guessing what the boundary conditions for the flow cells might be, the pressure conditions surrounding the cells are explicitly known). Another important benefit is that a poor separation of scales in the upscaling will no longer occur because the size of the computational region is the same as the geo model.

Although still in its research stage, global upscaling has much potential for improving today's permeability estimators, especially for models containing a complex facies architecture with large permeability contrasts between facies. In fact, according to Holden and Nielsen, preliminary results show an improvement factor of 10 in some cases.
considering a vertical segregation step along gravity lines after movement along all streamlines (Eq. 17.27). While simple in its approach, important details must be considered. In particular:

1. The algorithm is similar to an IMPES approach, in that the pressure is solved implicitly for a new time level \( t^{n+1} \) assuming saturations at level \( n \). The saturations at time \( n \) are given by mapping back solutions from each streamline onto the 3D grid at the previous timestep. Because of the implicit nature of the pressure solution, there is no limitation on the timestep to reach \( n+1 \). However, for compressible systems numerical convergence problems might limit the actual size of the timestep. This is no different than in FD simulation.

2. The tracing of the streamlines using Pollock’s algorithm assumes Cartesian cells. Nonorthogonal corner-point cells require an isoparametric transformation for tracing streamlines.\(^{118}\)

3. For incompressible systems, streamline will start at injection wells and end at production wells. For compressible systems, streamline can start/end anywhere in the system, because any gridblock in the system might act as a source (volume expansion) or a sink (volume contraction). Multiphase gravity effects can give rise to circulation cells for both incompressible and compressible systems.

4. Initial launching of streamlines from wells can be proportional to the total flux at the wells, though this will in general leave many cells in the system without a streamline passing through them. For missed cells, tracing begins at the center of the missed cell and then traced backward until a source is encountered. If a cell does not have a streamline pass through it, then it is not possible to assign an updated saturation back to that cell.

5. In practice, it is not possible to have all streamlines carry the same flux and ensure at least one streamline per cell. Thus, streamlines do not carry the same flux. Furthermore, for incompressible problems the flux along each streamline is a constant, while for compressible systems it is not.

6. The tracing of streamlines using the TOF variable produces a highly irregular 1D grid along each streamline. To numerically solve the 1D problem efficiently, the 1D grid must be regularized, solved using an implicit approach, or regridded in some way to allow for a more efficient solution.

7. The tracing of the streamlines relies on an accurate solution of the velocity field. Excessive distortions of the grid (nonorthogonal) or a pressure solution that has not been solved to a small enough tolerance can cause problems in tracing streamline paths.

17.5.5 The Computational Efficiency of Streamlines. One advantage of streamline simulation over more traditional approaches is its inherent efficiency, both in terms of memory and computational speed. Specifically, streamline-based simulation can exhibit a near-linear scaling in run times as a function of active cells in the model. Memory efficiency is a result of two key aspects of the formula: streamline-based simulation is an IMPES-type method and therefore involves only the implicit solution of pressure, and tracing of streamlines and solution of the relevant transport problem along each streamline is done sequentially. Only one streamline needs to be kept in memory at any given time.

Computational speed, on the other hand, is achieved because the transport problem is decoupled from the 3D grid and instead solved along each streamline. Because transport along streamlines is 1D, they can be solved efficiently. Because the number of streamlines increases linearly with the number of active cells, and streamlines only need to be updated infrequently, the computational time exhibits a near-linear scaling with increasing number of gridblocks.

The number of global timesteps is related to how often the flow field (streamlines) requires updating. Specifically, changing flow paths are a function of heterogeneity, mobility changes, gravity, and changing well conditions. For many practical problems, it is the changing well rates that introduced the greatest impact on a changing flow field and is therefore the limiting factor in deciding on global timestep sizes. Grouping well events into semiyearly or yearly intervals and assuming that the streamlines remain unchanged over each period is reasonable. This is why field simulations with 30- to 40-year histories are successfully and routinely simulated with 1-year timesteps.\(^{124}\)

A good example to demonstrate the efficiency of SL simulation is Model 2 of the 10th SPE comparative solution project.\(^1\) The total run time, \( T \), of any streamline simulation is approximately proportional to...
Streamlines can be used to:

1. Streamline production and management in the future. Specifically, streamlines can be used to:

   a. Visualize flow.
   b. Balance patterns.
   c. Determine efficiency of injectors and producers using data provided by streamlines.
   d. Aid in history matching.
   e. Enable ranking of production scenarios/geological models.
   f. Optimize and manage field injection/production.
   g. Conduct reservoir surveillance.

   It is important to underline that the theory on which streamline simulation rests is firmly rooted in the incompressible formulation of exact voidage replacement. Thus, streamline simulation is particularly powerful for modeling systems that are not a strong function of absolute pressure, but are instead governed by a pressure gradient. In addition, the strong assumptions of independence between streamlines favors modeling displacements that are not a strong function of diffusive phenomena, such as capillary pressure, transverse diffusion, or compressibility. For example, streamline simulation offers little or no advantage over conventional simulation for modeling primary production. This is because the main feature of modeling primary production is to accurately capture the pressure decline over time, not the movement of a saturation front.

A near-linear scaling arises because:

1. The number of timesteps (streamline updates) is independent of the model size, heterogeneity, and any other geometrical description of the 3D model. It is mainly a function of the number of well events and the actual displacement physics. For the SPE10 problem in Fig. 17.32, all cases were run with the exact same number of streamline updates—24.

2. An efficient pressure solver is expected to have a near-linear behavior as well.109

3. The number of streamlines tends to increase linearly with the number of gridblocks, all else being equal. Fig. 17.32 illustrates this behavior.

4. The time to solve the transport problem along each streamline can be made efficient by regularizing the underlying TOF grid and choosing the number of nodes to use along each streamline regardless of the size of the underlying 3D grid.

The linear behavior with model size is the main reason why streamline simulation is so useful in modeling large systems. In FDs, finer models not only cause smaller timesteps because of smaller gridblocks but usually face problems because of increased heterogeneity as finer models tend to have wider permeability and porosity distributions. The usual workaround is to use an implicit or adaptive-implicit formulation, but for large problems these solutions can become prohibitively expensive, both in terms of CPU time and memory.

### 17.5.6 Novel Data Produced by Streamlines

Streamlines produce new data not available with conventional simulators. Because streamlines start at a source and end in a sink, it is possible to determine which injectors are (or which part of an aquifer is) supporting a particular producer, and exactly by how much. A high water cut in a producing well can therefore be traced back to specific injection wells or boundaries with water influx. Conversely, it is possible to determine just how much volume from a particular injection well is contributing to the producers it is supporting—particularly valuable information when trying to balance patterns (Fig. 17.33) or optimize water injection over a field.

Streamlines can also identify the reservoir volume associated with any well in the system, because a block traversed by a streamline attached to a particular well will belong to that well's drainage volume. It is therefore possible to divide the reservoir into dynamically defined drainage zones attached to wells (Fig. 17.33). Properties normally associated with reservoir volumes can now be expressed on a per-well basis, such as oil in place, water in place, and average pressure, just to mention a few.

The most successful uses of new data produced by streamlines are in the area of waterflood management and reservoir surveillance.110,111 and in the area of history matching.112-115

### 17.5.7 Applications of Streamlines

Streamlines are a powerful complementary tool to more traditional simulation techniques, and they are expected to play an important part in optimizing field production and management in the future. Specifically, streamlines can be used to:

1. Validate upscaling techniques by allowing to generate reference solutions of fine-scale models.91
2. Efficiently perform parametric studies.
17.6.2 Modeling of Reservoir Compaction and/or Dilation. Modeling reservoir deformation is of considerable importance in soft and/or thick reservoirs where the results of compaction may need to be important production mechanisms, cause well failure, and/or cause ground subsidence or heave with environmental consequences. Review of the compaction mechanics and its consequences for field development is found in Ref. 121. Initial approach to modeling compaction was based on modifications of reservoir models. The common feature of such reservoir compaction models is that the compaction is treated as a 1D problem (uniaxial strain) by assuming that (a) only vertical deformations take place, and (b) each vertical column of blocks deforms independently. In such models, the porosity changes are calculated by modifying the conventional compressibility or as a function of pressure only, in the form of "compaction tables." The tables are based on results of uniaxial strain laboratory experiments, and the stress problem is not solved. The compaction of the reservoir is then obtained analytically assuming uniaxial deformation. The relation between reservoir compaction and surface subsidence can be then obtained by an independent solution of a stress problem using the computed compaction as a boundary condition. Dilation (increase of porosity) is an important geomechanical mechanism during steam injection into unconsolidated sands. This process has been also modeled by the "compaction-dilation" tables. In chalk reservoirs, increased water saturation in waterflooding causes weakening of the rock and therefore $S_b$ is an additional variable in Eq. 17.32.125

The major drawback of the use of compaction tables is that the dependence on stress indicated in Eq. 17.32 must be either ignored, or the change of stress must be estimated in terms of change in pressure. A more accurate modeling approach is to couple in some fashion the reservoir simulator with stress-strain (geomechanics) solution. Such models typically combine the solution of the multiphase flow in the reservoir and elastic-plastic solution of the deformations in a much larger domain including the reservoir, sideburden, underburden, and overburden. The majority of coupled models use the iterative coupling;126 the different variations and their shortcomings are discussed next.

Coupled models have much larger computing requirements compared with a reservoir model of the same reservoir, primarily for two reasons: first, the larger number of unknowns per gridblock, and second, the stress-solution grid must be usually much larger laterally than the reservoir grid to eliminate the effect of boundary conditions as well as extend up to surface to provide subsidence solution. Combination of these factors leads to computing times typically of one order of magnitude larger compared with conventional simulation, and even more if elastic-plastic solution is required for the stresses.

17.6.3 Modeling of Stress-Dependent Flow Properties. The primary flow-dependent property is permeability, and the problems to model its dependency in Eq. 17.33 are similar to modeling the pore-volume coupling (Eq. 17.32). However, the problem is somewhat easier because stress-dependence (or transmissibility) does not affect mass-balance formulation.

Again, the traditional approach is to use tables of $k$ vs. pressure in an uncoupled model. However, the problem remains one of replacing the dependency on effective stress by one on pressure only. Even in a single-phase, single-porosity gas-flow case, different assumptions about the stress change during depletion can lead to large errors in well decline.129 Different strategies for converting the stress-dependent data to pressure tables are based on local constraint assumptions.120

In coupled models, the permeability dependency can be usually computed explicitly on a timestep basis, and "loose" coupling can be used. In fact, a "coupled" model that deals only with flow properties coupling and ignores the pore-volume coupling can be run successfully even if the stress solution is done in larger intervals of time compared to the reservoir. Such models have been used extensively to study permeability changes in waterfloods, particularly in fractured or jointed media.131,132 Here, the advantage of coupled modeling is in its capability to predict the permeability changes from the geomechanics of reopening of fractures or failure (dilation) of joints.133 The development of anisotropy is dictated by the orientation of fractures or faults, and requires a "full tensor" treatment of transmissibilities in the flow model. In the stress-strain model, different methods of pseudoizing the fracture/joint networks into a continuum are used, which include predicting permeability as a function of effective stress and/or strain. While the need for the tensor transmissibilities in such models has been recognized,134 in injection processes dual-porosity media can be created. Therefore, reservoir description may be changing in time because of geomechanics; this aspect has been ignored in coupled models to date.

The same principles can be also applied to model hydraulically induced fractures being represented by dynamically changing transmissibility multipliers in the potential fracture plane.135 The effective stress dependency (as opposed to pressure in an uncoupled model) allows capturing the changes of the fracture propagation pressure with time, which can be large, in particular in steam injection. Another application is the prediction of production/injection-induced slippage on faults, which can induce communication between reservoir fault blocks and/or seismically.

17.6.4 Types of Coupled Models. Coupled models can be either fully coupled (i.e., all unknowns solved simultaneously) or modular (reservoir simulator and stress code). In the latter case, different coupling strategies can be used, with consequences for running speed and accuracy. The majority of coupled models use a conventional finite-difference reservoir simulator coupled with a finite-element (FEM) stress simulator. However, attempts have been made to develop fully coupled FEM codes,136 and a fully coupled geomechanics was implemented in a commercial model using a finite-difference stress solution.137 Considering the proliferation and sophistication of the geomechanics codes available outside the petroleum engineering, the modular approach offers the best solution.109,127 Generally, the reservoir simulator is the "host" or "master." Commercial stress simulators are, in principle, easy to couple to it (in particular if only permeability coupling is considered).

Because of the extreme complexity and large computing requirements of coupled models, different simplifications have been developed. The main types of models (in the order of increasing rigorosity, but also computing time) are as follows:

One-Way Coupling. Pressure and temperature changes are passed from the reservoir code to the geomechanics module, but no information is passed back on timestep basis. The geomechanics does not improve the flow solution, but the model can be useful for predictions of wellbore stability for infill drilling, fracturing pressures, and so on. Manual adjustment of "compaction tables" is possible manually through restarts. Such manually coupled solutions138 of the stress problem (at intervals of time) or one-way coupling139 were often used in early coupled modeling. The method can be satisfactory when the reservoir fluid system is highly compressible (i.e., in gas reservoirs), but can lead to errors when the porosity is strongly coupled to flow.

Loose Coupling. Reservoir and geomechanics modules are run sequentially on a timestep basis, passing converged solutions of flow and stress variables to each other. Pore volumes and permeabilities in the flow model are computed as a function of $p$, $T$, and $e_{avg}$ with stress variables lagged a timestep. However, the relationships are "distilled" into tables similar to the "compaction tables," but now a function of effective stress. The advantages are functional similarity to the uncoupled reservoirs with "compaction tables," no need for iteration during timestep, and the possibility of updating the stress solution less frequently than the reservoir solution. However, complex constitutive models of the solid (e.g., plasticity) may be difficult to represent.
that the porosity is determined directly by the constitutive model of the solid rather than by a simplified relationship.

Iterative Coupling. This method is shown schematically in Fig. 17.34. Iteration is carried out between the reservoir and stress solution at every timestep until the pore volumes and permeabilities calculated from the stress model and those used by the reservoir model agree. In each iteration, the previous guess of the \( V_n^{n+1} \) for the end of the timestep is used to converge the flow solution, and the changes of \( \rho \) and \( T \) over timestep are then used to solve for the new deformations and stresses, which in turn provide updated estimate of \( V_n^{n+1} \). The changes of permeability are also iterated on. Therefore, each "geomechanical" iteration costs the equivalent of a timestep solution of the previous methods. The original formulation of the coupling iteration\(^\text{128}\) is always convergent, and its efficiency has been recently improved.\(^\text{140}\)

The iterative coupling, when converged, is equivalent to a fully coupled code while it retains flexibility. In many problems, it is not necessary to fully converge the timestep, and if used with only one iteration per timestep, the method is similar to the loose coupling, except that the porosity is determined directly by the constitutive model of the solid rather than by a simplified relationship.

\[ n = 0 \]
\[ n_g = 0 \]

**Reservoir simulator**
\[ \rho, T, k, \phi^* \]

**Geomechanics module**
\[ u, s, e \]

**Updated properties**
\[ \phi^* = \phi^* (\rho, T, \sigma_m) \]
\[ k = k(u, s, e) \]

**Convergence**

Yes

No

\[ n_g = n_g + 1 \]

\[ n = n + 1 \]

**Fig. 17.34—Schematic of the iterative coupling algorithm.**

**Full Coupling.** This requires simultaneous formulation of the flow and stress variables and therefore results in larger matrices. The advantage is that consistent approach to discretization can be used, and the model is integrated from the point of view of code development. However, it is very costly to redevelop all the features of the physics and numerics now readily available in stress codes. Moreover, in published fully coupled models, the approach for solving the resulting matrix problem has been to partition the matrix in the same fashion as in the iterative coupling\(^\text{141}^\text{146}\) and to apply the geomechanical iteration at the matrix-solution level. Therefore, the fully coupled formulation, which results in larger, strongly nonlinear matrix equations, does not reduce the difficulty of the problem, and it may need to use geomechanical iteration in the solution process as the best strategy. These aspects need further study.

17.6.5 Future Trends and Needs. As a result of much larger computing requirements, coupled models lag behind conventional models in the size of the problems that can be currently handled. Therefore, they are a prime candidate for the use of massively parallel hardware and will require large future development effort in parallelization. Because of the increased understanding of the complexities of the geomechanics, the current trend is also toward more strongly coupled models with fewer simplifications. This further increases the computing requirements.

Given that not all problems require use of geomechanics, and the cost of the study may increase dramatically, it is important to be able to identify when coupled simulation is needed, and what approximations can be made without compromising the answers. There are no simple rules, but there is a growing need to conduct a "screening" process at an early stage of a reservoir study to determine if geomechanics is an issue. This process requires an integration of reservoir, production, and completion engineering data as well as field experience.

Finally, coupled geomechanical modeling is the future tool for truly integrated reservoir management. Conventional reservoir simulation studies ignore numerous constraints placed on the development scenarios from the point of view of drilling, completion, and operations. These constraints can be incorporated into coupled models, and additional modules can be integrated (e.g., long-term wellbore stability and sand production predictions, subsidence management, 4D seismic interpretation, and so on)\(^\text{147}\).

17.7 Pressure/Volume/Temperature (PVT) Treatment—Curtis H. Whitson

The PVT treatment of fluids in reservoir simulation describes the phase behavior of gas, oil, and water at reservoir and surface conditions. Phase behavior of a mixture with known composition consists of defining the number of phases, phase amounts, phase compositions, phase properties (molecular weights, density, and viscosity), and the interfacial tension between phases. In addition to defining the phase behavior of mixtures at a specific reservoir pressure, knowing the derivatives of all phase properties with respect to pressure and composition is important in reservoir simulation.

The calculation of phase behavior in a reservoir model can be made in one of two ways—using a "black-oil" approach\(^\text{122}^\text{143}^\text{146}\) based on simple interpolation of PVT properties as a function of pressure, or using a "compositional" approach based on a thermodynamically-consistent model such as a cubic EOS.\(^\text{14}\text{15}^\text{122}\) With either approach, the PVT quantities required by a reservoir simulator are essentially the same. Modern reservoir simulators\(^\text{14}^\text{122}\) are usually written with a general compositional formulation, whereas black-oil PVT properties are converted internally to a two-component "compositional" model; the two components are surface gas and surface oil.

A reservoir simulator keeps track of overall composition in each computational grid cell as a function of time. A grid cell’s overall composition changes because of cell-to-cell fluxes and because of production or injection in wells. The phase fluxes and component movement within the reservoir are greatly affected by phase behavior (e.g., the mobility of each phase and which
components are carried in each phase). The surface products from wells are dependent on the phase behavior of produced wellstreams, but at conditions far removed from reservoir pressure and temperature.

In most reservoir simulators, the water phase and water component are treated in a simplified manner—namely, that the water component does not partition into the hydrocarbon phases, and the hydrocarbon components do not partition into the water phase; the term “hydrocarbon” also includes N₂, CO₂, and H₂S. Because of relatively high CO₂ solubility in water, and the potential importance of CO₂ “accounting” in CO₂ floods, some compositional models allow CO₂ partitioning in the water phase.¹⁵⁵

Conceptually and computationally, it is feasible to allow complete partitioning of all components in all phases in reservoir simulation. The partitioning related to the water phase and the water component could be treated with simple pressure-dependent tables, or even with an EOS model. The water-related K-values have simple composition dependence that would make EOS-based fugacity updates almost trivial. The main problem would be treating the impact of changing salinity on water-related K-values. From a practical point of view, however, modeling water-related component partitioning will have a marginal impact on reservoir performance. Hereafter, only phase behavior of nonaqueous phases will be discussed.

17.7.1 Number of Phases and Phase Type. The starting point for PVT calculations in a model grid cell is to determine if the overall composition is single-phase or two-phase at the current pressure estimate. If a cell is single-phase, the phase “type” (gas or oil) may also be needed to select the proper relative permeability and capillary pressure curves. For a black-oil PVT model (i.e., any model with simple pressure-dependent K-values), determining the number of phases and phase identification is trivial.

For EOS models, two methods can be used for establishing how many phases exist: the Michelsen¹⁴⁸ phase stability test or a saturation pressure calculation.¹⁴⁸ The stability test is relatively “slow” because good K-value estimates are not used to initiate the calculation, whereas a saturation pressure can be quite fast because good K-value estimates are often available. A stability test is more reliable in the sense of giving a conclusive result, whereas a near-critical saturation pressure test may not be found¹⁴⁹ or converge correctly to a nontrivial solution. Another advantage of using saturation pressure is that it gives a consistent method to define phase type for single-phase cells, something that is not provided by the stability test. The choice of which method to use depends on the tradeoff between speed and reliability. Both methods, if detecting two phases, give excellent starting K-values to initiate the two-phase flash calculation.

17.7.2 Two-Phase Flash. Having established that two phases exist in a cell, one must perform a flash calculation. The flash calculation uses the overall molar n and molar composition z to determine the molar amounts of each phase (n₁ and n₂), and the phase compositions (y₁ and x₁)

For simple pressure-dependent K-value models, the Rachford-Rice¹⁵⁰,¹⁵¹ (RR) procedure is used to find molar amounts and compositions,

\[
\sum_{i=1}^{N} (y_i - x_i) = \sum_{i=1}^{N} z_i (K_i - 1) = 0, \quad \text{(17.34)}
\]

solving for \(a_1 = n_2 / (n_1 + n_2)\), with \(x_i = z_i / [1 + a_i (K_i - 1)]\) and \(y_i = K_i x_i\).

For an EOS model, using the RR solution can be used, but an additional constraint must be satisfied: the fugacity of each component must be the same in each phase.¹⁵₃ \(f_0 = f_r\). Using an initial set of \(K_i\) estimates, the RR equations is solved. Resulting compositions \(y\) and \(x\) are then used to calculate fugacities \(f_0\) and \(f_r\). If the constraint \(f_0 = f_r\) is not satisfied for all components, a new set of \(K\)-values is estimated using some form of a successive substitution (SS) update \(K_{new} = K_{old} \cdot f_r / f_0\).

For reservoir simulation, a Newton solution of the flash equation should be much faster than SS methods. Including the equal-fugacity constraint within the set of nonlinear equations used to solve for model pressures “automatically” provides the first Newton update of the flash equation.¹⁵₃ Usually only one additional Newton iteration of the flash equation is needed to converge the fugacity constraints to required tolerance.

In summary, K-values alone determine the phase amounts and phase compositions (from the Rachford-Rice equation). The EOS model guarantees rigorous thermodynamic consistency through the equal-fugacity constraint, ensuring that K-values properly account for pressure and composition dependence of the phase split and component partitioning.

Having completed the flash calculation in a cell, the following information is known: phase moles, phase masses, phase densities, phase volumes (saturations), and phase compositions. The viscosity and gas/oil IFT can be calculated using compositional correlations,¹⁵² or interpolated from pressure-dependent tables for a black-oil model. The impact of PVT properties on reservoir performance will consider individual PVT properties and their impact on individual terms in the flow equations.

17.7.3 Density. Phase molar amounts, \(n_1\) and \(n_2\), are converted to phase volumes using phase molecular weights and densities: \(V_s = \sum y_i M_i / \rho_i\) and \(V_w = \sum y_w \rho_w / \rho_i\), where \(S = V_s / V_w\) and \(S = V_w / V_s\). Saturation determine relative permeabilities, which can have a dramatic impact on phase fluxes. Typically, \(k_s < S\), where \(n = 2 \text{ to } 4\). For \(n = 3\), a 5% error in density results in a 15% error in \(k_s\).

For reservoirs with sufficiently high vertical permeability, gravity will often play an important role in recovery. Gravity segregation is caused by vertical fluxes driven by the potential energy \(\Delta P = \rho w g w (P_w - P) + \rho g (Pw - P)\), where \(\rho_w = \rho_w - \rho_g\). Likewise, the condition for equilibrium initialization \(\nabla \rho_w = \nabla \rho_o = 0\) results in saturation-depth distributions, which must honor input \(\rho_w g w = \rho_o g_o S_w\) and \(\rho_w g_w = \rho_o g_o S_o\) relationships.

A +5% error in oil density (e.g., 630 instead of 600 kg/m³) and a ~5% in gas density (e.g., 380 instead of 400 kg/m³) results in gravity segregation potential (\(\Delta \rho_w = \rho_w - \rho_o\)) in error by 25% (250 kg/m³). Similarly, in the initialization of a water/oil system, the inadvertent use of \(\rho_w = 62.5 \text{ lbm/ft}³\) instead of the correct value of 70.0 lbm/ft³ for a high-salinity brine could easily cause a 25% error in initial oil in place for a low-permeability (large capillary transition zone), low-API oil.

In black-oil models, it is particularly easy to model densities incorrectly because of the inadvertent use of “arbitrary” surface densities. The reservoir phase densities are not input directly in a black-oil model. Instead, they are calculated using the formation volume factors (PVF) \(B_w, B_o, B_{gr}\) and \(B_{gr}\) solution gas/oil and oil/gas ratios \(R_{gr}\) and \(r_o\) and surface densities \(\rho_o\) and \(\rho_w\), based on the relations \(\varphi_o = (\rho_o + \rho_w r_o) / B_w\) and \(\rho_w = \rho_w / B_o\).

Because reservoir densities are not usually tabulated as output by reservoir simulators, the user may not know how “wrong” the densities might be.

A well-tuned EOS using volume shift factors¹⁵₄ should always predict reservoir and surface densities within 2% to 4%, and often within 1 to 2%. The author’s experience has found it generally unnecessary to use different volume shift factors for reservoir and surface calculations unless all densities are needed for the grid. The use of either Peng-Robinson¹⁵₅ or Soave-Redlich-Kwong¹⁵₆ EOS with volume-shift factors provides densities as good or better than the more-complicated ZJKR¹⁵₇ EOS model; in this author’s opinion, the ZJKR no longer
has a role in reservoir simulation, as supported by decreasing industrial use (almost to nonexistence) during the past 5 years.

17.7.4 Component Partitioning. The partitioning of components in gas and oil phases, as dictated by $K$-values, is important for properly describing vaporization and condensation. Vaporization is the process of "stock-tank oil" (STO) components ($C_1^+$) moving from the oil phase to the gas phase, while condensation is the process of intermediate ($C_7^-$ to $C_{16}^-$) components moving from the gas phase to the oil phase. Retrograde condensation is particularly important in the depletion of gas condensate reservoirs, where in-situ retrograde condensation results in decreasing surface liquid yields.\cite{52,152,153}

For most gas-injection projects, vaporization will play an important role. Proper modeling of vaporization usually requires a compositional model that uses a detailed description of the heaviest ($C_{16}^-$) components, typically three to five fractions.\cite{22,159,160} The lightest fractions ($C_1$ to $C_4$) will typically vaporize 80 to 100%, while intermediate heavies ($C_7$ to $C_{16}$) may vaporize in varying amounts from 30 to 90%, and the heaviest components $C_{16}^-$, may vaporize from 0 to 50%. The degree of vaporization depends primarily on the local pressure and composition of displacing gas. \cite{144}

Cook et al.\cite{144} suggested a modification of the black-oil model that allows saturated properties to change as a function of how much injection gas has contacted a given cell, thereby allowing reasonable description of a vaporization-dominated process. Tang and Zick\cite{161} propose a limited three-component compositional model that allows accurate description of developed miscibility for use in models in which grid-related dispersion may result in underpredicted conditions of miscibility using an EOS-based simulator.

In some gas-injection processes, the injection gas may be enriched (1) during injection by adding gas to the reservoir (GTR), (2) within the reservoir by multiple contacts with reservoir oil, which gradually enriches the injection gas with intermediate components $C_7$ through $C_{16}$. For either type of enrichment, the vaporization process can become extremely efficient and result in a near-miscible or miscible displacement with near-100% recovery of the oil. When such a displacement results, it is often associated with the development of a complex process whereby the near-miscible front consists of upstream vaporization and downstream condensation—the "condensing/vaporizing" gas drive mechanism first described by Zick.\cite{162}

For an EOS model to properly describe complex phase behavior related to vaporization, condensation, and near-miscible mechanisms, special PVT experimental data should be measured and used to tune the EOS. For immiscible processes with significant vaporization, a single-cell multicontact PVT test is useful, quantifying the degree of vaporization in terms of oil volumetric stripping and gas compositional changes. For near-miscible or miscible processes, a swelling-type test which has the following features is strongly recommended: (1) five to seven mixtures of reservoir oil and injection gas are used, with two to three mixtures being dieselpoints and two to three mixtures being dewpoints; (2) a constant composition expansion is conducted on each mixture, where saturation pressure and the oil relative volumes are measured and reported; and (3) a single equilibrium "flash" somewhat near the critical point is used for obtaining a set of equilibrium gas and oil compositions ($K$-values).

Proper EOS tuning of the complex phase behavior measured in such an experiment is difficult, and requires the ability to match near-critical volumetric and compositional changes. Such predictions are almost never available a priori with an EOS model that has been tuned only to simple depletion data. Small but important modifications of the heavy-end properties and binary interaction parameters are usually required to obtain a satisfactory match of near-critical PVT data provided by this special "swelling" test. A tuned EOS model that is able to match all PVT data, including near-critical phase behavior, has a good chance of properly predicting true multicontact near-miscible/miscible behavior.\cite{152,162}

\textbf{17.7.5 Viscosity.} Gas viscosities are typically estimated by correlation\cite{163} within 5 to 10%, and they are almost never measured experimentally. Such accuracy is adequate for most applications, and gas viscosities seldom vary beyond the range of 0.02 to 0.03 cp. For gas-injection processes at high pressure or near-miscible displacements, gas viscosities can range from 0.03 to 0.1 cp. Gas-viscosity correlations are not usually accurate for this higher range of viscosity, and errors up to 20 to 30% may be expected; compositional viscosity correlations also have the same level of accuracy.

Oil viscosities are notoriously inaccurate based on correlations,\cite{164} at best being within 10 to 30%, but often in error by 50% or more. Oil viscosities should always be measured and used to tune a viscosity model. A minimum requirement would be measurement of stock-tank oil viscosities, and normally live-oil viscosities are available from a differential liberation test.

For gas condensates, oil viscosities are almost never measured. This may be a serious problem if condensate blockage has a significant impact on well deliverability.\cite{165} Condensate viscosities are difficult to measure because retrograde condensate volumes can be very small. The use of a separator condensate sample is recommended. Bring it to reservoir conditions ($T$ and $P$) for density and viscosity measurements, then tune the viscosity model to these data. This approach is feasible for any gas condensate, it is not expensive, and it is better than relying on untuned compositional viscosity correlations.

Our experience has been that the compositional viscosity correlation by Lorenz, Bray, and Clark\cite{166} (LBC) is not predictive, and it is highly dependent on accurate density predictions. Tuning the heavy-component critical volumes and, sometimes, careful modification of the higher-order constants in the LBC equation, provide the required accuracy for gas and oil viscosities most reservoir processes. Unfortunately, modification of the LBC correlation to match a limited number of viscosities (in a narrow range of pressure and composition typical of depletion experiments) can lead to unphysical viscosity predictions at conditions developed during a gas injection process.

The Pedersen et al.\cite{165} compositional viscosity correlation, though more complicated and CPU-intensive than the LBC correlation, has quite good predictive capabilities. The correlation is based on a corresponding states formulation in which methane viscosity is the reference compound.

17.7.6 Gas/Oil Interfacial Tension (IFT). Gas/oil IFTs may be used in a reservoir simulator\cite{167} to modify input capillary pressures $P_{g-o}$ relative permeabilities $k_{g-o}$ and $k_{o-g}$, and residual oil saturation $S_{o-r}$. As IFT decreases, $P_{g-o}$ decreases proportionately. Relative permeabilities and residual oil saturation change only marginally until IFT reaches a fairly low value (e.g., 1 mN/m), whereas at lower IFTs, the relative permeability-related properties change approximately as log(IFT).

Practically, IFT impact on capillary pressure is limited to fractured reservoirs\cite{168} in which displacement efficiencies may be strongly linked to the balance of gravity and capillary forces on a relatively small distance scale (e.g., block heights of 1 to 10 m). IFT impact on relative permeabilities and residual saturations may have an impact on some gas-injection processes, though near-miscible and miscible processes have minimal IFT dependence because they are usually dominated by the strong compositional effects (vaporization and condensation) that result in near-100% recoveries. In fact, displacements that are miscible should, by definition, be independent of the relative permeabilities and residual oil saturation. IFT impact on relative permeabilities in gas condensate reservoirs can, together with high velocities, result in large capillary numbers, which have the tendency to "straighten" relative permeabilities and improve flow performance. It has been shown\cite{169,170} in a number of recent publications that this effect can have an important impact on gas-condensate well deliv-
erabilities. Neglecting IFT and velocity dependencies of relative permeability can lead to overly conservative prediction of well deliverability (i.e., overprediction of condensate blockage).

17.7.7 Black-Oil PVT Models. Black-oil PVT properties are generated in one of two ways. For low- to medium-GOR oils (<150 Sm³/Sm³), a traditional differential liberation experiment (DLE) is used; with corrections for separator flash to calculate oil FVF, Bo, and solution GOR, as well as the gas GVF. This approach assumes the reservoir gas contains unsubstantial amounts of condensate in solution, with solution oil/gas ratio r₁ ~ 0.

The more common and general approach to generating black-oil PVT properties uses an EOS model to simulate a depletion-type PVT experiment (differential liberation, constant volume depletion, or constant composition expansion), with the equilibrium gas and equilibrium oil used in the depletion being individually processed to provide the four black-oil properties, Bₚ, Rₚ, Bₒ, and r₂. For highly volatile oils, the EOS method gives substantially different and improved black-oil properties compared with the traditional laboratory DLE/separatore-corrected approach.

The conversion of black-oil PVT data (Rₚ, Bₚ, rₚ, and Bₒ) to a compositional model uses K-values of surface gas and oil pseudo "components" to a compositional model uses K-values of surface gas and oil pseudo "components" Kₜ = (Rₚ + Cₒ) / R, and Kₘ = r(Rₚ - Cₒ) / (1 + rCₒ), with Cₒ = (RTₒ / pₒ) / Mₒ. The reservoir-phase densities are calculated from ρₜ = (pₒ + pₛ) / Bₜ, and ρₚ = (pₒ + Rₛ pₛ) / Bₚ, while phase molecular weights are given by Mₛ = (Mₒ + Rₛ Mₛ) / (1 + rCₒ) and Mₚ = (Rₛ Mₛ + Mₚ Cₛ) / (Rₛ + Cₛ). Viscosities and gas/oil IFTs are interpolated directly from input tables.

Coats et al., Coats et al., and Fevang et al. have shown that black-oil models can be used for practically any type of reservoir produced by depletion or waterflooding, including reservoirs with large compositional gradients. Some issues require special treatment for complex fluids systems, including fluid initialization and the method for generating black-oil tables. In the recommended procedures, the black-oil tables are generated using the black-oil EOS model using the fluids with the highest saturation pressure (e.g., at the gas/oil contact) and to initialize with solution GOR (Rₛ and rₛ) vs. depth—not saturation pressure vs. depth.

A common problem with black-oil models is the calculation of "negative compressibility," meaning that a small pressure drop results in a reduction in total gas (oil + volume). Another problem is physical extrapolation of saturated PVT properties to saturation pressures higher than given in the input tables (e.g., caused by gas injection, gravity segregation in undersaturated layers, or field behavior during rate reductions).

When can the black-oil PVT treatment not be used? Basically, for any gas-injection process with significant component partitioning that changes during the displacement. This could include processes with high-pressure vaporization using lean gas, condensation from enriched injection gas, and developed-miscibility processes such as the condensing/vaporizing mechanism. Surprisingly, a black-oil treatment is sometimes adequate even for complex gas injection problems, though it is not usually known a priori. To check the validity of a black-oil model, a black-oil model in a gas injection project, the reservoir model should first be simulated with a compositional model, and preferably a relevant 3D model that captures all important flow characteristics.

17.7.8 Equation-of-State Models. The most common EOS used in reservoir simulation are the PR and the SRK models. Both models have two constants, a and b. Each constant must be calculated for each component based on component critical properties (Tᵣ and pᵣ) and acentric factor (ω).

The PR EOS has two versions—the original 1976 version and the modified 1979 version; the latter uses a third-degree polynomial expression for the correction term to constant a. For some systems, the difference in equilibrium calculations for the two PR EOS models is significant.

The Penleoux volume shift factors should always be used with two-constant EOS models to provide oil and gas densities. The volume shift factors have no impact on calculated K-values or compressibility, only densities. As mentioned earlier, the ZIRK EOS is outdated and was before the volume-shift method was introduced in 1980, with complex correction functions to constants a and b to improve liquid density predictions.

Binary interaction parameters (BIPs) kᵣ are important for adjusting predictions of equilibrium properties (K-values and compositions). These parameters represent a correction of the form (1 - kᵣ) to the aᵣᵣ term in the quadratic mixing rule for EOS constant a. BIPs can be positive or negative; they are symmetric (kᵣ = kᵣᵣ), they are usually ~0 for most hydrocarbon-hydrocarbon pairs, except C₅ to C₈ pairs which may reach values as high as 0.25, and they are generally close to 0.1 for nonhydrocarbon (N₂, CO₂, H₂S)-hydrocarbon pairs.

17.7.9 Three-Phase PVT Behavior. Three-phase (L₁-L₂-V) behavior is an occasional but serious problem for EOS-based compositional models. The third phase (L₂) is usually a light liquid and typically appears at low temperatures (<140°F) in gas-injection processes using CO₂ or NGL-enriched gas. Physically, three phases may actually exist, and the EOS model is correctly predicting the behavior. Sometimes a three-phase system is predicted without any physically existing; this may result for lower temperatures when the heaviest component properties are improperly modified to fit two-phase gas/oil PVT data.

For reservoir simulators, the three-phase problem is caused by the EOS formulation "allowing" only two phases. If three phases actually exist, the two-phase flash may find any of the three possible two-phase combinations: L₁-V, L₂-V, or L₁-L₂. These false two-phase solutions may indicate a single-phase condition, or they may actually appear as a valid solution (meeting the equilibrium constraint). Unfortunately, the reservoir model, in a given cell during a given timestep, may flip-flop between two of the possible solutions, resulting in unstable behavior because the pressure solution is not continuous from one two-phase solution to the other. Models may have to simply give up because of repeated timestep reductions, which result from the inadequacy of the EOS two-phase model handling a three-phase condition.

17.7.10 Surface Phase Behavior. In compositional simulation, the surface calculations are usually made using multistage separation with an EOS model, with fixed K-values for each separator, or using so-called "gas plant" factors, which define the fraction of a wellstream component that is processed into the stock-tank oil. For black-oil models, the surface separation is "built in" to the PVT tables. Consequently, if the surface process changes during a model run, all black-oil PVT tables must be reentered at the appropriate time. This also requires that vertical flow performance (VFP) tables be reentered because surface rate and volume ratio nodes change with the process change.

It is difficult to use traditional black-oil models for fields with various well groups that have significantly different processing facilities. Additionally, thermal PVT requirements for thermal processes such as steamfloods include quantifying the temperature dependence of K-values, densities, and viscosities. Water-phase behavior of liquid and steam must also be defined in terms of pressure and temperature. Water-hydrocarbon phase behavior is still assumed to be simple, without water/hydrocarbon component partitioning.

An EOS model can be tuned to distillation data for improving the predictive capabilities of K-value dependence; otherwise, a simple correlation of the form Kₜ = a exp(-b/T) / p may be used for distillate components. Using distillation data is an indirect approach to defining K-value behavior, and it is useful in lieu of high-temperature gas/oil/water phase-behavior experiments, which are not usually available. Oil viscosities in thermal processes may be difficult to correlate with a compositional correlation, so empirical correlations may be used instead.
17.7.12 Fluid Initialization. As with rock and other petrophysical properties such as permeability and porosity, a reservoir simulator model must also initialize the spatial distribution of initial fluids. For an EOS-based model, the initial molar compositions are defined, \(z_i(x,y,z)\). For a black-oil model, the initial solution gas-oil ratio \(R_s\) and solution oil-gas ratio \(R_o\) ratio are defined, \(R_s(x,y,z)\) and \(R_o(x,y,z)\); sometimes saturation pressures are used instead, \(P_i(x,y,z)\) and \(P_s(x,y,z)\), but this is not recommended. Specifying a saturated gas-oil contact (GOC) is also a means to initialize fluids vertically in a reservoir simulator, where solution GOR \(R_o\) (and bubblepoint) are assumed constant below the GOC, while solution GOR \(R_o\) decreases upwards from the GOC to honor the model-imposed assumption that reservoir pressure equals dewpoint pressure, \(p(x,y,z)\).

Another fluid initialization data might include temperature \(T(x,y,z)\). Some black-oil models allow spatial variation of stock-tank oil density, \(\gamma_{AP}(x,y,z)\), where black-oil properties are correlated in multidimensional tables as a function of pressure and \(\gamma_{AP}\). Across barriers such as faults and sealing shales, discrete PVT model data may be defined, such as EOS parameters or black-oil tables; such “discontinuous” fluid descriptions may cause physical and model incompatibilities if fluids mix in the reservoir or at the surface.

A typical problem with initialization is that the specified fluid distribution, initial pressure specifications, and fluid contacts lead to fluid movement when the model is turned on (without production or injection). Initial fluid movement may be unimportant without significantly changing the user-specified fluid system; serious inconsistencies may lead to time-zero flow that has an impact on model performance.

17.8 High-Performance Computing and Reservoir Simulation—John E. Killough

The motivation for high-performance computing in reservoir simulation has always existed. From the earliest simulation models, computing resources have been severely taxed simply because the level of complexity desired by the engineer almost always exceeded the speed and memory capacity of the high-speed vector processors such as the Cray supercomputer. The earlier high-performance computing hardware known as vector processors for the speedup. More recently, parallel computers based on connecting tens of hundreds of processors have formed the basis of high-performance computers. The importance of the serial or single-processor computer cannot be overemphasized; however, the introduction of the reduced instruction set computer (RISC) with “superscalar” performance in 1990 spelled the end of the dominance of the Cray supercomputer in the high-performance reservoir-simulation market because of a rapid decrease in the price/performance of the new hardware. Similarly, the recent introduction of low-cost, commodity hardware based on the INTEL chipset has led to significant improvements in price/performance. The application of this hardware to reservoir simulation has been a natural evolution in the application of the lowest-cost, highest-performance computers to reservoir simulation.

17.8.2 Characteristics of High-Performance Computing. The basis of high-performance computing is the use of specialized hardware to achieve computational speeds that are much faster than conventional computers. This idea of “speedup” is encapsulated in what is known as Amdahl’s Law:

\[
\text{Speedup} = \frac{1}{(\text{scalar} + \text{special}/n)}
\]

where “scalar” is the fraction of the program which is scalar and “special” is the fraction of computations performed on specialized hardware (one minus scalar). Simply stated, Amdahl’s law indicates that if there is a piece of specialized hardware that is \(n\) times faster than the scalar processor(s) in a computer or group of computers, it must be used a large fraction of the time to have a significant impact on overall performance. The earlier supercomputers primarily used specialized hardware known as vector processors for the speedup. More recently, parallel computers based on connecting tens to hundreds of processors have formed the basis of high-performance computers. The importance of the serial or single-processor computer cannot be overemphasized; however, the introduction of the reduced instruction set computer (RISC) with “superscalar” performance in 1990 spelled the end of the dominance of the Cray supercomputer in the high-performance reservoir-simulation market because of a rapid decrease in the price/performance of the new hardware. Similarly, the recent introduction of low-cost, commodity hardware based on the INTEL chipset has led to significant improvements in price/performance. The application of this hardware to reservoir simulation has been a natural evolution in the application of the lowest-cost, highest-performance computers to reservoir simulation.

17.8.3 The Parallel Reservoir Simulator. The earlier high-performance computing hardware such as the Cray Computer required that modifications be made to the reservoir simulator programs to efficiently use the hardware. The modifications known as vectorization led to different approaches in the data organization. One of the best examples of this is given in Refs. 176 and 177, in which a compositional model’s data structure is reorganized so that like phases in the finite-difference gridblocks of the reservoir are grouped. The notion behind parallel reservoir simulation is similar; however, in this case, the reservoir grid is subdivided into 3D blocks or domains. The computational work for each of the domains is then performed in parallel by a separate CPU of one of the computers in the parallel hardware. Profiles of computing workload for a typical reservoir simulation model often show tens, if not hundreds, of subroutines that are involved in a substantial portion of the calculations. Because of this, major reprogramming of reservoir simulation models is required to achieve high parallel efficiencies. As Ref. 178 points out, there are numerous obstacles to parallelization for reservoir simulation: recursive nature of the linear equation solver, load imbalance caused by reservoir heterogeneity and/or physics, and diverse data structure of well and facility management routines.

Several papers 179–189 discuss techniques that have been used to bring about efficient parallel reservoir simulations. Each of these addresses the solutions to parallelization and, in particular, the obstacles mentioned previously in various ways. One of these simulators uses local grid refinement for parallelization. The main concept of parallelization with local grid refinement (LGR) is that LGR is not necessarily used to add additional cells to the model but simply to facilitate the subdivision of the grid. With LGR, the same simulation program can be used to perform simulations either serially on a single processor or in parallel on multiple processors.
simply through data manipulation. In addition, a great deal of flexibility for domain decomposition exists, which can lead to enhanced load balance for parallel simulations. Parallelization using LGR involves assigning each grid to a processor. The same processor can be assigned to many grids or in the limit; each grid can be assigned to a separate processor. Variation of processor assignment and the grid refinement can dramatically affect parallel performance. Different preconditioners for the parallel linear equation solvers can also have a dramatic effect on parallel efficiency. Finally, the flexibility of LGR and processor assignment can be used to achieve improved parallel efficiency. For parallel domain decomposition, the base grid may be totally refined into subgrids so that all gridblocks in the base grid become inactive. Each grid is assigned to a processor so that only a portion of the model resides on a given processor. In this manner, the simulator memory requirements become scalable. That is, as the number of processors is increased, the grid assigned to a given processor becomes smaller. Alternatively, only a portion of the base or root grid may be refined; the remaining unrefined cells in the root grid are then assigned to a processor. Because of this use of local grid refinement, only an extremely small number of global arrays are required on each processor—for example, the pore-volume array. The existence of horizontal wells in most modern simulations requires that no restriction be placed on the placement of wells relative to grid and/or processor boundaries in a parallel simulator. To alleviate any difficulty with this, one approach is for well information to be broadcast among all processors. The overhead for broadcasting completion level information for all wells is included in the following results and appears to have little overall effect on parallel efficiency. Tubing hydraulics calculations for a predictive well-management routine have also been parallelized. These calculations are computationally intensive enough to support gather/scatter-type operations. For example, in a case with 1,500 producing wells, more than 90% of the computational workload for well management was in the tubing hydraulics calculations. Parallelization of only these calculations has shown a substantial improvement in parallel efficiency for several test cases for the overall well-management routines.

17.8.4 The Parallel Linear-Equation Solver. A typical parallel linear solver involves composite grid decomposition and is based on the work of Wallis and Nolen. At each level of the composite grid, a red-black or block Jacobi ordering of the grid is used. Red-black refers to an ordering of the grids or domains. For example, if the grids form an equal areal subdivision of an areally square model, the red and black grids would appear as a checkerboard when viewed from the top of the grid. For a 1D or strip decomposition of the model, the red-black coloring would appear as alternately colored stripes when viewed perpendicular to the decomposition direction. An approximate block LU factorization approach is used for preconditioning and involves, for example, an L-factor for the red-black ordering as follows:

\[
L = \begin{bmatrix}
A_1 & A_2 \\
A_3 & A_4 \\
Q_1 & R_2 & B_1 \\
Q_3 & R_4 & B_2 & B_3
\end{bmatrix}
\] (17.35)

The \( A_i \) blocks are ordered first and represent the red grids or domains, while the \( B_i \) blocks are the black grids and are ordered second. The diagonals of the \( B_i \) submatrices are modified from the original coefficients to maintain rowsum constraints (see Ref. 190). This approximate factorization leads to independent solutions on each processor for the first the red \( A_i \) and then followed by the black \( B_i \) grid matrices. For the block Jacobi approach, only a single color exists, and all grids at a given level are solved independently. As described in Ref. 179, parallelization of this preconditioning can be achieved in a manner almost identical to the technique used for the coefficient matrices. Ref. 47 provides detail of a similar parallel solver developed independently, but almost simultaneously, by Burrows, Ponting, and Wood. As described previously, each grid is assigned to a processor. These may be the same or different processors. If all active grids lie on the same level, the composite grid solver performs all red or black preconditioning solutions independently (i.e., either block Jacobi (all red) or block red-black orderings are used). If grids are assigned to different levels, each grid of the same level and same color (Jacobi being all red) is solved simultaneously, with communication being performed after the entire solution of all grids on a level is completed. Factorization is performed so that rowsum constraints are also imposed. The flexibility of this solver allows the easy implementation of block red-black algorithms, or, as a special case, a red-black linear decomposition ("strip") decomposition could be used.

17.8.5 Parallel Computers and Message-Passing Interface. With the rapid changes in technology for parallel computing, it is difficult to provide the state-of-the-art because this appears to change daily; however, current classification practice divides parallel computers into two distinct categories: shared and distributed memory. Shared-memory systems allow all of the memory associated with the parallel computer to be accessed by all of the processors. For UNIX systems, the memory sizes currently can range up to several tens of billion of words with hundreds of processors. For systems based on PC products, shared-memory systems usually are limited to a few billion bytes of storage and fewer than 10 processors. (This may change with the recently introduced 64-bit processor Itanium.) The distributed memory systems allow access to local memory only by a small number of processors that reside on a "node" of the parallel system. Each of the "nodes" in the distributed memory parallel system often communicates with other nodes by a high-speed switch, although conventional Ethernet communications are possible for some applications. Either of these parallel computers is capable of delivering several hundred million floating point operations per second (MFLOPS) for each node, or several billions of floating point operations per second (GFLOPS) for applications that can use the parallel hardware. Parallel PC hardware can be further subdivided into operating systems: Windows- and UNIX-based systems. Current trends indicated that it is likely that the Windows-based systems will dominate the engineer's desktop because of the wide availability of engineering/business applications available for this operating system. UNIX-based systems will likely be limited to clusters of servers to perform computer-intensive work such as parallel simulations.

To perform calculations in parallel, data must be passed in the form of messages among processors of the parallel machine. Message-passing interface (MPI) has been primarily used for parallelization on all classifications of parallel machines because of the wide availability and portability of MPI. For shared-memory computers (Sun and SGI for example), other forms of message passing exist such as the recently popular OpenMP. There are generally three forms of message passing for a distributed memory system: synchronous, asynchronous, and global operations. Synchronous message passing requires that all messages be completed before computations can proceed. Asynchronous, on the other hand, allows some overlap of processing and communications. Global message-passing operations distribute data from single processors to one processor, or vice-versa. Often some form of arithmetic function, such as
It is likely that the introduction of unstructured grids and associated load balancing using technology effort will be required, especially if more than a few processors are to be used efficiently. Techniques such as "Hilbert space-filling curves" may well speedup a factor of 2 for slightly more than doubling the number of processors. The scaling falls off slightly from 27 to 54 processors with only a factor of 1.7, but from 54 to 108 processors, this scaling is maintained at a factor of 1.7. Overall, the speedup from 12 to 108 processors was a very good factor of 5.6. If perfect scaling on 12 processors is assumed, this indicates a factor of 68 overall for 107 processors for a parallel efficiency of approximately 64%. The total elapsed time for the 108 processor case was approximately 12 hours, indicating that overnight turnaround for this 3.4-million-cell model could be achieved. Another example application of parallel simulation with more than 1 million gridblocks was recently presented in a black-oil simulation study of complex water encroachment in a large carbonate reservoir in Saudi Arabia.

The Future of High-Performance Reservoir Simulation. It is clear that the uptake of parallel reservoir simulation has been limited by the hurdle of cost and the robustness of the technology. The recent advances in commodity hardware based on the INTEL architecture have provided a significant boost, however. Currently, the use of parallel INTEL-based clusters using inexpensive high-speed switches has lowered the entry point for significant parallel simulation by an order of magnitude. Just as the UNIX workstation caused a revolution in reservoir simulation, this new low-cost hardware will likely bring parallel computing to the engineer's desktop in the near future. The remaining hurdles for parallel simulation are thought to be much lower than for enhancing parallel simulation technology. In particular, emphasis must be placed on load balancing and efficient linear equation solvers. Load balancing refers to the fact that all processors must perform the same amount of work to achieve high parallel efficiency. If only a few processors perform most of the work for a parallel simulation involving large numbers of processors, then poor parallel efficiency results—a case known as load imbalance. The allocation of optimization techniques to solve the load balancing problem offers some promise. An area of particular importance for load balancing is the coupled surface network/reservoir simulator. For this case, the network often dominates the simulation by an almost an order of magnitude; therefore, to achieve reasonable parallel efficiency, the imbalance caused by the serial surface network calculations must be overcome. One approach to this is mentioned previously, the parallelization of the tubing hydraulic calculations; however, considerable additional parallelization effort will be required, especially if more than a few processors are to be used efficiently. Load imbalance is also brought about by complex geologies and complex locally refined grids. It is likely that the introduction of unstructured grids and associated load balancing using techniques such as space-filling curves may well lead to the solution of this problem. Finally, the linear equation solver as the heart of the parallel simulator must be enhanced to provide efficient and robust solutions. Solver degradation often results from the use of large numbers of domains (processors) or a poor choice of the decomposition geometry. Although the previous examples show good parallel performance, future models with larger, implicit systems must be routinely solved on large numbers of processors. Multigrid-like solvers may show the best promise for this improvement.

17.9 Reservoir Simulation Applications—L. Kent Thomas

Reservoir simulation is a widely used tool for making decisions on the development of new fields, the location of infill wells, and the implementation of enhanced recovery projects. It is the focal point of an integrated effort of geosciences, petrophysics, reservoir, production and facilities engineering, computer science, and economics. Geoscientists using seismic, well-log, outcrop analog data and mathematical models are able to develop geological models containing millions of cells. These models characterize complex geological features including faults, pinchouts, shales, and channels. Simulation of the reservoir at the fine geologic scale, however, is usually not undertaken except in limited cases. Generally, the fine-scaled geological model is partially integrated or "upscaled" to a coarse-grid model, which is computationally more tractable. The grid of the upscaled model is designed to capture the principal geological features of the geologic model to simulate the fluid flow. The grid may also be designed to capture the effects of complex wells. In the upsampling process, the laboratory relative-permeability and capillary-pressure functions may be upscaled to "pseudofunctions." These pseudofunctions attempt to capture fluid-flow behavior that is lost because of the integration of fine-scale geological features. The upsampling process often requires significant input from experts in other fields, such as reservoir engineering, production engineering, and economics.

The reservoir simulation model may either be used directly to forecast the performance of a new reservoir or adjusted so that it reasonably models the historical behavior of an existing reservoir and wells. This adjustment process is called history matching. Programs called "pre-processors" and "post-processors" enable the engineer to prepare data, manipulate the model, and view results. Once a history-matched model is obtained, then forecasts are made under a variety of operating conditions. These results are combined together with economic models to enable the engineer to make decisions concerning the operation of the reservoir.

17.9.1 Development of the Geological Model. A sound understanding of the structural elements of the reservoir and the depositional environment under which the sediments were deposited is critical to the development of an accurate geologic model. Today, the geologic model is frequently constructed of the reservoir and adjacent aquifer and is referred to as a static, or geocellular, model. This model provides the vehicle to capture and combine the seismic structural interpretation and well petrophysical data in a numerically consistent way with known depositional characteristics. Petrophysical properties such as porosity, permeability, and water saturation can be distributed throughout the interwell 3D volume using various techniques, many of which rely on geostatistics. Efforts are also underway to condition these numerical, static models with production data to further reduce geologic uncertainty. The construction of a geocellular model represents a significant collaborative effort between geoscientists, petrophysicists, and reservoir engineers.

Geocellular models today may consist of over 25 to 50 million cells on large and/or geocellular complex reservoirs. The ability to build static geologic models of this magnitude has outstripped the reservoir engineer’s ability to simulate an equal number of cells in a full physics reservoir simulator (and will continue to do so). Classical development geologic efforts have focused on defining and describing the reservoir geology using 2D maps, which depict the most likely interpretation of the depositional environment and the variability of the reservoir parameters between wells. These interpretations have historically been referred to as
“deterministic” reservoir descriptions. With the advent of geocellular models and the application of such technologies as geostatistics, it is now possible for geoscientists to generate multiple reservoir descriptions for the reservoir engineer to simulate. In some cases, one of these descriptions may be selected to represent the “deterministic” model. Regardless if one or several static models are handed over for reservoir simulation, it is generally necessary to reduce the cell count to run the problem with existing reservoir simulators. Significant effort is being spent improving techniques to reduce the number of reservoir cells in the areal and vertical dimension while maintaining the essential geologic character that impacts the recovery process under consideration. This approach is referred to as upscaling, and it will be discussed in greater detail in the following section. To date, the largest reservoir simulators consist of reservoir descriptions of 2 million grid cells and are run using massively parallel processing power.

### 17.9.2 Upscaling Geological Model to Reservoir Flow Model

Geological models, which contain the complex structural features of large oil and gas reservoirs, commonly have tens of millions of cells. These models, which contain pinchouts, faults, and other significant information including lithology and facies distributions, are upscaled in both the vertical and areal directions to tens or hundreds of thousands of cells for reservoir simulation. Several upscaling methods have been developed over the last several years including analytical techniques, local flow-based methods, and global flow-based methods. Analytical methods use geometric, harmonic, power law, and geometric averaging to calculate effective properties for each reservoir model gridblock. Local flow-based methods calculate effective gridblock properties by performing single-phase flow simulations in each direction across the upscaled block. The diagonal permeability tensor is calculated by sealing the boundaries perpendicular to the applied pressure gradient. The full-permeability tensor can be calculated in a similar manner by leaving the boundaries normal to the imposed pressure gradient open and applying periodic boundary conditions. Global flow-based methods use pressure gradients across the entire field subject to a specific set of wells to calculate the permeability tensor. Local and global flow-based techniques can be used to compute upscaled transmissibilities directly.

### 17.9.3 Inclusion of Faults in Reservoir Flow Model

Faults and pinchouts of geological layers are incorporated in geological models to capture the complex geometry of many reservoirs. This information is then upscaled into the reservoir model, and it results in both neighbor and non-neighbor connections across the faults and non-neighbor connections across the pinchouts. In Cartesian coordinates, the trace of a fault may need to be represented by a “stair-stepped” line, while a somewhat better representation of faults can be made with corner-point grids. PEBI grids, which will be discussed subsequently, are best suited to accurately model fault geometry.

Models for calculating the transmissibility across the fault and parallel to the fault have been developed based on fault type, displacement, geochemical deposition, and whether open joints occur along the fault. In general, transmissibilities across a fault can be at least an order of magnitude lower than those parallel to the fault. Inclusion of this information in a reservoir model is frequently a key parameter in reservoir description.

A recent paper describes the analysis that was performed to calculate fluid flow through conductive faults in the Khafji oil field in the Arabian Gulf. Two sandstone reservoirs separated by a thick continuous shale are both connected to the same large aquifer and had the same initial WOC. The top reservoir has edgewater drive, while the deeper reservoir is bottomwater drive. Early water breakthrough in the upper sand was determined to be a function of supplemental water influx from the aquifer of the lower sand through conductive faults.

### 17.9.4 Development of Pseudofunctions for Multiphase Flow

Pseudorelative permeability curves are developed for upscaled reservoir models to match multiphase fluid flow at the fine-grid level. Several methods for performing these calculations have been presented in the literature. In the “10th SPE Comparative Solution Project: A Comparison of Upscaling Techniques,” the fine-scale geological model was chosen to be sufficiently detailed such that it would be difficult to run the fine grid and use classical pseudosimulation methods. Several participants, however, used some level of fine-grid simulation to develop pseudorelative permeability curves, with two of the participants adjusting the exponents of the Corey equations to effect a reasonable match. This approach can be done manually or with an automated history-matching algorithm.

### 17.9.5 Gridding Techniques

The majority of reservoir simulation studies conducted today use Cartesian or corner-point structured grids with some application of local grid refinement to evaluate infill well locations or to more accurately calculate water and/or gas coning in a well. In a structured grid, cell locations are specified using a 3D, i, j, k indexing system. This allows for ready access either numerically or visually, using pre- and post-processing software, to multilayer well information or data and calculated results at any physical location in the reservoir model.

A more flexible approach for modeling reservoirs with complex geometries that still relies on structured gridding was presented by Jenny et al. Here, a hexahedral multiblock grid is used which is unstructured globally, but maintains an i, j, k structure on each subgrid. PEBI grids are now being used on a limited basis to simulate reservoirs with complex geological features that have been developed with nonconventional wells to maximize recovery. These grids are unstructured and are described internally in a simulator with a 1D index, i, that ranges from one to the number of nonzero pore volume cells in a model. Evaluation of simulator input and results relies heavily on pre- and post-processing software that allows the user to visually look at the model and make changes during the history-matching phase of a study.

### 17.9.6 Simulation of Nonconventional and Intelligent Wells

Nonconventional wells are routinely used to maximize production rates and ultimate recovery in oil and gas reservoirs. Wells in this category include deviated, horizontal, horizontal and vertical multilaterals, and multilateral fishbone designs. This latter well type is especially effective in low-permeability or heavy-oil reservoirs.

Simulation of nonconventional wells can be approached in several ways. First, the productivity of each perforation in a conventional model can be approximated by applying the appropriate skin and Peaceman’s equation. Second, simulation grids can be constructed that closely follow the well path and allow a more accurate calculation of well rates. Another approach, which is quite appealing, is based on a semi-analytical method. It results in a good approximation for PI's in nonconventional wells and incorporates the near-wellbore skin because of heterogeneity in this region as well as mechanical skin. This method, which is very efficient, can also include wellbore hydraulic effects.

Nonconventional wells coupled with intelligent completions can be used to improve sweep efficiency and optimize recovery. One example of this technology is the use of surface-controlled, downhole-adjustable chokes, which can be used to apply different pressure drawdowns to separate zones along the well. This allows a more uniform inflow in the well and control of early water or gas breakthrough. Real-time measurements of wellbore pressures and temperatures are being made for use in conjunction with PLT tests for inflow performance analysis.

### 17.9.7 Integrated Reservoir and Surface Facilities Models

Integration of reservoir and surface facilities simulation can result in improved production forecasts and allows optimization of...
the surface facilities structure and operating conditions. An integrated reservoir, well flow
string, and surface network model of the Prudhoe Bay oil field was built and successfully ap-
ploled to a facility optimization study.\textsuperscript{39} Production costs as a result of this effort were reduced
by defining the optimum number of separator stages and their connections, and defining the
 optimum separator operating conditions and by using excess capacity in the Prudhoe Bay facili-
ties to process production from satellite fields. Procedures for the simultaneous solution of the
reservoir and surface pipeline network flow equations are described in Refs. 210 and 211.

In the Ekofisk field in the Norwegian sector of the North Sea, integrated reservoir and facili-
ties simulations have been made to optimize throughput in existing surface facilities and to
forecast production from planned expansion of current facilities.\textsuperscript{212} This optimization project
has resulted in sustained high production of approximately 300,000 STB/D over the last several
years. Another important aspect in the management of this field is the inclusion of compaction
logic in the model based on both stress and water saturation changes during depletion and wa-
terflood.\textsuperscript{213} Treatment of geomechanical effects in stress-sensitive reservoirs has received
increased attention throughout the industry in recent years.

17.9.8 Simulation of Multiple Reservoirs. Simulation of multiple fields producing into a com-
mon production facility is routinely practiced today to capture the interplay between well
deliverability, water and gas injection, operating constraints, and contract rates. In the J-Block
area fields, in the U.K. sector of the North Sea, an integrated reservoir study was conducted
that included a gas condensate reservoir with gas cycling that was simultaneously modeled
with volatile oil reservoirs.\textsuperscript{214} The fields were developed with a single platform and one subsea
manifold completion and a combination of vertical and horizontal wells. Four separate PVT
regions were used to describe the fluid behavior. The integrated model used in this study re-

sults in an efficient reservoir management tool for making development and operating decisions.

Another example of reservoir management of multiple fields with shared facilities is the
Gannet cluster, located in the U.K. sector of the North Sea, which connects four fields.\textsuperscript{215}
Wells from one of the fields are directly linked to the production platform, and the other three
fields are subsea tiebacks to the platform. Three of the four fields are oil fields and the fourth
is a gas field. An integrated model was built to simulate the interaction of the subsurface and
surface processes. The well-management objective on this project was to maximize hydrocar-
on recovery while simultaneously meeting a long-term gas contract.

17.9.9 Use of Larger Models. The maximum practical model size has increased from tens of
thousands to hundreds of thousands of cells at essentially a linear rate vs. time during the last
decade. This trend has developed as a result of the dramatic increase in computer hardware
speed accompanied with larger memory and cache. Both high-speed UNIX workstations and
high-end PCs are used for reservoir simulation, with a close race developing between the two
platforms in regard to run times. Additional advances in computing speed for megamodels
have been achieved using parallel hardware along with the necessary developments in model
software, discussed in a previous section. An example application of this technology was recent-
ly presented in a simulation study of complex water encroachment in a large carbonate
reservoir in Saudi Arabia.\textsuperscript{189}

17.9.10 History Matching and Production Forecasting. Once a reservoir simulation model
has been constructed, the validity of the model is examined by using it to simulate the perfor-
mance of a field under past operating conditions. This is usually done by specifying historical
controlling rates, such as oil rate in an oil reservoir vs. time, and then making a comparison of the
nonspecified performance such as GOR, WOR, and reservoir pressure with measured data.
If there are significant differences between the calculated performance and the known perfor-
mane of the well/reservoir system, then adjustments to the reservoir simulation model are
made to reduce this difference. This process is called history matching. These adjustments
should be made in a geologically consistent manner.\textsuperscript{188} Modification of those parameters that
have the highest degree of uncertainty will give the maximum reduction in the error. The history-
matching process should be approached in a consistent manner to minimize the effort.\textsuperscript{216} In
addition to well rates and bottomhole pressures, and reservoir pressures measured at the time
the well is drilled, production logs, long-term pressure gauges, and time-lapse seismic data enable
the engineer to better constrain the model during the history-matching process. Time-lapse
(4D) seismic\textsuperscript{217,218} is becoming an integral part of the field performance monitoring and history
matching. Streamline models together with reservoir simulators\textsuperscript{19,18} can be used to improve
the history-matching process, especially in waterflood operations. Tools to assist in the history-
matching process consist of the use of parallel computers, sensitivity analysis, and gradient


\textbf{Nomenclature}
\begin{itemize}
  \item $A$ = a matrix
  \item $B$ = formation volume factor, reservoir volume/surface volume
  \item $b$ = right-side vector
  \item $C_{gas}$ = equivalent surface gas volume for a unit volume of surface oil
  \item $d$ = dimension in direction of flow
  \item $D$ = depth below datum, ft
  \item $d_{f}$ = fractional flow
  \item $f_{i}$ = fugacity of component i
  \item $F$ = flow rate, volume/time unit
  \item $F_{i}$ = see Eq. 17.2
  \item $f_{g}$ = gas-phase fractional flow, $\phi_{g}$ / ($\phi_{g} + \phi_{w}$)
  \item $g$ = phase velocity
  \item $K$ = permeability, md
  \item $K_{e}$ = relative permeability, fraction
  \item $k_{o}$ = formation volume factor, reservoir volume/surface volume
  \item $L$ = length, feet
  \item $L_{1}$ = liquid type 1
  \item $L_{2}$ = liquid type 2
\end{itemize}
References


Chapter 17—Reservoir Simulation


SI Metric Conversion Factors

<table>
<thead>
<tr>
<th>Unit</th>
<th>Conversion Factor</th>
<th>Conversion Factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>bbl</td>
<td>1.589 873 m³</td>
<td>1.1057 75 L</td>
</tr>
<tr>
<td>cp</td>
<td>1.0*</td>
<td>1.0*</td>
</tr>
<tr>
<td>dyne</td>
<td>1.0*</td>
<td>1.0*</td>
</tr>
<tr>
<td>ft</td>
<td>3.048 39*</td>
<td>0.304 80*</td>
</tr>
</tbody>
</table>
Chapter 18
Estimation of Primary Reserves of Crude Oil, Natural Gas, and Condensate
Ron Harrell, SPE, Ryder Scott Co. and Chap Cronquist, SPE, Consultant

18.1 Introduction
Most exploration and production companies' assets consist of the resources and reserves owned by that entity. Resources typically are classified as either contingent (discovered but presently uneconomic) or undiscovered, but their quantification is outside the scope of this work. This chapter discusses primary reserves, which are discovered quantities of hydrocarbons that can be produced at a profit and are classified by level of uncertainty. The discussion of estimated reserves in this chapter excludes limitations imposed by the terms of ownership.

Since publication of the first two petroleum handbooks,\(^1,2\) the international petroleum industry has increased its understanding of the need to develop more reliable estimates of petroleum reserves and to quantify the uncertainty associated with the classifications of reserves. Furthermore, the regulatory authorities of many countries, particularly of the U.S.A., have found it necessary to accommodate the increasing internationalization of the industry and to manage their increasing involvement and influence in the industry.

With this global expansion of the petroleum industry has come ongoing technological development that provides better tools and techniques for analyzing reservoirs and reservoir fluids as well as greater understanding of how reservoir geology affects reservoir performance. In developing mathematical models that replicate the geologic environment, the reservoir engineer must incorporate all available hard technical data and work closely with and use fully the experience of multidisciplinary teams of geophysicists, geologists, petrophysicists, and other specialists. The engineer must develop a working knowledge of the skills contributed by each member of the reserves assessment team and apply the professional judgment of each team member to the estimation and classification of reserves.

The estimation of reserves is more than just a periodic, statutory calculating and reporting of company assets (although those are important functions); it is an essential element of investment planning and resource management for every prudent operator. Estimating reserves begins with identifying a drillable prospect, and it continues while the prospect is developed and placed on production, and thereafter as warranted by well and/or reservoir performance, new geologic data, competitor (offset) operations, unitization, contract renegotiation, improved technology, and/or changing economic conditions.\(^3\)