CHAPTER 16

Domain Decomposition Techniques for Efficient Adaptive Local Grid Refinement

Richard E. Ewing*

ABSTRACT

In the multidimensional numerical simulation of certain multiphase fluid flow processes, many phenomena are sufficiently localized and transient that self-adaptive local grid refinement techniques are necessary to resolve the local physical behavior. For large-scale simulation problems, efficiency is the key to the choice of specific adaptive strategies. Purely local refinement techniques require complex data tree structures and associated specialized solution techniques. Although these tree structures of changing length are amenable to parallel computation, they are very difficult to use efficiently in a vector mode. Techniques which involve a relatively coarse macro-mesh with potential local refinement in each separate mesh will be discussed. The macro-mesh will be the basis for domain decomposition techniques and parallel solution algorithms. Uniform meshing in the subdomains will allow efficient vectorization as well as parallelization of the algorithms. Similarly, different solution processes can be applied to different sub-domains. A preconditioner, based upon the domain-decomposition techniques of Bramble, Pasciak, and Schatz is utilized to efficiently solve the combination domain-decomposition and local grid refinement problem. Techniques for applying this concept to resolve sharp, moving fluid interfaces in large-scale simulation problems will be discussed. Extensions to local time-stepping will also be presented.

1. INTRODUCTION

Domain decomposition and adaptive local grid refinement techniques possess enormous potential for local accuracy improvements in many large-scale problems. In order to illustrate this potential, we will discuss grid refinement

*Departments of Mathematics, Petroleum Engineering, and Chemical Engineering, University of Wyoming, Laramie, Wyoming 82071.

techniques in the context of large-scale simulation of petroleum recovery applications. We will also present ways to easily incorporate local grid refinement capabilities in large, existing codes.

The objective of reservoir simulation is to understand the complex chemical, physical and fluid flow processes occurring in a petroleum reservoir sufficiently well to be able to optimize the recovery of hydrocarbon. To do this, one must build mathematical and computational models capable of predicting the performance of the reservoir under various exploitation schemes. Many of the chemical and physical phenomena which govern enhanced recovery processes have extremely important local character. Therefore the models used to simulate these processes must be capable of resolving these critical local features.

Mathematical models of enhanced recovery processes involve coupled systems of nonlinear partial differential equations. In order to compare the results of these models with physical measurements to assess their validity and to make decisions based on these models, the partial differential equations must be discretized and solved on computers. Field scale hydrocarbon simulations normally involve reservoirs of such great size that uniform gridding on the length scale of the local phenomena would involve systems of discrete equations of such enormous size as to make solution on even the largest computers prohibitive. Therefore local grid refinement capabilities are becoming more important in reservoir simulation as the enhanced recovery procedures being used become more complex and involve more localized phenomena.

There are two distinct classes of local grid refinement techniques—fixed and dynamic. For problems with fixed wells, faults, pinchouts and large fractures, certain fixed local refinements have proven to be very effective. Dynamic and adaptive grid refinement to follow moving fluid interfaces is much more complex. Techniques which work well for fixed refinement can involve a data structure which is so complex that it can be very inefficient for dynamic applications. In this paper we present methods that can be applied to both fixed and dynamic refinement problems in an efficient and accurate manner. We present methods which are accurate discretizations on composite grids and are relatively easy to implement in existing simulators for various applications.

2. MODEL EQUATIONS FOR POROUS MEDIA FLOW

The miscible displacement of one incompressible fluid by another, completely miscible with the first, in a horizontal reservoir $\Omega \subset \mathbb{R}^2$ over a time period $J = [T_0, T_1]$, is given by

$$-\nabla \cdot \left(\frac{k}{\mu}\nabla p\right) \equiv \nabla \cdot \mathbf{u} = q, \quad x \in \Omega, \quad t \in J, \tag{2.1}$$

$$\phi \frac{\partial c}{\partial t} - \nabla \cdot (\mathbf{D}\nabla c - \mathbf{u}c) = q\tilde{c}, \quad x \in \Omega, \quad t \in J, \tag{2.2}$$

where p and $\mathbf{u}$ are the pressure and Darcy velocity of the fluid mixture, ϕ and k are the porosity and the permeability of the medium, μ is the concentration-dependent viscosity of the mixture, $\mathbf{D}$ is a diffusion/dispersion tensor, c is the

concentration of the invading fluid, q is the external rate of flow, and $\tilde{c}$ is the inlet or outlet concentration. In addition to Equations (2.1) and (2.2), initial and flow boundary conditions are specified. The flow at injection and production wells is modeled in Equations (2.1) and (2.2) via point sources and sinks.

The equations describing two phase, immiscible, incompressible displacement in a horizontal porous medium are given by

$$\phi \frac{\partial S_w}{\partial t} - \nabla \cdot \left(k \frac{k_{rw}}{\mu_w} \nabla p_w \right) = q_w, \quad x \in \Omega, \quad t \in J, \tag{2.3}$$

$$\phi \frac{\partial S_o}{\partial t} - \nabla \cdot \left(k \frac{k_{ro}}{\mu_o} \nabla p_o \right) = q_o, \quad x \in \Omega, \quad t \in J, \tag{2.4}$$

where the subscripts w and o refer to water and oil respectively, S_i is the saturation, p_i is the pressure, k_{ri} is the relative permeability, μ_i is the viscosity, and q_i is the external flow rate, each with respect to the i^{th} phase. The pressure between the two phases is described by the capillary pressure

$$p_c(S) = p_o - p_w. \tag{2.5}$$

Note that $\frac{dp_c}{dS} \leq 0$.

Although formally, the equations presented in (2.1) and (2.2) seem quite different from those in (2.3) and (2.4), the latter system may be rearranged in a form which very closely resembles the former system. In order to use the same basic simulator in our sample computations to treat both miscible and immiscible displacement, we define variables for total fluid pressure and Darcy velocity (see [7]). We combine Equations (2.3) and (2.4) to obtain

$$-\nabla \cdot (k\lambda(S)\nabla p) = q_w + q_o = q_t, \tag{2.6}$$

$$v_t = -k\lambda(S)\nabla p, \tag{2.7}$$

$$\phi \frac{\partial S}{\partial t} + \nabla \cdot \left(k\lambda(S)\overline{\lambda}_o \frac{dp_c}{dS} \nabla S \right) + \nabla \cdot (\overline{\lambda}_w v_t) = q_w, \tag{2.8}$$

where S is the water saturation. λ, λ_0 and λ_w, the mobilities of the total fluid and the oil and gas phases, respectively, are defined by ratios of the relative permeabilities and phase viscosities (see [13]).

The equations presented above describe both miscible and immiscible flow in porous media. They can be used to simulate various production strategies in an attempt to understand and hopefully optimize hydrocarbon recovery. However, in order to use these equations effectively, parameters that describe the rock and fluid properties for the particular reservoir application must be input. Since these rock and fluid properties cannot be measured directly in situ, they must be determined via history matching and reservoir characterization techniques (see [12]). The heterogeneities in the reservoir, which can be very localized often dominate the flow process and may require local grid refinement for adequate resolution [16].

A typical example of a fixed localized phenomenon which requires special treatment in simulation is fluid flow in the neighborhood of wells. If fluid flow

rates are specified at injection or production wells, the use of Dirac delta functions as point sources and sinks in the mathematical equations has been shown to be a good model for well-flow behavior beyond some minimal distance away from the wells. In this case, the pressure (which determines the flow) grows like $\ln r$ where r is the distance to that well. A different well model, involving specification of a bottom hole pressure as a boundary condition, also gives rise to a logarithmic growth in pressure up to a finite specified pressure. Because of the rapidly changing behavior of the pressure in the vicinity of wells, accurate pressure approximations require local grid refinement.

We will consider a simple example problem to illustrate our local refinement techniques. From Equation (2.1) or Equation (2.6), the pressure p of a fluid in a horizontal reservoir $\Omega \subset \mathbb{R}^2$ satisfies

$$-\nabla \cdot \frac{k}{\mu} \nabla p = q \quad \text{in} \quad \Omega. \tag{2.9}$$

Assuming no flow boundary conditions, we have

$$\frac{k}{\mu} \frac{\partial p}{\partial \nu} = 0 \quad \text{on} \quad \partial\Omega, \tag{2.10}$$

where $\frac{\partial}{\partial \nu}$ is the outward normal on $\partial\Omega$. For the existence of p we assume that the mean value of q is zero and for uniqueness we impose that p have mean value zero. If fluid flow rates at injection and production wells are specified via Dirac delta functions at the N_w wells x_i with associated flow rates q_i, then

$$q = \sum_{i=1}^{N_w} \delta(x - x_i) q_i \ . \tag{2.11}$$

Several techniques which assume radial flow near the well have been used to obtain local properties of p. One such technique involves subtracting out the singular behavior of p around the wells [8,13,17,20,21]. A radial flow assumption is probably not bad around injection wells, but may be inadequate for production wells where different techniques, such as local grid refinement, are often needed. It has been shown [29] that appropriate local grid refinement around these singularities can greatly increase the accuracy throughout the reservoir.

Two different types of self-adaptive grid refinement have been applied in reservoir simulation. The first technique is a truly local grid refinement where an arbitrary level of refinement can be applied at any region in space. Certain SPE references [9,10,22,28–30] utilize this type of refinement. This technique requires a special data structure for effective matrix set-up as well as special algorithms for efficient solution. A data structure has been developed [9,10,14,15] that will support truly local refinement and dynamic "unrefinement" in both space and time. The special tree structure allows truly local grid refinement and is implemented via an efficient multi-linked list. The dynamic multi-linked list representation efficiently allows both placement and removal of local meshes. A local grid analysis triggers the dynamic changes in the trees for adaptivity.

The data structures have proven to be very effective for elliptic or time-independent partial differential equations, and for fixed refinement applications.

However, the complexity of the data structures and the associated solution processes make many of the truly local refinement procedures inefficient for large-scale, time-dependent problems. If different grids are used for each time-step in a large problem, the overhead associated with the data structures and the grid generation can easily dominate the overall computation times. For this reason, alternate techniques which do not require complex data structures or regeneration of the grid at each time-step are desirable.

A technique termed patch refinement [1,4,11] is an attractive alternative to truly local refinement. This method does not require as complex a data structure but does involve ideas of passing information from one grid to another. The idea of a local patch refinement method is to pick a patch that includes most of the critical behavior requiring better resolution, and use a special, possibly uniform, refinement within this patch. If a uniform fine grid is utilized in the patch, very fast solvers, perhaps utilizing vector-based algorithms, can be applied locally in this region using boundary data from the original coarse grid.

The local patch refinement techniques [4,19,24,26,27] have proven to be very effective for obtaining local resolution around fixed singular points such as wells in a reservoir. We will discuss the patch approximation technique first in the context of local refinement around a fixed point or region like a well and then extend the concept to dynamic problems.

We have developed fast solution methods for the approximation of problems requiring mesh refinement. These techniques are related to various domain decomposition methods [2–6,19,23–25]. High accuracy throughout the computational region is obtained by incorporating local refinements around wells. A composite grid is obtained by superimposing these refinements on a quasi uniform grid on the original domain. Previous techniques usually had no systematic way of dealing with such questions as interface interpolation, mass conservation, and degree of grid overlap. They also usually involve the solution of the coarse grid problems with the regions corresponding to the refinement removed. This destroys the banded structure and ease of vectorization of the coarse grid regions.

In the methods to be discussed below, the problem is formulated with a composite operator on the composite grid. The techniques are iterative procedures which drive the residual of this composite grid operator to zero. Composite grid operators for finite element discretization are common and relatively easy to describe and analyze. Examples of accurate finite difference based composite grid operators for variable coefficient problems are presented in [18]. Complete error analyses for these difference stars will appear elsewhere. A new domain decomposition variant is presented to efficiently solve the resulting matrix equations. This involves the development of a preconditioner. This preconditioner is novel in that the task of computing its inverse applied to a vector reduces to the solving of separate matrix systems for the local refinements and the matrix system for the quasi uniform grid on the original domain. Note that this quasi uniform grid overlaps the regions of local refinement and its corresponding matrix problem remains invariant when local refinements are dynamically added or removed. This local refinement technique can be incorporated in existing reservoir codes without extensive modification. Furthermore, if the nodes on the quasi

uniform grid are chosen in a regular pattern, highly vectorizable algorithms for the solution of the corresponding matrix system can be developed.

In the next section, we describe our technique which allows ease of implementation in existing codes. First a composite grid, as well as a corresponding composite grid operator, is formed from a coarse uniform grid with superimposed refinements in subregions denoted collectively by Ω_2. The coarse grid remains in the region $\Omega_1 = \Omega/\Omega_2$. The discretization in this example is given via finite element techniques. However, by considering the matrix structure of this algorithm, we can see how the techniques can be extended to finite differences or other spatial discretizations.

3. LOCAL GRID REFINEMENT STRATEGY FOR FIXED POINTS

Multiplying (2.9) by an arbitrary (sufficiently regular) function ϕ, integrating by parts and using (2.10), we see that the solution p satisfies

$$A(p, \phi) = (f, \phi) \tag{3.1}$$

where

$$A(u, v) = \int_\Omega \frac{k}{\mu} \nabla u \cdot \nabla v dx$$

and

$$(u, v) = \int_\Omega uv \; dx \; .$$

The Galerkin approximation to (3.1) is to find a function P in a suitable finite dimensional subspace M_h of the Sobolev space $H^1(\Omega)$ such that

$$A(P, \phi) = (f, \phi), \quad \text{for all} \quad \phi \in M_h. \tag{3.2}$$

Since the bilinear form $A(\cdot, \cdot)$ corresponds to the composite operator, (3.2) is, in general, difficult to solve for P. Instead we will use a preconditioned iterative method to obtain P. We must then find a comparable form $B(\cdot, \cdot)$ such that, given g, the problem of finding $W \in M_h$ satisfying

$$B(W, \phi) = (g, \phi), \quad \text{for all} \quad \phi \in M_h. \tag{3.3}$$

is relatively easy.

As was described in [4], the problem of calculating the action of the inverse of the preconditioner essentially reduces to the solution of discrete mixed problems on the refined subgrids, and discrete Neumann problems on the original grid. Due to the regularity of the mesh geometry, such problems are generally easier to solve than the system resulting from the composite grid discretization.

We first split the bilinear form into parts $A(u, v) = A_1(u, v) + A_2(u, v)$, where

$$A_i(u, v) = \int_{\Omega_i} \frac{k}{\mu} \nabla u \cdot \nabla v \; dx \; . \tag{3.4}$$

Then we decompose any $V \in M_h(\Omega)$ as follows: $V = V_p + V_r$ where V_p equals V on Ω_1, $V_p \in M_h(\Omega_2)$ on Ω_2 and $V_r \in M_h(\Omega)$ satisfies

$$A(V_r, \phi) = 0 \quad \text{for all} \quad \phi \in M_h(\Omega_2). \tag{3.5}$$

Then, as in [4], we see that for $V \in M_h(\Omega)$,

$$A(V,V) = A_1(V,V) + A_2(V_p, V_p) + A_2(V_r, V_r) \ . \tag{3.6}$$

The action of the inverse of (3.6) is not easy to obtain. However, by replacing $A(V,V)$ by

$$B(V,V) = A_1(V,V) + A_2(V_p, V_p) + A_2(V_c, V_c) \ , \tag{3.7}$$

where V_c is determined by the original coarse grid $M_h^c(\Omega)$ and satisfies $V_c = V$ in Ω, and

$$A_2(V_c, \phi) = 0 \quad \text{for all} \quad \phi \in M_h^c(\Omega_2), \tag{3.8}$$

then the action of the inverse of (3.7) is relatively easy to obtain and the form $B(\cdot,\cdot)$ is comparable to the form $A(\cdot,\cdot)$ with comparability constants independent of the grid size h [5–6]. As described in [4], the following algorithm suffices for solving

$$B(W, \phi) = (g, \phi) \quad \text{for all} \quad \phi \in M_h \ , \tag{3.9}$$

given g.

Algorithm For Computing W [4]:

1. Find U_p by solving mixed problems on the regions Ω_2.
2. Pass the local information to the right hand side of the original problem and compute any solution U_c of the coarse grid problem

$$A(U_c, \phi) = (g, \overline{\phi}) - A_2(U_p, \overline{\phi}) \quad \text{for all} \quad \phi \in M_h^c \tag{3.10}$$

   where $\overline{\phi}$ is any function in M_h which equals ϕ on Ω_1.
3. Find U_r on Ω_2 by computing the discrete harmonic extension with respect to the refinement subspaces.
4. Compute $\overline{U}$, the mean value of $U = U_p + U_r$. Set $W = U - \overline{U}$.

Matrix Form of the Algorithm:

First, we consider the matrix A^c, generated by the finite element approximation of the equations (2.9)–(2.11) using a coarse quasi-uniform mesh. Let the solution P of the original coarse grid problem be decomposed in the form $P = (P_1, P_b, P_2)^T$, where P_1, P_2, and P_b are the parts of the coarse grid solution in Ω_1, Ω_2, and the intersection of the boundary of Ω_1 and Ω_2, respectively. The corresponding decomposition of the matrix A^c can be described in

$$A^c \begin{pmatrix} P_1 \\ P_b \\ P_2 \end{pmatrix} = \begin{pmatrix} A_{11}^c & A_{1b}^c & 0 \\ A_{b1}^c & A_{bb}^c & A_{b2}^c \\ 0 & A_{2b}^c & A_{22}^c \end{pmatrix} \begin{pmatrix} P_1 \\ P_b \\ P_2 \end{pmatrix}. \tag{3.11}$$

We assume that a code exists or can be easily written to solve (3.11) for a quasi-uniform grid which can take advantage of the banded structure of the matrix $\overline{A}^c$ which is equivalent to A^c except utilizing a standard lexicographical ordering of the unknowns.

Next assume that due to some localized process, grid refinement is desired in Ω_2. Let P_r be the new approximation on the refined grid in Ω_2 and A_{rr} be the local matrix on Ω_2. Let A_{br} and A_{rb} be the new connection matrices between the interface between Ω_1 and Ω_2 and the refined grid on Ω_2. Then, in order to maintain the sparsity of the composite grid matrix and a simple data structure obtained by concatonating P_r to P, we can write the composite matrix problem in the form

$$\tilde{A}\tilde{P} = \begin{pmatrix} A^c_{11} & A^c_{1b} & 0 & 0 \\ A^c_{b1} & A^c_{bb} & 0 & A_{br} \\ 0 & 0 & I & 0 \\ 0 & A_{rb} & 0 & A_{rr} \end{pmatrix} \begin{pmatrix} P_1 \\ P_b \\ P_2 \\ P_r \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ 0 \\ f_3 \end{pmatrix}. \tag{3.12}$$

We note that the I on the diagonal of (3.12) and the zeroes in the corresponding row, column and right hand side enforce the removal of P_2 from the system without destroying the relationship of

$$\begin{pmatrix} A^c_{11} & A^c_{1b} \\ A^c_{b1} & A^c_{bb} \end{pmatrix} \tag{3.13}$$

to A^c and hence $\overline{A}^c$.

As an initial guess for P_r, denoted P_r^0, we solve the local problem on Ω_2 with zero Dirichlet conditions on the interface between Ω_1 and Ω_2 (equivalent to setting $P_b^0 = 0$):

$$P_r^0 = A_{rr}^{-1} f_3. \tag{3.14}$$

This problem can be solved exactly or approximately by some iterative technique. This step could be considered as the first part of a block Gauss-Seidel iterative procedure for the solution of (3.11). The next step would be to use the approximation for P_r^0 and then invert the block (3.13) to obtain an approximation for P_1 and P_b. Since this block involves a complex region and may not be well-conditioned, we use an alternate solution method which involves a preconditioner, denoted by B, for the composite matrix $\tilde{A}$.

Using B, we define, for each iterate n, and an iteration parameter τ

$$\tilde{P}^{n+1} = \tilde{P}^n + \tau B^{-1}(\tilde{f} - \tilde{A}\tilde{P}^n). \tag{3.15}$$

Let Q be the residual vector given by

$$\tilde{f} - \tilde{A}\tilde{P}^n = \begin{pmatrix} f_1 - A^c_{11}P_1^n - A^c_{1b}P_b^n \\ f_2 - A^c_{b1}P_1^n - A^c_{bb}P_b^n - A_{br}P_r^n \\ 0 \\ f_3 - A_{rb}P_b^n - A_{rr}P_r^n \end{pmatrix} \equiv \begin{pmatrix} Q_1^n \\ Q_2^n \\ 0 \\ Q_4^n \end{pmatrix}. \tag{3.16}$$

Next we solve the original coarse grid problem

$$A^c \begin{pmatrix} W_1^{n+1} \\ W_b^{n+1} \\ W_2^{n+1} \end{pmatrix} = \begin{pmatrix} Q_1^n \\ Q_2^n - A_{br} A_{rr}^{-1} Q_4 \\ 0 \end{pmatrix} \tag{3.17}$$

(or its rearranged equivalent problem using $\overline{A}^c$ to take advantage of banding of $\overline{A}^c$) for W_1^{n+1} and W_b^{n+1}. We have in essence inverted (3.13) in an efficient and vectorizable manner. Then using W_b^{n+1}, we complete the block Gauss-Seidel analogy on (3.12) and obtain W_r^{n+1} by solving

$$A_{rr} W_r^{n+1} = Q_4 - A_{rb} W_b^{n+1}. \tag{3.18}$$

Finally, from (3.15), we set

$$\tilde{P}^{n+1} = \begin{pmatrix} P_1^n \\ P_b^n \\ 0 \\ P_r^n \end{pmatrix} + \tau \begin{pmatrix} W_1^n \\ W_b^n \\ 0 \\ W_r^n \end{pmatrix}.$$

Since this algorithm only requires two separate solutions of mixed problems on the subregions (each subregion problem possibly being solved via a different parallel processor) and one solution on the original, uniform coarse grid, it is relatively easy to perform. Similarly, no complex data structure is required and the algorithm can be implemented in existing large-scale codes without severely disrupting the solution process. Promising numerical results for the algorithm appeared in [4]. These results will be extended to more general reservoir simulation problems in a forthcoming paper in the Tenth SPE Symposium on Reservoir Simulation by Ewing, Boyett, Babu, and Heinemann.

As stated, the algorithm in its most general form involves two separate solutions on the subregions at each step. This comes from the desire to have a symmetric preconditioner from the form $B(\cdot,\cdot)$. As is mentioned in [23], the FAC algorithm [19,23,24] involves only one subregion solution per iteration. See [23] for a comparison of FAC and this algorithm [4] and their theories.

By considering the domain decomposition techniques presented in [5] which led to this algorithm, we can see that if the subregion problems ((3.14) and its sequels with updated guesses for P_b) are solved exactly, then Q_4 in (3.16) and (3.17) is identically zero and the action of the preconditioner is symmetric with only one subregion solution per iteration (from (3.18)). Preliminary computations indicate that if the subregion problem is solved iteratively with its own preconditioner, the full algorithm with two subregions solved will converge faster for some problems. Iterative solution of the unrefined region causes no difficulty with either version of the algorithm. This is an important consideration for the reservoir simulation applications when iterative solution of the unrefined problem is essential due to their size, while direct solution of the refined region problems is usually possible.

4. FINITE DIFFERENCE METHODS

In practical reservoir simulation applications, the finite difference system of equations for the unknown values of the fluid pressure at the grid points will be

nonlinear and will be coupled to other nonlinear partial differential equations like (2.2) and (2.8). Using a Newton method, we need to solve, for each iteration, a system with the associated Jacobian or linearization matrix. If the problem is linear, then the Jacobian matrix is the matrix of the finite difference scheme itself. Here we would like to explain how the local refinement technique can be naturally incorporated with the philosophy of the already existing codes for reservoir simulation and easily implemented without destroying their data structure.

We first consider the case for fixed local spatial refinement and a single partial differential equation. We write the approximation to the given equation and boundary conditions for each time step on the whole grid (coarse and fine) as a system of equations for the unknown values of the pressure Φ at the grid points at that time step with $\Phi^n(x) = \Phi(x, t^n)$. Consider the system

$$L\Phi^n = f, \tag{4.1}$$

where L is a positive definite Jacobian matrix, possibly different for each time step, resulting from the finite difference approximation described above, and f represents nonhomogeneous, initial, and boundary terms. The non-zero structure of the matrix L for most of the local grid refinement structures appearing in the SPE literature has been presented explicitly [9,22,29]. For these orderings of the unknowns, the band structure and corresponding efficiency of solution is lost. The solution algorithms presented in these papers have, in general, lost the potential for vectorization which can be so beneficial for the enormous problems encountered. We shall present solution ideas which maintain vectorization benefits and add important parallel capabilities which will be important for the emerging parallel computer architectures.

The local refinement approximations in all cases considered here and in [18] actually replace a coarse grid cell by a group of refined cells. Then the construction of the finite difference scheme may be considered to be created in the following manner: (a) construct within a global strategy of the existing code the finite difference approximation on the coarse grid; (b) treat all cells where the local refinement is introduced as dead cells by enforcing zero pressure values in these cells—the resulting matrix will be L_{CC}; (c) add to the system new equations approximating the problem on the fine grid. Let the matrix L_{FF} describe the regular difference stencil on the interior of the refined regions and the matrix L_{FC} describe the non-standard connections between coarse and refined grid cells described in [18].

This procedure can then be described in a matrix form as

$$\begin{pmatrix} L_{CC} & L_{CF} \\ L_{FC} & L_{FF} \end{pmatrix} \begin{pmatrix} \Phi_L \\ \Phi_F \end{pmatrix} = \begin{pmatrix} f_L \\ f_F \end{pmatrix} \tag{4.2}$$

where Φ_L are the unknowns in the coarse grid cells and Φ_F in the refined cells.

Of course, this ordering of the nodes will destroy the banded structure of L but will maintain the banded structure of L_{CC} and L_{FF}. Thus, any direct method for (4.2) will lead to a large amount of fill-in during the elimination

process and will greatly reduce the efficiency of the model. However, we can define an iterative procedure which takes advantage of the banded structure of L_{CC} and L_{FF} to allow full vectorization of the inversion of each in the algorithms.

We note that Equation (4.2) is not to be stored and used directly, but is intended to motivate our solution technique. The residual produced by (4.2) in an iterative procedure will be the same as the residual produced by the composite grid matrices presented in other papers [9,22,29] depending, of course, on the composite grid stencils. It is this residual that we want to drive to zero. However, there are efficient ways [4] to evaluate this residual without forming the matrix form (4.2). We next define an efficient and vectorizable preconditioner for use in an iterative solution for (4.2). The matrix solution for this preconditioned iterative procedure is very similar to the matrix solution for the finite element based algorithm presented in Section 3.

This procedure is the following: (a) for a given inital guess, solve the problem $L_{FF}\Phi_F = f_F$; (b) with the computed Φ_F solve $L_{CC}\Phi_C = f_C - L_{CF}\Phi_F$; (c) update the guess for the coarse grid and repeat by solving the problem $L_{FF}\Phi_F = f_F - L_{FC}\Phi_L$ using the new iterate for Φ_L. This block Gauss-Seidel iterative technique is clearly very similar to the local refinement algorithm by Bramble, Ewing, Pasciak, and Schatz [4].

Of course, as in the algorithm described in Section 3, the solutions of L_{cc} may not be very well-conditioned. Therefore, instead of a straightforward block Gauss-Seidel iterative procedure for the solution of (4.2), we suggest following the algorithm given in Section 2 by replacing finite element matrices by corresponding finite difference matrices. The extension of the finite element theory to the corresponding finite difference theory will be presented elsewhere.

In determining the data structure for this algorithm, the additional unknowns representing the local grid refinement are simply concatonated on the end of the existing vector of coarse grid unknowns with a pointer set to the beginning location and a length given for the vector of refined cell unknowns. If the coarse grid and refined grid are ordered with a lexicographical ordering, we maintain the banded structure of L_{CC} and L_{FF} which allows efficient vectorization. It is essential to point out that both systems involving L_{CC} and L_{FF} can be solved within the strategy of the existing codes with their data structure and the effective methods that have been developed.

In many current simulations, when wells are brought on line or shut in, strong local transients are generated which adversely affect the convergence of the Newton iterations for the nonlinear systems. The common remedy for this problem, to cut the time-step size across the entire reservoir, is wasteful since the transients are very local in both time and space. This motivates the use of local time-stepping techniques around the wells in combination with the local spatial refinement. In order to get more accurate initial guesses for the Newton method around the wells to enable convergence with the original large time-steps, local time-stepping problems can be defined using the same preconditioning techniques as for local spatial grid refinement. A full discussion with error analysis of this local time-stepping technique will be presented by Ewing, Lazarov, Pasciak, and Jacobs in another paper.

5. ADAPTIVE REFINEMENT FOR DYNAMIC PHENOMENA

Since the local reservoir processes are often dynamic, efficient numerical simulation requires the ability to perform dynamic self-adaptive local grid refinement. The need for adaptive techniques has provided the impetus for the development of local grid refinement software tools, some of which are used in day to day applications for small to mid size problems. Software and engineering tools capable of dynamic local-grid refinement need to be developed for large-scale, fluid flow applications. The adaptive grid refinement algorithms must also be closely matched with the architectural features of the new advanced computers to take advantage of possible vector and parallel capabilities.

For time-dependent problems, often there is much information which can be used from preceding time-steps to help drive our adaptivity process. In parabolic problems, where the solution changes smoothly in time, the "optimal" grid used at the previous time-step should be a very good approximation to the desired grid at the next time step. Thus beginning with a new coarse grid at each time step and using the elliptic techniques of error estimators to drive the local refinement would be wasteful. For small parabolic problems, when the grid is changing very slowly in time, a much better technique would be to take the grid from the last time-step, apply a grid analysis to determine where new grid is needed and where grid is no longer needed, and then change only the grid that indicates need for change. For large time-dependent problems, iterative solution processes are much more efficient than direct solution techniques. For problems with fairly smoothly changing solutions, the same preconditioner can generally be used for several time steps, because the matrices change smoothly, greatly saving in computational effort. If the size of the grid and hence the number of unknowns is constantly changing, clearly the preconditioner must be changed. Similarly, as mentioned earlier, changing the number of unknowns greatly hinders vectorization techniques. Therefore, a considerably more efficient alternative to constantly changing the grid is to use a larger refined area within which the action is maintained for several time steps and to move the patch less frequently, after several steps.

For hyperbolic or transport-dominated parabolic partial differential equations arising in fluid flow problems, sharp fluid interfaces move along characteristic or near-characteristic directions. The computed fluid velocities determine both the local speed and direction of the regions where local refinement will be needed at the upcoming time steps. This information can be utilized in the adaptive method to move the local refinement with the front. We are currently experimenting with using the computed fluid velocities to move the patch grids in quantum jumps. The analysis description and analyses for these methods are given in [11]. The techniques described in the last section are applied at the macro-cell level and the relationship with more general domain decomposition techniques are most apparent. In these techniques, great care must be taken to preserve mass balance when grid is removed and the flow properties must be averaged and described on the new coarser grid.

ACKNOWLEDGEMENTS

This research was supported in part by U.S. Army Research Office Contract No. DAAG29–84–K–0002, by U.S. Air Force Office of Scientific Research Contract No. AFOSR–85–0117, and by the National Science Foundation Grant Nos. DMS–8504360 and DMS–8712021.

REFERENCES

1. M.J. Berger and J. Oliger, *Adaptive mesh refinement for hyperbolic partial differential equations*, Man. NA-83-02, Computer Science Department, Stanford University, 1983.

2. P.E. Bjorstad and O.B. Widlund, *Iterative methods for the solution of elliptic problems on regions partitioned into substructures*, preprint.

3. P.E. Bjorstad and O.B. Widlund, *Solving elliptic problems on regions partitioned into substructures*, in *Elliptic Problem Solvers II*, G. Birkhoff and A. Schoenstadt, eds., Academic Press, 1984, pp. 245-256.

4. J.H. Bramble, R.E. Ewing, J.E. Pasciak, and A.H. Schatz, *A Preconditioning technique for the Efficient Solution of Problems with Local Grid Refinement*, Computer Methods in Applied Mechanics and Engineering, (to appear).

5. J.H. Bramble, J.E. Pasciak, and A.H. Schatz, *An Iterative Method for Elliptic Problems on Regions Partitioned into Substructures*, Math. Comput., 46 (1986), pp. 361–370.

6. J.H. Bramble, J.E. Pasciak, and A.H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comput., 47 (1986), pp. 103–134.

7. G. Chavent, *A new formulation of diphasic incompressible flows in porous media*, in *Lecture Notes in Mathematics*, No. 503, Springer-Verlag, 1976.

8. B.L. Darlow, R.E. Ewing, and M.F. Wheeler, *Mixed finite element methods for miscible displacement in porous media*, Proc. Sixth SPE Symposium on Reservoir Simulation, New Orleans, 1982, pp. 137-146; and Soc. Pet. Eng. J., 4 (1984), pp. 391-398.

9. J.C. Diaz, R.E. Ewing, R.W. Jones, A.E. McDonald, L.M. Uhler, and D.U. von Rosenberg, *Self-Adaptive Local Grid Refinement for Time-Dependent, Two-Dimensional Simulation*, in Finite Elements in Fluids, Vol. VI, Wiley, New York (1985), pp. 279–290.

10. J.C. Diaz and R.E. Ewing, *Potential of HEP-like MIMD Architectures in Self-Adaptive Local Grid Refinement for Accurate Simulation of Physical Processes*, Proc. Workshop on Parallel Processing Using the HEP, Norman, Oklahoma, 1985, pp. 209–226.

11. M. Espedal and R.E. Ewing, *Characteristic Petrov-Galerkin subdomain methods for two-phase immiscible flow*, Computer Methods in Applied Mechanics and Engineering, 64 (1987), pp. 112–115.

12. R.E. Ewing, *Determination of coefficients in reservoir simulation*, in *Numerical Treatment of Inverse Problems for Differential and Integral Equations*, P. Deuflhardt and E. Hairer, eds., Birkhauser, Berlin, 1982, pp. 206-226.

13. R.E. Ewing, *Problems arising in the modeling of processes for hydrocarbon recovery*, in *The Mathematics of Reservoir Simulation*, R.E. Ewing, ed., Frontiers in Applied Mathematics, Vol. 1, SIAM, Philadelphia, 1983, pp. 3-34.

14. R.E. Ewing, *Adaptive Mesh Refinement in Large-Scale Fluid Flow Simulation*, Accuracy Estimates and Adaptivity for Finite Elements, Ch. 16, I. Babuska, O. Zien- kiewicz, and E. Oliveira, eds., John Wiley and Sons, New York, 1986, pp. 299-314.

15. R.E. Ewing, *Efficient Adaptive Procedures for Fluid Flow Applications*, Comp. Meth. Appl. Mech. Eng., 55 (1986), pp. 89–103.

16. R.E. Ewing, *Adaptive grid-refinement techniques for treating singularities, heterogeneities, and dispersion, IMA Volume 11, Numerical Simulation in Oil Recovery*, M.F. Wheeler, ed., Springer-Verlag, New York, 1988, pp. 133–148.

17. R.E. Ewing, J.V. Koebbe, R. Gonzalez, and M.F. Wheeler, *Mixed finite element methods for accurate fluid velocities*, Finite Elements in Fluids, Vol. 4, John Wiley, (1985) pp. 233-249.

18. R.E. Ewing and R.D. Lazarov, *Adaptive locl grid refinement*, Proceedings SPE Rocky Mountain Regional Meeting, SPE No. 17806, Casper Wyoming, May 11–13, 1988.

19. R.E. Ewing, S. McCormick, and J. Thomas, *The fast composite grid method for solving differential boundary-value problems*, Proc. Fifth ASCE Specialty Conference, Laramie, Wyoming, 1984, pp. 1453-1456.

20. R.E. Ewing, T.F. Russell, and M.F. Wheeler, *Simulation of miscible displacement using mixed methods and a modified method of characteristics*, Proc. Seventh SPE Symposium on Reservoir Simulation, SPE No. 12241, San Francisco, November 15-18, 1983, pp. 71-82.

21. R.E. Ewing, T.F. Russell, and M.F. Wheeler, *Convergence analysis of an approximation of miscible displacement in porous media by mixed finite elements and a modified method of characteristics*, Computer Meth. Appl. Mech. Eng., R.E. Ewing, ed., 47 (1984), pp. 73-92.

22. Z.E. Heinemann, G. Gerken, and G. von Hantelmann, G., *Using Local Grid Refinement in a Multiple Application Reservoir Simulator*, paper SPE 12255 presented at the Seventh SPE Symposium on Reservoir Simulation, San Francisco, November, 1983.

23. J. Mandel and S. McCormick, *Iterative solution of elliptic equations with refinement: the two-level case*, Proceedings of Second International Symposium on Domain Decomposition Methods, Los Angeles, California, January, 1988, (to appear).

24. S. McCormick and J. Thomas, *The fast adaptive composite grid method for elliptic boundary value problems*, Math. Comp., 46 (1986), pp. 439–456.

25. J. Pasciak, *Domain decomposition preconditioners for elliptic problems in two and three dimensions: First approach*, First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. Golub, G. Meurant, and J. Periaux, eds., SIAM, Philadelphia, 1988.

26. O.A. Pedrosa, Jr., *Use of Hybrid Grid in Reservoir Simulation*, Ph.D. Thesis, Stanford University, California, December, 1984.

27. O.A. Pedrosa and K. Aziz, *Use of Hybrid Grid in Reservoir Simulation*, paper SPE 13507 presented at the SPE 1985 Middle East Technical Conference, Bahrain, March, 1985.

28. P. Quandalle and P. Besset, *Reduction of Grid Effects Due to Local Sub-Gridding in Simulations Using a Composite Grid*, paper SPE 13527 presented at the SPE 1985 Reservoir Simulation Symposium, Dallas, February, 1985.

29. D.U. Von Rosenberg, *Local Grid Refinement for Finite Difference Methods*, paper SPE 10974 presented at the 57th Annual Fall Technical Conference, New Orleans, September, 1982.

30. M.L. Wasserman, *Local Grid Refinement for Three-Dimensional Simulators*, paper SPE 16013 presented at the Ninth SPE Symposium on Reservoir Simulation, San Antonio, February, 1987.