1 COUPLING DIFFERENT NUMERICAL ALGORITHMS FOR TWO PHASE FLUID FLOW

Malgorzata Peszyńska, Qin Lu, Mary F. Wheeler
Texas Institute for Computational and Applied Mathematics,
University of Texas, Austin, TX 78712 U.S.A.

ABSTRACT

We discuss the coupling of different numerical algorithms for the solution of two phase immiscible flow problems. The computational domain for the flow is split into several subdomains (blocks) on which different numerical algorithms or *models* are implemented. We discuss in particular two algorithms: a fully implicit and a sequential formulation. Both are based on a mixed finite element discretization in space and differ by discretization in time. The codes for the two formulations are run on individual subdomains (blocks) and are coupled across the interface by a set of conditions imposing continuity of primary variables and of the component mass fluxes. The interface code is part of the multiblock multimodel framework which uses mortar spaces to handle nonmatching grids, and is capable of controlling different time stepping in different subdomains. We discuss numerical, mathematical and implementation issues involved in the coupling.

Key words. domain decomposition, multiphase flow, multiblock, mortar spaces, implicit solution, sequential solution, multimodel.

1.1 INTRODUCTION

It is generally believed that the coupling of different models or codes may be the only way to achieve progress in modelling and simulation of problems with complex geometry and physics. Currently many highly specialized algorithms and codes exist which can perform the local tasks in an already optimal or nearly optimal manner. The coupling of these specialized codes with applications to multiphase flow and subsurface modelling is the focus of this paper.

problem size	n=500	n=4000	n=16000
Sequential	.040	.580	4.650
Implicit	.250	4.370	58.070
Black oil in two phase	2.510	41.190	338.000

Table 1.1: Computational cost (in seconds) of running a two phase quarter five–spot problem with n gridblocks using different codes.

In industrial practice many codes have been coupled together in a loose fashion for example by using interface values delivered by one code as boundary values for the next time step of another code. Our multiblock multimodel framework allows for tight coupling. The interface values are the unknowns at every time step. Their values are sought iteratively with a domain decomposition procedure which stops when the conservation of the quantities in question has been satisfied to a given tolerance level. In our applications to multiphase flow in subsurface, the quantities matched at interface are phase pressures or other primary variables, and the conservation of mass across the interface is achieved by iterating the difference in the component fluxes to zero (or desired tolerance). More precisely, the matching condition is imposed in a weak sense.

In the traditional setting, if any part of the computational domain is occupied by n phases, then the n phase simulation code has to be run in the whole reservoir. However at a given time or at all times large parts of that reservoir may be occupied by fewer than n phases or components. For example, a black oil or compositional code (in petroleum industry this corresponds to a three or n phase code, respectively) may have to be run at higher elevations and around wells of an oil reservoir, while in the rest of the computational domain the flow can be simulated efficiently by a two phase code or even by a single phase code. In addition, these codes come in many flavors corresponding to different time and space discretization schemes, well models, solvers, etc. These different numerical algorithms may have optimal applications depending on the magnitude of flow rates, refinement needs, etc. However, without the multiblock multimodel approach, the use of a single complicated code is mandatory and it may be very costly (see Table 1.1 for timings). The multiblock multimodel allows for splitting of the computational domain into blocks in which different codes can be run efficiently.

In this paper we discuss the coupling of different numerical algorithms for the two phase code. Specifically, we consider two immiscible slightly compressible fluids, for example oil and water, and their pressures P_o, P_w, saturations S_o, S_w, concentrations N_o, N_w, and densities ρ_o, ρ_w, respectively, with $N_o = S_o\rho_o$ and $N_w = S_w\rho_w$, flowing in a three–dimensional porous medium (reservoir) with gravity G and depth $D(x)$. Note that phases here are equivalent to components. The porous medium is characterized by the porosity and permeability values $\phi(x), K(x)$, as well as by the values of relative permeabilities and capillary pressure functions k_o, k_w, P_c which are rock– and fluid–specific and are functions of the water saturation S_w and may also be x–dependent. The flow is described by the classical equations of conservation of mass and momentum (Darcy's law) complemented by a set of algebraic constraints and constitutive equations (see [11]). The densities ρ_o, ρ_w are known functions of pressure of each phase with known compressibility constants c_w

and c_o. The rock is assumed to be incompressible, but this constraint is easily removed.

$$\phi \frac{\partial N_o}{\partial t} - \nabla \cdot (\rho_o K \frac{k_o}{\mu_o} (\nabla P_o - \rho_o G \nabla D)) = 0, \tag{1.1}$$

$$\phi \frac{\partial N_w}{\partial t} - \nabla \cdot (\rho_w K \frac{k_w}{\mu_w} (\nabla P_w - \rho_w G \nabla D)) = 0, \tag{1.2}$$

$$S_w + S_o = 1, \tag{1.3}$$

$$P_o = P_w + P_c(S_w). \tag{1.4}$$

The above equations are complemented with a set of initial conditions and the no flow boundary conditions on the external boundary of the reservoir. The flow is driven by injection / production wells which are implemented in the numerical model using the *Peaceman model* (see [12]) and appear in the equations 1.1 and 1.2 as the right hand side terms $q_o(x), q_w(x)$, respectively. Several numerical algorithms were proposed (see [11]) and numerous codes in petroleum industry, environment management, as well as in research labs exist which employ now standard discretization techniques and are validated against field data and results.

While these codes could be understood as the single block approach, the domain decomposition or multiblock approach has been succesfully applied to the multiphase flow, see [2, 16]. Several algorithms exist for coupling with grids matching or nonmatching across interface. However, to our knowledge the coupling of different numerical algorithms is a new research direction in this domain of applications. In this paper we present results obtained as a joint effort of the research group at the Center for Subsurface Modelling at TICAM, UT Austin. The multiblock as well as the multiblock multimodel framework is a part of our in–house simulator framework IPARSv2 [14] (*Integrated Parallel Accurate Reservoir Simulator*). We would like to acknowledge several colleagues who were part of the project, most notably John Wheeler, Ivan Yotov, Manish Parashar, Steven Bryant and Srinivas Chippada.

The outline of the paper is as follows. In Section 1.2 we briefly formulate the multiblock multimodel technique including subdomain and interface algorithms. In Section 1.3 we present some numerical examples using which we discuss related mathematical, numerical and implementation issues. See [15] for more complex examples.

1.2 FORMULATION

In this section we briefly review the numerical algorithm used in the subdomains and on the interface.

1.2.1 Fully Implicit in Time Formulation

In the fully implicit formulation we choose as primary variables the pressure and the concentration of oil P_o, N_o. We first discuss semidiscretization in time. The discrete in time equations at the time t_{n+1} are obtained by the backward Euler formula and are solved for P_o^{n+1}, N_o^{n+1} which satisfy

$$\phi \frac{N_o^{n+1} - N_o^n}{\triangle t_{n+1}} - \nabla \cdot (\rho_o^{n+1} \frac{K}{\mu_o} k_o^{n+1} (\nabla P_o^{n+1} - \rho_o^{n+1} G \nabla D)) = 0, \tag{1.5}$$

$$\phi \frac{N_w^{n+1} - N_w^n}{\triangle t_{n+1}} - \nabla \cdot (\rho_w^{n+1} \frac{K}{\mu_w} k_w^{n+1} (\nabla P_w^{n+1} - \rho_w^{n+1} G \nabla D)) = 0. \tag{1.6}$$

The discretization in space is achieved through the use of expanded mixed finite element methods of lowest order Raviart Thomas on a rectangular grid which by the appropriate quadrature reduce to the cell centered finite differences (see [4, 3, 1]). The edge values are computed by upwinding. The resulting nonlinear system of equations is solved by the Newton method and the Jacobian equation is solved by one of the suite of iterative solvers capable of handling non–symmetric and non–positive systems arising from the Jacobian, for example GMRES. The Newton method stops when the residuals are less than a given tolerance ν.

The fully implicit formulation is known unconditionally stable and permits the use of large time steps which may vary adaptively while keeping the error in mass conservation to minimum. The pitfall of the fully implicit method is the complexity of the Newtonian iteration which may be costly (see Table 1.1). The implementation of the wells in the implicit system allows perfect mass balances with small ν; however, it may critically affect the convergence of the Newtonian procedure. The implicit formulations are therefore applied in a limited number of simulations, in spite of the increase in the computational power and the wide spread of parallel computing. Other formulations known as *IMPES* (*implicit pressures, explicit saturations*) are attractive alternatives. The sequential formulation presented below is another example.

1.2.2 Sequential Formulation

The IPARSv2 two phase sequential model relies on the splitting of the model equations into the time lagged (formally) elliptic part and the parabolic–hyperbolic part. The splitting which we propose goes back to the papers of [6] where different time steps were proposed to be used for the pressure and for the concentration equations. Other formulations related to the one proposed here are those of *total pressure* [11] or *streamlines–streamtubes* methods [5]. Several others not mentioned here exist. The main idea behind the formulation is that even though the individual phase mobilities vary strongly with the water content function, the sum of them (the total mobility) remains close to a constant over many time steps. Therefore the pressure profiles remain stable even though water saturation varies strongly. Once the pressure profiles are known one solves for the saturation analytically (along streamlines) or numerically as we show below.

The primary variables in the sequential formulation are water pressure P_w and saturation S_w. The first equation defines the value of water pressure at the new time step P_w^{n+1} as a solution to the problem

$$\begin{aligned} -\nabla \cdot \left(K \lambda_t^n (\nabla P_w^{n+1}) \right) &= \nabla \cdot (K \lambda_o^n \nabla P_c(S_w^n)) \\ &- \nabla \cdot (K(\lambda_o^n \rho_o^n + \lambda_w^n \rho_w^n) G \nabla D) \end{aligned} \tag{1.7}$$

where the oil, water, and total mobilities are $\lambda_o = \frac{k_o}{\mu_o}$, $\lambda_w = \frac{k_w}{\mu_w}$, and $\lambda_t = \lambda_o + \lambda_w$. Using the values of P_w^{n+1} the densities, the phase velocities u_o, u_w (defined as terms under $\nabla \cdot$ in (1.1–1.2)) and the total velocity $u_t = u_o + u_w$ are computed. Then the saturation equation is solved for S_w^{n+1}

$$\begin{aligned} \frac{\phi S_w^{n+1}}{\triangle t_{n+1}^S} \frac{\rho_w^{n+1}}{\rho_w^n} &+ \nabla \cdot [K \frac{\lambda_o^n \lambda_w^n}{\lambda_t^n} P_c'(S_w^n) \nabla S_w^{n+1}] = \frac{\phi}{\triangle t_{n+1}^S} S_w^n \\ &- \nabla \cdot [\frac{\lambda_w^n}{\lambda_t^n} u_t^n] - \nabla \cdot [K \frac{\lambda_o^n \lambda_w^n}{\lambda_t^n} (\rho_w^n - \rho_o^n) G \nabla D)]. \end{aligned} \qquad (1.8)$$

The time step for saturation may be much smaller than the pressure step or, alternatively, pressure solution can be skipped and redone only every K saturation steps, with K as large as 10 or 60. The saturation time step is limited by the CFL–type stability condition $\frac{v}{\phi} \frac{\triangle t}{\triangle x} < 1$ on the time and spatial discretization steps $\triangle t$, $\triangle x$ in terms of the velocity v and porosity ϕ. Another limitation to the size of time step is a consequence of the presence of wells which are implemented using the Peaceman model. Since the densities as well as mobilities in the pressure equation are time-lagged (explicit) in the well terms, the (strict) material balances show discrepancy in mass conservation which can only be controlled by the time step size, as no parameter ν can be imposed.

The discretization in space is done analogously to what was described in the previous section. The set of two separate fully discrete equations (or more if multiple saturation steps are used) is solved each by a simple iterative linear solver like PCG for symmetric positive definite system. Since the system (1.7–1.8) is effectively a linearized version of (1.5–1.6), the computational cost per time step is much lower (see Table 1.1), but again, the time step may be severely restricted for reasons of accuracy and stability.

1.2.3 Interface coupling

In the previous sections we defined two alternative algorithms used to solve the same set of two phase flow equations. Below we describe the interface coupling of these two algorithms.

For simplicity we shall consider two blocks A and B with interface I. Wells are located in any part of $D = A \cup B \cup I$. The Neumann no–flow boundary conditions are imposed on ∂D. In the multiblock multimodel approach we use the implicit algorithm in the domain (block) A and the sequential algorithm in block B, and we seek the interface values of the primary unknowns on I such that the fluxes of the oil and water components match in a prescribed weak sense. This domain decomposition formulation for mixed methods stems from the classical paper [8] and was extended to the nonmatching grids with mortar spaces in [2] as well as implemented in our framework, see [15].

The choice of the primary unknowns on the interface is problem dependent. Physics of the flow imposes conditions on the interface which reflect the conservation of mass expressed by matching of the component fluxes as well as the conservation of momentum that is equivalent to the equality of pressures. For simplicity we assume below that the capillary pressure function is independent of the position x and then the matching of pressures is equivalent to the equality of (any pair of) saturations and concentrations. This assumption is true for a large class of problems. In such a case, the choice of primary variables influences and is motivated by the convergence properties, computational efficiency or coding convenience. In particular, in the tests presented below we use (P_o, N_o) as the interface primary variables. Note that this set matches the set of primary variables in the implicit formulation used in block A but is different from the one in block B.

We focus again on the semidiscrete in time coupling. The space discretization on the interface with the use of *mortar spaces* and the mathematical form of the matching conditions in the weak form have been described in [2, 16, 15]. The *mortar spaces* technique is capable of handling non–matching grids across the interface and uses suitable projections between the subdomain grids and the interface grids for both primary variables and fluxes. For notational convenience in the discussion below, depending on the context, we will understand as "values" the values of primary unknowns (denoted by $\mathbf{\Lambda}$) or the values of their projections into suitable spaces. Similar convention applies to the values of the fluxes of oil and water across I outward to subdomains $Flux_o, Flux_w$, respectively, and to their jump across I denoted below as $B(\mathbf{\Lambda})$. Furthermore, the interface problem $B(\mathbf{\Lambda}) = 0$ is understood in a weak sense. See [2, 16] and references therein for details.

Interface values applied to the implicit equations in the block A.

Because of the choice of interface primary variables that we assumed above, the application of Dirichlet boundary values to the problem (1.5–1.6) is straightforward, and for a given current guess $\mathbf{\Lambda}^{n+1} = (P_o^{*,n+1}, N_o^{*,n+1})$, we need to solve the problem (1.5–1.6) with

$$P_o^{n+1}|_I = P_o^{*,n+1} \tag{1.9}$$

$$N_o^{n+1}|_I = N_o^{*,n+1}. \tag{1.10}$$

Once the subdomain problem is solved, the normal fluxes of oil and water $Flux_o^A, Flux_w^A$ across I outward to A are computed.

Interface values applied to the sequential equations in the block B.

In the block B, one needs to find a map from the set of primary unknowns on the interface $\mathbf{\Lambda}^{n+1} = (P_o^{*,n+1}, N_o^{*,n+1})$ to the set of primary unknowns in the subdomain for the sequential algorithm $(P_w^{*,n+1}, S_w^{*,n+1})$. There exists a direct algebraic relationship

$$P_w^* = P_o^* - P_c(S_w^*) \quad , \quad S_w^* = 1 - \frac{N_o^*}{\rho_o(P_o^*)} \tag{1.11}$$

which follows the capillary pressure relationship 1.4. However, note that the values $\mathbf{\Lambda}^{n+1}$ are imposed implicitly (in time) whereas the sequential solution uses time–lagged or explicit saturations and mobilities in the pressure equation. The direct applicaiton of 1.11 leads to inconsistency and failure of the interface algorithm. We propose instead to use the consistent formula

$$P_w^{*,n+1} = P_o^{*,n+1} - P_c(S_w^{*,n}) \quad , \quad S_w^{*,n+1} = 1 - \frac{N_o^{*,n+1}}{\rho_o(P_o^{*,n+1})}. \tag{1.12}$$

The pressure equation 1.7 is then modified by the Dirichlet condition

$$P_w^{n+1}|_I = P_w^{*,n+1} \tag{1.13}$$

where the saturation values used for the mobilities are $S_w^{*,n}$. Next, the saturation equation 1.8 is solved and it is complemented by

$$S_w^{n+1}|_I = S_w^{*,n+1}. \tag{1.14}$$

The computation of fluxes $Flux_o^B, Flux_w^B$ outward to B across I follows.

1.2.4 Solution of the interface problem.

The goal of the interface algorithm is to find, at every time step t_{n+1}, the interface values $\mathbf{\Lambda} = \mathbf{\Lambda}^{n+1} = (P_o^{*,n+1}, N_o^{*,n+1})$ so that

$$B(\mathbf{\Lambda}) = |Flux_o^A - Flux_o^B| + |Flux_w^A - Flux_w^B| = 0$$

or, practically, $B(\mathbf{\Lambda}) < \epsilon$, where ϵ is some prescribed tolerance and $|\cdot|$ is a suitable norm.

The problem $B(\mathbf{\Lambda}) = 0$ can be solved by various solvers appropriate for general nonlinear problems. In the results reported below we use the inexact Newton–Krylov method for which the Jacobian $B'(\mathbf{\Lambda})S$ is approximated by a finite difference and the equation to be solved in an interface Newton step is

$$B'(\mathbf{\Lambda})\boldsymbol{S} \approx \frac{B(\mathbf{\Lambda} + \sigma\boldsymbol{S}) - B(\mathbf{\Lambda})}{\sigma}. = -B(\mathbf{\Lambda}),$$

see [15]. Several parameters determine the efficiency and convergence of this technique; see [7, 9, 10]. For lack of space we only comment on the optimal choice of σ. For the multiblock implicit where all subdomain solvers are fully implicit the values of σ are controlled by the ν, and we used $\sigma \approx 10^{-8}$ or less with $\nu \approx 10^{-10}$. For multiblock sequential the optimal values, in the absence of ν, were rather large ($\approx 10^{-4}$). Therefore it is hard to choose a "perfect" σ for multiblock multimodel (some subdomain solvers are implicit, some are sequential), but in practice we use a large one $\sigma \approx 10^{-4}$.

1.3 COMPUTATIONAL EXAMPLES

1.3.1 Heterogeneous permeability example

In our first example we present the so–called quarter five–spot problem for a heterogeneous reservoir (Figure 1.1) with wells located in the lower left (water injection) and upper right (oil and water production) corners. The permeability was $K = 2md$ or $K = 200md$ as shown. We first solve the problem using a uniformly fine grid and a fully implicit code. Then we apply the multiblock multimodel strategy. The location of high permeability layers and wells suggests the use of fine grid and implicit algorithm in the high permeability zones and the use of coarse grid and sequential code in the remaining part of the reservoir where velocities are small. Further decomposition of the high permeability layers into regions around wells (implicit) and elsewhere (handled sequentially) is possible but not as desirable, because the high velocities in the sequential regions will limit the size of the time step. The gridding and the multimodel decomposition is shown in Figure 1.1. Figure 1.1 shows the agreement of oil production rate calculated by the single block and by the multiblock multimodel codes, respectively.

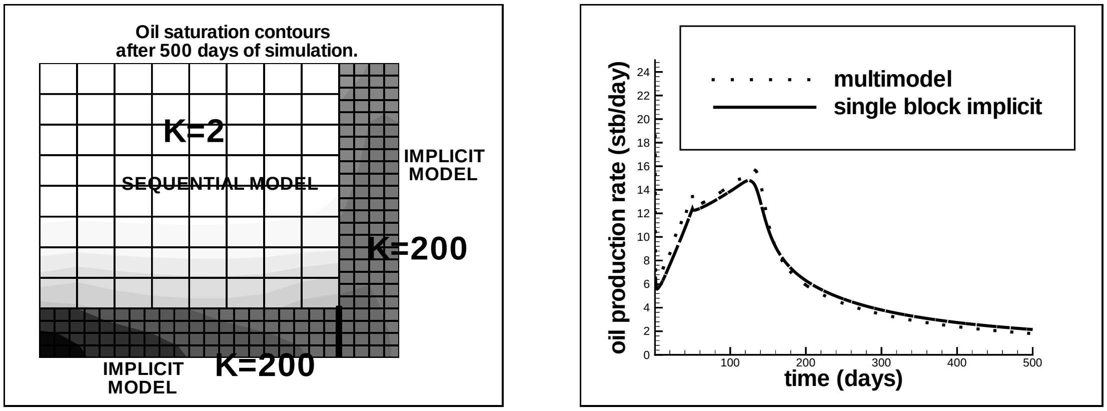

Figure 1.1: Grid and well rates for heterogeneous permeability example.

1.3.2 Numerical and mathematical issues

As it was mentioned above, several elements determine the effectiveness of the interface algorithm for the multiblock multimodel problems. One of them is the difference in how the subdomain problems respond to the imposed boundary values. Consider a 1D (thin and long reservoir) with wells at opposite ends which has been split into three blocks so that I consists of the two disjoint parts. The flow is simulated by the multiblock implicit, mutliblock sequential and multiblock multimodel codes. Figure 1.2 presents the values of the total jump $B(\mathbf{\Lambda})$ as a function of $\mathbf{\Lambda} = (P_o^1, N_o^1)$ for the first time step of simulation. More precisely, at $t = t_1$ the waterflood front has not yet reached either part of the interface I and for the purposes of this experiment we fix the value of N_o to be equal to the initial value. We let P_o imposed on the first interface vary and use the value of P_o on the second as $P_o + dp$ where the optimal value of dp is different for each experiment. Then the jump $B(\mathbf{\Lambda})$ is computed and plotted.

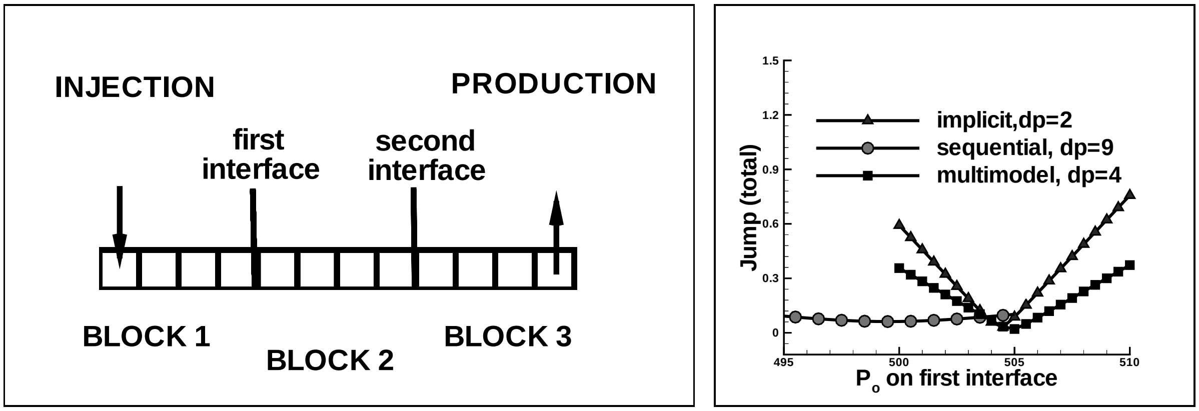

Figure 1.2: Qualitative behavior of $B(\mathbf{\Lambda})$ for 1D example.

One can easily notice the qualitative and quantitative differences between $B(\mathbf{\Lambda})$ plotted for the three cases which are explained by the differences in approximation properties of the implicit and the sequential formulations.

1.3.3 Implementation and parallel computation issues

In our last example we discuss the implementation issues. Aside from the coding effort spent on memory management, visualization, etc., the parallel issues are the most interesting to tackle, as the multimodel code by definition is really an MIMD (Multiple Instruction Multiple Data) code. Our implementation uses multiple MPI communicators [13]. Load balancing is an issue here. The traditional (non–multimodel) load balancing strategy is that the cells are divided more or less evenly between processors. Our experiments show that for optimal load balancing, one processor should never handle more than one model / code, if possible. Figure 1.3 shows the optimal load balancing decomposition and the speedup for the traditional and for the optimal load balancing strategies.

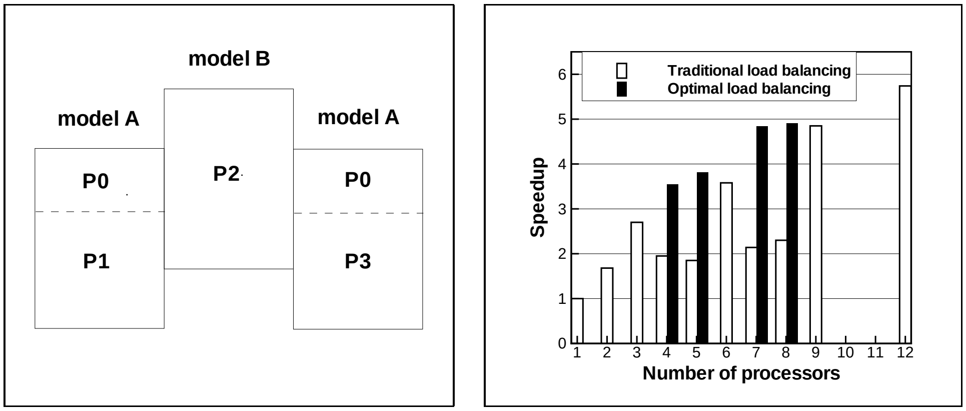

Figure 1.3: Load balancing issues in multimodel implementation.

1.4 CURRENT RESEARCH

The mortar spaces can be time dependent and chosen as to weakly couple or strongly couple the subdomain problems. The interface solver type and convergence parameters may vary adaptively. This is a current research topic.

REFERENCES

[1] T. Arbogast, C. N. Dawson, P. T. Keenan, M. F. Wheeler and I. Yotov (1998). Enhanced cell-centered finite differences for elliptic equations on general geometry. *SIAM J. Sci. Comp.*, 19:404-425.

[2] T. Arbogast, L. C. Cowsar, M. F. Wheeler and I. Yotov. Mixed finite element methods on non-matching multiblock grids. *SIAM J. Num. Anal.*, to appear.

[3] T. Arbogast, M. F. Wheeler and I. Yotov (1997). Mixed finite elements for elliptic problems with tensor coefficients as cell-centered finite differences. *SIAM J. Numer. Anal.*, 34, 828-852.

[4] T. Arbogast, M. F. Wheeler and I. Yotov (1996). Logically rectangular mixed methods for flow in irregular, heterogeneous domains. *Computational Methods in Water Re-*

sources XI, A. A. Aldama and others editors, Computational Mech. Publ., Southampton, 621-628.

[5] R. P. Batycky, M. J. Blunt, M. R. Thiele (1996). A 3D Multi–Phase Streamline Simulator With Gravity and Changing Well Conditions. *17th Intl. Energy Agency Coll. Project on Enhanced oil Recovery*, Sydney, Australia, Sept. 29–Oct. 2

[6] J. Douglas,, R. E. Ewing, M. F. Wheeler (1983). A time-discretization procedure for a mixed finite element approximation of miscible displacement in porous media *R.A.I.R.O. Analyse Numerique, vol. 17, pp. 249–265.*

[7] S. C. Eigenstat and H. F. Walker (1994). Globally convergent inexact Newton method. *SIAM J. Sci. Optim.*, 4:393-422

[8] R. Glowinski and M. F. Wheeler (1988). Domain decomposition and mixed finite element methods for elliptic problems, *Domain Decomposition Methods for Partial Differential Equations*, SIAM, Philadelphia, 144-172.

[9] C. T. Kelly (1995). *Iterative methods for linear and nonlinear equations.* SIAM, Philadelphia.

[10] H. Klie (1996). Krylov-secant Methods for Solving Large Scale Systems of Coupled Nonlinear Parabolic Equations. *PhD thesis*, Rice University, Houston, Texas.

[11] D. W. Peaceman (1977). *Fundamentals of numerical reservoir simulation.* Elsevier, Amsterdam.

[12] D. W. Peaceman (1983). Interpretation of well-block pressure in numerical reservior simulation with non-square grid blocks and anisotropic permeability. *Tran. AIME*, Vol. 275.

[13] M. Snir, S. Otto, S. Huss-Lederman, D. Walker and J. Dongarra (1996). *MPI: the complete reference.* The MIT Press.

[14] P. Wang, I. Yotov, M. Wheeler, T. Arbogast, C. Dawson, M. Parashar and K. Sephernoori (1997). A New Generation EOS Compositional Reservoir Simulator: Part I – Formulation and Discretization. *SPE 37979, 1997 SPE Reservoir Simulation Symposium*, Dallas, Texas.

[15] M. F. Wheeler T. Arbogast, S. Bryant, J. Eaton, Q. Lu, M. Peszynska and I. Yotov (1999). A parallel multiblock/multidomain approach for reservoir simulation. *SPE 51884, 1999 SPE Symposium on Reservoir Simulation*, Houston, Texas.

[16] I. Yotov (1997). A mixed finite element discretization on non-matching multiblock grids for a degenerate parabolic equation arising in porous media flow. *East-West J. Numer. Math.*, 5:211-230