Multiphysics Coupling of Codes

Małgorzata Peszyńska, Qin Lu & Mary F. Wheeler

Texas Institute for Comp. and Applied Math.,
The University of Texas at Austin,
Austin, TX 78712, USA

January 12, 2000

Abstract

We discuss the coupling of codes for the solution of multi–component multi–phase flow problems in the subsurface. The coupled codes simulate the flow and transport in a porous reservoir which can be decomposed into several parts, based on the number and type of fluids which flow there. Each part is associated with a *physical model* which comprises a set of conservation equations, constitutive laws and a numerical algorithm implemented. The physical model codes are coupled across interfaces by a set of matching or approximately matching conservation equations and constitutive laws. In particular, we discuss the coupling of a black–oil model with a two–phase model and a single–phase model for simulation of oil recovery in a reservoir with strong gravitational phase segregation.

Keywords multi–physics, domain decomposition, multi–phase flow, multi–block, implicit solution, single–phase, two–phase, black–oil, multi–model.

1 INTRODUCTION

Most simulators in reservoir engineering follow the traditional approach in which one complex code code is used to simulate multi–phase and/or multi–component processes occuring in the whole reservoir. Even with the use of *domain decomposition* techniques which allow for flexible treating of nonuniform geometry due to geological faults, surface irregularities, etc., the codes which run in the individual subdomains are essentially "clones" of one given code of a fixed computational complexity.

In mega–size applications it is highly desirable to reduce the overall computational cost of the simulation by individually selecting the most appropriate code to be executed in a given subdomain. For example, in an aquifer part of the reservoir one only needs a single–phase, possibly also a multi–component, code, while in some other part of the reservoir the presence of a *gas cap* requires application of a black–oil or of a full *compositional* code. Furthermore, the regions around wells which are typically characterized by high flow rates require grid refinement and careful time stepping as well as implicit formulation necessary to maintain mass balance, while in the far field a sequential code on a coarse grid can give satisfactory results. This is the motivation for the use of different codes in different parts of the computational domain. Of course, these codes have to be coupled as quantities have to be conserved across the interfaces. The interface coupling and the implementation itself raise many interesting and delicate mathematical and computational issues. The multi–physics concept of selecting, running, and coupling the individual codes associated with different subdomains is the focus of this paper. Our approach allows for efficient use of computational resources in a heterogeneous computing environment while it does not compromise accuracy of the simulation.

The research reported in this paper is based on the general purpose simulator framework IPARS (Integrated Parallel Accurate Reservoir Simulator), see [21, 14], which has been developped in CSM. IPARS presently combines 8 different physical models in a general multi–model multi–block or domain decomposition framework [21, 22]. The physical models range from single–phase to three–phase and they realize different discretization and time stepping approaches, and they come in implicit, semi–implicit or explicit versions. The framework allows for coupling of sequential and implicit codes as well as for variable time–stepping in different subdomains [22, 17]. The interface algorithm allows for nonmatching grids across the interface and it uses the *mortar spaces technique* [2, 4, 3, 1, 22, 23] combined with the inexact Newton–GMRES procedure [22, 11, 9].

In this paper we focus on the coupling of the three model codes which are part of IPARS: the black–oil, two–phase, and single–phase codes. The issues involved are those of multi–physics in that the interfaces fall in the regions where one or more

phases are absent. The multi–physics algorithm is responsible for determining which variables are relevant in the computation and which will be used as primary variables on the interface. The multi–physics multi–block algorithm is formulated as a domain decomposition problem [10] so that the unknowns are values of primary variables on the interface. The resulting nonlinear problem is solved by the previously mentioned interface Newton–GMRES algorithm. Additionally, the multi–physics algorithm resolves matching or approximation of the matching of the constitutive equations across interface. This is necessary because the individual models are build to fit traditional engineering approaches and formulations and therefore they do not fit into a common frame as concerns the definitions of subdomain primary variables, their units or data. Consider for example a dipping reservoir 1500' x 1300' x 40', of layered permeability

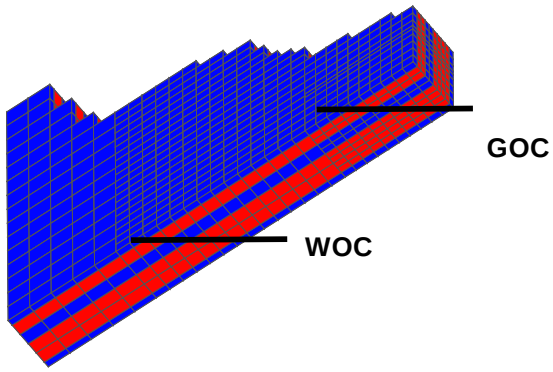

Figure 1: Grid and permeability. Middle layers have permeability of 200 md, outside layers have permeability 20 md.

field, surrounded by an irregular boundary made of impermeable shales, see Figure 1 [13]. The hydrostatic distribution of fluids in equilibrium is as on Figure 2 where contours of oil concentration N_O are shown. Due to gravity, oil and gas prevail in upper parts of the reservoir and disappear towards the bottom where mostly water (the aqueous phase) is present. Below the water–oil contact (WOC) only water is present. The gas phase is present only at the top above the gas–oil contact (GOC). In fact, the GOC can be located outside the reservoir, in which case the pressure in the whole field is above the *bubble point*. Further, the oleic phase contained between GOC and WOC is composed of light and heavy hydrocarbon components. The concentration of lighter hydrocarbon component depends on the production history of the reservoir. In particular, lower parts of the reservoir may only contain the heavy *dead–oil* component.

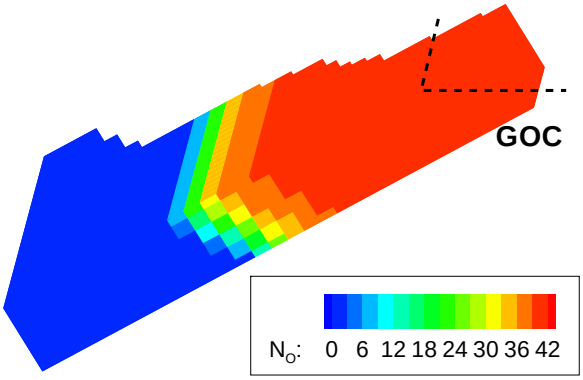

Figure 2: Initial fluids distribution.

For this reservoir we further assume that there is a production well located at the top of the formation and that, in order to maintain pressure, several water injection wells are placed in the bottom part of the region. Simulation of hydrocarbon production in this reservoir requires, with the traditional approach, the use of a black–oil code [15, 20]. This is due to the need to account for phase behavior of light and heavy hydrocarbons. However, in the main part of the reservoir the fluids operate under typical two–phase or single–phase conditions. Meanwhile, the black–oil code is 6–10 times slower than the two–phase code, while both, the black–oil and the two–phase codes, are orders of magnitude more costly than the single–phase code. This provides the motivation for the use of the multi–physics approach. The computational domain is split into three parts arranged vertically with the dip along with the distribution of fluids (see Figure 3). The single–phase (or two–phase) model is assigned to the bottom

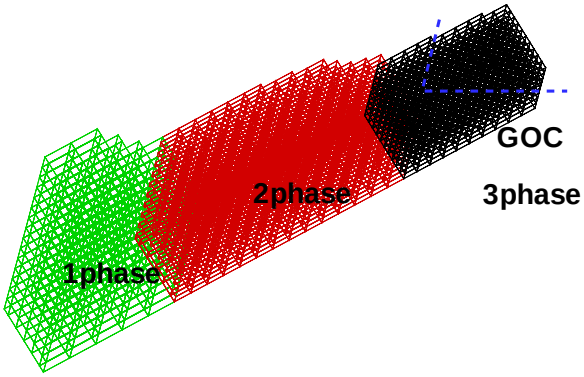

Figure 3: Assignment of models to subdomains.

part, the two–phase and the black–oil model to the middle and the top part, respectively. The codes are coupled across interfaces which are in the planes perpendicular to the dip direction.

The plan of the paper is as follows. In Section 2, we recall the discrete–in–time equations for all three physical models used in the coupling. These are the single–phase, the two–phase and the black–oil models of flow. Section 3 describes the interface coupling and Section 4 presents the implementation issues. In Section 5 we discuss results of the computational example introduced above.

The following notation is used in the paper: capital subscripts W, O, G are used for fluid components; respectively, for water, heavy hydrocarbon or oil and light hydrocarbon or gas component. Small subscripts w, o, g are used for phases: aqueous or water, oleic and gaseous phase, respectively. For simplicity we assume that the rock (solid) phase is immobile and that no adsorption, reaction, or dispersion takes place. The variables used are phase pressures P_w, P_o, P_g, phase saturations S_w, S_o, S_g, and component concentrations N_W, N_O, N_G. The system satisfies the volume constraint

$$S_w + S_o + S_g = 1. \tag{1}$$

The capillary pressure relationships are given (for three–phase relationships we use Stone's models [7]):

$$P_{cow}(S_w) = P_o - P_w, \tag{2}$$

$$P_{cgo}(S_g) = P_g - P_o. \tag{3}$$

1.1 *Acknowledgements*

We would like to acknowledge several contributors to the multi–block IPARS project, in particular Manish Parashar, Ivan Yotov, and John Wheeler.

2 FORMULATION OF SUBDOMAIN MODELS

In this section we briefly describe the *physical models* used in the subdomains: we recall the conservation equations, constitutive laws and their discrete–in–time formulation with the backward Euler formula. The space discretization is based on the expanded mixed finite element methods of lowest order Raviart Thomas type on a rectangular grid which by the appropriate quadrature reduce to the cell centered finite differences [3, 4, 2]. The edge values are computed by upwinding.

All the models discussed here are fully implicit and they are solved by a subdomain Newton iteration on the values of primary variables. The Newton method stops when the residuals are less than a given tolerance ν. The primary variables are different for

each model or subdomain and as a consequence the matching conditions on interface have to include mappings from one set of variables to another.

In the equations below, the porosity ϕ and permeability tensor K are spatially varying and constant in time reservoir rock data. Other rock properties involve relative permeability and capillary pressure relationships which are given functions of saturations and possibly also of position in the case of different *rock types*. The well injection / production rates q_W, q_O, q_G of the components are defined using the Peaceman well model [16] and they describe typical well conditions for pressure or rate specified wells.

We start with the least complicated case, the single–phase equations. These describe the flow in this part of the domain where $S_o = S_g = 0$, and $S_w = 1$. Then we define the two–phase model where $S_g = 0$ and $S_w + S_o = 1$, and finally we discuss the black–oil equations in which all three phases may be present.

2.1 *Single–phase model*

The single–phase model describes the (saturated) flow of a (slightly compressible) fluid, in the context of this paper considered to be water (aqua), of constant compressibility c_w and density ρ_w, which depends on the pressure P_w as follows

$$\rho_w = \rho_W^{ref} e^{c_w P_w}. \tag{4}$$

The pores of the rock are saturated with water, i.e., $S_w = 1$, and so $N_W = \rho_w$. The mass conservation equation and Darcy's law discretized in time are

$$\frac{(\phi\rho_w)^{n+1} - (\phi\rho_w)^n}{\triangle t_{n+1}} - \nabla \cdot U_w^{n+1} = q_W^{n+1}, \tag{5}$$

$$U_w^{n+1} = \frac{K}{\mu_w} (\rho_w)^{n+1} (\nabla P_w^{n+1} - \rho_w^{n+1} G \nabla D). \tag{6}$$

After discretization in space, the above system is solved for water pressure P_w. In the incompressible case and in the absence of gravity and wells, the system becomes linear; otherwise, it is quasilinear.

2.2 *Two–phase model*

The two–phase model in this paper describes the flow of two slightly compressible fluid phases (aqueous and oleic) identified with immiscible (pseudo) components water and oil. We have constitutive equations

$$N_M = S_m \rho_m = S_m \rho_M^{ref} e^{c_m P_m} \tag{7}$$

for each component, $M = O, W$, identified with a phase m=o,w, respectively. The discrete–in–time mass conservation equation and Darcy's law read

$$\frac{(\phi N_M)^{n+1} - (\phi N_M)^n}{\triangle t_{n+1}} - \nabla \cdot U_m^{n+1} = q_M^{n+1}, \tag{8}$$

$$U_m^{n+1} = \frac{K}{\mu_m} (\rho_m k_m)^{n+1} (\nabla P_m^{n+1} - \rho_m^{n+1} G \nabla D). \tag{9}$$

The system is solved for P_o, N_O: this choice of variables gives all the other variables directly from functional relationships. For example, $S_w = 1 - \frac{N_O}{\rho_o(P_o)}$. Note that other choices of primary variables, for example, the choice of P_w, N_O, may require an implicit solve to get S_w (see Section 3).

2.3 *Black–oil formulation*

The black–oil model is a three phase (water, oil and gas) model describing the flow in petroleum reservoir. It is assumed that the aqueous phase contains only water component and that the water component does not exist in other phases. Furthermore, the gas phase contains only the light hydrocarbon component. These are standard assumptions [15, 12], also [18, 6, 20, 13, 5].

In the fully implicit black–oil model considered here, the primary variables are water pressure P_w, oil component concentration N_O, and gas component concentration N_G. The discretized mass conservation equations are

$$\frac{(\phi N_W)^{n+1} - (\phi N_W)^n}{\triangle t_{n+1}} - \nabla \cdot U_w^{n+1} = q_W^{n+1}, \tag{10}$$

$$\frac{(\phi N_O)^{n+1} - (\phi N_O)^n}{\triangle t_{n+1}} - \nabla \cdot U_O^{n+1} = q_O^{n+1}, \tag{11}$$

$$\frac{(\phi N_G)^{n+1} - (\phi N_G)^n}{\triangle t_{n+1}} - \nabla \cdot (U_G + R_o U_O)^{n+1} = q_G^{n+1}. \tag{12}$$

Note that the gas component flux includes the amount of gas or light hydrocarbon component present in the gas phase as well as the amount dissolved in the oil phase through the gas-oil ratio $R_o = \frac{N_G}{N_O}$ [8]. Darcy's law for multi–phase flow is used to calculate the mass velocity of phase m, for m equal to w, o, g, reads

$$U_m^{n+1} = \frac{K}{\mu_m} \left(\frac{k_m}{B_m} \right)^{n+1} (\nabla P_m^{n+1} - \rho_m^{n+1} G \nabla D), \tag{13}$$

where B_m is the formation volume factor of phase m.

If the pressure in the reservoir is high enough (above the *bubble point*) then all of the light hydrocarbon component is dissolved in the oil phase. The following criterion is used to determine whether (locally) a cell is in three–phase or in the two–phase regime:

$N_g > R_{so}N_o$: three–phase , oil phase saturated with gas,
$N_g \leq R_{so}N_o$: two–phase, oil phase undersaturated.

where R_{so} is the solution gas-oil ratio given as reservoir data.

3 INTERFACE COUPLING

The subdomain physical models presented above are summarized in Table 1. At the beginning of the simulation the variables of these models are initialized according to assumed initial conditions. In IPARS, most commonly, some form of hydrostatic equilibrium between phases is assumed. The interface variables are initialized from the conditions in the surrounding subdomains which should be consistent. In order

single–phase	two–phase	black–oil
P_w	P_o	P_w
	N_O	N_O
		N_G

Table 1: Primary variables for the subdomain models.

to illustrate the difficulties related to mapping different sets of primary variables, suppose that we are given interface values of P_w^*, N_O^* that are to be mapped to the primary variables P_o, N_O in two–phase model. Note that the (water) saturation S_w for the two–phase model can be computed from the implicit relationship

$$N_O = (1 - S_w)\rho_O^{ref} \exp(P_w + P_{cow}(S_w)). \tag{14}$$

Once S_w is known, the value of $P_o = P_w + P_{cow}(S_w)$ can be computed. This may require inverting usually degenerate capillary pressure relationships. Also, generalization of this kind of computation for the black–oil or compositional model would require an interface *flash*.

Consider now the case of a reservoir similar to the example introduced in Section 1. After gravitational segregation, the bottom of the reservoir is filled with water only

(an aquifer) and the remaining part contains water and hydrocarbons in two or three phases with the gas phase appearing only at the top of the reservoir.

The interfaces between the subdomains or models are located in the regions where one or more phases are absent, so that it is legitimate to consider a simple model on one side and a more complex one on the other. The more complex model operates in the situation of residual content of fluids. Further, assume that the direction of the flow during the simulation is such that these residual conditions are not violated.

The interface conditions come from physical principles of conservation of momentum and mass across the interface. These are realized, respectively, by the matching of phase pressures and by the matching of component mass fluxes. The matching is an iterative process. Regardless of the mechanism, the solution procedure is by trial–and–error: a certain guess for the values of primary variables, e.g., pressures, is used as a Dirichlet condition for subdomain models. The solution corresponding to this set of values on the interface is found. This solution gives rise to fluxes across the interface. If these fluxes match, the solution has been found. Otherwise, another guess for the pressures is sought. In fact, the solution procedure is more sophisticated and comprises the use of mortar spaces to account for nonmatching spatial grids, an inexact Newton-Krylov solver, different time steps in the models etc. [22, 17].

Consider the interface between the single–phase and the two–phase models which as in Figure 2 is situated in the aquifer part of the reservoir where $S_o = 0, N_O = 0$. More precisely, S_o is equal to the residual oil saturation. The fluids are assumed to flow (generally) from the aquifer to the two–phase region or from left to right as on Figure 2. The quantity to be determined on the interface is the value of P_w^*, which is used as a (Dirichlet) boundary condition imposed for the single–phase. The two–phase model uses this value of P_w^* to determine the boundary conditions for its primary variables: P_o^* comes from capillary pressure at residual conditions, and N_O^* reflects the residual conditions.

The interface conditions between two–phase and black–oil are defined similarly. The underlying assumption on conditions in the neighborhood of this interface is that the oil phase is *undersaturated* with gas, in other words, that the pressure is well above the *bubble point*. The momentum conservation across the interface requires that the pressures of water P_w^* and of oil P_o^* match. For simplicity, assume that the rock is of the same *rock type* in the neighborhoood of the interface, which implies that it is described by the same capillary pressure relationship. As a consequence, the matching of pressures is equivalent to the matching of saturations and to the matching of the component concentrations. Because of complexity of the black–oil model, it is convenient to choose the set of interface primary variables as equivalent to the set of black–oil subdomain primary variables P_w, N_O, N_G.

In fact, because the variable N_G is not represented in the two–phase model part, the value of N_G near interface is assumed to be small and such that the oil–gas ratio R_o is small and remains approximately constant throughout the simulation. This allows for variable N_O to have (approximately) the same meaning on both sides of the interface. In particular, in the *dead oil* case we have $N_G = 0, R_o = \frac{N_G}{N_0} = 0$ and the oil component is the only component in the oil phase on both sides of the interface. We do not need to solve for N_G^*, as it is known to be zero. In the case when R_o is nonzero, however, we still do not solve for N_G^* as it is not represented on the two–phaseside of the interface. However, its value is needed for constitutive equations for the oil phase and in the computation of fluxes. For that we use the values of N_G and R_o, which are projected from the neighboring cells.

Additional considerations must be given to constitutive equations. In the *dead–oil* case, B_o is a function only of P_o and then oil phase density is $\rho_o = \frac{\rho_O^{ref}}{B_o}$. In the case other than *dead–oil*, the density of oil phase in the black–oil model is dependent on P_o, N_O, and N_G through $\rho_o = \rho_o(P_o, R_o)$, while it is only a function of P_o in the two–phase model subdomain as in 7. In order to accomodate that discrepancy, for a given set of R_{so}, B_o etc. we find the closest matching values of reference oil component density and compressibility for use in 7.

In summary, the interface primary variables whose values are used as Dirichlet data for different models in our multi–model implementation are P_w^*, N_O^*, N_G^*. Not all of these are relevant on each interface. If a phase is absent, in other words, if an interface variable takes residual values, then we do not solve for this variable: such variables are enclosed in parenthesis in Table 2.

single–phase&two–phase	two–phase&black–oil
P_w^* (N_O^*)	P_w^* N_O^* (N_G^*)

Table 2: Primary variables on the interface.

4 IMPLEMENTATION

The physical models described above are built in IPARS framework (see Section 1), which handles general input/output, memory management, grid generation, visualization, parallelism, etc. The code for interface algorithm has been merged with the framework. The framework can have multiple (fault) blocks (or subdomains), each of

which may have associated its own physical model. The neighboring blocks are connected via an interface. The values of primary variables and fluxes are projected back and forth between subdomains and interface and the subdomain solvers with Dirichlet data are executed until the fluxes from the two sides match to a given tolerance. Unit conversion between different models may be necessary during projection.

For the subdomain cells adjacent to the interface, the Dirichlet boundary condition is applied using values of primary variables delivered by the interface code. Transmissibilities, mobilities, fluxes, etc. are computed and stored. The Dirichlet condition is applied to the Jacobian and residuals of the discrete system.

The linear solvers for different physical models can be either different or the same. A parallel GMRES solver has been extended for solving multiple models simultaneously. The basic idea for the extension is to expand the work space from a scalar to an array, so that each model has its own entry of work space.

The parallelism is the most delicate and interesting issue to tackle. Different from the traditional single model simulator, the multimodel problem is actually a MIMD (multiple instruction multiple data) problem. We use multiple MPI communicators [19, 17] in implementation. The processors are split into multiple groups (or *communicators*) so that each physical model has its own communicator. Within a communicator, the adjacent processors can exchange boundary information which is necessary for parallel computation. The message passing between different communicators is also allowed.

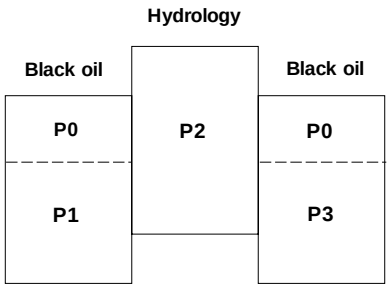

Figure 4: An example of optimal load balancing with 4 processors and 3 blocks in multimodel problem.

The load balance is important for parallel efficiency. The traditional (single model) load balancing strategy is that the grid cells are divided more or less evenly between processors. In the multi–model implementation a different strategy is used. Our experiments show two rules to achieve optimal load balancing. First, because of synchronization issues between different communicators, if possible, one processor should never handle more than one model / code. Second, the number of processors assigned to each model should be proportional to its simulation speed. Figure 4 shows an example of optimal load balancing decomposition with 4 processors. Figure 5 shows the speedup for the traditional and for the optimal load balancing strategies for coupling of two–phase and of the black–oil models. We note that the speed ratio between two models when they are running independently (around 10) is higher than the ratio when they are coupled (2-6). One reason is that running with no–flow boundary conditions as opposed to running with Dirichlet boundary conditions changes the efficiency of subdomain Newton solves, in particular, because the subdomain solvers, when run within interface iterations, usually start with a fairly good initial guess as opposed to the one from previous time step. Another reason is the parallel communication overhead. Table 3 lists the number of processors assigned to different models to achieve optimal load balancing when they are coupled, assuming the grid sizes of all blocks are the same.

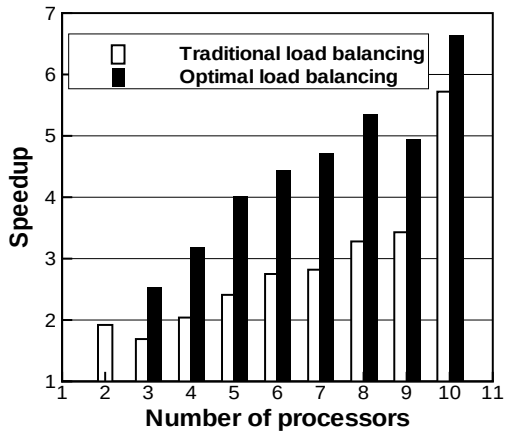

Figure 5: Speedup for the traditional and for the optimal load balancing in multimodel problem.

Total number of processors	Number of processors per block	
	black-oil	hydrology
2	1	2
3	1	1
4	2	1
5	2	1
6	2	2
7	3	2
8	3	2
9	4	2
10	4	2

Table 3: Number of processors assigned to different models for a 3-block example.

5 EXAMPLE

Below we discuss and compare the results of the simulation for the motivating example from Section 1. We ran the case in A) the multi–physics mode, using two models: the two–phase and the black–oil. The case with single–phase model will be discussed elsewhere. Additionally, we ran the case using B) only the black–oil code.

The case was run over 1000 days. The time step was originally equal to 1 day and increased over the course of the simulation to maximum of 3 days; the flow rates were relatively high as the breakthrough occured soon after 200 days. The whole simulation required about 390 time steps to complete. The field was originally at 2000 psi at the depth of 200 ft, which corresponds to a location within the two–phase model with the dip angle about 25'. The permeability was 20 and 200 milidarcies in the vertical and horizontal direction, respectively, and was smaller by a factor of 10 in the layers as shown in Figure 1. There were about 5000 gridblocks in the reservoir. In the multi–physics run, they were split to around 3000 in the black–oil part of the reservoir and to about 2000 in the two–phase. The multi–physics case was run with a minimum number of interface degrees of freedom, but, since the flow direction was well defined and the case had relatively few local heterogeneities, the interface iterations converged fast: it required on the average 25 subdomain evaluations before breakthrough and only 6 after breakthrough.

The simulation results are shown in Figure 6. Profiles of oil concentration should

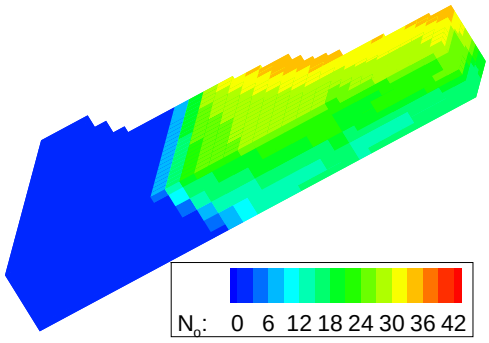

Figure 6: Oil concentration after 1000 days.

be compared to those in Figure 2. It is clear how the middle block has been swept out by water with most of the flow going through high permeability channel. There is a small discontinuity in the coloring of profiles at the bottom of the reservoir close to the interface. This can be explained in two ways. First, the coarse interface discretization which we used to speed up the interface convergence, cannot give continuity at fine resolution scale for nonmatching grids. Secondly, there is a visualization artifact which occurs for nonmatching grids, even if the computed field is continuous, and it is caused by plotting of the contours separately for each faultblock.

The results obtained are close enough to the results of the second simulation run in which only the black–oil code was used. The meaningful comparison of point values of simulation variables for this complex run, however, is virtually impossible. As a reasonable alternative, we compare the well output (instantaneous and cumulative injection and production rates) which prove to be very close for the two models, see Figure 7. The only siginificant discrepancy appears in the injection rates which will be further studied.

6 CONCLUSIONS

We presented the multi–physics algorithm for multi–phase flow. The multi–physics approach allows for coupling of different codes or *physical models* which are run in separate subdomains and are coupled across the interfaces between subdomains. We discussed the implementation issues and presented an example which was run using the coupling of a black–oil and a two–phase model.

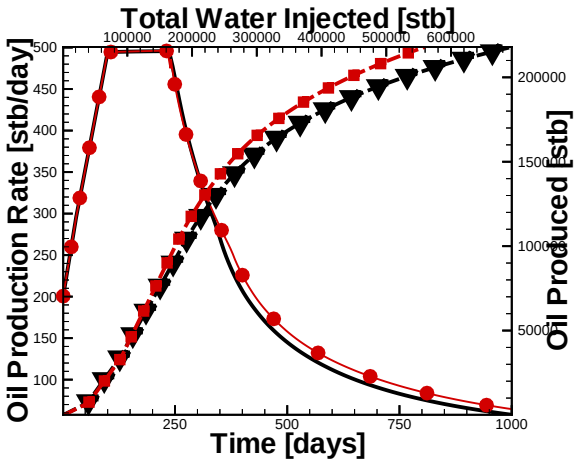

Figure 7: Comparison of well rates obtained for multi–physics (run A) and black–oil (run B). Solid line: oil production rate vs time for run A (no symbols) and run B (circles). Dashed line: oil produced vs water injected for run A (triangles) and run B (squares).

The multi–physics approach allows to select the most appropriate code to run in a given subdomain without compromising the accuracy of computations. The efficiency of the implementation has yet to be assessed, especially on parallel machines where load balancing is critical.

References

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

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

[3] T. Arbogast, M. F. Wheeler, and I. Yotov. Logically rectangular mixed methods for flow in irregular, heterogeneous domains. In A. A. Aldama et al., editors, *Computational Methods in Water Resources XI*, pages 621–628, Southampton, 1996. Computational Mech. Publ.

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

[5] L. Bergamaschi, S. Mantica, and G. Manzini. A mixed finite element-finite volume formulation of the black-oil model. *SIAM J. Sci. Comp.*, 20(3):970–997, 1998.

[6] K. H. Coats, L. K. Thomas, and R. G. Pierson. Compositional and black oil reservoir simulator. In *the 13th SPE symposium on reservoir simulation*, San Antonio, Texas, Feb 12-15 1995.

[7] M. Delshad and G. A. Pope. Comparison of the three-phase oil relative permeability models. *Transport in Porous Media*, 4:59–83, 1989.

[8] M. J. Economides and A. D. Hill. *Petroleum production systems.* Prentice Hall, Englewood Cliffs, New Jersey, 1994.

[9] S. C. Eigenstat and H. F. Walker. Globally convergent inexact newton method. *SIAM J. Sci. Optim.*, 4:393–422, 1994.

[10] R. Glowinski and M. F. Wheeler. Domain decomposition and mixed finite element methods for elliptic problems. In *Domain Decomposition Methods for Partial Differential Equations*, pages 144–172, Philadelphia, 1988. SIAM.

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

[12] L. W. Lake. *Enhanced oil recovery.* Prentice Hall, Englewood Cliffs, New Jersey, 1989.

[13] C. C. Mattax and R. L. Dalton. Reservoir simulation. In *SPE Monograph Series, volume 13*, Richardson, Texas, 1990.

[14] M. Parashar, J. A. Wheeler, J. C. Browne, G. Pope, K. Wang, and P. Wang. A new generation eos compositional reservoir simulator: Part ii – framework and multiprocessing. In *1997 SPE Reservoir Simulation Symposium*, Houston, Texas, 1997. SPE 37977.

[15] D. W. Peaceman. *Fundamentals of numerical reservoir simulation.* Elsevier Scientfic Publishing Company, Amsterdam-Oxford-New York, first edition, 1977.

[16] D. W. Peaceman. Interpretation of well-block pressure in numerical reservior simulation with non-square grid blocks and anisotropic permeability. *Tran. AIME*, 275:10–22, 1983.

[17] M. Peszynska, Q. Lu, and M. F. Wheeler. Coupling different numerical algorithms for two phase fluid flow. In *Proceedings for the Math. of Finite Elements and Application X, MAFELAP 1999*, Uxbridge, Middlesex, UK, Jun 1999.

[18] D. K. Ponting, B. A. Foster, P. F. Naccache, M. O. Nicholas, R. K. Pollard, J. Rae, D. Banks, and Walsh S. K. An efficient fully implicit simulator. In *European Offshore Petroleum Conference and Exihibition*, 1980.

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

[20] J. A. Trangenstein and J. B. Bell. Mathematical structure of the black-oil model for petroleum reservoir simulation. *SIAM J. Appl. Math.*, 49(3):749–783, 1989.

[21] P. Wang, I. Yotov, M. Wheeler, T. Arbogast, C. Dawson, M. Parashar, and K. Sephernoori. A new generation eos compositional reservoir simulator: Part i – formulation and discretization. In *1997 SPE Reservoir Simulation Symposium*, Houston, Texas, 1997. SPE 37979.

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

[23] I. Yotov. 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, 1997.