On the development of a high-performance tool for the simulation of CO_2 injection into deep saline aquifers

Craig Douglas[1], Frederico Furtado[2], Victor Ginting[2], Marcos Mendes[2], Felipe Pereira[1]*, and Mohammad Piri[3]

[1]University of Wyoming, Department of Mathematics and School of Energy Resources, 1000 E. University Ave., Laramie, Wyoming 82071-3036, U.S.A.

[2]University of Wyoming, Department of Mathematics, 1000 E. University Ave., Laramie, Wyoming 82071-3036, U.S.A.

[3]University of Wyoming, Department of Chemical and Petroleum Engineering, 1000 E. University Ave., Laramie, Wyoming 82071-3295, U.S.A.

**Correspondence should be addressed to: lpereira@uwyo.edu*

ABSTRACT

We report on the development of a multiscale parallel simulator for porous media flow problems. We combine state-of-the-art numerical techniques in a new object-oriented, high-performance simulation tool. The new multiscale parallel software will adapt itself to the type and number of available processing cores. The combination of:

- physically based operator splitting for multiscale time discretization of nonlinear systems of partial differential equations arising in multiphase flows in porous media,
- domain decomposition for the parallel solution of elliptic and parabolic problems, and
- semi-discrete central finite volume schemes for hyperbolic systems

allows us to produce new very accurate simulations of multiphase flow in porous media problems that are of interest in many areas of science and technology, such as petroleum reservoir and environmental engineering. The new simulation code may aid the assessment and monitoring of CO_2 sequestration projects by providing accurate predictions of the migration and trapping of injected CO_2 plumes.

KEYWORDS: CO_2 injection, CPU/GPU clusters, high-performance computing, operator splitting, saline aquifers.

INTRODUCTION

Scientifically correct models based on first principles and accurate numerical simulators are of utmost importance in producing reliable predictions of multiphase flows in the subsurface. Traditional simulations of multiphase flows rely on ad hoc up-scaling techniques along with coarse grid simulations of the up-scaled models. The models for multiphase flows are defined at the lab (or Darcy) scale (a few centimeters) while simulations of interest to important problems (such as the migration, trapping, and possible leakage of CO_2 plumes in the subsurface; enhanced oil recovery; or production of gas from unconventional resources) have to be performed in the field scale (a few kilometers). Figure 1 indicates the length scales that are of importance for the modeling of flows through porous media. Depending on the spatial scales involved, there are well-established theories to perform the upscaling from a finer to a coarser scale. For example, in going from the atomic to the pore scale one could use Statistical Mechanics. However, in going from the lab to the field scale, ad hoc up-scaling techniques frequently developed and tested for some flow regimes for single- or two-phase problems may produce serious errors when applied to more complex compositional flows. There are not rigorous theories to perform this up-scaling: difficulties in deriving up-scaled models arise both from the highly nonlinear fluid-fluid interactions in the flow of multiphase fluids and from the

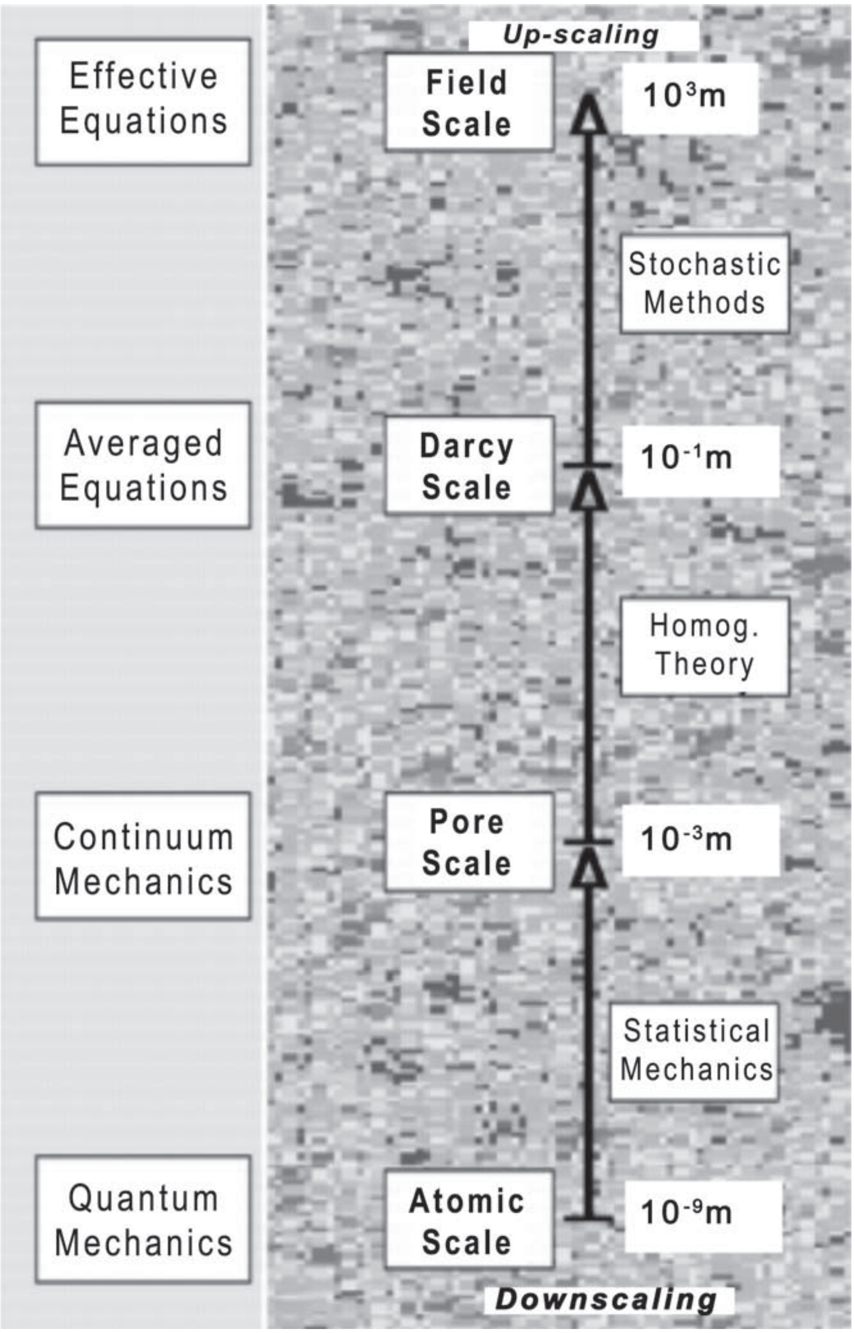


Figure 1. Important length scales for the modeling of flows through porous media along with theories that allow up-scaling from a shorter to a longer length scale. Depending on the scales involved there are well-established theories to up-scale between scales. We are concerned with models defined at the Darcy scale for making predictions in the field scale. There are no rigorous theories to perform this up-scaling. The authors believe that the field-scale behavior should be determined by a combination of fine-grid simulations along with stochastic models for the underlying heterogeneity.

complex interactions between heterogeneities and nonlinearities (see Furtado and Pereira, 2003).

Although up-scaled solutions may be important in some areas, e.g., in identifying trends of flow patterns in oil reservoir simulation, there are important practical problems where the fine-scale details of the numerical simulations should be numerically captured. We mention, for example, the simulation of injected CO_2 leakage from saline aquifers: the fine-scale preferential paths for flow that may lead to leakage cannot be captured by coarse grid, up-scaled simulations. We use state-of-the-art numerical techniques aiming at solving large-scale multiphase flows directly in fine computational grids that properly resolve the underlying physical heterogeneities. Here we report on the development of software for a Linux cluster. This is the first step toward the use of hybrid central processing unit/graphics processing unit (CPU/GPU) machines, which are becoming common in computational science environments. Within each processing unit of such systems, memory and computing power vary: while most of the memory of such devices is connected to the CPUs, most of the actual computations take place in the GPUs. GPUs have less memory but are much faster by up to one or two orders of magnitude. Such state-of-the-art hybrid machines are useful in studies aimed at quantifying the underlying uncertainty of field-scale, realistic models for multiphase flows. In such studies thousands of simulations have to be performed to produce reliable predictions in a statistical, Bayesian-type framework (Glimm and Sharp, 1999; Efendiev et al., 2005; Douglas et al., 2006).

To put the work to be described here in perspective, we remark that there is a difference in the algorithms we use based on the number of cores or processors available. There are three distinct ranges of cores that are important:

- Range 1: 1–16 cores, suitable for desktops or laptops based on multicore processors (CMPs).
- Range 2: 64–256 cores, suitable for small clusters or large shared memory multiprocessors (SMPs).
- Range 3: 10,000+ cores, suitable for use on large clusters that could be petascale in performance.

Most commercial software that is widely used for field-scale simulations of porous media flow problems is in range 1. This is by far the most common computing platform in use at corporations that purchase products instead of developing in-house software. Eclipse is a good example for this range. As clusters become commonplace in engineering labs, many research groups have pursued range 2. The computational tool described in this paper fits within this range. We are in the process of developing new optimized kernels for range 3. Most numerical methods currently in use for porous media flow problems do not scale well in this range of cores. It is far more common to run multiple problems in parallel as an ensemble with some sort of statistical filtering method that merges the parallel predictions into a common analytic prediction during a time step. Making the code switch automatically among the three ranges of cores will provide a multiscale parallel code that will be useful on all three ranges of cores. Our research group at the University of Wyoming (UW) has recently installed a new 8 CPU/64 GPU cluster that has 15,360 GPU cores that will be useful for testing different levels of parallel scaling. We note that recent scaling results on machines like Texas Advanced Computing Center's (TACC's) Ranger system show that codes that scale well at 10,000–20,000 cores seem to scale well with 60,000 cores with little or no changes. A vital understanding of the code at 10,000 cores seems to indicate that the parallelism is understood to a point that the code should work well at the order of 100,000 cores unless there is a radical change of architecture in the near future, which is distinctly possible with all of the recent accelerator (GPU and field-programmable gate array (FPGA)) announcements for enhancing traditional computers inexpensively (Exascale, 2010).

We refer the reader to Schnaar and Digiulio (2009) for a recent review of the simulation tools that have been applied in the study of CO_2 sequestration. Most of them are designed for ranges 1 or 2 indicated above. None of these codes can make effective use of all three ranges of cores. Moreover they cannot make effective use of the new, state-of-the-science hybrid CPU/GPU systems.

In the UW-team simulator to be described here, we combine state-of-the-art numerical techniques in a new object-oriented, high-performance simulation tool. Our new computational tool is based upon: (1) physically based operator splitting for multiscale time discretization of nonlinear systems of partial differential equations arising in multiphase flows in porous media; (2) domain decomposition for the numerical solution of elliptic and parabolic problems; and (3) semi-discrete central finite volume schemes for hyperbolic systems. The new simulator allows us to produce new and accurate simulations of multiphase flows in multiscale heterogeneous porous media.

The new simulation code may aid the assessment and monitoring of CO_2 sequestration projects by providing accurate predictions of the migration and trapping of injected CO_2 plumes.

The impetus of this study was our participation in a geological site characterization of the Moxa Arch in southwestern Wyoming. This interdisciplinary project, funded by the U.S. Department of Energy's National Energy Technology Laboratory, recognizes the potential for geologic sequestration of impure carbon dioxide in deep reservoirs of the Moxa Arch in Wyoming. The Moxa Arch is a 120-mile-long (~120 km) north–south-trending anticline plunging beneath the Wyoming thrust belt on the north and bounded on the south by the Uinta Mountains. Several oil and gas fields along the Moxa Arch contain natural accumulations of CO_2. The largest of these is the La Barge Platform, which encompasses approximately 800 mi^2. Several formations may be suitable for storage of impure CO_2 gas, foremost among them the Madison Limestone and Nugget Sandstone. The simulator presented here will provide an accurate computational tool to monitor plume migration if CO_2 is injected into aquifers, whether in Wyoming or elsewhere.

This paper is organized as follows. In the next section we define the model that we use. Then, we review operator-splitting methods that are applied to our model. Central schemes for solving hyperbolic systems of conservation laws are discussed and the object-oriented software design we use is described. In the last section we provide numerical evidence that our methodology works.

MODELING

We use the model proposed in Obi and Blunt (2006) to perform numerical simulations of the injection of CO_2 in saline aquifers. Our focus is on the primary physical processes that affect the transport of two components, CO_2 and water, in a two-phase system encountered during the injection of supercritical CO_2 into deep saline aquifers. Most of the injected CO_2 will initially reside in its own phase, which will contact and displace the resident fluid (brine). The displacement of the resident brine is controlled by the relative permeability functions, viscosities, and densities of the CO_2 and aqueous (brine) phases, as well as by the spatial structure of the permeability and porosity fields. Some CO_2 will dissolve into the aqueous phase, and the extent of dissolution is controlled by pressure and temperature alone; we neglect the effect of salinity in dissolution. We also ignore evaporation of water from the aqueous phase to the CO_2 phase.

Our focus is on the relatively short-term period of injection, so we emphasize convective processes. Slower flow processes, such as diffusion and buoyancy effects of dissolved species, are not included. Chemical reactions between acidic brine and supercritical CO_2 with rock minerals and non-isothermal effects are also ignored.

We assume incompressible two-phase flow and neglect capillary pressure and dispersive effects. Supercritical CO_2 is relatively incompressible in the high-pressure (well above the critical pressure) conditions of deep aquifers that we intend to simulate. Moreover, most deep consolidated rocks also have low compressibility, so we ignore mechanical stress effects.

The governing equation for transport of CO_2 in its own phase (indicated by a subscript *c*) is then given by the following (volume) balance equation:

$$\phi\frac{\partial S_c}{\partial t}+\nabla\bullet\left(uF_c-k\frac{\lambda_c\lambda_a}{\lambda}(\rho_a-\rho_c)g\nabla d\right)=-T_d+q_c, \quad (1)$$

where ϕ is porosity, k is absolute permeability, S_c is the saturation of the CO_2 phase, u is the total Darcy velocity, T_d represents the transfer of CO_2 to the aqueous phase due to dissolution, and q_c denotes sinks and sources. ρ_α denotes the density of phase α (*c* for CO_2 and *a* for aqueous), the parameter *d* represents depth, and *g* is the acceleration of gravity. The total mobility $\lambda=\lambda_c+\lambda_a$ is the sum of the phase mobilities, the latter defined in terms of the phase relative permeabilities k_α and the phase viscosities μ_α as follows: $\lambda_\alpha=k_\alpha/\mu_\alpha$.

Moreover, the fractional flow functions are given by $F_\alpha=\lambda_\alpha/\lambda$.

The transport equation for the CO_2 dissolved in the aqueous phase is:

$$\phi\frac{\partial(1-S_c)C}{\partial t}+\nabla\bullet C\left(u(1-F_c)+k\frac{\lambda_c\lambda_a}{\lambda}(\rho_a-\rho_c)g\nabla d\right)=m_cT_d. \quad (2)$$

Here C is the concentration of CO_2 in the aqueous phase, measured in moles per unit volume, and m_c is the molar density of CO_2.

The two equations above can be combined with the transport equation for the aqueous phase to yield an equation that expresses the incompressibility of the two-phase system:

$$\nabla\bullet u=q_c. \quad (3)$$

This last equation can be viewed as a "pressure equation" once the total velocity *u* is related to the pressure *p* by a multiphase version of Darcy's law:

$$u=-k\left(\lambda\nabla p-\left(\rho_c\lambda_c+\rho_a\lambda_a\right)g\nabla d\right). \quad (4)$$

The specification of appropriate initial and boundary conditions completes the description of our physical model. Sample boundary conditions are contained later in this paper in the Numerical Experiments section.

OPERATOR SPLITTING

Operator-splitting schemes have recently been developed for the approximation of multiphase flows in porous media (Abreu et al., 2006). These schemes are aimed at decomposing the underlying physical processes and treating each such component appropriately. Thus, instead of solving the governing system of differential equations in the form which results directly from the basic conservation laws supplemented by constitutive relations (or other fundamental equations), the equations are rewritten in such a way as to exhibit clearly each physical process. Then they are solved by appropriate numerical techniques that furnish effective and efficient schemes designed to resolve the sharp gradients and dynamics evolving at vastly different rates that are the hallmarks of multiphase flows in porous media.

In addition to the ability to vary the numerical schemes with the physics, operator splitting enables the use of multiscale time step discretizations: different time steps are used for the different physical processes. These properly chosen time steps lead to a significant reduction in overall computing times. This set of ideas has been successfully applied to several multiphase flow problems: multicomponent gas flow (Douglas et al., 2003), two-phase flows in unfractured and fractured porous formations (Douglas, 1997a; Douglas, 1997b; Douglas et al., 2000), and three-phase flows (Abreu et al., 2006; Abreu et al., 2008; Abreu et al., 2009).

For the simulation of the injection of CO_2 into saline aquifers, we employ an operator-splitting technique based on separating the underlying physical processes and treating each such process appropriately to define the mathematical model that was investigated in this project. The obvious primary physics-based splitting is the separation of the pressure-velocity calculations from the saturation-concentration calculation. Since only the total Darcy velocity u is needed in the transport equations, a natural choice of a numerical method for the pressure equation is a mixed finite element method, well known for producing accurate velocity fields in the presence of rapidly varying coefficients (e.g., Douglas, 1997a) so that u is approximated directly.

The secondary splitting is to separate the convective transport from the dissolution process. For the numerical solution of the nonlinear system of conservation laws that is associated with convective transport, we employ higher-order, conservative, central finite difference schemes to capture the sharp fronts that typically develop in this problem. We shall discuss the recent development of these schemes later in the paper. Finally, the dissolution of CO_2 into the aqueous phase is accounted for using Henry's law (Smith et al., 2005) and assuming ideal mixing.

CENTRAL DIFFERENCE SCHEMES FOR MULTIPHASE FLOWS

Recently, high-resolution, semi-discrete central difference schemes have been used for the solution of systems of hyperbolic conservation laws arising in the simulation of multiphase flows in multidimensional heterogeneous porous media. These schemes have some desirable properties for the approximation of porous media flow, which we shall discuss below.

The Lax-Friedrichs (LxF) scheme (Lax, 1954) is the canonical first-order central scheme, which is the forerunner of all central differencing schemes. It is based on piecewise constant approximate solutions. It also enjoys simplicity, that is, it does not employ Riemann solvers and characteristic decomposition. Unfortunately, the excessive numerical dissipation in the LxF scheme (of order $O(\Delta X^2/\Delta t)$) yields poor resolution, which seems to have delayed the development of high-resolution central schemes when compared with the earlier developments of the high-resolution upwind methods. Only in 1990 did Nessyahu and Tadmor (NT) introduce a second-order generalization to the LxF scheme (Nessyahu and Tadmor, 1990). They used a staggered form of the LxF scheme and replaced the first-order piecewise constant solution with a van Leer's Monotone Upstream-centered Schemes for Conservation Laws (MUSCL)-type piecewise linear second-order approximation (Leer, 1979). The numerical dissipation in this new central scheme has an amplitude of order $O(\Delta X^4/\Delta t)$. When applying these methods to multiphase flows in highly heterogeneous aquifers we need to use decreasing time steps as the heterogeneity increases, yielding greater numerical diffusion. Kurganov and Tadmor (KT; Kurganov and Tadmor, 2000) combined ideas from the construction of the NT scheme with Rusanov's method (Rusanov, 1961) to obtain the first second-order central scheme that admits a semi-discrete formulation, which is then solved with an appropriate ordinary differential equation (ODE) solver. The resulting scheme exhibits a much smaller numerical diffusion than the NT scheme. In fact, due to the semi-discrete formulation, this numerical diffusion is independent of the time step used to evolve the ordinary differential equation. This property guarantees that no extra numerical diffusion will be added if the time step is forced to decrease. The KT scheme for multidimensional problems is based on a dimensional splitting scheme. Some of the authors and their collaborators have recently derived a multidimensional version of the KT scheme that retains the advantages of the earlier semi-implicit procedures while avoiding dimensional splitting (see also Kurganov and Petrova, 2001 and Balbas and Qian, 2009 for related efforts). In the numerical simulations reported here, we will use the original KT schemes, because some preliminary results indicate that the dimension-by-dimension approach leads to a much better speed up of the numerical solution produced in a hybrid CPU/GPU system (Pereira and Rahunanthan, 2010).

SOFTWARE DESIGN

To explain the framework of the software developed, we first discuss the main features that simulation software must have for reliable computational modeling of multiphase flows in porous media. These were the main principles that guided us during the software design phase.

One of the main characteristics about this type of simulation software is that some of its requirements change often. This happens because the computational modeling of physical systems is an iterative procedure. We begin by describing the physical phenomena with simple equations that model the most fundamental processes in the problem with trivial constitutive laws. This original system is incremented in subsequent iterations by increasing the complexity of the mathematical model. This has a great impact on the software requirements, because numerical methods must be added or extended in the code to solve the new system of equations.

Another interesting aspect of the software design is that the same set of equations can be solved by many numerical methods with different degrees of programming labor, computational efficiency, and precision. In fact, we often implement many methods to solve the same set of equations. By reproducing verified results available in the referenced literature we can investigate which method is better suited for the class of problems of interest.

Due to frequent additional requirements for our simulation software, we must be able to add new numerical methods written by different collaborators while preserving the correctness of the code. Although efficiency is a desirable feature of any scientific software, we also have to carefully consider the relative cost of efficiency and code maintenance.

The considerations presented above motivated the UW team to build a simulator in modules that can be chosen by the user in a configuration file. Each module corresponds to a specific numerical method used to solve a specific set of equations. This framework is easily implemented in the context of C++ template-based object-oriented programming.

In the simulator developed for CO_2 sequestration, the governing system of partial differential equations was decomposed through an operator-splitting procedure in three sets of equations (this is our mathematical model described above):

- *Transport system*: It is composed of the transport equations of the fluid phases, for example, the supercritical CO_2, the brine, and the water vapor. It is generally formulated in terms of the phase saturations.

- *Poro-mechanics system*: It defines the tension interaction between the fluids and the porous media. The main variables are the pressure of the fluids and the solid displacement in the case of deformable media. In the current stage of the project we have worked only with rigid porous media.

- *Thermodynamics system*: It defines the equilibrium flash calculations used to define the chemical interactions among the components in the phases (solid, brine, and supercritical CO_2).

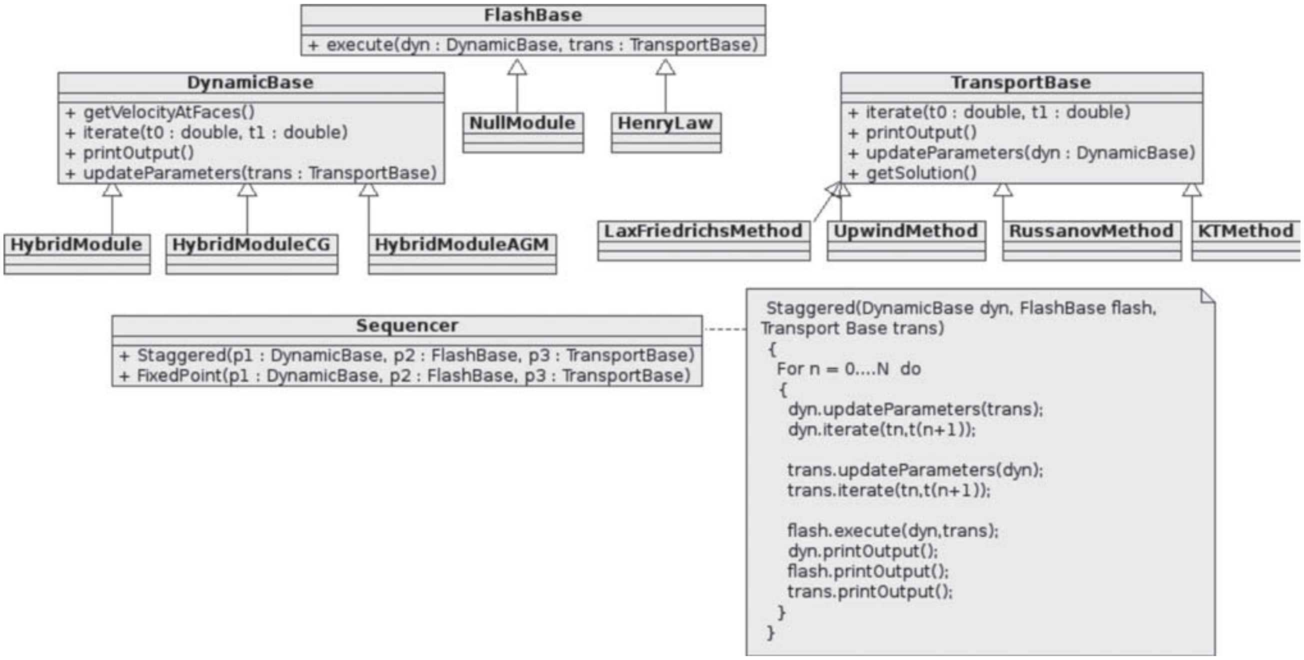
For each subsystem there is a family of modules. To solve the transport system, for instance, there are three modules corresponding to four numerical schemes: Lax-Friedrichs, Upwind, Russanov, and Kurganov-Tadmor (KT). For the poro-mechanics we implemented the mixed and the mixed-hybrid finite element methods that can be combined with three linear solvers: algebraic multigrid (AMG), a direct frontal solver, the unsymmetric multifrontal sparse LU factorization package (UMFPACK), and conjugate gradient with a symmetric successive overrelaxation method (SSOR) preconditioner. We also used the message passing interface library (MPI) to implement an iterative domain decomposition scheme (Douglas et al., 1997a) for the parallel implementation of the mixed hybrid finite element method in Linux clusters. The thermodynamics system has just two modules: a null module (trivial thermodynamic equilibrium without dissolution) and another one based on Henry's Law (Obi and Blunt, 2006).

The subsystems are coupled among themselves since the primary variables in one subsystem can be the input parameters for other modules in a dataflow programming style. Once a module is chosen for each subsystem, the set of modules needs to evolve through the simulation time in a consistent manner using a staggered algorithm that exchanges module solutions every time step. We have implemented two algorithms: one that simply evolves the subsystems without checking for convergence between them and another based on the search for a fixed point. These algorithms receive as parameters the three modules, one for each subsystem. The unified modeling language (UML) diagram (Fig. 2) illustrates the framework. The base classes *FlashBase*, *DynamicBase*, and *TransportBase* define the interface among the modules that implement, respectively, the three subsystems: thermodynamics, poromechanics, and transport. This design favors encapsulation and teamwork. For example, to add a new transport method a collaborator just creates a new class inherited from *TransportBase* and overrides its four methods. Constant parameters (in time) like the permeability field, initial conditions, and time-independent boundary conditions are passed to the classes on instantiation.

The UML diagram shows some of the methods already implemented:

- For transport methods we have the classes *LaxFriedrichsMethod*, *UpwindMethod*, *RussanovMethod,* and the *KTMethod*. Each one corresponds to a specific finite volume method.
- For the poromechanics modules we have *HybridModule*, *HybridModuleCG,* and *HybridModuleAGM*. All of them discretize the poromechanics subsystem using the hybridized mixed finite element method. They differ only by the type of linear solver they employ. We also have implemented a domain decomposition version of each one of the poromechanics modules (not shown in the UML diagram).

When the program runs, it instantiates one of the modules for each type of subsystem and calls a procedure of the sequencer class while passing the instantiated classes as parameters. The UML diagram shows the code of one of the sequencer's methods called *Staggered*. This method simply advances the modules in time and exchanges their solutions each time step. For the modules based on finite element methods, we use the deal.II C++ library (Differential Equations Analysis Library) that implements a wide range of finite element methods for 3D quadrilateral non-structured meshes. It is bundled with a large collection of iterative linear solvers that we use extensively. We also used the UMFPACK (Davis et al., 2010) for debugging purposes. For data output we used a data input/output system, HDF5, which is designed specifically for scientific data (HDF, 2010); it is a versatile data model that


FlashBase
+ execute(dyn : DynamicBase, trans : TransportBase)
DynamicBase
+ getVelocityAtFaces()
+ iterate(t0 : double, t1 : double)
+ printOutput()
+ updateParameters(trans : TransportBase)
NullModule
HenryLaw
TransportBase
+ iterate(t0 : double, t1 : double)
+ printOutput()
+ updateParameters(dyn : DynamicBase)
+ getSolution()
HybridModule
HybridModuleCG
HybridModuleAGM
LaxFriedrichsMethod
UpwindMethod
RussanovMethod
KTMethod
Sequencer
+ Staggered(p1 : DynamicBase, p2 : FlashBase, p3 : TransportBase)
+ FixedPoint(p1 : DynamicBase, p2 : FlashBase, p3 : TransportBase)

```
Staggered(DynamicBase dyn, FlashBase flash,
Transport Base trans)
 {
  For n = 0....N  do
  {
   dyn.updateParameters(trans);
   dyn.iterate(tn,t(n+1));

   trans.updateParameters(dyn);
   trans.iterate(tn,t(n+1));

   flash.execute(dyn,trans);
   dyn.printOutput();
   flash.printOutput();
   trans.printOutput();
  }
 }
```

Figure 2. UML class diagram for the software architecture.

can represent very complex data objects and a wide variety of metadata commonly used on high-performance computing systems. It supports a large variety of data types and is designed for flexible and efficient input/output and for large volumes of complex data.

NUMERICAL EXPERIMENTS

We first present the result of a grid refinement study for the numerical simulation of the injection of supercritical CO_2 in a three-dimensional aquifer having 128 meter (m) versus 64 m versus 64 m. The CO_2 is injected through two horizontal wells located at 32 m from the bottom of the aquifer. Dirichlet conditions are imposed for the pressure on the lateral (left and right) boundaries and no-flow conditions are imposed on the other boundaries. Due to the effect of gravity the plume is displaced upwards. In the simulations, the reservoir initially contains 100 percent brine. CO_2 is injected into the aquifer at a constant rate of 93 m^3 per day for 3.3 years. We consider a scalar, heterogeneous absolute permeability field with six horizontal layers. The constant-by-parts absolute permeability field is defined on a 64 versus 32 versus 32 uniform grid. Each layer is taken to be (the log of) a realization of a (Gaussian) random fractal field (see Glimm et al., 1992, 1993) with moderately large heterogeneity strength where the mean permeability value is prescribed as 100 millidarcy. The data below are held fixed in all flow studies:

- Viscosity: 5 10^{-4} Pas (aqueous); 6 10^{-5} Pas (CO_2)
- Porosity: 0.3
- Residual saturations: $s_{ra} = 0; s_{rc} = 0$
- Relative permeability functions: $k_{ra}(S_c) = (1 - (1 - s_{ra})^{-1} S_c)^2; k_{rc}(S_c) = (1 - s_{rc})^{-2}(S_c - s_{rc})^2$

We use the same parameters for Henry's Law as those values given in Obi and Blunt (2006). Figure 3 displays the result of a grid refinement study where the dissolution was not taken into account. The grids used are finer in going from top to bottom, and have 64 versus 32 versus 32 (top), and 128 versus 64 versus 64 (bottom) elements. CO_2 saturation surface plots are shown after 1200 days of simulation. The numerical results indicate that the coarse and fine grid results are quite similar; this is an indication that an adequate level of numerical convergence has been achieved in the coarser grid simulation. The numerical results for both grids used show perfect mass conservation.

The coarse grid simulation depicted on Figure 4 takes dissolution into account. In Figure 4 the plume on the top was simulated without dissolution, and the plume on the bottom was simulated with dissolution. As expected, the volume of the plume simulated with dissolution is smaller.

CONCLUSIONS

We remark that details of the implementation of the software in both multiple CPU and multiple hybrid CPU/GPU computer systems will appear elsewhere. The authors and their collaborators intend to continue the development of the software described here. We intend to add a geomechanics module and investigate the robustness of our splitting procedure in the approximation of extremely difficult problems involving strong couplings, e.g., porosity and permeability dependence on CO_2 acidity, that can imply wormhole formation processes.

ACKNOWLEDGMENTS

This research was supported in part by DOE grant DE-NT0004730-Task9, NSF grants CNS-1018072 and CNS-1018079, and Award No. KUS-C1-016-04, made by King Abdullah University of Science and Technology (KAUST). The authors wish to thank the Guest Editor Professor C. Frost for her helpful suggestions. This manuscript was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

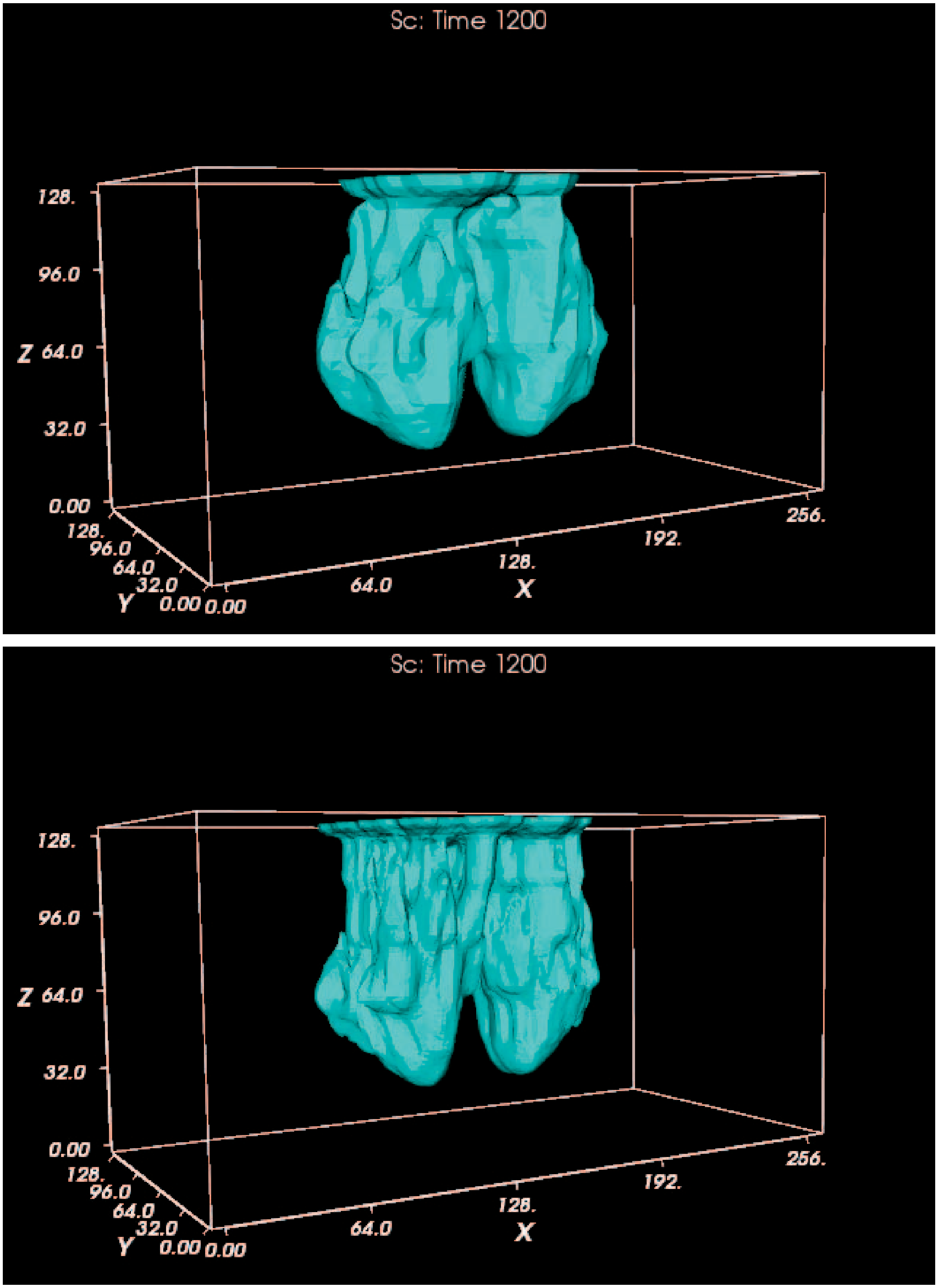


Figure 3. CO_2 injection in a heterogeneous, 3D brine aquifer. The pictures show the saturation of the injected CO_2 in a mesh refinement study. The picture on the top refers to a coarser computational grid; the picture on the bottom refers to a finer computational grid. As the computational grid is refined one can observe finer-scale details of the injected CO_2 plume.

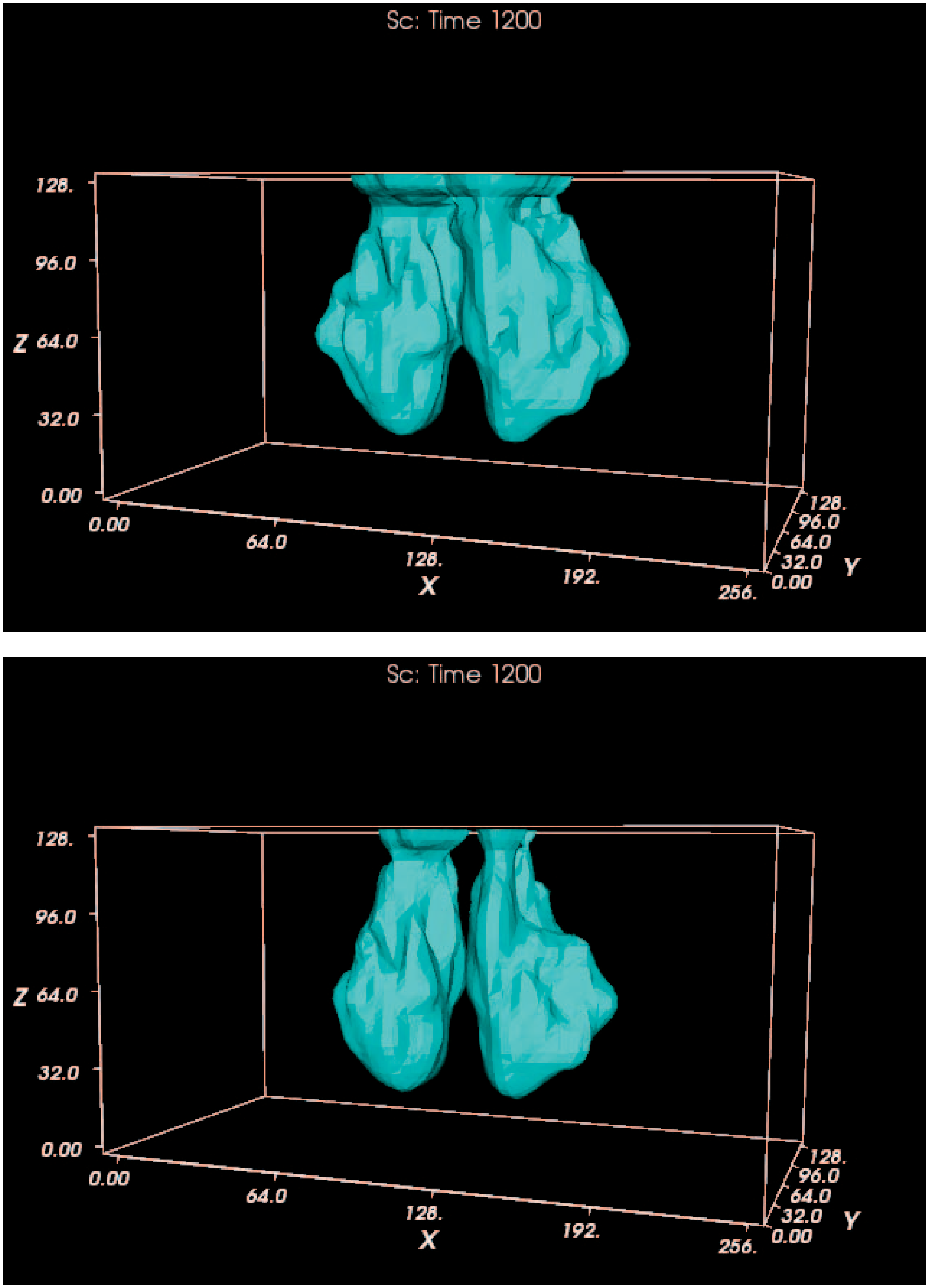


Figure 4. Simulated plumes without dissolution (top picture) and with dissolution (bottom picture). As expected, the volume of the plume in the bottom picture is smaller because a fraction of the injected CO_2 is dissolved in the resident brine phase.

REFERENCES CITED

Abreu, E., Douglas, J., Jr., Furtado, F., and Pereira, F., 2008, Operator splitting based on physics for flow in porous media: International Journal of Computational Science, v. 2, p. 315–335.

_____2009, Operator splitting for three-phase flow in heterogeneous porous media: Communications in Computational Physics, v. 6, p. 72–84.

Abreu, E., Douglas, J., Jr., Furtado, F., Marchesin, D., and Pereira, F., 2006, Three-phase immiscible displacement in heterogeneous petroleum reservoirs: Mathematics and Computers in Simulation, v. 73, p. 2–20.

Balbas, J., and Qian, X., 2009, Non-oscillatory central schemes for 3D hyperbolic conservation laws: Hyperbolic Partial Differential Equations, Theory, Numerics and Applications, *in* Proceedings of the 12th International Symposia in Applied Mathematics, College Park, University of Maryland.

Davis, T., Duff, I., Amestoy, P., Larimore, S., Gilbert, J., Ng, E., and Hadfield, S., 2010, UMFPACK, http://www.cise.ufl.edu/research/sparse/umfpack.

Douglas, C., Efendiev, Y., Ewing, R., Ginting, V., and Lazarov, R., 2006, Dynamic data driven simulations in stochastic environments: Computing, v. 77, p. 321–333.

Douglas, J., Furtado, F., and Pereira, F., 1997a, On the numerical simulation of waterflooding of heterogeneous petroleum reservoirs: Computational Geosciences, v. 1, p. 155–190.

Douglas, J., Pereira, F., and Yeh, L. M., 1997b, A parallelizable method for two-phase flows in naturally fractured reservoirs: Computational Geosciences, v. 1, p. 333–368.

_____2000, A locally conservative Eulerian-Lagrangian numerical method and its application to nonlinear transport in porous media: Computational Geosciences, v. 4, p. 1–40.

Douglas, J., Frias, D., Henderson, N., and Pereira, F., 2003, Simulation of single-phase multicomponent flow problems in gas reservoirs by Eulerian-Lagrangian techniques: Transport in Porous Media, v. 50, p. 307–342.

Efendiev, Y., Datta-Gupta, A., Ginting, V., Ma, X., and Mallick, B., 2005, An efficient two-stage Markov chain Monte Carlo method for dynamic data integration: Water Resources Research, v. 41, W12423.

Exascale, 2010, International Exascale software project, White paper, http://www.exascale.org.

Furtado, F., and Pereira, F., 2003, Crossover from nonlinearity controlled to heterogeneity controlled mixing in two-phase porous media flows: Computational Geosciences, v. 7, p. 115–135.

Glimm, J., and Sharp, D., 1999, Prediction and the quantification of uncertainty: Physica D, v. 133, p. 152–170.

Glimm, J., Lindquist, W. B., Peierls, R., and Pereira, F., 1992, The fractal hypothesis and anomalous diffusion: Computational and Applied Mathematics, v. 11, p. 75–93.

Glimm, J., Lindquist, W. B., Pereira, F., and Zhang, Q., 1993, A theory of macrodispersion for the scale-up problem: Transport in Porous Media, v. 13, p. 97–122.

HDF, 2010, The HDF group, http://www.hdfgroup.org/HDF5.

Kurganov, A., and Petrova, G., 2001, A third-order semi-discrete genuinely multidimensional central scheme for hyperbolic conservation laws and related problems: Numerische Mathematik, v. 88, p. 683–729.

Kurganov, A., and Tadmor, E., 2000, New high-resolution semi-discrete central schemes for Hamilton-Jacobi equations: Journal of Computational Physics, v. 160, p. 720–742.

Lax, P., 1954, Weak solutions of non-linear hyperbolic equations and their numerical computation: Communications on Pure and Applied Mathematics, v. 7, p. 159–193.

Leer, B., 1979, Toward the ultimate conservative difference scheme V: A second-order sequel to Godunov's method: Journal of Computational Physics, v. 32, p. 101–136.

Nessyahu, H., and Tadmor, E., 1990, Non-oscillatory central differencing for hyperbolic conservation laws: Journal of Computational Physics, v. 87, p. 408–463.

Obi, E., and Blunt, M., 2006, Streamline-based simulation of carbon dioxide storage in a North Sea aquifer: Water Resources Research, v. 42, W03414.

Pereira, F., and Rahunanthan, A., 2010, Numerical simulation of two-phase flows on a GPU, *in* Proceedings, High Performance Computing for Computational Science, 9th international meeting (VECPAR 2010), Berkeley, California.

Rusanov, V., 1961, The calculation of the interaction of non-stationary shock waves with barriers: Journal of Computational Mathematics and Mathematical Physics USSR, v. 1, p. 267–279.

Schnaar, G., and Digiulio, D., 2009, Computational modeling of the geologic sequestration of carbon dioxide: Vadose Zone Journal, v. 8, p. 389–403.

Smith, J., Van Ness, H., and Abbott, M., 2005, Introduction to chemical engineering thermodynamics: New York, McGraw-Hill, 7th edition, 817 p.

Manuscript Submitted April 8, 2010

Revised Manuscript Submitted June 28, 2010

Manuscript Accepted July 29, 2010