Partitioning Algorithms and Their Application to Massively Parallel Computations of Multi-Phase Fluid Flows in Porous Media

A Final Report to the U.S. Environmental Protection Agency

Date of Final Report: October 20, 2000

EPA Agreement Number: R 825207

Title: Partitioning Algorithms and Their Application to Massively Parallel Computations of Multi-Phase Fluid Flows in Porous Media

Investigators: **Principal Investigator:**
Richard E. Ewing, Texas A&M University

Co-Principal Investigators:
Hristo Djidjev, Rice University
Raytcho Lazarov, Texas A&M University
Moshe Vardi, Rice University

Institutions: Texas A&M University
Rice University

1   OBJECTIVES OF THE RESEARCH PROJECT

This project focused on the development of methods for simulation of multi-phase and multi-component fluid flows in porous media that combine recent advances in numerical methods with techniques for partitioning and load balancing based on graph theory and graph algorithms. According to the research plan, we have concentrated our efforts and achieved our goals in the following interrelated groups of problems, which combine theoretical investigations with computational experimental work:

- Construction, analysis, and implementations of finite volume element methods for reactive flows in porous media,
- Local error estimators and refinement strategies for finite volume methods,
- Algorithms for partitioning and load balancing,
- Domain decomposition for parallel computations in fluid flow modeling, and
- Testing the developed methods on fluid flows in porous media.

The first four points are connected in the following computational strategy. We use a mesh generator (triangle for 2-D meshes and NETGEN for 3-D meshes) to generate a good coarse mesh. Then the considered problem is solved sequentially on the coarse mesh (by every processor). The solution is used to compute *a posteriori* error estimates, which are used as weights in an element-based

splitting of the coarse mesh into sub-domains. Such splitting ensures that the local refinements that follow will produce a computational mesh with a number of triangles/tetrahedrons balanced over the sub-domains. Every sub-domain is "mapped" to a processor. Then, based on *a posteriori* error analysis, each processor refines consecutively its region independently. After every step of independent refinement, there is communication between the processors to make the mesh on that level globally conforming. The last point concentrates on computer simulation and testing the developed techniques on fluid flows in porous media. In Section 3 we give the results from applying the developed computational methodology over the so-called *Bioscreen Problem*.

Additional information about this research project, including examples of results, can be found on the Internet at http://www.isc.tamu.edu/EPA.

2   SUMMARY OF THE RESEARCH

2.1   Mathematical Model

The mathematical model of steady-state ground-water flows and transport in porous media yields two basic equations. These are the Darcy equation for pressure, discussed in Subsection 2.1.1, and the advection-dispersion (transport) equation, discussed in Subsection 2.1.2. The transport equation describes the steady-state distribution of a passive substance dissolved in the water, transported by the flow, and absorbed by the soil. Further, this method can be extended to the case of transport of multiple chemicals that react. Since many clean-up, remediation, and exploration strategies in aquifers and petroleum reservoirs are based on treatment/injection/production through wells, we also briefly discuss various well models.

2.1.1   Diffusion (pressure) equation

The fluid flow is due to the velocity $\underline{v}$ defined by Darcy's law: $\underline{v} = -D\nabla p$, where p is the pressure, and D is the permeability of the porous media. The pressure p satisfies the following equation subject to various boundary conditions:

$$\begin{cases} \nabla \cdot \underline{v} \equiv -\nabla \cdot D\nabla p & = f, & \text{in } \Omega, \\ p & = p_D, & \text{on } \Gamma_D, \\ -D\nabla p \cdot \underline{n} - \gamma p & = p_N, & \text{on } \Gamma_N, \\ p & = p_w, & \text{on } \Gamma_w. \end{cases} \tag{1}$$

Here Ω is a bounded polyhedral domain in R^3 with boundary $\partial\Omega \equiv \Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_w$, D is symmetric, bounded and uniformly positive definite matrix in Ω, $\underline{n}$ is the outer unit vector normal to the boundary of Ω, p_D, p_N, and $\gamma \geq 0$ are given functions, p_w is a given constant (called well-bore pressure), and f is the given source term. The last three equations prescribe Dirichlet, Neumann, and well boundary conditions, respectively. The last one models injection/extraction of fluid through a well, which is assumed to be a cylinder with radius r_w. Since the well radius r_w is very small compared to the reservoir size, the wells can be classified as small features of the media and well boundary conditions will lead to solutions with almost singular behavior. For discussion of well boundary conditions, including nonlinear ones, we refer to [20].

2

Another boundary condition that models injection/extraction of fluid from a reservoir is well condition with a prescribed production rate Q, but with unknown pressure p_w on the well surface:

$$p = p_w, \text{ on } \Gamma_w, \ p_w \text{ unknown constant and } \int_{\Gamma_w} D\nabla p \cdot \underline{n}\,ds = Q. \tag{2}$$

Finally, on Γ_w we can prescribe the same type of boundary condition as on Γ_N. Namely, we have the boundary condition

$$-D\nabla p \cdot \underline{n} = \gamma p + \tilde{Q} \quad \text{ on } \ \Gamma_w \tag{3}$$

with $\tilde{Q}$ a given constant. Note that here $\tilde{Q}$ is the pointwise flux, while Q in (2) is the total debit of the well. For $\gamma = 0$ they are related by $Q = \tilde{Q}S_w$, where S_w is the area of the lateral surface of the well. Our computations involving well models were done for condition (3) with $\gamma = 0$.

2.1.2 Convection-diffusion-reaction (transport) equation

The second basic equation gives the concentration of a passive chemical dissolved and distributed in the water due to the processes of advection, diffusion, and absorption. The equation describes the conservation of mass of the chemical. The steady-state distribution of the concentration c is described by the following general boundary value problem for convection-diffusion-reaction equation:

$$\begin{cases} -\nabla \cdot K\nabla c + \nabla \cdot (\underline{b}c) + ac &= f, \quad \text{ in } \ \Omega, \\ c &= c_D, \quad \text{ on } \ \Gamma_D, \\ (-K\nabla c + \underline{b}c) \cdot \underline{n} &= c_N, \quad \text{ on } \ \Gamma_N^{in}, \\ -K\nabla c \cdot \underline{n} &= 0, \quad \text{ on } \ \Gamma_N^{out}, \\ c &= c_w, \quad \text{ on } \ \Gamma_w. \end{cases} \tag{4}$$

Again, Ω is a bounded polyhedral domain in R^3 with boundary $\partial\Omega$, that is split into Dirichlet, Neumann, and well parts, namely $\partial\Omega \equiv \Gamma = \Gamma_D \cup \Gamma_N \cup \Gamma_w$. Further, the Neumann boundary is divided into two parts: $\Gamma_N = \Gamma_N^{in} \cup \Gamma_N^{out}$, where $\Gamma_N^{in} = \{x \in \Gamma_N : \underline{n}(x) \cdot \underline{b}(x) < 0\}$ and $\Gamma_N^{out} = \{x \in \Gamma_N : \underline{n}(x) \cdot \underline{b}(x) \geq 0\}$. We assume that the diffusion-dispersion tensor K is a symmetric, bounded, and uniformly positive definite matrix in Ω, $\underline{b}$ is the given convection vector field, $\underline{n}$ as before is the outer unit vector normal to $\partial\Omega$, $a \geq 0$, f, c_D, c_N and $\gamma \geq 0$ are given functions. The boundary condition on Γ_w models the case of a given concentration on the well surface, which corresponds to the case of an injection well.

In our computations, we take the advection vector-field $\underline{b} = \underline{v}$, where the Darcy velocity $\underline{v}$ is obtained after solving the problem in Equation (4). Then the diffusion-dispersion tensor is given by $K = k_{diff}I + k_t\underline{v}^T\underline{v}/|\underline{v}| + k_l(|\underline{v}|^2I - \underline{v}^T\underline{v})/|\underline{v}|$, where k_{diff}, k_t, and k_l are constants characterizing correspondingly the diffusion, transverse dispersion, and longitudal dispersion.

In the case of a production well, we have to impose boundary conditions that model appropriately the extraction of the dissolved substance by the well activity. In the case when the flow is determined by the solution of problem (4) with well boundary condition (3), we get the following boundary condition for the concentration:

$$K\nabla c \cdot \underline{n} = 0 \ \text{ on } \Gamma_w. \tag{5}$$

3

2.2   Finite Volume Element Methods for Flows in Porous Media

Many clean-up, remediation, and exploration strategies in aquifers and petroleum reservoirs are based on treatments/actions through wells. Therefore, accurate numerical simulation of the well is an important problem. We have restricted our research on the following two main directions: accurate numerical well models and adaptive local grid refinement.

First, we have theoretically studied the well models for Darcy's and Forchheimer's flows treated numerically by finite differences, finite elements, and mixed finite elements. This research has resulted in a paper by Ewing, R. Lazarov, S. Lyons, D. Papavassiliou, J. Pasciak, and G. Qin, which has appeared in *Computational Geosciences* (see paper 9 of the List of Publications Resulting from This Grant). This study is a basis for our further research in 3-D problems, as well as more complex flows.

Mathematical formulations in terms of parabolic problems with integral terms in time have been used in various engineering models, such as nonlocal reactive transport in underground water flows in porous media done by Dagan [12] and Cushman and collaborators [11], and radioactive nuclear decay in fluid flows investigated by Shelukhin [19]. These equations model quite adequately the process of multi-component flows in porous media and lead to better understanding of the physical processes and their properties. A very important characteristic of all of these models is that they all express a conservation of a certain quantity (mass, heat, etc.) in any moment for any sub-domain. In many applications, this is a desirable feature of the approximation method when it comes to numerical solution, and application of the corresponding models to real life problems.

We have studied finite volume element methods for one-dimensional and two-dimensional problems of this type. Namely, we have developed a general framework for obtaining finite volume element approximations and studying their error analysis. In 1-D, we consider the lowest-order (linear and L-splines) elements. These schemes are locally conservative and have optimal approximation properties. We have shown that the finite volume element approximations are convergent with optimal order in H^1 norm, suboptimal in the L^2 norm and super-convergent in a discrete H^1 norm. In 2-D we have also studied the convergence in L^∞-norm and shown optimal error estimates. This work has been reported by Ewing, Lazarov, and Lin in paper 11 of the List of Publications Resulting from This Grant. We have implemented an upwind version of the approxiamtaion which has the same properies as the stibilized methods of [18] and at the same time has local conservation.

2.3   Local Error Estimators and Refinement Strategies

The solutions of flow and transport problems in porous media exhibit local behavior. Therefore, it is essential that local grid refinement, based on *a posteriori* error analysis, is applied. Our goal was to implement and experiment with various local error estimators and indicators, to develop adaptive finite element and finite volume codes for both 2-D and 3-D problems. The methods were based on the previous research by Babuska and Rheinboldt [4], [5]; Zienkiewicz and Zhu [23]; Becker and Rannacher [8]; Bank and Weiser [7]; Brenner [10].

We started the problem investigation by developing, implementing, and testing a 2-D grid refinement strategy based on the known error estimator and indicators. In the context of the finite element method, our work was mostly concentrated on the "h-version" adaptive refinement, where the error is reduced by adaptively refining the grid and leaving the degree of the approximating polynomials the same. The results of this research are summarized in the technical report by R. Lazarov, J. Pasciak, S. Tomov, *Error control, local grid refinement, and efficient solution algorithms for singularly perturbed problems* (this is paper 18 of the List of Publications Resulting from This

Grant). Here we show examples of a homogeneous reservoir and a non-homogeneous reservoir with two wells.

We extended the developed 2-D adaptive code to 3-D. We worked on, tested, and implemented three error indicators, namely residual based refinement (first introduced by Babuska and Rheinboldt [4]), Zienkiewicz-Zhu technique (see [23]), and hierarchical refinement (see [6], [7]). We have adapted the first two techniques for the finite volume element method. In R. Lazarov, S. Tomov, (paper 21 of the List of Publications Resulting from This Grant), *Adaptive finite volume element method for convection-diffusion-reaction problems in 3-D*, we present an adaptive numerical technique for solving steady-state diffusion and convection-diffusion-reaction equations in 3-D (including the case of dominant convection) using finite volume approximations. Computational results of various model simulations of fluid flow and transport of passive chemicals in non-homogeneous aquifers are presented and discussed in this work.

In our research we have mostly used triangular and tetrahedral meshes for the 2-D and 3-D case, correspondingly. The actual mesh refinement is important since the consecutive refinements have to be done in such way that the element shapes do not degenerate. Our adaptive 3-D mesh generation is based on the bisection algorithm (see, e.g. Arnold *et al.* [3]). In the 2-D case, the triangles marked for refinement are split uniformly into four equal parts, and the mesh is refined to conformity by bisection through the longest element edge.

Finally, we implemented the refinement in parallel using the Message Passing Interface Library (MPI). Each processor refines its part of the domain independently. After achieving local conformity, the processors communicate to obtain global conformity. Realistic 2-D and 3-D examples of flows in porous media are given in the Bioscreen Problem.

2.4   Algorithms for Partitioning and Load Balancing

We have concentrated our efforts on implementation issues, working on two versions of our partitioning algorithms. The first implementation uses a simpler strategy for searching the graph based on the levels of a breadth-first search combined with recursive bisection. In addition, separators that give a better ratio of component to boundary sizes were were given higher priority than those that give a better balance. In addition, issues related to the optimal combination of smaller components into components of the final partition were addressed. The advantages of this approach are that the resulting algorithms are very fast and produce good results for triangulations with good quality characteristics and that the algorithms will work with almost no modification for 3-D meshes. The second implementation is of an algorithm that exploits the planar nature of the mesh and divides the graph directly into p parts, instead of recursively subdividing. This algorithm works better for triangulations of high aspect ratios (see paper 3 of the List of Publications Resulting from This Grant and [13, 16, 17]).

2.4.1   Constructing good separators under complex criteria

We studied graph separation as a basic tool for achieving even distribution of data among processors and to reduce the cost of communication between processors in a parallel computer. In graph separation algorithms, one has to construct a small set of vertices or edges called a *separator* whose deletion divides the input graph into small and roughly equal parts. If the input graph describes a finite element mesh, then the data associated with the elements corresponding to each component of the partition will be assigned to the same processor. The separator theorems give a guarantee that

the data partition is optimal according to certain criteria (see paper 7 of the List of Publications Resulting from This Grant and [14, 16, 17]).

According to the research plan, we studied separation in graphs where vertices or edges will have one or more types of weights and costs associated with them. We constructed an algorithm that finds in $O(n)$ time a separator of an arbitrary planar graph of n vertices. In case of vertex costs in the original graph the total sum of vertex costs of the separator is at most $\sqrt{8 \sum_{v \in V(G)} (cost(v))^2}$ and in case of edge costs the total sum of edge costs is at most $4\sqrt{d \sum_{v \in V(G)} (cost(v))^2}$, where d is the maximum vertex degree. This result is tight within a constant factor.

Furthermore, we studied applications of our result for solving problems in diverse areas. In graph embedding, we showed that any n-vertex planar graph of maximum degree d and non-negative edge costs whose squares sum up to n can be embedded into a binary tree, a hypercube, a butterfly graph, a shuffle-exchange graph, a cube-connected-cycles graph, or a de Brujn graph with $O(\log d)$ average weighted dilation. For the graph pebbling problem, which has applications in register allocation and memory management, we showed that any n-vertex planar acyclic directed graph G with non-negative vertex costs can be pebbled so that, the sum of the costs of all vertices with pebbles be $O(\sqrt{\sum_{v \in V(G)} (cost(v))^2} + I \log n)$. (Here I denotes the maximum sum of costs on the predecessors of any vertex of G.) We also found efficient solution of the tree congestion problem using our separation algorithms. No previous efficient algorithms were known for those problems. These results have been presented in paper 11 of the List of Publications Resulting from This Grant and in [2, 15, 16, 17]).

2.4.2 Maintaining partitions in a dynamic environment

Using graph partitioning algorithms, one can distribute the data among the processors so that the computational load is balanced and the communications are minimized. This initial partitioning, however, might later need to be updated after such operations on the mesh as mesh refinement (adding new points or edges) and mesh coarsening (removing existing points and edges). The currently existing graph partitioning algorithms construct the new partitioning from scratch, which is time consuming, and they also may result in a new data reallocation that differs much from the original one and thus might require many data transfers.

We develop a new algorithm for maintaining a balanced partitioning for dynamic planar graphs, i.e., such that can be changed by local vertex or edge addition or deletion operations. Our algorithm uses a tree-like data structure that takes only $O(n)$ space, where n is the size of the graph, and can be updated in $O(\log n)$ time after any local change of the graph of the above type. Furthermore, the updated partition is close to the original one, which means that only a few ($O(\log n)$ in the worst case) blocks of data have to be moved between processors in order to maintain the required properties of the partition (see paper 1 of the List of Publications Resulting from This Grant).

2.4.3 Using separators for adaptive mesh partitioning

The generation of a mesh with appropriate characteristics is a very important phase of the simulation, since the shapes of the finite elements have influence on the accuracy of the approximation and the numerical stability of the method. We investigated issues of mesh smoothing based on the so call *spring embedding* method, which has recently been studied in graph drawing applications. The idea of this method is to find a placement of the vertices of the mesh in such positions that

minimize the total sum of squares of the edge lengths. This simulates a mechanical system, where edges are replaced by strings which attract the corresponding endpoint of of the edge with force proportional to the length of the edge. There are also variants where repulsive forces are assigned to some of the edges. Our study showed that in many cases this method produces bad solutions, but there are cases where the resulting mesh be very bad or even the method can produce an invalid mesh. We investigated an improved version of the method, where real valued weights are assigned on the vertices and edges of the mesh. We showed that appropriate combination of weights can produce an optimal placement for any mesh.

2.5 Ready-made software

We have also used METIS in our codes. This is a software package for partitioning large irregular graphs/meshes and computing fill, thus reducing the sequencing of sparse matrices. This software has been developed by George Karypis and Vipin Kumar from the University of Minnesota. The algorithms in METIS are based on multilevel graph partitioning, where the graph is made progressively coarse, the coarse graph is partitioned, and then the computed partitions are projected into the fine graph. Below we give several 2-D and 3-D examples of sub-domain partitioning using METIS.

2.6 Implementation of the Domain Decomposition Structure on Parallel Computers

We have completed a domain decomposition implementation featuring the following:

- Sub-domain partitioning providing good load balance and minimization of the number of shared nodes between the sub-domains nodes (called interface nodes);
- *A posteriori* error estimators and indicators leading the refinement process and providing input for weighted sub-domain partitioning;
- Parallel multilevel local refinement preserving the mesh quality and providing (combined with the above two features) load balance and minimization of the interface node number over the different refinement levels;
- Relatively independent sub-domains "mapped" to different processors with data structures providing efficient access to the interface nodes; and
- Efficient solvers, visualization etc.

The implementation, using C++, is in an object-oriented style. The parallel computations are done using the Message Passing Interface Library (MPI). The main classes, their hierarchy, data members, and methods are given in Figure 1.

The PCG method that solves a symmetric and positive definite system of linear equations and the GMRES method that solves non-singular non-symmetric systems of linear equations are implemented as C++ templates. To perform the algorithm, the templates rely on four routines (to solve $Ax = b$):

- Given x compute Ax;
- Given x and y, compute the inner product (x, y);
- Given scalars a and b, and vectors x and y, compute $ax + by$; and
- Given x, compute Bx, where B is a preconditioner to A.

Subdomain

Data

> **Packet**
> Data - vertex indices; sharing subd. and corresponding packet number; owner subdomain, etc.
> Methods - send, receive; init. procedures

- Subdomain number; number of packets and array of packets

Methods

- Initialization procedures
- Parallel solve : CG, PCG (with multigrid preconditioner)
- Update -add contrib. from no-owners to owner, make no-owners 0.
- UpdateNoOwners - update no-owners with values from owners

Method

Data

> **Matrix**
> Data - pointer to the sparce structure and values; pointers to other useful data
> Methods - Action; ActionTranspose; Matrix mult; Gauss Seidel iteration

- Arrays of unknowns, RHS (put in Method for convinience)

Methods

- Functions initializing different stiffness matrices
- Solvers : CG, PCG, GMRES
- Preconditioners : multigrid; hierarchical basis multigrid
- A posteriori error indicators : RB, ZZ, Hierarchical, Second derivative
- Error computing functions : in discrete L^2, H^1, maximum and energy norms

Mesh

Data

> **tetrahedron**
> - 4 vertex indices
> - 4 face indices
> - 1 integer attribute

> **face**
> - 3 vertex indices
> - 2 tetrahedron indices

> **[edge]**
> - 2 vertex indices

> **vertex**
> - 3 coordinates (double)

- array of nodes on Dirichlet boundary
- sparse vertex connectivity structure (used for the stiffness matrix)
- [sparse vertex connectivity structure for hierarchical basis]
- [sparse vertex connectivity structure for interpolation]
- number of elements, faces, edges, vertices, levels

Methods

- Mesh constructor - initializes the mesh data from NETGEN output file
- Local Refine - input is list of tetrahedrons marked for refinement - the result is updated Mesh (a new conforming mesh level)
- Printing procedures (for Maple, MCGL, MTVPlot)
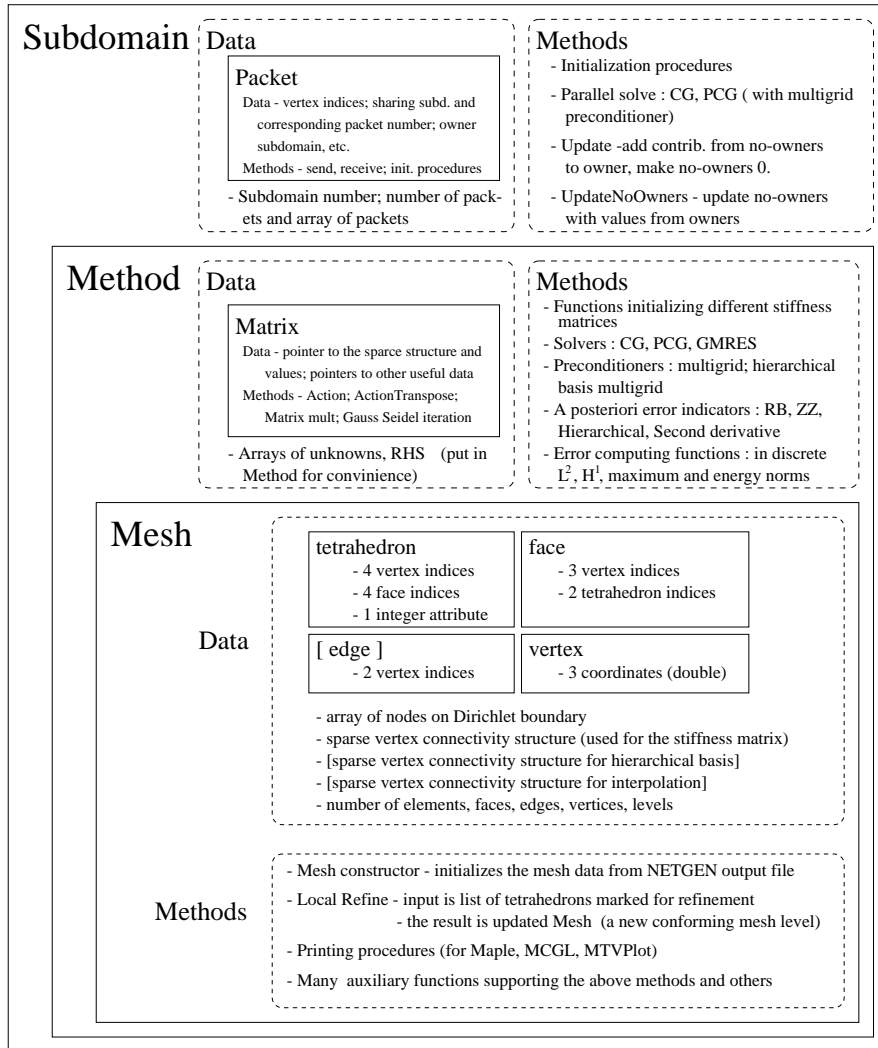- Many auxiliary functions supporting the above methods and others

Figure 1: Code structure. Main classes along with their data members and methods.

As a result of this template mechanism, we have highly reusable pieces of software.

We have also investigated a new and very promising technique for domain decomposition computations using non-matching grids. Namely, a multi-grid technique for uniformly preconditioning linear systems arising from a mortar finite element discretization of second-order elliptic boundary value problems has been introduced and analyzed by Gopalakrishnan and Pasciak (paper 16 of the List of Publications Resulting from This Grant). These problems are posed on domains partitioned into sub-domains, each of which is independently triangulated in a multilevel adaptive fashion. The multilevel mortar finite element spaces based on such triangulations (which need not align across sub-domain interfaces) are in general not nested. Suitable grid transfer operators and smoothers have been developed that lead to a variable V-cycle preconditioner resulting in a uniformly preconditioned algebraic systems.

We have also implemented and tested initialization procedures connecting our software to packages such as HYPRE and PETSc. HYPRE is being developed at the Center for Applied Scientific

Computing (CASC) at Lawrence Livermore National Laboratory. It is developed to fill a need for highly parallel preconditioners with scalable performance. PETSc is a toolkit providing variety of scalable (parallel) solvers. This direction of our work is in close cooperation with the LLNL team developing HYPRE and is now under intensive development.

3 SUMMARY OF FINDINGS & CONCLUSIONS: BIOSCREEN PROGLEM: A PARTICULAR EXAMPLE OF FLOW IN POROUS MEDIA

Here we present a particular problem of flow in porous media. It is based on data supplied to us by the Center for Biofilm Engineering at Montana State University. The setting is described below.

A steady state flow, with Darcy velocity $\underline{v}$ (measured in ft/yr), has been established in a parallelepiped shaped reservoir of size $1000 \times 1000 \times 500$. The problem setting (see below) gives us symmetry with respect to the plane $x_2 = 0$, so the equations are solved only in half of the domain, the parallelepiped $(0, 1000) \times (0, 500) \times (0, 500)$. The transport of a contaminant, in our case benzene, dissolved in the water is described by the convection-diffusion-reaction equation (4), where c is the benzene concentration, $\underline{b}$ is the Darcy velocity $\underline{v}$, K is the dispersion-diffusion tensor, and a is the biodegradation rate. We assume that the Darcy velocity $\underline{v}$ is obtained by solving the pressure equation (1) for fully saturated porous media under appropriate boundary conditions.

3.1 Simulation Results

Here we present simulation results for the case of a homogeneous reservoir, a non-homogeneous reservoir, and a non-homogeneous reservoir with a well.

3.1.1 Homogeneous aquifer

The pressures at the the faces $x_1 = 0$ and $x_1 = 1000$ are constants, 1000 and 0, respectively, and the permeability tensor is $D = 32I$, I is the identity matrix. On the rest of the boundary, a homogeneous Neumann condition is specified. This setting creates a uniform Darcy velocity $\underline{v} = (32, 0, 0)$ ft/yr. A steady-state leakage on boundary strip $x_1 = 0$, $x_3 = 50..350$ of 30 mg/l has been established. The dispersion/convection process causes the dissolved benzene to disperse in the reservoir. The dispersion tensor has the form $K = k_{diff}I + k_t\underline{v}^T\underline{v}/|\underline{v}| + k_l(|\underline{v}|^2I - \underline{v}^T\underline{v})/|\underline{v}|$, where $k_{diff} = 0.0001$, $k_t = 21$ and $k_l = 2.1$. The biodegradation is transforming the pollutant into a solid substance which is absorbed by the soil. This leads to a decrease in the benzene. Its concentration level curves are shown on Figure 2 for the case of low biodegradation rate $a = 0.1$ mg/yr and on Figure 3 for medium biodegradation rate $a = 0.2$ mg/yr. We have started with an initial coarse mesh with 52 nodes.

3.1.2 Non-homogeneous aquifer

Here the problem setting is as above, but a layer is added (see Figures 2, 3, and 4). In the layer strip we take the permeability D_{layer} to be twice smaller than in the rest of the domain, i.e. $D_{layer} = 16I$. Now the Darcy velocity is not constant and the error estimators force the grid to be refined around the layer. The obtained grid is shown on Figure 2.

After the pressure is found with prescribed accuracy we solve the corresponding problem for the concentration. Figure 3 shows the obtained mesh and the isolines for the concentration in
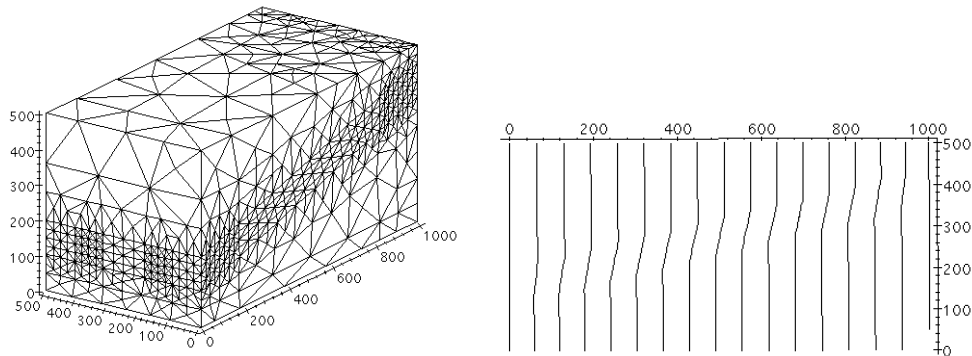
Figure 2: Pressure computations for a non-homogeneous reservoir; (left) the locally refined 3-D mesh on level 2 with 4,053 nodes; (right) Contour curves of the pressure for level 2
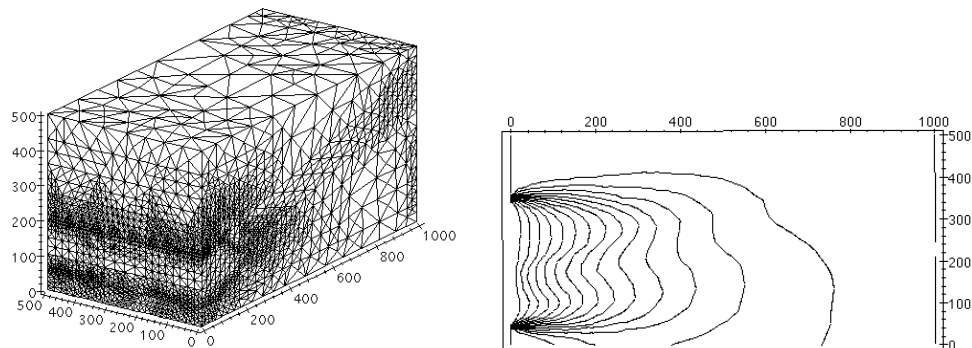
.

Figure 3: Concentration computations for a non-homogeneous reservoir; (left) the 3-D mesh on refinement level 4 with 39,445 nodes; (right) contour curves of the concentration for the cross-section $x_2 = 0$; the permeability in the layer is two time smaller than the rest of the domain

.

the reservoir cross-section $x_2 = 0$ on grid refinement level 4. Two more experiments varying the permeability tensor are shown on Figure 4. The first one shows the concentration isoline in the reservoir cross-section $x_2 = 0$ when the permeability in the layer is 5-times and 10-times smaller than the permeability in the rest of the reservoir. The initial coarse mesh in both cases has 235 nodes. The solution method is borroed from [9] and utilizes multigrid concepts. The highest level of refinements has 67,509 nodes.

3.1.3 Non-homogeneous aquifer with a well

Finally, we consider a problem with one well using the well model described in Section 2 and the approximation given in Section 3. The well has an axis along the segment $x_1 = 250$, $x_2 = 0$, $x_3 = 0..400$, and its production rate is $Q = 200,000\ l/yr$. On Figure 5, we show the adapted mesh and the level curves for the pressure in the reservoir cross-section $x_2 = 0$. On Figure 6 we show the computational mesh and the level curves obtained for the concentration in the reservoir cross-section $x_2 = 0$ on grid refinement level 5.
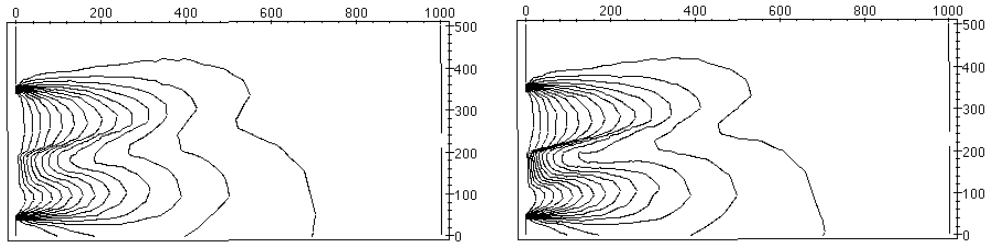
Figure 4: Concentration computations for a non-homogeneous reservoir; contour curves of the concentration for permeabilities in the layer 5 times (left) and 10 times (right) smaller than the permeability in the rest of the domain
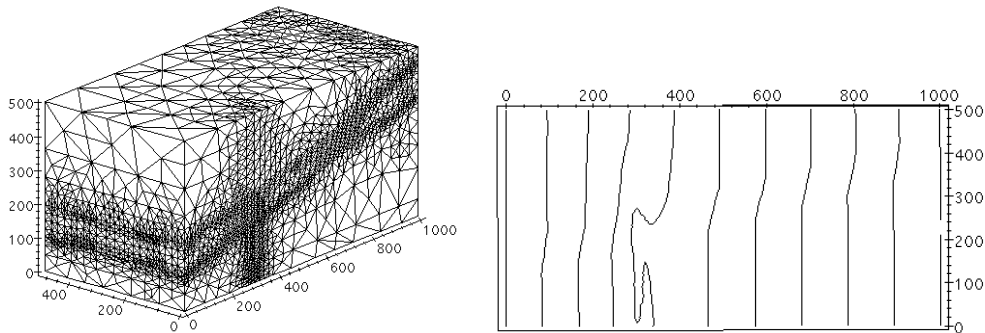

Figure 5: Pressure computations for a non-homogeneous reservoir with a well; (left) the 3-D Mesh on level 3 with 34,236 nodes; (right) contour curves of the pressure on level 3

3.2 Conclusions

The developed methodology allowed us to simulate fairly large problems related to groundwater reservoirs (flow, diffusion and dispersion of passive chemicals) in a parallel computer environment. The resulting problems could easily reach more than one million unknonws which, when combined with a grid refinement technique, ensure high resolution and qulity of the numerical results.

4 PUBLICATIONS RESULTING FROM THIS GRANT

1. L.G. Aleksandrov and H.N. Djidjev, Maintaining partitions of dynamic planar meshes, *Proceedings of the Canadian Conference on Computational Geometry (CCCG'98)*, pp. 84-85.
2. H. Chen, R. Ewing, and R. Lazarov, Superconvergence of mixed finite element methods for parabolic problems with non-smooth initial data, *Numerische Mathematik*, **78** (1998), 495-521.
3. T. Chernogorova, R.E. Ewing, O.P. Iliev, and R.D. Lazarov, On the discretization of interface problems with perfect and imperfect contact, *Lecture Notes in Physics*, **552** (2000), 93–103.
4. H. Djidjev, Computing the girth of a planar graph, International Colloquium on Automata, Languages and Programming (ICALP'00), *Lecture Notes in Computer Science*, **1853**, 2000, pp. 821–831.
5. H. Djidjev, A Dynamic Algorithm for Maintaining Graph Partitions, Scandinavian Workshop on Algorithm Theory (SWAT'00), *Lecture Notes in Computer Science*, **1851**, 2000, pp. 71–82.
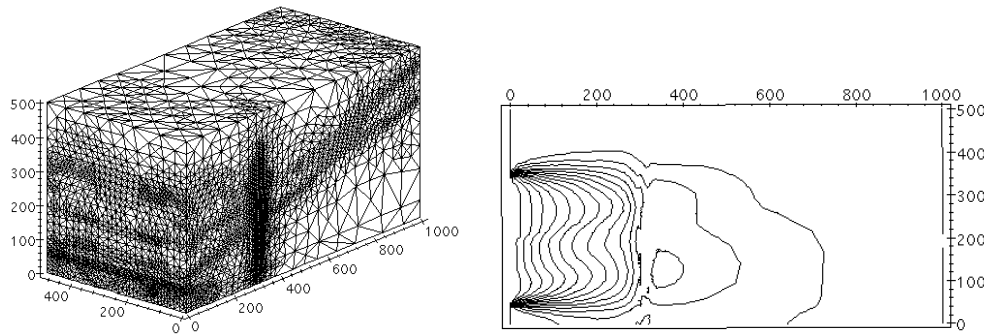
11

Figure 6: Concentration computations for a non-homogeneous reservoir with a well; (left) the 3-D mesh with 67,509 nodes in half of the domain obtained after 5 levels of refinement; (right) contour curves of the concentration in the plane $x_2 = 0$

6. H. Djidjev, Force-directed methods for smoothing unstructured triangular and tetrahedral meshes, *Proceedings of the 9th International Meshing Roundtable*, October 2-5, 2000, New Orleans, pp. 395-406.

7. H. Djidjev, Weighted graph separators and their application, European Symposium on Algorithms (ESA'97), *Lecture Notes in Computer Science*, **1284** (1997), 130-143.

8. R.E. Ewing, R.D. Lazarov, and Y. Lin, Finite volume element approximations of integro-differential parabolic problems, *Recent Advances in Numerical Methods and Applications*, (O. Iliev, M. Kaschiev, S. Margenov, Bl. Sendov, P. Vassilevski, Eds.), World Scientific, 1999, 3–15.

9. R.E. Ewing, R.D. Lazarov, S. Lyons, D. Papavassiliou, J.E. Pasciak, and G. Qin, Numerical well model for non-Darcy flow through isotropic porous media, *Computational Geosciences*, **3** (1999), 185-204.

10. R.E. Ewing, R. D. Lazarov, T. Lin, and Y. Lin, Domain decomposition capabilities for the mortar finite volume element methods *Technical Report, ISC-00-04-MATH*, Texas A&M University, 11th International Conference on Domain Decomposition Methods in Science and Engineering, London, 1998, C.H. Lai, P.E. Bjorstad, M. Cross, and O. Widlund, Eds. DD Press, Bergen, 1999, 220-227.

11. R. Ewing, R. Lazarov, and Y. Lin, Finite volume element approximations of nonlocal in time one-dimensional flows in porous media, *Computing*, **64** (2000), 157-182.

12. R.E. Ewing, R.D. Lazarov, and Y. Lin, Finite volume element approximations of nonlocal reactive flows in porous media, *Numerical Methods for PDEs*, **16** (2000), 285-311.

13. R.E. Ewing, R.D. Lazarov, T. Lin, and Y. Lin, Mortar finite volume element approximations of second-order elliptic problems, Technical Report ISC-98-08-MATH, *East-West J. Numer. Math.*, **8** (2000), 93-110.

14. R.E. Ewing, O.P. Iliev, and R.D. Lazarov, A modified finite volume approximation of second-order wlliptic equations with discontinuous coefficients, *Technical Report, ISC-99-01-MATH*, Texas A&M University, 1999, (submitted to *SIAM J. Sci. Comput.*).

15. R.E. Ewing, Up-scaling of biological processes and multi-phase flow in highly heterogeneous media, *IMA Volumes in Mathematics and its Applications*, J. Chadam, A. Cunningham, R.E. Ewing, Ortoleva, and M.F. Wheeler, eds., Springer-Verlag, to appear.

16. J. Gopalakrishnan and J. Pasciak, Multi-grid for the mortar finite element method, *SIAM J.*

Numer. Anal., **37** (2000), 1029-1052.

17. C.S. Kim, R.D. Lazarov, J.E. Pasciak, and P.S. Vassilevski, Multiplier spaces for the mortar finite element method in three dimensions, *Technical Report ISC-00-07-MATH*, (submitted to *SIAM J. Numer. Anal.*)

18. R.D. Lazarov, J.E. Pasciak, and S.Z. Tomov, Error control, local grid refinement, and efficient solution algorithms for singularly perturbed problems, *Technical Report ISC-99-04-MATH*, 1999 Analytical and Numerical Methods for Singularly Perturbed Problems, L.G. Vulkov, J.J.H. Miller, and G.I. Shishkin, Eds, Nuova Science Publishing House, 2000, 309 pages.

19. R.D. Lazarov, J.E. Pasciak, and P.S. Vassilevski, Mixed finite element methods for elliptic problems on non–matching grids, *Technical Report ISC-99-10-MATH*, *Large-Scale Scientific Computations of Engineering and Environmental Problems*, *Notes on Numerical Fluid Mechanics*, **73** (2000), 25–35

20. R.D. Lazarov, J.E. Pasciak, and P.S. Vassilevski, Iterative solution of a coupled mixed and standard Galerkin discretization method for elliptic problems, (to appear in *Numerical Linear Algebra and Applications*).

21. R.D. Lazarov, S.Z. Tomov, Adaptive finite volume element method for convection-diffusion-reaction problems in 3-D, *Technical Report, ISC-00-02-MATH*, (to appear in Proc. Int. Symposium on Large Scale Scientific Computing, Kananaskis, Canada, June 2000)

22. R.D. Lazarov, S.Z. Tomov, and P.S. Vassilevski, Interior penalty discontinuous approximations of elliptic problems, *Technical Report ISC-00-04-MATH*, (submitted to *SIAM J. Sci. Comp.*)

5   STUDENT PARTICIPATION AND SUPPORT

Four Ph.D. students participated and were supported by this grant. J. Gopalakrishnan (defened Ph.D. in 1999) developed efficient multi-grid/multilevel methods for non-conforming domain decomposition and mortar methods. C. Kim (Ph.D. expected in 2001) developed mortar spaces based on a dual approach. S. Tomov (Ph.D. expected in 2002) was the main constructor of the data structure for parallel computations and implemented both finite element and finite volume methods with local grid refinement. C. Bacuta (defended Ph.D. in 2000) participated in the analysis of various multilevel methods and refinement strategies.

References

[1] M. AINSWORTH AND J. T. ODEN, *A unified approach to a posteriori error estimators based on element residual methods*, Numer. Math., 65 (1993), pp. 23-50.

[2] L. ALEKSANDROV, M. LANTHIER, A. MAHESHWARI, AND J.R. SACK, *An ϵ-approximation algorithm for weighted shortest paths on polyhedral surfaces*, Proceedings of the 6th Scandinavian Workshop on Algorithm Theory, SWAT'98, Stockholm, LNCS 1432, (1998), pp. 11-22.

[3] D. N. ARNOLD, A. MUKHERJEE, AND L. POULY, *Locally adapted tetrahedral meshes using bisection*, SIAM J. on Sci. Computing, **22 (2)** (2000), pp. 387–405.

[4] I. BABUSKA AND W. C. RHEINBOLDT, *Error estimates for adaptive finite element computations*, SIAM J. Numer. Anal., 15 (1978), pp. 736–754.

[5] I. BABUSKA AND W. C. RHEINBOLDT, *A-posteriori error estimates for the finite element method*, Int. J. Numer. Meth. Engng., 12 (1978), pp. 1597–1615.

[6] R. E. Bank, *A simple analysis of some a posteriori error estimates*, Appl. Numer. Math., 26 (1998), pp. 153–164.

[7] R.E. Bank and A. Weiser, *Some a posteriori error estimators for elliptic partial differential problems*, Math. Comp., 44 (1985), pp. 283–301.

[8] R. Becker and R. Rannacher, *Weighted a posteriori error control in finite element methods*, Preprint 96-1 (SFB 359), Proc. ENUMATH - 95, Paris, 1995.

[9] J.H. Bramble and X. Zhang, *The Analysis of Multi-grid Methods*, North Holland, N.Y., 2000 (to appear).

[10] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer, 1996.

[11] J.H. Cushman, X. Hu, and F. Deng, *Nonlocal reactive transport with physical and chemical heterogeneity: Localization error*, Water Resources Research, 31 (1995), pp. 2219–2237.

[12] G. Dagan, Flow and Transport in Porous Formations, 1989.

[13] H.N. Djidjev and J.R. Gilbert, *Separators in graphs with negative or multiple vertex weights*, Algorithmica, 23 (1999), pp. 57-71.

[14] H.N. Djidjev, *Force-directed methods for mesh improvement*, Proceedings of the 2nd Workshop on Algorithm Engineering and Experiments (ALENEX 00), (2000), pp. 29-42.

[15] H.N. Djidjev, *Partitioning planar graphs with vertex cosos: Algorithms and applications*, Algorithmica, 28 (2000), pp. 51-75.

[16] R. Fagin, J.Y. Halpern, Y. Moses, M.Y. Vardi, *Knowledge-based programs*, Distributed Computing, **10** (1997), pp. 199-225.

[17] J. Feigenbaum, S. Kannan, M.Y. Vardi, M. Viswanathan, *Complexity of problems on graphs represented by OBDDs*, Proc. 15th Symp. on Theoretical Aspects of Computer Science, Springer-Verlag, Lecture Notes in Computer Science **1373**, (1998), 216-226.

[18] L. P. Franca, S. L. Frey, and T. J. R. Hughes, *Stabilized finite element methods. I: Application to advective-diffusive problems*, Comput. Meth. Appl. Mech. Engrg., 95 (1992), pp. 253–271.

[19] V.V. Shelukhin, *A non-local in time model for radionuclides propagation in Stokes fluids*, in Dynamics of Fluids with Free Boundaries, Siberian Branch of the Russian Academy of Sciences, Institute of Hydrodynamics, **107** (1993), pp. 180–193.

[20] M. Slodicka and R. Van Keer, *A nonlinear elliptic equation with non-local boundary condition solved by linearization*, Preprint 1, (2000), Department of Mathematics, University of Gent, Belgium.

[21] S. Z. Tomov, *Domain decomposition structure for massively parallel computations with application to fluid flow modeling in 3-D*, Texas A&M University, ISC Technical report (2000) (in progress).

[22] J. Xu and L. Zikatanov, *A monotone finite element scheme for convection-diffusion equations*, Math. Comp., 68 (1999), pp. 1429–1447.

[23] O. C. Zienkiewicz and J. Z. Zhu, *A Simple Error Estimator and Adaptive Procedure for Practical Engineering Analysis*, Int. J. Numer. Meth. Engng., 24 (1987), pp. 337-357.