ADAPTIVE FINITE VOLUME ELEMENT METHOD FOR CONVECTION-DIFFUSION-REACTION PROBLEMS IN 3-D*

R.D. LAZAROV† AND S.Z. TOMOV‡

Abstract. We present an adaptive numerical technique for solving steady-state diffusion and convection-diffusion-reaction equations in 3-D 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.

Key words. finite elements, finite volumes, a posteriori error estimates, local grid refinement, convection-diffusion-reaction problems

AMS subject classifications. 65N15, 65C15, 65Y15

1. Introduction. We aim to develop, implement, and test a computational technique for simulation of fluid flow and transport of passive chemicals in porous media. We consider the pressure equation describing fully saturated single phase flow in 3-dimensional bounded aquifers with variable permeability. Further, we consired the case when chemicals are transported by the flow and absorbed by the media. In this paper we discuss the steady-state solutions of these two problems. The corresponding mathematical models for both problems, flow and transport, are elliptic equations of second order (diffusion and convection-diffusion-reaction equations) subject to various boundary conditions.

The solutions of these problems exhibit local behavior due to discontinuity in the boundary data and the coefficients of the differential equations, from extraction/injection wells, and/or other local phenomena. Here we describe a computational technique that utilizes both finite volume and finite element approximations of the differential equations and a posteriori error estimators and indicators that will lead to adaptive local grid refinement. This technique is implemented (with appropriate tools for grid generation, partitioning and parallelization) in a multilevel fashion and tested on various boundary value problems for diffusion, convection-diffusion, and reaction-diffusion equations that exhibit local or singular behavior. We also present a number of numerical simulations of flows in inhomogeneous aquifers and transport dispersion and absorbtion of benzene that has been dissolved in the water.

The paper is organized in the following manner. In Section 2 we formulate various boundary value problems for the diffusion equation for the pressure and the advection-diffusion equation for the concentration of the chemicals. Further, we introduce some notations and give the weak formulations of the problems. In Section 3 we present the finite element and the finite volume element methods. Further, in Section 4 we discuss residual type of error estimators for the finite volume element method that will lead to adaptive local grid refinement. In Section 5 we describe the object-oriented structures for the computer implementation of our strategy. Finally, in Section 6 we demonstrate the performance of the developed adaptive grid refinement method on various model second-order problems.

*This work has been partially supported by the US Environmental Protection Agency under Grant R 825207 and by the National Science Foundation under Grant DMS-9973328

†Institute for Scientific Computation, Texas A& M University, College Station, Texas 77843 (lazarov@math.tamu.edu).

‡Department of Mathematics, Texas A& M University, College Station, Texas 77843 (tomov@math.tamu.edu).

2. Problem formulation. The mathematical model of steady state groundwater flows and transport in porous media yields two basic equations. These are the *Darcy* equation for the pressure, discussed in Subsection 2.1, and the advection-dispersion (transport) equation, discussed in Subsection 2.2. 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. Diffusion (pressure) equation. The fluid flow is due to the velocity $\underline{v}$ defined by the *Darcy's law*: $\underline{v} = -D\nabla p$, where p is the pressure, D is the permeability of the porous media. The pressure p satisfies the following equation subject to various boundary conditions:

$$\begin{cases} \nabla \cdot \underline{v} \equiv -\nabla \cdot D\nabla p &= f, \qquad \text{in } \Omega, \\ p &= p_D, \qquad \text{on } \Gamma_D, \\ -D\nabla p \cdot \underline{n} - \gamma p &= p_N, \qquad \text{on } \Gamma_N, \\ p &= p_w, \qquad \text{on } \Gamma_w. \end{cases} \tag{2.1}$$

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, correspondingly. 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 [14].

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

$$p = p_w, \text{ on } \Gamma_w, \ p_w \text{ unknown constant and } \int_{\Gamma_w} D\nabla p \cdot \underline{n} ds = Q. \tag{2.2}$$

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

$$-D\nabla p \cdot \underline{n} = \gamma p + \tilde{Q} \quad \text{ on } \ \Gamma_w \tag{2.3}$$

with $\tilde{Q}$ a given constant. Note that here $\tilde{Q}$ is the pointwise flux, while Q in (2.2) 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 model were done for condition (2.3) with $\gamma = 0$.

2.2. 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:

$$\begin{cases} -\nabla \cdot K\nabla c + \nabla \cdot (\underline{b}c) + ac &= f, \quad \text{in } \Omega, \\ c &= c_D, \quad \text{on } \Gamma_D, \\ (-K\nabla c + \underline{b}c) \cdot \underline{n} &= c_N, \quad \text{on } \Gamma_N^{in}, \\ -K\nabla c \cdot \underline{n} &= 0, \quad \text{on } \Gamma_N^{out}, \\ c &= c_w, \quad \text{on } \Gamma_w. \end{cases} \tag{2.4}$$

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 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 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 (2.1). Then the diffusion-dispersion tensor is given by $K = k_{diff} I + k_t \underline{v}^T \underline{v}/|\underline{v}| + k_l(|\underline{v}|^2 I - \underline{v}^T \underline{v})/|\underline{v}|$, where k_{diff}, k_t, and k_l are constants characterizing correspondingly the diffusion, transverse dispersion, and longitudal dispersion.

In the case of 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 (2.1) with well boundary condition (2.3) we get the following boundary condition for the concentration:

$$K\nabla c \cdot \underline{n} = 0 \quad \text{on } \Gamma_w. \tag{2.5}$$

2.3. Notations. We denote the solution to both problems formulated above by u, i.e. $u = p$ or $u = c$. For simplicity we consider only homogeneous Dirichlet boundary conditions on Γ_D and assume that Γ_D is nonempty. We further introduce the space $H_D^1(\Omega) = \{v \in H^1(\Omega) : \ v|_{\Gamma_D} = 0\}$. In this space we shall use the standard L^2 and H^1-norms: $||u|| = (u,u)^{1/2}$, $||u||_{1,\Omega} \equiv ||u||_1 = \{(u,u) + (\nabla u, \nabla u)\}^{1/2}$, where $(\cdot,\cdot)$ is the inner product in L^2 and ∇u is the gradient of u.

In order to simplify our notation we shall present the weak formulation for the problem (2.4) with boundary condition (2.5) on Γ_w. Next, we introduce the bilinear from $a(\cdot,\cdot)$ defined on $H_D^1(\Omega) \times H_D^1(\Omega)$:

$$a(u,v) \equiv (K\nabla u - \underline{b}u, \nabla v) + (au, v) + \int_{\Gamma_N^{out}} \underline{b} \cdot \underline{n} \ u \ v ds + \int_{\Gamma_w} \tilde{Q} \ u \ v \ ds.$$

Here, $\tilde{Q}$ is the constant in the boundary condition (2.3).

Further, we assume that the coefficients of problem (2.4) with boundary conditions (2.5) ensure the following conditions: (a) the form is coercive in $H^1_D(\Omega)$, i.e. there is a constant $c_0 > 0$ s.t. $a(u,u) \geq c_0||u||^2_1, \ \ \forall u \in H^1_D(\Omega)$; and (b) the form is bounded in $H^1_D(\Omega)$, i.e. there is a constant $c_1 > 0$ s.t. $a(u,v) \leq c_1||u||_1||v||_1, \ \ \forall u, \ v \in H^1_D(\Omega)$. A sufficient condition for the coercivity is $a(x) + 0.5\nabla \cdot \underline{b} \geq 0$ for all $x \in \Omega$.

Then we rewrite (2.4) in the following weak form for the case of homogeneous Dirichlet boundary conditions on Γ_D and boundary condition (2.5) in the presence of injection/production wells: Find $u \in H^1_D(\Omega)$ such that

$$a(u,v) = (F,v) \equiv (f,v) - \int_{\Gamma_N^{in}} c_N v ds \ \ \text{for all} \ \ v \in H^1_D(\Omega). \tag{2.6}$$

3. Approximation method. Here we use finite element and finite volume element as approximation methods. Both methods use partition of the domain Ω into tetrahedra, called finite elements and denoted by T. The partition is denoted by $\mathcal{T}_h$. The space V_h is a finite dimensional subspace of $H^1_D(\Omega)$ of continuous piece-wise linear functions over the partition $\mathcal{T}_h$. The finite element solution u_h is computed by the Galerkin method. Namely, we introduce an approximation $a_h(\cdot,\cdot)$ of the form $a(\cdot,\cdot)$ in V_h and find $u_h \in V_h$ such that

$$a_h(u_h,v) = F(v) \equiv (f,v) - \int_{\Gamma_N^{in}} c_N \ v ds \ \text{for all} \ v \in V_h.$$

In the case when there is no convection or small convection $a_h(\cdot,\cdot)$ is defined as a straightforward evaluation of $a(\cdot,\cdot)$ over V_h. For convection-dominated problems this approximation gives oscillating numerical results which we would like to avoid. For such problems we are interested in approximation methods that produce solutions satisfying the maximum principle (for diagonal K) and are locally conservative. Such schemes are also known as monotone schemes. A well-known sufficient condition for a scheme to be monotone is that the corresponding stiffness matrix is an M–matrix. Good choices of monotone schemes are: Tabata's upwind scheme [15], the stream-line upwind Galerkin method (SUPG scheme) of Franca, Frey, and Hughes [10] and the scheme of Xu and Zikatanov [19], which constructs a finite element discretization by an appropriate averaging of the differential equation coefficients on the element edges. For construction, analysis, and use of methods for convection-diffusion problems we refer to the monograph of H.-O. Ross, M. Stynes, and L. Tobiska [13].

For deriving the finite volume approximation we shall need the so-called dual partition of Ω into finite volumes. This partition is described below. First, we introduce the set $N_h = \{p : p \ \ \text{is a vertex of element} \ T \in \mathcal{T}_h\}$ and the set $N_h^0 \subset N_h$ of the vertices except those on Γ_D. For a given vertex x_i we denote by $\Pi(i)$ the index set of all neighbors of x_i in N_h, i.e. all vertices that are connected to x_i by an edge.

For a given finite element partition $\mathcal{T}_h$ we construct a dual mesh $\mathcal{T}_h^*$ (based upon $\mathcal{T}_h$), whose elements are called control volumes. In the finite volume methods there are various ways to introduce the control volumes. Almost all approaches can be described in the following general scheme. In each tetrahedron $T \in \mathcal{T}_h$ a point q is selected. On each of the four faces $\overline{x_i x_j x_k}$ of T a point x_{ijk} is selected and on each of the six edges $\overline{x_i x_j}$ a point x_{ij} is selected. Then q is connected to the points x_{ijk}, and in the corresponding faces the points x_{ijk} are connected to the points x_{ij} by straight lines (see Figure 3.1). Control volumes are associated to each vertex $x_i \in N_h$. Control volume associated with vertex x_i is denoted by V_i and defined as the union of the

"quarter" elements $T \in \mathcal{T}_h$, which have x_i as a vertex (see Figure 3.1). The interface between two control volumes, V_i and V_j, is denoted by γ_{ij}.

In our implementation q is the medicenter of the tetrahedron, x_{ijk} is the medicenter of the face defined by the vertices x_i, x_j, and x_k, and x_{ij} is the midpoint of the edge connecting the vertices x_i and x_j (as on Figure 3.1).

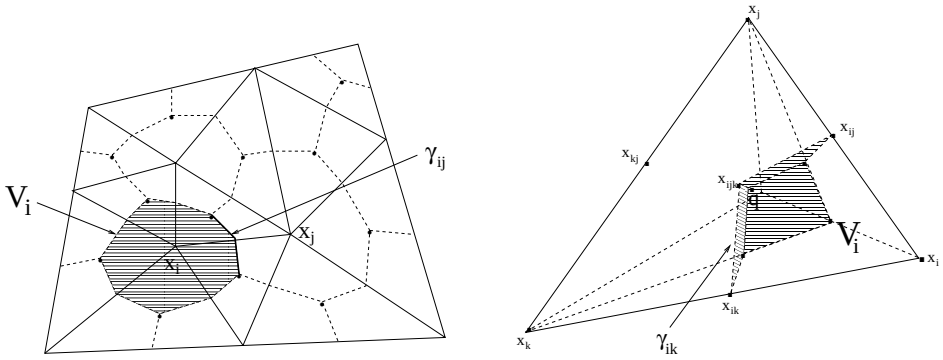

FIG. 3.1. *Left: Finite element and finite volume partitions in 2-D; Right: Contribution from one element to control volume V_i in 3-D; Point q is the element's medicenter and internal points for the faces are the medicenters of the faces.*

Now $V_h = \text{span}\{\phi_i(x) : x_i \in N_h^0\}$ is the finite element space and $V_h^* = \text{span}\{\chi_i(x) : x_i \in N_h^0\}$ is its dual finite volume space. Here ϕ_i is the standard continuous hat function associated with the node x_i and χ_i is the characteristic function of the volume V_i.

The discrete finite volume element approximation u_h of (2.4) is the solution to the problem: Find $u_h \in V_h$ such that

$$a_h(u_h, v^*) \equiv A(u_h, v^*) + C(u_h, v^*) = F(v^*), \text{ for all } v^* \in V_h^*. \tag{3.1}$$

The bilinear form $A(u_h, v^*)$ and the linear from $F(v^*)$ are defined by

$$A(u_h, v^*) = \sum_{x_i \in N_h^0} v_i^* \left\{ - \int_{\partial V_i \setminus (\Gamma_N \cup \Gamma_w)} K \nabla u_h \cdot \underline{n} ds + \int_{V_i} a u_h dx + \int_{\Gamma_w \cap V_i} \tilde{Q} u_h ds \right\}, \tag{3.2}$$

$$F(v^*) = \sum_{x_i \in N_h^0} v_i^* \left\{ \int_{V_i} f dx - \int_{\partial V_i \cap \Gamma_N^{in}} c_N ds \right\}, \tag{3.3}$$

for $u_h \in V_h$ and $v^* \in V_h^*$. Here and further we use the notation $v_i^* = v^*(x_i)$. We use two different approximations for computing $C(u_h, v^*)$. The first one is a straightforward evaluation of $C(u_h, v^*)$:

$$C(u_h, v^*) = \sum_{x_i \in N_h^0} v_i^* \int_{\partial V_i \setminus \Gamma_N^{in}} \underline{b} \cdot \underline{n} u_h ds, \quad u_h \in V_h, \quad v^* \in V_h^*. \tag{3.4}$$

However, this approximation is not monotone for large convection and up-winding or other stabilization is required. In such cases we define the convection form in a

different way. We split the integral over ∂V_i on integrals over $\gamma_{ij} = \partial V_i \cap \partial V_j$, (see Figure 3.1) and introduce outflow and inflow parts of the boundary of the volume V_i. This splitting can be characterized by the quantities $(\underline{b} \cdot \underline{n}_i)_+ = \max(0,\ \underline{b} \cdot \underline{n}_i)$ and $(\underline{b} \cdot \underline{n}_i)_- = \min(0,\ \underline{b} \cdot \underline{n}_i)$, where $\underline{n}_i$ is the outer unit vector normal to ∂V_i. Then the convection form $C(u_h, v^*)$ is defined as

$$C(u_h, v^*) = \sum_{x_i \in N_h^0} v_i^* \sum_{j \in \Pi(i)} \int\limits_{\gamma_{ij}} \left[(\underline{b} \cdot \underline{n}_i)_+ u_h(x_i) + (\underline{b} \cdot \underline{n}_i)_- u_h(x_j) \right] \, ds. \tag{3.5}$$

This is an upwind approximation of the convection term and is closely related to the discontinuous Galerkin approximation.

4. Local error estimators. The behavior of the physical process is greatly affected by local smoothness properties of the coefficients, the source term, and the boundary data as well as the singularities due to corners, boundary layers, wells or nonlinear behavior. For such cases it is essential that the numerical method has capabilities to resolve the local behavior of the solution. In the context of the finite element method there are two main techniques for the error reduction. The first approach, the so called h–refinement, uses polynomials of the same degree, but adaptively refines the grid by decreasing the mesh size h (see, e.g. [3], [4], [9]). The second approach, the so called p–refinement, increases the order of the algebraic polynomials used in the approximation process (see, e.g. [1], [18]).

For the finite element method we have implemented and tested (for both 2-D and 3-D problems) three error indicators based on the h-version, namely: (1) residual based refinement (see, e.g. [1], [3], [4], [7], [18]); (2) Zienkiewicz-Zhu technique (see, e.g. [20]), and (3) hierarchical refinement (see, e.g. [6]).

For the finite volume element method we have developed analogues of the residual and Zienkiewicz-Zhu techniques. We explain the residual method first in the case of small convection and then we give the modifications needed for the case of dominant convection. The method expresses the error in terms of the residual of the approximate solution. This residual is a sum of the residuals of the differential equation evaluated for the approximate solution over each element and the jumps of the conormal defivative along the element faces. The main idea is illustrated on the model problem (2.4).

We first demonstrate the method in the case when no upwinding is used for the approximation of the convection term and on the entire boundary Γ we have homogeneous Dirichlet boundary conditions, namely we consider the problem: Find $u_h \in V_h$ such that

$$a(u_h, v^*) \equiv \sum_{x_i \in N_h^0} v_i^* \left\{ \int\limits_{\partial V_i} (-K \nabla u_h + \underline{b} u_h) \cdot \underline{n} ds + \int\limits_{V_i} a u_h dx dx \right\} = \sum_{x_i \in N_h^0} v_i^* \int\limits_{V_i} f dx,$$

for all $v^* \in V_h^*$. We give a posteriori estimate for the error $e = u - u_h$, where u is the solution of the *weak* problem (2.6). Using the divergence theorem over the volumes and regrouping the sum over the volumes as sum over the tetrahedra give us the following:

$$a(e, v^*) = \sum_{x_i \in N_h^0} v_i^* \left\{ \int\limits_{V_i} \left(f + \nabla \cdot (K \nabla u_h - \underline{b} u_h) - a u_h \right) dx - \sum_{j \in \Pi(i)} \int\limits_{\gamma_{ij}} [K \nabla u_h] \cdot \underline{n} ds \right\}$$

$$= \sum_{T\in\mathcal{T}_h} \left\{ \int_T (f + \nabla\cdot K\nabla u_h - \nabla\cdot(\underline{b}u_h) - au_h)v^* dx - \frac{1}{2}\int_{\partial T} [K\nabla u_h]\cdot \underline{n}v^* ds \right\}$$
$$\equiv \sum_{T\in\mathcal{T}_h} \{(R_T, v^*)_T + (R_{\partial T}, v^*)_{\partial T}\} \quad for\ all\ \ v^* \in V_h^*.$$

Here $[K\nabla c_h]$ denotes the jump of $K\nabla c_h$ across the finite element boundary. The last equality defines R_T as the residual over the element T and $R_{\partial T}$ is the jump across the element boundary. Using the weak formulation given by (2.6), integrating by parts, we get similar expression for e as well:

$$a(e,v) = \sum_{T\in\mathcal{T}_h} \{(R_T, v)_T + (R_{\partial T}, v)_{\partial T}\} \quad for\ all\ v \in H_0^1(\Omega).$$

In what follows the second argument of the bilinear form $a(\cdot,\cdot)$ will determine whether it is the bilinear form for finite volumes, $a(\cdot, v^*)$, or the bilinear form for finite elements, $a(\cdot, v)$. Using the Petrov-Galerkin orthogonality for the finite volume method $a(e, v^*) = 0$, for all $v^* \in V_h^*$, and applying Hölder's inequality on each element leads to the following estimate for the error in the energy norm:

$$\begin{aligned} c_0||e||_1^2 &\leq a(e,e) = a(e,e) - a(e,v^*) \\ &= \sum_{T\in\mathcal{T}_h} \{(R_T, e-v^*)_T + (R_{\partial T}, e-v^*)_{\partial T}\} \leq \sum_{T\in\mathcal{T}_h} \rho_T\omega_T \end{aligned} \tag{4.1}$$

Here the local residuals ρ_T and the weights ω_T are defined by

$$\rho_T := h_T||R_T||_T + h_T^{1/2}||R_{\partial T}||_{\partial T},$$
$$\omega_T := max\left\{h_T^{-1}||e-v^*||_T, h_T^{-1/2}||e-v^*||_{\partial T}\right\}.$$

The local approximation properties of the finite volume elements ensure that there is a $v^* \in V_h^*$ such that $\omega_T \leq C_{I,T}||\nabla e||_T$, where $\max C_{I,T} = C_I \approx 1$ is an interpolation constant (see [8]). Using this fact and Schwartz inequality we finally get

$$||e||_1 < CC_I \left(\sum_{T\in\mathcal{T}_h} \rho_T^2\right)^{1/2},$$

where $C = 1/c_0$ with c_0 being the coercivity constant for the bilinear form $a(\cdot,\cdot)$. An estimate for the error in L^2-norm can be obtained through duality argument.

The mesh generation aims to equilibrate the local residuals ρ_K, i.e. for a given tolerance δ, the elements $T \in T_h$ are refined according to the criteria $\rho_T \approx \delta/(CC_I\sqrt{N})$, where N is the number of tetrahedrons in T_h.

For the case of convection dominated problems we use the approximation (3.1), where $a_h(\cdot,\cdot)$ is defined by (3.1) and the convection part is determined by (3.5). The up-winding will bring additional error term and we modify the above argument in the following way. From $a_h(u_h, v^*) = F(v^*)$ and $a(u, v^*) = F(v^*)$ for $v^* \in V_h^*$ we get the orthogonality condition:

$$a(u, v^*) - a_h(u_h, v^*) = 0.$$

Following (4.1), the estimate for the error in the energy norm now becomes

$$\begin{aligned} c_0||e||_1^2 \leq a(e,e) &= a(e,e) - a(u,v^*) + a_h(u_h,v^*) \\ &= \{a(e,e) - a(e,v^*)\} + \{a_h(u_h,v^*) - a(u_h,v^*)\}. \end{aligned}$$

For the first term, $a(e,e) - a(e,v^*)$, we use the already derived estimate. For the second term, $a_h(u_h,v^*) - a(u_h,v^*)$, we get

$$\begin{aligned} a_h(u_h,v^*) - a(u_h,v^*) &= \sum_{x_i \in N_h^0} v_i^* \int\limits_{\partial V_i} \left[(\underline{b}\cdot\underline{n}_i)_+ u_h(x_i) + (\underline{b}\cdot\underline{n}_i)_- u_h(x_j) - \underline{b}\cdot\underline{n}_i u_h \right] ds \\ &= \sum_{T\in\mathcal{T}_h} \sum_{\gamma_{ij}\subset T} (v_i^* - v_j^*) \int\limits_{\gamma_{ij}} \underline{b}\cdot\underline{n}(u_h(x_i) - u_h) ds. \end{aligned}$$

In the last equality $\underline{n}$ is taken to be the normal to γ_{ij} such that $\underline{b}\cdot\underline{n} \geq 0$ and the indices (ij) are such that $(x_i - x_j)\cdot\underline{n} \leq 0$. We denote by $[v^*]$ the jump of v^* across γ_{ij} and by $R_{\gamma_{ij}}$ the expression $\underline{b}\cdot\underline{n}(u_h(x_i) - u_h)|_{\gamma_{ij}}$. Then, by Schwartz inequality, we get the bounds

$$a_h(u_h,v^*) - a(u_h,v^*) \leq \sum_{T\in\mathcal{T}_h} \sum_{\gamma_{ij}\subset T} |||[e-v^*]|||_{\gamma_{ij}} ||R_{\gamma_{ij}}||_{\gamma_{ij}} \leq \sum_{T\in\mathcal{T}_h} w_T^\gamma \; h_T^{1/2} ||R_{\gamma_T}||.$$

In the last inequality we have used the notations

$$w_T^\gamma = h_T^{-1/2} \left(\sum_{\gamma_{ij}\subset T} |||[e-v^*]|||_{\gamma_{ij}}^2 \right)^{1/2}, \; ||R_{\gamma_T}|| = \left(\sum_{\gamma_{ij}\subset T} ||R_{\gamma_{ij}}||_{\gamma_{ij}}^2 \right)^{1/2},$$

where $||\cdot||_{\gamma_{ij}}$ denotes the L^2-norm on γ_{ij}. Again, by the local approximation properties of the finite volume elements we have $w_T^\gamma \leq C_{I,T}||\nabla e||_T$, where $C_{I,T} \approx 1$ is an interpolation constant (see [8]). This means that we have to add to the local residuals ρ_T additional term $h_T^{1/2}||R_{\gamma_T}||$, i.e.

$$\rho_T := h_T||R_T||_T + h_T^{1/2}||R_{\partial T}||_{\partial T} + h_T^{1/2}||R_{\gamma_T}||,$$

and proceed for the equilibration of the local residuals ρ_T as in the previous case.

5. Data structure and implementation. Here we describe our computational strategy and the developed set of object-oriented structures useful in solving the class of problems given in Section 2. First, in Subsection 5.1 we introduce the mesh generator *NETGEN*. The overall code structure is given in Subsection 5.2. Finally, in Subsection 5.3 we give a short description of the solvers that have been implemented in the code.

5.1. Adaptive mesh generation. Finding a "good" computational mesh is one of the key elements in the development of any efficient computational methodology based on finite element or finite volume method. Both methods require partitioning a given domain into a set of elements (coarse mesh), which have certain regularity properties. Additionally, in order to produce an approximation within a given tolerance, adaptive mesh refinement, based on a posteriori error analysis, has to be used.

We have used *NETGEN* for generating coarse meshes. This is a 3-D stand-alone mesh generator based on advancing front method. The input is 3-D domain

described by boolean operations (or, and, not) on primitives, such as planes, cylinders, spheres, cones and tubes. The splitting is into tetrahedra. *NETGEN* is developed by *Joachim Schöberl*, Johannes Kepler University, Linz, Austria. More information about *NETGEN* can be found on *Joachim Schöberl*'s homepage [1]. The software is free for non-commercial applications and is available for *Unix/Linux* and *Windows 98/NT* platforms.

Our adaptive mesh generation is based on the bisection algorithm (see, e.g. Arnold et al. [2]). The algorithm features data structure that simplifies both the selection of refinement edge and the recursive refinement to conformity once some tetrahedra have been refined. Repeated application of the algorithm leads to only finitely many tetrahedral shapes, i.e. tetrahedra shape cannot degenerate as the mesh is refined.

5.2. Code structure. The code is written in *C++* and has object-oriented structure. The implementation is done in the framework of multilevel refinement and the corresponding problems on the composite grid can be solved in a multilevel fashion. There are three main classes and their dependencies. These are classes *Mesh*, *Method* and *Matrix*.

Class *Mesh* keeps the data for the mesh. Our mesh is composed of tetrahedra, faces, vertices and edges (if needed). All these are objects of the corresponding classes given below. In *Mesh* they are given as *vectors*. The simplest and in many cases the most efficient *STL* container class is **vector**. It supports random access to elements (fast as array), constant time insertion and removal of elements at the end. The number of elements in a vector may vary dynamically, memory management is automatic. We have also the following classes

- *tetrahedron* – four node and face indices and attribute;
- *face* – three node indices and two pointers to the neighboring tetrahedra; if the face is on the boundary the second pointer gives the type of the boundary;
- *edge* – if the edges are generated they are composed of two node indices;
- *vertex* – three coordinates and attribute.

The *Mesh* also maintains information on how the nodes are connected, which is used in storing the stiffness matrix. Similar information, giving the connectivity between new nodes, is generated (if needed) for hierarchical bases. Interpolation information, for example, giving that node k is between i and j, may also be generated. The latter may be used in multi-grid preconditioner and for derefinement, when nodes are deleted in order to coarsen the mesh. Using the interpolation information one can restore the previous mesh and from there to start some other refinement procedure.

The most important *Mesh* routine is the one for local refinement. Its input is an array of indices of the elements that have to be refined. The result is a new level of the mesh, refined to conformity, after the given elements were bisected.

Class *Method* is used to *encapsulate* and *prepare* the data necessary for a finite element (or finite volume) method solve. The class inherits *Mesh* and according to its construction input arguments creates on each level the necessary matrices (stiffness, interpolation and so on). The created stiffness matrix is an object of class matrix, described below. It is passed to *PCG* or *GMRES* routine, described in Section 5.3. If the exact solution is available other useful routines are the ones for computing the approximation error in discrete *energy*-norm, L^2-norm and *maximum*-norm.

To refine the mesh, class *Method* has methods for computing *Zienkiewicz-Zhu*, *Residual based* and *Hierarchical* error estimates. Also, there is option were the local

[1] `http://www.sfb013.uni-linz.ac.at/~joachim/`

refinement is governed by user defined function. All implemented error indicators take as argument some error tolerance and the approximate solution. The output is a list of tetrahedra, which based on the error estimate and the tolerance, have to be refined. This list is the input for the local refinement routine.

Class *Matrix* encapsulates the information about the matrix. We have a sparse storage format. The vertex connectivity information is generated and stored in *Mesh*. *Matrix* has only pointers to that information. The implemented routines are *Matrix-Vector* product, *Matrix Transpose-vector* product, *Matrix-Matrix* product, *Gauss-Seidel forward* and *Gauss-Seidel backward* iteration (used as smoother in multi-grid preconditioner).

5.3. Solvers. To solve the symmetric positive definite system arising in approxiamtion of the pressure equation and the non-singular non-symmetric system (arising from the concentration euquation) we have implemented and used *Preconditioned Conjugate Gradient (PCG)* and *Generalized Minimum Residual (GMRES)* methods, correspondingly. Both methods are implemented as *C++* templates. There is a provision for using a preconditioner if one is available. The templates rely on matrix, which provides matrix-vector product, routines for computing Bx, (x, y) and $ax + by$, where x and y are vectors, a and b scalars, B is a preconditioner to the stiffness matrix. The solvers can estimate the conditioner number using *LAPACK* library.

For symmetric and positive definite problems we have implemented multi-grid preconditioner with *Gauss-Seidel* smoother. We have also hierarchical smoother, where the smoothing is performed only over the new nodes. Regarding the preconditioners for both symmetric and non-symmetric problems we have developed *Domain Decomposition* structure [16] for parallel computations.

6. Computational results. Our computational results demonstrate the performance of the adaptive grid refinement strategy on verious problems with singularities due to discontinuity of the boundary data and also due to corners, edges, and wells.

6.1. Problems with corner and edge singularities. Here we consider the cases of corner and edge singularities.

Edge singularity. We solve the problem (2.1) with $f = 0$ and *Dirichlet* boundary $\Gamma_D = \partial\Omega$ such that the solution is a harmonic function given in cylindrical coordinates by $u(r, \theta, z) = r^{2/3} sin\frac{2\theta}{3}$ (note that u is not in $H^{5/3}(\Omega)$). Here Ω is the L shaped domain given on Figure 6.1 (left).

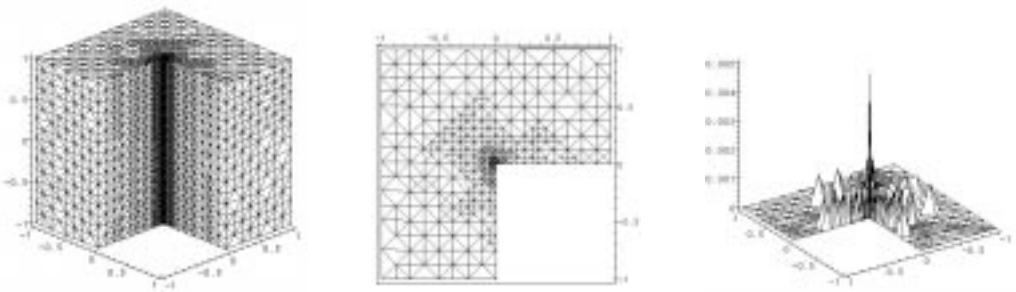

FIG. 6.1. *L-shaped domain in* 3*-D with solution given in cylindrical coordinates by* $u(r, \theta, z) = r^{2/3} sin\frac{2\theta}{3}$*; the* 3*-D Mesh on level* 6 *(left) with 28,768 nodes, the mesh (middle) and the error (right) for* $z = 0$*.*

We have singularity along the edge determined by the points $(0, 0, -1)$ and $(0, 0, 1)$.

As expected, the mesh shown on Figure 6.1 is repeatedly refined near this edge. On Figure 6.1 we show the 3-D mesh (left) on the sixth refinement level (see also Table 6.1), the mesh for $z = 1$ (middle) and the error (right) at $z = 0$ (again on level six). Table 6.1 gives the results for the mesh and the error for the different levels of

TABLE 6.1
Local Refinement

level	# nodes	$\|\|e\|\|_{max}$	$\|\|e\|\|_{L^2}$	$\|\|e\|\|_1$
1	509	0.050494	0.028660	0.123884
2	2,420	0.034936	0.010644	0.078138
3	8,813	0.022735	0.004112	0.049559
4	17,312	0.014491	0.001757	0.033087
5	22,448	0.009164	0.001048	0.024555
6	28,768	0.005807	0.000850	0.020348

TABLE 6.2
Uniform Refinement

level	# nodes	$\|\|e\|\|_{max}$	$\|\|e\|\|_{L^2}$	$\|\|e\|\|_1$
1	509	0.050494	0.028660	0.123884
2	3,333	0.034952	0.010643	0.077947
3	23,817	0.022750	0.004073	0.048743
4	179,729	0.014511	0.001586	0.030489
5	1,395,745	0.009186	0.000625	0.019107

refinement obtained using Zienkiewicz-Zhu error estimator. Error tolerance of 2% in the energy norm was imposed and obtained on level 6. Indeed, the energy norm of the exact solution is approximately $\sqrt{\pi/8} \approx 0.627$, i.e. on the last level we have approximately 3% error in the energy norm. The errors on the last level in the discrete L^2 and *maximum* norms are correspondingly 0.06% and 0.46%.

In order to compare these results with the results when no local refinement is applied in Table 6.2 we have included computations with *uniform* refinement. Here by *uniform* we mean that every tetrahedron has been split into 8 tetrahedra. The accuracy of the solution obtained on a mesh with $1,395,745$ nodes (after 5 levels of refinement) is comparable to the accuracy of the locally refined grid with $28,768$ nodes on level 6. Note, that the locally refined grid has about 40 times less grid points than the uniform grid.

The other error indicators give similar results with small differences in the obtained meshes. In all cases, the meshes are refined in the area where the solution has some type of singularity or very steep gradient and quantitatively they have almost the same number of nodes.

Corner singularity in 3-D. Now we shall demonstrate the local refinement performance for problems where the singularity is located at a corner. We consider two problems and show on Figure 6.2 the obtained computational meshes after few levels of refinement.

The first test is again for the homogeneous Poisson equation with $f = 1$. We consider an L-shaped domain Ω shown on Figure 6.2 (left). Here the singularity is due to the non-convex corner at the origin. The second test is for the Poisson in the unit cube shown on Figure 6.2 (right) with Dirichlet boundary condition on $x = 0$, $y = 0$, $z = 0$ and Neumann boundary conditions on the rest of the boundary. The

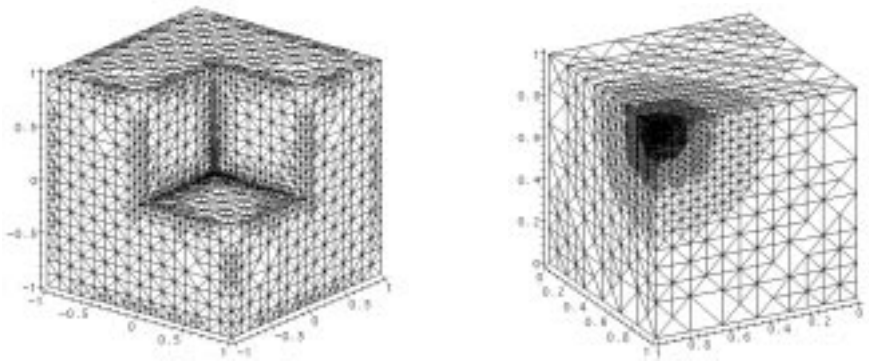

FIG. 6.2. *Computational meshes obtained by local refinement due to L-shaped corner singularity (left) and delta function source term (right). Both meshes are for level six with correspondingly* $12,350$ *and* $11,770$ *nodes.*

singularity is due to the source term, which is taken to be δ-function concentrated at the corner $(1,1,1)$. In both cases, as expected, the meshes are refined around the corners where the singularities are located.

6.2. Simulation results for fluid flow in porous media. In this section we show the results from applying the developed computational technology to fluid flow problems in porous media. The problems that we consider are based on real data and the setting is described in below. The mathematical model is based on problems (2.1) and (2.4) and includes the well model described in Section 2. The singularities, coming from jumps in the input data, the wells, domain non-homogeneity and so on, make the application of local grid refinement essential.

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 (2.4), 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 (2.1) for fully saturated porous media under appropriate boundary conditions. We consider two cases: (1) homogeneous reservoir; and (2) non-homogeneous reservoir.

Homogeneous reservoir. The pressures at the the faces $x_1 = 0$ and $x_1 = 1000$ are constants, correspondingly 1000 and 0 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 $K = k_{diff} I + k_t \underline{v}^T \underline{v}/|\underline{v}| + k_l(|\underline{v}|^2 I - \underline{v}^T \underline{v})/|\underline{v}|$, 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 6.3 for the case of low biodegradation rate $a = 0.1$ *mg/yr* and on Figure 6.4 for medium biodegradation rate $a = 0.2$ *mg/yr*. We have started with an initial coarse

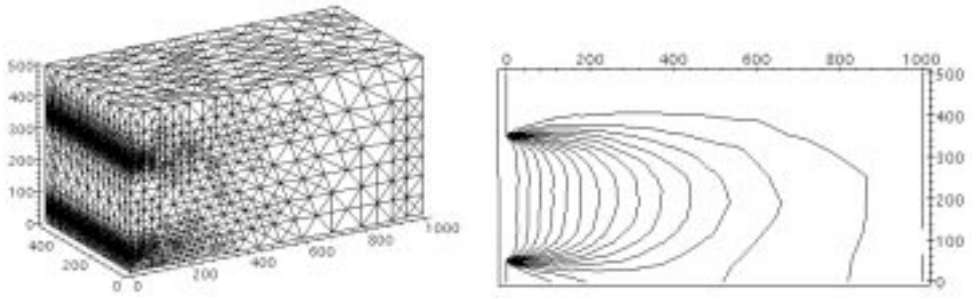

FIG. 6.3. *Homogeneous reservoir; the* 3*-D mesh on refinement level* 6 *(left) with 54,129 nodes; the concentration level curves (right) in the plane* $x_2 = 0$ *for the case of low biodegradation rate*

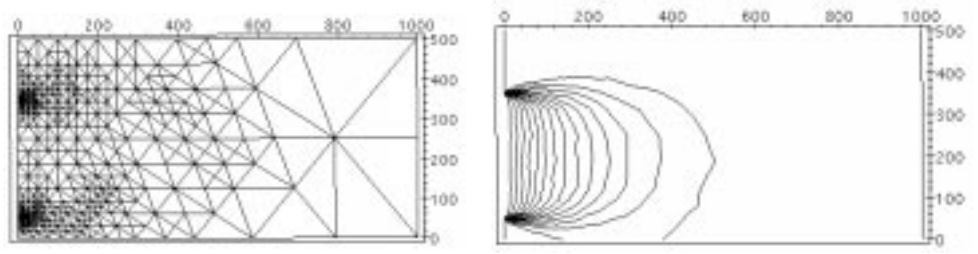

FIG. 6.4. *Homogeneous reservoir; the mesh in plane* $x_2 = 0$ *on refinement level* 5 *(globally with 36,427 nodes); the concentration level curves in plane* $x_2 = 0$ *for the case of medium biodegradation rate*

mesh with 52 nodes.

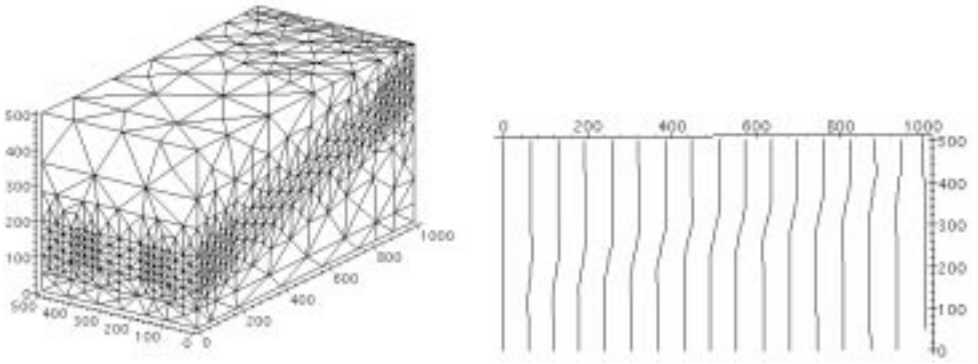

FIG. 6.5. *Pressure computations for a non-homogeneous reservoir; (left) the locally refined* 3*-D Mesh on level* 2 *with 4,053 nodes; (right) Contour curves of the pressure for level 2*

Non-homogeneous reservoir. Here the problem setting is as above but a layer is added (see Figures 6.5, 6.6, and 6.7). 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 6.5. After the pressure is found with prescribed accuracy we solve the corresponding problem for the concentration. Figure 6.6 shows the obtained mesh and the isolines for the concentration

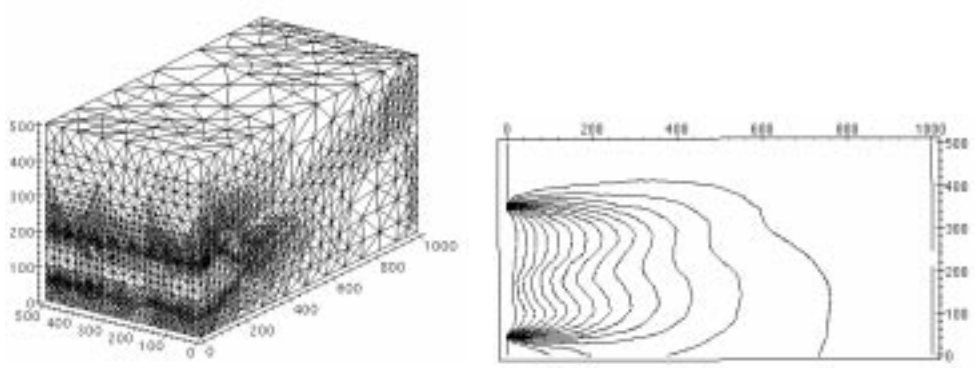

FIG. 6.6. *Concentration computations for a non-homogeneous reservoir; (left) the* 3*-D mesh on refinement level* 4 *with 39,445 nodes; (right) contour curves of the concentration for the cross-section* $x_2 = 0$*; the permeability in the layer is two time smaller than the rest of the domain*

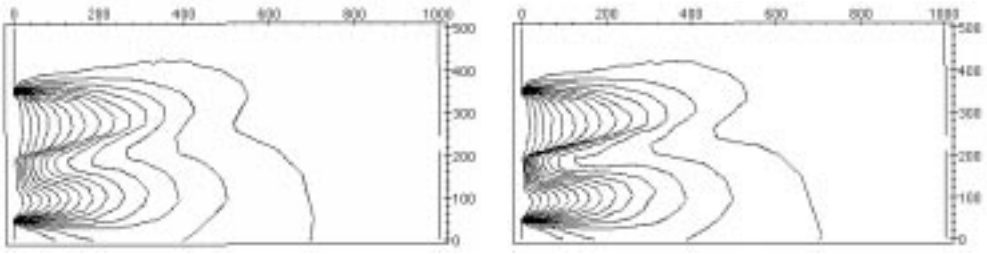

FIG. 6.7. *Concentration computations for a non-homogeneous reservoir; contour curves of the concentration for permeabilities in the layer* 5 *times (left) and* 10 *times (right) smaller than the permeability in the rest of the domain*

in the reservior cross-section $x_2 = 0$ on grid refinement level 4. Two more experiments varying the permeability tensor are shown on Figure 6.7. The first one shows the concentration isoline in the reservior 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 reservior. The initial coarse mesh in both cases has 235 nodes.

Non-homogeneous reservoir 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 6.8 we show the adapted mesh and the level curves for the pressure in the reservior cross-section $x_2 = 0$. On Figure 6.9 we show the obtained computational mesh and the level curves for the concentration in the reservior cross-section $x_2 = 0$ on grid refinement level 5.

7. Conclusions. We have presented a computational methodology for adaptively solving second order elliptic problems (diffision and convection-diffusion). We begin the solution process with an initial coarse mesh which describes adequately the given problem (domain, coefficients, boundary conditions, and right hand side). During the solution process the grid is refined (based on a criteria formulated from one of the three error estimators) until maximum refinement level is reached or the local error is found below a given threshold δ. The grids obtained from all error estimators differ slightly, but in all cases they are refined in the same areas and produce compa-

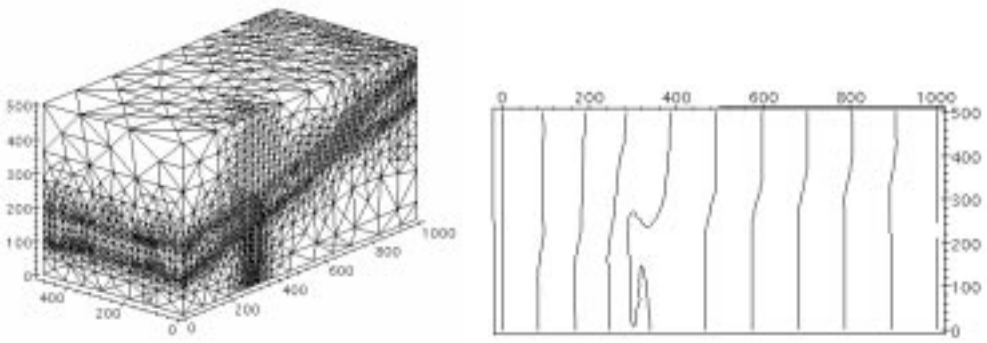

FIG. 6.8. *Pressure computations for a non-homogeneous reservoir with a well; (left) the* 3*-D Mesh on level* 3 *with 34,236 nodes; (right) contour curves of the pressure on level* 3

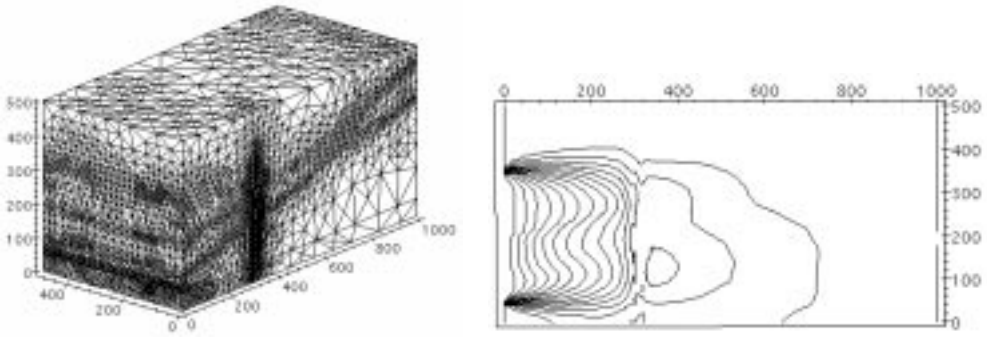

FIG. 6.9. *Concentration computations for a non-homogeneous reservoir with a well; (left) the* 3*-D mesh with 67,509 nodes in half of the domain obtained after* 5 *levels of refinement; (right) contour curves of the concentration in the plane* $x_2 = 0$

rable results. The proposed methodology is part of our tool-box for adaptive parallel simulation of flow and transport in porous media (see for details [12]).

Acknowledgment. Part of this research has been done during the summer visits of the authors to Lawrence Livermore National Laboratory. The authors thank the Institute for Scientific Computing Research and the Center for Applied Scientific Computing for thier hospitality and for the technical and financial support.

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] D. N. ARNOLD, A. MUKHERJEE, AND L. POULY, *Locally adapted tetrahedral meshes using bisection*, SIAM J. on Sci. Computing (to appear).

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

[4] 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.

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

[6] R. E. BANK AND R. K. SMITH, *A posteriori error estimates based on hierarchical bases*, SIAM J. Numer. Anal., 30 (1993), pp. 921–932.

[7] R. BECKER, C. JOHNSON, AND R. RANNACHER, *Adaptive error control for multigrid finite*

element methods, Computing, Springer-Verlag, 1995.

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

[9] K. Eriksson and C. Johnson, *An Adaptive Finite Element Method for Linear Elliptic Problems*, Math. Comp., 50 (1988), pp. 361–382.

[10] 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.

[11] T. Ikeda, *Maximum Principle in Finite Element Models for Convection-Diffusion Phenomena*, Lecture Notes in Numer. Appl. Analysis, Vol. 4, North-Holland, Amsterdam New York Oxford, 1983.

[12] R. D. Lazarov and S. Z. Tomov, *Tool-box for large scale parallel simulation of 3-D flow and transport in porous media*, Texas A&M University, TAMU, ISC Technical Report, 2000 (in preparation).

[13] H.-O. Ross, M. Stynes, and L. Tobiska, *Numerical Methods for Singularly Perturbed Differential Equations*, Springer, 1996.

[14] 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.

[15] M. Tabata, *A finite element approximation corresponding to the upwind finite differencing*, Mem. Numer. Math., 4 (1977), pp. 47–63.

[16] 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).

[17] R. Verfürth, *A Review of a Posteriori Error Estimators and Adaptive Mesh Refinement Techniques*, Teubner-Wiley, Stuttgart, 1996.

[18] R. Verfürth, *A posteriori error estimators for convection-diffusion equations. Numer. Math.*, 80 (1998), pp. 641–663.

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

[20] 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.