Adaptive Moving Mesh Methods for Simulating One Dimensional Groundwater Problems with Sharp Moving Fronts

Weizhang Huang [*], Li Zheng [†], Xiaoyong Zhan [‡]

May 1, 2001

Abstract

Accurate modeling of groundwater flow and transport with sharp moving fronts often involves high computational cost, when a fixed/uniform mesh is used. In this paper, we investigate the modeling of groundwater problems using a particular adaptive mesh method called the moving mesh partial differential equation approach. With this approach, the mesh is dynamically relocated through a partial differential equation to capture the evolving sharp fronts with a relatively small number of grid points. The mesh movement and physical system modeling are realized by solving the mesh movement and physical partial differential equations alternately. The method is applied to the modeling of a range of groundwater problems, including advection dominated chemical transport and reaction, nonlinear infiltration in soil, and the coupling of density dependent flow and transport. Numerical results demonstrate that sharp moving fronts can be accurately and efficiently captured by the moving mesh approach. Also addressed are important implementation strategies, e.g. the construction of the monitor function based on the interpolation error, control of mesh concentration, and two-layer mesh movement.

keyword: groundwater modeling, mesh adaption, mesh movement

1 Introduction

Great concern over world wide groundwater problems has prompted intensive research in groundwater modeling in the past three decades. Much progress has been made, yet the

[*]Department of Mathematics, The University of Kansas, Lawrence, KS 66045, U.S.A. (*huang@math.ukans.edu*).

[†]Kansas Geological Survey, The University of Kansas, Lawrence, KS 66047, U.S.A. (*lizheng@kgs.ukans.edu*).

[‡]Kansas Geological Survey, The University of Kansas, Lawrence, KS 66047, U.S.A. (*zhan@kgs.ukans.edu*).

quest for more accurate and efficient methods remains active, especially in solving problems involving sharp moving fronts. Conventional methods tend to produce oscillatory solutions and excessive numerical dispersion in regions around sharp fronts for such problems. A current remedy is a type of methods referred to as the mixed Eulerian-Lagrangian method (see [21] for a detailed review), which solves, for example, the advection part of the transport problem using particle tracking in the Lagrangian frame while simulating the dispersion part in the Eulerian frame.

Many have come to recognize mesh adaption as an effective tool for simulating sharp fronts and reducing numerical dispersion and oscillation. It has been amply demonstrated that significant improvements in accuracy and efficiency can be gained by adapting the mesh nodes so that they remain concentrated in regions of sharp fronts. There have been a number of applications of mesh adaptation in groundwater modeling. For example, Yeh et al. [19] combine the method of characteristics (MOC) and the modified MOC (MMOC) with local grid refinement and effectively avoid the numerical dispersion and oscillation problem in solving advection-dispersion equations. Trompert [18] applies local-uniform-grid refinement to modeling transport in heterogeneous media. Gottardi and Venutelli [12] apply the moving finite element method (MFE) of Miller and Miller [16] to one-dimensional infiltration problems. Gamliel and Abriola [10, 11] modify the MFE and apply it to the simulation of multi-phase flow in porous media. Zegeling et al. [20] analyze the coupling of density dependent flow and transport using the moving grid method developed by Dorfi and Drury [8].

The so called moving mesh approach, as advocated by Miller and Miller [16] and Zegeling et al. [20], differs from the grid refinement [18, 19] in that the former keeps the same number of mesh points throughout the entire solution process. Thus, the size of computation and data structure are fixed, which enables a much easier implementation. The widely used method of characteristics (MOC) can also be considered as a type of moving mesh method. MOC implicitly defines the coordinate transformation using the characteristic equation of the partial differential equation (PDE) and moves the mesh points along the characteristic lines. However, the MOC often generates a very skewed mesh which is not suitable for use in simulating other physical processes such as dispersion. Indeed, the MOC is known to be more feasible for the solution of hyperbolic systems. It is interesting to point out that for its connection to the MOC or the Lagrangian method, the moving mesh method is sometimes referred to as the quasi-Lagrangian method.

In this paper, we consider a particular moving mesh method, the so-called moving mesh PDE (MMPDE) approach proposed by Huang et al. [14, 15]. With this approach, adaptive moving meshes are generated as images of a fixed (computational) mesh in the auxiliary domain through a time dependent coordinate transformation or the solution of an MMPDE. The mesh points are thus continuously relocated and dynamically adapted to solution behavior, which provides an ideal adaptive strategy to capture evolving sharp fronts without using a large number of grid points. The MMPDE is defined as the gradient or heat flow equation of a functional, which is related to the well-known equidistribution

principle [7] and measures the difficulty in approximating the solution. The method has been successfully applied to various one- and two-dimensional problems, e.g. see [3, 4]. Theoretical and numerical studies have shown that the approach generates non-singular meshes at least in one and two dimensions [3, 14, 15].

In the rest of this paper, we first introduce the theoretical base and methodology of the MMPDE approach. Several implementation strategies are then explored. In particular, we investigate in detail the proper choice of the monitor function, control of mesh adaption, and two-layer mesh movement. A numerical example is used to demonstrate the effectiveness of the implementation strategies. We apply the method to a variety of one-dimensional groundwater problems, including advection dominated chemical transport and reaction, nonlinear infiltration in soil, and the coupling of density dependent flow and transport. The accuracy and efficiency of the MMPDE approach are demonstrated. Conclusions and comments are given in the final section.

2 Numerical Method

For simplicity we describe the numerical method only for the advection-dispersion equation

$$\frac{\partial u}{\partial t} + V\frac{\partial u}{\partial x} = D\frac{\partial^2 u}{\partial x^2}, \quad x \in (0,1) \text{ and } t > 0 \tag{1}$$

where D is the dispersivity and the flow velocity is taken as $V = 1$. The method can be straightforwardly extended to other systems.

2.1 Adaptive Moving Mesh Method

We use here the MMPDE method first proposed in [14, 15]. Specifically, the method is based on a coordinate transformation $x = x(\xi, t) : \Omega_c \equiv (0,1) \to \Omega \equiv (0,1)$ between the physical and computational domains. Adaptive moving meshes are then generated as images of a fixed, uniform computational mesh under the coordinate transformation. Hereafter the mesh and the coordinate transformation are used synonymously.

The basic idea behind the MMPDE moving mesh method is that a transient partial differential equation, often referred to as a moving mesh PDE (MMPDE), is employed to define the coordinate transformation. In one dimension, the equidistribution principle [7] plays an important role in mesh adaption. It employs the so-called monitor function to measure the difficulty in the numerical solution of the physical PDE and places more mesh points in regions where the monitor function is large. The equidistribution can be achieved by minimizing the functional

$$I[\xi] = \frac{1}{2}\int_0^1 \frac{1}{g}\left(\frac{\partial \xi}{\partial x}\right)^2 dx,$$

where $\xi = \xi(x,t)$ is the inverse coordinate transformation of $x = x(\xi,t)$ and $g = g(x,t) > 0$ is the monitor function. The MMPDE is then defined as a modified gradient (or heat) flow equation of $I[\xi]$, i.e.,

$$\tau \frac{\partial \xi}{\partial t} = -\frac{1}{p}\frac{\delta I}{\delta \xi} \qquad \text{or} \qquad \tau \frac{\partial \xi}{\partial t} = \frac{1}{p}\frac{\partial}{\partial x}\left(\frac{1}{g}\frac{\partial \xi}{\partial x}\right),$$

where p is a to be determined positive function, $(\delta I)/(\delta \xi)$ is the functional derivative, and τ is the user prescribed parameter used for adjusting the time scale of mesh movement. From the relation

$$\frac{\partial x}{\partial t} = -\frac{\partial x}{\partial \xi}\frac{\partial \xi}{\partial t}$$

that can be obtained by differentiating $x = x(\xi(x,t),t)$ with respect to t while x is fixed, we can rewrite the MMPDE into

$$\tau \frac{\partial x}{\partial t} = \frac{1}{p(g x_\xi)^2}\frac{\partial}{\partial \xi}\left(g x_\xi\right),$$

where $x_\xi = (\partial x)/(\partial \xi)$. As in [13], we choose $p = \sqrt{g^2 + g_\xi^2}/(g x_\xi)^2$, where $g_\xi = (\partial g)/(\partial \xi)$, in order to make the resultant mesh equation invariant under transformations $g \to c_1 g$ and $x \to c_2 x$ for any positive constants c_1 and c_2. Thus, we obtain

$$\tau \frac{\partial x}{\partial t} = \frac{1}{\sqrt{g^2 + g_\xi^2}}\frac{\partial}{\partial \xi}\left(g\frac{\partial x}{\partial \xi}\right). \tag{2}$$

Generally speaking, it is not difficult to choose a value for τ. Our limited experience shows that a value of 10^{-2} or 10^{-3} works well for most problems. In contrast, the choice of g is much more difficult. Since the success of the moving mesh strategy relies crucially on the choice of g, we leave the discussion to the next subsection.

Several features of (2) are worth mentioning. (2) is nonlinear but parabolic. (2) is always numerically integrable. Moreover, it can be shown that the solution or the coordinate transformation exists and is non-singular. Further, the equidistribution principle can be obtained by simply taking $\tau = 0$. In this sense, τ can be regarded as a relaxation parameter and (2) as an approximation of the equidistribution principle. The smaller τ is, the prompter the mesh responds to the change of the monitor function.

The coupled system of the physical and mesh equations (1) and (2) must be solved for both the physical solution and the mesh. We use here the method of lines approach and discretize MMPDE (2) and physical PDE (1) in space via standard central finite differences. We note that (1) is discretized through its transformed form in the computational coordinate, i.e.

$$\frac{\partial v}{\partial t} + \frac{(V - x_t)}{x_\xi}\frac{\partial v}{\partial \xi} = \frac{D}{x_\xi}\frac{\partial}{\partial \xi}\left(\frac{1}{x_\xi}\frac{\partial v}{\partial \xi}\right), \tag{3}$$

where $v = v(\xi,t) \equiv u(x(\xi,t),t)$.

The resultant system of ordinary differential equations can be integrated either simultaneously or alternately. Note that the coupled system is generally highly nonlinear, even when the original physical PDE is linear. This can be easily seen from the transformed physical PDE (3) where x_ξ appears in denominators. Also, the size of the coupled system is larger than that of the original one. This increase in size is more significant in higher dimensions. These features of the coupled system make the simultaneous solution procedure less attractive than the alternating procedure. Based on this consideration, we adopt the following alternating solution procedure.

Alternating Procedure: Assume that the physical solution v^n, mesh x^n, and time step Δt_n are given at time $t = t_n$.

(i) Compute monitor function $g(x, t_n)$ using u^n and x^n and smooth it. Note that the monitor function is understood as a continuous function of x in the sense of interpolation.

(ii) Integrate the MMPDE from $t = t_n$ to $t = t_n + \Delta t_n$ using the backward Euler scheme with variable step sizes. More than one sub-step may be necessary before $t = t_n + \Delta t_n$ is reached. When this happens, the monitor function is updated from mesh to mesh via linear interpolation. The obtained mesh is denoted by x^{n+1}.

(iii) Integrate the physical PDE over a fixed or variable step. The mesh and mesh speed at the intermediate stages are calculated through

$$x(t) = \frac{t - t_n}{\Delta t_n} x^{n+1} + \frac{t_n + \Delta t_n - t}{\Delta t_n} x^n. \tag{4}$$

(iv) When a variable step size is used in step (iii), the physical PDE may actually be integrated over a smaller step $\hat{\Delta} t_n < \Delta t_n$. In this case, the mesh at the actual new time level $t_{n+1} := t_n + \hat{\Delta} t_n$ should be recalculated as $x^{n+1} := x(t_n + \hat{\Delta} t_n)$ before marching on to the next time step.

(v) Go to the next step with either the fixed step size or Δt_{n+1} predicted in (iii) by the physical solver.

It is emphasized that more than one sub-step in (ii) for integrating the MMPDE is often necessary. This is because the time scale of the MMPDE or the mesh movement relies on the choice of τ and is often different from the time scale of the physical PDE. Time step Δt_n, predicted by the physical solver, is thus inadequate for integrating the MMPDE, and a different (and often smaller) time step should be used to maintain the stability and accuracy of mesh movement. Following [13], we impose the condition $\Delta t_{\text{mesh}} \leq \min\{\Delta t_n/3, 0.1\tau\}$ on the sub-step sizes. Under this condition, at least three sub-steps are taken in step (ii), and the scaled step for mesh movement $\Delta t_{\text{mesh}}/\tau$ is less than 0.1.

In our computation we take $\tau = 10^{-3}$ and integrate the physical PDE using the third order singly diagonally implicit Runge-Kutta (SDIRK) scheme [5]. An error estimate

e, provided by the SDIRK scheme as a by-product, is used for controlling the step size according to the standard technique in the context of numerical ODEs, viz.

$$\Delta t_{n+1} = \Delta t_n \min\left(4, \max\left(0.1, 0.84\sqrt{\frac{tol}{\|e\|_\infty}}\right)\right),$$

where $\|e\|_\infty$ is the maximum norm of e and the tolerance tol is taken as 10^{-6}.

2.2 Monitor Functions

The key to the success of the above described mesh movement method is to define an appropriate monitor function g. Inappropriately chosen monitor functions can result in wrong mesh concentrations and also make the MMPDE difficult to integrate.

2.2.1 Monitor functions based on solution gradients

One of the most commonly used, one dimensional monitor functions is the arc-length monitor function

$$g = \sqrt{1 + u_x^2}, \tag{5}$$

where $u_x = (\partial u)/(\partial x)$, which tends to concentrate mesh points in regions where the solution gradient is large. This seems reasonable and intuitive, because more mesh points are concentrated in the regions where the solution changes significantly. Another and perhaps more important feature is that the monitor function is much smoother than other monitor functions based on higher order derivatives. Thus, the corresponding mesh equation can be easier to integrate, see [2] for the comparison study of the arc-length and curvature monitor functions. However, it should be pointed out that the error of a numerical approximation is often related to higher order derivatives of the solution, and thus the arc-length monitor function may not lead to optimal mesh adaption and can even fail for some cases.

2.2.2 Monitor functions based on interpolation error indicators

It is clear that monitor functions must be computed based on some sort of error estimates. The comparison study in [3, 4] for two dimensional mesh movement suggests that the interpolation error estimate serve the purpose best in terms of efficiency and ability to accurately predict the large error regions where higher mesh concentration is needed.

Following the two dimensional study in [4], we compute the interpolation error estimate and define the monitor function as follows. Denote by $p^L(x)$ the piecewise linear polynomial connecting points (x_j, u_j), $j = 0, ..., J$, where x_j is a mesh point and u_j is an approximation of the solution at this point. Let $I_j = (x_j, x_{j+1})$, $x_{j+1/2} = (x_j + x_{j+1})/2$, and $du_{j+1/2} = (u_{j+1} - u_j)/(x_{j+1} - x_j)$. Consider node x_j and its two neighboring cells I_{j-1} and I_j. Compute the linear polynomial $p_j(x)$ in $I_{j-1} \cup I_j$ which passes through $(x_{j-1/2}, du_{j-1/2})$ and $(x_{j+1/2}, du_{j+1/2})$. Obviously p_j can be regarded as the derivative of

some quadratic interpolation polynomial (say p^Q) in $I_{j-1} \cup I_j$. Then the error indicator is defined as

$$e_j = \frac{1}{|I_{j-1}| + |I_j|} \sqrt{\int_{I_{j-1}} \left| \frac{d}{dx} (p^Q - p^L) \right|^2 dx + \int_{I_j} \left| \frac{d}{dx} (p^Q - p^L) \right|^2 dx}, \tag{6}$$

where $|I_{j-1}|$ and $|I_j|$ denote the lengths of I_{j-1} and I_j, respectively. Accordingly, the monitor function is computed as

$$g_j = \sqrt{1 + e_j^2}. \tag{7}$$

2.2.3 Control of mesh adaption

It should be noted that the aforementioned monitor functions lack a mechanism to explicitly control the mesh adaption. Indeed, the monitor functions may over- or under-concentrate mesh points in the region of large solution variations or errors, upon the distribution of the solution gradient or the error indicator. Since the distribution of the solution gradient or the error indicator is different for different problems, the degree of mesh concentration will be different even if the same monitor function is used for solving these problems. In this sense, the monitor functions defined above are not robust.

A common remedy is to introduce the intensity parameter α into the monitor function. For example, the arc-length monitor function can be modified as

$$g = \sqrt{1 + \alpha u_x^2}.$$

Interestingly, Beckett and Mackenzie [1] introduce

$$g = 1 + \alpha |u_{xx}|^{1/m},$$

where m is an integer and α is computed automatically through

$$\alpha = \frac{1}{< |u_{xx}|^{1/m} >}$$

and $< \cdot >$ denotes the average. They consider finite difference solution of a singularly perturbed problem and obtain a uniform convergence rate.

Motivated by the idea of Beckett and Mackenzie, we define

$$g = 1 + \alpha v, \quad \alpha = \frac{\beta}{< v > (1 - \beta)}, \tag{8}$$

where $v = \sqrt{1 + u_x^2} - 1$ for the gradient based monitor function or $v = \sqrt{1 + e^2} - 1$ for the monitor function based on the error indicator, and $\beta \in (0, 1)$ is a user defined parameter. It is easy to see that when $v \approx < v >$, $g \approx 1/(1 - \beta)$ and a nearly uniform mesh results. Note that we use $v = \sqrt{1 + u_x^2} - 1$ instead of $v = |u_x|$ to avoid the non-smoothness at

$u_x = 0$. (Note that function $|x|$ is non-smooth at $x = 0$.) Introduction of parameter β allows us to control the mesh concentration. Indeed, it is not difficult to verify

$$\beta = \frac{\int_0^1 g dx - 1}{\int_0^1 g dx} = \frac{\alpha \int_0^1 v dx}{\int_0^1 g dx}. \tag{9}$$

In words, β is a good indicator of the percent of mesh points concentrated in the region of large v or g.

2.2.4 Smoothing of Monitor functions

It is common practice to smooth the monitor function in moving mesh methods. This is because the computed monitor function is often very non-smooth. At the same time, a smoother monitor function leads to a smoother mesh and also makes the MMPDE easier to integrate. In the spatial direction, we apply the low-pass filter four times to g,

$$g_j \leftarrow \frac{1}{4} g_{j-1} + \frac{1}{2} g_j + \frac{1}{4} g_{j+1}, \tag{10}$$

where $g_j = g(x_j, t)$.

We also found that temporal smoothing of the monitor function is often useful. This is especially true for the case of generating the initial adaptive mesh using the MMPDE approach, where the temporal smoothing helps to obtain convergent meshes. We use

$$g^n \leftarrow 0.2 g^{n-1} + 0.8 g^n, \tag{11}$$

where $g^n = g(x, t_n)$, in our computation although this has only a minor effect on time accurate integration.

2.3 Two-layer Mesh Movement

The efficiency of the moving mesh method can further be improved when combined with a two-layer mesh movement strategy [13]. With this strategy, the mesh movement is performed on a relatively coarse mesh whereas the physical PDE is solved on a finer mesh which is obtained from the coarse mesh via interpolation or a refinement procedure. This is reasonable since, unlike the physical solution, the locations of mesh points do not have to be calculated to high accuracy. A similar idea has been used by Fiedler and Trapp [9] for the dynamic generation of adaptive meshes using an elliptic differential equation system.

We use here a simple relation between the coarse and fine mesh. Denote the coarse mesh by $\{x_j^c, j = 0, ..., J^c\}$ and the fine mesh by $\{x_j, j = 0, ..., J\}$. They are related through the projection

$$x_j^c = x_{JC(j)}, \quad j = 0, ..., J^c \tag{12}$$

where the array JC is defined as

$$JC(j) = j \cdot JM, \quad j = 0, ..., J^c \tag{13}$$

with a prescribed integer JM. (Note that J and J^c must satisfy $J = J^c \cdot JM$.) This relation actually defines the $(j \cdot JM)$th node of the fine mesh as the jth node of the coarse one. To capture fine structures of the physical solution, the monitor function is computed on the fine mesh and then projected via area averaging to the coarse mesh. Having obtained the new coarse mesh by solving the MMPDE, we compute the fine mesh at the new time step via linear interpolation.

Note that the coarse mesh cannot be chosen too coarse to catch basic features of the physical solution. Our experience shows that choices $JM = 2, 3, 4$ can often lead to accurate results and good efficiency.

3 A Numerical Example

In this section numerical results obtained with the moving mesh method for model problem (1) are presented to demonstrate the basic features of the method. The Dirichlet boundary conditions at the left and right ends $x = 0$ and $x = 1$ are chosen such that the problem has the exact solution

$$u(x,t) = \frac{1}{2}\text{erfc}\left(\frac{x-t}{\sqrt{4Dt}}\right) + \frac{1}{2}\exp\left(\frac{x}{D}\right)\text{erfc}\left(\frac{x+t}{\sqrt{4Dt}}\right), \tag{14}$$

where $\text{erfc}(x)$ is the complementary error function. Note that this exact solution is singular at $t = 0$. To avoid this problem, we start the integration at $t = 10^{-4}$ and stop at $t = 1$. All the moving mesh computations start with an adaptive initial mesh, which is obtained by solving the MMPDE with the analytical initial solution until the L_2 norm of the mesh speed is less than 10^{-6}. The error is measured as

$$|||e||| = \int_0^1 ||u(\cdot,t) - u^{comp}(\cdot,t)||_{L^2} dt, \tag{15}$$

where $||u(\cdot,t) - u^{comp}(\cdot,t)||_{L^2}$ denotes the L_2 norm of the error at time t. The CPU time is listed in seconds on a single MIPS 300MHZ R12000 processor of SGI Origin2000.

First of all, we show in Fig. 1 the results obtained with uniform meshes to demonstrate the numerical difficulty in solving this simple model problem. Fig. 1a shows the computed solution obtained with 5121 uniform nodes for the case $D = 10^{-5}$. The computation takes 3407 seconds of the CPU time and has the error $|||e||| = 2.74 \times 10^{-3}$. The convergence history against the number of nodes is plotted in Fig. 1b for two cases, $D = 0.01$ and $D = 10^{-5}$. We can see that for the smooth case with $D = 0.01$, uniform meshes lead to very accurate results even for small numbers of nodes, and the second order convergence of the method (using central finite differences) is clear. However, for the difficult case with $D = 10^{-5}$, the situation is different. Reasonably accurate solutions cannot be obtained with small and even intermediate numbers of nodes. In fact, oscillations are visible behind the sharp front in the computed solution shown in Fig. 1a where 5121 nodes are used. The second order convergence is shown only for large numbers of nodes.

Fig. 2 shows a typical result obtained using the moving mesh method with 81 nodes for the case $D = 10^{-5}$. The monitor function based on the interpolation error indicator is used and the mesh concentration parameter is taken as $\beta = 0.5$. The solution at various time instants and the mesh trajectories clearly show the ability of the moving mesh method to catch the moving front while maintaining sufficient mesh points in the front region. The computation takes 79.3 seconds of the CPU time and the error is $|||e||| = 4.72 \times 10^{-4}$. Note that to get this level of error, more than 10241 uniform nodes have to be used and more than a factor of 65 CPU time is needed.

In Fig. 3 we plot the solutions obtained with the monitor functions based on the solution gradient and the interpolation error indicator. For both cases, the moving front is caught well, the mesh points are concentrated in the front region, and very mild oscillations can be seen in the solution. Comparatively, the monitor function based on the interpolation error indicator leads to more accurate results than the gradient based one does. It can also be seen that the mesh points are concentrated in different regions. For the gradient based monitor function, the points are concentrated and distributed uniformly throughout the region of large gradient, whereas the monitor function based on the error indicator results in more concentration at the two ends (where the solution curvature is large) than the middle of the large gradient region. This indicates that the gradient based monitor function concentrates mesh points in the wrong region. Fortunately, since the layer is very thin and the large error regions are very close to the large gradient region, the spatial smoothing of the monitor function helps to maintain enough mesh concentrations in the large error regions and therefore reasonably accurate solutions are obtained.

Next we demonstrate how mesh adaption (concentration) can be controlled with parameter β. We recall that β indicates the percent of mesh points concentrated in the region of large v or g. To see this, we define the actual percent of mesh points concentrated in the large g region as the ratio R of the number of the mesh points at which the value of the monitor function is strictly greater than its average to the total number of the mesh points. R is shown against time in Fig. 4 for various values of β and for the two monitor functions. The results clearly show that $R \approx \beta$, especially for the cases with 161 nodes. In other words, the control of mesh concentration defined in (8) is very effective, namely, a smaller value of β leads to lower mesh adaptivity while a larger value results in higher mesh concentration.

We show in Fig. 5 the computed solutions obtained with the monitor functions based on the solution gradient and interpolation error estimate and with/without concentration control. Since a relatively small number of nodes is used in the computation, there are not enough nodes concentrated in the sharp front area for both the case without control and the one with $\beta = 0.5$. As a result, oscillations are visible behind the front, especially when the gradient monitor function is used. On the other hand, when $\beta = 0.8$ is used, more nodes are concentrated in the front area and both the magnitude of the oscillation and the error are reduced considerably.

We list in Table 1 the error and CPU time for the computations with various fixed

and moving meshes for the two cases with $D = 10^{-2}$ and $D = 10^{-5}$. It can be expected that the central finite difference discretization on fixed and moving meshes leads to a second order convergence rate. However, this may not occur until the number of nodes is large, depending on the difficulty of the numerical approximation of the underlying problem. For the easy case $D = 10^{-2}$, both fixed and moving meshes lead to a second or nearly second order convergence rate in the considered range (from 41 to 321) of the number of nodes. However, the situation is different for the more difficult case $D = 10^{-5}$. In this case, uniform meshes lead to a very slow convergence rate in this range of mesh nodes, whereas moving meshes maintain almost the same convergence rate as in the case $D = 10^{-2}$. (The second order convergence rate can be achieved with uniform meshes for large numbers of nodes, see Fig. 1b.)

Finally we investigate the efficiency improvement by the two-layer mesh movement. We list in Table 2 the results obtained using the two-layer mesh movement technique with a coarse mesh of 41 nodes. We note that in this case, the coarse (moving) mesh is fixed and only the mesh used for solving the physical PDE is refined while the convergence is considered. The coordinate transformation determined by the moving mesh can be thus regarded as almost fixed. As a consequence, the second order convergence can be expected to occur even for small numbers of nodes. Indeed, the results in Table 2 show clearly the second order convergence. It should be emphasized that unlike the case with uniform meshes, the convergence rate does not deteriorate with D. It can also be seen from Tables 1 and 2 that both the efficiency and convergence rate of the moving mesh method is improved significantly with the two-layer mesh movement strategy. For instance, the two-layer mesh movement with $J^c = 41$ and $J = 161$ leads to a result for the case $D = 10^{-5}$ as accurate as that obtained using the one grid movement with $J = 161$, but the former requires only about 75% of the CPU time used by the latter. Moreover, the percent of the CPU time used for moving the mesh is reduced from 50% for the one-grid case to only about 13% for the two-layer mesh movement.

To conclude this section we summarize briefly the observations we made through the simple advection dispersion equation. The key to the success of the moving mesh method is to design an appropriate monitor function. The monitor function based on the interpolation error indicator is more robust and leads to more accurate results than that based on the solution gradient. This is consistent with the observations made in [3, 4] for two dimensional problems. Mesh concentration can be controlled by introducing the intensity parameter in the monitor function. The technique presented in section 2.3 seems very effective. The two-layer mesh movement strategy not only improves significantly the efficiency of the moving mesh method but also helps recover the exact second order convergence rate on moving meshes.

Table 1: Results with fixed and moving meshes for model problem (1).

D	Fixed/Moving	Mesh	$\|\|\|e\|\|\|$ (ratio)	Total CPU
1e-2	Fixed	41	8.75e-3	1.2
		81	3.24e-3 (2.6)	2.1
		161	9.50e-4 (3.4)	4.5
		321	2.36e-4 (4.0)	10
	Moving	41	1.13e-3	4.6
		81	2.64e-4 (4.3)	7.8
		161	7.92e-5 (3.0)	19
		321	2.34e-5 (3.1)	52
1e-5	Fixed	41	1.22e-1	2.3
		81	9.88e-2 (1.2)	5.9
		161	7.84e-2 (1.3)	20.1
		321	5.96e-2 (1.3)	65.9
	Moving	41	1.77e-3	43.1
		81	4.72e-4 (3.8)	79.3
		161	1.67e-4 (2.8)	217.6
		321	5.88e-5 (2.8)	465.4

4 Applications

In this section, we apply the moving mesh approach and the implementation strategies to the simulation of a selection of groundwater problems, including chemical transport and reaction, water infiltration into soil, and the coupling of density dependent flow and transport. All of these examples involve sharp moving fronts. We use the monitor function based on the interpolation error indicator and take the concentration parameter as $\beta = 0.5$. All of the presented results are obtained with 81 moving nodes. For Examples 2 and 3, there are no exact solutions available. For comparison purpose, a reference solution is also presented which is obtained using the two-layer moving mesh method with $J = 321$ (the number of the physical mesh points) and $J^c = 41$ (the number of the moving nodes).

4.1 Advection dominated chemical transport and reaction

In subsurface, mass in the form as ions, molecules, or solid particles undergoes multiple processes, including transport processes such as advection and dispersion and chemical, nuclear, and biological processes. Acid-base reactions, solution, volatilization, and precipitation, , solute reactions, oxidation-reduction reactions, hydrolysis reaction, isotropic

Table 2: Results with the two-level mesh movement for model problem (1). The moving mesh has 41 nodes or $J^c = 41$. See Table 1 for more results obtained with $JM = 1$.

D	JM	Physical Mesh	$\|\|\|e\|\|\|$ (ratio)	Total CPU	(mesh CPU)/(total CPU)
1e-2	1	41	1.13e-3	4.6	0.50
	2	81	2.57e-4 (4.4)	7.4	0.31
	4	161	6.78e-5 (3.8)	14.2	0.16
	8	321	1.90e-5 (3.6)	23.3	0.10
1e-5	1	41	1.77e-3	43.1	0.50
	2	81	4.21e-4 (4.2)	109.4	0.20
	4	161	1.14e-4 (3.7)	161.7	0.13
	8	321	3.04e-5 (3.7)	306.3	0.07

reactions, adsorption and desorption are examples of chemical, nuclear, and biological processes. An accurate description of these behaviors is prerequisite to the successful groundwater protection and management. One of the mathematical models of these behaviors can be written as the so called advection-dispersion-reaction (ADR) equation

$$R\frac{\partial C}{\partial t} = \frac{\partial}{\partial x}\left(D\frac{\partial C}{\partial x}\right) - V\frac{\partial C}{\partial x} - \lambda RC, \tag{16}$$

where C is the concentration of a certain type of chemical, D is the dispersivity, V is the Darcy velocity, R is the retardation factor, and λ is the reaction factor. We take the physical parameters as $D = 10^{-5}$, $V = 1$, $R = 1.1$, and $\lambda = 1.1$. The computation is performed on interval $(0, 1)$ with the initial and boundary conditions

$$C(x, 0) = 0, \quad C(0, t) = 1, \quad C(1, t) = C^*, \tag{17}$$

where C^* is taken as the value of the exact solution at $x = 1$. The exact solution to this problem is known [17]

$$C(x,t) = \frac{1}{2}\exp\left(\frac{Vx}{2D}\right) \quad \left[\exp\left(-\frac{\sqrt{V^2+4D\lambda R}x}{2D}\right)\operatorname{erfc}\left(\frac{x-\sqrt{V^2+4D\lambda R}t/R}{\sqrt{4Dt/R}}\right)\right.$$
$$\left. + \exp\left(\frac{\sqrt{V^2+4D\lambda R}x}{2D}\right)\operatorname{erfc}\left(\frac{x+\sqrt{V^2+4D\lambda R}t/R}{\sqrt{4Dt/R}}\right)\right].$$

Fig. 6 shows the numerical solution at various time instants obtained with the moving mesh method. Also given is the the analytical solution for comparison. The figure demonstrates that the MMPDE method provides accurate tracking of the sharp moving front.

4.2 Nonlinear infiltration in unsaturated soil

The movement of water through unsaturated soil or vadose zone infiltration is of interest with hazardous and radioactive waste transport and storage in environmental protection and rainfall run-off problems in watershed management. The movement of water moisture in soil is another common problem in groundwater hydrology. The modeling of infiltration is complicated by the variation in both water pressure and saturation, and the presence of a nonlinear relationship between saturation and physical parameters such as hydraulic conductivity. The most common mathematical description of moisture movement during infiltration is the Richard equation [6]

$$\frac{\partial \theta}{\partial t} = \frac{\partial}{\partial z}\left(D\left(\theta\right)\frac{\partial \theta}{\partial z}\right) - \frac{\partial K\left(\theta\right)}{\partial z}, \tag{18}$$

where z is the vertical coordinate, θ is the water content, $D(\theta)$ is the moisture diffusivity defined as a function of θ, and $K(\theta)$ is the unsaturated hydraulic conductivity, also depending on θ.

We consider the case where $z \in (0,1)$ and $t \in (0,1)$ with $D(\theta) = 2\theta$, and $K(\theta) = \theta^3$ as used in [6]. The initial and boundary conditions are chosen as

$$\theta(x,0) = 0, \quad \theta(0,t) = 1, \quad \frac{\partial \theta}{\partial z}(1,t) = 0. \tag{19}$$

Fig. 7 shows the computed saturation and water flux at several time instants. No analytic solution is available for this problem. The water flux is calculated according to the formula

$$q = -K\nabla H = -K\left(1 + \frac{dh}{dz}\right) = -K\left(1 + \frac{K}{D}\frac{d\theta}{dz}\right), \tag{20}$$

where h is the water head.

To demonstrate the ability of the moving mesh method to capture sharp fronts, we reduce the diffusivity to $D = 2\times 10^{-4}\theta$ and increase K by a factor of 2.5. This corresponds to the situation where water moves in unsaturated clay or highly impermeable soil. The inflow boundary condition at ground surface is also modified to a time dependent condition $\theta(0,t) = e^{-t}$. Fig. 8 presents the computed saturation at different time instants and the location of the sharp moving front. One may notice that the movement of the front is not uniform with time because of non-linearity of the problem.

For comparison, a reference solution is also plotted in Figs. 7 and 8. These results, particularly the location of the moving front, proves that the moving front is captured very well by the moving mesh method with a relatively small number of nodes.

4.3 Coupling of density dependent flow and transport

The last example is a coupled problem for flow and brine transport in porous media. This problem has been analyzed by Zegeling et al. [20]. It deals with the interaction between

the flow and brine concentration, where the salt concentration is high and influences the fluid density. On the other hand, the fluid density has effect on the flow, and then the brine movement or transport. This isothermal, single phase, two component saturated flow model in one-dimensional space is given [20] as

$$\begin{aligned} \omega\rho\beta\frac{\partial P}{\partial t} + \omega\rho\gamma\frac{\partial C}{\partial t} &= -\frac{\partial(\rho v)}{\partial x}, \\ \omega\rho\frac{\partial C}{\partial t} &= -\rho v\frac{\partial C}{\partial x} + \frac{\partial}{\partial x}\left(\rho\lambda|v|\frac{\partial C}{\partial x}\right), \end{aligned} \tag{21}$$

where the velocity is related to pressure through $v = -\frac{k}{\mu}(\frac{\partial P}{\partial x} + \rho g)$ and the equation of state is $\rho = \rho_0 \exp(\beta(P - P_0) + \gamma C)$. Here, ρ_0 is the constant reference density, P_0 is the constant reference pressure, β is the constant compressibility coefficient, γ is the constant salt coefficient, k is the permeability, g the gravity, μ the viscosity, and λ the dispersion length.

Using the same values of the parameters (listed in Table 1 of [20]) and the same initial and boundary conditions we obtain similar results to those in [20], and for this reason we do not present them here. Instead, we consider here a more interesting case. We note that the initial and boundary conditions used for the first example of [20] are

$$P(x,0) = 1.7(1-x) + x, \quad C(x,0) = 0$$

and

$$P(0,t) = 1.7, \quad P(1,t) = 1.0, \quad C(0,t) = 1, \quad \frac{\partial C}{\partial x}(1,t) = 0.$$

Consider a case where the left boundary conditions are made to be time dependent, i.e.,

$$P(0,t) = 1.7e^{-0.3t}, \quad C(0,t) = e^{-0.3t}.$$

The computed pressure and concentration, as well as the reference solution, at $t = 0$, 0.1, 0.2, 0.3, 0.4, and 0.5 are shown in Fig. 9. Once again the results demonstrate the capability of the moving mesh method to capture sharp fronts for coupled problems.

5 Conclusions

A moving mesh method based on the so-called moving mesh PDE, introduced in [14], has been presented. Our emphasis is on the important implementation strategies, including the construction of the monitor function based on the interpolation error indicator, control of mesh concentration, and two-layer mesh movement. An advection-dispersion equation, a typical model problem in groundwater simulation, is used to demonstrate the effectiveness of the implementation strategies. Of particular interest, the strategy of controlling the mesh concentration is effective, and the two-layer mesh movement strategy improves significantly the efficiency of the moving mesh method.

Numerical results are presented for three application examples in the field of groundwater. The first deals with advection dominated contaminant transport, the second example is a nonlinear infiltration problem, and the last example is a coupled density dependent flow and transport problem. All three examples involve sharp moving fronts, and all of them contain high degree of difficulty for the numerical solution. The excellent results we obtained clearly demonstrate the capability of the MMPDE method and indicate the high potential of this method in handling difficult groundwater modeling problems.

Although only one dimensional problems are addressed here, our ultimate goal is to apply this approach to realistic two- and three-dimensional groundwater systems. Our future research includes the detailed comparison of the moving mesh approach with other adaptive mesh methods such as refinement and the application to more complex groundwater systems with multiple species, reactions, and moving chemical fronts.

Acknowledgment. This work was supported in part by the NSF under grant DMS-0074240 and the Kansas Geological Survey.

References

[1] G. Beckett and J. A. Mackenzie, On a uniformly accurate finite difference approximation of a singularly perturbed reaction-diffusion problem using grid equidistribution, *J. Comput. Appl. Math.* (to appear).

[2] J. G. Blom and J. G. Verwer, On the use of the arc-length and curvature monitor functions in a moving-grid method which is based on the method of lines, CWI (Stichting Mathematisch Centrum, Centrum voor Wiskunde en Informatica, Amsterdam) Report. NM-N8902, 1989.

[3] W. Cao, W. Huang, and R. D. Russell, A two-dimensional r-adaptive finite element method based on a posteriori error estimates, *J. Comput. Phys.* (to appear).

[4] W. Cao, W. Huang, and R. D. Russell, Comparison of two-dimensional r-adaptive finite element methods using various error indicators, *Math. Comput. Simulation* (to appear).

[5] J. R. Cash, Diagonally implicit Runge-Kutta formulae with error estimates, *J. Inst. Math. Appl.* 24 (1979) 293 – 301.

[6] C. L. Cox and T. H. Payne, Mathematical modeling of unsaturated porous media flow and transport, in *Applied mathematical modeling, A multidisplinary approach* Edited by D. R. Shier and K. T. Wallenius (Chapman and Hall/CRC, 2000), 185 – 201.

[7] C. de Boor, Good approximation by splines with variables knots II, in Springer Lecture Notes Series 363 (Springer-Verlag, Berlin, 1973).

[8] E. A. Dorfi and L. O'c Drury, Simple adaptive grids for 1-D initial value problems, *J. Comput. Phys.* 69 (1987), 175 – 195.

[9] B. H. Fiedler and R. J. Trapp, A fast dynamic grid adaption scheme for meteorological flows, *Mon. Weather Rev.* 121 (1993), 2879 – 2888.

[10] A. Gamliel and L. M. Abriola, A One-dimensional moving grid solution for the coupled nonlinear equations governing multi-phase flow in porous media. 1: Model development, *Int. J. Numer. Meth. Fluids* 14 (1992), 25 – 45.

[11] A. Gamliel and L. M. Abriola, A One-dimensional moving grid solution for the coupled nonlinear equations governing multi-phase flow in porous media. 2: Example simulations and sensitivity analysis, *Int. J. Numer. Meth. in Fluids* 14 (1992), 47 – 69.

[12] G. Gottardi and M. Venutelli, Moving finite element model for one-dimensional infiltration in unsaturated soil, *Water Resour. Res.* 28 (1992), 3259 – 3267.

[13] W. Huang, Practical aspects of formulation and solution of moving mesh partial differential equations, Mathematics Research Report 99-11-02, Department of Mathematics, The university of Kansas, Lawrence, KS 66049. (Submitted for publication)

[14] W. Huang, Y. Ren, and R. D. Russell, Moving mesh partial differential equations (MMPDEs) based upon the equidistribution principle, *SIAM J. Numer. Anal.* 31 (1994), 709 – 730.

[15] W. Huang, Y. Ren, and R. D. Russell, Moving mesh methods based on moving mesh partial differential equations, *J. Comput. Phys.* 113 (1994), 279 – 290.

[16] K. Miller and R. N. Miller, Moving finite elements I, *SIAM J. Numer. Anal.* 18 (1981), 1019 – 1032.

[17] T. Lee, Applied Mathematics in Hydrology (Lewis Publishers, 1999).

[18] R. Trompert, Local-uniform grid refinement and transport in heterogeneous porous media, *Adv. Water Resour.* 16 (1993), 293 – 304.

[19] G. T. Yeh, J. Chang, and T. E. Short, An exact peak capturing and oscillation-free scheme to solve advection-dispersion transport equations, *Water Resour. Res.* 28 (1992), 2937 – 2951.

[20] P. A. Zegeling, J. G. Verwer, and J. C. H. Van Eijkeren, Application of a moving grid method to a class of 1D brine transport problems in porous media, *Int. J. Numer. Meth. Fluids* 15 (1992), 175 – 191.

[21] C. Zheng and G. D. Bennett, Applied contaminant transport modeling: theory and practice, (Van Nostrand Reinhold, new York, 1995).

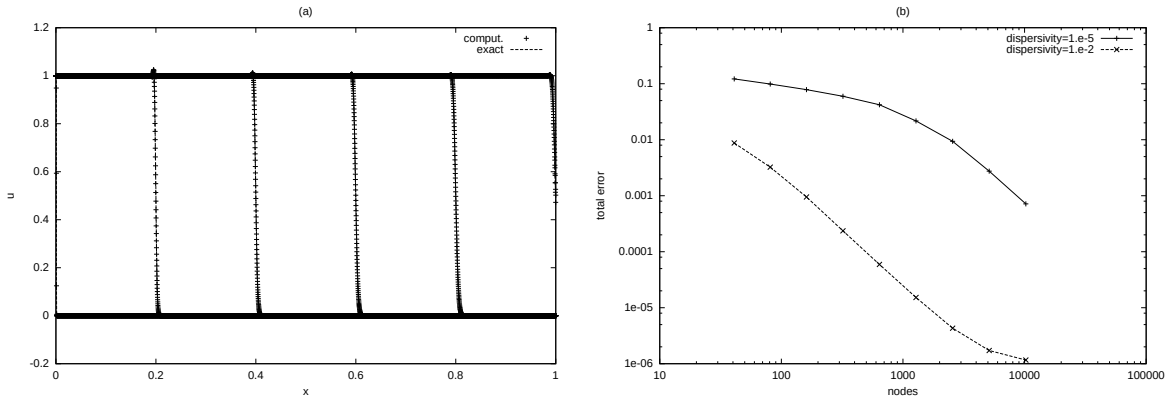

Figure 1: Numerical results obtained with uniform meshes for model problem (1). (a): The computed solution obtained with 5121 nodes at time $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0 for the case $D = 10^{-5}$. (b): Converegence history for the cases $D = 10^{-5}$ and $D = 10^{-2}$.

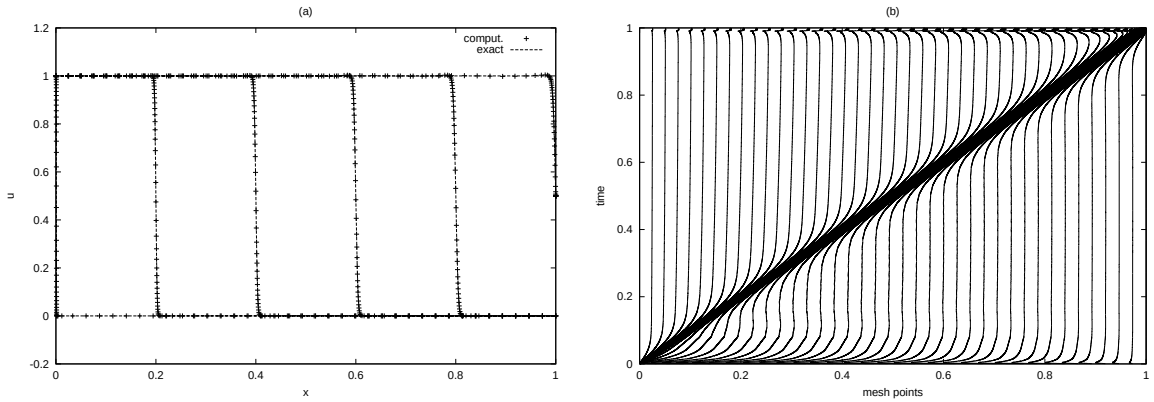

Figure 2: A typical result obtained using the moving mesh method with 81 nodes for model problem (1) with $D = 10^{-5}$. (a): The computed solution at time $t = 10^{-4}$, 0.2, 0.4, 0.6, 0.8, and 1.0. (b): The mesh trajectories.

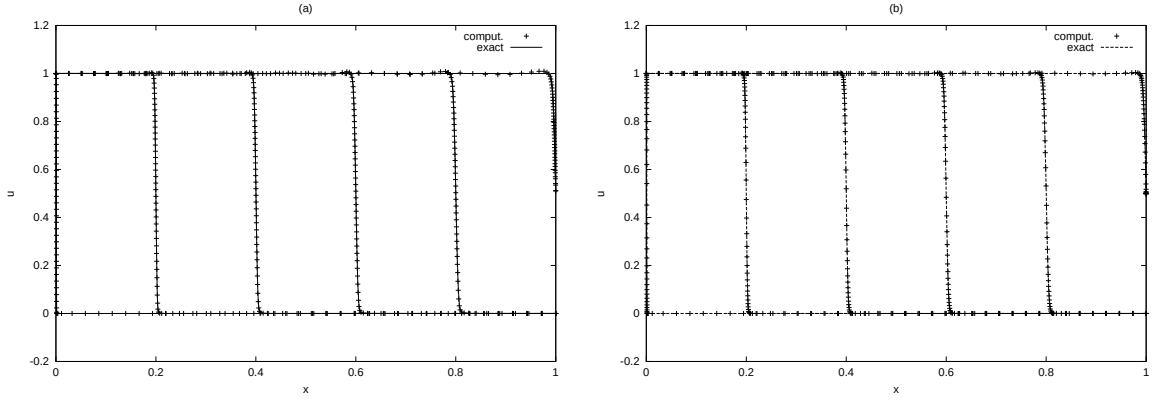

Figure 3: Computed solutions obtained using 81 moving nodes and the monitor function based on either (a) the solution gradient or (b) the interpolation error indicator for model problem (1) with $D = 10^{-5}$.

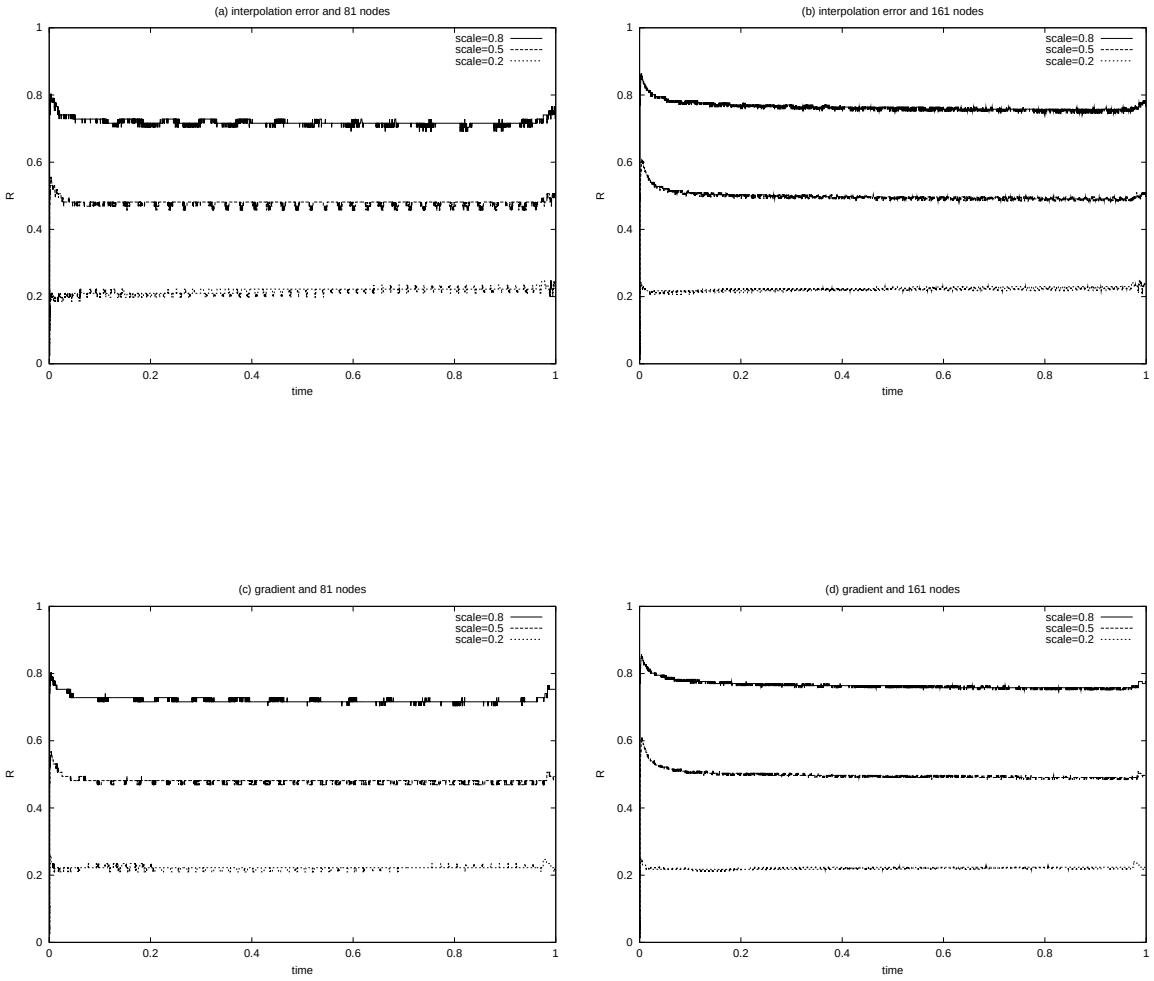

Figure 4: R, the ratio of mesh points at which the value of the monitor function is strictly greater than its average to the number of mesh points is shown as function of time for various values of β, two numbers of mesh points, and the monitor functions based on the error indicator (a,b) and the solution gradient (c,d).

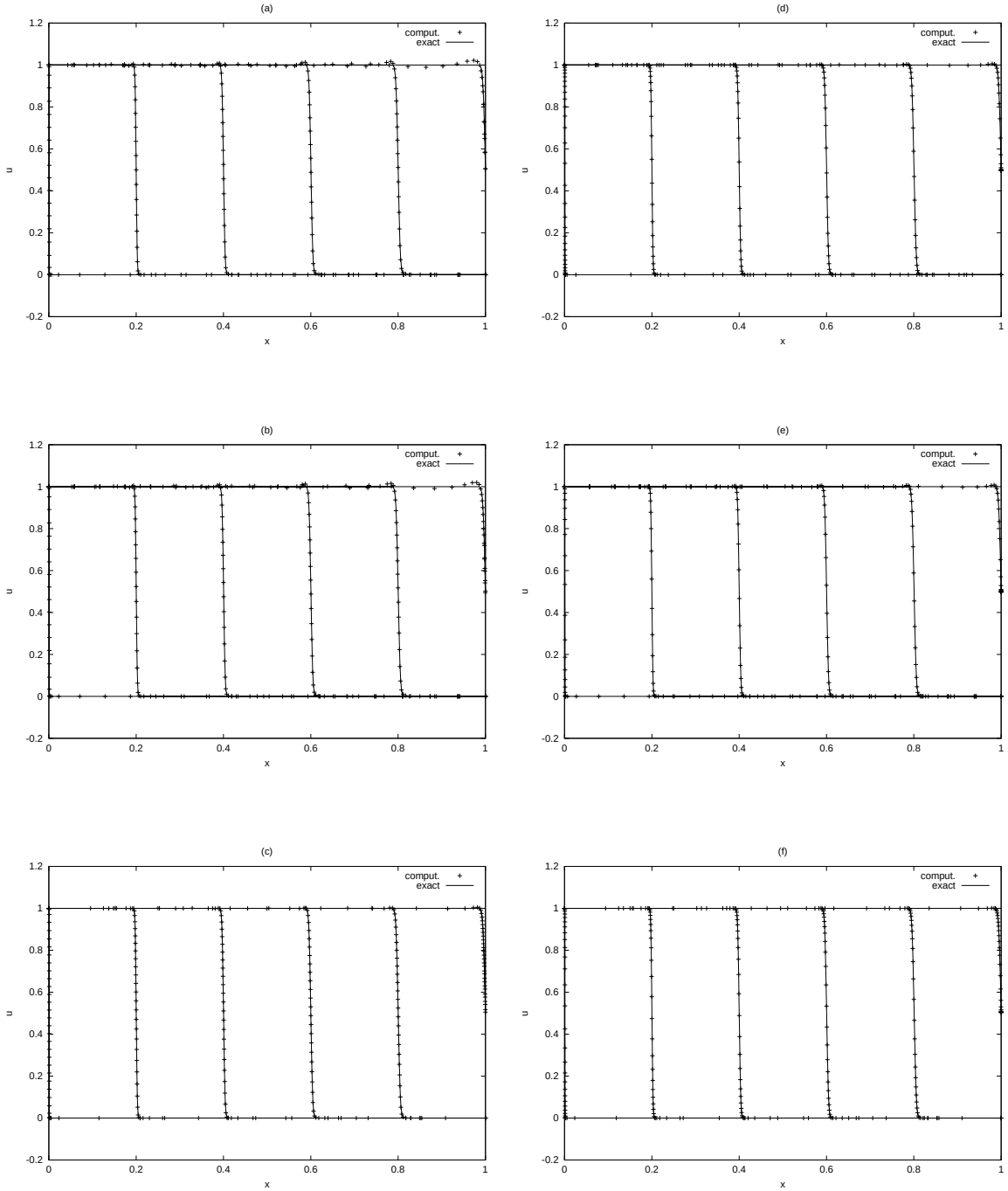

Figure 5: Results obtained with a mesh of 41 nodes for model problem (1). (a): Arc-length g without control, $|||e||| = 4.48e-3$; (b): Arc-length g with $\beta = 0.5$, $|||e||| = 4.42e-3$; (c): Arc-length g with $\beta = 0.8$, $|||e||| = 9.71e-4$; (d): Error estimate g without control, $|||e||| = 8.49e-4$; (e): Error estimate g with $\beta = 0.5$, $|||e||| = 1.76e-3$; and (f): Error estimate g with $\beta = 0.8$, $|||e||| = 4.21e-4$.

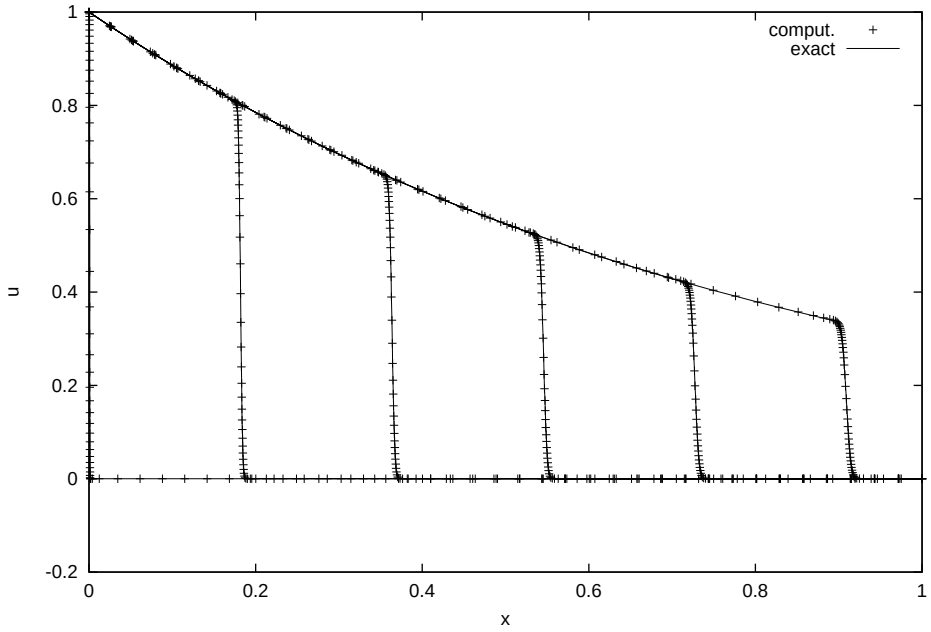

Figure 6: The computed and exact solutions at time $t = 0$, 0.2, 0.4, 0.6, 0.8, and 1.0 for the chemical reactive and retardation equation. The computed solution is obtained using the moving mesh method with 81 nodes.

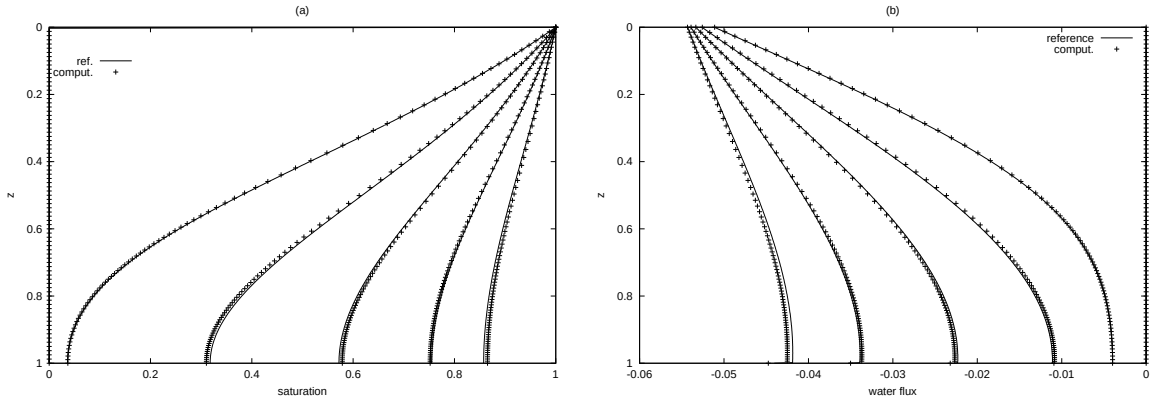

Figure 7: The computed saturation (a) and infiltrating water flux (b) at time $t = 0.0$, 0.2, 0.4, 0.6, 0.8, and 1.0 with 81 moving nodes for the nonlinear infiltration problem with $D = 2\theta$ and $K = \theta^3$.

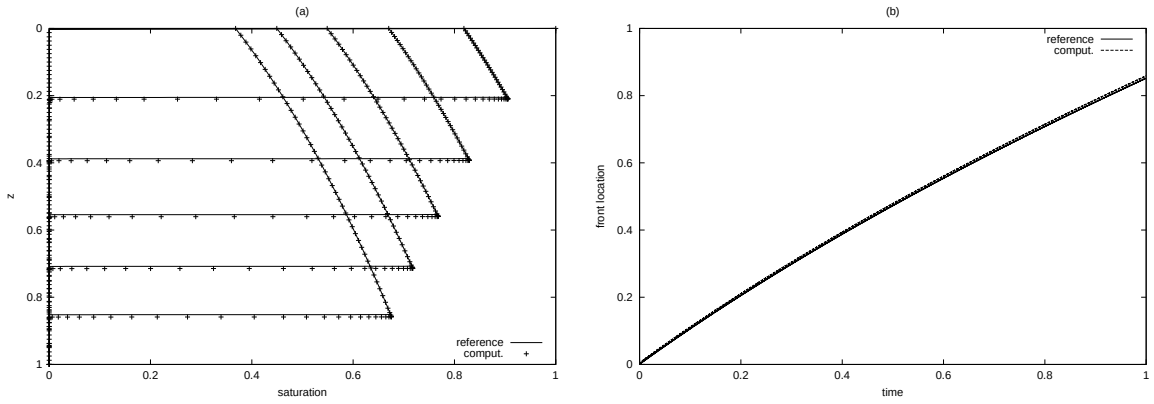

Figure 8: Numerical results for the nonlinear infiltration problem with sharp fronts. (a) Saturation at time $t = 0.0$, 0.2, 0.4, 0.6, 0.8, and 1.0. (b) The computed location of the moving front as function of time.

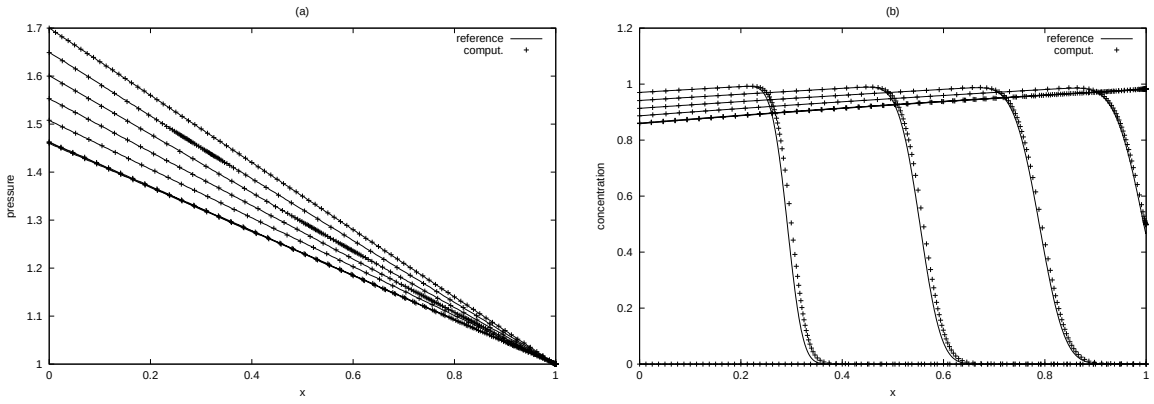

Figure 9: The computed pressure (a) and concentration (b) at $t = 0.0$, 0.1, 0.2, 0.3, 0.4, and 0.5 for the coupled fluid pressure and concentration problem.