Stabilization of the Evolutionary Convection-Diffusion Problem: Introduction and Experiments

Noel Heitmann
and
Stephen Peurifoy

Department of Mathematics
Millersville University of Pennsylvania
Millersville, PA 17551

March 24, 2007

Abstract

In [2] a variationally consistent discretization is developed for the time-dependent convection diffusion problem via an extension of work done in [3]. In this paper, we explain the fundamental difficulties with modeling the convection diffusion problem and present experimental results for a finite difference approximation of the methods used in [2, 3].

1 Introduction

The time-dependent convection diffusion problem seeks a scalar solution $u : \Omega \times (0, T] \to \mathbb{R}$ satisfying

$$\begin{cases} u_t - \epsilon \Delta u + \mathbf{v} \cdot \nabla u + gu = f & \mathbf{x} \in \Omega \\ u = 0 & \mathbf{x} \text{ on } \partial\Omega \end{cases} \tag{1}$$

In this problem u represents a scalar quantity (such as temperature or contaminant level) associated with a fluid flow. ϵ is a diffusion coefficient, $\mathbf{v}$ is the convection (or velocity) field, g is an absorbtion/reaction coefficient, and f is a forcing function.

Applications of the convection diffusion problem are virtually limitless. In nuclear engineering, convection diffusion is used to track a destructive substance called "crud" in a nuclear reactor. In modeling the extraction of oil, this problem is coupled with the porous media problem (which determines the convection field). Convection diffusion is also instrumental in modeling pollutants in water systems and in modeling global climate change. These are only a few of the many applications of this important problem. It stands to reason then, that determining a solution has great practical importance. Unfortunately, for any but the most trivial of parameter settings, this partial differential equation does not admit an analytic solution. As a result, we must turn to numerical approximations.

In this paper, we investigate broadly the process of producing a numerical model from start to finish. In section two we present the basic idea of how one can obtain a numerical solution for such a problem. In section 3 we highlight the major difficulty, namely instability of solutions in the spatial domain, that become manifest for a particular class of parameters and boundary conditions. Section 4 introduces a stabilization method that seeks to restore stability to the solution with minimal loss of solution quality. Section 5 presents a number of one and two spatial dimensional examples of the time-dependent problem and the resultant solutions. Finally, we give some conclusions in section 6.

2 Finite Difference Approximations for Differential Equations

As most real world differential equations are not solveable by analytic means, solutions must be obtained numerically. This process requires the translation of the continuous model into a discrete model. The typical finite difference approximation for the solution of a differential equation involves three steps: (1) discretization of the domain, (2) approximation of the derivatives, and (3) solving the resultant linear system of equations.

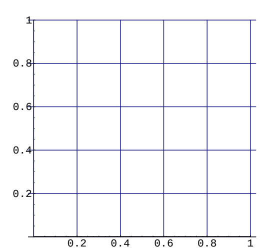

Figure 1: $m = n = 5$ uniform mesh

As an introduction, we will assume that our two dimensional spatial domain has been normalized to the unit square, $\Omega = [0, 1] \times [0, 1]$. One and three dimensional variants follow quite naturally (although the computational work increases exponentially with the three dimensional problem). The x and y intervals are partitioned into n and m intervals, $0 = x_0 < x_1 < ... < x_n = 1$ and $0 = y_0 < y_1 < ... < y_m = 1$. For ease of explanation, we will assume that $m = n$ and that the mesh is uniform with the mesh width defined by $h \equiv \Delta x = \Delta y = 1/n$. This is represented for the case $n = m = 5$ in figure 1. The intersections of grid lines are called mesh points. The goal is to approximate u on the $(n-1)^2$ interior mesh points, the boundary values being known.

The second step is to use difference quotients to approximate the derivatives of the differential equation. This can be done in a variety of ways. Let u be the true solution and U be the approximation. The indices i and j and superscripts n are used such that $U^n_{i,j} = U(i\Delta x, j\Delta x, n\Delta t) = U(ih, jh, n\Delta t)$. In this paper, we use the following approximations for the derivatives in equation (1),

$$\begin{aligned} (u_t)^{n+1}_{i,j} &\approx \tfrac{U^{n+1}_{i,j} - U^n_{i,j}}{\Delta t} \\ \Delta u^{n+1}_{i,j} = (u_{xx} + u_{yy})^{n+1}_{i,j} &\approx \tfrac{U^{n+1}_{i,j-1} + U^{n+1}_{i-1,j} - 4U^{n+1}_{i,j} + U^{n+1}_{i+1,j} + U^{n+1}_{i,j+1}}{h^2} \\ \nabla u^{n+1}_{i,j} &\approx \left\langle \tfrac{U^{n+1}_{i+1,j} - U^{n+1}_{i-1,j}}{2h}, \tfrac{U^{n+1}_{i,j+1} - U^{n+1}_{i,j-1}}{2h} \right\rangle^T \end{aligned} \tag{2}$$

For interior points adjacent to one or more boundaries, the approximate values at boundary mesh points in the above approximations are replaced with the known values of u.

Finally, for each interior mesh point, the approximations are inserted into the differential equation (1). This creates $(n-1)^2$ linear equations in $(n-1)^2$ unknowns, or a system

$$A\mathbf{U}^{n+1} = \mathbf{b}$$

which can be solved for the unknown vector $\mathbf{U}^{n+1}$ of interior mesh point approximations $U^{n+1}_{i,j}$. If desired, a continuous approximation can then be constructed via interpolation over the discrete approximations.

3 Fundamental Difficulties

The methods outlined above work very well for a number of problems. However, frequently in applications the inequality

$$\frac{\epsilon}{|\mathbf{v}|h} \ll 1 \tag{3}$$

holds. Problems of this type are called convection dominated problems. It should be noted, that often in practice, problems are normalized such that $|\mathbf{v}| = 1$, thus the condition is reduced to the ratio of ϵ and h. True solutions of such problems often exhibit interior layers or boundary layers (a term coined by Ludwig Prandtl in 1904). In this case numerical solutions become unstable, manifested by wild node to node fluctuations as shown in figure 2.

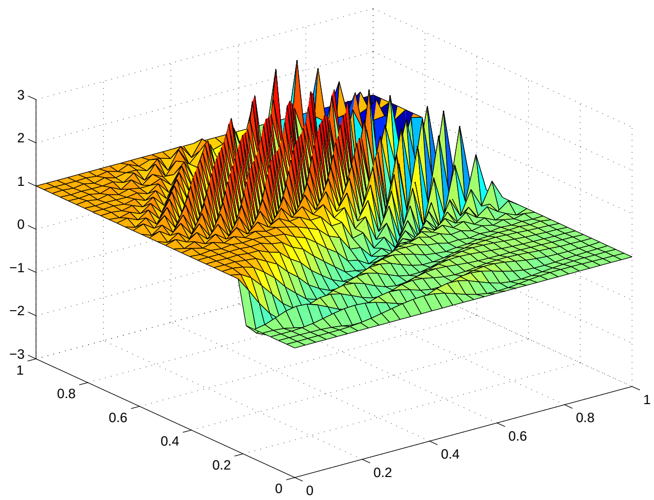

Figure 2: Numerical instability with an interior layer

Refering back to equation (3) with $|\mathbf{v}| = 1$, this instability can be managed either by increasing ϵ or decreasing h. The former solution involves changing the fundamental nature of the physical reality that we are attempting to model. For this reason, it might be considered a poor choice, but despite this flaw, it is a technique that is used with some regularity. At first glance, the latter solution, decreasing h, would seem to be the obvious solution. Unfortunately, the cost, in terms of needed computational resources such as the availability of memory and the time needed to compute a solution, makes this solution impossible (or at least impractical) in many cases. For this reason, a great deal of mathematical research has been devoted towards numerical methods that attempt to satisfy three criterion:

1. accuracy, as measured by the order of h
2. spatial stability, and
3. cost effectiveness.

4 Artificial diffusion on small scales

As mentioned in the previous section, one solution for obtaining stability is simply to blanket the domain of the problem with added diffusion by increasing ϵ. This type of solution changes equation (1) into

$$u_t - (\epsilon + \alpha)\Delta u + \mathbf{v} \cdot \nabla u + gu = f$$

where the added diffusion is represented by the parameter α. This type of solution has the effect of smoothing out sharp boundary and/or interior layers resulting in stability. However, this approach also adds diffusion to areas of the domain where stability is not an issue.

The goal of the methods presented in [2, 3] is to add diffusion only where it is needed to attain stability. Consider a decomposition of the true solution u into its large and small scales. The large scales represent the overall structure of the solution where the small scales represent fluctuations from this structure. Another way to think of this is that the large scales are represented by average values of the solution over some radius and the small scales represent fluctuations from that average. In figure 4, the "jagged sine" curve represents the true solution. The "smooth sine" function represents the average values, or large scale structure of the solution. The remaining curve represents fluctuations from average or the small scales of the solution.

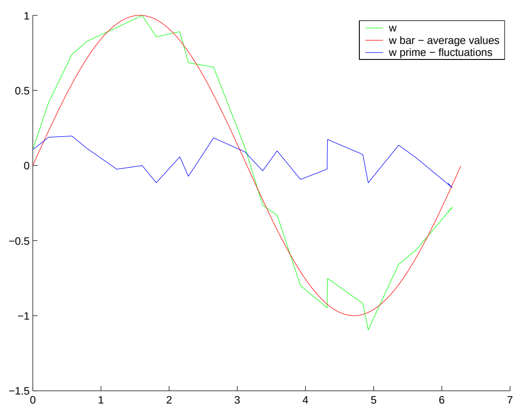

Figure 3: A decomposition of u into $\overline{u}$ and fluctuations

Indeed, let $\overline{u}$ denote average values of the solution. Then $u' = u - \overline{u}$ denotes fluctuations from the average solution. Thus, artificial diffusion can be added only to the small scales of the problem by amending equation (1) to

$$u_t - \epsilon \triangle u - \alpha \triangle (u - \overline{u}) + \mathbf{v} \cdot \nabla u + gu = f.$$

Rearranging terms gives

$$u_t - (\epsilon + \alpha) \triangle u + \mathbf{v} \cdot \nabla u + gu = f - \alpha \triangle \overline{u}.$$

At this point we can apply our numerical method using the chosen discrete approximations from (2) for the derivatives (denoted by the superscript h) giving

$$\frac{\partial}{\partial t}^{h} U^{n+1} - (\epsilon + \alpha)\Delta^h U^{n+1} + \mathbf{v} \cdot \nabla^h U^{n+1} + gU^{n+1} = f^{n+1} - \alpha \Delta^h \overline{U^n}. \tag{4}$$

Note that the true solution u has been replaced with its approximate values U in the above equation. Also note that the method is *mostly* implicit with the exception of the averaging operator on the right hand side. This is done to allow changes to the averaging operator without changing the entire matrix of coefficients.

5 Experiments

In order to evaluate the effectiveness of this stabilization technique, MATLAB code was written to implement the technique for several different types of problems. These included:

1. A one (spatial) dimensional evolutionary problem with constant boundary conditions at inflow and outflow which achieves steady state.
2. A one dimensional evolutionary problem modeling a "semi-infinite" medium in which the propagation of a time-varying boundary condition "pulse" is observed.
3. A two dimensional evolutionary problem with a non-constant (half sine) boundary condition along an inflow edge and fixed boundary conditions $u = 0$ at all other edges which reaches steady state.

In all cases studied thus far, a constant convection field with $|v| = 1$ has been prescribed. We note that examples 1 and 3, which reach steady state, can also be solved directly as stationary problems (i.e. with $u_t = 0$). However, we pose them as evolutionary problems to test our time-dependent code.

The underlying finite difference scheme employed in these experiments was a single step backward difference method in which the spatial derivatives are approximated using mesh values of the dependent variable at the current time step. This is an implicit method which entails solving a simultaneous system of equations involving the spatial mesh values of U at each time step. In order to simplify the implementation, the averaging function used to compute $\overline{U}$ operated on values from the previous time step as noted in section 4. A weighted average of U at each spatial point along with its nearest neighbors was used. In the one and two dimensional cases respectively,

$$\begin{aligned} \overline{U_i} &= \beta\, U_i + (1 - 2\beta)\,(U_{i-1} + U_{i+1}) \\ \overline{U_{i,j}} &= \beta\, U_{i,j} + (1 - 4\beta)\,(U_{i,j-1} + U_{i-1,j} + U_{i+1,j} + U_{i,j+1}) \end{aligned} \tag{5}$$

where β is a parameter controlling the relative weighting of the central point versus its neighbors. For the 1-d problems we used $\beta = 0.3$ and for the 2-d problem we used $\beta = .125$. The experimental parameters to be varied for a given problem are thus the amount of artificial diffusion added, α, the diffusion coefficient ϵ, and the averaging coefficient, β.

5.1 Example 1: Temperature of fluid through a thin pipe

Assume that a fluid flows through a thin pipe. The (normalized) temperature at the inflow ($x = 0$) is held constant at $u = 0$ and at the outflow ($x = 1$), the (normalized) temperature is $u = 1$. The convection moves the colder fluid through the pipe and the diffusion of heat makes the heat of the warmer fluid at the outflow transfer backwards. This scenario is modeled by

$$\begin{cases} u_t - .0001\Delta u + u_x &= 0 \\ u(x,0) &= 0, \quad 0 \leq x < 1 \\ u(0,t) &= 0, \quad t \geq 0 \\ u(1,t) &= 1, \quad t \geq 0. \end{cases}$$

The temperature profile then will show a solution that begins at zero and climbs to 1. If the diffusion coefficient ϵ is relatively large, the climb will be gradual. However, in the interesting case, when ϵ is small, the climb will be steep and abrupt at the outflow, creating a boundary layer.

One advantage to this problem is that the true solution is known to be

$$u_\epsilon(x) = \frac{1 - e^{x/\epsilon}}{1 - e^{v/\epsilon}}.$$

For our experiment, we chose $v = 1$ and $\epsilon = .001$. For the discretization, we used $h = .01$. Thus, equation (3) gives $\epsilon/(|v|h) = 0.1$. As suggested in section 3, instability occurs if no stabilization is implemented. This is seen in figure 4, where the first picture shows the complete numerical solution and the second picture zooms in on the outflow behavior.

When the stabilization scheme of equation (4) is implemented, with $\alpha = 0.3$, the results are as shown in figure 5. Here, we can clearly see that the spatial instability has been removed to a large extent and the sharpness of the boundary layer has been maintained. However, it is equally clear that the solution is not perfect. We see what is called an undershoot at the boundary where solution values take on physically impossible values less than zero. Nonetheless, the stabilization scheme has produced a significantly better qualitative solution.

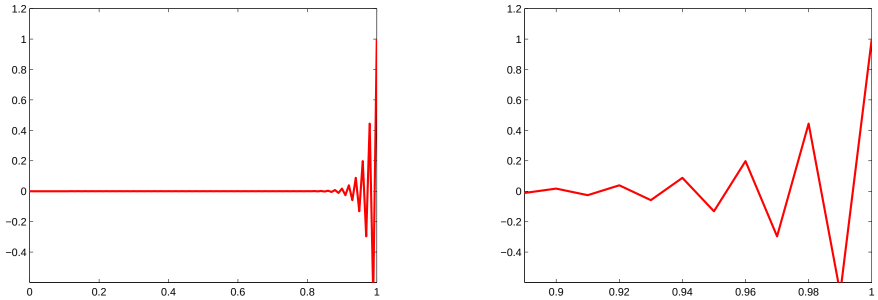

Figure 4: Example 1 with no stabilization

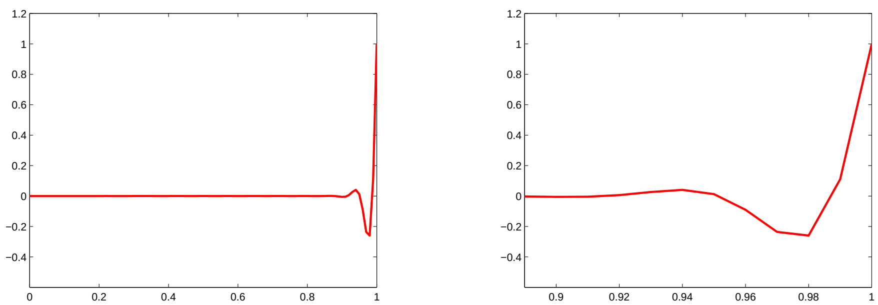

Figure 5: Example 1 with stabilization

5.2 Example 2: Traveling Pulse

Our second example is modeled by the evolutionary problem

$$\begin{cases} u_t - .0001\Delta u + u_x &= 0 \\ u(x,0) &= 0, \qquad x \geq 0 \\ u(0,t) &= g(t), \quad t \geq 0 \end{cases}$$

where the function $g(t)$ creates a a gaussian pulse with a width on the order of one tenth the overall time scale that then travels along with the velocity field. Again, we see a drastic difference between the solution when no artificial diffusion is used (figure 6) and when the stabilization method is used with $\alpha = 0.3$ (figure 7). The same type of behavior in the stabilized solution is manifested with the undershoot coming only at the tail end of the pulse.

One of the problems with any artificial diffusion stabilization method is that by its nature it adds diffusion to the problem, thus tamping down structures to some extent. In the case

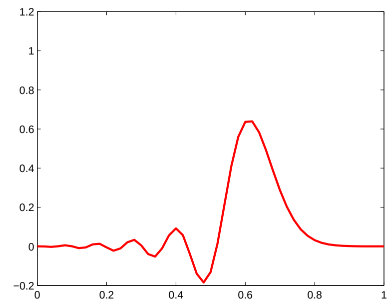

Figure 6: Example 2 with stabilization

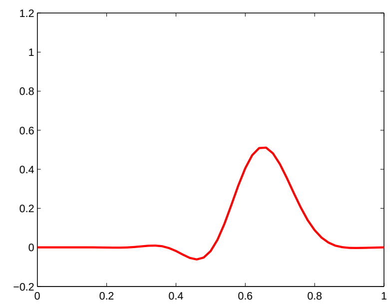

Figure 7: Example 2 with no stabilization

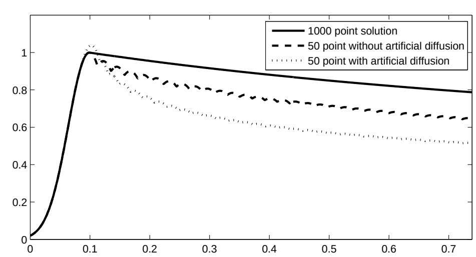

Figure 8: Envelopes of the true, unstabilized, and stabilized solutions

of the traveling pulse problem this is manifest by the height of the pulse becoming smaller as it travels. In figure 8 we see the height of the pulse for an approximation of the true solution (obtained by approximating with a very fine mesh), the solution with no artificial diffusion added, and the solution obtained via the stabilization method employed. We can see clearly that both approximate solutions underestimate the height of the structure and that the stabilized structure is the bigger culprit. This is the cost associated with controlling the instability that occurs at the tail end of the structure when no stabilization is used.

5.3 Example 3: 2-D Half-Sine Inflow

For our final example we consider a problem with two spatial dimensions. In this problem, all initial values are set to $u = 0$ except at the inflow boundary. The problem is given by

$$\begin{cases} u_t - .0001\Delta u + <1,0> \cdot \nabla u & = & 0 & \\ u(x,y,0) & = & 0, & 0 \le x < 1,\ 0 \le y \le 1 \\ u(x,0,t) = u(x,1,t) = u(1,y,t) & = & 0, & t \ge 0 \\ u(0,y,t) & = & \sin(\pi y), & t \ge 0 \end{cases}$$

With small ϵ as time marches on, the half sine inflow profile will move across the domain losing only a small amount of its height until it gets close to the outflow, at which point it drops quickly to the outflow boundary condition of zero, creating the boundary layer. Two views each are shown for the solution with no stabilization and with stabilization. The results are similar to the other two examples. When no stabilization is used the solution is clearly a qualitative disaster as shown in figure 9. When the stabilization method is used, stability is achieved as shown in figure 10. However, as usual, this comes at the cost of the structure being pushed down more than it should be. We also see an overshoot around the boundary layer.

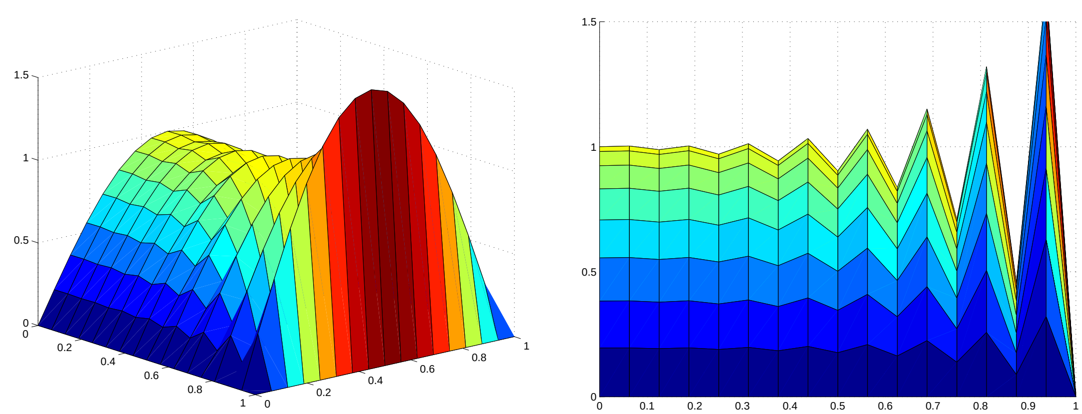

Figure 9: Two views of Example 3 with no stabilization

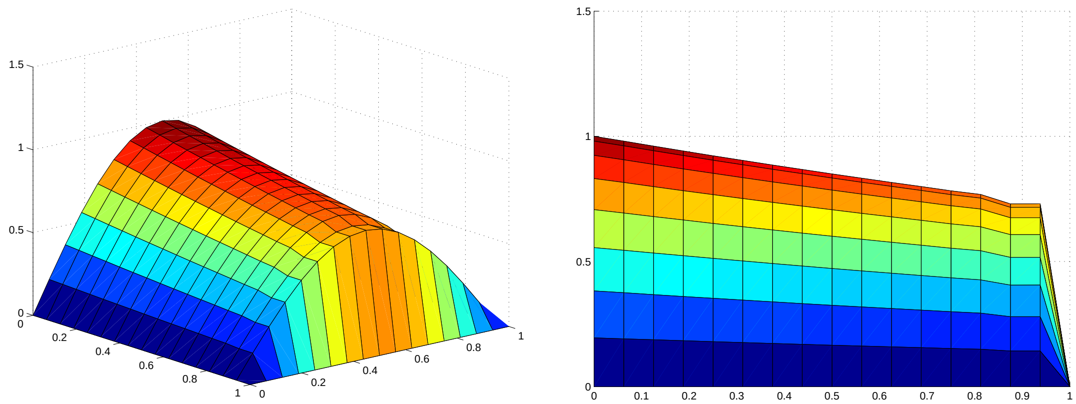

Figure 10: Two views of Example 3 with stabilization

6 Conclusions

The experiments carried out so far indicate that the introduction of artificial diffusion can be an effective technique for improving the stability of numerical solutions to the convection-diffusion equation. These stability improvements have not come without cost however. The goal of course, is to efficiently obtain a solution as close as possible to the "true" physical solution and although artificial diffusion as implemented here has clearly reduced the magnitude of oscillatory artifacts, some features of the desired solution are being affected as well (e.g. the increased attenuation of the traveling pulse in the one dimensional evolutionary experiment).

There are an essentially unlimited number of variations on a theme that can be employed in the averaging step of the artificial diffusion scheme. In order to make objective decisions regarding the merits of these algorithms, it is essential to have a quantitative measure of the error of a given numerical solution. Future work on artificial diffusion should therefore start with a comprehensive error analysis. It should be kept in mind that the significance of an error depends on the nature of the physical problem under study. It may be that sizeable errors in one part of the domain are of little concern while small errors in another are unacceptable. For problems without analytical solutions, the "gold standard" reference solution may have to be obtained through the use of a very fine mesh (albeit computationally expensive) numerical approximation.

Given a quantitative measure of the error in a numerical solution, a more systematic approach to the optimization of the algorithm's parameters can be carried out. A combination of theoretical analysis and experimentation is likely to be useful in this phase.

As previously noted, there are numerous degrees of freedom in the choice of averaging algorithm. There are similarities between the averaging problem here and problems that arise in other fields such as engineering. It may be that there is opportunity to borrow ideas from these other fields to good effect.

References

[1] R. Codina, Comparison of some finite element methods for solving the diffusion-convection-reaction equation, Comput. Methods Appl. Mech. Engrg. 156 (1998) 185-210

[2] N. Heitmann, Subgridscale stabilization of time-dependent convection dominated diffusive transport, J. Math. Anal. Appl (2006), doi:10.1016/j.jmaa.2006.08.049

[3] W.J. Layton, A Connection between subgrid scale eddy viscosity and mixed methods, *Appl. Math. and Comput.* **133** (2002) 147–157.