[10] A. J. M. van Gasteren, *Experimenting with a refinement calculus* in Beauty is our Business, W. H. J. Feijen et al. (ed.), Springer Verlag, 1990.

D. C. Ince,
Department of Computing,
The Open University,
Walton Hall,
Milton Keynes MK7 6AA,
England.

NUMERICAL SOLUTION OF CONVECTION-DIFFUSION PROBLEMS

Martin Stynes

Abstract: An overview is given of the nature of convection-diffusion problems and of some methods commonly used to solve these problems.

1. Introduction

Think of a still pond. At a point in this pond you pour a small amount of liquid dye. Approximately what shape will the dye stain take on the surface of the water as time passes? I think that we would all agree that the answer is a disc of slowly increasing radius, as the dye *diffuses* outwards from the initial point.

Consider next a more complicated situation: suppose that I replace the still pond above by a river which is flowing strongly and smoothly. What now is the shape of the dye stain?

The answer is a long thin curved wedge. This shape is the result of two physical processes: there is as before a tendency for the dye to diffuse slowly through the water, but the dominant mechanism present is the swift movement of the water, which rapidly sweeps (this is *convection*) the dye downstream. Convection alone would carry the dye along a (one-dimensional) curve on the surface; diffusion gradually spreads that curve, resulting in a wedge shape.

Physical situations such as this, where convection and diffusion are both present but convection dominates, are known

This article is the text of an invited lecture given by the author at the September meeting of the Irish Mathematical Society held at the Regional Technical College, Waterford, September 3–4, 1992.

as *convection-diffusion problems*. Convection-diffusion problems arise when modelling airflow over cars, aircraft wings and through jet engines, in weather forecasting, in the modelling of electrical currents in semiconductor devices and in many other applications. Consequently there is great interest in their analysis and numerical solution.

In this article we shall begin by discussing the nature of solutions to convection-diffusion problems. Then we move on to the construction of accurate numerical methods for the solution of these problems. Finally we outline the main tools which are used to analyse such numerical methods.

2. Structure of convection-diffusion solutions

The simplest mathematical model of a convection-diffusion problem is a two-point boundary value problem of the following form:

$$\varepsilon u''(x) + a(x)u'(x) + b(x)u(x) = f(x) \qquad \text{for} \quad 0 < x < 1, \quad (1)$$

with $u(0)$ and $u(1)$ given, where ε is a small positive parameter and a, b and f are some given functions. Here the term u'' corresponds to diffusion and its coefficient ε is small. The term u' represents convection, while u and f play the rôles of a source and driving term respectively. (For an explanation of why diffusion and convection should be modelled by second and first order derivatives respectively, see for example Spriet & Vansteenkiste [10].)

Problems of this type, where the highest order derivative has a small coefficient, are *singularly perturbed* differential equations. We begin by considering a single generic example in detail.

Example 1 Suppose that

$$-\varepsilon u''(x) + u'(x) = 1 \qquad \text{for} \quad 0 < x < 1, \qquad (2)$$

with $u(0) = u(1) = 0$ and $0 < \varepsilon << 1$. Here we have taken $-\varepsilon$ rather than ε as the coefficient of u'', since this turns out later to be more convenient; one can clearly move from either formulation to the other by multiplying the differential equation by -1.

The solution to this boundary value problem is easily seen to be

$$u(x) = x + \frac{e^{-1/\varepsilon} - e^{-(1-x)/\varepsilon}}{1 - e^{-1/\varepsilon}}.$$

It is more revealing if we write this as

$$u(x) = x - e^{-(1-x)/\varepsilon} + O(e^{-1/\varepsilon}). \qquad (3)$$

These three terms should be interpreted in the following way. The first, x, is the solution of the *initial* value problem

$$u'(x) = 1 \quad \text{on } (0,1) \quad \text{subject to } u(0) = 0. \qquad (4)$$

(This problem is obtained by formally setting ε equal to zero in (2) and taking one of the original boundary conditions.) The second term in (3) has a negligible influence on the solution when x is not near 1 (recall that ε is positive and small). It is essentially a correction to the solution of (4) which is required in order that the other boundary condition $u(1) = 0$ of the original problem be satisfied. The last term in (3) is of negligible size.

Thus from (3) we can see that a graph of $u = u(x)$ will closely approximate the straight line $u = x$ on almost all of [0,1]. When x approaches 1, the graph (while, of course, remaining continuous) suddenly departs from this straight line and plunges downwards to satisfy the condition $u(1) = 0$. We say that the graph has a *boundary layer* at $x = 1$.

This behaviour may be summarized as follows. Except on a narrow region near one of the boundaries, the solution of the original boundary value problem closely approximates the solution of an associated initial value problem.

Several further examples of this type are given in O'Riordan [8].

In two dimensions the situation is similar, as we now show.

Example 2 Consider the second order elliptic convection-diffusion problem

$$-\varepsilon \Delta u + u_x + u = f \quad \text{on } \Omega, \qquad (5)$$

with $u = 0$ on $\partial\Omega$, the boundary of Ω. Here to avoid any technical complications we assume that Ω is a bounded strictly convex region in $\mathbf{R}^2$ with smooth boundary $\partial\Omega$. We also assume that $f \in L^2(\Omega)$. As before, we take ε to be a small positive parameter. These hypotheses imply that (5) has a unique solution $u(x, y)$.

Write $\vec{\imath}$ for the unit vector in the direction of the positive x-axis and $\vec{n}$ for the outward pointing unit normal to $\partial\Omega$. Set

$$\partial^-\Omega = \{p \in \partial\Omega : \vec{\imath}.\vec{n} < 0 \text{ at } p\}$$

and

$$\partial^+\Omega = \{p \in \partial\Omega : \vec{\imath}.\vec{n} > 0 \text{ at } p\}.$$

Then, analogously to Example 1, the solution u on all of Ω, except close to $\partial^+\Omega$, is equal (modulo a little diffusion) to the solution v of the first order hyperbolic problem

$$v_x + v = f \quad \text{on } \Omega, \tag{6}$$

with initial data $v = 0$ on $\partial^-\Omega$. At $\partial^+\Omega$ the function u will have a boundary layer, i.e., close to $\partial^+\Omega$ the solution u changes rapidly in order to satisfy the boundary condition $u = 0$ on $\partial^+\Omega$.

This example in two dimensions is related to our earlier "dye in the river" problem. Think of the direction in which (6) propagates (the positive x-axis) as the direction of flow of the river. Then the first part of the previous paragraph states that the solution at any point (i.e., the presence or absence of dye at any point) *depends only on what happens almost directly upstream of that point*, which is what we observed when we stated that the dye spread as a long thin wedge.

This identification of the direction of propagation of a first order hyperbolic problem with the direction of flow of a fluid dynamics problem is often tacitly made in convection-diffusion terminology.

For further examples in two dimensions (and some graphs) see Johnson [1]. A feature which may occur in two dimensions is that a discontinuity in the boundary data on $\partial^-\Omega$ will in general cause

an *internal layer* in the solution; this is a narrow region, centred on one of the characteristics of the first order hyperbolic problem (6) - i.e, following the direction of flow - in which the solution changes rapidly. For a pictorial example of this see Johnson [1].

3. Numerical solution

In this Section we discuss some numerical methods which are commonly used to compute approximate solutions for convection-diffusion problems.

3.1 Why standard numerical methods fail

It is not immediately evident why convection-diffusion problems merit special attention from numerical analysts. After all, a problem such as Example 1 of the previous Section is a linear two-point boundary value problem. The average undergraduate numerical analysis textbook will give several methods applicable to this class of problems. (We shall refer to such standard textbook methods as *classical* methods, in order to distinguish them from methods which are designed specifically for convection-diffusion problems.) However if you try any classical method on Example 1, you will probably find that your computed solution displays wild oscillations and yields a very poor approximation of the true solution. What has gone wrong?

The answer may be found by a careful inspection of the convergence analysis of the classical method. This reveals that the accuracy of the method depends in general on the size of the greatest lower bound for the absolute value of the coefficient of the highest order derivative in the differential equation. In many problems this lower bound is not close to zero and then classical methods are often satisfactory. In the case of Example 1 however, this coefficient is $-\varepsilon$. When ε is close to zero, classical methods tend to be destabilized, resulting in the oscillations mentioned above.

We now investigate this phenomenon in more detail, in order to see how to devise methods which will not misbehave so badly.

Consider again Example 1. We shall attempt to generate an approximate numerical solution by means of a standard finite

difference method. First partition [0,1] by a uniform mesh of $N+1$ points, where N is some positive integer. That is, we set $x_i = i/N$ for $i = 0, \ldots, N$. Put $h = 1/N$.

A typical classical finite difference approach would begin by approximating

$$u''(x_i) \quad \text{by} \quad \frac{u(x_{i+1}) - 2u(x_i) + u(x_{i-1})}{h^2}$$

and

$$u'(x_i) \quad \text{by} \quad \frac{u(x_{i+1}) - u(x_{i-1})}{2h}. \tag{7}$$

(If these expressions are unfamiliar, use Taylor expansions to see that each approximation is $O(h^2)$ accurate; as h is small in practice - perhaps 0.1 at most - this approximation is sufficiently accurate for most purposes.)

For each i, we write $u^h(x_i)$ for the solution which we will compute at x_i. Then based on the above difference approximations and the differential equation (2), we compute the $u^h(x_i)$ from the following linear system of equations:

$$-\varepsilon \frac{u^h(x_{i+1}) - 2u^h(x_i) + u^h(x_{i-1})}{h^2} + \frac{u^h(x_{i+1}) - u^h(x_{i-1})}{2h} = 1,$$

for $i = 1, \ldots, N-1$, where $u^h(x_0) = u^h(x_N) = 0$. This set of equations comprises our *difference scheme*.

Writing this system of equations in matrix-vector form, it is easy to see that we obtain a tridiagonal square matrix whose i th row is

$$0 \ldots 0 \quad \left(-\frac{\varepsilon}{h^2} - \frac{1}{2h}\right) \quad \frac{2\varepsilon}{h^2} \quad \left(-\frac{\varepsilon}{h^2} + \frac{1}{2h}\right) \quad 0 \ldots 0,$$

for $i = 1, \ldots, N-1$. Its first ($i = 0$) and last ($i = N$) rows, which correspond to the boundary conditions, are $1\ 0 \ldots 0$ and $0 \ldots 0\ 1$ respectively.

At this point we introduce the reader to M-matrices. This class of matrices is frequently encountered in numerical analysis. We say that a matrix $A = (A_{ij})$ is an M-matrix iff

$$A_{ij} \leq 0\ \forall i \neq j,\ A^{-1} \textit{ exists and } (A^{-1})_{ij} \geq 0\ \forall i, j.$$

The significance of M-matrices in finite difference methods is that, loosely speaking, methods which give rise to M-matrices are stable and well-behaved.

Using some well-known results for M-matrices (see, e.g., Ortega & Rheinboldt [9]), one can quickly see that our difference scheme matrix above is an M-matrix if the nonzero off-diagonal entries are negative. This is equivalent to requiring that

$$-\frac{\varepsilon}{h^2} + \frac{1}{2h} < 0,$$

i.e., that

$$h < 2\varepsilon. \tag{8}$$

When (8) is satisfied, one expects the finite difference method to be stable and to yield an accurate approximation to the true solution of Example 1. In practice this is what happens. Also, if in practice $h >> 2\varepsilon$, then one's computed solution oscillates wildly and is useless.

Note here that if $\varepsilon = 1$ (i.e., if we no longer have a convection-diffusion problem), then (8) obviously holds, so one obtains an M-matrix and hence a stable numerical method. This is why classical methods are satisfactory for problems which are not of convection-diffusion type.

One might consider the above analysis as merely indicating that classical methods may be satisfactorily employed to solve convection-diffusion problems, provided only that the mesh is chosen so that some inequality such as (8) holds. Theoretically this is so, but from a practical viewpoint (8) asks too much. For in reality one wishes to solve two- or three-dimensional problems with, say, $\varepsilon = 0.0001$ (in fact ε is smaller in many applications). With this value of ε, a problem in two dimensions for which (8) is satisfied will in its finite difference formulation (which uses a square grid with $(N+1)^2$ mesh points) have about 25,000,000 unknowns! By usual computing standards, this is an absurdly large number. In three dimensions the situation is substantially worse (exercise: about 1.25×10^{11} unknowns).

The message here is that to get a classical method to work satisfactorily, one has to provide it with an unacceptably large number of mesh points. This restriction can be avoided by constructing methods which are specially suited to convection-diffusion problems. We now show how this may be done.

3.2 Upwinding

Our troubles above with large numbers of mesh points stemmed from the fact that our matrix became an M-matrix only when h was roughly the same size as ε, i.e., only when h was small. Looking at how the entries in our matrix are related to our chosen difference approximations to u'' and u', we are led to make the following modification to our method: instead of approximating

$$u'(x_i) \qquad \text{by} \qquad \frac{u(x_{i+1}) - u(x_{i-1})}{2h},$$

approximate it by

$$\frac{u(x_i) - u(x_{i-1})}{h}. \tag{9}$$

The motivation for this alteration is that it leads to a tridiagonal difference scheme matrix whose i th row is

$$0 \ldots 0 \quad \left(-\frac{\varepsilon}{h^2} - \frac{1}{h}\right) \quad \left(\frac{2\varepsilon}{h^2} + \frac{1}{h}\right) \quad -\frac{\varepsilon}{h^2} \quad 0 \ldots 0,$$

for $i = 1, \ldots, N-1$. As the nonzero off-diagonal terms are negative, it can be shown that this is an M-matrix, irrespective of the relative sizes of h and ε. Thus when a reasonable number of mesh points is used, this difference scheme will be stable. Consequently its computed approximation will be much closer than that of our original difference scheme to the true solution.

However in stabilizing the scheme we have paid a certain price in accuracy. As we mentioned previously, (7) is an $O(h^2)$ approximation. A similar Taylor expansion reveals that (9) is only an $O(h)$ approximation. Consequently our computed solution is not expected to be extremely accurate. It turns out to be moderately accurate outside the boundary layer but inaccurate inside this layer.

In fact to obtain accuracy *inside* boundary layers requires the construction of more complicated difference schemes. We refer the reader to O'Riordan [8] for an introduction to this topic.

The technique we described above, which consisted of replacing a centred difference approximation to a first derivative by a one-sided difference approximation, is known in the research literature as *upwinding*. This name comes from the fact that stability is achieved by taking this one-sided approximation in the upstream direction (recall the discussion after Example 2 earlier). If for example one tries instead a one-sided difference approximation in the downstream direction, this does not yield stability.

Upwinding has certain drawbacks, one of which is its mediocre degree of accuracy, as we described above. Another is the difficulty of generalizing it in a satisfactory way to problems in two or three dimensions. For this reason we now consider an alternative way of stabilizing our original difference scheme.

3.3 Artificial diffusion

We return once more to the differential equation

$$-\varepsilon u''(x) + u'(x) = 1$$

of Example 1. Suppose that we have a uniform mesh with the same notation as before. We generate a difference scheme by the following two-step procedure:

(*i*) *change* ε *to* $\varepsilon + \dfrac{h}{2}$

(*ii*) *apply our original method (i.e., centred difference approximation) to this modified differential equation.*

That is, we first modify the differential equation then apply a classical difference method. Due to step (*i*), this approach is known as the *artificial diffusion* method.

On working through the details of this procedure, one finds that it yields precisely the same difference scheme matrix as upwinding! We thus have two superficially different approaches

which turn out to have identical outcomes for our one-dimensional problem. However, unlike upwinding, (a variant of) the artificial diffusion method can readily be generalized in a satisfactory manner to two or more dimensions, as we shall see in subsection 3.4.

Now recall the differential equation of Example 2:

$$-\varepsilon(u_{xx} + u_{yy}) + u_x + u = f. \tag{10}$$

Working with a square mesh of diameter h in two dimensions, the obvious generalization of our one-dimensional artificial diffusion method would be to replace ε in (10) by $\varepsilon + h/2$ then to apply a classical method to the modified differential equation. This will give a stable method, but as we describe below, it does not cope successfully with internal layers if these are present.

Recall that internal layers are narrow regions in the interior of the domain where the solution changes rapidly. If we visualize the surface $u = u(x, y)$, then an internal layer is a steep, almost sheer cliff forming part of this surface and running in the direction of the flow across the domain Ω from $\partial^-\Omega$ to $\partial^+\Omega$.

When the artificial diffusion method is applied to a problem with an internal layer, the computed solution will not include an almost sheer cliff. Instead, a moderately steep slope will be generated (this is often described by saying that the layer has been "smeared out"). The basic reason is that the method adds diffusion in all directions, including the direction perpendicular to the internal layer, so the cliff is diffused in this direction.

3.4 Streamline diffusion

The artificial diffusion method is considerably improved if the added diffusion is confined to act only in the direction of flow and not perpendicular to this direction. This idea is the basis for the *streamline diffusion* method, which is fully described in Johnson [1]. When this method is applied to (10), the diffusion term $-\varepsilon(u_{xx}+u_{yy})$ is essentially modified to $-hu_{xx}-\varepsilon u_{yy}$. The method computes reasonably sharp internal layers.

The streamline diffusion method has a further advantage over the artificial diffusion method. As we indicated in subsection 3.3,

the artificial diffusion method is somehow closely related to upwinding, while upwinding is based on an $O(h)$ approximation to the true solution. Consequently both upwinding and artificial diffusion can at best be $O(h)$ accurate, even in parts of the solution which are distant from layers. Now, using a finite element approach, the streamline diffusion method can be generated in a manner which yields better than $O(h)$ accuracy away from layers. The analysis of this higher order accuracy is discussed later.

3.5 Cell vertex finite volume method

We close our list by briefly describing one class of *finite volume* methods. Finite volume methods are a standard tool in the aerospace industry, where extremely complex numerical problems (such as modelling the airflow over an entire aircraft) are commonplace.

Our concern here is with the *cell vertex* finite volume method. To apply this method, one first divides the domain of the differential equation into many small pieces or "cells" (intervals in one dimension, rectangles in two dimensions). Here let's consider the two-dimensional formulation. One seeks a computed solution, u^h say, which is a continuous piecewise bilinear function (i.e., bilinear on each cell). The unknowns in the problem are the values of u^h at the cell corners. One generates a system of equations in these unknowns in the following way. Let C be a typical cell. In the differential equation replace u at each occurrence by u^h, then integrate the resulting equation over C using Gauss' divergence theorem. This entails computations such as

$$\int_C u^h_x dx\, dy = \int_C \nabla \cdot (u^h, 0) dx\, dy = \int_{\partial C} u^h dy,$$
$$\int_C u^h_{yy} dx\, dy = \int_C \nabla \cdot (0, u^h_y) dx\, dy = -\int_{\partial C} u^h_y dx.$$

The resulting integrals over ∂C can be expressed in terms of the values of u^h at the cell corners; this is clear for the first such integral above, while for the second some form of differencing yields a reasonable interpretation of $\int_{\partial C} u^h_y dx$. See Mackenzie & Morton [3] for details.

The above integration over C yields one equation. You might expect *a priori* that one performs this computation over each cell, but it is a curious feature of the method that this does not in general yield the correct number of equations (i.e., the number of equations may then not match the number of unknowns). One can obtain the correct number of equations by discarding or dividing cells as needed, as described in Morton [5].

3.6 Summary

In the above subsections we have given short descriptions of some numerical methods which are suited to convection-diffusion problems. Our list is by no means exhaustive (for other approaches see, e.g., Miller [4]). No panacea currently exists; for each method, one can exhibit examples to which an application of the method yields disappointingly inaccurate results.

Any proposed method for convection-diffusion problems, if it is to have any chance of success, must somehow mimic the behaviour of the true solution discussed in Section 2. That is, away from layers its computed solution at each point should depend only on what happens in a narrow region directly upstream of that point. Each of our methods has this property to a greater or lesser extent.

4. Numerical analysis

In Section 3 we described some standard numerical methods which are suited to convection-diffusion problems. We now indicate briefly the techniques which are used to prove that such methods do indeed yield accurate numerical approximations to the true solutions of these problems.

For finite difference methods, the basic ideas are those of *inverse monotonicity* and *barrier functions*. These can be described quite simply. Suppose that the discrete linear system of equations is

$$L^h u^h = f^h,$$

where L^h is the matrix arising from the difference scheme and u^h will be our computed solution. The right hand vector f^h is known. Suppose (this is inverse monotonicity) that

$$(L^h)^{-1} \geq 0,$$

where the inequality holds for each entry in the matrix. It's of course reasonable to assume that $(L^h)^{-1}$ exists (otherwise we could not compute a solution u^h); the essential feature here is that all its entries are non-negative.

Let $\tilde{u}$ denote the restriction of the true solution u to the mesh points. The consistency error $L^h(u^h - \tilde{u})$ can be estimated by Taylor expansions. Warning: this calculation is tedious and involves a lot of careful estimation of terms!

Using this consistency error estimate, one next tries to construct a discrete function w^h which satisfies the vector inequalities $w^h \geq 0$ and $L^h w^h \geq |L^h(u^h - \tilde{u})|$. This is often not as difficult as it looks; one chooses w^h to mimic certain properties that one expects in the true solution u. The function w^h is known as a barrier function.

Finally, combining the last inequality with the inverse monotonicity property, we deduce that

$$|u^h - \tilde{u}| \leq w^h,$$

which is a satisfactory result provided that w^h is small.

A rather famous example of the use of this technique is provided by Kellogg & Tsan [2].

The main drawback to this method of analysis is the assumption of inverse monotonicity. In the context of convection-diffusion problems, this property often holds for the matrices arising from ordinary differential equations but is less frequently true for problems in two and three dimensions. Thus other analytical techniques are needed.

For the streamline diffusion method in two dimensions, various global error estimates have been proven. However these are not satisfactory since they are expressed in terms of Sobolev norms of the true solution u, which become excessively large when ϵ is small. The best local estimates available have been obtained by

Niijima [7]. He shows that, away from all layers, $O(h^{11/8}\ln(1/h))$ accuracy is achieved when ε is small. (Here h denotes the mesh diameter, just as in our one-dimensional investigations.) His approach uses finite element techniques to obtain local bounds on a discrete Green's function (this function is basically the inverse of the difference scheme matrix). Hence one can readily deduce convergence of the streamline diffusion method in regions that are not close to any layers. This analysis is very technical but it works.

Analysis of convergence of the cell vertex finite volume method has lagged far behind the application of the method. Up to now, no fully satisfactory analysis of this method has been published. The best estimates available are in Morton & Stynes [6], where a sharp convergence result for the one-dimensional case is obtained in a weighted discrete Sobolev H^1 norm. This bound is obtained by using techniques from finite element analysis. At present we are working on an extension of this result to the two-dimensional case.

References

[1] C. Johnson, Numerical solution of partial differential equations by the finite element method. Cambridge University Press: U.K., 1988.

[2] R. B. Kellogg and A. Tsan, *Analysis of some difference approximations for a singular perturbation problem without turning points*, Math. Comp. **32** (1976), 1025–1039.

[3] J. A. Mackenzie and K. W. Morton, *Finite volume solutions of convection-diffusion test problems*, Math. Comp. (to appear, Jan. 1993).

[4] J. J. H. Miller (ed.), Computational methods for boundary and interior layers in several dimensions. Boole Press: Dublin, 1991.

[5] K. W. Morton, *Finite volume methods and their analysis* in The mathematics of finite elements and applications VII. Proceedings of MAFELAP 1990, Brunel, April 1990, J. R. Whiteman (ed.), Academic Press: London, 1991, 189–214.

[6] K. W. Morton and M. Stynes, *An analysis of the cell vertex scheme* (1992) (Report Number 91/7, Oxford University Computing Laboratory).

[7] K. Niijima, *Pointwise error estimates for a streamline diffusion finite element scheme*, Numer. Math. **56** (1990), 707–719.

[8] E. O'Riordan, *Numerical methods for singularly perturbed differential equations*, Bull. Irish Math. Soc. **27** (1986), 14–24.

[9] J. M. Ortega and W. C. Rheinboldt, Iterative solution of nonlinear equations in several variables. Academic Press: New York, 1970.

[10] J. A. Spriet and G. C. Vansteenkiste, Computer-aided modelling and simulation. Academic Press: London, 1982.

Martin Stynes,
Department of Mathematics,
University College,
Cork.