

Numerical stability of unsteady stream-function vorticity calculations

E. Sousa^{1,*},[†] and I. J. Sobey²

¹*Departamento de Matemática, Universidade de Coimbra, Apartado 3008, 3000 Coimbra, Portugal*

²*Oxford University Computing Laboratory, Wolfson Building, Parks Road, Oxford OX1 3QD, U.K.*

SUMMARY

The stability of a numerical solution of the Navier–Stokes equations is usually approached by considering the numerical stability of a discretized advection–diffusion equation for either a velocity component, or in the case of two-dimensional flow, the vorticity. Stability restrictions for discretized advection–diffusion equations are a very serious constraint, particularly when a mesh is refined in an explicit scheme, so an accurate understanding of the numerical stability of a discretization procedure is often of equal or greater practical importance than concerns with accuracy. The stream-function vorticity formulation provides two equations, one an advection–diffusion equation for vorticity and the other a Poisson equation between the vorticity and the stream-function. These two equations are usually not coupled when considering numerical stability. The relation between the stream-function and the vorticity is linear and so has, in principle, an exact inverse. This allows an algebraic method to link the interior and the boundary vorticity into a single iteration scheme. In this work, we derive a global time-iteration matrix for the combined system. When applied to a model problem, this matrix formulation shows differences between the numerical stability of the full system equations and that of the discretized advection–diffusion equation alone. It also gives an indication of how the wall vorticity discretization affects stability. Despite the added algebraic complexity, it is straightforward to use MATLAB to carry out all the matrix operations. Copyright © 2003 John Wiley & Sons, Ltd.

KEY WORDS: Navier–Stokes; stream-function vorticity; finite differences; stability analysis

1. INTRODUCTION

The simplest formulation of the Navier–Stokes equations for two-dimensional unsteady flow is in terms of a stream-function, ψ , and the vorticity, ω . There are two non-dimensional parameters which occur in the non-dimensionalized Navier–Stokes equations, a Reynolds number, R , and a Strouhal number, St . If the time scale for the flow is taken as the ratio of the length scale to the velocity scale, then $St = 1$ and the Reynolds number is the only parameter. The

*Correspondence to: E. Sousa, Departamento de Matemática, Universidade de Coimbra, Apartado 3008, 3000 Coimbra, Portugal.

[†]E-mail: ecs@mat.uc.pt.

stream-function vorticity formulation gives two equations, one an advection–diffusion equation for the vorticity and the other a relation between the vorticity and the stream-function.

In this case, the vorticity equation is

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{1}{R} \nabla^2 \omega \quad (1)$$

the domain is some part of (x, y) space and the fluid velocities are defined by

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \quad (2)$$

where the stream-function, ψ , is connected to the vorticity, ω , by

$$\omega = -\nabla^2 \psi \quad (3)$$

Equations (1) and (3) each require a boundary condition on each part of the boundary and a number of different ways are possible to specify the two conditions: for instance in a channel flow the velocities may vanish on the channel walls and the inlet and outlet conditions may involve specifying inlet velocities and outlet conditions on the stream-function and the vorticity. In channel flows it is common to specify the stream-function and its normal derivative on channel walls rather than the velocities, so for example the stream-function could be fixed on each of the channel walls and the difference between the two wall values of the stream-function would determine the flux through the channel. The normal derivative vanishing on the wall would then be used in determining the wall vorticity in a numerical scheme.

This means that numerical solution of the stream-function vorticity form of the equations is usually particularly straight forward using an explicit scheme. In that case, the variables are discretized in time and space and the discrete form of the advection–diffusion equation is used to calculate the vorticity in the interior of the flow at a new time step. The Poisson equation between ψ and ω is then solved in the interior of the flow domain to determine the new value of ψ there and the updated values of the stream-function enable the Poisson equation to be used a second time to determine the vorticity on solid boundaries. In this process the main computational element is not the vorticity update procedure, which is relatively quick, but rather the solution of the Poisson equation which normally proceeds using multigrid and which is relatively time consuming.

There is of course a large literature dealing with the stream-function vorticity approach and most computational fluid mechanics texts describe techniques for discretizing the equations, see for instance Hirsch [1] or Peyret and Taylor [2]. Our first concern here is not with the fine detail of how the equations are discretized but rather in formulating the iterative procedure in terms of a global iteration matrix, something which does not seem to have been done previously.

As an example we consider a relatively simple one-dimensional model problem which mimics the major features of the stream-function vorticity formulation. Our experience in examining the stability constraints of an advection–diffusion equation alone is that general guidelines which come from one-dimensional model problems are largely applicable to two-dimensional equations so we think the results of this simple model problem are sufficiently interesting to be examined now although we intend in a further work to consider more

complicated two-dimensional problems using the same general formulation for the global iteration matrix and its stability.

2. FORMULATION

In this section, to bring out the basic features of the global iteration matrix in as clear a way as is possible, we derive the global iteration matrix for the simplified case where the walls are not moving. We consider the formulation for a more general situation in the final section.

Divide the computational domain into two sets: points in the interior of the region where the vorticity is updated through the advection–diffusion equation, (1), and points on the boundary where the vorticity update occurs after the stream-function has been updated to the new time level. The set of interior points is denoted by a subscript I while the boundary points are denoted by a subscript B , so that \mathbf{W}_I^n and Ψ_I^n are vorticity and stream-function values in the interior (which may of course, include inflow or outflow boundary points where the vorticity is still determined by advection–diffusion) at time $t = t_n$. The boundary sets are then \mathbf{W}_B^n and Ψ_B^n . Underlying the discussion here is an assumption of a uniform mesh of size h in both directions but it is not an essential assumption and the theory could be developed analogously for general meshes.

Most explicit time marching discretization for the vorticity equation, (1), can be written

$$\mathbf{W}_I^{n+1} = \mathbf{A}\mathbf{W}_I^n + \mathbf{B}\mathbf{W}_B^n \quad (4)$$

for suitable matrices \mathbf{A} and \mathbf{B} . This of course hides the non-linearity of the Navier–Stokes equations since the matrices \mathbf{A} and \mathbf{B} are both functions of the stream-function but for the moment that can be kept in the background.

Next the stream-function in the interior has to be determined from the updated vorticity in the interior. This is a linear equation, a discrete form of (3), which can be written

$$\frac{1}{h^2} \mathbf{L}\Psi_I^{n+1} + \frac{1}{h^2} \mathbf{N}\Psi_B^{n+1} = -\mathbf{W}_I^{n+1} \quad (5)$$

where \mathbf{L} , \mathbf{N} are suitable matrices for discretization of the Laplace operator. The values of the stream-function on the boundaries will have been specified by the flux through the flow region.

The boundary vorticity values are then found from applying a discrete form of (3) at the boundaries using the updated values of the stream-function,

$$\mathbf{W}_B^{n+1} = \frac{1}{h^2} \mathbf{M}\Psi_I^{n+1} + \frac{1}{h^2} \mathbf{J}\Psi_B^{n+1} + \mathbf{F}\mathbf{W}_I^{n+1} \quad (6)$$

where \mathbf{M} , \mathbf{J} and \mathbf{F} are suitable matrices. The matrix \mathbf{F} is zero in many formulations but for instance in Woods' method [3] the interior vorticity is used in updating the wall vorticity with second order accuracy.

This enables the stream-function to be eliminated,

$$\mathbf{W}_B^{n+1} = (\mathbf{F} - \mathbf{M}\mathbf{L}^{-1})\mathbf{W}_I^{n+1} + \frac{1}{h^2} (\mathbf{J} - \mathbf{L}^{-1}\mathbf{N})\Psi_B^{n+1} \quad (7)$$

and then the update of the vorticity in the interior can be replaced using (4), so that

$$\mathbf{W}_B^{n+1} = (\mathbf{F} - \mathbf{M}\mathbf{L}^{-1})(\mathbf{A}\mathbf{W}_I^n + \mathbf{B}\mathbf{W}_B^n) + \frac{1}{h^2} (\mathbf{J} - \mathbf{L}^{-1}\mathbf{N})\mathbf{\Psi}_B^{n+1} \quad (8)$$

This essentially completes the derivation of the global iteration matrix, since we now have

$$\begin{bmatrix} \mathbf{W}_I^{n+1} \\ \mathbf{W}_B^{n+1} \end{bmatrix} = \begin{bmatrix} \mathbf{A} & \mathbf{B} \\ (\mathbf{F} - \mathbf{M}\mathbf{L}^{-1})\mathbf{A} & (\mathbf{F} - \mathbf{M}\mathbf{L}^{-1})\mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{W}_I^n \\ \mathbf{W}_B^n \end{bmatrix} + \frac{1}{h^2} \begin{bmatrix} 0 \\ (\mathbf{J} - \mathbf{L}^{-1}\mathbf{N})\mathbf{\Psi}_B^{n+1} \end{bmatrix} \quad (9)$$

or

$$\mathbf{W}^{n+1} = \mathbf{K}\mathbf{W}^n + \mathbf{S}^{n+1} \quad (10)$$

where we use \mathbf{K} to denote the overall iteration matrix and the vector \mathbf{S}^{n+1} is known from the wall stream-function. The usual method to consider stability is to examine the eigenvalues of the matrix \mathbf{A} , the advection–diffusion part. In the case of the Navier–Stokes equations that is not sufficient and it is the eigenvalues of a global iteration matrix such as \mathbf{K} which need to be considered.

3. MODEL PROBLEM

The iteration matrix \mathbf{K} may be difficult to calculate for an arbitrary domain and associated mesh but all the matrix algebra, including matrix inversion is within the capability of MATLAB. It is true that the separation of the domain into interior and exterior points introduces an ordering which is a little involved from an implementation viewpoint and we discuss this further in the conclusion. In the case of one space dimension the ordering is fairly simple and we have examined the following model problem.

The ‘vorticity’, ω , is described by

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} = \frac{1}{R} \frac{\partial^2 \omega}{\partial x^2}, \quad 0 \leq x \leq 1 \quad (11)$$

and the ‘vorticity’ is related to a function $\psi(x, t)$ by

$$\frac{\partial^2 \psi}{\partial x^2} = -\omega \quad (12)$$

In the examples below we assume that the velocity, u , is a positive constant and that the boundary conditions on the function ψ are $\psi(0, t) = \psi(1, t) = 0$. Of course the longtime solution to this problem will be $\psi = \omega = 0$ but that does not alter the utility of the model problem for studying numerical stability of the time marching iteration process.

4. DISCRETIZATION OF THE VORTICITY EQUATION

Assume then that this system is discretized in space at points $x_j = jh$, $j = 0, \dots, M$ with $h = 1/M$ and discretized in time with a time step Δt so that $t_n = n\Delta t$. The interior of the

domain will be the points $j = 1, \dots, M - 1$ while the boundary points are at $j = 0$ and $j = M$. A simple explicit forward difference in time is used so that

$$\frac{\partial \omega}{\partial t} \approx \frac{1}{\Delta t} (W_j^{n+1} - W_j^n) \quad (13)$$

In this model problem the interior and boundary vectors are $\mathbf{W}_I^n = [W_1^n, W_2^n, \dots, W_{M-1}^n]^T$, and $\mathbf{W}_B^n = [W_0^n, W_M^n]^T$. We consider a number of different space discretization schemes, both for the advection–diffusion equation and for specifying the boundary vorticity. The first set use the simplest discretization for the boundary vorticity,

$$W_0^{n+1} = -2\Psi_1^{n+1}/h^2, \quad W_M^{n+1} = -2\Psi_{M-1}^{n+1}/h^2 \quad (14)$$

so that the matrix \mathbf{F} vanishes and the matrix \mathbf{M} is simply

$$\mathbf{M} = \begin{bmatrix} -2 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \cdots & 0 & -2 \end{bmatrix} \quad (15)$$

with $M - 1$ columns. If the Poisson equation, (12) is discretized using second order central differences, the square matrix \mathbf{L} of size $M - 1$ is also simple to write down and will not vary in any of the following, the matrix $\mathbf{L} = (l_{i,j})$ is a tridiagonal matrix defined by

$$l_{i,i} = -2, \quad i = 1, \dots, M - 1, \quad l_{i,i-1} = 1, \quad i = 2, \dots, M - 1, \quad l_{i,i+1} = 1, \quad i = 1, \dots, M - 2 \quad (16)$$

The remaining matrices \mathbf{A} and \mathbf{B} depend on the discretization method but once specified they can be calculated by MATLAB and the matrix \mathbf{K} assembled and analysed. In the next few sections we use two numerical parameters, a Courant number, and a parameter representing the effect of viscosity, respectively, given by

$$v = \frac{u\Delta t}{h} \quad \text{and} \quad \mu = \frac{\Delta t}{Rh^2} \quad (17)$$

The calculations we present are, unless otherwise stated, for the case $M = 30$. Experiments with larger values of M only change the results a little and not in any significant way. Define the operators,

$$\Delta_0 W_j := \frac{1}{2} (W_{j+1} - W_{j-1}), \quad \delta^2 W_j := W_{j+1} - 2W_j + W_{j-1}$$

$$\Delta_- W_j := W_j - W_{j-1}, \quad \Delta_+ W_j := W_{j+1} - W_j$$

4.1. Lax–Wendroff

If the advection–diffusion equation is discretized using a compact three point Lax–Wendroff scheme [4],

$$W_j^{n+1} = [1 - v\Delta_0 + (\mu + v^2/2)\delta^2]W_j^n \quad (18)$$

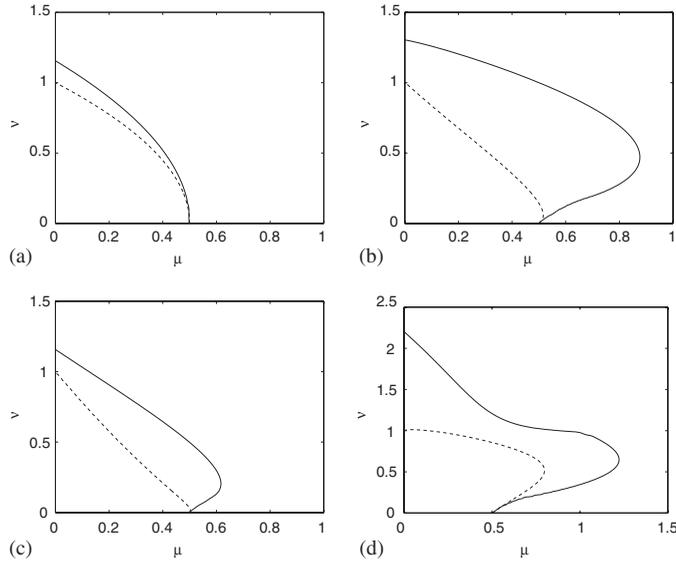


Figure 1. Contours of unit spectral radius for first order boundary vorticity. (—) advection–diffusion operator, (– –) Navier–Stokes operator. The region of stability is the area between the curves and the axes: (a) Lax–Wendroff scheme; (b) Quickest with Lax–Wendroff at the first point; (c) Quickest with downwind at the first point; and (d) Quickest using Lax–Wendroff at inlet. Note that the scale in figure (d) is different from the others.

then the $(M - 1) \times (M - 1)$ matrix $\mathbf{A} = (a_{i,j})$ is a tridiagonal matrix with entries given by

$$\begin{aligned} a_{i,i} &= 1 - 2\lambda, & i = 1, \dots, M - 1 \\ a_{i,i-1} &= \lambda + \nu/2, & i = 2, \dots, M - 1 \\ a_{i,i+1} &= \lambda - \nu/2, & i = 1, \dots, M - 2 \end{aligned} \tag{19}$$

where $\lambda = \mu + \nu^2/2$. The $(M - 1) \times 2$ matrix $\mathbf{B} = (b_{i,j})$ is given by

$$b_{1,1} = \lambda + \nu/2, \quad b_{M-1,2} = \lambda - \nu/2 \tag{20}$$

and zero on the other entries. The matrices \mathbf{A} and \mathbf{B} (and hence \mathbf{K}) depend on the two numerical parameters μ and ν and in Figure 1(a) we show the contour in (μ, ν) -space where the spectral radius $\rho(\mathbf{A}) = 1$ and $\rho(\mathbf{K}) = 1$. It can be seen that the region of stability is reduced somewhat for the full system compared to the advection–diffusion equation alone.

4.2. Quickest with Lax–Wendroff at first point

The application of a higher order method such as the third order Quickest scheme introduced by Leonard [5] requires two upstream points and is given by

$$W_j^{n+1} = \left[1 - \nu \Delta_0 + \left(\mu + \frac{1}{2} \nu^2 \right) \delta^2 + \nu \left(\frac{1}{6} - \frac{\nu^2}{6} - \mu \right) \delta^2 \Delta_- \right] W_j^n \tag{21}$$

Of course this is not straightforward at the first interior point where there is only one upstream point. In previous studies we have considered a number of options to get round this problem, see Sousa [6, 7] and Sousa and Sobey [8]. The first scheme we examine here is just to use the three point Lax–Wendroff scheme at the first interior point and Quickest at subsequent points. This modifies the **A** and **B** matrices a little. Quickest introduces a third order difference with coefficient $\gamma = v(1 - v^2 - 6\mu)/6$. If we keep the Lax–Wendroff scheme at the first point, and let

$$s_{ll} = -\gamma, \quad s_l = \lambda + v/2 + 3\gamma, \quad s_c = 1 - 2\lambda - 3\gamma, \quad s_r = \lambda - v/2 + \gamma \quad (22)$$

then the matrix **A** is a band matrix, with bandwidth 4, given by

$$\begin{aligned} a_{1,1} &= 1 - 2\lambda, & a_{1,2} &= \lambda - v/2, & a_{i,i-2} &= -s_{ll}, \quad i=3, \dots, M-1 \\ a_{i,i-1} &= s_l, \quad i=2, \dots, M & a_{i,i+1} &= s_r, \quad i=2, \dots, M-2 & a_{i,i} &= s_c, \quad i=2, \dots, M-1 \end{aligned} \quad (23)$$

The matrix **B** is given by

$$b_{1,1} = \lambda + v/2, \quad b_{2,1} = s_{ll} \quad b_{M-1,2} = s_r \quad (24)$$

and the other entries are zero.

The stability boundaries for this scheme is shown in Figure 1(b) and the region of stability for the Navier–Stokes type problem is considerably smaller than that for the advection–diffusion operator.

4.3. Quickest with downwinded first point

A second solution for the problem of which method to apply at the first point is to use a downwinded third difference. In principle this does not affect the order of accuracy of the solution. In this case the matrix **A** is a band matrix, with bandwidth 5, defined by

$$\begin{aligned} a_{1,1} &= 1 - 2\mu + 3\gamma, & a_{1,2} &= \lambda - v/2 - 3\gamma, & a_{1,3} &= \gamma \\ a_{i,i} &= s_c, \quad i=2, \dots, M-1 & a_{i,i+1} &= s_r, \quad i=2, \dots, M-2 & a_{i,i+2} &= 0, \quad i=2, \dots, M-3 \\ a_{i,i-1} &= s_l, \quad i=2, \dots, M & a_{i,i-2} &= s_{ll}, \quad i=3, \dots, M-1 \end{aligned} \quad (25)$$

and the **B** matrix is given by

$$b_{1,1} = \lambda + v/2 - \gamma, \quad b_{2,1} = s_{ll} \quad b_{M-1,2} = s_r \quad (26)$$

and the other entries are zero.

The computed stability boundaries are shown in Figure 1(c) and again the region of stability of the model problem is somewhat smaller than the stability region for the advection–diffusion operator alone but is very similar to that obtained in Figure 1(b).

4.4. Quickest modified by Lax–Wendroff at inlet: fictitious point

The final Quickest type method we consider is one whereby Lax–Wendroff is applied at $x=0$ using the boundary variation of the advected quantity at inlet to obtain an estimate for a fictitious point at $x = -h$ so that Quickest may be applied consistently at the first interior point. A word of warning: it is not clear how this scheme might be implemented in a practical stream-function vorticity solver, we are able to do so here because the Poisson equation for the stream-function is solved exactly. If the solution were to proceed through an iterative solver it is unclear how the updated boundary vorticity could be incorporated in practice. This example is included to show how improving the treatment of the first interior point may lead to significant stability improvement.

The premise of this method is to assume the existence of an upstream point at $x = -h$ and apply Lax–Wendroff at $x=0$ giving

$$W_0^{n+1} = d_l W_{-1}^n + d_c W_0^n + d_r W_1^n \quad (27)$$

where the coefficients are

$$d_l = \mu + \frac{1}{2} v^2 + v/2, \quad d_c = 1 - 2\mu + v^2, \quad d_r = \mu + \frac{1}{2} v^2 - v/2 \quad (28)$$

What is assumed is that the boundary values W_0^{n+1} and W_0^n are known, so that this equation can be rearranged to give

$$W_{-1}^n = \frac{1}{d_l} W_0^{n+1} - \frac{d_c}{d_l} W_0^n - \frac{d_r}{d_l} W_1^n \quad (29)$$

Of course in the case of advection–diffusion of a material quantity such as concentration, the boundary values at the inlet are known as an explicit boundary condition, it is only in the case of this form of the Navier–Stokes equations that the boundary values are coupled to the solution in the interior of the domain at the new time level.

This approximation enables Quickest to be applied in unmodified form at the first interior point, $x=h$, so that

$$W_1^{n+1} - \frac{s_{ll}}{d_l} W_0^{n+1} = \left(s_l - \frac{s_{ll} d_c}{d_l} \right) W_0^n + \left(s_c - \frac{s_{ll} d_r}{d_l} \right) W_1^n + s_r W_2^n \quad (30)$$

This gives the matrix **A** a band matrix, with bandwidth 4, defined as

$$\begin{aligned} a_{1,1} &= s_c - s_{ll} d_r / d_l, & a_{i,i} &= s_c, & i=2, \dots, M-1 & \quad a_{i,i+1} &= s_r, & i=2, \dots, M-2 \\ a_{i,i-1} &= s_l, & i=2, \dots, M & \quad a_{i,i-2} &= s_{ll}, & i=3, \dots, M-1 \end{aligned} \quad (31)$$

and the **B** matrix is given by

$$b_{1,1} = s_l - s_{ll} d_c / d_l, \quad b_{2,1} = s_{ll} \quad b_{M-1,2} = s_r \quad (32)$$

and the other entries are zero.

The complication that this modification introduces to the scheme is now clear because the iteration (10) has to be modified to

$$\mathbf{VW}^{n+1} = \mathbf{KW}^n \quad (33)$$

Fortunately in this case the matrix \mathbf{V} is particularly simple,

$$\mathbf{V} = \begin{bmatrix} 1 & 0 & \cdots & -s_{ll}/d_l & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \end{bmatrix} \quad (34)$$

Hence it is straight forward to apply Gauss elimination to the matrix \mathbf{K} to effectively apply \mathbf{V}^{-1} by adding a multiple s_{ll}/d_l of the second last row of \mathbf{K} to the first row and reduce (33) to the same form as (10).

The results of computing the stability boundary are shown in Figure 1(d). Both the advection–diffusion operator and the Navier–Stokes type operators show much improved stability region over the other methods we have considered although again, the Navier–Stokes type equation has a more restricted stability region compared to the advection–diffusion operator.

4.5. HODS

As an example of application of this theory to an alternative discretization, consider the HODS scheme described in Castro and Jones [9]. In their case the advection term was discretized by

$$u \frac{\partial \omega}{\partial x} \approx \frac{u}{h} \left(\frac{3}{2} W_j - 2W_{j-1} + \frac{1}{2} W_{j-2} \right) \quad (35)$$

We denote $h_{ll} = v/2$, $h_l = -2v + \mu$, $h_c = 1 + 3v/2 - 2\mu$ and $h_r = \mu$. This gives the matrix \mathbf{A} a band matrix, with bandwidth 4, defined as

$$\begin{aligned} a_{1,1} &= 1 - 2v - 2\mu, & a_{1,2} &= (3/2)v + \mu, & a_{i,i} &= h_c, & i &= 2, \dots, M-1 \\ a_{i,i+1} &= h_r, & i &= 2, \dots, M-2 & a_{i,i-1} &= h_l, & i &= 2, \dots, M \\ a_{i,i-2} &= h_{ll}, & i &= 3, \dots, M-1 \end{aligned} \quad (36)$$

and the \mathbf{B} matrix is given by

$$b_{1,1} = h_{ll} + h_r, \quad b_{2,1} = h_{ll} \quad b_{M-1,2} = h_r \quad (37)$$

and the other entries are zero.

We have combined this with a central difference for the diffusion term and applied first order upwinding at the first interior mesh point. The results for stability are shown in Figure 2.

5. DISCRETIZATION OF THE VORTICITY ON THE WALLS

A number of different discretization formulae for the boundary vorticity have been proposed. For a survey and discussion of boundary vorticity formulae, see for instance Napolitano

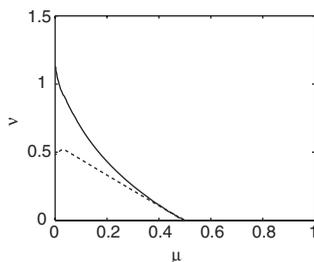


Figure 2. Contours of unit spectral radius for first order boundary vorticity with the HODS scheme. (—) advection–diffusion operator, (– –) Navier–Stokes operator. The region of stability is the area between the curves and the axes.

et al. [10]. In this section we illustrate how different discretization for the boundary vorticity can be incorporated into the global iteration matrix.

An approximation for the wall vorticity which uses an additional point for the stream-function in the domain to increase the accuracy of the vorticity approximation and which is a second order scheme, is given by

$$h^2 W_0 = -4\Psi_1 + \frac{1}{2}\Psi_2, \quad h^2 W_M = -4\Psi_{M-1} + \frac{1}{2}\Psi_{M-2} \quad (38)$$

This variation can be easily implemented by varying the matrix \mathbf{M} . The matrix \mathbf{M} is simply

$$\mathbf{M} = \begin{bmatrix} -4 & 1/2 & 0 & \cdots & 0 & 0 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1/2 & -4 \end{bmatrix} \quad (39)$$

The effect on numerical stability of combining this boundary vorticity discretization with Lax–Wendroff and the three Quickest schemes (which use different numerical boundary conditions at the inlet) described in Sections 4.1–4.4 is illustrated in Figure 3(a)–3(d). It is evident that attempting to improve accuracy in this particular way has a substantial stability penalty for each of the different discretizations.

6. CONCLUSION

In this study we consider the global iteration matrix for explicit time marching stream-function vorticity calculations. The usual approach to numerical stability of such calculations is by examination of the discretized advection–diffusion equation alone. Using the algebraic capabilities of MATLAB we have shown that the global iteration matrix can be calculated and studied for a one-dimensional problem which mimics the two dimensional stream-function vorticity equations. There are two numerical parameters and the study of this model problem show us that in terms of these numerical parameters, the region of numerical stability for a Navier–Stokes like equation is smaller than the region for a discretized advection–diffusion equation alone. We also examine the influence of a different boundary vorticity discretization technique and note that the use of a higher-order discretization may have a significant penalty on the numerical stability of the full system of equations.

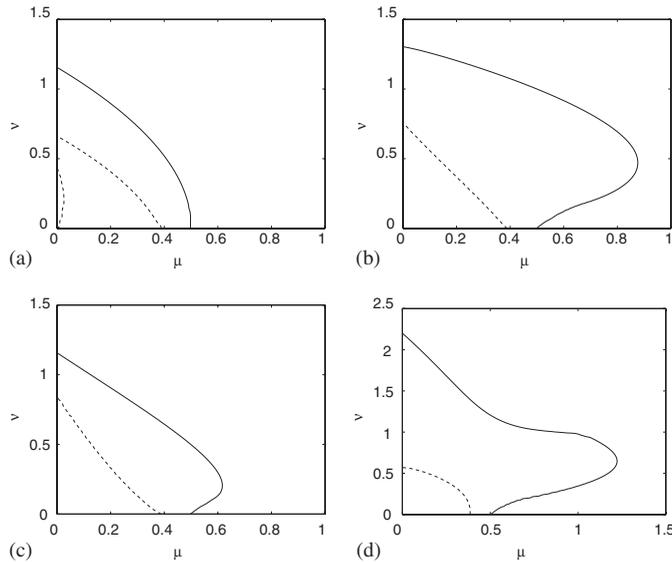


Figure 3. Contours of unit spectral radius for second order boundary vorticity. (—) advection–diffusion operator, (- -) Navier–Stokes operator. The region of stability is the area between the curves and the axes: (a) Lax–Wendroff scheme; (b) Quickest with Lax–Wendroff at the first point; (c) Quickest with downwind at the first point; and (d) Quickest using Lax–Wendroff at inlet. Note that the scale in figure (d) is different from the others.

Of course the real test of this approach will come from examination of the proper two-dimensional stream-function vorticity equations. Our view is that such an extension of this work is achievable. Whereas the coefficient matrices in Equations (4)–(6) were straightforward to write down for a simple one-dimensional problem they are less obvious for a two-dimensional problem since we need to map the physical approximation points to the two vectors of interior and boundary points. This is possible by observing that the vector (Ψ_B, Ψ_I) is just a permutation of the stream-function at mesh points ordered by more physical considerations. For example, if we consider a driven cavity flow then the natural ordering of the stream-function values gives an ordering of values $\psi_{rs} = \psi(rh, sh)$, $r, s = 0, \dots, m = 1/h$ as $(0, 0), (1, 0), \dots, (m, 0), (0, 1), \dots, (m, m)$. Suppose we denote a vector of values Ψ_{rs} which approximate ψ_{rs} by this natural ordering $\hat{\Psi} = [\Psi_{0,0}, \Psi_{1,0}, \Psi_{2,0}, \dots, \Psi_{m,m}]^T$, then there will exist a permutation matrix \mathbf{P} such that

$$\hat{\Psi} = \mathbf{P} \begin{bmatrix} \Psi_B \\ \Psi_I \end{bmatrix} \tag{40}$$

where Ψ_B are values on the boundary (r or s equal to 0 or m , in this case $4m$ values), Ψ_I those values in the interior ($0 < r, s < m$, in this case $(m - 1)^2$ values) and if \mathbf{P} is suitable partitioned as $\mathbf{P} = [\mathbf{P}_1, \mathbf{P}_2]$ then

$$\hat{\Psi} = \mathbf{P}_1 \Psi_B + \mathbf{P}_2 \Psi_I \tag{41}$$

In a similar manner the vorticity approximation \mathbf{W}_{rs} , can be written

$$\hat{\mathbf{W}} = [\mathbf{W}_{0,0}, \mathbf{W}_{1,0}, \mathbf{W}_{2,0}, \dots, \mathbf{W}_{m,m}]^T = \mathbf{P}_1 \mathbf{W}_B + \mathbf{P}_2 \mathbf{W}_I \quad (42)$$

Next, the stream-function vorticity equation, can be discretized at the interior points by

$$\mathbf{R} \hat{\Psi} = -h^2 \mathbf{W}_I \quad (43)$$

where \mathbf{R} is easy to write down in terms of the natural ordering of $\hat{\Psi}$, so that in Equation (5) $\mathbf{L} = \mathbf{R} \mathbf{P}_2$ and $\mathbf{N} = \mathbf{R} \mathbf{P}_1$. In the driven cavity problem \mathbf{R} is a $(m-1)^2 \times (m+1)^2$ and \mathbf{P}_2 is an $(m+1)^2 \times (m-1)^2$ matrix keeping \mathbf{L} as a $(m-1)^2 \times (m-1)^2$ matrix. If the vorticity equation is discretized at the interior points (again using the natural ordering of $\hat{\Psi}^n$),

$$\mathbf{W}_I^{n+1} = \mathbf{G} \hat{\mathbf{W}}^n = \mathbf{G}[\mathbf{P}_1 \mathbf{W}_B^n + \mathbf{P}_2 \mathbf{W}_I^n] \quad (44)$$

then

$$\mathbf{W}_I^{n+1} = \mathbf{G} \mathbf{P}_1 \mathbf{W}_B^n + \mathbf{G} \mathbf{P}_2 \mathbf{W}_I^n \quad (45)$$

so that the matrix \mathbf{A} and \mathbf{B} in (4) are determined by $\mathbf{A} = \mathbf{G} \mathbf{P}_2$ and $\mathbf{B} = \mathbf{G} \mathbf{P}_1$. The boundary vorticity is a little more complicated, but if we use a natural ordering to obtain on the boundary

$$\mathbf{D}_1 \hat{\mathbf{W}}^{n+1} = \frac{1}{h^2} \mathbf{D}_2 \hat{\Psi}^{n+1} + \mathbf{v}^{n+1} \quad (46)$$

where \mathbf{v}^{n+1} might for instance arise in a driven cavity problem where the walls are moving, then

$$\mathbf{D}_1[\mathbf{P}_1 \mathbf{W}_B^{n+1} + \mathbf{P}_2 \mathbf{W}_I^{n+1}] = \frac{1}{h^2} \mathbf{D}_2[\mathbf{P}_1 \Psi_B^{n+1} + \mathbf{P}_2 \Psi_I^{n+1}] + \mathbf{v}^{n+1} \quad (47)$$

so that

$$\mathbf{D}_1 \mathbf{P}_1 \mathbf{W}_B^{n+1} = \frac{1}{h^2} \mathbf{D}_2 \mathbf{P}_1 \Psi_B^{n+1} + \frac{1}{h^2} \mathbf{D}_2 \mathbf{P}_2 \Psi_I^{n+1} - \mathbf{D}_1 \mathbf{P}_2 \mathbf{W}_I^{n+1} + \mathbf{v}^{n+1} \quad (48)$$

Consequently the matrices \mathbf{M} and \mathbf{F} in Equation (6) are, respectively, given by the equalities $\mathbf{M} = (\mathbf{D}_1 \mathbf{P}_1)^{-1} \mathbf{D}_2 \mathbf{P}_2$, $\mathbf{F} = -(\mathbf{D}_1 \mathbf{P}_1)^{-1} \mathbf{D}_1 \mathbf{P}_2$. In this way the global iteration matrix can still be calculated using MATLAB to carry out all the various matrix calculations and then stability analysed by examining eigenvalues of that matrix.

REFERENCES

1. Hirsch C. *Numerical Computation of Internal and external flows*, Vol I and II. Wiley Interscience: Chichester, 1990.
2. Peyret R, Taylor TD. *Computational Methods for Fluid Flow*. Springer: Berlin, 1986.
3. Woods LC. A note on the numerical solution of fourth order differential equations. *Aeronautical Quarterly* 1954; **5**:176.
4. Lax PD, Wendroff B. Systems of conservations laws. *Communications on Pure and Applied Mathematics* 1960; **13**:217–237.
5. Leonard BP. A stable and accurate convective modeling procedure based on quadratic upstream interpolation. *Computer Methods in Applied Mechanics and Engineering* 1979; **19**:59–98.
6. Sousa E. Finite differences for the convection–diffusion equation: On stability and boundary conditions. *Ph.D. Thesis*, Oxford University, 2001.

7. Sousa E. A Godunov-Ryabenkii instability for a quickest scheme. *Lecture Notes in Computer Science* 2001; **1988**:732–740.
8. Sousa E, Sobey IJ. On the influence of boundary conditions. *Applied Numerical Mathematics* 2002; **41**: 325–344.
9. Castro IP, Jones JM. Studies in numerical computations of recirculating flows. *International Journal for Numerical Methods in Fluids* 1987; **7**:793–823.
10. Napolitano M, Pascazio G, Quartapelle L. A review of vorticity conditions in the numerical solution of the $\omega - \psi$ equations. *Computers and Fluids* 1999; **28**:139–185.