Computing eigenvalues of ordinary differential equations

D. L. Harrar II[*] M. R. Osborne[†]

(Received 1 June 2001; revised 18 October 2002)

Abstract

Discretisations of differential eigenvalue problems have a sensitivity to perturbations which is asymptotically least as $h \to 0$ when the differential equation is in first order system form. Both second and fourth order accurate discretisations of the first order system are straightforward to derive and lead to generalised eigenvalue problems of the form

$$(A(h) - \lambda B(h))\,\mathbf{v} = 0\,,$$

where both A and B are narrow-banded, block bidiagonal (hence unsymmetric) matrices, and typically B is singular. Solutions of the differential equation associated with eigenvalues of small magnitude are best determined by the discretisations. Thus Krylov subspace methods (for example)

[*] MegaSense Inc., 1215 Bordeaux Drive, Sunneyvale, CA 94089, U.S.A.

[†] Mathematical Sciences Institute, Australian National University, ACT 0200, Australia. mailto:mike.osborne@maths.anu.edu.au

[0] See http://anziamj.austms.org.au/V44/CTAC2001/Harr for this article, © Austral. Mathematical Soc. 2003. Published 1 April 2003. ISSN 1446-8735

require A to be invertible and seek large solutions of

$$\left(\mu I - A^{-1}B\right)\mathbf{v} = 0\,.$$

This already requires rational methods in principle. It follows that rapidly convergent methods based on inverse iteration applied to the original formulation as a nonstandard generalised eigenvalue problem prove attractive for the narrow-banded systems considered here. Also they have the advantage that they are applicable under the weaker condition $A^{\perp} \cap B^{\perp} = \emptyset$. We have had extensive experience with a method combining aspects of Newton's method and inverse iteration and having a convergence rate of 3.56. Our implementation combines this basic algorithm with a limiting form of Weilandt deflation to find a sequence of eigenvalues. It has proved extremely satisfactory in a range of applications. This formulation has the further advantage that it is easy to insert the eigenvalue calculation inside an outer loop to satisfy a constraint on an auxiliary parameter. Examples to illustrate both the robustness of the deflation and the flexibility of the approach are provided.

Contents

1 Introduction

Eigenvalue problems for systems of ordinary differential equations provide a class of problems for which special purpose methods are frequently appropriate:

- Standard discretization methods cannot, in general, provide accurate estimates of highly oscillatory eigensolutions so there is little point in trying to compute them in this way. It is more important to have methods that can probe regions of the spectrum in a reasonably controlled manner.
- The discretisations lead to generalized eigenvalue problems (GEVPs)
  $$\mathbf{s}^*\,(A-\lambda B)=(A-\lambda B)\,\mathbf{u}=0\,, \tag{1}$$
  where the component matrices A and B are narrow-banded and typically have the block bidiagonal form
  $$M=\begin{bmatrix} M_{1,1} & & & & M_{1,k} \\ M_{2,1} & M_{2,2} & & & \\ & M_{3,2} & M_{3,3} & & \\ & & \ddots & \ddots & \\ & & & M_{k,k-1} & M_{k,k} \end{bmatrix}, \tag{2}$$
  and $\mathbf{u}$, $\mathbf{s}$ are the right and left eigenvectors, respectively.
- Frequently the question of interest involves adjusting system parameters so that a particular eigenvalue satisfies a required condition — is purely real or purely imaginary, for example.

- Stability considerations may require that parameter values are such that no eigenvalues lie in a particular region. This region may have to be computed as part of the investigation.

The methods discussed are recommended as appropriate in just these contexts.

We assume that the system is given as a set of first order differential equations:

$$\frac{d\mathbf{v}}{dz} + C(z,\lambda)\mathbf{v} = 0\,, \tag{3}$$

where $C : \mathbb{R}^m \to \mathbb{R}^m$ is a smooth enough matrix function both of the independent variable z and of the eigenvalue parameter λ. Two discretization schemes are employed:

Trapezoidal rule This is a second-order scheme and is derived by integrating (3) between adjacent mesh points using the trapezoidal rule to obtain

$$\left[-I + \frac{h}{2}C_{i-1}\right]\mathbf{v}_{i-1} + \left[I + \frac{h}{2}C_i\right]\mathbf{v}_i = \mathcal{O}\left(h^3\right)\,, \quad i = 2,\ldots,n\,. \tag{4}$$

Collocation/Simpson's rule This is a fourth-order scheme which uses collocation at midpoints of mesh intervals as well as at mesh points [7]

$$\begin{aligned} &\left\{-I + \frac{h}{6}C_{i-1} + \frac{4h}{6}C_{i-\frac{1}{2}}\left[\frac{1}{2}I - \frac{h}{8}C_{i-1}\right]\right\}\mathbf{v}_{i-1} \\ &+ \left\{I + \frac{h}{6}C_i + \frac{4h}{6}C_{i-\frac{1}{2}}\left[\frac{1}{2}I + \frac{h}{8}C_i\right]\right\}\mathbf{v}_i = \mathcal{O}\left(h^5\right)\,, \\ &\text{for } i = 2,\ldots,n\,. \end{aligned} \tag{5}$$

Here $h = L/(n-1)$ is the mesh spacing, n is the number of mesh points, and $(0, L)$ is the interval on which the problem is defined. The mesh spacing is assumed constant, but variable meshes are certainly possible. The subscript indicates evaluation at the corresponding mesh value, $C_j = C\left((j-1)\,h, \lambda\right)$. Note that (5) involves products of C and conceivably this could force a nonlinear dependence on λ. However, this possibility can be ruled out in many cases and does not occur in our examples [3].

The specification of the eigenvalue problem is completed by adding boundary conditions. This is discussed separately for each example considered. The resulting block bidiagonal eigenvalue problem has dimension $nm \times nm$ and bandwidth $2m$. Typical values are $2 \leq m \leq 8$ and $n = \mathcal{O}\left(10^3\right)$. Working with the first-order system form has the important advantage that the sensitivity of the eigenvalue problem, defined as

$$\rho = \frac{\|\mathbf{s}\| \,.\, \|\mathbf{v}\|}{|\mathbf{s}^* B \mathbf{v}|}\,, \tag{6}$$

is typically $\mathcal{O}\left(h^{-1}\right)$ — this is a consequence of the h-dependence in (4) and (5). For higher-order equations it is typically $\mathcal{O}\left(h^{-q}\right)$, where q is the order of the differential equation. It has the disadvantage that B is typically both nonsymmetric and indefinite. Consequently some useful structure may be lost — for example, if the original equation is of Sturm-Liouville type.

The organization of this paper is as follows: In the next section we discuss the solution of generalized eigenvalue problems using inverse iteration, Newton's method, and deflation. In order to implement these techniques efficiently for block bidiagonal problems on vector processors we use wrap-around partitioning [4]. We give examples of how to transform problems in the study of chemical reactions (§3), and in hydrodynamic stability (§4) into block bidiagonal GEVPs using both second- and fourth-order accurate discretisations. The numerical results presented were obtained using the Fujitsu VPP300 at the Australian National University.

2 Newton-based methods for eigenvalue problems

Let $K : \mathbb{C}^n \to \mathbb{C}^n$ and consider the eigenvalue problem

$$K(\lambda)\mathbf{u}(\lambda) = 0\,, \quad K(\lambda) \equiv (A - \lambda B)\,. \tag{7}$$

Although $K(\lambda)$ is assumed to be linear in λ here, the techniques discussed can also be applied in the case that $K(\lambda)$ is nonlinear in λ. The basic idea behind using Newton's method to solve GEVPs is to replace the problem (7) with that of finding zeros of a nonlinear function. To this end, (7) is embedded in the more general family

$$K(\lambda)\mathbf{u}(\lambda) = \beta(\lambda)B\mathbf{x}\,, \tag{8}$$

$$\widetilde{\mathbf{s}}^* B\mathbf{u}(\lambda) = \kappa\,. \tag{9}$$

Then, as λ approaches an eigenvalue, $K(\lambda)$ becomes singular so the solution $\mathbf{u}$ of (8) becomes unbounded for almost all right-hand sides $\beta(\lambda)B\mathbf{x}$ and choices of the scaling vector $\widetilde{\mathbf{s}}$. Hence, (9)—a scaling condition—can be satisfied only if $\beta(\lambda) \to 0$ as λ approaches an eigenvalue. The vectors $\widetilde{\mathbf{s}}$ and $\mathbf{x}$ can be chosen adaptively as the iteration proceeds, and a good strategy is to choose them as the best current estimates of the left and right eigenvectors, respectively. An important advantage facilitated by this freedom is the possibility of exceeding the second-order convergence rate characteristic of procedures based on Newton's method [9].

Differentiating equations (8–9) with respect to λ gives

$$K\frac{d\mathbf{u}}{d\lambda}+\frac{dK}{d\lambda}\mathbf{u}=\frac{d\beta}{d\lambda}B\mathbf{x}\,, \tag{10}$$

$$\widetilde{\mathbf{s}}^*B\frac{d\mathbf{u}}{d\lambda}=0\,. \tag{11}$$

Multiplying through on the left by $\widetilde{\mathbf{s}}^*BK^{-1}$, (10) becomes

$$\begin{aligned}\widetilde{\mathbf{s}}^*B\frac{d\mathbf{u}}{d\lambda}+\widetilde{\mathbf{s}}^*BK^{-1}\frac{dK}{d\lambda}\mathbf{u}&=\frac{d\beta}{d\lambda}\widetilde{\mathbf{s}}^*BK^{-1}B\mathbf{x}\\&=\frac{1}{\beta(\lambda)}\frac{d\beta}{d\lambda}\widetilde{\mathbf{s}}^*B\mathbf{u}\,,\end{aligned}$$

since $K^{-1}B\mathbf{x}=\beta(\lambda)^{-1}\mathbf{u}$ by (8). The first term on the left drops out by (11), and, solving for the Newton correction, the Newton iteration takes the form

$$\lambda\longleftarrow\lambda+\Delta\lambda\,,\quad \Delta\lambda=-\frac{\beta(\lambda)}{d\beta/d\lambda}=\frac{\widetilde{\mathbf{s}}^*B\mathbf{u}}{\widetilde{\mathbf{s}}^*BK^{-1}B\mathbf{u}}\,,$$

where we have made the substitution $dK/d\lambda=-B$. The main computational component is essentially inverse iteration with the matrix K. Convergence rates, including conditions under which third-order convergence is possible, are discussed in [8]. This requires that the iteration function be redefined at each step by updating $\mathbf{x}$ so that it corresponds to the best current estimate of the right eigenvector

$$\mathbf{x}\leftarrow K^{-1}B\mathbf{u}/\|K^{-1}B\mathbf{u}\|\,.$$

A more recent reference is [9] in which the development is in terms of generalized, rather than standard, eigenvalue problems. Here it is shown that a convergence rate of 3.56 is obtained if the iteration is organised to simultaneously compute a rapidly convergent estimate of the left eigenvector to use as the scale vector:

$$\begin{aligned} \mathbf{w} &= K^{-T} B^T \widetilde{\mathbf{s}} \\ \widetilde{\mathbf{s}} &\leftarrow \mathbf{w}/\|\mathbf{w}\| \,. \end{aligned}$$

Also discussed in this reference are deflation techniques. These are now sketched in outline.

2.1 Deflation

A suitable approach has proved to be a limiting case of multiplicative Wielandt deflation [10, e.g.]. This involves working with the matrix

$$\hat{A} = (I - \sigma B \mathbf{v}_i \mathbf{s}_i^*) A \,,$$

where $\mathbf{v}_i$ and $\mathbf{s}_i$ are the right and left eigenvectors, normalized so that $\mathbf{s}_i^* B \mathbf{v}_i = 1$, corresponding to the eigenvalue λ_i . This maps $\lambda_i \rightarrow (1-\sigma)\lambda_i$ while leaving the remaining eigenvalues unchanged. Note that this form of algorithm sits well with accelerated forms of the Newton algorithm which compute estimates of the right and left eigenvectors as an integral part of the computation. The basic iteration (8) is replaced by

$$\hat{K}(\lambda)\mathbf{u}(\lambda) = \beta(\lambda) B \mathbf{x} \,, \quad \hat{K}(\lambda) \equiv (\hat{A} - \lambda B) \,.$$

Multiplying through by

$$(I - \sigma B \mathbf{v}_i \mathbf{s}_i^*)^{-1} = \left(I + \frac{\sigma}{1-\sigma} B \mathbf{v}_i \mathbf{s}_i^* \right) \,,$$

(here we have assumed, without loss of generality, that $\kappa = 1$ in (9)) it is apparent that the deflated iteration is equivalent to

$$\left(A - \lambda \left[I + \frac{\sigma}{1-\sigma} B \mathbf{v}_i \mathbf{s}_i^* \right] B \right) \mathbf{v} = \beta(\lambda) \left[I + \frac{\sigma}{1-\sigma} B \mathbf{v}_i \mathbf{s}_i^* \right] B \mathbf{x} \,.$$

So, letting $\sigma \to \infty$ effectively moves $\lambda_i \to \pm\infty$, and, if we define the (non-orthogonal) projection

$$P_i \equiv (I - B\mathbf{v}_i\mathbf{s}_i^*)\,,$$

then as $\sigma \to \infty$ the deflated iteration becomes

$$(A - \lambda P_i B)\mathbf{v} = \beta(\lambda) P_i B\mathbf{x}\,,$$

and the solution $\mathbf{v}$ of this equation is the same as that of

$$(A - \lambda B)\mathbf{v} = \beta(\lambda) P_i B\mathbf{x}\,.$$

In this form, multiplicative Wielandt deflation preserves the block bidiagonal structure. Maintenance of special structure is, of course, highly desirable for banded systems in general. This technique will be recognised as repeated biorthogonalisation with respect to already computed eigenvectors. It has proved very satisfactory even in quite severe tests. For example, eighty eigenvalues have been found for the discretized Brusselator equation (Section 3). Here the matrix was 14000×14000, and the computation of each eigenvalue was started from a small random perturbation about $\lambda = 0$. This requires that each already computed eigenvalue must be bypassed in order to reach the next in sequence. As Gram-Schmidt orthogonalisation is notoriously unstable when applied to produce a set of orthogonal vectors it is worth pointing out that this is not how it is used here. Here it is required to remove components of the already computed vectors from the inverse iteration right-hand sides, and this the Gram-Schmidt method does adequately. Problems can only arise in the initial stage of an iteration through catastrophic cancellation which could “reflate” unwanted components as a result of rounding error. We have found the use of randomly generated vectors satisfactory for initialising each iteration.

3 Chemical reactions in a tubular reactor

A frequently used test problem for eigensolvers is solution of the “Brusselator model” of the Belousov-Zhabotinski reaction in a tubular reactor. The model comprises a system of two coupled diffusion equations,

$$\begin{aligned}\frac{\partial\phi}{\partial t} &= \frac{\nu_\phi}{L^2}\frac{\partial^2\phi}{\partial z^2} + \alpha - (\beta+1)\phi + \phi^2\psi\,,\\ \frac{\partial\psi}{\partial t} &= \frac{\nu_\psi}{L^2}\frac{\partial^2\psi}{\partial z^2} + \beta\phi - \phi^2\psi\,,\end{aligned}$$

where $z \in [0,1]$ is the distance along the tube, ϕ and ψ are the chemical concentrations of the two reactants, ν_ϕ/L and ν_ψ/L are diffusion coefficients, and α and β arise from the boundary conditions:

$$\phi(t,0) = \phi(t,1) = \alpha\,, \quad \psi(t,0) = \psi(t,1) = \beta/\alpha\,.$$

Investigation of the linear stability of this system at the stationary solution $(\phi,\psi) = (\alpha,\beta/\alpha)$ leads to a coupled ODE eigenvalue problem,

$$\begin{aligned}\frac{\nu_\phi}{L^2}\frac{d^2\phi}{dz^2} + (\beta-1)\phi + \alpha^2\psi &= \lambda\phi\,,\\ \frac{\nu_\psi}{L^2}\frac{d^2\psi}{dz^2} - \beta\phi - \alpha^2\psi &= \lambda\psi\,,\end{aligned}$$

and corresponding boundary conditions given by

$$\phi(0) = \phi(1) = 0\,, \quad \psi(0) = \psi(1) = 0\,.$$

Let

$$\omega_\phi \equiv \frac{\sqrt{\nu_\phi}}{L}\,, \quad \omega_\psi \equiv \frac{\sqrt{\nu_\psi}}{L}\,,$$

and set

$$u = \omega_\phi \frac{d\phi}{dz}\,, \quad v = \omega_\psi \frac{d\psi}{dz}\,,$$

then this is written as the first-order system

$$D\frac{d\mathbf{v}}{dz} + C\mathbf{v} = 0\,, \tag{12}$$

where

$$\mathbf{v} = \begin{bmatrix} \phi \\ \psi \\ u \\ v \end{bmatrix}, \quad D = \begin{bmatrix} \omega_\phi & & & \\ & \omega_\psi & & \\ & & \omega_\phi & \\ & & & \omega_\psi \end{bmatrix},$$

$$C = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \\ (\beta-1)-\lambda & \alpha^2 & 0 & 0 \\ -\beta & -\alpha^2-\lambda & 0 & 0 \end{bmatrix}.$$

Second and fourth order discretisations have been implemented, and details can be found in [3]. The matrices of the resulting GEVPs are $4n \times 4n$. In the second order case the form of B ensures that at most $2n-2$ distinct eigenvalues can be found, and this is the number that would be obtained by solving the eigenvalue problem obtained by direct second-order discretization of the higher-order system. In the fourth order discretization B drops rank only in the equations corresponding to the boundary conditions with the result that the GEVP possibly has as many as $4n-4$ distinct eigenvalues. However, the difference between the two forms of B is only $\mathcal{O}\left(h^2\right)$, so the additional eigenvalues depend on a form of singular perturbation of relative magnitude $\mathcal{O}(h)$. It has been reported that the spurious eigenvalues tend to ∞ with h^{-1} as $h \to 0$ in a related application [11].

3.1 Numerical results

In this case the generalized eigenvalue problem is actually a two parameter problem having the form

$$\mathbf{s}^* M(\lambda, L) = M(\lambda, L)\,\mathbf{v} = 0\,. \tag{13}$$

where $\mathbf{s}$ and $\mathbf{v}$ are the left and right eigenvectors respectively. Two different types of problem are considered:

1. Determine the value of L such that the right-most conjugate eigenvalue pair is purely imaginary; this corresponds to a Hopf bifurcation and signals the onset of periodic behavior. Newton-based procedures can be used to compute the bifurcation parameter L. This computation proves to sit nicely with our eigenvalue solver and hence provides a good illustration of the flexibility of our approach.

2. The Brusselator example has proved a very popular test problem for eigenvalue codes. The basic idea is to take the value of L associated with the Hopf bifurcation and to compute, say, the twelve eigenvalues nearest zero. The eigenvalues are not hard to compute and give small values of the sensitivity parameter (6), but the resulting plot of the spectrum is most attractive (Figure 1). As noted in Section 2.1 it is not a bad test of the deflation procedure when carried out in the manner indicated. Of course, in a practical computation it would be much more satisfactory to start the computation of each new eigenvalue beginning from a perturbation about the previous eigenvalue computed.

The Newton correction to a guess at the bifurcation parameter L is

$$L \leftarrow L - \frac{\Re(\lambda)}{\Re(d\lambda/dL)}$$

where $\lambda = \lambda(L)$ is obtained by solving the eigenvalue problem (13) for λ, $\mathbf{v}$, $\mathbf{s}$. Note that our approach permits the current solution to the eigenvalue problem to be used as a first approximation for the next iteration. Calculation of the Newton correction requires a knowledge of $d\lambda/dL$ which is obtained by differentiating the eigenvalue problem. This gives

$$\left(\frac{\partial M}{\partial L} + \frac{d\lambda}{dL}\frac{\partial M}{\partial \lambda}\right)\mathbf{v} + M\frac{d\mathbf{v}}{dL} = 0\,.$$

Multiplying by the left vector gives

$$\frac{d\lambda}{dL} = -\frac{\mathbf{s}^*\frac{\partial M}{\partial L}\mathbf{v}}{\mathbf{s}^*\frac{\partial M}{\partial \lambda}\mathbf{v}}\,.$$

The parameter values used are taken from [10]:

$$\nu_x = 0.0008\,, \quad \nu_y = \frac{1}{2}\nu_x\,, \quad \alpha = 2\,, \quad \beta = 5.45\,.$$

Our computations give $L \approx 0.51302$ in agreement with [10].

Results for the second set of calculations are summarized in Figure 1, where a portion of the spectrum is shown, and the twelve eigenvalues of interest shown as solid circles. The results are identical to those obtained using, for example, the block Arnoldi method reported on in [2].

4 Hydrodynamic stability

The next application considered here is the computation of the neutral curve for the Orr-Sommerfeld equation. This equation is of considerable importance in studies of transition to turbulence and

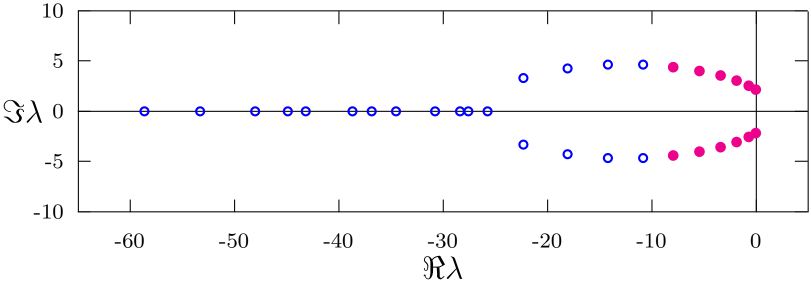

Figure 1: Right-most eigenvalues for the Brusselator model.

has been used to predict instability in a variety of fluid flow configurations including boundary layers, jets, and shear layers, as well as other flows on unbounded domains; for a survey see [5]. Each of these types of flows is amenable to solution using the computational techniques presented here, but we restrict our attention to two examples for which the respective velocity profiles are quite different: Poiseuille flow between parallel flat plates and Blasius (boundary layer) flow over a flat plate.

The Orr-Sommerfeld equation has the form of a fourth-order ODE eigenvalue problem,

$$\frac{i}{\alpha R}\left\{\frac{d^2}{dz^2}-\alpha^2\right\}^2\phi+(U(z)-\lambda)\left\{\frac{d^2}{dz^2}-\alpha^2\right\}\phi-\frac{d^2U}{dz^2}\phi=0\,, \quad (14)$$

where $U(z)$ is the velocity profile for the flow to be investigated, α is the wave number, R is the Reynold's number, and the eigenvalue parameter λ is the wave speed. This differential equation is complex and hence our numerical solution techniques for this problem have used complex arithmetic throughout.

The ODE (14) can be written as a system of four first-order ODEs in a variety of ways. Noting that $\frac{d^2}{dz^2}-\alpha^2=(\frac{d}{dz}-\alpha)(\frac{d}{dz}+\alpha)$, the first-order system is formulated as follows:[1]

$$\frac{d\phi}{dz} + \alpha\phi - u = 0\,, \tag{15}$$

$$\frac{du}{dz} - \alpha u - v = 0\,, \tag{16}$$

$$\sqrt{\frac{i}{\alpha R}}\left(\frac{dv}{dz} + \alpha v\right) - w = 0\,, \tag{17}$$

$$\sqrt{\frac{i}{\alpha R}}\left(\frac{dw}{dz} - \alpha w\right) - \frac{d^2U}{dz^2}\phi + U(z)v = \lambda v\,. \tag{18}$$

The first problem we consider is that of Poiseuille flow which models the flow of a fluid between infinite parallel plates. The Poiseuille velocity profile is

$$U(z) = 1 - z^2\,, \quad -1 \le z \le 1\,, \tag{19}$$

so that the coefficient for ϕ in the last term of (14) is constant:

$$\frac{d^2U}{dz^2} = -2\,.$$

The boundary conditions in this case are

$$\phi = \frac{d\phi}{dz} = 0\,,$$

at the surfaces of the plates. The second problem we consider is associated with transition to turbulence in flow over a flat plate. Strictly speaking it is a similar [1, e.g.] rather than a parallel flow, and this

[1] Here we have used a factor of $\sqrt{i/(\alpha R)}$ in the last two equations, rather than a factor of $i/(\alpha R)$ in the last only, solely due to numerical stability considerations — this artifice should in some cases result in a smaller range of values in matrix elements.

has practical implications in terms of problem scaling as well as raising theoretical questions. The Blasius velocity profile $U(z)$ is assumed to have the form

$$U(z) = f'(\eta) \tag{20}$$

where η is the similarity variable, and f satisfies the Blasius differential equation

$$f''' + ff'' = 0\,, \tag{21}$$

subject to the boundary conditions

$$f\,(0) = f'\,(0) = 0\,, \quad f'\,(\infty) = 1\,,$$

where the prime indicates differentiation with respect to η. Values of f, f', and f'' are obtained by solving the Blasius equation (21) using a fourth-order Runge-Kutta scheme in our computations. The boundary conditions in the case of boundary layer flow are taken to be

$$\phi(0) = \left.\frac{d\phi}{dz}\right|_{z=0} = 0\,, \quad \phi \sim e^{-\alpha z}\left(A + \frac{B}{z} + \frac{C}{z^2} + \cdots\right)\,, \quad z \to \infty\,.$$

Evaluating the asymptotic condition at $z = L$ and ignoring terms that are $\mathcal{O}\left(e^{-\alpha L}/L^3\right)$ we obtain terminal boundary conditions

$$2\alpha\frac{d\phi}{dz} + \frac{d^2\phi}{dz^2} = 0\,, \tag{22}$$

$$\frac{d^3\phi}{dz^3} = 0\,. \tag{23}$$

The setting up of the discretized equations is discussed in detail in [3]. The weight matrix B is distinctly sparse in the second order case, and the resulting GEVP has at most $n-1$ distinct eigenvalues. This agrees with the number that would be obtained by direct discretization of the Orr-Sommerfeld equation. Again there is a relation of singular perturbation type between the spectra derived from the second and fourth order discretisations, with a richer spectrum (at most $2n-2$ eigenvalues) being obtained for the fourth-order discretization for all finite h. The difference between the B matrices in the two cases is again $\mathcal{O}(h^2)$.

4.1 Numerical results

One goal is to compute the "neutral curve", or curve of neutral stability, which separates the stable and unstable regions in the (α, R)-plane. Computationally, this is the locus of points in the (α, R)-plane with vanishing imaginary part:

$$\Im\left\{\lambda(\alpha, R)\right\} = 0\,.$$

(α, R)-regions outside the neutral curve correspond to combinations of α and R for which the flow is stable, while those within the neutral curve correspond to unstable flow. Newton's method is used to find a zero of $\Im\left\{\lambda\right\}$ either as a function of α or of R using

$$\frac{\partial\lambda}{\partial\alpha} = \frac{s^*\frac{\partial A}{\partial\alpha}v}{s^*Bv}\,, \qquad \frac{\partial\lambda}{\partial R} = \frac{s^*\frac{\partial A}{\partial R}v}{s^*Bv}\,.$$

To decide which of α or R to vary, a comparison of the magnitude of the respective derivatives is made; for example, R is varied if it passes the scale-invariant test,

$$\left|R_0\frac{\partial\lambda}{\partial R}\right| \geq \left|\alpha_0\frac{\partial\lambda}{\partial\alpha}\right| \tag{24}$$

where α_0 and R_0 are typical parameter values. To move on the neutral curve, the alternate variable is incremented by a fixed amount

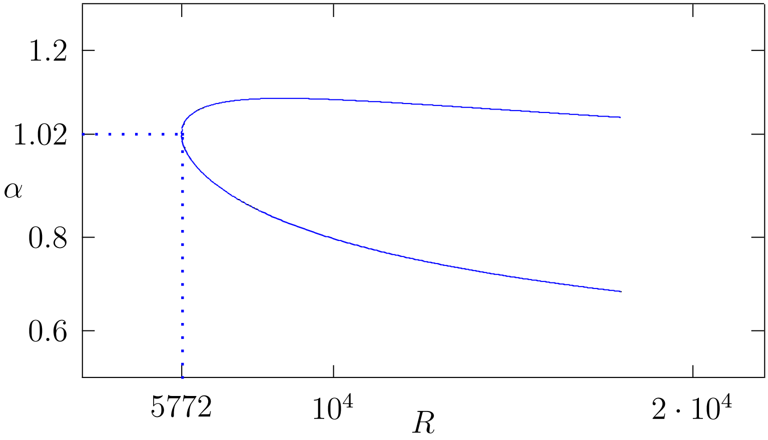

Figure 2: Neutral stability curve for Poiseuille flow.

(due consideration must be given to the correct choice of sign) and the computation sequence restarted. The advantage of the 3.56 rate of convergence is very much in evidence — no more than two iterations are required at any stage in order to compute the next λ.

In Figures 2 and 3 we show the neutral curves for the Poiseuille velocity profile (19) and for the Blasius profile (20–21), respectively. Computations were carried out in parallel on three processors of the vpp300, each one computing different portions of the curves (top, bottom, left). All results used 5000 finite difference grid points, corresponding to complex banded matrices of dimension $n = 20,000$. The solution of the block bidiagonal linear systems achieved about 92% vectorization for $n = 5000$. We ran experiments using up to 750,000 points, that is, complex matrices of order $n = 3,000,000$, for which a (complex) matrix factorization takes about ten seconds (in shared mode) and achieves 99.6% vectorization.

An interesting feature of the Poiseuille computation is that somewhat large numbers of grid points are required before the trapezoidal

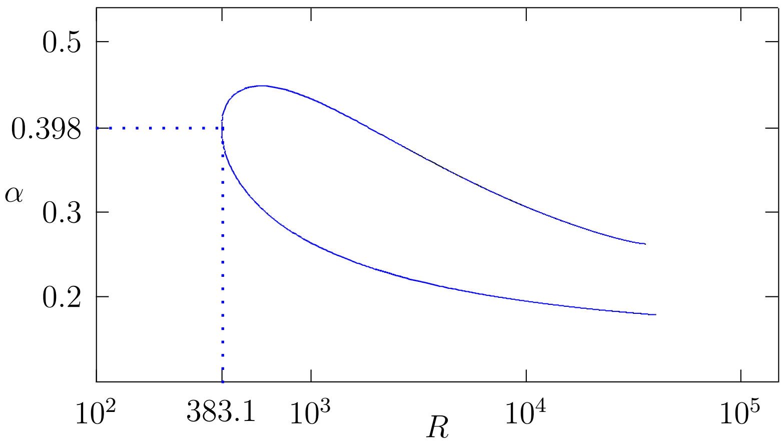

Figure 3: Neutral stability curve for Blasius flow (logarithmic scale).

rule gives adequate accuracy. This is an indication that the differential equation is approaching a singular perturbation problem for the parameter values of interest (also that the limiting equation is a singular differential equation). Experiments have been initiated using the higher-order differencing scheme (5), and it proves useful in the case of large R. The value of the Reynold's number at the turning point, $R = 5772$, is to four significant digits the value given in [1] and has been confirmed experimentally [6].

To conclude, we illustrate the 3.56 rate of convergence of the Newton-based eigensolver by showing in Table 1 the convergence history for the beginning of the computation of the neutral curve for Blasius boundary layer flow. Tolerance for the computation of λ is 10^{-12} and for R or α is 10^{-6}. For this choice of initial data it is R which is updated at each step according to the test (24). The 3.56 rate of convergence of the Newton procedures is evident.

Table 1: Convergence history in computing a first point on the neutral curve for Blasius boundary layer flow using an initial guess of $\{\alpha_0, R_0\} = \{0.27, 1000.0\}$.

Iter	$\Im\{\lambda\}$	$\Im\{\Delta\lambda\}$
0	$R = 1000.0$	$\Delta R = 0.0$
1	2.2796675864805e-03	1.0296675864805e-03
2	2.2818890819293e-03	2.2214954488240e-06
3	2.2818890819283e-03	-9.837883193987e-16
*	$R = 892.335200$	$\Delta R = 107.66480$
1	-2.222756013814e-04	-2.504164683309e-03
2	-2.222756013806e-04	8.2926469548662e-16
*	$R = 901.048651$	$\Delta R = -8.713450$
1	-1.708633408248e-06	2.2056696797234e-04
2	-1.708633413609e-06	-5.3607273978914e-15
*	$R = 901.116680$	$\Delta R = -0.068030$
1	-6.756906093322e-12	1.70862665677034e-06
2	-6.759492083016e-12	-2.5859896939883e-15
*	$R = 901.116681$	$\Delta R = 0.$

References

[1] D. J. Acheson, *Elementary Fluid Dynamics*, Oxford Applied Mathematics and Computing Science Series, Clarendon Press, Oxford, U.K., 1992, Second edition. C327, C331

[2] D. L. Harrar II, *A shift-invert block Arnoldi method with deflation and adaptive parameter selection*, Tech. Report MRR.033.99, Mathematics Department, Australian National University, 1999. http://wwwmaths.anu.edu.au/research.reports/mrr/99.033/MRR99-033.ps.gz C325

[3] D. L. Harrar II and M. R. Osborne, *Computational techniques for differential equation eigenvalue problems on vector processors*, Mathematics Research Report No. MRR99-032, Australian National University. http://wwwmaths.anu.edu.au/research.reports/mrr/99.032/MRR99-032.ps.gz C317, C323, C328

[4] M. Hegland and M. R. Osborne, *Wrap-around partitioning for block bidiagonal systems*, IMA J. Num. Anal. **18** (1998), 373–383. C317

[5] I. H. Herron, *The Orr-Sommerfeld equation on infinite intervals*, SIAM Review **29** (1987), no. 4, 597–620. C326

[6] M. Nishioka, S. Iida, and Y. Ichikawa, J. Fluid Mech. **72** (1975), 731–751. C331

[7] M. R. Osborne, *A method for finite-difference approximation to ordinary differential equations*, Computer J. **7(1)** (1964), 58–65. C316

[8] M. R. Osborne, *Inverse iteration, Newton's method, and nonlinear eigenvalue problems*, The Contributions of J. H.

Wilkinson to Numerical Analysis, Symposium Proc. Series No. 19, The Inst. for Math. and its Appl., 1979. C319

[9] M. R. Osborne and D. L. Harrar II, *Inverse iteration and deflation in general eigenvalue problems*, Mathematics Research Report No. MRR97- 012, Australian National University. `http://wwwmaths.anu.edu.au/research.reports/mrr/97.012/MRR97-012.dvi.gz` C318, C319

[10] Y. Saad, *Numerical Methods for Large Eigenvalue Problems*, Manchester University Press (Series in Algorithms and Architectures for Advanced Scientific Computing), Manchester, 1992. C320, C325

[11] W. Sun, 1997, Private communication. C323