BIFURCATION ANALYSIS FOR TIMESTEPPERS

LAURETTE S. TUCKERMAN* AND DWIGHT BARKLEY†

Abstract. A collection of methods is presented to adapt a pre-existing time-stepping code to perform various bifurcation-theoretic tasks. It is shown that the implicit linear step of a time-stepping code can serve as a highly effective preconditioner for solving linear systems involving the full Jacobian via conjugate gradient iteration. The methods presented for steady-state solving, continuation, direct calculation of bifurcation points (all via Newton's method), and linear stability analysis (via the inverse power method) rely on this preconditioning. Another set of methods can have as their basis any time-stepping method. These perform various types of stability analyses: linear stability analysis via the exponential power method, Floquet stability analysis of a limit cycle, and nonlinear stability analysis for determining the character of a bifurcation. All of the methods presented require minimal changes to the time-stepping code.

Key words. bifurcation analysis, continuation, Stokes preconditioning, Newton's method, Arnoldi's method.

AMS(MOS) subject classifications. 35B32, 58F39, 65F10, 65F50, 65H10, 65H17, 65Mxx, 76Exx.

1. Introduction.

The goal of the dynamical systems approach to time-evolution equations is a full conceptual picture resting on such basic building blocks as steady states, limit cycles, and bifurcations. For partial differential equations, constructing such a picture is a formidable challenge, whether the approach is analytic or numerical. From a numerical perspective, the spatial discretization of fields results in a high-dimensional phase space. The vectors representing phase-space points are then so large that the matrix operations required by standard dynamical systems algorithms are prohibitively expensive. (See [21] for a survey and references.) In addition, the basic interface for communication with dynamical systems software is generally the right-hand-side of the evolution equation. For partial differential equations, this function may not be immediately available due to constraints such as incompressibility and boundary conditions.

Instead, the basic numerical tool for studying partial differential equations such as the Navier-Stokes or reaction-diffusion equations has usually been temporal integration, or time-stepping. Since time-stepping codes can represent a considerable investment – on the order of a few years for development and verification – it is desirable to be able adapt a time-stepping code to carry out bifurcation analysis.

Any timestepping scheme can already be, and often is, used for bifurcation analysis without further modification. Integration can proceed until a stable steady state is reached, or a control parameter gradually increased until a transition takes place, indicating a bifurcation. However, these techniques rely on waiting out slow exponential decay, or on arduous binary searches, and make highly ineffi-

*LIMSI, B.P. 133, 91403 Orsay Cedex France. Email: laurette@limsi.fr

†Mathematics Institute, University of Warwick, Coventry CV4 7AL, United Kingdom. Email: barkley@maths.warwick.ac.uk

cient use of both machine and human resources. Other tasks cannot be carried out using time-integration at all: time-stepping will never converge to an unstable steady state nor to an eigenvector. Rapid algorithms have been developed for obtaining steady states, bifurcation points, and eigenvectors directly, rather than as by-products of time-integration. Our goal is to adapt time-stepping codes to perform these algorithms efficiently.

We consider the Navier-Stokes equation in the following symbolic form:

$$\partial_t U = -(U\cdot\nabla)U - \nabla P + \frac{1}{R}\nabla^2 U \tag{1.1a}$$

$$= -(I - \nabla\nabla^{-2}\nabla\cdot)(U\cdot\nabla)U + \frac{1}{R}\nabla^2 U \tag{1.1b}$$

$$= \Pi(U\cdot\nabla)U + \frac{1}{R}\nabla^2 U \tag{1.1c}$$

$$= N(U) + LU \tag{1.1d}$$

In equation (1.1b), we have set the pressure P to the solution of a Poisson equation whose right-hand-side is the divergence of the nonlinear inertial term. The boundary conditions for this Poisson equation are chosen and imposed in various ways, depending on the particular physical problem and numerical method; these features are summarized by the projection operator Π of (1.1c). R represents the Reynolds number. Equation (1.1d) serves to define N as the nonlinear operator corresponding to the combined inertial and pressure terms, and L as the linear operator corresponding to viscous diffusion. Formulation (1.1) can be generalized to include additional fields, such as temperature, by including the corresponding balance equations, although we continue to refer to U as the velocity.

The velocity field U, and hence the operators L and N, are spatially discretized according to the numerical method chosen. We denote the size of U by M. M is the number of spatial gridpoints (x, y, z) multiplied by the number of fields (U_x, U_y, U_z, temperature, etc.) and so can be quite large, on the order of 1000–50000. We shall use U to denote both the continuous fields appearing in the partial-differential equations, e.g. (1.1d), and also a single vector of length M of discretized values.

It is a fortunate feature of the Navier-Stokes equations and of many reaction-diffusion equations that the fastest timescales in the system arise from the linear operator L. Hence it is L which poses the severest constraint on numerical timestepping. Stability restrictions on the timestep arising from the linear operator are overcome by treating the linear term implicitly, leaving the nonlinear term to be integrated by an easier, explicit method. The simplest implicit/explicit scheme is first-order Euler timestepping:

$$\begin{aligned} U(t+\Delta t) &= U(t) + \Delta t\,[N(U(t)) + LU(t+\Delta t)] \\ &= (I - \Delta t L)^{-1}(I + \Delta t N)U(t) \end{aligned} \tag{1.2}$$

We will assume that a computer program for time stepping by this method is available for the equations or applications of interest. (It is generally easy to

transform a higher-order timestepping scheme, such as Adams-Bashforth, to a first-order Euler scheme.) In equation (1.2), the operator $(I - \Delta t L)^{-1}$ is not computed as a matrix inverse; most spatial discretization methods include tricks for acting with $(I - \Delta t L)^{-1}$ economically. It is precisely these tricks which we wish to exploit.

Most bifurcation-theoretic tasks are specified in terms of the Jacobian $N_U + L$ of $N+L$, i.e. for the discretized system the matrix of partial derivatives $\partial(NU+LU)_i/\partial U_j$ where $1 \leq i, j \leq M$. Although $N_U + L$ is defined formally as an $M \times M$ matrix, we emphasize that we never intend for it to be constructed explicitly; our methods are all *matrix-free*. Instead, we require only the action of $N_U + L$ on a vector u. In the case of the Navier-Stokes equations, the action of the operator $N_U + L$ on a vector u is obtained from (1.1) by replacing $(U \cdot \nabla)U$ by $(U \cdot \nabla)u + (u \cdot \nabla)U$. Similar replacements lead to the linearization of other nonlinear terms such as the advection of temperature $U \cdot \nabla T$. It is straightforward to adapt the timestepping algorithm (1.2) to carry out timestepping of the linearized system, i.e.

$$u(t + \Delta t) = (I - \Delta t L)^{-1}(I + \Delta t N_U)u(t) \tag{1.3}$$

2. Steady-state solving. Steady states are solutions to:

$$N(U) + LU = 0 \tag{2.1}$$

Equations of type (2.1) are solved by Newton's method. One Newton step for (2.1) is:

$$\begin{aligned} (N_U + L)u &= (N + L)U \\ U &\longleftarrow U - u \end{aligned} \tag{2.2}$$

U is the current estimate for the steady state, and u is a decrement whose subtraction from U would yield an exact solution if (2.1) were linear.

Although $N_U + L$ may be full (depending on the choice of spatial discretization method) in the sense that most of its elements are non-zero, it should be considered as sparse in the sense that acting with $N_U + L$ on a vector u requires far fewer than the M^2 operations required to multiply by an arbitrary $M \times M$ matrix. No matter what spatial discretization is used, $N_U + L$ will have some kind of regular structure.

If $N_U + L$ is too large to be stored (M^2 words), it certainly cannot be inverted or factored directly (operation count $O(M^3)$). Conjugate gradient iteration, or one of its variants for matrices which are not symmetric positive definite, is the method of choice for solving sparse linear systems [18]. (We will use the name conjugate gradient iteration to mean any of these variants.) Conjugate gradient algorithms can be written in matrix-free implementations, where only the action of the matrix on a vector is required, rather than the individual matrix elements. However, $(N_U + L)$ is poorly conditioned meaning, roughly, that it has a large

range of eigenvalues. Hence, iterative solution via conjugate-gradient type methods converges slowly. A poorly conditioned linear system will require $O(M)$ matrix-vector multiplications to converge, or may not even converge at all.

The remedy for this slow convergence is *preconditioning*, i.e. multiplication of both sides of (2.2) by a matrix which is an approximate inverse of $N_U + L$. Recall from section 1 that L is responsible for the large range of timescales in the temporal evolution (1.1). For the same reason, L is the primary cause of the poor conditioning of $N_U + L$. However, the implicit timestepping of (1.2) used to alleviate this difficulty in time integration also provides a ready preconditioner for the steady state problem.

We multiply both sides of (2.2) by the operator $(I - \Delta t L)^{-1}\Delta t$ and perform some formal algebraic manipulations:

$$\begin{aligned}
&(I - \Delta t L)^{-1}(N_U + L)u = (I - \Delta t L)^{-1}\Delta t(N(U) + LU) \\
&(I - \Delta t L)^{-1}[I + \Delta t N_U - (I - \Delta t L)]u = \\
&\qquad\qquad (I - \Delta t L)^{-1}[I + \Delta t N(U) - (I - \Delta t L)]U \\
&[(I - \Delta t L)^{-1}(I + \Delta t N_U) - I]u = \\
&\qquad\qquad [(I - \Delta t L)^{-1}(I + \Delta t N(U)) - I]U
\end{aligned} \tag{2.3}$$

The key point is that the right-hand-side of (2.3) is the action of the time-stepping operator (1.2) minus the identity, i.e. the difference between consecutive timesteps, and thus easily constructed using the existing timestepping code. More importantly, the left-hand-side of (2.3) is the action of the linearized time-stepping operator (1.3) minus the identity, i.e. the difference between two consecutive *linearized* timesteps.

Note that, unlike the time-stepping scheme (1.2), whose validity in approximating the solution to the differential equation (1.1d) is limited to $\Delta t \ll 1$, the derivation of (2.3) does not depend at all on the size of Δt. The replacement of (2.2) by (2.3) is legitimate for all Δt, no matter how large. The criterion to be used in choosing the value of Δt in (2.3) is exclusively that of efficiency: how fast does conjugate gradient iteration on (2.3) converge? Empirically, for the problems we have investigated, we have found that the fastest convergence is achieved for a Δt which is 10-1000 times the Δt used for timestepping. Heuristically, we reason that, since L is the source of the most widely spaced eigenvalues of $N_U + L$, we seek a preconditioner that resembles L^{-1}. Because $(I - \Delta t L)^{-1}$ is the time-stepping operator for Stokes flow (where there is no nonlinear term), we call this technique Stokes preconditioning.

We have used the BCGS (Bi-Conjugate Gradient Squared) algorithm implemented in the NSPCG (Non-Symmetric Preconditioned Conjugate Gradient) software package [18] For cases which we have studied, the solution of (2.3) has taken on the order of 30-60 iterations, far fewer than our M of 5000-10000. Three to five Newton steps usually suffice to converge to a steady state. Otherwise, a better initial guess, i.e. closer to a previously computed steady state, is usually necessary.

This method has been used to calculate steady states in spherical Couette flow [15] and in a wide variety of convective flows: buoyancy-driven [2, 9, 22, 23, 24, 25] and capillary-driven [7, 9], in rectangular [7, 25] and axisymmetric [2, 9, 22, 23, 24] geometries, with vertical [2, 7, 22, 23, 24] and horizontal [9, 25] gradients, in a magnetic field [22], and in a binary fluid [7, 25].

3. Continuation. The most common steady bifurcations are saddle-nodes, also called turning points. At a saddle-node bifurcation, the solution vector U ceases to be a function of the control parameter R. However, there remains a single smooth curve of solutions (U, R). Near a saddle-point bifurcation, $(\bar{U}, R)$ is a function of U_m, where U_m is some typical component of U and $\bar{U}$ is the vector consisting of all components of U except U_m; see, e.g., [13, 21]. Techniques which calculate steady states along a branch, recognizing and adapting to saddle-node bifurcations, are called continuation techniques. These techniques include strategies for extrapolating to predict the new steady state, for choosing the step-size in R or in U_m, for backtracking along a branch when Newton iteration does not succeed, and for determining when a bifurcation is imminent. For discussions of these important issues, see, e.g. , [21].

The method described in section 2 for finding steady states can be adapted to continuation. We rewrite our schematic equations (1.1) so that R multiplies the nonlinear term. The system to be solved for (U, R) is:

$$0 = RN(U) + LU \tag{3.1a}$$

$$0 = p(U, R) - p^* \text{ where } p(U, R) \equiv \begin{cases} U_m & \text{near a saddle-node} \\ R & \text{otherwise} \end{cases} \tag{3.1b}$$

In practice, we diagnose an imminent saddle-node bifurcation by detecting that one component U_m (when appropriately weighted) begins to change more quickly along the branch than R. This component U_m is then fixed. Specifying a solution to (3.1a) by appending an equation of type (3.1b) is called local [21] or natural [13] parametrization. The projection $p(U, R)$ in (3.1b) serves to distinguish whether it is U_m or R that is to be fixed. For the Navier-Stokes equations, the usual control parameter R is the Reynolds number. For convection, R is the Rayleigh number. The prescribed value p^* depends on which of U_m or R is fixed.

Substituting $(U - u, R - r)$ for (U, R) in (3.1a), and expanding to first order we obtain the linear system to be solved for the decrements (u, r), either:

$$\begin{bmatrix} RN_U + L & N(U) \\ \dots 0 \dots & 1 \end{bmatrix} \begin{bmatrix} u \\ r \end{bmatrix} = \begin{bmatrix} RN(U) + LU \\ R - p^* \end{bmatrix} \tag{3.2a}$$

or near a saddle-node:

$$\begin{bmatrix} RN_U + L & N(U) \\ \dots 1 \dots & 0 \end{bmatrix} \begin{bmatrix} u \\ r \end{bmatrix} = \begin{bmatrix} RN(U) + LU \\ U_m - p^* \end{bmatrix} \tag{3.2b}$$

In (3.2a) and (3.2b), the vector of decrements (u, r) and the right-hand-side are $(M + 1)$-dimensional. In adapting a time-stepping code to perform continuation, it is desirable that vectors remain M-dimensional, in order to facilitate

communication between subroutines performing tasks like Newton and conjugate gradient iteration and those performing fluid-mechanical computations such as $N(U)$ or $(I - \Delta t L)^{-1}$.

In fact, although it is notationally convenient to write (3.2a) and (3.2b) as $(M + 1) \times (M + 1)$ systems, because the last equation involves only one unknown, each of the systems is easily reduced to an $M \times M$ system. R or U_m may immediately be set to its prescribed value p^* (this would would be true after one Newton iteration, since the last equation is linear) and r or u_m set to zero. Equation (3.2a) is then merely a restatement of equation (2.2), while equation (3.2b) reduces to

$$(RN_U + L)\bar{u} + N(U)r = RN(U) + LU \tag{3.3}$$

where $\bar{u} = u$ except that $\bar{u}_m = 0$. Thus (3.3) is an equation for the M unknowns $(\bar{u}, r)$.

We wish to use the same data structures for conjugate gradient iteration as for fluid-mechanical computations. We therefore store the control parameter decrement r in the location, u_m, of the velocity decrement which is fixed at zero. In the subroutine which computes the left-hand-side in (3.3), we precede the matrix-vector multiplication by unpacking the data:

$$\bar{u}_i \leftarrow u_i, \quad \text{for } i \neq m \tag{3.4a}$$

$$\bar{u}_m \leftarrow 0 \tag{3.4b}$$

$$r \leftarrow u_m \tag{3.4c}$$

i.e. $(\bar{u}, r) \leftarrow u$. When the conjugate gradient iteration converges, the solution vector must again be unpacked via (3.4) to update U and R.

The system (3.3) can be preconditioned in the same way as system (2.2). By multiplying both sides of (3.3) by $(I - \Delta t L)^{-1}\Delta t$

$$\begin{aligned}(I - \Delta t L)^{-1}\Delta t[(RN_U + L)\bar{u} + N(U)r] = \\ (I - \Delta t L)^{-1}\Delta t(RN(U) + LU)\end{aligned}$$

we obtain

$$\begin{aligned}[(I - \Delta t L)^{-1}(I + \Delta t(RN_U + N(U)r) - I]\bar{u} = \\ [(I - \Delta t L)^{-1}(I + \Delta t RN) - I]U\end{aligned} \tag{3.5}$$

The right-hand-side of (3.5) is again the difference between consecutive time-steps. The left-hand-side is the difference between consecutive linearized time-steps, with the additional replacement $RN_U u \rightarrow RN_U u + N(U)r$. The system (3.5) with a large Δt is again rapidly solved by conjugate gradient iteration, e.g. by BCGS.

4. Bifurcation points. We now wish to calculate bifurcation points directly using the techniques described in sections 1–3 above. A steady state U undergoing a steady bifurcation at control parameter value R in the direction determined by H satisfies the following system:

$$0 = RN(U) + LU \tag{4.1a}$$

$$0 = RN_U H + LH \tag{4.1b}$$

$$0 = p(H) - p^* \equiv H_m - 1 \tag{4.1c}$$

Equation (4.1c) represents one choice of normalization for the eigenvector, and serves to exclude $H = 0$.

Writing these equations for the decremented vector $(U - u, H - h, R - r)$ and expanding to linear order we get:

$$\begin{bmatrix} RN_U + L & 0 & N(U) \\ RN_H & RN_U + L & N_U H \\ 0 \dots & \dots 1 \dots & 0 \end{bmatrix} \begin{bmatrix} u \\ h \\ r \end{bmatrix} = \begin{bmatrix} RN(U) + LU \\ RN_U H + LH \\ H_m - 1 \end{bmatrix} \tag{4.2}$$

In writing (4.2), we have expanded (4.1b) for our quadratic nonlinearity $N(U)$ as follows:

$$\begin{aligned} N_{U-u}(H-h) &= \Pi\big[\ ((U-u)\cdot\nabla)(H-h) + ((H-h)\cdot\nabla)(U-u)\big] \\ &= \Pi\big[\ (U\cdot\nabla)H - (u\cdot\nabla)H - (U\cdot\nabla)h \\ &\qquad + (H\cdot\nabla)U - (H\cdot\nabla)u - (h\cdot\nabla)U + O(u,h)^2\big] \\ &= \quad N_U H - N_H u - N_U h + O(u,h)^2 \end{aligned}$$

so that

$$\begin{aligned} (R-r)N_{U-u}(H-h) + L(H-h) = \\ RN_U H + LH - RN_H u - RN_U h - Lh - N_U H r + O(u,h,r)^2 \end{aligned}$$

It may be that a continuous known branch of solutions U exists for all R. This situation is widespread, at least in the literature, since the study of the stability of and bifurcations from a known branch of solutions is amenable to analysis. Examples of such branches of solutions are the motionless conductive state in convection, azimuthal Couette flow in Taylor-Couette flow, and plane Couette and Poiseuille flows. Rather than saddle-nodes, the steady bifurcations that occur are transcritical and, if the system has a symmetry of some kind, pitchforks.

Elimination of (4.1a) by using a known solution U greatly reduces (4.2). The linear system to be solved for (h, r) at each Newton step becomes

$$\begin{bmatrix} RN_U + L & N_U H \\ \dots 1 \dots & 0 \end{bmatrix} \begin{bmatrix} h \\ r \end{bmatrix} = \begin{bmatrix} RN_U H + LH \\ H_m - 1 \end{bmatrix} \tag{4.3}$$

Henry [7] observed that system (4.3) is almost identical to (3.2b) and can be preconditioned and solved in the same way.

This method has been used to calculate bifurcation points in various two-dimensional convective flows [7, 9, 22, 25].

5. Linear stability analysis. We now consider the problem of determining the linear stability of steady states. The stability of U is governed by the eigenvalues λ of the Jacobian $A \equiv N_U + L$:

$$(N_U + L)u = \lambda u \tag{5.1}$$

This follows from the fact that infinitesimal perturbations from a steady state U evolve according to the linear stability equations:

$$\partial_t u = (N_U + L)u, \tag{5.2}$$

(we suppress the dependence on R). Knowing whether any eigenvalue has a positive real part is sufficient to determine the stability of U. In addition, it is also useful to know how many eigenvalues are positive, if others are negative but close to zero, and the structure of the corresponding eigenvectors. In other words, we wish to know several leading eigenpairs – the eigenvalues of maximal real part and corresponding eigenvectors.

Our matrices are considered to be sufficiently large that diagonalization, i.e. calculating all of the eigenvectors and eigenvalues via the QR algorithm (operation count $O(M^3)$), is not an option. Indeed, the vast majority of the eigenvalues are superfluous for our needs.

The basic technique for iterative calculation of selected eigenvalues is the power method. In its simplest form, one acts repeatedly with a matrix A on an arbitrary initial vector u_0. The sequence of vectors $u_n \equiv A^n u_0$ approaches the dominant eigenvector, i.e. that whose corresponding eigenvalue is largest in magnitude, and the sequence of Rayleigh quotients $h_n \equiv u_n^T A u_n / u_n^T u_n$ converges to that eigenvalue.

The power method algorithm must be modified in two respects for our purposes. First, we seek more than one eigenpair. Since several, possibly complex, eigenvalues may be competing, we generally want to compute 2-4 eigenpairs accurately. The 2-4 eigenpairs desired are calculated more accurately if we also calculate, to lower accuracy, an equal number of unneeded eigenpairs; these serve as an error-absorbing buffer. Thus, we typically compute 4-8 eigenpairs. The calculation of several eigenpairs is accomplished by the Arnoldi method, or the block power or orthogonal subspace iteration methods, which are all closely related generalizations of the power method [1, 19]. We form a sequence u_0, Au_0, ..., $A^{K-1}u_0$, whose span defines the *Krylov space*. K is the number of eigenpairs sought, i.e. about 8. These vectors are orthonormalized to form a basis $v_1, v_2, \ldots v_K$ for the Krylov space. We define the $M \times K$ matrix $V(i,k) \equiv v_k(i)$ and the $K \times K$ Hessenberg matrix $H \equiv V^T A V$. (H is a K-dimensional generalization of the Rayleigh quotient.) When H is diagonalized, its eigenvalues approximate K of the eigenvalues of A, and its eigenvectors, multiplied by V, approximate K of the eigenvectors of A. Care should be taken not to split a complex conjugate pair; if this situation occurs, it is easily remedied by incrementing K by 1.

The second modification required to adapt the power method for linear stability analysis is to change the region of the complex plane in which eigenvalues are

sought. The dominant eigenvalues are of no interest to us: in the Navier-Stokes equations and in most reaction-diffusion equations, it is the negative eigenvalues corresponding to the most quickly damped modes that have the largest magnitude. This property of the Jacobian, inherited from L, has already been encountered in our discussions of timestepping (where fast timescales necessitate implicit timestepping) and steady-state solving (where poor conditioning requires preconditioning). We are instead interested in the leading eigenvalues, i.e. those of largest real part. We consider two options.

5.1. Exponential power method. The solution to the linearized evolution equation (5.2) is

$$u(t + \Delta t) = e^{\Delta t(N_U + L)} u(t) \tag{5.3}$$

The leading eigenvalues of any matrix A are the dominant ones of $\exp(\Delta t A)$ for any positive Δt. For $\Delta t \ll 1$, the linearized time-stepping scheme

$$u(t + \Delta t) = (I - \Delta t L)^{-1}(I + \Delta t N_U)u(t) \tag{5.4}$$

that is already available provides an approximation to (5.3). The power or Arnoldi method can be carried out on $\exp(\Delta t A)$ by integrating the linearized equations via (5.4). Each linearized timestep serves as one iteration of the power method.

The drawback of this comes from the requirement that $\Delta t \ll 1$ in order for (5.4) to approximate (5.3). (Other timestepping schemes can be used that are more accurate than the first-order Euler scheme of (5.4), but this has a minimal effect on the maximum Δt allowed.) Let $\lambda_1 > \lambda_2 > \dots$ be the leading eigenvalues of A (all real, for simplicity), so that $\exp(\Delta t \lambda_1) > \exp(\Delta t \lambda_2) \dots$ are the dominant eigenvalues of $\exp(\Delta t A)$. Suppose we seek to calculate λ_1 via the simple power method on $\exp(\Delta t A)$. It is easily shown that one multiplication reduces the component in u_n corresponding to λ_2 by a factor of $\exp(\Delta t(\lambda_2 - \lambda_1))$. For $\Delta t \ll 1$, this factor is close to one, and convergence thus very slow. Similar reasoning applies to the block power or Arnoldi methods.

However, the exponential power method is easily implemented and very reliable. Complex eigenpairs can be found as easily as real ones. The exponential power method has been successfully used to compute leading eigenpairs in many problems of hydrodynamic stability [2, 5, 6, 7, 9, 10, 11, 15, 16, 17, 20, 22, 23, 25].

5.2. Inverse power method. In the most straightforward case, we seek the eigenvalue(s) nearest zero. The method of choice is then the inverse power method, which calls for acting repeatedly with A^{-1} instead of with A. The simple convergence analysis above then shows that the error at each step is reduced by a factor of λ_1/λ_2. Near bifurcations, where $\lambda_1 \approx 0$, the convergence is thus extremely rapid. However, we need a way to act with A^{-1}.

In section 2, we showed that the linear system involving $N_U + L$ arising from Newton's method could be rapidly solved by conjugate gradient iteration if preconditioned with $(I - \Delta t L)^{-1}\Delta t$, The following calculations show that

the same preconditioning can be used to carry out the inverse power iteration $u_{n+1} = A^{-1}u_n$:

$$\begin{aligned} u_{n+1} &= (N_U + L)^{-1}u_n \\ (N_U + L)u_{n+1} &= u_n \\ (I - \Delta tL)^{-1}\Delta t(N_U + L)u_{n+1} &= (I - \Delta tL)^{-1}\Delta t u_n \\ (I - \Delta tL)^{-1}\left[(I + \Delta tN_U) - (I - \Delta tL)\right]u_{n+1} &= (I - \Delta tL)^{-1}\Delta t u_n \\ \left[(I - \Delta tL)^{-1}(I + \Delta tN_U) - I\right]u_{n+1} &= (I - \Delta tL)^{-1}\Delta t u_n \end{aligned} \tag{5.5}$$

We see that, just as in equation (2.3), the left-hand-side of (5.5) requires taking only the difference of consecutive linearized timesteps (5.4). The right-hand-side consists of taking one linear (not linearized) timestep, multiplied by Δt. System (5.5), like (2.3) is well-conditioned in many cases of interest, and thus is solved rapidly by the same conjugate gradient variants, such as BCGS and GMRES. Equation (5.5) can be incorporated into the Arnoldi or block power methods to compute the eigenpair more accurately and to calculate several eigenpairs whose eigenvalues are closest to zero.

The calculation of complex leading eigenpairs by the inverse power method is more complicated. The eigenvalues sought are no longer necessarily those closest to zero. Shifting must be used to bring the leading eigenvalues close to zero: instead of solving $Au_{n+1} = u_n$, one solves

$$(A - \lambda^{(n)}I)u_{n+1} = u_n \tag{5.6}$$

where $\lambda^{(n)}$ is the current estimate of one member of the complex conjugate pair of leading eigenvalues. Although shifting is in principle accomplished merely by substituting $N_U - \lambda^{(n)}I$ for N_U in (5.4), several difficulties arise: Initial estimates of $\lambda^{(n)}n$ may be difficult to obtain. Different shifts are required for eigenvalues with different imaginary parts. Finally, for complex shifts either complex arithmetic must be used or equivalently a real problem of twice the size must be solved.

This method has been used to compute real leading eigenvalues in spherical Couette flow [4] and complex leading eigenvalues in natural convection [8].

6. Floquet stability analysis. Often one is interested in the stability of periodic orbits rather than of steady states. The exponential power method described in section 5.1 can be applied to this case with little modification. Consider a T-periodic solution $U(t \bmod T)$ of equation (1.1). Through its dependence on U, the operator N_U appearing in the linear-stability equations (5.2) is now also time periodic and the stability of a periodic solution cannot be determined from the eigenvalues of the constant Jacobian matrix. Rather, stability is determined by the eigenvalues of the monodromy operator (matrix) B defined formally by:

$$B = \exp\left(\int_{t_0}^{t_0+T} dt' \left(N_U(t') + L\right)\right). \tag{6.1}$$

The operator B takes an infinitesimal perturbation u of U at an initial time t_0 and evolves it forward under the linear flow to give the perturbation at time $t_0 + T$. Heuristically (6.1) can be understood as stating that the stability or instability of a periodic orbit is a consequence of the way linear growth and decay combine around the entire orbit. Hence it is necessary to follow a perturbation once around the orbit to assess overall growth or decay. In practice, the action of B is approximated by integrating (5.2) over $T/\Delta t$ timesteps.

The eigenvalues μ of B are known as Floquet multipliers and are independent of t_0. The corresponding eigenmodes do depend on t_0. For an initial condition $u(t_0)$ to (5.2) which is an eigenmode of B, the solution to (5.2) is of the form

$$u(t) = \tilde{u}(t \bmod T)e^{\lambda t} \tag{6.2}$$

where $\lambda = \log(\mu/T)$ is called a Floquet exponent and $\tilde{u}(t \bmod T)$ is called a Floquet mode.

The dominant Floquet multipliers (leading Floquet exponents) are those of interest for stability and bifurcation analysis. These can be computed by applying the power method to the matrix B. In fact one can view B as a generalization of the operator $\exp(\Delta t(N_U + L))$ of (5.3), considered in the exponential power method for steady states. The generalization to Floquet analysis leads to the following new considerations:

The first issue for implementation is that acting with B on a vector u means integrating the linear stability equations over one period and this means knowing the base solution U at a large number of time points (the integration time steps). Because the solutions are periodic it is natural to represent $U(t)$ as a Fourier series and keep only enough modes to represent $U(t)$ to some desired accuracy. Then U can be found at any time by interpolation. In studies of cylinder wake flow [3, 12] it was found that 16 Fourier modes (corresponding to 32 time points over one period) were sufficient to represent $U(t)$ at arbitrary times to within the accuracy of the simulation that produced U.

The second issue is that, if the period T is long, the first few dominant eigenvalues of B may differ greatly from one another. It should always be possible to calculate the dominant eigenvalue by the exponential power method, but smaller multipliers may be difficult or impossible to obtain if multiplication by B makes the corresponding components orders of magnitude smaller than the dominant ones. The method will break down altogether if the periodic orbit approachs a homoclinic or heteroclinic situation in which the period T goes to infinity. (See [14] for ways to handle such situations.)

This method has been used to calculate the three-dimensional instability of limit cycles in two-dimensional open flows, in particular cylindrical wake flow [3, 12] and perturbed plane Poiseuille flow [20].

7. Nonlinear stability analysis. In addition to computing bifurcation points we want to distinguish between those which are subcritical and those that are supercritical. The distinction arises in pitchfork and Hopf bifurcations and it can

be explained most simply in terms of the normal form for a pitchfork bifurcation written as:

$$\partial_t a = \sigma(R - R_c)a + \alpha a^3 \tag{7.1}$$

where a is the amplitude of the bifurcating solution, R_c is the bifurcation point, σ is a positive constant of proportionality relating changes in R to changes in the leading eigenvalue at the bifurcation, and α, called the Landau coefficient, determines the nonlinear character of the bifurcation: $\alpha < 0$ describes a supercritical bifurcation and $\alpha > 0$ describes a subcritical bifurcation. One can view the difference between these two cases either in terms of the direction of the bifurcating branches or in terms of the nonlinear flow along the center manifold at the bifurcation. Viewed the second way, the distinction is between a nonlinear flow at $R = R_c$ that is stable (supercritical case) and a flow at $R = R_c$ that is unstable (subcritical case).

We can exploit the difference in the dynamics along the center manifold to distinguish the two cases through a relatively simple computation. We evolve the nonlinear equations (1.1) starting near the bifurcation point (near refers both to parameter space and to phase space) and ascertain whether the nonlinear term is stabilizing or destabilizing. In practice we find that setting R slightly above R_c is the best approach (in part because R_c is not known exactly). We compute the steady (slightly unstable) solution U at this R. (In the case of a symmetry-breaking pitchfork bifurcation, this can be accomplished by time-stepping the equations restricted to the symmetric subspace.) We then compute its leading eigenvector u by the exponential power method. We then start a nonlinear simulation using the initial condition $U + \epsilon u$ for some small ϵ. Initially the simulation shows linear growth consistent with a small positive eigenvalue: $\sigma(R - R_c) > 0$. When the dynamics deviates from linear growth, it is simple to estimate α from the time series and to thereby determine whether the bifurcation is subcritical or supercritical. Note that the dynamics in the two cases is very different. In the subcritical case, the nonlinear growth is faster than the linear growth, whereas in the supercritical case it is the other way around. Therefore the *sign* of α, which is the essential bit, can be found very reliably.

This method has been used to demonstrate the subcriticality of pitchfork bifurcations in cylindrical wake flow [3, 12], in perturbed plane Couette flow [5], and in double-diffusive natural convection [25].

REFERENCES

[1] W.E. ARNOLDI, *The principle of minimized iterations in the solution of the matrix eigenvalue problem*, Q. Appl. Math. **9** (1951), 17–29.

[2] D. BARKLEY AND L.S. TUCKERMAN, *Traveling waves in axisymmetric convection: the role of sidewall conductivity*, Physica D **37** (1989), 288–294.

[3] D. BARKLEY AND R. HENDERSON, *Floquet stability analysis of the periodic wake of a circular cylinder*, J. Fluid Mech. **322** (1996), 215–241.

[4] D. BARKLEY AND L.S. TUCKERMAN, *Stokes preconditioning for the inverse power method*, in Lecture Notes in Physics: Proc. of the Fifteenth Int'l. Conf. on Numerical Methods in

Fluid Dynamics, P. Kutler, J. Flores, and J.-J. Chattot, eds., Springer, New York, 1997, 75–76.

[5] D. BARKLEY AND L.S. TUCKERMAN, *Stability analysis of perturbed plane Couette flow*, Phys. Fluids, submitted (1998).

[6] D. BARKLEY, M.G.M. GOMES, AND R. HENDERSON, *Three-dimensional stability analysis of flow over a backward facing step*, J. Fluid Mech., submitted (1998).

[7] A. BERGEON, D. HENRY, H. BENHADID, AND L.S. TUCKERMAN, *Marangoni convection in binary mixtures with Soret effect*, J. Fluid Mech., in press (1998).

[8] F. BERTAGNOLIO, L.S. TUCKERMAN, P. LE QUÉRÉ, AND O. DAUBE, *Calculation of leading eigenmodes in natural convection by the inverse power method with Stokes preconditioning*, Scientific report, Laboratoire d'Informatique pour la Mécanique et les Sciences de l'Ingénieur, Orsay, 1998.

[9] E. CHÉNIER, *Etude de la stabilité linéaire des écoulements thermocapillaries et thermogravitationnels en croissance cristalline* Thesis, Université de Paris XI; Notes et Documents LIMSI No. 97-26, 1997.

[10] K.N. CHRISTODOULOU AND L.E. SCRIVEN, *Finding leading modes of a viscous free surface flow: an asymmetric generalized eigenproblem*, J. Sci. Comput. **3** (1988), 355–406.

[11] I. GOLDHIRSCH, S.A. ORSZAG AND B.K. MAULIK, *An efficient method for computing leading eigenvalues and eigenvectors of large asymmetric matrices*, J. Sci. Comput. **2** (1987), 33–58.

[12] R.D. HENDERSON AND D. BARKLEY, *Secondary instability in the wake of a circular cylinder*, Phys. Fluids **8** (1996), 1683–1685.

[13] Y.A. KUZNETSOV, *Elements of Applied Bifurcation Theory*, Springer, New York, 1995.

[14] K. LUST, *Numerical bifurcation analysis of periodic solutions of partial differential equations*, PhD thesis, Katholieke Universiteit Leuven, 1997.

[15] C.K. MAMUN AND L.S. TUCKERMAN, *Asymmetry and Hopf bifurcation in spherical Couette flow*, Phys. of Fluids **7** (1995), 80–91.

[16] P.S. MARCUS AND L.S. TUCKERMAN, *Numerical simulation of spherical Couette flow. Part I: Numerical methods and steady states*, J. Fluid Mech. **185** (1987), 1–30.

[17] P.S. MARCUS AND L.S. TUCKERMAN, *Numerical simulation of spherical Couette flow. Part II: Transitions*, J. Fluid Mech. **185** (1987), 31–65.

[18] D.R. KINCAID, T.C. OPPE, AND W.D. JOUBERT, *An overview of NSPCG: A nonsymmetric preconditioned conjugate gradient package*, Report CNA-228, Center for Numerical Analysis, University of Texas at Austin, 1988.

[19] Y. SAAD, *Variations on Arnoldi's method for computing eigenelements of large unsymmetric matrices*, Linear Alg. Appl. **34** (1980), 269–295.

[20] M.F. SCHATZ, D. BARKLEY, AND H.L. SWINNEY, *Instabilities in spatially periodic channel flow*, Phys. Fluids **7** (1995), 344–358.

[21] R. SEYDEL, *Practical Bifurcation and Stability Analysis*, Second edition, Springer, New York, 1994.

[22] R. TOUIHRI *Stabilité des écoulements dans une cavité cylindrique chauffée par le bas en présence d'un champ magnétique*, Thesis, Ecole Centrale de Lyon, 1998.

[23] L.S. TUCKERMAN AND D. BARKLEY, *Global bifurcation to travelling waves in axisymmetric convection*, Phys. Rev. Lett. **61** (1988), 408–411.

[24] L.S. TUCKERMAN, *Steady-state solving via Stokes preconditioning; recursion relations for elliptic operators*, in Lecture Notes in Physics: Proc. of the Eleventh Int'l. Conf. on Numerical Methods in Fluid Dynamics, D.L. Dwoyer, M.Y. Hussaini, and R.G. Voigt, eds., Springer, New York, 1989, 573–577.

[25] S. XIN, P. LE QUÉRÉ, AND L.S. TUCKERMAN, *Bifurcation analysis of double-diffusive convection with opposing horizontal thermal and solutal gradients* Phys. of Fluids. **10** (1998), 850–858.