Scalable bifurcation analysis algorithms for large parallel applications

Andrew G. Salinger [a,*], Roger P. Pawlowski [a], Louis A. Romero [b]

[a] *Parallel Computational Sciences Department, Sandia National Laboratories, P.O. Box 5800, MS-1111, Albuquerque, NM 87185-1111, USA*

[b] *Computational Mathematics and Algorithms Department, Sandia National Laboratories, P.O. Box 5800, MS-1111, Albuquerque, NM 87185-1111, USA*

Abstract

A set of stability analysis algorithms have been developed for analysis of large-scale nonlinear applications on parallel computers, and applied to 2D and 3D incompressible flow applications. These analysis tools include several continuation algorithms for locating and tracking bifurcations and a linear stability analysis capability. The continuation algorithms are developed to be readily linked to application codes that already use Newton's method.

Keywords: Bifurcations; Eigenvalues; Parallel computers; Finite element; CFD; Incompressible flow; Jets; Pitchfork bifurcations

1. Introduction

The nonlinear stability of a system must be well understood in order to design, operate, and control an engineering system with a reasonable level of certainty. While sophisticated algorithms exist for studying the stability of numerical models consisting of small sets of ODEs, they have not been broadly applied to the high-fidelity engineering models that are being solved today on parallel computers. A set of tools for performing stability analysis of large-scale nonlinear systems have been developed at Sandia National Laboratories, and include the following algorithms:

- Parameter continuation algorithms, including pseudo arc-length for locating solution multiplicity.
- A turning point bifurcation (a.k.a. fold) tracking algorithm.
- A Pitchfork (symmetry breaking) bifurcation tracking algorithm.
- A Hopf bifurcation tracking algorithm.
- A linear stability analysis tool for approximating leading eigenvalues.

The bifurcation analysis algorithms are collected in the LOCA library, that is designed to be readily interfaced with codes that already use a full Newton method. While results for all the algorithms will be shown in the oral presentation, in the limited space available here we briefly describe just the pitchfork bifurcation tracking algorithm. The algorithm is demonstrated on a CFD application modeled by the MPSalsa code. This code is an unstructured grid finite element code that simulates incompressible reacting flows on massively parallel computers [1–4] using a full Newton method and the Aztec iterative linear solver library [5].

2. Numerical methods overview

The algorithms fall into two distinct categories: a set of bifurcation analysis tools and an eigensolver.

The bifurcation algorithms are used to detect regions of multiple steady states and delineate regions of qualitatively different behaviors. The algorithms currently implemented include zeroth order, first order, and pseudo arc-length continuation algorithms [6,7], a turning point bifurcation tracking algorithm, pitchfork bifurcation tracking algorithm, and a Hopf tracking algorithm. An excellent review article on algorithms for performing numerical bifurcation analysis has recently been published [8]. Each of the algorithms was implemented in the LOCA library using bordering algorithms, which require minimal intrusiveness to codes that

[*] Corresponding author. Tel.: +1 (505) 845-3523; Fax: +1 (505) 845-7442; E-mail: agsalin@sandia.gov

are already set up to do a fully coupled Newton method. All routines except the Hopf bifurcation tracking algorithm where written to have just three main calls to the application code: (1) calculate a residual vector given a solution vector and a parameter value; (2) calculate a Jacobian matrix given a solution vector and a parameter value; and (3) solve a linear system given a Jacobian matrix and right hand side. Since the linear systems all involve the same Jacobian matrix as solved by the steady-state code, these algorithms do not require modification of the matrix fill routine, sparse matrix allocation, or parallel communication maps. The Hopf tracking algorithm also requires the computation of a mass matrix (the coefficient matrix of the time derivative terms) and a linear solver for a complex-valued matrix [9].

Due to limited space, only the outline of the pitchfork bifurcation tracking algorithm will be shown. The algorithm is an extension of a usual Newton iteration for reaching a steady-state solution vector x for a given parameter p, which is to solve

$$R(x, p) = 0 \quad \text{with Newton iteration } \mathbf{J}\delta x = -R \tag{1}$$

where R is the residual equations from the PDE discretization, $\mathbf{J}$ is the Jacobian matrix, and δx is the update to the latest estimate of the solution vector.

To locate a pitchfork bifurcation, we want to find the point on the symmetric solution branch where one eigenvalue is zero. We use these conditions to formulate a Newton method. To start the algorithm, a vector Ψ that is antisymmetric with respect to the symmetry that is broken at the pitchfork is required. This vector is generated using an eigensolver, by calculating the eigenvector associated with the eigenvalue that is crossing zero at the pitchfork. The algorithm consists of a system of system of $2N_x + 2$ unknowns (x, n, ε, and p), where N_x is the length of x (and the order of $\mathbf{J}$), n is the null vectors at the pitchfork bifurcation, and ε is a slack parameter representing the asymmetry in the system. (If the numerical system is truly symmetric, then the algorithm will drive ε to zero, but if there is a slight asymmetry in the problem — perhaps due to a nonuniform mesh — then ε will be nonzero.) The $2N_x + 2$ equations specifying the pitchfork bifurcation are,

$$\begin{aligned} R(x, p) + \varepsilon\Psi &= 0 \\ \mathbf{J}(x, p)n &= 0 \\ \langle x, \Psi\rangle &= 0 \\ l^t n &= 1 \end{aligned} \tag{2}$$

The first equation specifies that the solution is a steady state when the slack parameter is driven to zero, and the second equation specifies that the system is singular. The third equation (containing an inner product) forces the solution vector x to be off of the asymmetric branches, and the last equation normalizes the length of the null vector. The form of l^t, often chosen to be Ψ, can be chosen by the user.

This algorithm does not require the user to formulate the pitchfork problem different from the physical problem. The symmetry is found by the eigensolver. An alternative approach requires the user to specify the symmetry being broken through meshing and boundary conditions. This involves more user intervention and in some cases can be difficult to implement, when the boundary conditions destroy the sparsity pattern of the finite element method.

These equations are solved using a Newton method using Ψ as an initial guess for n. A bordering algorithm is used to solve the Newton step that requires six linear solves of the matrix $\mathbf{J}$, and is equivalent to solving the size $2N_x + 2$ once. The biggest numerical difficulty in solving for bifurcations using a bordering algorithm is that we use an iterative solver (which is required for efficient solves of large systems) to solve the same matrix which we are driving singular. However, we have found the iterative solvers work well unless we try to locate the bifurcation to more than 4 digits of accuracy.

The algorithms and parallel implementation of linear stability analysis algorithms have been detailed in published articles [10–12]. A linearization of the problem about a steady state leads to a generalized eigenvalue problem. A Cayley transformation is used to transform the eigen spectrum in a way that an Arnoldi iteration will converge to the eigenvalues of interest. We use the ARPACK library to perform the Arnoldi iteration [13,14]. The main computational hurdle for a scalable algorithm is the solution of the linear set of equations to sufficient accuracy with an parallel iterative matrix solver. Details are found in a previous paper, which contains the approximation of the 6 rightmost eigenvalues of an order 4 million unknown reactor analysis problem [11].

3. Application of scalable stability analysis

To briefly illustrate the scalable bifurcation analysis capabilities, we present results for the symmetry breaking of two opposed jets. The problem comes from an analysis of the counterflow jet reactor, an experimental system that has been used to measure gas phase kinetics in a wall-less environment [15,16]. It has been shown that at steady state, the stagnation point between two opposing jets can be located away from the midpoint distance between them, even if the jets have equal mass flow rates and the system is isothermal. It is found that for high enough flow rates, described by the Reynolds number (based on the gap length between the jets), Re, the symmetric solution becomes unstable to the asymmetric solutions. Fig. 1 depicts two steady-state flow profiles, for the symmetric and asymmetric solution branches.

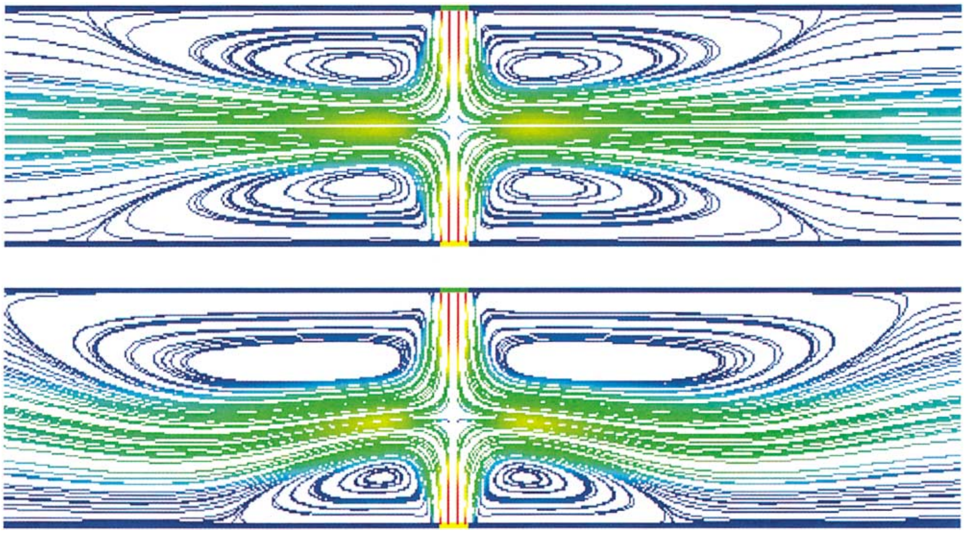

Fig. 1. Streamline profiles of the counterflow jet reactor, showing both the symmetric solution (top) and the asymmetric solution (bottom) that arise due to a pitchfork bifurcation.

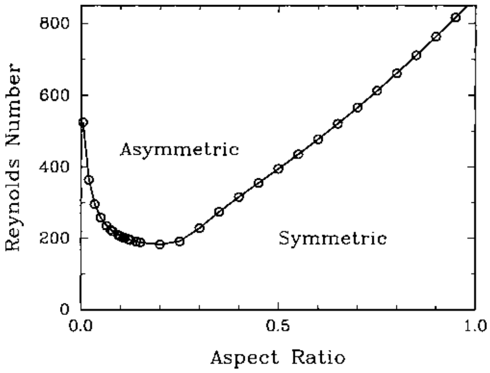

Fig. 2. Two-parameter bifurcation set showing the locus of pitchfork bifurcations for 2D infinite planar jets (i.e. Cartesian coordinates). The curves delineate regions where stable steady state solutions exhibit symmetric or asymmetric flow profiles. The model consisted of 19,000 unknowns and was run on 24 processors.

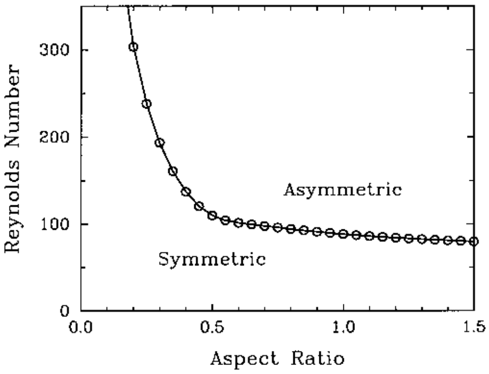

Fig. 3. Two-parameter bifurcation set showing the locus of pitchfork bifurcations for opposed circular jets in 2D (using cylindrical coordinates). The curves delineate regions of symmetric and asymmetric flow profiles. The 2D model consisted of 15,000 unknowns and was run on 24 processors.

The initial pitchfork bifurcation is detected using the linear stability analysis capability. At each step along a continuation run, where steady state solutions are calculated at increasing values of *Re*, the rightmost eigenvalues were approximated. When a single real eigenvalue crossed through zero to the positive half plane, signaling a pitchfork bifurcation, the pitchfork tracking algorithm (2) was started. The flow rate (*Re*) at which the pitchfork occurs is converged to using Newton’s method. Once one pitchfork is located, it is easily tracked as a function of a second parameter, a geometric aspect ratio comparing the diameter of the inlet jet to the gap between the jets. Fig. 2 show the curve of pitchfork bifurcations in the two-parameter space for a 2D calculation of rectangular jets. Fig. 3 and Fig. 4 show the same results for circular jets, a more accurate model for the counterflow reactor, which are calculated in 2D using cylindrical coordinates and repeated in 3D using Cartesian coordinates. By tracking the bifurcation point, an operability chart is directly generated that shows the maximum flow rate at which the reactor can be operated to still avoid the asymmetric solutions.

4. Conclusions

A set of bifurcation analysis algorithms and a linear stability analysis capability have been successfully imple-

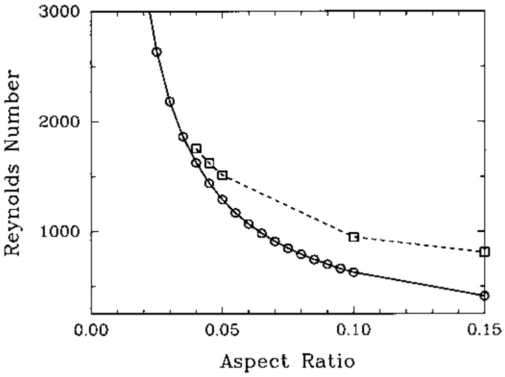

Fig. 4. Two-parameter bifurcation set showing the locus of pitchfork bifurcations for opposed circular jets in 2D (using cylindrical coordinates) and in 3D, for small aspect ratios. The curves delineate regions of symmetric and asymmetric flow profiles, and differ only due to the lower resolution of the 3D calculation. The 2D model consisted of 15,000 unknowns and was run on 24 processors, and the 3D model consisted of 350,000 unknowns and was run on 256 processors.

mented around a massively parallel incompressible flow code. The algorithms rely only on iterative matrix solvers and so are scalable to large problems, and are designed to be easily implemented around application codes. In one example problem, the locus of pitchfork bifurcations delineating symmetric and asymmetric operation of a counterflow jet reactor were automatically calculated. The pitchfork algorithm has been shown to work on problems as large as 350,000 unknowns distributed across 256 processors of a distributed memory parallel computer.

Acknowledgements

The authors with to acknowledge the work of John Shadid, Rich Lehoucq, Ray Tuminaro, TJ Mountziaris, and the others who in various ways have laid the groundwork for this contribution. This work was partially supported by the MICS division of the Office of Science at DOE. Sandia is a multiprogram laboratory operated by Sandia Corporation, and Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000.

References

[1] Shadid JN, Moffat HK, Hutchinson SA, Hennigan GL, Devine KD, Salinger AG. MPSalsa: a finite element computer program for reacting flow problems, part 1 — theoretical development. Sandia National Laboratories Technical Report, SAND95-2752, 1996.

[2] Salinger AG, Devine KD, Hennigan GL, Moffat HK, Hutchinson SA, Shadid JN. MPSalsa: a finite element computer program for reacting flow problems, part 2 — user's guide. Sandia National Laboratories Technical Report, SAND96-2331, 1996.

[3] Salinger AG, Shadid JN, Hutchinson SA, Hennigan GL, Devine KD, Moffat HK. Analysis of gallium arsenide deposition in a horizontal CVD reactor using massively parallel computations. J Cryst Growth 1999;203:516–533.

[4] Shadid JN. A fully-coupled Newton–Krylov solution method for parallel unstructured finite element fluid flow, heat and mass transfer simulations. Int J CFD 1999;12:199–211.

[5] Hutchinson SA, Shadid JN, Tuminaro RS. Aztec User's Guide: Version 1.0. Sandia National Laboratories Technical Report, SAND95-1559, 1995.

[6] Keller HB. In: Rabinowitz PH (Ed), Applications of Bifurcation Theory. New York: Academic, 1997, p. 359.

[7] Salinger AG, Brandon S, Aris R, Derby JJ. Buoyancy driven flows of a radiatively participating fluid in a vertical cylinder heated from below. Proc R Soc A 1993;442:313–341.

[8] Cliffe KA, Spence A, Tavener SJ. The numerical analysis of bifurcation problems with application to fluid mechanics. Acta Numer 2000;39–131.

[9] Day D, Heroux M. Solving complex-valued linear systems via equivalent real formulations. SIAM J Sci Comput, in press.

[10] Lehoucq RB, Salinger AG. Massively parallel linear stability analysis with P_ARPACK for 3D fluid flow modeled with MPSalsa. In: Agstrom B, Dongarra J, Elmroth E, Wasniewski J (Eds), Applied Parallel Computing, PARA'98. Lecture Notes in Computer Science, No. 1541. Berlin: Springer-Verlag, 1998, pp. 286–295.

[11] Lehoucq RB, Salinger AG. Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers. Int J Numer Methods Fluids, in press.

[12] Morzynski M, Afanasiev K, Thiele F. Solution of the eigenvalue problem resulting from global non-parallel flow stability analysis. Comput Methods Appl Mech Eng 1999;169:161–176.

[13] Lehoucq RB, Sorensen DC, Yang C. ARPACK Users Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods. Philadelphia, PA: SIAM press, 1998.

[14] Maschhoff KJ, Sorensen DC. P_ARPACK: An efficient portable large scale eigenvalue package for distributed memory parallel architectures. In: Wasniewski J, Dongarra J, Madsen K, Olesen D (Eds), Applied Parallel Computing in Industrial Problems and Optimization. Lecture Notes in Computer Science, Volume 1184. Berlin: Springer-Verlag, 1996.

[15] Gupta V, Safvi SA, Mountziaris TJ. Gas phase decomposition kinetics in a wall-less environment using a counterflow jet reactor: design and feasibility studies. Ind Eng Chem Res 1996;35:3248–3255.

[16] Pawlowski RP, Salinger AG, Mountziaris TJ. Bifurcation analysis of impinging isothermal jets. In preparation.