INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2002; **40**:1059–1073 (DOI: 10.1002/fld.392)

Computational bifurcation and stability studies of the 8:1 thermal cavity problem

Andrew G. Salinger[*,†], Richard B. Lehoucq, Roger P. Pawlowski and John N. Shadid

Sandia National Laboratories [‡], *Albuquerque*, *NM 87185-1111*, *U.S.A.*

SUMMARY

Stability analysis algorithms coupled with a robust Newton–Krylov steady-state iterative solver are used to understand the behavior of the 2D model problem of thermal convection in a 8:1 differentially heated cavity. Parameter continuation methods along with bifurcation and linear stability analysis are used to study transition from steady to transient flow as a function of Rayleigh number. To carry out this study the steady-state form of the governing PDEs is discretized using a Galerkin/least-squares finite element formulation, and solved on parallel computers using a fully coupled Newton method and preconditioned Krylov iterative linear solvers. Linear stability analysis employing a large-scale eigenvalue capability is used to determine the stability of the steady solutions. The boundary between steady and time-dependent flows is determined by a Hopf bifurcation tracking capability that is used to directly track the instability with respect to the aspect ratio of the system and with respect to mesh resolution. The effect of upwinding stabilization terms in the finite element formulation on the computed value of critical Rayleigh number is investigated. The Hopf bifurcation signaling the onset of flow is determined to occur at a critical Rayleigh number of $Ra = 3.0604 \times 10^5$. Copyright © 2002 John Wiley & Sons, Ltd.

KEY WORDS: bifurcations; stability; eigenvalues; parallel computers; finite element; CFD

1. INTRODUCTION

This manuscript presents a computational stability analysis of the model problem of confined thermal convection flow in an 8:1 enclosure. The model problem is fully described in Reference [1]. The computational method employs a robust steady-state Newton–Krylov solver for the non-linear systems, continuation methods for tracking solutions, and linear stability

[*] Correspondence to: A. G. Salinger, Parallel Computational Sciences Department, Sandia National Laboratories, Albuquerque, New Mexico 87185-1111, U.S.A.

[†] E-mail: agsalin@sandia.gov

[‡] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed-Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

Contract/grant sponsor: Department of Energy

Received 31 December 2001
Revised 1 July 2002

analysis capabilities. We will show how these capabilities provide a powerful tool for providing design information for systems with bifurcations. Using these techniques computationally efficient maps for stability regions for entire parameter spaces can be generated. However, while the locus of transitional states can be determined with these methods, transient simulations are needed to study the details of the time-dependent supercritical response of the system.

The power of applying bifurcation analysis algorithms to the study of flow instabilities has been well documented in numerous applications, most famously the Rayleigh–Benard and Taylor–Couette systems. An excellent review of this area by Ciiffe *et al.* [2] has recently been published. The reader is referred to that work and Govaerts [3] for various formulations of the bifurcation tracking algorithms. There is limited experience in applying these algorithms for large-scale PDE discretization that use approximate iterative linear solvers.

The model problem of buoyancy driven flow in a differentially heated cavity is described in the first contribution to this issue by Christon *et al.* [1]. The fluid flow and heat equations are discretized using the MPSalsa unstructured grid finite element code, which has been developed for robust steady-state solves on distributed memory parallel computers [4–9]. This code uses a Galerkin/least-squares discretization scheme, and includes a switch that can turn off the SUPG upwinding terms present in that formulation. The formulation is described in Section 2.1 and steady-state solution method in Section 2.2.

A linear stability analysis capability has been implemented with MPSalsa by combining a Cayley transformation with the Arnoldi-based P_ARPACK eigensolver [10, 11]. This capability has been verified and validated for numerous fluid flow applications and has demonstrated parallel scaling to millions of unknowns [12, 13], and is briefly described in Section 2.3. The library of continuation algorithms (LOCA) library [14] has also been interfaced with the MPSalsa code for directly calculating bifurcations [15]. A Newton-based algorithm in LOCA is used to converge directly to the instability, converging the parameter value and solution simultaneously. The Hopf tracking algorithm is presented in detail in Section 2.4.

In Section 3.1, the critical value of the Rayleigh number for the transition between stable steady flows and time-dependent flows is found. The instability is located with the eigensolver, and found to be a Hopf bifurcation signifying an oscillatory instability. Continuation of the Hopf point with respect to a second parameter, the aspect ratio of the box, traces neutral stability curves and provides insight into the structures of solution branches and the behavior of the system. A mesh resolution study is performed in Section 3.2, using the Hopf algorithm to find the instability on each mesh. The effect of the upwinding terms on the convergence with mesh of the critical Rayleigh number is documented in Section 3.3.

2. NUMERICAL METHODS OVERVIEW

2.1. Galerkin/least-squares finite element formulation

The governing transport PDEs describing fluid flow and thermal energy transfer are presented in Table I in residual form. In these equations, the unknown quantities are $\mathbf{u}$, P, and T; these are, respectively, the fluid velocity vector, the hydrodynamic pressure, and temperature.

Table I. Governing transport PDEs.

Momentum	$\mathbf{R}_m = \rho \dfrac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) - (\nabla \cdot \mathbf{T}) - \rho \mathbf{g}$
Total mass	$R_P = \dfrac{\partial \rho}{\partial t} + (\nabla \cdot \rho \mathbf{u})$
Thermal energy	$R_T = \rho \hat{C}_P \left[\dfrac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right] + \nabla \cdot \mathbf{q}_c$

Table II. Galerkin least-squares formulation of transport PDEs.

Momentum	$F_{m,i} = \int_\Omega \Phi R_{m,i} \, d\Omega + \int_{\Omega_e} \rho \tau_m (\mathbf{u} \cdot \nabla \Phi) R_{m,i} \, d\Omega$
Total mass	$F_P = \int_\Omega \Phi R_P \, d\Omega + \int_{\Omega_e} \rho \tau_m (\nabla \Phi \cdot \mathbf{R}_m) \, d\Omega$
Thermal energy	$F_T = \int_\Omega \Phi R_T \, d\Omega + \int_{\Omega_e} \rho C_p \tau_T (\mathbf{u} \cdot \nabla \Phi) R_T \, d\Omega$

The constitutive relations for a Newtonian stress tensor $\mathbf{T}$ and the Fourier law for the heat flux vector $\mathbf{q}_c$ are used to close this system of equations.

The continuous problem, defined by the transport equations, is approximated by a Galerkin least-squares (GLS) formulation [16–19]. This formulation allows for equal order interpolation of pressure and velocity (without spurious pressure solutions), and for stabilization of highly convected flows. The resulting GLS equations are shown in Table II.

The GLS total mass residual equation in expanded form is given in the following equation. The first term is the Galerkin term while the second pressure stabilization term is what allows for equal order interpolation.

$$F_P = \int_\Omega \Phi \left(\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right) d\Omega + \int_{\Omega_e} \rho \tau_m \nabla \Phi \cdot \left[\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} + \nabla P - \nabla \cdot \Upsilon - \rho \mathbf{g} \right] d\Omega \quad (1)$$

The Newtonian stress tensor, $\mathbf{T}$, is expanded to include the pressure, P, and the viscous stress tensor term, Υ. This expansion exhibits the weak form of a Laplacian operator acting on pressure

$$\int_{\Omega_e} \rho \tau_m \nabla \Phi \cdot \nabla P \, d\Omega \quad (2)$$

produced by the GLS formulation of the total mass conservation equation. The existence of this well-conditioned matrix in the FE discretization of the GLS equations allows the solution of the linear systems with a number of algebraic and domain decomposition-type preconditioners. This is in contrast to other formulations, such as Galerkin methods using mixed interpolation, that produce a zero block on the total mass continuity diagonal.

The second term in the GLS formulations of momentum and thermal energy are the up-winding terms for stabilizing highly convective flows. In Section 3.3 we will present results with these terms turned off. This is done by setting τ_m and τ_T to zero in these two equations without setting τ_m to zero in the total mass balance.

2.2. Overview of parallel Newton–Krylov implementation

In this section, a brief overview of the parallel numerical solution procedure for computing steady states is presented in varying degrees of completeness. References are provided to more complete sources on each of the topics.

2.2.1. Problem partitioning. Chaco [20], a general graph partitioning tool is used to partition the FE mesh into subdomains and make subdomain to processor assignments. *Chaco* constructs partitions and subdomain mappings that have low communication volume, good load balance, few message start-ups and only small amounts of network congestion. For the results in this paper, multi-level methods with Kernighan–Lin improvement were used. For a detailed description of parallel FE data structures and a discussion of the strong link between partitioning quality and parallel efficiency see Reference [21].

2.2.2. Newton–Krylov methods. A Newton–Krylov method [22] is an implementation of Newton's method in which a Krylov iterative solution technique is used to approximately solve the linear systems that are generated at each step of Newton's method. Specifically, to solve the non-linear system $\mathbf{F}(\mathbf{x})=\mathbf{0}$, the Krylov iterative solver is applied to determine an approximate solution of the Newton equation

$$\mathbf{J}(\mathbf{x})\mathbf{s} = -\mathbf{F}(\mathbf{x}) \tag{3}$$

where $\mathbf{J}(\mathbf{x})$ is the Jacobian matrix of $\mathbf{F}$ at the current iterate of $\mathbf{x}$. A Newton–Krylov method is usually implemented as an inexact Newton method. That is, one chooses a forcing term $\eta \in [0,1)$ and then applies a Krylov method until an iterate $\mathbf{s}_\mathrm{k}$ satisfies the inexact Newton condition

$$\|\mathbf{F}(\mathbf{x}) + \mathbf{J}(\mathbf{x})\mathbf{s}_\mathrm{k}\| \leqslant \eta\|\mathbf{F}(\mathbf{x})\| \tag{4}$$

A more complete discussion of the details of this inexact Newton implementation can be found in Reference [22].

2.2.3. Parallel preconditioned Krylov implementation. The linear subproblems generated from the inexact Newton method are solved by preconditioned Krylov methods as implemented in the *Aztec* solver library [7]. The parallel Krylov algorithms implemented in *Aztec* include techniques such as the restarted generalized minimal residual [GMRES(k)] and transpose-free quasi-minimal residual techniques for non-symmetric systems. It is well known that the overall performance of Krylov methods can be substantially improved when one uses preconditioning.

The preconditioners that we use in our subsequent calculations are based on algebraic additive Schwarz domain decomposition (DD) preconditioners [23] with variable overlapping between subdomains. This method corresponds to projecting the equations onto a series of overlapping subdomains and solving each subsystem. Since these subdomain solves are independent, they can be performed concurrently. Overlapping corresponds to increasing the size of the locally defined subdomain to include additional levels of FE nodes outside of the processor's assigned nodes. Thus, a single level of overlapping uses only information from

FE nodes that are connected by an edge (in the FE connectivity graph) that was cut by the original subdomain partition. Successive levels of overlap now use this method recursively by considering the previously overlapped points to now be *assigned* nodes to the expanded subdomain.

2.3. Linear stability analysis algorithms

Having a fully assembled Jacobian matrix and robust linear solvers enables the use of stability analysis tools. Details relating to the methods and parallel implementation of the linear stability analysis algorithms can be found in References [12, 13, 24, 25]. The analysis begins at a given steady-state solution point. A normal mode linear stability analysis produces a linearization of the evolution equations around this steady-state solution and produces a generalized eigenvalue problem of the form

$$\mathbf{J}z = \lambda\mathbf{B}z \tag{5}$$

where **J** is the Jacobian matrix, **B** is the mass matrix (i.e. coefficient matrix of time derivative terms), z is an eigenvector (generally complex), and λ its associated eigenvalue (also complex). A generalized Cayley transformation, which includes two adjustable real parameters, σ and μ, is used to reformulate the generalized eigenvalue problem into an ordinary eigenvalue problem for the transformed eigenvalues γ:

$$(\mathbf{J} - \sigma\mathbf{B})^{-1}(\mathbf{J} - \mu\mathbf{B})z = \gamma z \tag{6}$$

A simple relationship exists between the transformed and original eigenvalues, $\gamma = (\lambda - \mu)/(\lambda - \sigma)$. Appropriate choices of σ and μ are made so that the eigenvalues of interest (those λ with largest real part) are mapped to large γ. More details on appropriately choosing the Cayley parameters are given in previous works [12, 13].

The eigenvalue problem defined in Equation (6) is solved using Arnoldi's method with a version of the P_ARPACK software [10, 11] driven by software in the LOCA library for performing the generalized Cayley transformation [14]. The approximate matrix inversions are solved using the Aztec package, exactly the same as in the Newton iterations.

2.4. Hopf bifurcation tracking algorithm

A set of Newton algorithms for directly locating and tracking a bifurcation points has been implemented as part of the LOCA library at Sandia National Labs [14]. These have been developed to be relatively non-invasive to simplify implementation within an application code and to work with codes that are based on iterative linear solvers.

Background on bifurcation theory can be found in several texts [26, 27]. Briefly, bifurcations are points on a steady-state solution branch where the real part of one or more eigenvalues passes through zero as the branch is followed with respect to a key parameter. In the differentially heated cavity application, we will see that steady solutions encounter a Hopf bifurcation at a certain value of the Rayleigh number, referred to as the critical Rayleigh number. A Hopf bifurcation is an instability where a steady solution becomes unstable to oscillatory modes, and is characterized by a complex conjugate pair of eigenvalues that are purely imaginary (i.e. $\lambda = \pm\omega\mathrm{i}$). This definition, together with Equation (5), are used to define a system of equations that define the Hopf bifurcation [2, 3, 28]. In real arithmetic, this leads to a system of $3N_x+2$ unknowns ($\mathbf{x}, \mathbf{y}, \mathbf{z}, \omega$ and p). Here, N_x is the length of $\mathbf{x}$ (and the order of $\mathbf{J}$), while $\mathbf{y}$ and $\mathbf{z}$ are vectors (of length N_x) containing the real and imaginary parts of the eigenvector, $\mathbf{z} = \mathbf{y} + \mathrm{i}\mathbf{z}$. The $3N_x + 2$ equations specifying the Hopf bifurcation are then,

$$\begin{aligned} \mathbf{F} &= 0 \\ \mathbf{J}\mathbf{y} + \omega\mathbf{B}\mathbf{z} &= 0 \\ \mathbf{J}\mathbf{z} - \omega\mathbf{B}\mathbf{z} &= 0 \\ \phi\cdot\mathbf{y} &= 1 \\ \phi\cdot\mathbf{z} &= 0 \end{aligned} \tag{7}$$

The first vector equation requires a steady-state solution (where $\mathbf{F}(\mathbf{x}, p)$ is the vector of residuals), the next two vector equations specify that a purely imaginary eigenvalue exists, and the last two scalar equations are used to set the length and phase of the eigenvector. The unknown p is the critical parameter value, which is solved for as part of the solution, and is assigned to the Rayleigh number for all calculations in this paper. The scaling vector ϕ can be almost any arbitrary vector and is fixed throughout the calculations.

These $3N_x+2$ equations are solved using a Newton method. The linearized system used to determine the solution updates is

$$\begin{bmatrix} \mathbf{J} & 0 & 0 & 0 & \dfrac{\partial\mathbf{F}}{\partial p} \\ \dfrac{\partial(\mathbf{J}\mathbf{y})}{\partial\mathbf{x}} + \omega\dfrac{\partial(\mathbf{B}\mathbf{z})}{\partial\mathbf{x}} & \mathbf{J} & \omega\mathbf{B} & \mathbf{B}\mathbf{z} & \dfrac{\partial(\mathbf{J}\mathbf{y})}{\partial p} + \dfrac{\partial(\omega\mathbf{B}\mathbf{z})}{\partial p} \\ \dfrac{\partial(\mathbf{J}\mathbf{z})}{\partial\mathbf{x}} - \omega\dfrac{\partial(\mathbf{B}\mathbf{y})}{\partial\mathbf{x}} & -\omega\mathbf{B} & \mathbf{J} & -\mathbf{B}\mathbf{y} & \dfrac{\partial(\mathbf{J}\mathbf{z})}{\partial p} - \dfrac{\partial(\omega\mathbf{B}\mathbf{y})}{\partial p} \\ 0 & \phi^{\mathrm{t}} & 0 & 0 & 0 \\ 0 & 0 & \phi^{\mathrm{t}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\mathbf{x} \\ \Delta\mathbf{y} \\ \Delta\mathbf{z} \\ \Delta\omega \\ \Delta p \end{bmatrix} = \begin{bmatrix} -\mathbf{F} \\ -\mathbf{J}\mathbf{y} - \omega\mathbf{B}\mathbf{z} \\ -\mathbf{J}\mathbf{z} + \omega\mathbf{B}\mathbf{y} \\ 1 - \phi\cdot\mathbf{y} \\ \phi\cdot\mathbf{z} \end{bmatrix} \tag{8}$$

This derivation allows the mass matrix to depend on the solution vector and the parameter, though these terms can often be neglected. Instead of forming and solving the $3N_x + 2$ by $3N_x + 2$ matrix system in Equation (8), a bordering algorithm is used at each Newton step. This has the advantage of requiring fewer modifications to an existing code and of using less memory. The new formulation requires two linear solves of the matrix $\mathbf{J}$ and three solves of the complex matrix $\mathbf{J} + \mathrm{i}\omega\mathbf{B}$ (shown below in expanded real form)

$$\mathbf{J}\mathbf{a} = -\mathbf{F} \tag{9}$$

$$\mathbf{J}\mathbf{b} = -\frac{\partial\mathbf{F}}{\partial p} \tag{10}$$

$$\begin{bmatrix} \mathbf{J} & \omega\mathbf{B} \\ -\omega\mathbf{B} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} \mathbf{B}\mathbf{z} \\ \mathbf{B}\mathbf{y} \end{bmatrix} \tag{11}$$

$$\begin{bmatrix} \mathbf{J} & \omega\mathbf{B} \\ -\omega\mathbf{B} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{e} \\ \mathbf{f} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial(\mathbf{Jy})}{\partial\mathbf{x}}\mathbf{a} + \omega\dfrac{\partial(\mathbf{Bz})}{\partial\mathbf{x}}\mathbf{a} \\ \dfrac{\partial(\mathbf{Jz})}{\partial\mathbf{x}}\mathbf{a} - \omega\dfrac{\partial(\mathbf{By})}{\partial\mathbf{x}}\mathbf{a} \end{bmatrix} \tag{12}$$

$$\begin{bmatrix} \mathbf{J} & \omega\mathbf{B} \\ -\omega\mathbf{B} & \mathbf{J} \end{bmatrix} \begin{bmatrix} \mathbf{g} \\ \mathbf{h} \end{bmatrix} = \begin{bmatrix} \dfrac{\partial(\mathbf{Jy})}{\partial p} + \dfrac{\partial(\mathbf{Jy})}{\partial\mathbf{x}}\mathbf{b} + \omega\dfrac{\partial(\mathbf{Bz})}{\partial p} + \omega\dfrac{\partial(\mathbf{Bz})}{\partial\mathbf{x}}\mathbf{b} \\ \dfrac{\partial(\mathbf{Jz})}{\partial p} + \dfrac{\partial(\mathbf{Jz})}{\partial\mathbf{x}}\mathbf{b} - \omega\dfrac{\partial(\mathbf{By})}{\partial p} - \omega\dfrac{\partial(\mathbf{By})}{\partial\mathbf{x}}\mathbf{b} \end{bmatrix} \tag{13}$$

The temporary vectors named alphabetically **a** through **h**, which are computed by those linear solves, are used to calculate the updates to the solution vector as follows:

$$\Delta p = \frac{(\phi\cdot\mathbf{d})(1+\phi\cdot\mathbf{e}) - (\phi\cdot\mathbf{f})(\phi\cdot\mathbf{c})}{(\phi\cdot\mathbf{h})(\phi\cdot\mathbf{c}) - (\phi\cdot\mathbf{g})(\phi\cdot\mathbf{d})} \tag{14}$$

$$\Delta\omega = -\left[\frac{1+\phi\cdot\mathbf{e}+(\phi\cdot\mathbf{g})\Delta p}{(\phi\cdot\mathbf{c})}\right] \tag{15}$$

$$\Delta\mathbf{x} = \mathbf{a} + \mathbf{b}\Delta p \tag{16}$$

$$\Delta\mathbf{y} = -\mathbf{y} - \mathbf{e} - \mathbf{g}\Delta p - \mathbf{c}\Delta\omega \tag{17}$$

$$\Delta\mathbf{z} = -\mathbf{z} - \mathbf{f} - \mathbf{h}\Delta p - \mathbf{d}\Delta\omega \tag{18}$$

The three complex matrix equations (shown in real form in Equations (11)–(13)) are solved using a novel implementation for the solution of complex matrices with an existing real-valued sparse iterative linear solver [29]. This step is the main numerical difficulty in solving for the Hopf. Not only is this linear system double the order of Equation (3), but also the matrix is singular at the Hopf point. Although this algorithm would break down if one attempted to converge to the Hopf point to machine precision, our initial experience is that this algorithm works well as long as the iterative linear solver tolerance is set to require high accuracy, such as a 10^{-6} reduction in the linear residual.

3. APPLICATION OF SCALABLE STABILITY ANALYSIS ALGORITHMS

In this section, we will present results for applying the numerical methods presented in Section 2 to the benchmark problem. In this section, we use the terms centro-symmetric to describe eigenmodes that preserve the symmetry of the equations and boundary conditions, as described in Reference [1], and symmetry-breaking to describe those that break the centro-symmetry.

3.1. Results for 80×180 *mesh*

Results are shown for studying bifurcations in the thermal cavity problem. A mesh of 80×180 bilinear finite elements, highly graded towards the walls, was used for these calculations.

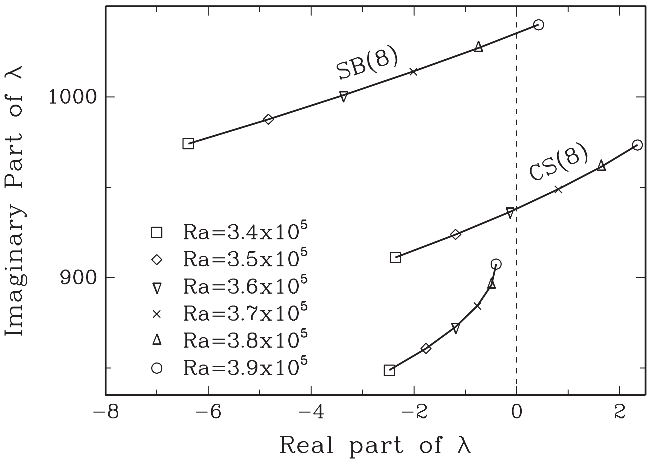

Figure 1. A plot showing the movement of the three leading eigenvalues as a function of the Rayleigh number shows two Hopf bifurcations, the first occurring near $Ra = 3.61 \times 10^5$. The mesh of 58 644 unknowns requires 1–2min for a steady-state solve and about 30min for an eigenvalue calculation on 24 processors of the Sandia-Intel Tflop (ASCI Red) machine. The curve labelled CS has centro-symmetric eigenfunctions, and the curve labelled SB has symmetry-breaking eigenfunctions.

The problem of 58 644 unknowns was solved in parallel on 24 333 MHz Pentium processors of the Sandia-Intel Tflop machine (ASCI Red). A typical matrix fill requires 0.35 s and an iterative matrix solve (using a domain decomposition preconditioner with overlap and a GMRES solver) about 15 s, and 3–6 N iterations were sufficient to converge to a steady state using a guess from a nearby parameter value, for a total of 1–2 min. Figure 1 shows the evolution of the three rightmost eigenvalues as a function of the Rayleigh number. Two Hopf bifurcations are detected, the first is a centro-symmetric mode near $Ra = 3.61 \times 10^5$ and a second symmetry-breaking mode near $Ra = 3.86 \times 10^5$. Because of the large imaginary parts, an Arnoldi space of 180 was needed to converge the first several eigenvalues using Cayley parameters of $\sigma = 2000$ and $\mu = 5000$. An eigensolve required about 30 min. While we believe the non-linear solver and the eigensolver are converged to 3 or more digits, the calculation is not converged with mesh spacing. This will be addressed in Section 3.2.

Three streamline plots are shown in Figure 2: the solution at the bifurcation point, one of the eigenvectors ($\mathbf{y}$) for the centro-symmetric instability at this point, and one of the eigenvectors for the symmetry-breaking instability at the second Hopf bifurcation. The symmetry of the solutions is diagnosed visually by observing that symmetric solutions having zero velocity at the center of the cavity while the symmetry-breaking eigenmodes have a non-zero streamfunction contour passing through this point.

The Hopf bifurcation tracking algorithm was run using results from the eigenvalue calculation as initial guesses for $\mathbf{y}$, $\mathbf{z}$, and ω. The results of tracking the Hopf bifurcation with respect to the aspect ratio of the cavity are shown in Figure 3. Calculating a Hopf bifurcation starting from a converged solution at a different parameter value required about 30 min. Once started, the curves were traced automatically. The two bifurcations seen in Figure 1 were initially tracked, and it was found that (for this mesh) the centro-symmetric

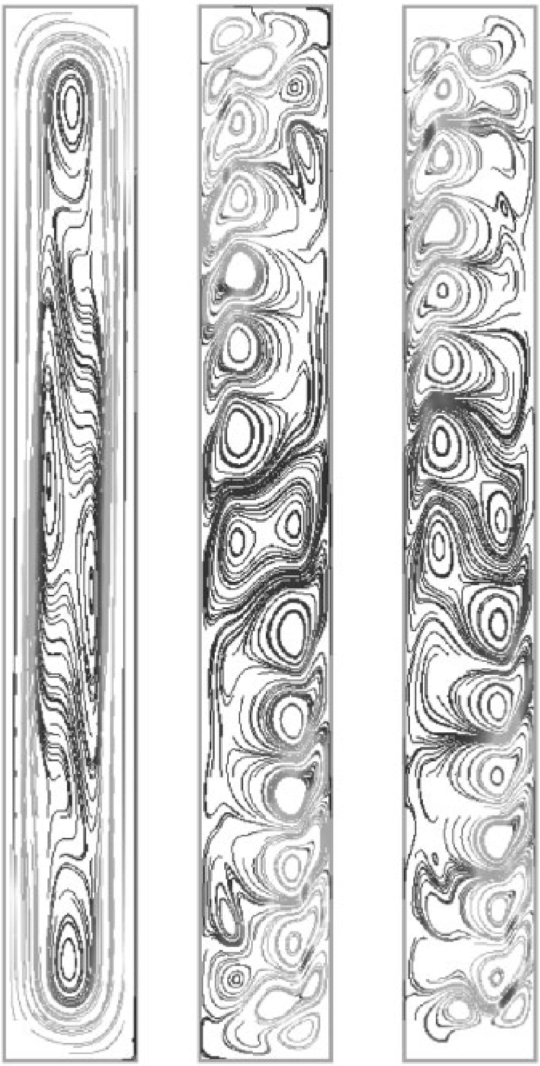

Figure 2. Three streamline plots are shown: the first is the solution at the first Hopf bifurcation at $Ra = 3.61 \times 10^5$, the second is the centro-symmetric eigenfunction at that point, and the third is the symmetry-breaking eigenfunction at the second Hopf bifurcation at $Ra = 3.86 \times 10^5$.

bifurcation (labelled CS(8)) always occurs at lower parameter values than the symmetry-breaking (labelled SB(8)). Eigenvalue calculations at aspect ratios of 7.0 and 9.0 revealed that other modes had overtaken these modes. Tracking the locus of neutral stability points of these two symmetry-breaking modes (labelled SB(7) and SB(9)) show how the leading destabilizing mode transitions from the SB(7) to the CS(8) to the SB(9) mode. The two transition points (near aspect ratios of 7.4 and 8.6) are codimension 2 bifurcation points

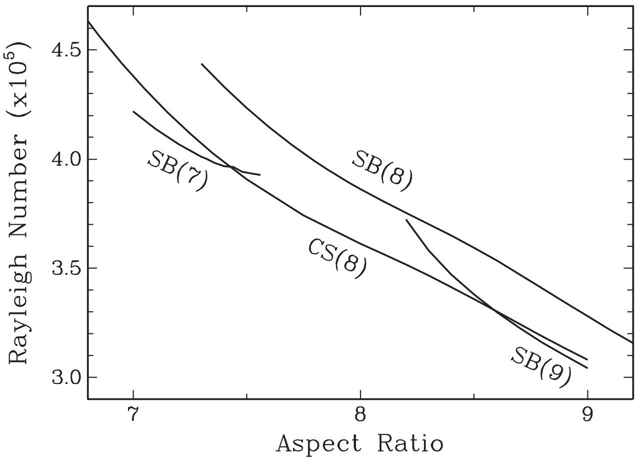

Figure 3. Neutral stability curves showing the locus of Hopf bifurcations for a range of aspect ratios, calculated directly using the Hopf tracking algorithm. The destabilizing mode at an aspect ratio of 8.0 is no longer the destabilizing mode at aspect ratios below 7.4 or above 9.6. The curve labelled CS has centro-symmetric eigenfunctions, and the curve labelled SB has symmetry-breaking eigenfunctions.

where two Hopf bifurcations occur simultaneously. Much more complicated dynamics would be expected if supercritical Rayleigh numbers were studied at these aspect ratios.

3.2. Mesh resolution study

In the previous section, we determined the critical Rayleigh number to be $Ra = 3.61 \times 10^5$ for the mesh of 80×180 elements (corresponding to 58 644 unknowns) with the GLS discretization. The destabilizing mode was found to be centro-symmetric. The results of a mesh resolution study to see how the critical Rayleigh number for this mode depends on mesh resolution is shown in Figure 4. Seven predictions are shown, two for coarser meshes, and four at finer meshes then the previous results. Since the meshes are not uniform, but graded towards the edges, we chose a measure of the mesh spacing $h = 1/(\sqrt{N})$, where N is the number of total unknowns for that given mesh. For the mesh refinement study, no eigenvalue calculations were needed, just the Hopf tracking algorithm. For each successively finer mesh, the solution vector and the two eigenvectors were interpolated from the previous mesh and used as initial guesses (along with the previous values for the Rayleigh number and frequency). In this way, the critical Rayleigh number for this mode was directly located in 4–5 N iterations for each new mesh. This is a great savings in both compute and user time over locating the bifurcation points with just an eigenvalue approximation capability.

The number of unknowns for the seven meshes were 23 940, 43 508, 58 644, 102 212, 178 020, 302 036, and 708 292. The critical Rayleigh number on the finest mesh of 256×688 elements was found to be $Ra = 3.156 \times 10^5$. This calculation required four Newton iterations to converge and took 3.5 h on 160 processors. The convergence of the critical Rayleigh number with mesh is shown to be second order in Figure 5. Here, the error in the critical Rayleigh number (using $Ra = 3.115 \times 10^5$ as the reference solution, as computed in Section 3.3) is plotted against the mesh spacing on a log–log plot and found to have a slope of 2.

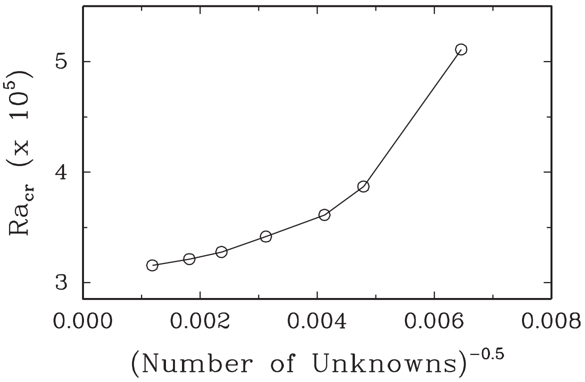

Figure 4. Convergence study of the location of the critical Rayleigh number for the Hopf bifurcation with mesh resolution. The x-axis is an estimate of a mesh spacing, h. Starting with the third coarsest mesh, which had a critical $Ra = 3.61 \times 10^5$ and has solutions shown in Figure 2, the results on the final meshes were calculated directly using the Hopf tracking algorithm using the results of a mesh interpolation utility. The finest mesh has 708 292 unknowns and has a critical Rayleigh number of $Ra = 3.156 \times 10^5$.

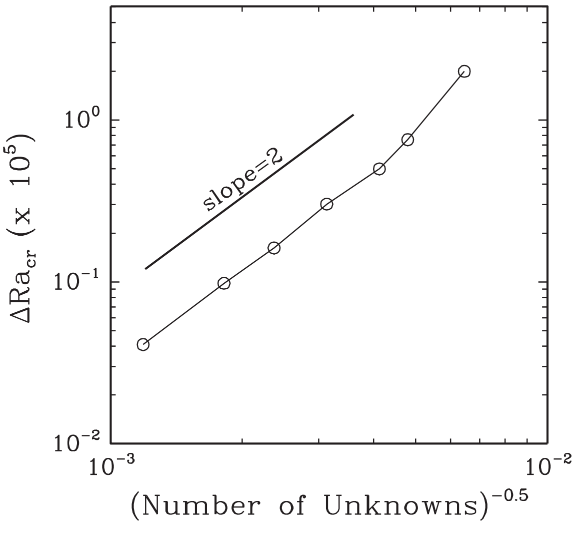

Figure 5. The same mesh convergence data for the critical Rayleigh number is plotted on a log–log plot to demonstrate the second-order convergence rate. The reference solution of $Ra = 3.115 \times 10^5$ is taken from the results described in Section 3.3.

To verify the results of the tracking algorithm, an eigenvalue approximation run was performed at these conditions. Using an Arnoldi space of size 150 and Cayley parameters of $\sigma = -\mu = 1000$, the largest 27 eigenvalues were converged within the specified tolerance in less then 4 h on 128 processors. A plot of the computed eigen spectrum together with data for the five rightmost pairs is shown in Figure 6. While there is a purely imaginary pair of

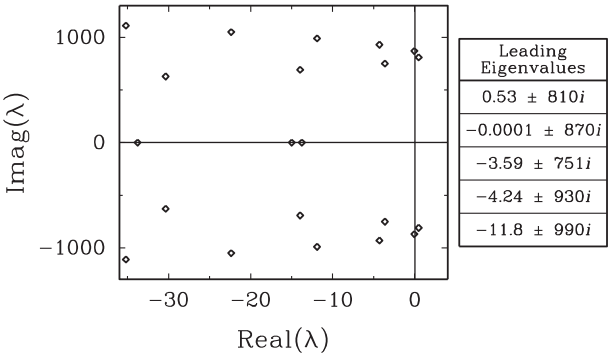

Figure 6. Part of the eigen spectrum for the finest mesh at $Ra = 3.156 \times 10^5$, where the Hopf bifurcation tracking algorithm located the instability. In addition to the expected pair of eigenvalues on the imaginary axis, we can see that another complex conjugate pair of eigenvalues is in the right half-plane. The five rightmost eigenpairs are tabulated.

eigenvalues as expected, the results show a complex conjugate pair of eigenvalues with positive real part. This results shows that, as the mesh was refined, the symmetry-breaking mode overtook the centro-symmetric mode as the first destabilizing mode. This is not a failing of the Hopf tracking algorithm, which successfully converged to a solution based on the initial guess it was given, yet points to the need to have both the bifurcation tracking and the complementary eigenvalue approximation capabilities. The Hopf algorithm was relaunched using these unstable modes for the **y** and **z** vectors, and converged to a critical Rayleigh number of $Ra = 3.115 \times 10^5$.

3.3. *Effects of upwinding*

All of the previous results were obtained with the GLS formulation, which includes the convective stabilization terms that are essentially equivalent to the streamwise upwinding Petrov–Galerkin (SUPG) method. To study the effect of the upwinding terms on the prediction of the instability for this highly convective flow, we turned off the upwinding terms in our formulation. This results in a pressure stabilized Galerkin (PSG) formulation for which we recalculate the critical Rayleigh number. For the finest mesh, there was no significant difference in the number of iterations needed to solve the linear sub-problems for the two formulations. A comparison of the critical Rayleigh number for the three finest meshes for both the GLS solution (with SUPG terms) and the PSPG formulation (without upwinding) are shown in Table III. In addition, the extrapolated value of the critical Rayleigh number as $h \rightarrow 0$ is shown in the final row of the table. This value is obtained using the first and third rows, where the mesh has been doubled in each direction, and assuming $O(h^2)$ convergence for both schemes.

From the extrapolated values, there is strong evidence that both formulations are converging to the same prediction of the critical Rayleigh number. This confirms the expectation that the effect of the upwinding terms disappears as $h \rightarrow 0$. The prediction of the PSG formulation is found to be much more mesh insensitive than the upwinded solution. From these results,

Table III. Location of the first critical Hopf bifurcation as a function of mesh resolution for the GLS formulation, which includes the SUPG upwinding terms, and the PSPG formulation, where upwinding terms are NOT included in formulation. The third mesh has double the elements in each direction of the first mesh, so an $O(h^2)$ extrapolation of the critical value is shown in the final row.

# of Unknowns	$\mathrm{Ra_{cr}}$ SUPG	$\mathrm{Ra_{cr}}$ PSPG
178 020	$3.2637\times10^{5*}$	3.0768×10^5
302 036	$3.1833\times10^{5*}$	3.0678×10^5
708 292	3.1153×10^5	3.0645×10^5
Extrapolation	3.0658×10^5	3.0604×10^5

*These numbers were calculated after the conference in June 2001 to better compare the convergence of the same mode with two different formulations.

our best estimate for the critical Rayleigh number is the extrapolated value from the PSPG formulation: $Ra = 3.0604\times10^5$.

The period of an oscillatory solution starting at this point is given by $(2\pi)/\omega$, which (in the time units given in the problem description) is $T = 3.67$. The stability analysis algorithms in LOCA do not presently include the capability of determining whether the bifurcation is subcritical or supercritical, which is to say, whether a stable branch of oscillating solutions emanates from the bifurcation point, though such algorithms are being developed [30]. The second mode to go unstable is the centro-symmetric mode, which was found on the finest mesh with the PSPG formulation to bifurcate at $Ra = 3.115\times10^5$ with a period of $T = 3.39$. This is the same mode that was tracked in the initial mesh resolution study in Section 3.2, and so this value of the critical Rayleigh number was used as the reference value in determining order of convergence.

4. SUMMARY AND CONCLUSIONS

Stability analysis algorithms have been used to locate the Hopf bifurcations where steady flow of a Boussinesq fluid in an 8:1 thermal cavity goes unstable. The set of four coupled PDEs are discretized using a Galerkin/least-squares formulation for unstructured grids on parallel computers as implemented in the MPSalsa code. A fully coupled inexact Newton method together with a preconditioned Krylov iterative solver from the Aztec package are used to solve directly for steady-state solutions on a parallel computer. An eigenvalue approximation capability, based on the Cayley transformation and the ARPACK library, and which has been implemented to work with iterative linear solvers, is used to initially locate the bifurcations. A Hopf tracking algorithm form the LOCA library is then used, for one mesh, to track out the neutral stability curves as a function of the aspect ratio. Two double Hopf bifurcations are found to exist nearby in parameter space.

The Hopf tracking algorithm is also used to perform mesh resolution studies on the critical Rayleigh number without the need of searching for the bifurcations with the eigensolver. We find that the GLS formulation, which includes SUPG-type upwinding terms, would require a finer mesh to predict the instability to the same accuracy as the PSG formulation, which does not include upwinding terms.

Our best prediction for the critical Rayleigh number for the onset of unsteady convection is $Ra = 3.0604 \times 10^5$. Although our linearized system about the steady-state predicts a period of 3.67, our methods do not speak to the stability of the oscillatory branch that starts at this critical Rayleigh number or to the existence of any stable periodic orbits.

The methods described here, which use the MPSalsa, Aztec, ARPACK, and LOCA libraries, have been implemented to be scalable on distributed memory parallel computers and to work with unstructured grid discretizations in two or three dimensions. The same code can be used to aid in the design of systems with chemical reactions, variable physical properties, and with complex geometries.

We believe that stability analysis algorithms are the appropriate tool for locating the parameter value for the onset of an instability, and tracking how the instability is dependent on a second system parameter. This is the crucial information needed when trying to design a system to operate on a given side of the instability (usually the steady side). When the dynamical behavior of an unsteady system is desired, an efficient and accurate transient capability is the appropriate tool, though having a stability analysis capability can be valuable in that case as well, to help target the transient runs.

ACKNOWLEDGEMENTS

The authors would like to thank Louis Romero for providing important insight into the importance of symmetry in the problem, and Mark Christon for his guidance on what issues to focus a stability analysis capability. Funding for this work was provided by the Department of Energy, both through the Office of Science's MICS program and the NNSA's ASCI program.

REFERENCES

1. Christon MA, Gresho PM, Sutton SB. Computational predictability of natural convection flows in enclosures. *International Journal for Numerical Methods in Fluids* 2002; **40**:953–980.
2. Cliffe KA, Spence A, Tavener SJ. The numerical analysis of bifurcation problems with application to fluid mechanics. *Acta Numerica* 2000; 39–131.
3. Govaerts W. *Numerical Methods for Bifurcations of Dynamic Equilibria*. SIAM: Philadelphia, PA, 2000.
4. Shadid JN. A fully-coupled Newton–Krylov solution method for parallel unstructured finite element fluid flow, heat and mass transport. *International Journal of Computational Fluid Dynamics* 1999; **12**:199–211.
5. Shadid JN, Moffat HK, Hutchinson SA, Hennigan GL, Devine KD, Salinger AG. MPSalsa: a finite element computer program for reacting flow problems: Part I—theoretical development. *Sandia National Laboratories Technical Report*, SAND95-2752, 1996.
6. 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.
7. Hutchinson SA, Prevost L, Shadid JN, Tong C, Tuminaro RS. Aztec User's Guide Version 2.0. *Sandia National Laboratories Technical Report*, SAND99-8801J, 1999.
8. 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. *Journal of Crystal Growth* 1999; **203**:516–533.
9. Pawlowski RP, Salinger AG. Numerical simulation of the partial catalytic oxidation of ethane to ethylene in short contact time reactors. *Sandia National Laboratories Technical Report*, SAND2001-1338, 2001.
10. Maschhoff KJ, Sorensen DC. P_ARPACK: an efficient portable large scale eigenvalue package for distributed memory parallel architectures. In *Applied Parallel Computing in Industrial Problems and Optimization*, Wasniewski J, Dongarra J, Madsen K, Olesen D (eds), Lecture Notes in Computer Science. Springer-Verlag: Berlin, 1996.
11. Lehoucq RB, Sorensen DC, Yang C. *ARPACK USER'S GUIDE*: *Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods.* SIAM: Philadelphia, PA, 1998.
12. Lehoucq RB, Salinger AG. Large-scale eigenvalue calculations for stability analysis of steady flows on massively parallel computers. *International Journal for Numerical Methods in Fluids* 2001; **36**:309–327.

13. Burroughs EA, Romero LA, Lehoucq RB, Salinger AG. Large-scale eigenvalue calculations for computing the scalability of buoyancy driven flows. *Sandia National Laboratories Technical Report*, SAND2001-0113, 2001.
14. Salinger AG, Bou-Rabee NM, Burroughs EA, Lehoucq RB, Pawlowski RP, Romero LA, Wilkes ED. LOCA 1.0: Theory and implementation manual. *Sandia National Laboratories Technical Report*, SAND2002-0396, 2002. http://www.cs.sandia.gov/LOCA.
15. Salinger AG, Pawlowski RP, Romero LA. Scalable bifurcation analysis algorithms for large parallel applications. In *Computational fluid and Solid Mechanics*, Bathe KJ (ed.), *Proceedings of the 1st Conference on Computational Fluid and Solid Mechanics*. Elsevier: Amsterdam, 2001; 1647–1650.
16. Brooks AN, Hughes JR. Streamline upwind/Petrov–Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier–Stokes equations. *Computational Methods in Applied Mechanics and Engineering* 1982; **32**:199–259.
17. Hughes JR, Franca LP, Balestra M. A new finite element formulation for computational fluid dynamics: V. Circumventing the Babuska–Brezzi condition: a stable Petrov–Galerkin formulation of the Stokes problem accommodating equal-order interpolations. *Computational Methods in Applied Mechanics and Engineering* 1986; **59**:85–99.
18. Shakib F. Finite element analysis of the compressible Euler and Navier–Stokes Equations. *Ph.D. Thesis*, Division of Applied Mechanics, Stanford University, 1989.
19. Tezduyar TE. Stabilized finite element formulations for incompressible flow computations. *Advances in Applied Mechanics* 1992; **28**:1–44.
20. Hendrickson B, Leland R. A user's guide to Chaco, Version 1.0. *Sandia National Laboratories Technical Report*, SAND93-2339, Albuquerque, NM, 1993.
21. Shadid JN, Hutchinson SA, Hennigan GL, Moffat HK, Devine KD, Salinger AG. Efficient parallel computation of unstructured finite element reacting flow solutions. *Parallel Computing* 1994; **23**:1307–1325.
22. Shadid JN, Tuminaro RS, Walker HF. An inexact Newton method for full-coupled solution of the Navier–Stokes equations with heat and mass transport. *Journal of Computational Physics* 1997; **137**:155–185.
23. Smith B, Bjorstad P, Gropp W. *Domain Decomposition*: *Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press: NY, 1996.
24. Lehoucq RB, Salinger AG. Massively parallel linear stability analysis with P_ARPACK for 3D fluid flow modeled with MPSalsa. In *Applied Parallel Computing*, *PARA'98*, Agstrom B, Dongarra J, Elmroth E, Wasniewski J (eds), Lecture Notes in Computer Science, vol. 1541. Springer-Verlag: Berlin, 1998; 286–295.
25. Morzynski M, Afanasiev K, Thiele F. Solution of the eigenvalue problem resulting from global non-parallel flow stability analysis. *Computer Methods in Applied Mechanics and Engineering* 1999; **169**:161–176.
26. Golubitsky M, Schaeffer DG. *Singularities and Groups in Bifurcation Theory*, vol. I. Springer: New York, 1985.
27. Seydel R. *Practical Bifurcation and Stability Analysis*: *From Equilibrium to Chaos*. Springer: New York, 1994.
28. Jepson AD. Numerical Hopf bifurcation. *Ph.D. Thesis*, Caltech, Pasadena, CA, 1981.
29. Day D, Heroux M. Solving complex-valued linear systems via equivalent real formulations. *SIAM Journal of Science and Computers* 2001; **23**(2):480–498.
30. Bou-Rabee NM, Romero LA, Salinger AG. A multi-parameter, numerical stability analysis of a standing cantilever conveying fluid. *SIAM Journal for Applied Dynamical Systems* 2002; at press.