

On the multiple shooting continuation of periodic orbits by Newton-Krylov methods.

Juan Sánchez * and Marta Net

Departament de Física Aplicada, Universitat Politècnica de
Catalunya, Mòdul B4. Campus Nord, 08034 Barcelona, Spain

January 14, 2010

Abstract

The application of the multiple shooting method to the continuation of periodic orbits in large-scale dissipative systems is analyzed. A preconditioner for the linear systems which appear in the application of Newton's method is presented. It is based on the knowledge of invariant subspaces of the Jacobians at nearby solutions. The possibility of speeding up the process by using parallelism is studied for the thermal convection of a binary mixture of fluids in a rectangular domain, with positive results.

KEYWORDS: Continuation methods, periodic orbits, Poincaré maps, multiple shooting, parallelism, variational equations, Krylov methods, periodic Schur decomposition, Krylov-Schur method

1 Introduction

Continuation methods have become a basic tool in the numerical analysis of nonlinear dynamical systems depending on parameters, since their introduction in the 1960s in several different fields. Designed in principle for the computation of fixed points, they are now also used for the study of other invariant manifolds, for instance periodic orbits or connecting manifolds, and for the computation of loci of bifurcations (saddle-node, Hopf, etc.) by using extended systems. Different implementations of these tools are described in several textbooks and lecture notes [Rheinboldt, 1986; Keller, 1987; Simó, 1990; Allgower & Georg, 1990; Seydel, 1994; Kuznetsov, 1998; Govaerts, 2000; Doedel, 2007], and related continuation and bifurcation software packages, such as AUTO [Doedel,

*Corresponding author: e-mail address sanchez@fa.upc.edu; Tel. 34-93-4017981; Fax 34-93-4016090

1986], PITCON [Rheinboldt & Burkardt, 1983], DsTool [Guckenheimer *et al.*, 1995], CONTENT [Kuznetsov & Levitin, 1996], Multifario [Henderson, 2002] (multi-parameter continuation), MatCont [Dhooge *et al.*, 2003], are available.

In a continuation process, most of the computing time is spent in numerical linear algebra calculations. Mainly in solving linear systems of equations at Newton's iterations, and in the computation of eigenvalues and eigenvectors in the study of the stability of the solutions and in the search of new branches of solutions at bifurcation points. Direct methods are employed for low-dimensional systems. The extension to high-dimensional problems, most of them coming from the discretization of partial differential equations, is not trivial. The main obstacle is, precisely, the implementation of the numerical linear algebra. In many cases the matrices are not directly available but only their actions. In these cases *matrix-free* methods are required. Methods requiring the transpose of a linear operator might be useless, because it is possible that neither the matrix of the operator nor the action by its transposed be available, unless also the adjoint operator can be discretized [Sánchez *et al.*, 2006]. An example is provided by the computation of periodic orbits. The Jacobian of the Poincaré map is a full matrix. It might be impossible to store it if the dimension of the system is large, but its action can be computed by integrating a system of twice the original dimension of the system.

The development of modern linear algebra techniques during the 1980s and 1990s, in many cases based on Krylov or Arnoldi methods [Saad, 1992; 1996], has allowed the study of large-scale systems; many of them dealing with the computation of steady states in fluid dynamics or reaction-diffusion problems. Inexact Newton-Krylov methods [Dembo *et al.*, 1982] are used to find the fixed points, and subspace iteration or Arnoldi methods to study their stability (see [Edwards *et al.*, 1994; Böhmer *et al.*, 2000; Lopez *et al.*, 2001] among many others). One of the main difficulties is to achieve a fast convergence of the linear solvers; in many cases the Generalized Minimal Residual Method, GMRES [Saad & Schultz, 1986]. To accelerate them, some kind of preconditioning is frequently used. In the case of discretizations by finite differences, finite volumes, or finite elements, the use of incomplete LU decompositions usually provides efficient preconditioners, because of the sparsity of the matrices involved (see [Saad, 1994; Molemaker & Dijkstra, 2000; Sánchez *et al.*, 2002] to mention only a few). For spectral discretizations of incompressible fluid problems, the use of the Stokes operator as preconditioner is suggested in [Edwards *et al.*, 1994]. When a pseudo-spectral method is employed, the preconditioners based on discretizations by finite differences or finite elements, on the same mesh, are a very efficient possibility as it is shown in [Canuto *et al.*, 2007]. To summarize, it can be stated that the continuation of steady solutions and its bifurcation loci of large-scale systems has now become a known tool among the scientific computing community. The only available software package is LOCA [Salinger *et al.*, 2002], which allows tracking steady solutions, pitchfork and Hopf bifurcations, and a more specific phase transition tracking.

The continuation of periodic orbits in large-scale dissipative systems has only been

tackled recently by Newton-Picard algorithms [Lust *et al.*, 1998], implemented in the package PDECONT, limited memory Broyden methods [van Noorden *et al.*, 2004], and Newton-Krylov methods [Sánchez *et al.*, 2004]. In this case the matrices are always dense, independently of the discretization method used for the governing partial differential equations. As far as we know, there has been no attempt to develop specific preconditioners for this kind of computations.

A new numerical algorithm for the continuation of periodic orbits of high-dimensional dissipative dynamical systems, using multiple shooting and parallelism, is presented in this article. The equations of the multiple shooting are solved by Newton-Krylov methods, but it will be seen that a direct application, with each partial shoot computed in a different processor, does not provide an important speedup. To achieve a linear speedup some kind of preconditioning for the linear systems must be used. We show how a preconditioner can be constructed from the information on the stability of nearby periodic orbits. This information is available from the continuation and bifurcation analysis, since the stability is computed frequently. Therefore the preconditioner can be obtained at a low extra cost. The codes can be designed so that its computation does not interfere with that of the periodic orbits. Two implementations are suggested, although other ways are possible. We focus on the problem of accelerating the computation of the periodic orbits. The parallelization of the computation of the stability and the preconditioner, which is straightforward, is not considered here.

The idea of using the information on the stability to accelerate the convergence of variants of the Picard iteration was described in [Shroff & Keller, 1993], for the case of fixed points, and in [Lust *et al.*, 1998] for the computation of periodic orbits. We employ this information for preconditioning the linear systems in the Newton's iterations. The preconditioner described here was used in [Erhel *et al.*, 1996] to accelerate the convergence of the restarted version of GMRES [Saad & Schultz, 1986], (GMRES(M)), in order to make its convergence closer to that of the full version. The approximations to the partial Schur decompositions were obtained after each cycle of the restarted GMRES. Our case is different since this information is obtained from the calculation of the stability of a previous periodic solution, which is useful by itself. Moreover, the number of iterations is so small that, in almost all cases, the GMRES solver ends before restarting.

More recently, similar ideas based on invariant subspaces have been applied. In [Parks *et al.*, 2006], sequences of linear systems are considered, with small differences from one to the next. They use the information of the Krylov subspace generated during the solution of a linear system to preconditioning the next. The authors refer to this process as *Krylov subspace recycling*. In [Carpentieri *et al.*, 2007], an initial preconditioner, used as the smoother in the framework of an algebraic multigrid solver, is improved by using information on invariant subspaces of the preconditioned system.

The test problem shows that linear speedups are attainable very efficiently for a moderate number of intermediate sections, m . We explain why the method is efficient if

m is not too large. The message passing library MPI [Message Passing Forum, 1994] has been used, having in mind distributed memory machines, although codes using this method could also be run very efficiently on the emerging computer architectures, based on shared-memory multi-core processors, using MPI or OpenMP [Chapman *et al.*, 2007], provided the memory access does not become a serious bottleneck.

The layout of the paper is as follows, Sec. 2 describes the multiple shooting algorithm, in Sec. 3 we recall some results on the cyclic matrices of the linear systems which appear in this study, and on their spectra. In Sec. 4 the preconditioner, and different ways of obtaining it are described. Section 5 describes the test problem employed to show the benefits of preconditioning, and what happens if no preconditioner is used. The results for the preconditioned computations are described in Sec. 6. Finally, the paper closes, in Sec. 7, with further comments, a summary of the results, and a brief description of some possible extensions.

2 Multiple Shooting

Consider a system of autonomous differential equations

$$\dot{x} = f(x, \lambda), \quad (1)$$

$(x, \lambda) \in \mathcal{U} \times I \subset \mathbb{R}^{n+1}$, depending on a single parameter λ , with I an open interval. Suppose that (1) has periodic orbits for $\lambda \in I$, and that $\Pi_i, i = 1, \dots, m$, are m hyperplanes, which intersect transversally an initial periodic orbit $x^0(t)$. Let $x_i^0 \in \Pi_i$ be the intersections, \mathcal{U}_i an open neighborhood of x_i^0 in Π_i , $P_i : \mathcal{U}_i \times I \subset \Pi_i \times I \rightarrow \Pi_{i+1}$ (with $\Pi_{m+1} = \Pi_1$) the partial Poincaré maps from \mathcal{U}_i to Π_{i+1} (see Fig. 1), and $P : \mathcal{U}_1 \times I \subset \Pi_1 \times I \rightarrow \Pi_1$ the full Poincaré map. It is clear that $P(x_1, \lambda) = (P_m \circ P_{m-1} \circ \dots \circ P_1)(x_1, \lambda)$ in a neighborhood of x_1^0 , and that $D_x P(x_1, \lambda) = D_x P_m(x_m, \lambda) D_x P_{m-1}(x_{m-1}, \lambda) \dots D_x P_1(x_1, \lambda)$, if $x_{i+1} = P_i(x_i, \lambda)$, $i = 1, \dots, m-1$. We also define $X = (x_1, \dots, x_m) \in \mathbb{R}^{mn}$, and the maps \mathcal{P} , and $\mathcal{G} = \mathcal{I} - \mathcal{P}$ from $\mathcal{U}_1 \times \dots \times \mathcal{U}_m \times I$ to $\Pi_1 \times \dots \times \Pi_m$, as

$$\mathcal{P}(X, \lambda) = (P_m(x_m, \lambda), P_1(x_1, \lambda), \dots, P_{m-1}(x_{m-1}, \lambda)),$$

and

$$\mathcal{G}(X, \lambda) = X - \mathcal{P}(X, \lambda) = (x_1 - P_m(x_m, \lambda), x_2 - P_1(x_1, \lambda), \dots, x_m - P_{m-1}(x_{m-1}, \lambda)).$$

The points x_1, \dots, x_m are on a periodic orbit if

$$\mathcal{G}(X, \lambda) = 0. \quad (2)$$

It must be noticed that each x_i in Eq. (2) is parameterized by $n-1$ coordinates because it lies on the known hyperplane Π_i , and the same holds for the $P_i(x_i, \lambda)$. Instead of working

with the maps P_i defined from Π_i to Π_{i+1} , we work with \bar{P}_i defined from a reference hyperplane $\bar{\Pi}_i$ to another, $\bar{\Pi}_{i+1}$ as depicted in Fig. 2. The maps $\bar{P}_i(x_i, \lambda)$ are defined as $\bar{P}_i(x_i, \lambda) = R_{i+1}(P_i(R_i^{-1}(\bar{x}_i), \lambda))$, R_i and R_{i+1} being the orthogonal projectors from Π_i and Π_{i+1} onto $\bar{\Pi}_i$ and $\bar{\Pi}_{i+1}$, respectively. The hyperplanes $\bar{\Pi}_i$ have equations $x_{k_i} = 0$, k_i being the index of the largest component of the vector orthogonal to Π_i . The action of the Jacobian of \bar{P}_i can be trivially computed from that of P_i by applying the chain rule to the definition of \bar{P}_i .

If λ is fixed, Eq. (2) completely determines the intersections of the periodic orbit with the hyperplanes. The period of the orbit is obtained as a byproduct; if x_i are the solutions of (2), and T_i is the integration time from x_i to $P_i(x_i, \lambda)$, then the period of the orbit is $T = \sum_{i=1}^m T_i$. If, as in the example in Sect. 5, the time integrator is a multistep one, T_i and $P_i(x_i, \lambda)$ can be computed by interpolation once the hyperplane Π_{i+1} has been crossed.

In the context of a continuation method, (2) is complemented by an equation of the form

$$U^\top(X - X_0) + u_\lambda(\lambda - \lambda_0) = 0, \quad (3)$$

(X_0, λ_0) being a prediction of a new point on the curve of solutions, parametrized, for instance, by an approximation to its arclength (pseudo-arclength continuation), and $(U, u_\lambda) = (u_1, \dots, u_m, u_\lambda)$ is taken such that Eq. (3) defines a hyperplane, which intersects transversally the curve of solutions (for instance, a prediction of the tangent to the curve). Then the system (2)-(3) determines locally a unique new solution pair (X, λ) . This is the system which defines the multiple shooting method in a continuation context. A particular case of interest corresponds to the selection $U = 0$ and $u_\lambda = 1$; it fixes the parameter, and gives rise to parameter continuation, which can be used away from turning points.

To solve the system (2)-(3) by a Newton-Krylov method, the action by its Jacobian

$$\begin{pmatrix} \mathcal{I} - D_X \mathcal{P} & -D_\lambda \mathcal{P} \\ U^\top & u_\lambda \end{pmatrix} = \begin{pmatrix} I & \dots & \dots & 0 & -A_m & -b_m \\ -A_1 & \ddots & & & 0 & -b_1 \\ 0 & \ddots & \ddots & & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ 0 & \dots & 0 & -A_{m-1} & I & -b_{m-1} \\ u_1^\top & \dots & u_{m-2}^\top & u_{m-1}^\top & u_m^\top & u_\lambda \end{pmatrix}, \quad (4)$$

with $A_i = D_x P_i(x_i, \lambda)$ and $b_i = D_\lambda P_i(x_i, \lambda)$, is required. For instance, the action of $\mathcal{I} - D_X \mathcal{P}(X, \lambda)$ on a vector $V = (v_1, \dots, v_m)$ is

$$(v_1 - D_x P_m(x_m, \lambda)v_m, v_2 - D_x P_1(x_1, \lambda)v_1, \dots, v_m - D_x P_{m-1}(x_{m-1}, \lambda)v_{m-1}).$$

The computation of $\mathcal{P}(X, \lambda)$ involves the time integration of (1) with m initial conditions, $x(0) = x_i$, which can be done in parallel. The calculation of $D\mathcal{P}(X, \lambda)(V, \mu)$ is also fully

parallelizable, and involves the time integration of the system of first variational equations

$$\dot{v} = D_x f(x, \lambda)v + D_\lambda f(x, \lambda)\mu, \quad (5)$$

with initial conditions $v(0) = v_i$, together with the original system (1), with $x(0) = x_i$.

In the case of parameter continuation the linear systems to be solved have matrices of the form

$$\begin{pmatrix} \mathcal{I} - D_X \mathcal{P} & 0 \\ 0 & 1 \end{pmatrix}, \quad (6)$$

because we fix λ to the predicted value, the term $D_\lambda f(x, \lambda)\mu$ is not included in the integration of (5), and we do not allow increments in λ during Newton's method.

For low-dimensional systems, and not very unstable periodic orbits, it is common to solve the systems with matrix $\mathcal{I} - D_X \mathcal{P}$ by a condensation process [Ascher *et al.*, 1995]. It requires to form the product $D_x P(x_1, \lambda) = D_x P_m(x_m, \lambda) D_x P_{m-1}(x_{m-1}, \lambda) \cdots D_x P_1(x_1, \lambda)$, which is computed with each matrix $D_x P_i(x_i, \lambda)$ being calculated on a different processor. Then Newton's method can be applied to obtain x_1 . For large-scale problems the only tools available are the computation of the vectors $\mathcal{P}(X, \lambda)$ and $D\mathcal{P}(X, \lambda)(V, \mu)$. The matrices $D_x P_i(x_i, \lambda)$ are never computed explicitly, and it can be even impossible to store them. Therefore, as far as we know, there is no way to extend the method to high-dimensional systems.

When using multiple shooting in parallel, the CPU times spent in the integration of equations (1) or (5), from each Poincaré section to the next, must be essentially the same to avoid regions at which the code behaves as serial. This synchronization can be accomplished by a careful selection of the sections, which can be changed adaptively during the continuation process. In our test problem, we use a fixed time step integrator, except for a very few initial steps. Then the CPU time it takes to compute any integration is proportional to the integration time interval. Therefore the hyperplanes have been selected to be normal to an initial periodic orbit, and at equally spaced times. Changing them involves the integration of a periodic orbit during its period. If it is possible, the first hyperplane can be kept fixed to have consistent outputs.

3 Block Cyclic Matrices

In this section some results for block cyclic matrices are recalled. They state the relationship between the eigenpairs of $D_x P(x_1, \lambda)$ and those of $D_X \mathcal{G}(X, \lambda)$, and provide a way to obtain an invariant subspace of $D_X \mathcal{G}(X, \lambda)$ if an invariant subspace of $D_x P(x_1, \lambda)$ is known. These results will be used in the next sections.

Consider the matrices

$$\mathcal{A} = \begin{pmatrix} 0 & \dots & \dots & 0 & A_m \\ A_1 & \ddots & & & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & A_{m-1} & 0 \end{pmatrix},$$

and $A = A_m A_{m-1} \cdots A_1$, with $\mathcal{A} \in \mathbb{R}^{mn \times mn}$, and $A_i \in \mathbb{R}^{n \times n}$. The matrix \mathcal{A} is said to have a block cyclic structure. The next results establish the relation between the eigenpairs of A and \mathcal{A} , and a way to obtain an orthonormal basis of an invariant subspace of \mathcal{A} from a partial periodic Schur decomposition of A . Their proofs are straightforward.

Proposition 1 *If A and \mathcal{A} are defined as above then*

- a) *if (μ, V) is an eigenpair of \mathcal{A} , with $\mu \in \mathbb{C}$, and $V = (v_1, \dots, v_m)^\top$ with $v_i \in \mathbb{C}^n$ ($i = 1, \dots, m$), then (μ^m, v_1) is an eigenpair of A .*
- b) *if (λ, u_1) is an eigenpair of A , with $\lambda \in \mathbb{C}$ and $u_1 \in \mathbb{C}^n$, and if $\mu^m = \lambda$, then (μ, V) , with $V = (\mu^{m-1}u_1, \mu^{m-2}u_2, \dots, \mu u_{m-1}, u_m)^\top$, and $u_{i+1} = A_i u_i$ ($i = 1, \dots, m-1$), is an eigenpair of \mathcal{A} .*

This result states that the eigenvalues of \mathcal{A} are on circles centered at the origin, and then, those of $\mathcal{I} - \mathcal{A}$ on circles centered at $+1$.

Proposition 2 *Consider now a partial periodic Schur decomposition of $A = A_m A_{m-1} \cdots A_1$. Let $Q_1, \dots, Q_m \in \mathbb{R}^{n \times k}$ be matrices such that $Q_i^\top Q_i = I_k$ ($I_k \in \mathbb{R}^{k \times k}$ being the identity matrix of dimension $k < n$), $R_1, \dots, R_{m-1}, R_m \in \mathbb{R}^{k \times k}$, the first $m-1$ being upper triangular, and R_m upper block triangular, with diagonal 1×1 and 2×2 blocks corresponding, respectively, to real and pairs of complex conjugate eigenvalues, verifying*

$$A_i Q_i = Q_{i+1} R_i, \quad i = 1, \dots, m-1, \quad (7)$$

$$A_m Q_m = Q_1 R_m. \quad (8)$$

Then it follows that

$$\mathcal{A} \mathcal{Q} = \mathcal{Q} \mathcal{R}, \quad (9)$$

with

$$\mathcal{Q} = \begin{pmatrix} Q_1 & \dots & \dots & \dots & 0 \\ \vdots & \ddots & & & \vdots \\ \vdots & & \ddots & & \vdots \\ \vdots & & & \ddots & \vdots \\ 0 & \dots & \dots & \dots & Q_m \end{pmatrix}, \quad \text{and} \quad \mathcal{R} = \begin{pmatrix} 0 & \dots & \dots & 0 & R_m \\ R_1 & \ddots & & & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \dots & 0 & R_{m-1} & 0 \end{pmatrix},$$

$Q \in \mathbb{R}^{mn \times mk}$, and $\mathcal{R} \in \mathbb{R}^{mk \times mk}$.

Equation (9) states that the columns of Q form an orthonormal basis of an invariant subspace of \mathcal{A} , of dimension mk . It is also clear that

$$(\mathcal{I} - \mathcal{A})Q = Q(\mathcal{I} - \mathcal{R}). \quad (10)$$

The matrix Q_1 is such that $AQ_1 = Q_1R$ is a partial real Schur decomposition of A , with $R = R_m R_{m-1} \cdots R_2 R_1$. If this decomposition is known, the periodic Schur decomposition can be computed by calculating $A_i Q_i$ and reorthogonalizing to obtain Q_{i+1} and R_i for $i = 1, \dots, m-1$, and finally $R_m = Q_1^\top A_m Q_m$, although this is not the most accurate way.

The periodic Schur decomposition, and more accurate algorithms to compute it are described in [Bojanczyk *et al.*, 1992], [Varga & Dooren, 2001], and [Kressner, 2005]. See also the review [Watkins, 2005] on product eigenvalue problems. In the context of the multiple shooting computation of periodic orbits it has been used in [Lust, 1997] in the framework of the Newton-Picard method, and in [Lust, 2001] to improve the computation of Floquet multipliers. The software psSchur [Lust, 2000], which computes the full periodic Schur decomposition, is available. The subroutine MB03WD of the LAPACK-based library SLICOT [Benner *et al.*, 1999] also computes it. Unfortunately, as far as we know, there is no available software to compute partial periodic Schur decompositions. However, the Krylov-Schur algorithm is described in detail in [Kressner, 2006].

4 Preconditioners Based on Leading Invariant Subspaces

In this section a preconditioner, which can be used to accelerate the convergence of the linear solvers, is described. It will be shown in Sect. 6 that its use helps to obtain linear speedups when using parallel shooting.

Let us consider a generic linear system $Cx = b$, with non-singular matrix $C \in \mathbb{R}^{n \times n}$. We look for a right preconditioner, M , based on the knowledge of approximate invariant subspaces of C , such that the convergence of the iterative methods applied to

$$CM^{-1}y = b, \quad x = M^{-1}y, \quad (11)$$

be faster. Since, for the matrices of the form $\mathcal{I} - \mathcal{A}$ we will consider, most of the spectrum is clustered around $+1$, we assume this situation for C .

Let the columns of $Q = [q_1, \dots, q_k] \in \mathbb{R}^{n \times k}$ form an orthonormal basis of an invariant subspace of C corresponding to the first k leading (maximal distance to $+1$) eigenvalues of C , with $Q^\top Q = I_k$, and $k \ll n$. It can be obtained by any of the methods previously described. The matrix Q verifies

$$CQ = QR, \quad \text{and} \quad C^{-1}Q = QR^{-1}, \quad (12)$$

with R an invertible $k \times k$ matrix. Then, we define M as

$$M = QRQ^\top + (I - QQ^\top), \quad (13)$$

so that

$$M^{-1} = QR^{-1}Q^\top + (I - QQ^\top), \quad \text{and} \quad CM^{-1} = QQ^\top + C(I - QQ^\top). \quad (14)$$

The operator QQ^\top is the orthogonal projector onto $\text{Span}\{q_1, \dots, q_k\}$. Therefore, if $z = Q\alpha$, with $\alpha \in \mathbb{R}^k$, then $CM^{-1}z = z$, and if $Q^\top z = 0$, then $CM^{-1}z = Cz$. So, if $z = z_1 + z_2$ with $z_1 \in \text{Span}\{q_1, \dots, q_k\}$, and $z_2 \in \text{Span}\{q_1, \dots, q_k\}^\perp$, $CM^{-1}z = z_1 + Cz_2$, i.e. CM^{-1} acts as the identity on $\text{Span}\{q_1, \dots, q_k\}$, and as the original C on its orthogonal complement.

In the case of multiple shooting we need preconditioners, M , for matrices of the form $\mathcal{I} - \mathcal{A}$. They can be obtained as described in Sect. 3 from a partial Schur decomposition of $\mathcal{I} - \mathcal{A}$ (see Eq. (10)). Computing the action $CM^{-1}z$ implies k dot products to form $Q^\top z$, $k + 1$ BLAS AXPY operations to calculate $QQ^\top z$ and $(I - QQ^\top)z$, and a matrix product by C . In the case of $\mathcal{I} - \mathcal{A}$, \mathcal{Q} , and \mathcal{R} have block structure, which can be utilized to compute efficiently the DOT and AXPY operations. No reference to the matrix R (or \mathcal{R}) or its inverse is needed during the iterations. It is only needed, at the end, to compute x from y (see Eq. (11)). The effect of the preconditioner for $\mathcal{I} - \mathcal{A}$, is to bring its eigenvalues on the external circles back to a neighborhood of $+1$.

There are several ways to compute the matrix \mathcal{Q} needed for the preconditioner. The first is to compute the leading multipliers, and the corresponding invariant subspaces of the orbit by any variant of the power method: the implicitly restarted Arnoldi method [Sorensen, 1992] for which the ARPACK software [Lehoucq *et al.*, 1998] is available, the Krylov-Schur method [Stewart, 2002], or any variant of the slower but fully parallelizable subspace iteration method. This provides Q_1 in formula (7). Then Q_i and R_i , with $i = 2, \dots, m$, can be computed sequentially by using equations (7) and (8). This is the method we have adopted in our numerical experiments. Another more stable possibility, which should be used if the periodic orbit is so unstable that the leading multipliers cannot be obtained by simple shooting, is to compute Q_i directly by using a recent algorithm [Kressner, 2006], which computes partial periodic Schur decompositions. It is an extension of the Krylov-Schur algorithm, which in turn is a modification of the implicitly restarted Arnoldi algorithm. This latter version is parallelizable. The cost of computing \mathcal{Q} depends on the method used. As \mathcal{Q} and \mathcal{R} have the block structure shown in Sect. 3, \mathcal{Q} only needs to be stored as a set of m matrices of dimension $n \times k$, and \mathcal{R} as m matrices of dimension $k \times k$. The products by \mathcal{Q} or \mathcal{R} are computed by exploiting this block structure.

There are several ways of linking the computation of the stability and the preconditioner with the calculation of the periodic orbits. Two of them are schematized in Fig. 3. A first option (scheme a in Fig. 3) is to have m processors computing the periodic orbits,

and another one calculating the stability, updating the hyperplanes and preparing the preconditioner. As soon as the preconditioner is updated, the last computed solution is sent to the processor which will obtain the leading invariant subspace. When the task is finished, this processor communicates with the main process to obtain the current last computed solution to update the hyperplanes. The preconditioner is then computed and sent to the main processor to be updated, and the full process starts again.

Another possibility (scheme b in Fig. 3) is to use the information of the number of iterations required to solve the linear systems to decide when to renew the preconditioner. When this is needed the m processors stop computing the periodic orbits, and they are used to compute the stability, and to update the hyperplanes and the preconditioner from the last computed solution. In this case a parallelizable method to obtain the invariant subspaces must be used. This second scheme was used in [Sánchez *et al.*, 2002] to compute fixed points in a Fluid Mechanics problem. The preconditioner was an incomplete LU decomposition, and it was found that long continuations could be done without changing the preconditioner. We have checked that the invariant subspaces need not be computed with a high precision for the preconditioner to be effective.

Systems with matrix (4), can be preconditioned with

$$\begin{pmatrix} M & -D_\lambda \mathcal{P} \\ U^\top & u_\lambda \end{pmatrix}, \quad \text{or} \quad \begin{pmatrix} M & 0 \\ 0 & 1 \end{pmatrix}, \quad (15)$$

and those with matrix (6) with the second. The former is a bordered system, which can be solved in a stable way by the BEM algorithm (see [Govaerts, 2000]). It requires solving systems with M and M^\top , and to know explicitly the column $-D_\lambda \mathcal{P}$. Equation (14) shows how to apply M^{-1} , and it is easy to see that $(M^\top)^{-1} = Q(R^\top)^{-1}Q^\top + (I - QQ^\top)$. Finally, the column $-D_\lambda \mathcal{P}$ can be obtained by integrating the first variational equations (5) with initial conditions $v(0) = 0$, and $\mu(0) = 1$.

5 Application of Newton-Krylov Methods to the Multiple Shooting.

In order to perform some numerical experiments we have applied the multiple shooting method to the thermal convection problem of a Boussinesq binary mixture, filling a two-dimensional rectangular domain Ω heated from below. In non-dimensional units $\Omega = [0, \Gamma] \times [0, 1]$, Γ is the width to the height ratio, and x and y are the horizontal and vertical coordinates, respectively.

The problem is governed by the mass, momentum, energy and one of the concentrations (the denser in what follows) equations. The basic conductive and linearly stratified state, which is a solution of the equations for any value of the parameters, is given by zero velocity $\mathbf{v}_b = 0$, and linear profiles for the temperature $T_b = T_b(0) - y$, and the concentration

$C_b = C_b(0) - y$. The values $T_b(0)$ and $C_b(0)$ are constants related by the parameters of the problem.

The equations for the perturbation of the basic state (\mathbf{v}, Θ, C) , in non-dimensional form, are

$$\begin{aligned}\partial_t \mathbf{v} + (\mathbf{v} \cdot \nabla) \mathbf{v} &= \sigma \nabla^2 \mathbf{v} - \nabla p + \sigma Ra (\Theta + SC) \hat{e}_y, \\ \partial_t \Theta + (\mathbf{v} \cdot \nabla) \Theta &= \nabla^2 \Theta + v_y, \\ \partial_t C + (\mathbf{v} \cdot \nabla) C &= L(\nabla^2 C - \nabla^2 \Theta) + v_y, \\ \nabla \cdot \mathbf{v} &= 0.\end{aligned}$$

The problem depends on the non-dimensional Rayleigh, Prandtl, Lewis, and Separation ratio numbers, denoted respectively by Ra , σ , L , and S . The boundary conditions considered are nonslip for the velocity field ($\mathbf{v} = 0$ on $\partial\Omega$), constant temperatures at the top and bottom lids, insulating lateral walls, and non-porous boundaries (see [Batiste *et al.*, 2002] for more details of the formulation). In the continuation experiments we fix $\Gamma = 4$, $\sigma = 0.6$, $L = 0.03$ and $S = -0.1$. The last three values correspond to a mixture of two isotopes of Helium in liquid state. The Rayleigh number, which is proportional to the difference between the bottom and top temperatures, will be the control parameter. If S is below a negative critical value, as it is in the test problem, the primary bifurcation from the basic state is a Hopf bifurcation.

The above equations are rewritten in terms of a stream-function, ψ , i.e., $\mathbf{v} = (-\partial_y \psi, \partial_x \psi)$, and an auxiliary function $\eta = C - \Theta$. They are

$$\begin{aligned}\partial_t \nabla^2 \psi + J(\psi, \nabla^2 \psi) &= \sigma \nabla^4 \psi + \sigma Ra [(S + 1) \partial_x \Theta + S \partial_x \eta], \\ \partial_t \Theta + J(\psi, \Theta) &= \nabla^2 \Theta + \partial_x \psi, \\ \partial_t \eta + J(\psi, \eta) &= L \nabla^2 \eta - \nabla^2 \Theta,\end{aligned}$$

with $J(f, g) = \partial_x f \partial_y g - \partial_y f \partial_x g$, and the boundary conditions become

$$\begin{aligned}\psi = \partial_n \psi = \partial_n \eta &= 0 \quad \text{on} \quad \partial\Omega, \\ \Theta &= 0 \quad \text{on} \quad y = 0, 1, \\ \partial_x \Theta &= 0 \quad \text{on} \quad x = 0, \Gamma.\end{aligned}$$

In this way the incompressibility condition is identically fulfilled, the boundary conditions for Θ and η decouple, and the number of unknowns is reduced.

The group of symmetries of this system is $\mathbb{Z}_2 \times \mathbb{Z}_2$ generated by the reflections R_x and R_y , with respect to the vertical, and horizontal mid-planes, i.e., changing x by $\Gamma - x$ and the sign of ψ , or changing y by $1 - y$ and the sign of all three functions, leaves the system invariant. These symmetries give rise to pitchfork bifurcations of fixed points, and periodic orbits, at which any of the two symmetries can be broken.

To obtain the numerical solutions, the functions ψ , Θ , and η are approximated by a pseudo-spectral method. Collocation on a mesh of $n_x \times n_y = 64 \times 16$ Gauss-Lobatto points has been used in all the calculations shown. This gives a total dimension $n = 3072$. This mesh is enough in the interval of Ra considered because the solutions are quite smooth. A finer resolution with $n_x \times n_y = 128 \times 32$ ($n = 12288$) has also been used to check some of the results.

The stiff system of ODEs obtained after the spatial discretization can be written as $B\dot{u} = Lu + N(u)$, where the vector $u = (\psi_{ij}, \Theta_{ij}, \eta_{ij})$ contains the values of ψ , Θ and η at the mesh of collocation points. The operators L and N represent the linear and non-linear terms in the equations. They are integrated by using fixed-time-step fourth-order BDF-extrapolation formulas

$$\frac{1}{\Delta t} B \left(\gamma_0 u^{n+1} - \sum_{i=0}^{k-1} \alpha_i u^{n-i} \right) = \sum_{i=0}^{k-1} \beta_i N(u^{n-i}) + Lu^{n+1},$$

where the superscripts indicate the step. The coefficients, up to sixth order, are given, for instance, in [Sánchez *et al.*, 2004]. The initial points required to start the time integration are obtained by a fully implicit BDF method. The subroutine DLSODPK of the ODEPACK package [Hindmarsh, 1983] has been used.

Figure 4 shows two branches of solutions for this problem. The horizontal line corresponds to the basic state. It becomes unstable at $Ra = 2075$ at a subcritical Hopf bifurcation giving rise to the branch of periodic orbits, which is initially unstable. It becomes stable at $Ra = 2061$ at a saddle-node bifurcation and loses stability at $Ra = 2067$ at a Neimark-Sacker bifurcation. By following the unstable branch of periodic orbits three pitchfork bifurcations are found at $Ra = 2113$, $Ra = 2168$, and $Ra = 2196$. At them, one of the spatio-temporal reflection symmetries of the periodic orbits is broken.

Table 1 shows the first twenty Floquet multipliers of the periodic orbit at $Ra = 2305$ ordered by their moduli. Three of them, which are real, are outside the unit circle. The complex conjugate multipliers, which crossed the unit circle at the Neimark-Sacker bifurcation, become real when they collapse at the real axis, and the multiplier responsible for the second pitchfork bifurcation gets back into the unit circle at the third. The invariant subspaces, corresponding to this set of eigenvalues, will be used in Sect. 6 to obtain a preconditioner for the linear systems of the multiple shooting. Fig. 5 shows the temporal evolution of this periodic orbit. It contains the contour plots of ψ , θ , and C during a period T in increments of $T/8$. It can be seen that it is a fixed cycle with respect to the transformation R_x , and a symmetric cycle with respect to R_y [Kuznetsov, 1998], and the composition $R_x R_y$. The periodic orbit consists on reversals of the rotating sense of the vortices, as can be seen in the left column of plots, which produce an oscillation of the temperature and the concentration.

The multiple shooting method has always been started with the periodic orbit at

$Ra = 2320$ (shown in Fig. 4 as an empty circle) as initial condition, and with $1 \leq m \leq 10$ sections. Pseudo-arclength and parameter continuation methods have been employed with an arclength-step control based on the curvature of the curve of solutions, and on the number of Newton's iterations. They are stopped when eqs. (2-3) are satisfied with a tolerance $10^{-8}\sqrt{m}$ in euclidean norm, and the norm of the difference of the last two iterates is below $10^{-8}\sqrt{m}$ times the norm of the last iterate. The factors \sqrt{m} have been included to have, approximately, the same accuracy in the periodic orbit for each value of m . The control of the arclength-step is relative to the euclidean norm of the solution to try that the same number of points be computed during the continuation for every m . Two close initial conditions, on the curve of solutions, are used to start the continuation. Linear extrapolation is used as predictor for the first step, and quadratical for the rest. The linear systems are solved by the restarted version of GMRES (GMRES(M)), with tolerances which ensure the quadratical convergence of Newton's method.

Let us recall here that GMRES is a projection method which, from an initial approximation y_0 of the solution of a linear system $\mathcal{M}y = b$ (\mathcal{M} is the matrix (4) in our case), obtains a sequence y_m , eventually convergent to the solution, defined by the conditions $y_m \in y_0 + \mathcal{K}_m$, with $\mathcal{K}_m = \{r_0, \mathcal{M}r_0, \mathcal{M}^2r_0, \dots, \mathcal{M}^{m-1}r_0\}$ (the Krylov subspace of dimension m) and $r_0 = b - \mathcal{M}y_0$, and y_m minimizing $\|b - \mathcal{M}y\|_2$ over $y \in y_0 + \mathcal{K}_m$. Then it is easy to see that $r_m = b - \mathcal{M}y_m = P_m(\mathcal{M})r_0$, P_m being a polynomial of degree m with $P_m(0) = 1$. In the restarted version of the method (GMRES(M)), in order to prevent the dimension of \mathcal{K}_m from growing too much, the iteration is stopped when the dimension reaches a size M , and restarted by using the last approximation y_M as the new initial seed.

When a system with matrix (4) is solved without preconditioning, we take the maximal dimension of the Krylov subspace for GMRES(M) as $M = mK_G$, K_G being the dimension used for $m = 1$, which in all the computations shown has been $K_G = 10$. This is done to ensure that GMRES(M) sees the same fraction of the eigenvalues of each matrix. With this value, GMRES(M) is almost never restarted, so it works as if we were using its full version.

In the preconditioned case there is no need to take $M = mK_G$ because, for the range of m considered, the maximum number of iterations required is reduced in such a way that M can be taken independent of m . For instance, if the dimension of the base invariant subspace is $k = 20$, the maximum number of iterations has always been below 25, and, in order to have no restarting, $M = 25$ is enough for all m . Therefore, the memory requirement of GMRES is not a problem for the application of the method. Moreover, the computational cost of the linear solver has always been completely negligible compared with the time integrations.

All the computations were performed using a small cluster of six Core 2 Quad Intel processors Q9450 with 12Mb of cache memory, connected to a full-duplex Gigabit ethernet local network. In order to minimize cache conflicts, at most two time integrations were

computed on each processor at the same time. The parallelism was implemented by using MPI, and only to compute $\mathcal{P}(X, \lambda)$ and $D\mathcal{P}(X, \lambda)(V, \mu)$. These time integrations, represent more than 99% of the total CPU time. The relative differences, between the CPU times to compute each partial shoot, remained below 4% during all the continuations performed, so that they are synchronized enough to allow good speedups.

We have check that, in all our calculations, the time spent in the communications is negligible. This time must be compared with the time integration to obtain $P_i(x_i, \lambda)$ or $DP_i(x_i, \lambda)(v_i, \mu)$, which depends on the problem, but is orders of magnitude larger. In the example shown, the computation of each action $DP_i(x_i, \lambda)(v_i, \mu)$ takes of the order of several tens of seconds, depending on the number of intermediate shoots, m . The only communications required, when using MPI, are sending the current value of the parameter λ , sending x_i and v_i , and receiving $P_i(x_i, \lambda)$ or $DP_i(x_i, \lambda)(v_i, \mu)$ and T_i .

To study the efficiency of the multiple shooting method, we define $T(m, k)$ as the wall-clock time required to do a continuation with m sections, with each partial shoot computed on a different processor, and using a preconditioner based on an invariant subspace of dimension mk . By definition, $k = 0$ means no preconditioning. Notice that due to the spreading of the eigenvalues, which can be seen in Fig. 6, as m increases k must also increase. This prevents the use of high values of m because the cost of the computation of the stability and the preconditioner must be limited.

The (absolute) speedup, $S(m, k)$, is defined as

$$S(m, k) = T_{ref}/T(m, k), \quad (16)$$

T_{ref} being the wall-clock time of the best algorithm without using parallelism. In this definition we compare the response time of doing a continuation for m sections using parallelism, and preconditioning by using a subspace of dimension mk , with a reference time corresponding to the best serial algorithm.

The efficiency, $E(m, k)$, is

$$E(m, k) = S(m, k)/m. \quad (17)$$

It can be greater than one if the serial and the parallel algorithms are different. If $E(m, k) = \gamma$, the initial time T_{ref} is reduced by a factor γm . The optimal case is $\gamma = 1$, but suboptimal cases ($\gamma < 1$) are also of interest if γ is close to one.

Several issues must be taken into account to decide which is the best serial algorithm. Let us define $T'(m, k)$ as the wall-clock time required to do a continuation with m sections, using a preconditioner based on an invariant subspace of dimension mk , but now doing all the computations on a single processor. If one wants to compare the same algorithm using one or m processors, the (relative) speedup should be defined as $T'(m, k)/T(m, k)$. The only added time in the parallel case is due to the communications, which are negligible, and to the possible lack of synchronization of the different shoots, which can also be neglected

by changing the sections frequently, as stated before. Therefore, $T'(m, k) \approx mT(m, k)$, and the efficiency would always be close to one, no matter the values of m and k , giving the false impression that a linear speedup can be obtained regardless of how large m is. This only indicates that this serial algorithm is a good candidate to parallelization. Notice that the time $T'(m, k)$ increases with m at least for two reasons. The first is that the number of expensive initializations of the time integrator is m , one for each partial shoot. The second is that as m increases, the number of iterations to solve the linear systems also increases proportionally to m (see below for an explanation). Therefore, for a given k , one should use the lower possible value of m , allowed by the stability of the solution, to compute the periodic orbit using a single processor. In our example we can still compute the periodic orbits with $m = 1$. On the other hand, we have found that if $m = 1$ the preconditioner is only effective if the value of the parameter at which the periodic orbit is being computed is close to that at which the preconditioner was constructed. Therefore, we have taken (16) as the definition of the speedup with $T_{ref} = T(1, 0)$.

We assume that the computation of the stability and the preconditioner is done on a different processor, which would be also busy in the case $m = 1$, computing the stability of the orbits. Consequently, the time needed to prepare the preconditioner is not included in the definition of the speedup. In the unlikely case that the stability was of no interest, the efficiency should be multiplied by a factor $m/m + 1$ because there is another processor being used.

As a first test, five periodic orbits were computed by both the pseudo-arclength, and the parameter continuation methods without preconditioning. Figure 7 shows the speedup, and the efficiency as a function of m . The reference line, in red, is that corresponding to the linear speedup, $S(m, 0) = m$ and $E(m, 0) = 1$, which will be plotted for comparison in all the figures. The line in black corresponds to a process such that the wall-clock time is independent of the number of processors. This is nearly what happens in the computations shown. From these results it is clear that the only use of parallel multiple shooting does not accelerate significantly the computation of periodic orbits, if the same number of points is computed for each m . The wall-clock time does not decrease as $T(m, 0) \approx T(1, 0)/m$ as should be for a linear speedup $S(m, 0) = m$. The speedup is below 1.7 even for $m = 10$. This is due to the already mentioned increase of the number of iterations of the linear solver with m . Therefore, by using parallelism, we are dividing by, approximately, m the response time of the time integrations, but also increasing the number of integrations required, in such a way that the total response time is essentially the same.

It is easy to understand the linear dependence of the number of iterations with m , if the following well known result on the convergence of GMRES (see [Saad & Schultz, 1986]) for a system $\mathcal{M}y = b$ is used.

Proposition 3 *Suppose that a matrix \mathcal{M} diagonalizes with $\mathcal{M} = V\Lambda V^{-1}$, where $\Lambda = \text{diag}\{\lambda_1, \dots, \lambda_n\}$ is the diagonal matrix of eigenvalues, P_q is the set of polynomials of*

degree at most q , and $\kappa_2(V) = \|V^{-1}\|_2 \|V\|_2$ is the norm-2 condition number of V . Then, if y_q is the approximation at the q -th step of GMRES, with initial condition y_0 ,

$$\frac{\|b - \mathcal{M}y_q\|_2}{\|b - \mathcal{M}y_0\|_2} \leq \kappa_2(V) \inf_{\substack{p \in P_q \\ p(0)=1}} \sup_{i=1, \dots, n} |p(\lambda_i)|. \quad (18)$$

In our case \mathcal{M} is the matrix $(\mathcal{I} - \mathcal{A})$ bordered as in (4). As the clustering of the eigenvalues of $(\mathcal{I} - \mathcal{A})$ is not destroyed by the bordering, we can reason as if the matrix were $(\mathcal{I} - \mathcal{A})$. If for $m = 1$ a polynomial of degree q takes a certain maximum value on the spectrum of $I - A$, the degree required to attain the same maximum on the spectrum of $(\mathcal{I} - \mathcal{A})$ is mq because of the spreading of its eigenvalues (see Proposition 1). The leading eigenvalues of $I - A$, and $(\mathcal{I} - \mathcal{A})$ for $m = 5$ and $m = 10$ are shown in Fig. 6 for the periodic orbit at $Ra = 2305$ (see also table 1). Therefore, the number of iterations required to solve a linear system with GMRES would increase proportionally to m if the condition number $\kappa_2(V)$ were independent of m . Then $T(m, 0) \approx T(1, 0)$, and $S(m, 0) \approx 1$. However, Fig. 7 shows that some speedup is obtained, although small. There are two main reasons for this. The first is the better conditioning of the matrices of the multiple shooting, and of the matrices of their eigenvectors. The second is the better prediction of the new points along the curve of solutions. The periodic orbits we are computing are quite unstable, and the predicted point gives, at the first Newton's iteration, lower residuals as m increases.

6 Results for the Preconditioned System

In order to conform to the scheme of Fig. 3a, the length of the interval of Ra , that can be computed during the time the stability and the preconditioner are calculated, was estimated. The CPU time required to obtain the leading spectra (see table 1), and the invariant subspaces of dimensions $k = 6, 12, 16$, and 20 of a periodic orbit at $Ra = 2305$ was found. Afterwards, the preconditioner was calculated for all these values of k , and m from 2 to 10. Tables 2 and 3 show the CPU times needed. The worst case corresponds to $m = 9$ or 10, and $k = 20$. A total of 1665 seconds were needed to compute the stability, and to prepare the preconditioner. Then, starting at $Ra = 2305$, it was checked that the preconditioners were effective in all the continuations lasting at least 3330 seconds. This means that the number of iterations to solve the linear systems does not grow during the continuations more than 30%. During this time, an interval of more or less 30 units of Ra could be computed. Therefore, following the scheme of Fig. 3a, the computation of the periodic orbits can proceed without degrading during the time the preconditioner is prepared for updating.

The preconditioner used in all the tests presented is that defined by the right-hand-side matrix in (15), computed from the stability at $Ra = 2305$. The hyperplanes used for the

computation are normal to the initial periodic orbit at $Ra = 2320$, and placed at equally spaced times. In addition, and for the reasons explained before, the reference time, T_{ref} in Eq. (16) has always been that corresponding to $m = 1$ and no preconditioning ($k = 0$).

We have not included the preconditioned $m = 1$ case in the figures, because the number of iterations needed to solve the linear systems without preconditioning is close to 10, and then it is very difficult to reduce it. Moreover, the first leading eigenvalues shown in table 1 for $Ra = 2305$ depend strongly on Ra . This makes that, in the single shooting case, the preconditioner be effective only close to the point at which the stability was computed. Instead of accelerating the convergence it slows it down at the end of the interval considered.

To isolate the effect of the preconditioner from that of the multiple shooting method, the computation of the five points of Fig. 7 was repeated using the preconditioner. The results are presented in Figs. 8, and 9. They show the speedups, $S(m, k)$, and the efficiencies, $E(m, k)$, achieved, the graph of the linear speedup, $S(m) = m$ or $E(m) = 1$, and those computed by preconditioning with $k = 0, 6, 12, 16$, and 20. The curves of $k = 0$ were already represented in Fig. 7. It can be seen that close to linear speedups can be obtained for low m and large k , if only the effect of the preconditioner is taken into account to accelerate the computation of the periodic orbits.

The speedup is better for the parameter than for the pseudo-arclength continuation, since we use in both cases, the right-hand-side matrix in (15) as preconditioner, which is best suited for the second case. The bordering of the matrix in the pseudo-arclength continuation introduces an additional spreading of the eigenvalues, which we do not capture with this preconditioner. The left-hand-side matrix in (15), together with the BEM algorithm, already mentioned, could be used in this case. We have not implemented this option, and therefore we will restrict ourselves to parameter continuation from now on.

Figs. 10 and 11 correspond to the interval $Ra = 2320 < Ra < 2335$. So, all the computations done in this interval take more than 1665 seconds. There are some general characteristics which are common to the following figures. For a fixed m , both $S(m, k)$, and $E(m, k)$ are, in general, increasing functions of k , as could be expected. The larger the subspace used to preconditioning, the lower the number of iterations to solve the linear systems. For a fixed k , $S(m, k)$ increases with m in the range considered, but $E(m, k)$ has a maximum and then decreases. This indicates that, as m grows, the expansion of the spectra shown in Fig. 6 is noticed by GMRES, which *feels* the circles of eigenvalues. Then, an increase in k is required to have the same efficiency. A decrease of $S(m, k)$ with m can be expected if m is increased further.

The degrading of the preconditioner, explained before for the case $m = 1$, can be perceived in Figs. 8 to 11 for $m = 2$, which is the next m for which this effect is present. When $m = 2$, only if $k = 20$ there is a significant increase in $E(m, k)$ with respect to the unpreconditioned case. In Fig. 8, when $m = 2$, and $k = 6$ there is, in fact, a decrease in $E(m, k)$. This is what happens for $m = 1$ in all the continuations performed for the selected

interval of the parameter Ra . Therefore, it is better not to precondition the case $m = 1$. Except a first few, the multipliers are strongly clustered at zero. Preconditioning the simple shooting case would be interesting if there were more dispersion of the multipliers, even for stable solutions. As stated before, if $m > 1$, the speedup grows almost monotonically with k . In this case, the dependence of the leading eigenvalues of $\mathcal{I} - \mathcal{A}$ and of the corresponding invariant subspaces on the parameter becomes less pronounced. The greater m the weaker the dependence. Therefore the size of the interval for which the preconditioner is still valid grows with m .

In Fig. 10 the maximal continuation step size is limited to force that the number of points computed be similar for all the m considered. Figure 11 corresponds to the same computation, but without limitation on the continuation step size. The periodic orbits being computed are quite unstable, and the step taken by the single shooting code must be small to have a good starting condition for Newton's method. This is not the case for the multiple shooting, which can cover the considered interval by computing less points. By adding this effect to the improvement due to the preconditioner, it has been possible to obtain linear or better speedups for low values of m , without the need of a large value of k . Efficiencies above 0.8 have been obtained by using $k = 12$. To study the stability of the periodic orbits in the range of Ra considered, it is enough to know, for instance, the first six multipliers, but, from table 2, it is clear that increasing to twelve the number of multipliers has only a 30% extra cost. The increase in performance, due only to the multiple shooting algorithm, can be seen in the $k = 0$ curves of all figures. In Fig. 9 (green curve) the speedup arrives, approximately, to 1.6 for $m = 10$ (see also Fig. 7), but it is close to 2.6 in Fig. 11.

In the last plots in Fig. 12 the same calculations as in Fig. 11 are presented, but $T(m, k)$ has been redefined as the average time to compute a single continuation point. This figure is very similar to Fig. 8, and shows, again, the speedup and the efficiency due only to the preconditioner.

In all figures $E(m, k)$ decreases when $m \geq 4$ and k is fixed. Therefore, as m increases the dimension of the base invariant subspace k must be higher to keep $E(m, k)$ at the same value. For low values of m , a small k is enough to achieve close to optimal efficiencies. This result is important because, in many applications it is interesting to compute stable, or not very unstable periodic orbits, which do not require a high m . If we move along a line $E(m, k) = \gamma$, the storage required for the linear solver, increases only proportionally to m (it is mainly $K_G mn$ with K_G fixed) because the number of iterations of the linear solver is almost independent of m . These storage requirements are not a problem to use the method. As k must be increased with m to have $S(m, k) = m$, and the dimension of the invariant subspace is mk , the storage requirement for the preconditioner grows with m^2 (it is $O(m^2 n)$). This might limit the application of the method to relatively low values of m , if linear speedups were needed, the dimension of the system n were very large, and the action of the preconditioner were to be computed on one processor. If $n = 10^6$ and $m = 10$

the required storage is less than 1Gb, which can be stored in the memory of a standard current PC. If, for stability reasons, larger values of m were required, and the storage were limited, then k should be small, and only sub-linear speedups would be available. Another option would be to distribute the matrices Q_i among the different processors, which would only be done each time the preconditioner is updated, and to compute the action by M^{-1} in (14) in parallel.

7 Conclusions

We have shown that the computation of periodic orbits by parallel shooting can be accelerated by the use of a preconditioner based on the information on their stability. This information must be already obtained if a study of possible bifurcations is needed. The efficiency of the preconditioner will depend on the distance from the point being computed to that at which the invariant subspace was calculated. In the case of simple shooting, the preconditioner presented provides a way of accelerating the convergence for problems with small dissipation, or with mixed dissipative and small-dimension non-dissipative parts.

There are two issues which make it difficult to achieve linear speedups for large values of m . The first one is the time integration initialization process. In the example presented we have used DLSODPK to start the time integrations, which is computationally expensive because it requires solving a nonlinear system of equations at each step. A better option would be using a variable-step variable-order semi-implicit method, based on BDF-extrapolation formulas [Garcia *et al.*, 2010], which could be used not only in the starting process, but also during all the integration. In this case only a linear system must be solved at each step. This would have reduced the starting time; anyway, this is an intrinsic problem of any algorithm based on the multiple shooting method. The second is related to the expansion of the eigenvalues as m increases. To keep the efficiency close to the optimal, k must be increased with m to return the external circles of eigenvalues of $\mathcal{I} - \mathcal{A}$ back to a small neighborhood of $+1$. The computation of the corresponding invariant subspace, and preconditioner will become too expensive for large m . Despite these problems, we have seen that the computation time can be reduced easily by a factor of ten.

The dimension k , required to have close to linear speedups, depends on how the moduli of the multipliers of the orbits decay to zero near 1. For parabolic equations this decay is, in general, faster for three-dimensional than for lower dimensional problems. Therefore, best performances could be expected in the cases for which the acceleration of the process is more necessary.

The method proposed is particularly useful for the computation of unstable periodic orbits after one or several period-doubling bifurcations, because at each of them the multipliers are squared.

Other invariant objects could be computed in a similar way. For example, fixed points of $f(x)$, which are difficult to obtain by preconditioned Newton-Krylov methods due to the poor convergence of the linear solvers, could be computed as fixed points of the map $x \rightarrow \psi(T, x)$, if $\psi(T, x)$ is the solution of $\dot{x} = f(x)$ with initial condition x at time T . Now T is a time adequately chosen. If the fixed point x_0 is stable, the method is an acceleration of the time evolution to x_0 . If x_0 is unstable, T must be short enough to avoid escaping too far away from x_0 , but long enough to let the contractive properties on almost all the directions of the flow $\psi(t, x)$ act. This method, based on Newton-Krylov iterations, was suggested, for instance, in [Sánchez *et al.*, 2004], although the original idea of using stabilized time evolutions to find unstable fixed points goes back to [Jarausch & Mackens, 1987; Shroff & Keller, 1993]. Similar multiple shooting techniques could also be employed in the computation of invariant tori, if the unknown of the problem is a single point of the invariant curve on a certain generalized Poincaré section [Sánchez *et al.*, 2010]. In fact the method applies to any fixed point problem for dissipative maps obtained by time integration.

8 Acknowledgments

This research has been supported by MEC-DGI project FIS2007-64993, and by AGAUR-GENCAT project 2009-SGR-67.

References

- Allgower, E. L. & Georg, K. [1990]. *Numerical Continuation Methods: An Introduction*, volume 13 of *Computational Mathematics*. Springer.
- Ascher, U. M., Mattheij, R. M. M., & Russell, R. D. [1995]. *Numerical solution of boundary value problems for ordinary differential equations*. Classics in Applied Mathematics. SIAM, Philadelphia, PA.
- Batiste, O., Knobloch, E., Mercader, I., & Net, M. [2002]. “Simulations of Oscillatory Binary Fluid Convection in Large Aspect Ratio Containers,”. *Phys. Rev. E*, **65**, 0163031–01630319.
- Benner, P., Mehrmann, V., Sima, V., Van Huffel, S., & Varga, A. [1999]. “SLICOT – A Subroutine Library in Systems and Control Theory,”. In Biswa Nath Datta, editor, *Applied and Computational Control, Signals, and Circuits*, volume 1(10), pages 499–539. Birkhäuser.
- Böhmer, K., Mei, Z., Schwarzer, A., & R., S. [2000]. “Path-following of large bifurcation problems with iterative methods,”. In E. Doedel & L. S. Tuckerman, editors, *Numerical*

- methods for bifurcation problems and large-scale dynamical systems*, volume 119 of *The IMA volumes in Mathematics and its applications*, pages 35–65. Springer.
- Bojanczyk, A., Golub, G., & van Dooren, P. [1992]. “The periodic Schur decomposition: algorithms and applications.”. *Proceedings of the SPIE — The International Society for Optical Engineering*, **1770**, 31–42.
- Canuto, C., Hussaini, M. Y., Quarteroni, A., & Zang, T. A. [2007]. *Spectral Methods. Evolution to complex geometries and applications to Fluid Dynamics*. Springer.
- Carpentieri, B., Giraud, L., & Gratton, S. [2007]. “Additive and multiplicative two-level spectral preconditioning for general linear systems.”. *SIAM J. Sci. Comput.*, **29**(4), 1593–1612.
- Chapman, B., Jost, G., & van der Pas, R. [2007]. *Using OpenMP. Portable Shared Memory Parallel Programming*. Scientific and Engineering Computation. MIT Press.
- Dembo, R., Eisenstat, S., & Steihaug, T. [1982]. “Inexact Newton methods.”. *SIAM J. Numer. Anal.*, **19**(2), 400–408.
- Dhooge, A., Govaerts, W. J. F., & Kuznetsov, Y. [2003]. “MatCont: A MATLAB package for numerical bifurcation analysis of ODEs.”. *ACM Trans. Math. Soft.*, **29**, 141–164.
- Doedel, E. [1986]. *AUTO: Software for continuation and bifurcation problems in ordinary differential equations*. Report Applied Mathematics, California Institute of Technology, Pasadena, USA.
- Doedel, E. [2007]. “Lecture Notes on Numerical Analysis of Nonlinear Equations.”. Concordia University, Canada.
- Edwards, W. S., Tuckerman, L. S., Friesner, R. A., & Sorensen, D. C. [1994]. “Krylov methods for the incompressible Navier-Stokes equations.”. *J. Comput. Phys.*, **110**, 82–102.
- Erhel, J., Burrage, K., & Pohl, B. [1996]. “Restarted GMRES preconditioned by deflation.”. *J. Comp. Appl. Math.*, **69**, 303–318.
- Garcia, F., Net, M., García-Archilla, B., & Sánchez, J. [2010]. “A comparison of high-order time integrators for the Boussinesq Navier-Stokes equations in rotating spherical shells.”. *Preprint*.
- Govaerts, W. J. F. [2000]. *Numerical Methods for Bifurcations of Dynamical Equilibria*. SIAM.

- Guckenheimer, J., Myers, M. R., Wicklin, F. J., & Worfolk, P. A. [1995]. *DsTool: A dynamical system toolkit with an interactive graphical interface*. Center for Applied Mathematics, Cornell University, Ithaca NY, USA.
- Henderson, M. E. [2002]. “Multiple Parameter Continuation: Computing Implicitly Defined k-manifolds,”. *Int. J. Bifurcation and Chaos*, **12**(3), 451–476.
- Hindmarsh, A. C. [1983]. “ODEPACK, A Systematized Collection of ODE Solvers,”. In R. S. S. et al., editor, *Scientific Computing*, pages 55–364. North-Holland, Amsterdam.
- Jarausch, H. & Mackens, W. [1987]. “Solving large nonlinear systems of equations by an adaptive condensation process,”. *Numer. Math.*, **50**(6), 633–653.
- Keller, H. B. [1987]. *Lectures on Numerical Methods in Bifurcation Theory*. Tata Institute of Fundamental Research, Lectures on Mathematics and Physics. Springer-Verlag, New York.
- Kressner, D. [2005]. *Numerical methods for general and structured eigenvalue problems*, volume 46 of *Lecture Notes in Computational Science and Engineering*. Springer.
- Kressner, D. [2006]. “A periodic Krylov-Schur algorithm for large matrix products,”. *Numer. Math.*, **103**(3), 461–483.
- Kuznetsov, Y. A. [1998]. *Elements of Applied Bifurcation Theory, Second Edition*. Springer.
- Kuznetsov, Y. A. & Levitin, V. V. [1996]. “CONTENT, a multiplatform continuation environment,”. Technical report, CWI, Amsterdam, The Netherlands.
- Lehoucq, R. B., Sorensen, D. C., & Yang, C. [1998]. *ARPACK User’s Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. SIAM.
- Lopez, J. M., Marqués, F., & Sánchez, J. [2001]. “Oscillatory modes in an enclosed swirling flow,”. *J. Fluid Mech.*, **439**, 109–129.
- Lust, K. [1997]. *Numerical Bifurcation Analysis of Periodic Solutions of Partial Differential Equations*. Ph.D. thesis, Department of Computer Sciences, KU Leuven, Belgium.
- Lust, K. [2000]. psSchur: A code for computing the periodic Schur decomposition, The code is available on the web at https://perswww.kuleuven.be/~u0006235/ACADEMIC/r_psSchur.html.
- Lust, K. [2001]. “Improved numerical Floquet multipliers,”. *Internat. J. Bifur. Chaos Appl. Sci. Engrg.*, **11**(9), 2389–2410.

- Lust, K., Roose, D., Spence, A., & Champneys, A. [1998]. “An adaptive Newton-Picard algorithm with subspace iteration for computing periodic solutions,”. *SIAM J. Sci. Comput.*, **19**(4), 1188–1209.
- Message Passing Forum [1994]. “MPI: A Message-Passing Interface Standard,”. Technical report, University of Tennessee, Knoxville, TN, USA.
- Molemaker, M. J. & Dijkstra, H. A. [2000]. “Multiple equilibria and stability of the North-Atlantic wind-driven ocean circulation,”. In E. Doedel & L. S. Tuckerman, editors, *Numerical methods for bifurcation problems and large-scale dynamical systems*, volume 119 of *The IMA volumes in Mathematics and its applications*, pages 35–65. Springer.
- Parks, M., De Sturles, E., Mackey, G., Johnson, D., & Maiti, S. [2006]. “Recycling Krylov subspaces for sequences of linear systems,”. *SIAM J. Sci. Comput.*, **28**, 1651–1674.
- Rheinboldt, W. C. [1986]. *Numerical Analysis of Parametrized Nonlinear Equations*. J. Wiley.
- Rheinboldt, W. C. & Burkardt, J. V. [1983]. “Algorithm 596: a program for a locally parameterized continuation process,”. *ACM Trans. Math. Softw.*, **9**(2), 236–241.
- Saad, Y. [1992]. *Numerical Methods for Large Eigenvalue Problems*. Manchester University Press, Manchester.
- Saad, Y. [1994]. “Preconditioned Krylov subspace methods for CFD applications,”. Technical Report umsi-94-171, Minnesota Supercomputer Institute, Minneapolis, MN 55415.
- Saad, Y. [1996]. *Iterative methods for sparse linear systems*. PWS pub. company, New York.
- Saad, Y. & Schultz, M. H. [1986]. “GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems,”. *SIAM J. Sci. Stat. Comput.*, **7**, 865–869.
- Salinger, A. G., Bou-Rabee, N. M., Pawlowsky, R. P., Wilkes, E. D., Burroughs, E. A., Lehoucq, R. B., & Romero, L. A. [2002]. *LOCA 1.1. Library of Continuation Algorithms: Theory and Implementation Manual*. Sandia National Laboratories, Albuquerque, USA.
- Sánchez, J., Marqués, F., & López, J. M. [2002]. “A continuation and bifurcation technique for Navier-Stokes flows,”. *J. Comput. Phys.*, **180**, 78–98.
- Sánchez, J., Net, M., García-Archilla, B., & Simó, C. [2004]. “Newton-Krylov continuation of periodic orbits for Navier-Stokes flows,”. *J. Comput. Phys.*, **201**(1), 13–33.

- Sánchez, J., Net, M., & Vega, J. [2006]. “Amplitude equations close to a triple-(+1) bifurcation point of D_4 -symmetric periodic orbits in $O(2)$ -equivariant systems,”. *Discrete and Continuous Dynamical Systems-Series B*, **6**(6), 1357–1380.
- Sánchez, J., Net, M., & Simó, C. [2010]. “Computation of invariant tori by Newton-Krylov methods in large-scale dissipative systems,”. *Physica D*, **239**, 123–133.
- Seydel, R. [1994]. *Practical bifurcation and stability analysis. From equilibrium to chaos*. Springer, New York.
- Shroff, G. M. & Keller, H. B. [1993]. “Stabilization of unstable procedures: the recursive projection method,”. *SIAM J. Numer. Anal.*, **30**(4), 1099–1120.
- Simó, C. [1990]. “Analytical and numerical computation of invariant manifolds,”. In C. Benest & C. Froeschlé, editors, *Modern methods in celestial mechanics*, pages 285–330. Editions Frontières.
Also at <http://www.maia.ub.es/dsg/2004/>.
- Sorensen, D. C. [1992]. “Implicit application of polynomial filters in a k -step Arnoldi method,”. *SIAM J. Matrix Anal. Appl.*, **13**(1), 357–385.
- Stewart, G. W. [2002]. “A Krylov–Schur Algorithm for Large Eigenproblems,”. *SIAM J. Matrix Anal. Appl.*, **23**(3), 601–614.
- van Noorden, T. L., Verduyn Lunel, S. M., & Blik, A. [2004]. “A Broyden rank $p+1$ update continuation method with subspace iteration,”. *SIAM J. Sci. Comput.*
- Varga, A. & Dooren, P. [2001]. “Computational methods for periodic systems - an overview,”. In *Proc. of IFAC Workshop on Periodic Control Systems, Como, Italy*, pages 171–176.
- Watkins, D. S. [2005]. “Product eigenvalue problems,”. *SIAM Rev.*, **47**(1), 3–40.

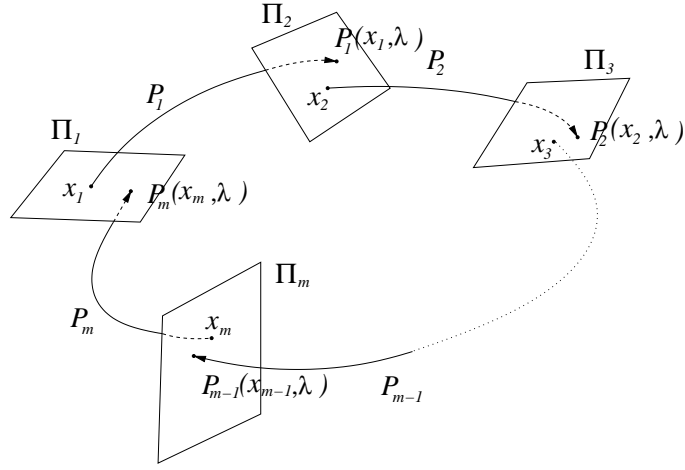


Figure 1: Scheme of the multiple shooting method.

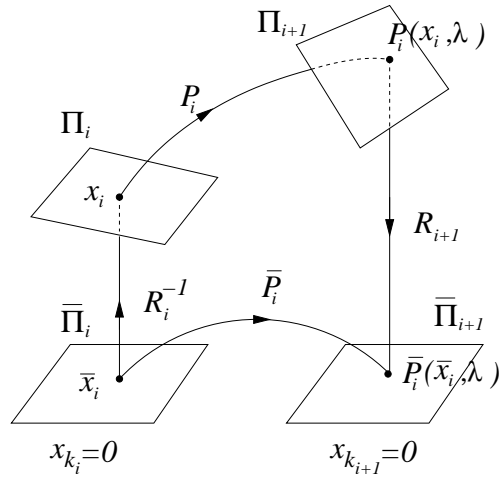


Figure 2: Parameterization of the partial Poincaré maps.

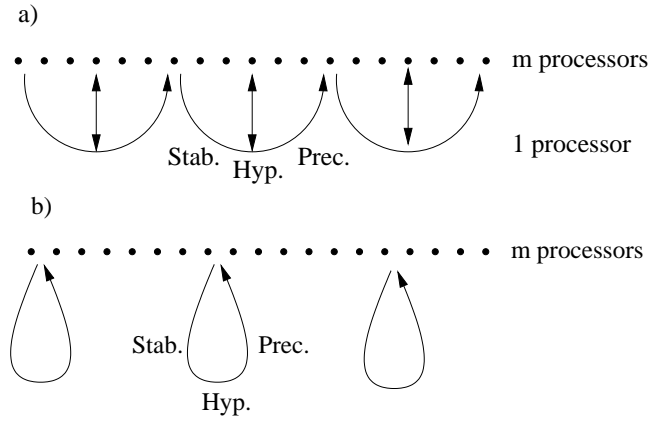


Figure 3: Two possible ways of linking the continuation and the updating of the preconditioner processes. a) m processors are used to compute the periodic orbits and another one is updating the preconditioner. b) the m processors are used to do both tasks.

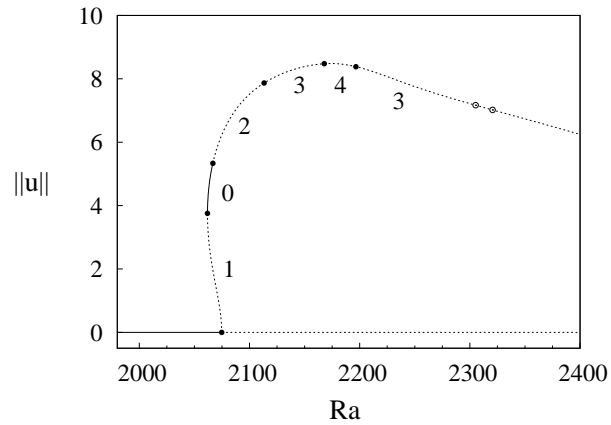


Figure 4: Bifurcation diagram. The full circles indicate bifurcation points, and the empty circles the point at which the continuation is started in the numerical experiments ($Ra = 2320$), and that at which the preconditioner has been computed ($Ra = 2305$). Solid and dashed lines mean stable and unstable branches, respectively. The labels beside the branch of periodic orbits indicate the number of multipliers outside the unit circle.

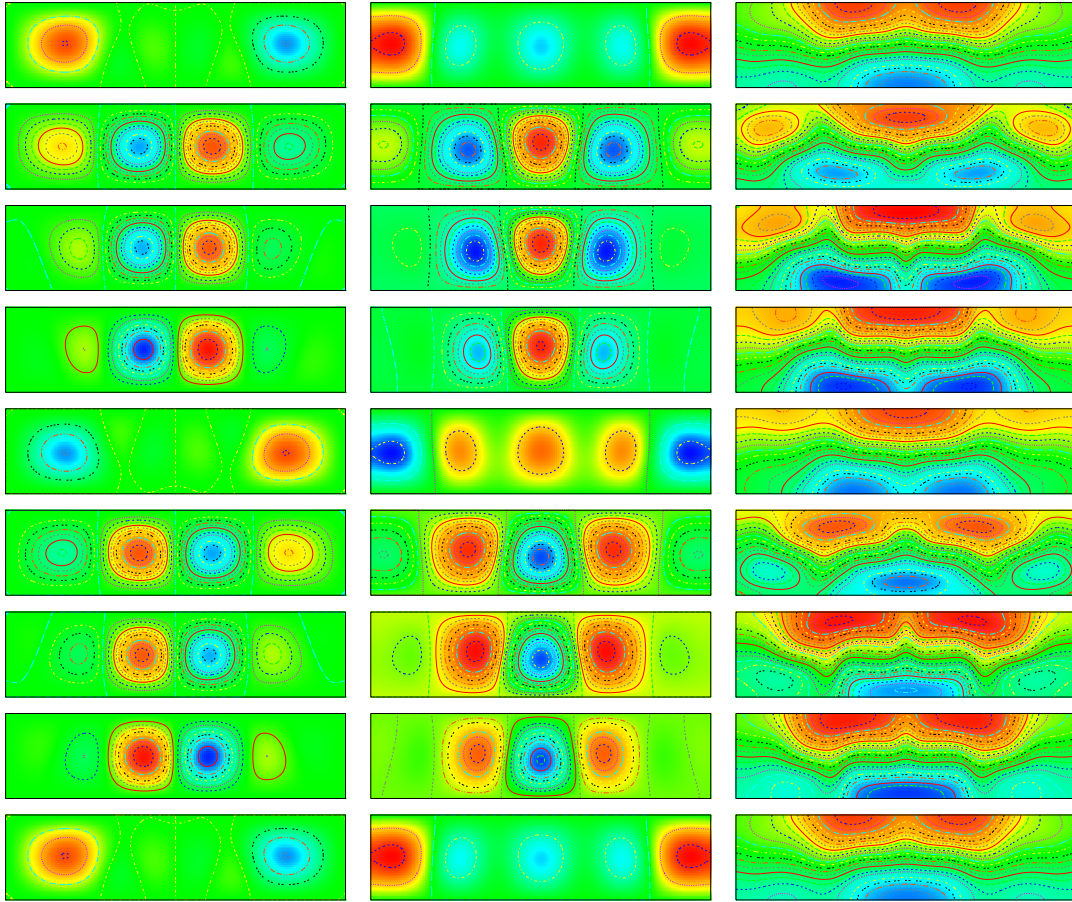


Figure 5: Temporal sequence during a period, T , for the periodic orbit at $Ra = 2306$, showing its spatio-temporal symmetries. Time increases from top to bottom in increments of $T/8$. The left, center, and right columns correspond to the streamfunction ψ , the perturbation of the temperature Θ , and that of the concentration C , respectively.

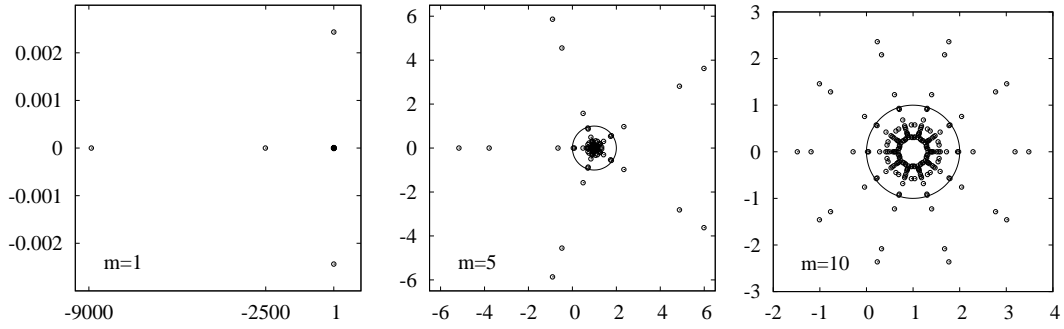


Figure 6: Leading eigenvalues of $\mathcal{I} - \mathcal{A}$ for $m = 1$, $m = 5$, and $m = 10$. The closed curve is the unit circle centered at $+1$.

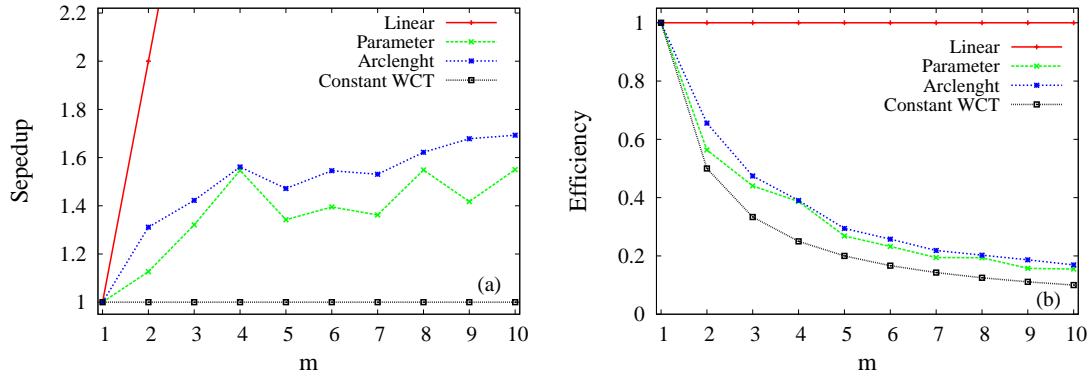


Figure 7: (a) Speedup, and (b) efficiency versus the number of sections m , corresponding to the computation of five points without preconditioning, starting at $Ra = 2320$.

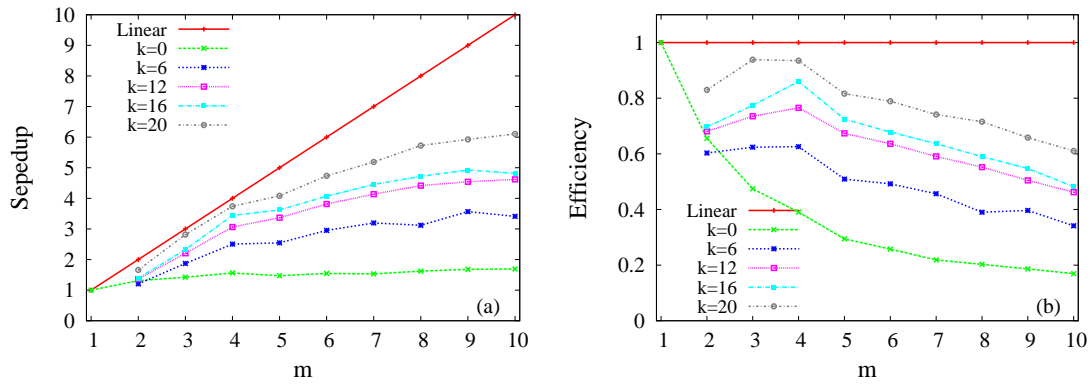


Figure 8: Pseudo-arclength continuation of 5 points. (a) Speedup, and (b) efficiency.

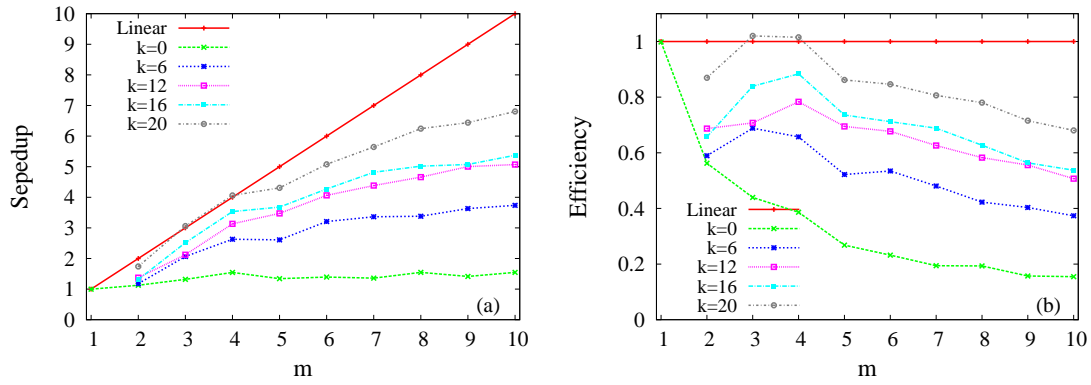


Figure 9: Parameter continuation of 5 points. (a) Speedup, and (b) efficiency.

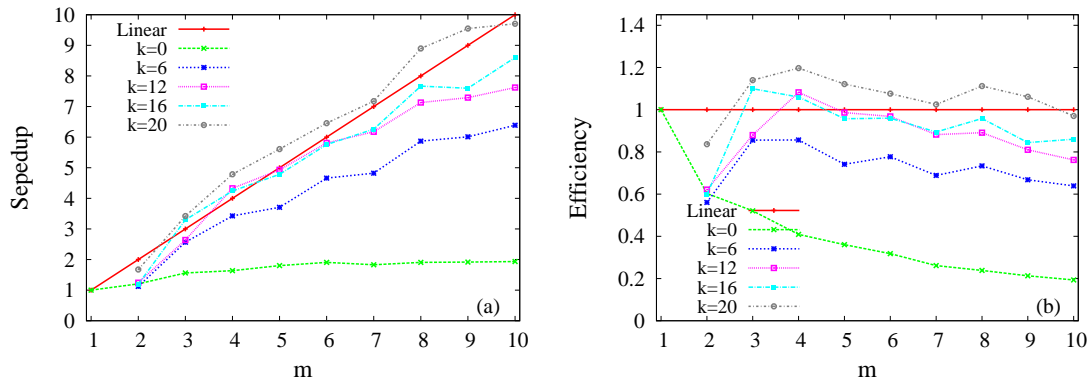


Figure 10: Parameter continuation of an interval of 15 units in Ra limiting the step size. (a) Speedup, and (b) efficiency.

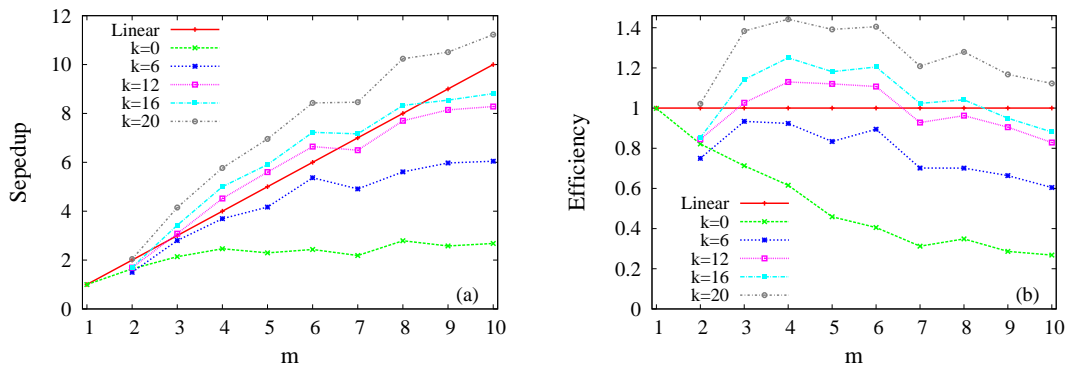


Figure 11: Parameter continuation of an interval of 15 units in Ra without limitation of the step size. (a) Speedup, and (b) efficiency.

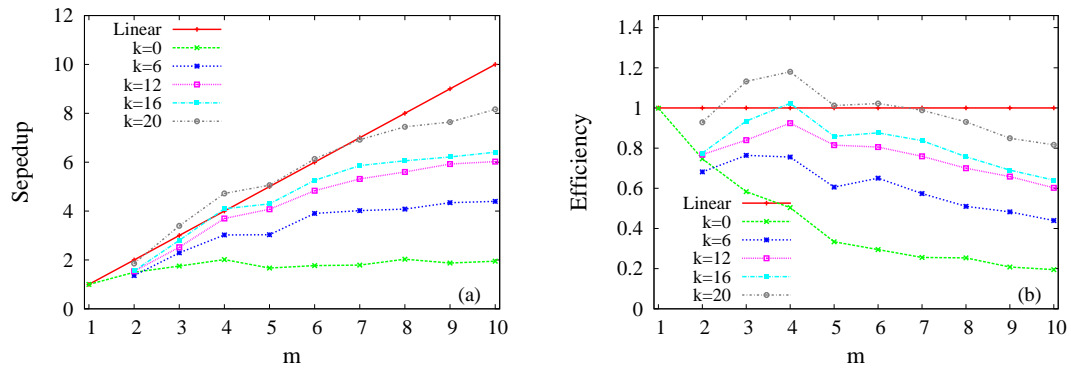


Figure 12: Parameter continuation of an interval of 15 units in Ra without limitation of the step size. (a) Speedup, and (b) efficiency considering the average time to compute a new point.

Real Part	Imag. Part	Modulus	Real Part	Imag. Part	Modulus
0.8912E+4	0.0000E+0	0.8912E+4	0.1161E-2	0.0000E+0	0.1161E-2
0.2513E+4	0.0000E+0	0.2513E+4	0.2353E-3	0.0000E+0	0.2353E-3
0.1250E+2	0.0000E+0	0.1250E+2	0.1029E-3	0.5231E-5	0.1030E-3
0.8042E+0	0.0000E+0	0.8042E+0	0.1029E-3	-0.5231E-5	0.1030E-3
0.6312E+0	0.0000E+0	0.6312E+0	0.5525E-4	0.0000E+0	0.5525E-4
0.3638E-1	0.0000E+0	0.3638E-1	0.1190E-4	0.0000E+0	0.1190E-4
0.4632E-2	0.0000E+0	0.4632E-2	-0.5218E-5	0.1025E-4	0.1150E-4
-0.3338E-2	0.2436E-2	0.4132E-2	-0.5218E-5	-0.1025E-4	0.1150E-4
-0.3338E-2	-0.2436E-2	0.4132E-2	-0.5560E-5	0.4723E-5	0.7295E-5
0.2572E-2	0.0000E+0	0.2572E-2	-0.5560E-5	-0.4723E-5	0.7295E-5

Table 1: First twenty leading Floquet multipliers at $Ra = 2305$.

k	6	12	16	20
CPU time	449	586	895	858

Table 2: CPU time, in seconds, required to compute the stability at $Ra = 2305$.

m	$k = 6$	$k = 12$	$k = 16$	$k = 20$
2	207	417	557	695
3	212	428	573	717
4	219	443	592	742
5	219	442	591	742
6	226	456	613	769
7	230	466	625	788
8	229	464	622	783
9	235	478	642	807
10	235	477	642	805

Table 3: CPU time, in seconds, required to compute the preconditioner at $Ra = 2305$.