Matrix-free continuation of limit cycles for bifurcation analysis of large thermoacoustic systems

Iain Waugh[1,*], Simon Illingworth, Matthew Juniper

Cambridge University Engineering Department, Trumpington Street, Cambridge, CB2 1PZ, United Kingdom, Fax: +44 1223 332585.

Abstract

In order to define the nonlinear behaviour of a thermoacoustic system, it is important to find the regions of parameter space where limit cycles exist. Continuation methods find limit cycles numerically in the time domain, with no additional assumptions other than those used to form the governing equations. Once the limit cycles are found, these continuation methods track them as the operating condition of the system changes.

Most continuation methods are impractical for finding limit cycles in large thermoacoustic systems because the methods require too much computational time and memory. In the literature, there are therefore only a few applications of continuation methods to thermoacoustics, all with low-order models.

Matrix-free shooting methods efficiently calculate the limit cycles of dissipative systems and have been demonstrated recently in fluid dynamics, but are as yet unused in thermoacoustics. These matrix-free methods are shown to converge quickly to limit cycles by implicitly using a 'reduced order model' property. This is because the methods preferentially use the influential bulk motions of the system, whilst ignoring the features that are quickly dissipated in time.

The matrix-free methods are demonstrated on a model of a ducted 2D diffusion flame, and the stability limits are calculated as a function of the Peclet number and the heat release parameter. Both subcritical and supercritical Hopf bifurcations are found. Physical information about the flame-acoustic interaction is found from the limit cycles and Floquet modes. Invariant subspace preconditioning, higher order prediction techniques, and multiple shooting techniques are all shown to reduce the time required to generate bifurcation surfaces. Two types of shooting are compared, and two types of matrix-free evaluation are compared.

Keywords: matrix-free, continuation, limit cycles, thermoacoustic, diffusion flame

1. Introduction

Thermoacoustic oscillations can occur whenever combustion takes place inside an acoustic resonator. Unsteady combustion is an efficient acoustic source, and combustors have acoustic resonances at certain frequencies. Therefore for suitable phase differences between combustion and acoustic perturbations, large-amplitude self-excited limit cycles can occur.

The amplitude and phase between the acoustic perturbations and the heat release perturbations determine the thermoacoustic behaviour. This is often a strong function of the system parameters. Consequently, suitable relationships for limit cycles are only found at certain parameter-amplitude combinations. Because limit cycles are generally unwanted in a combustion system, we are interested in finding the safe operating region of parameter space, where no limit cycles exist (and where no other high-amplitude states, such as chaos, exist).

*Corresponding author (icw26@cam.ac.uk). Supported by EPSRC and IMechE funding.

Many methods have been used in the literature to find safe operating regions of thermoacoustic systems, both in the frequency domain and the time domain. These methods can be divided into those that use linear analysis and those that use nonlinear analysis.

The linear analyses describe the stability of infinitesimal perturbations around a steady state (fixed-point). The operating region of the combustor is then defined to be safe when the steady state of the combustor is linearly stable. In the frequency domain, a Flame Transfer Function (FTF) is often used for linear analysis, both in computational [1, 2] and experimental [3, 4, 5, 6] studies. The FTF is a transfer function between an acoustic perturbation and a heat release perturbation. It has both a frequency-dependent phase and a frequency-dependent gain. The FTF is then coupled with a linear acoustic model of the combustor, and the stability of the system is analysed using standard techniques from control theory. The system is linearly unstable at operating conditions for which any frequency has a positive growth rate in the coupled system.

In the time domain, the behaviour of small perturbations about the fixed point is described by a Jacobian matrix. The eigenvalues of this matrix are used to define the linear stability of the system [7, 8, 9]. The system is linearly unstable at operating conditions for which any eigenvalue, or complex pair of eigenvalues, has a positive real part (growth rate). This analysis calculates the same safe operating region as that calculated by the FTF approach.

Linear stability analyses can calculate the operating conditions at which the steady state is unstable, but cannot predict the resulting behaviour of the system. The system can evolve to a limit cycle, a period-2^n limit cycle, chaos or another fixed point [10, 11, 12]. Linear stability analysis is often used in thermoacoustics to estimate the frequencies of large-amplitude limit cycles. This is successful only when the heat release varies approximately linearly with the acoustic amplitude.

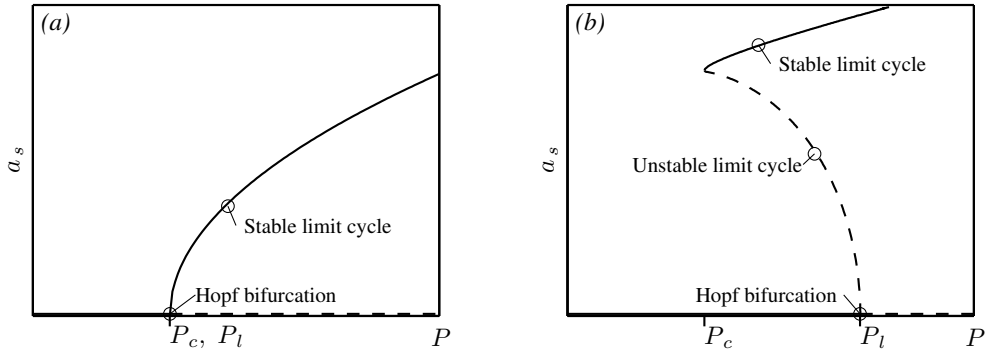


Figure 1: Schematic supercritical (a) and subcritical (b) Hopf bifurcations, in terms of the limit cycle amplitude, a_s and a system parameter P. The critical parameter, P_c, is the parameter value where limit cycles first occur. The linear stability limit, P_l, is the same as the Hopf bifurcation point. In the subcritical Hopf bifurcation, there is a bistable region where there is both a stable fixed point and a stable limit cycle ($P_c < P < P_l$). Triggering can occur in the bistable region.

Nonlinear analyses are required in order to calculate limit cycle amplitudes and mode shapes, and to quantify the extent of any bistable operating regions. In thermoacoustics, it is common to see a branch of limit cycles emerging from a Hopf bifurcation. A Hopf bifurcation is where a complex pair of eigenvalues of the Jacobian matrix (for the fixed point) have zero real part. Hopf bifurcations in thermoacoustics are either supercritical or subcritical, as shown schematically in Figure 1(a) and (b). Both types of bifurcation have been seen in experimental combustors [13, 10, 11]. When subcritical Hopf bifurcations are present in a system, triggering can occur from a linearly stable fixed point to a large amplitude limit cycle. This triggering can be caused by a pulse [14, 15], background noise, or combustion noise [16, 17].

In the frequency domain, the Flame Describing Function (FDF) is commonly used to estimate limit cycle amplitudes and to characterise the behaviour of experimental flames [18, 19]. The FDF is similar to the FTF, but its gain and phase depend on the acoustic amplitude, as well as on the acoustic frequency.

The FDF is nonlinear, but still has some remnants of a linear framework: it relies on a single-frequency in, single-frequency out relationship. In other words, the flame response is discarded at frequencies other than the forcing frequency. This assumption can lead to inaccuracy in the predicted limit cycle amplitudes if the flame responds strongly at frequencies other than the forcing frequency.

In the time domain, continuation methods have been developed in the field of nonlinear dynamics to track solutions to the governing equations, whilst varying system parameters. The solutions can be fixed points, limit cycles, or bifurcation points. The behaviour of a fluid system, which may seem complex, can often be understood in terms of attraction to and repulsion from these solutions [20]. Software packages such as AUTO [21], MATCONT [22] and DDE-BIFTOOL [23] have been developed to track solutions of generic time dependent systems. These software packages are powerful for systems with $\mathcal{O}(10)$ variables, but are impractical for larger systems because they use direct solvers for the underlying linear algebra. Continuation methods rely on the solution of a series of linear equations to find limit cycles, and the exact solution of these linear equations becomes prohibitively expensive for larger systems, both in terms of computational time and memory usage.

Matrix-free iterative methods can reduce both the time and memory required to solve these linear equations [24, 25]. Matrix-free methods have recently been used to find cusp bifurcations [26] and limit cycles in thermal convection [27], with $\mathcal{O}(10^3)$ variables. Continuation of steady states has been performed for thermal convection [28] using the matrix-free algorithms of the LOCA package [29], of which some subroutines are used in this paper. Limit cycles have also been extracted from turbulent Couette flow [30] with $\mathcal{O}(10^5)$ variables, and turbulent pipe flow [31] with $\mathcal{O}(10^4)$ variables, using matrix-free methods and hook-step optimisation routines. For larger systems, multiple-shooting techniques have been used to reduce the computational time to find limit cycles, and have been used for Navier-Stokes flows [32] and ocean modelling [33], with up to $\mathcal{O}(10^6)$ variables. The stability of the resulting solutions can also be found using matrix-free methods, such as the Arnoldi [34] or Krylov-Schur [35] algorithms.

Because combustion and fluids systems are dissipative, only a few bulk fluid motions are influential in the long time limit. This means that matrix-free methods are particularly well suited to finding limit cycles. The iterative methods inexactly solve the linear equations by implicitly using these influential bulk motions, whilst ignoring features that are quickly dissipated in time.

Continuation methods have been used in the field of thermoacoustics to calculate limit cycles and sub-/supercritical bifurcations [7, 15, 8, 9], but only for small systems with $\mathcal{O}(10)$ variables. To investigate coupled flame-acoustic interaction, however, it is necessary to model the flame shape, which even in reduced order models requires $\mathcal{O}(10^2 - 10^3)$ variables.

The aim of this paper is to present a method for finding limit cycles in large thermoacoustic systems, using an iterative matrix-free continuation technique. The technique is able to calculate the safe operating region of the thermoacoustic system in the time domain, and find the mode shape and frequencies of any limit cycles. The paper begins by introducing two shooting methods and by introducing the iterative techniques used to converge to the limit cycles. The paper then describes how the iterative process can be achieved with matrix-free techniques. Additional methods of increasing the efficiency of the continuation process, such as invariant subspace preconditioning, adaptive step sizing, and higher order prediction are then discussed. The numerical methods are then demonstrated on a model of a 2D diffusion flame in an acoustic duct in section 6. The shooting methods and matrix-free techniques are compared and discussed. The paper ends with a description of multiple shooting methods and a discussion of how they can decrease the time required to find a limit cycle.

2. Finding limit cycles

Continuation methods examine nonlinear systems whose evolution is governed by:

$$\frac{\mathrm{d}\underline{x}(t)}{\mathrm{d}t} = F(\underline{x}(t), \lambda), \quad \underline{x}(t) \in \mathbb{R}^N \tag{1}$$

where $\underline{x}$ is the current state of the system and λ are parameters. The governing equations are in most cases derived from the discretisation of a PDE.

Limit cycles satisfy:

$$\underline{x}(0) = \underline{x}(T)\,, \quad \left\{T \in \mathbb{R}^+ | T \neq 0\right\}, \tag{2}$$

where T is the period of the cycle. For nonlinear thermoacoustic systems with delays, however, this condition is less simple because the governing equation becomes $\dot{\underline{x}}(t) = F(\underline{x}(t), \underline{x}(t-\tau), \lambda)$ and a limit cycle must satisfy $\underline{x}(-\tau \leq t \leq 0) = \underline{x}(T-\tau \leq t \leq T)$. Although systems with delays are common in thermoacoustics [8, 9], only systems without delays will be considered for this paper.

Shooting methods are most applicable for finding limit cycles of large systems. Shooting methods find states at distinct locations on the limit cycle, but do not attempt to find the shape of the cycle.[1] The current guess for a state on the limit cycle, $\underline{x}(0)$, is iterated in order to satisfy the condition for a limit cycle that $\underline{x}(0) = \underline{x}(T)$, where $\underline{x}(T)$ is found with a timemarching process. Numerically, the iteration process is stopped when a specific level of convergence is reached, $||\underline{x}(T) - \underline{x}(0)|| < \epsilon$, $\{\epsilon \in \mathbb{R}^+ | \epsilon \neq 0\}$, where ϵ is small.

There are two main shooting methods, which will be referred to in this paper as standard shooting and Poincaré shooting.

2.1. Standard shooting

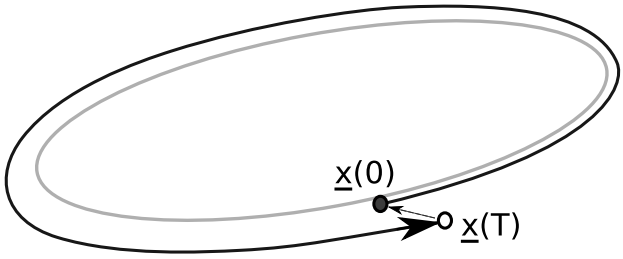


Figure 2: Standard shooting method to find a limit cycle. Given a current guess for a state on a limit cycle, $\underline{x}(0)$, we timemarch forward T time units to $\underline{x}(T)$, where T is our guess for the period. We then iterate our starting guess, $\underline{x}(0)$, to minimise the length of the residual vector, $\underline{x}(0) - \underline{x}(T)$, (dashed arrow).

The standard shooting method used in this paper finds, by iteration, a state on the limit cycle, $\underline{x}(0)$, and the period of the limit cycle, T. The magnitude of the residual vector, $\underline{r} = \underline{x}(0) - \underline{x}(T)$ (Fig. 2), is reduced to a predefined tolerance by a two-step iteration process. First, we consider the evolution of the system when started from small perturbations around our current guess $[\underline{x}(0), T]$. We generate a $(N+1) \times (N+1)$ Jacobian matrix, which relates a general small change in $[\underline{x}(0), T]$ to the resulting change in $[\underline{x}(0) - \underline{x}(T), T]$. Second, we solve a linear equation with the Jacobian matrix to find the $[\underline{\Delta x}, \Delta T]$ that we should add to our current guess, $[\underline{x}(0), T]$, in order to move closer to the limit cycle. If the magnitude of the residual is still too large, we repeat the first step from the improved guess.

Equation (3) shows the linear equation for the n^{th} iteration, where i and j are the row and column indices of the matrix [36]. It has the standard form for multi-dimensional Newton iteration, $J\underline{\Delta x} = -\underline{r}$.

$$\overset{(N+1)\times(N+1)}{\begin{bmatrix} I - M & \underline{b} \\ \underline{c} & d \end{bmatrix}} \overset{(N+1)\times 1}{\begin{bmatrix} \underline{\Delta x} \\ \Delta T \end{bmatrix}} = - \overset{(N+1)\times 1}{\begin{bmatrix} (\underline{x}(0) - \underline{x}(T))^n \\ \theta^n \end{bmatrix}} \tag{3}$$

[1] In contrast, collocation methods attempt to find the shape of the cycle in terms of piecewise polynomial splines. This is too expensive for large systems.

$$\overset{N\times N}{M_{i,j}} = \frac{\partial \underline{x}_i(T)}{\partial \underline{x}_j(0)},\ \overset{N\times 1}{\underline{b}_i} = -\frac{\partial \underline{x}_i(T)}{\partial T},\ \overset{1\times N}{\underline{c}_j} = \frac{\partial \theta}{\partial \underline{x}_j(0)},\ \overset{1\times 1}{d} = \frac{\partial \theta}{\partial T}$$

$$\underline{x}(0)^{n+1} = \underline{x}(0)^n + \underline{\Delta x}, \qquad T^{n+1} = T^n + \Delta T$$

There is an infinite number of points on a limit cycle that satisfy $\underline{r} = 0$, however, so a condition is required to fix the phase of the limit cycle (θ) and therefore provide a unique solution state. For a limit cycle in a thermoacoustic system, a suitable phase condition is that the instantaneous acoustic pressure in the fundamental mode is zero, or that the instantaneous acoustic pressure at a set location (x/L) of the combustor is zero. Where multiple acoustic modes are important, the value x/L must be irrational so that this location does not coincide with a node of any acoustic modes.

The j^{th} column of the Jacobian matrix can be numerically found in two ways: first, by perturbing $\underline{x}_j(0)$, then timemarching forward and measuring the resultant change in $\underline{x}(T)$, or second, by timemarching the first variational equations (section 3). To fill the Jacobian matrix for each linear equation, N timemarches are required. For large thermoacoustic systems, with $\mathcal{O}(10^3)$ variables, it is therefore impractical to form the Jacobian matrices, because the computational expense of timemarching is too high. This is the primary driver for the use of matrix-free methods.

In equation (3), the characteristics of the system are contained in the monodromy matrix M, which relates a change in $\underline{x}(0)$ to a change in $\underline{x}(T)$. In a dissipative system, such as in thermoacoustics, most of the eigenvalues of this matrix are clustered near zero. These correspond to quickly dissipated motions, because a change in $\underline{x}(0)$ causes very little change in $\underline{x}(T)$. The remaining few eigenvalues are not clustered near zero. These correspond to the bulk motions of the system, because a change in $\underline{x}(0)$ causes a significant change in $\underline{x}(T)$. These bulk motions will govern the flame-acoustic interaction (section 7.3.2).

2.2. Poincaré shooting

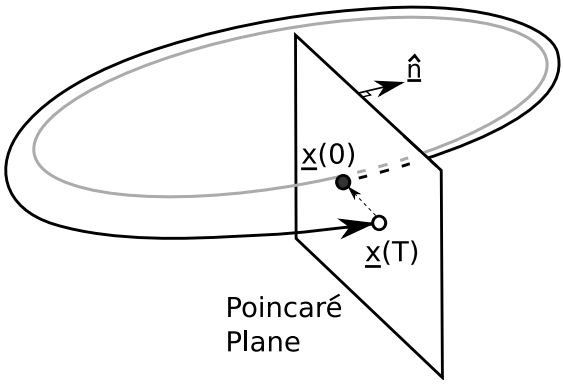


Figure 3: Poincaré shooting method to find a limit cycle. The starting state, $\underline{x}(0)$, and end state, $\underline{x}(T)$, are constrained to lie on a hyperplane perpendicular to the limit cycle. The system is timemarched until it crosses the Poincaré plane again, in the same direction as $\underline{\hat{n}}$. The period T is found as a byproduct.

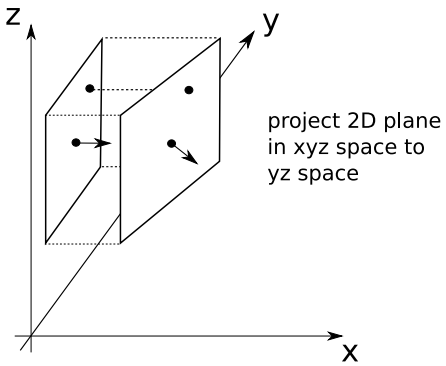
In the Poincaré shooting method, the solution is constrained to lie on a hyperplane that is perpendicular (or nearly perpendicular) to the limit cycle [27]. The starting state lies on the hyperplane, and is marched forward in time until it crosses the hyperplane again, in the same direction as $\underline{\hat{n}}$ (Figure 3). With the Poincaré shooting method, therefore, the iteration takes place in an $(N-1)$ dimensional space, defined by $\underline{\hat{n}} \cdot (\underline{x} - \underline{x}_0) = 0$, where $\underline{x}_0$ is a point on the hyperplane. The iteration is equivalent to finding the fixed point of a Poincaré map. The point on the hyperplane is taken as the current guess, or that of a nearby converged solution, $\underline{x}_0 = \underline{x}(0)$. The normal to the hyperplane is taken as the normalised time differential, $\underline{\hat{n}} = \underline{\hat{\dot{x}}}(0)$. In the standard shooting formulation, the period, T, is required as a variable, but in the Poincaré shooting formulation, the period is found as a byproduct of the timemarching process.

The linear equation for Poincaré shooting is in an $(N-1)$ dimensional space, so an efficient technique for projecting from N dimensions is required (and vice-versa). This is known as parametrisation. One method of parametrisation is shown in Figure 4 for a 3 dimensional case [27]. In Figure 4a, a 2D test plane is shown in xyz space. To parametrise the test plane, it is projected onto yz space and then only information in the yz plane is used in the iteration process. In this example the yz plane is chosen because the angle between the test plane and the yz plane is less than the angle between the test plane and the yx or xz planes. This is equivalent to projecting away the information corresponding to the dimension k with the highest value of $|\underline{n}_k|$, $1 \leq k \leq N$. Using the notation and method of Sanchez [27], the projection operator to discard the information in the k^{th} dimension is defined as R_k, where:

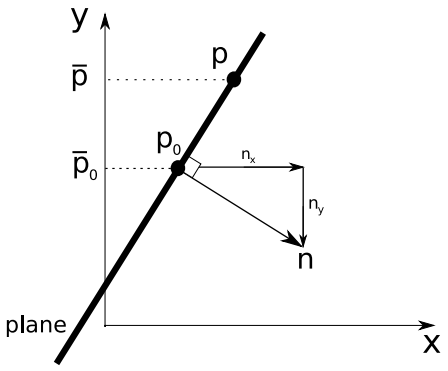
$$R_k \{x_1, x_2, \ldots, x_{k-1}, x_k, x_{k+1}, \ldots, x_N\} = \{x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_N\}$$

In the two dimensional example of Figure 4b, the action of the projector is $R_1(p) = \bar{p}$, $R_1(p_0) = \bar{p}_0$, where an overbar is used to denote an $(N-1)$ dimensional state on the plane. To reverse the process, if the components of the points are written as $\underline{p} = [p_x, p_y, p_z]$, then $p_x = p_{0x} + {}^{n_y(\bar{p}-\bar{p}_0)}/_{n_x}$, and $p_y = \bar{p}_y$ (note that n_y is negative). The inverse process to $\bar{R}_k$ is to return from the projected plane to the original plane, which is achieved by the operator E_k. Following the two dimensional argument, this is defined as:

$$E_k \{x_1, x_2, \ldots, x_{k-1}, x_{k+1}, \ldots, x_N\} = \left\{ x_1, x_2, \ldots, x_{k-1}, x_{0k} - \frac{\bar{\underline{n}} \cdot (\bar{\underline{x}} - \bar{\underline{x}}_0)}{\underline{n}_k}, x_{k+1}, \ldots, x_N \right\}$$



(a) 3D diagram of parametrisation of a plane.



(b) 2D diagram of parametrisation of a plane.

Figure 4: Parametrisation method, which takes an $(N-1)$ dimensional hyperplane with N variables, and projects it onto a space with only $(N-1)$ variables. This is a quick method of reducing the number of variables, so that the Poincaré shooting method can iterate in $(N-1)$ dimensions.

The iterative procedure is modified from equation (3), and now involves minimising the residual, $\bar{r}(\underline{x}(0), \lambda) = \bar{\underline{x}}(0) - \bar{\underline{x}}(T)$, with the following steps :

$$\overset{(N-1)\times(N-1)}{\begin{bmatrix} I - M \end{bmatrix}} \overset{(N-1)\times 1}{\begin{bmatrix} \Delta\bar{\underline{x}} \end{bmatrix}} = - \overset{(N-1)\times 1}{\begin{bmatrix} (\bar{\underline{x}}(0) - \bar{\underline{x}}(T))^n \end{bmatrix}} \tag{4}$$

$$\overset{(N-1)\times(N-1)}{M_{i,j}} = \frac{\partial \bar{\underline{x}}_i(T)}{\partial \bar{\underline{x}}_j(0)} \quad , \quad \underline{x}^{n+1}(0) = \underline{x}^n(0) + E_k\left(\Delta\bar{\underline{x}}\right), \quad \underline{x}_0 = \underline{x}^n(0), \quad \underline{n} = \hat{\underline{x}}^n(0)$$

The Poincaré shooting formulation has several advantages, most notably that if the plane is perpendicular to the cycle, the eigenvalues of the monodromy matrix (M) are the Floquet multipliers (without the trivial

multiplier of 1 which has an eigenvector in the direction of $\underline{\dot{x}}(0)$), and that the eigenvalues of the Jacobian matrix are shifted Floquet Multipliers, $eig_J = 1 - eig_M$. This is not true for the standard shooting method, because the border of the monodromy matrix contains terms that relate to the period ($\underline{b}, \underline{c}, d$ in equation (3)), and therefore the eigenvalues are not shifted Floquet multipliers. In continuation methods, the stability of the limit cycles is determined by these Floquet multipliers. With the Poincaré shooting method, the Floquet multipliers that are calculated when finding the stability can then be used directly to form a preconditioner for the Jacobian. This can significantly reduce the number of timemarches required to solve the next linear equation (see sections 4.1 and 7.4.1). The other main advantage of the Poincaré shooting formulation is that it is easier to find bifurcations of the limit cycle, such as Neimark-Sacker or period-doubling bifurcations.

The disadvantage of the Poincaré shooting method is that accurate techniques must be used to detect the exact crossing point on the hyperplane, $\underline{x}(T)$. Depending on the timemarching method used, this may be difficult to retrofit into existing thermoacoustic codes, and this may introduce some noise into the derivatives that form the Jacobian. In the ducted diffusion flame model of section 6, the exact crossing point of the hyperplane is found by varying the final timestep with Newton-Raphson iteration until the distance between the end of the trajectory and the hyperplane is less than a predefined tolerance.

3. Matrix-free methods

As stated in section 2.1, when finding limit cycles by the standard shooting method, N timemarches are required to form the Jacobian matrix ($N - 1$ for Poincaré shooting). This is unfeasible for large thermoacoustic systems. Alternative methods are therefore required to solve the Newton equation without having to form the Jacobian.

Matrix-free methods are those that solve the linear equation, $J\underline{\Delta x} = -\underline{r}$, without ever requiring the matrix J to be explicitly defined. The methods are iterative and only require matrix-vector products, i.e. $J\underline{v}$, where $\underline{v}$ is an arbitrary vector. This differs from many conventional methods of solving linear equations, where the matrix J is defined and then decomposed.

3.1. GMRES

The matrix-free method used in this paper to solve $J\underline{\Delta x} = -\underline{r}$ is the Generalised Minimal Residual method (GMRES) [37]. GMRES uses k matrix-vector products to define a k-dimensional Krylov subspace:

$$\mathcal{K}_k = \text{span}\left\{\underline{r}_0, J\underline{r}_0, J^2\underline{r}_0, J^3\underline{r}_0, ..., J^{k-1}\underline{r}_0\right\},$$

where $\underline{r}_0 \equiv -\underline{r} - J\underline{\Delta x}_0$, and $\underline{\Delta x}_0$ is an initial solution guess, often taken as the right hand side, $-\underline{r}$.

The vectors $\underline{r}_0, J\underline{r}_0, J^2\underline{r}_0, \ldots$ become almost linearly dependent [38], so the standard Arnoldi method is used to find orthonormal vectors, $\underline{q}_1, \ldots, \underline{q}_k$, that span the Krylov subspace $\mathcal{K}_k$. Modified Gram-Schmidt methods with re-orthogonalisation are generally used to orthonormalise the vectors for large systems, because the standard Gram-Schmidt method suffers from numerical problems with large state vectors.

In each Krylov subspace, the current guess for the solution, $\underline{\Delta x}_k$, is changed to minimise the residual:

$$\text{res}_k = ||-\underline{r} - J\underline{\Delta x}_k|| \quad \text{where} \quad \underline{\Delta x}_k \in \mathcal{K}_k$$

If convergence occurs, where $\text{res}_k < \epsilon_{conv}$, the iterative procedure stops. If $\text{res}_k > \epsilon_{conv}$, another matrix-vector product is taken to form $\mathcal{K}_{k+1}$. Another Arnoldi step is taken, to add another orthogonal direction $\underline{q}_{k+1}$. The residual is then minimised in this extra direction. The vectors $\underline{q}_1, \ldots, \underline{q}_k$ are unchanged in $\mathcal{K}_{k+1}$, and therefore the GMRES method converges monotonically.

3.2. Finite difference matrix-vector products

Matrix-free methods require accurate evaluation of general matrix-vector products ($J\underline{v}$). In the case of the Jacobian matrix, the matrix-vector product can be approximated by finite differences, because the Jacobian matrix is formed of partial derivatives [25]. In this section, a mapping operator, A, which represents the time marching process, is defined as:

$$\underline{x}(T) = A\left(\underline{x}(0)\right)$$

The spatial part of the Jacobian matrix is defined in equation (3), and can be written as:

$$J_{ij} = \frac{\partial \left(\underline{x}_i(0) - \underline{x}_i(T) \right)}{\partial \underline{x}_j(0)} \tag{5}$$

The matrix-vector product for arbitrary vector $\underline{v}$ can therefore be approximated by equation (6), where δ is small:

$$J\underline{v} = \underline{v} - \frac{A(\underline{x}(0) + \delta\underline{v}) - A(\underline{x}(0))}{\delta} + \mathcal{O}(\delta) \tag{6}$$

During the GMRES solution, evaluating each matrix-vector product therefore requires one timemarch, to calculate the $A(\underline{x}(0) + \delta\underline{v})$ term. In the standard shooting formulation, the phase (θ) components of the Jacobian matrix are also partial differentials ($\underline{b}, \underline{c}, d$ in equation (4)), and can similarly be calculated by a finite difference approach. The finite difference matrix-vector product approximation is implemented in the Trilinos NOX/LOCA solvers [29]. Finite difference matrix-vector products are simple to evaluate, but numerical problems may arise if the variables in the state vector are weighted poorly. This can occur when the variables are not non-dimensionalised, or when the variables act at different scales.

3.3. First variational matrix-vector products

The matrix-vector product can also be calculated by taking the first variational equations, sometimes referred to as tangent linear equations, and integrating them in time. This approach is used in the Navier-Stokes continuation of Sanchez [27, 32]. The first variational equations linearise about a trajectory, rather than a point. As in equation (1), for the k^{th} variable, the discretised nonlinear system evolves according to:

$$\frac{d\underline{x}_k}{dt} = F_k\left(\underline{x}\right)$$

If the first variational state is defined as $\underline{X}'$, then for the k^{th} variable, the first variational system evolves according to:

$$\frac{d\underline{X}'_k}{dt} = \sum_j \frac{\partial F_k\left(\underline{x}\right)}{\partial \underline{x}_j} \underline{X}'_j$$

If $\underline{X}'(0) = \underline{v}$, then, for the standard shooting formulation, the matrix-vector product for a general vector is:

$$J\underline{v} = \underline{v} - \underline{X}'(T) \tag{7}$$

For the Poincaré shooting formulation the first variational approach requires an extra term in the matrix-vector product evaluation. This arises because the time that the trajectory crosses the hyperplane depends on the initial state. The trajectory end state, $\underline{x}(T)$, lies on the plane, but the combined trajectory and perturbation state, $\underline{x}(T) + \underline{X}'(T)$, lies off the plane. If $\underline{X}'(0) = E_k(\bar{v})$, then the first variational matrix-vector product is [38]:

$$J\bar{v} = \bar{v} - R_k \left(\underline{X}'(T) - \dot{\underline{x}}(T) \frac{\hat{\underline{n}} \cdot \underline{X}'(T)}{\hat{\underline{n}} \cdot \dot{\underline{x}}(T)} \right)$$

The matrix-vector products calculated by the first variational equations are more accurate than those calculated by finite differences, and for highly nonlinear and chaotic systems the finite difference products can be qualitatively different [39, 40]. More calculations are required per timestep for the first variational equations, but the timestep can be made larger for the same matrix-vector product accuracy. This could lead to a faster timemarching process, which dominates the time taken to find a limit cycle. Although the first variational equations produce more accurate matrix-vector products, they may be time consuming to retrofit to existing thermoacoustic codes.

The accuracy of the finite difference and first variational approaches are compared in section 7.3.2, for the test model of a 2D diffusion flame in an acoustic duct.

4. Efficient continuation techniques

The matrix-free methods presented in the last section are efficient when converging to a single limit cycle. This section presents three additional techniques that are efficient when finding a series of limit cycles that span a parameter range: invariant subspace preconditioning, adaptive step sizing, and higher order prediction methods.

4.1. Invariant subspace preconditioning

To increase the convergence of GMRES, preconditioning can be used to reduce the spread of the eigenvalues, or to decrease the non-normality of the matrix. The linear system can either be left preconditioned, $P^{-1}J\underline{\Delta x} = -P^{-1}\underline{r}$, or right preconditioned, $JP^{-1}\left(P\underline{\Delta x}\right) = -\underline{r}$. The preconditioner is worth evaluating if the convergence rate of the GMRES algorithm is improved enough to outweigh the time in forming the preconditioning matrix. In continuation methods, if the step between solutions is small, then information about the previous solution can be used to form a preconditioner for the next. This is particularly useful where Krylov decompositions can be recycled, with no need for further timemarching.

Preconditioning can be achieved with many techniques, such as ILU factorisations, multigrid methods and approximate inverse methods. For the continuation of limit cycles in large systems, it is only practical to use preconditioners that can be evaluated with a matrix-free method. One such method is to take the spectral information of J that is available from the Arnoldi algorithm, and then use it to define an invariant subspace that is associated with a particular set of eigenvalues [41, 42, 32] (an invariant subspace is simply one that, under the action of an operator, maps onto itself). The eigenvalues of this invariant subspace can then be altered to make the linear equation easier to solve, in a similar manner to eigenvalue deflation in eigenvalue problems.

In a dissipative system, most of the eigenvalues of the monodromy matrix are clustered near zero (section 2.1). When finding limit cycles in a dissipative system, most of the eigenvalues of the Jacobian matrix are therefore clustered near $(+1, 0)$, because the Jacobian matrix is dominated by the $(I - M)$ block (equation (3)).To reduce the eigenvalue spread in a shooting formulation, the eigenvalues that are furthest away from the cluster at $(+1, 0)$ should be moved towards the cluster. The eigenvalues of interest are the eigenvalues of the monodromy matrix that have the largest magnitude (from section 2.1). The eigenvalues can be found in a matrix-free fashion with the Arnoldi algorithm or the Krylov-Schur algorithm of Stewart [35].

A k-dimensional partial-Schur decomposition of a matrix, J, takes the form $JU_k = U_kS_k$, where U_k is a $(N \times k)$ matrix with orthonormal columns, and S_k is an upper triangular $(k \times k)$ matrix.

$$\overset{N\times N}{\begin{bmatrix} & & \\ & J & \\ & & \end{bmatrix}}\overset{N\times k}{\begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \underline{\hat{u}}_1 & \underline{\hat{u}}_2 & \cdots & \underline{\hat{u}}_k \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}} = \overset{N\times k}{\begin{bmatrix} \uparrow & \uparrow & & \uparrow \\ \underline{\hat{u}}_1 & \underline{\hat{u}}_2 & \cdots & \underline{\hat{u}}_k \\ \downarrow & \downarrow & & \downarrow \end{bmatrix}}\overset{k\times k}{\begin{bmatrix} (J\underline{\hat{u}}_1)\cdot\underline{\hat{u}}_1 & (J\underline{\hat{u}}_2)\cdot\underline{\hat{u}}_1 & \cdots & (J\underline{\hat{u}}_k)\cdot\underline{\hat{u}}_1 \\ & (J\underline{\hat{u}}_2)\cdot\underline{\hat{u}}_2 & \cdots & (J\underline{\hat{u}}_k)\cdot\underline{\hat{u}}_2 \\ & & \ddots & \vdots \\ & & & (J\underline{\hat{u}}_k)\cdot\underline{\hat{u}}_k \end{bmatrix}} \tag{8}$$

From the construction of the partial-Schur decomposition, the inverse must exist, $J^{-1}U_k = U_kS_k^{-1}$, because S_k^{-1} must exist as long as $trace(S_k) \neq 0$. A partial-Schur decomposition is usually formed by taking the Arnoldi decomposition, $JV_k = V_kH_k$, where H_k is a k dimensional upper Hessenberg matrix, and further taking a QR decomposition, $H_kQ_k = Q_kS_k$, giving $JU_k = U_kS_k$ where $U_k = V_kQ_k$. Using reordering techniques, such as Given's rotations, it is possible to get a partial-Schur decomposition where the k columns of U_k correspond to the invariant subspace of the eigenvalues that we wish to deflate. The operator P is then defined as having the same eigenvalues of J in the k-dimensions of the invariant subspace, and as having an eigenvalue of 1 in all directions orthogonal to the invariant subspace.

$$P = U_kS_kU_k^T + \left(I - U_kU_k^T\right)$$

The preconditioner P^{-1}, which deflates the eigenvalues in the invariant subspace to (+1,0), is therefore of the form:

$$P^{-1} = U_k S_k^{-1} U_k^T + \left(I - U_k U_k^T\right)$$

When premultiplying by J, the matrix-preconditioner product simplifies to:

$$JP^{-1} = U_k U_k^T + J\left(I - U_k U_k^T\right)$$

The operator $U_k U_k^T$ is the projection operator onto the k-dimensional invariant subspace, which acts like an eigenvalue of 1 in the invariant subspace. Conversely, the operator $\left(I - U_k U_k^T\right)$ gives zero for all parts of the invariant subspace, and is the identity for all perpendicular subspaces. Therefore JP^{-1} acts as the identity on the invariant subspace, and as J on all perpendicular subspaces. The preconditioner can be applied in a matrix-free fashion if S_k and U_k are stored.

The concept of this preconditioner is similar to that used in the Newton-Picard method, where an invariant subspace is found that contains all eigenvalues above a certain magnitude [43, 36]. The iterative process is then split into Newton iteration in the invariant subspace and Picard iteration in all other directions. However, it is simpler to keep the invariant subspace information in a preconditioner, rather than splitting the iteration process [27].

4.2. Adaptive step sizing

Adaptive step sizing is used in continuation algorithms to examine a parameter range more quickly, whilst maintaining efficient and stable convergence of the iterative solver. Because Newton iteration can become unstable when started far from a solution, initial residuals must be kept low, or the iterative process may diverge[2]. Adaptive step sizing reduces the step size in regions where the solution is changing rapidly, and increases the step size in regions where the solution is changing slowly.

In this paper, the adaptive step size routine from the LOCA framework is used [29]. The step size is adapted based on the number of Newton iterations required for the last solution (ITS), the maximum number of Newton iterations allowed (ITS_{max}), and the angle between the last secant vectors [29]:

$$\Delta S = \Delta S_{last} \times \tau^y \times \left(1 + a\left(1 - \frac{ITS_{last}}{ITS_{max}}\right)^2\right)$$

where τ is the cosine of the angle between the last two pairs of solutions, and is a measure of how fast the solution is changing:

$$\tau = \left(\frac{\hat{\partial \underline{x}}}{\partial S}\right) \cdot \left(\frac{\hat{\partial \underline{x}}}{\partial S}\right)_{last}$$

This adaptive step size routine is basic, but can easily be tuned to suit different thermoacoustic problems. The factor a defines how aggressive the routine is in increasing step size, and the factor y defines how much the step size decreases when the solution is changing rapidly. A more robust implementation would be based on the total number of matrix-vector products required for the last solution, not just the number of Newton iterations [27].

4.3. Higher order prediction

Once a limit cycle is found, the limit cycle at the next parameter value must be predicted. The prediction can be a simple secant extrapolation from the last two solutions, or it can use a higher order prediction method. With a higher order prediction method, it is possible to make a prediction that has a lower initial residual, which is therefore likely to converge faster and has less chance of divergence. Alternatively, larger steps can be taken for the same magnitude of initial residual, reducing the number of points required

[2] In the literature, there are several globalisation strategies that can be used in tandem with Newton methods to increase the zone of convergence [44].

per bifurcation diagram. Because the previous solutions are already known and stored, the higher order prediction methods require no additional timemarching, so the computational expense is low.

One efficient method of implementing higher order prediction is to fit an n^{th}-order polynomial function to the last k solution vectors, where $n < k$. For each variable in the state vector, $\underline{x}$, a polynomial can be fitted using least squares regression. The polynomial can be defined as a function of a system parameter or as a function of the arclength of the solution curve. If the polynomial is a function of a parameter, the prediction will break down around fold bifurcations, because the polynomial is single valued. For this reason, the arclength will be used in this paper.

The arclength can be estimated as the cumulative sum of the distance between successive solution vectors. The least squares fitting procedure uses the arclength values from the last k solutions, $S_k, S_{k-1}, \dots S_1$, and the i^{th} variable from the last k solutions, $\underline{x}_{i,k}, \underline{x}_{i,k-1}, \dots, \underline{x}_{i,1}$ to find the polynomial coefficients for that variable, $\underline{p}_{i,n}, \underline{p}_{i,n-1}, \dots, \underline{p}_{i,1}$. We form a linear equation for each variable in the state:

$$\overset{k \times (n+1)}{\begin{bmatrix} S_k^n & \dots & S_k^2 & S_k & 1 \\ S_{k-1}^n & \dots & S_{k-1}^2 & S_{k-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_1^n & \dots & S_1^2 & S_1 & 1 \end{bmatrix}} \overset{(n+1) \times 1}{\begin{bmatrix} \underline{p}_{i,n} \\ \vdots \\ \underline{p}_{i,2} \\ \underline{p}_{i,1} \\ \underline{p}_{i,0} \end{bmatrix}} = \overset{k \times 1}{\begin{bmatrix} \underline{x}_{i,k} \\ \underline{x}_{i,k-1} \\ \vdots \\ \\ \vdots \\ \underline{x}_{i,1} \end{bmatrix}} \tag{9}$$

If we use the notation for this linear equation, $A\underline{p} = \underline{b}$, then we solve this equation using the least squares formulation $A^T A\underline{p} = A^T\underline{b}$. The columns of A will be independent if no repeated solutions are included by the continuation process. It may seem time consuming to solve a linear equation for each variable, but the matrix $A^T A$ is small, $((n+1) \times (n+1))$, and is the same for each variable, so its decomposition need only be found once. The polynomial can therefore be found efficiently for each variable by using a QR decomposition. Because the polynomial is fitted to each variable in the state vector separately, the method is efficient for vectors that are distributed across multiple processors.

The polynomial prediction works with either fixed parameter or pseudo-arclength continuation methods. With the latter, some iteration is required after the polynomials have been estimated, to find the predicted state that is both projected along the solution curve, and is the correct distance from the last solution.

The order of the prediction technique cannot be increased indefinitely, however, due to numerical errors arising from the scaling of the different terms in equation (9). Larger values of S^n increase the condition number of the matrix A, so if the values of S are far from 1, they should be shifted and scaled. A suitable shifting and scaling routine is $S_{new} = {}^{S-\bar{S}}/_{\sigma}$, where σ is the standard deviation of S. The order of the prediction technique that is most effective will be problem specific (§4.3).

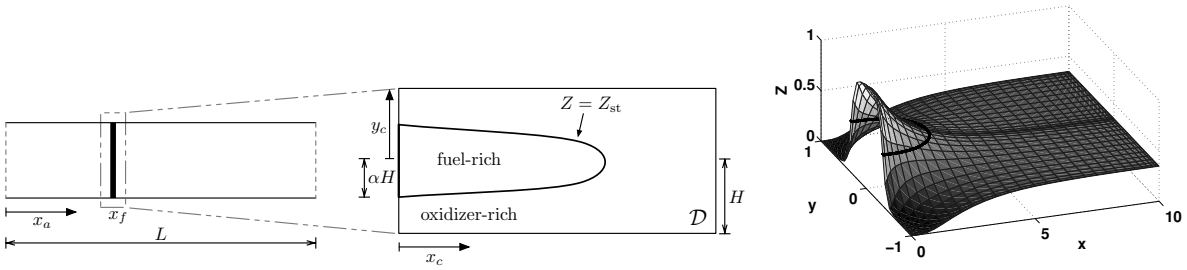
5. Software implementation

We have written a continuation code in C++, as a supplementary package to the Trilinos framework written by Sandia National Labs [45]. The package uses the GMRES solver from the NOX package, the parallelisable vectors from the Epetra package, some subroutines from the LOCA package [29] and the Krylov-Schur eigensolver from the Anasazi package [46]. The continuation code is designed to perform matrix-free continuation of limit cycles for general time dependent systems, and has been tested with several thermoacoustic models with serially and parallelly distributed discretisations. An object-oriented approach is used to switch simply between fixed point, standard shooting, Poincaré shooting, multiple shooting and Hopf bifurcation tracking algorithms.

6. Ducted 2D diffusion flame model

As described in the introduction, limit cycles occur in thermoacoustics when there is suitable coupling between an unsteady heat source and an acoustic field. The continuation routines are now applied to a

model of a thermoacoustic system containing a ducted 2D diffusion flame. Earlier versions of this model were discretised using a Galerkin method in the flame domain [47, 48]. This version was discretised by Illingworth using a spectral method in the flame domain [49]. This improves its numerical accuracy, which is important for the application of continuation methods. The model contains a 2D flame domain, where the mixture fraction, Z, is discretised on a Chebyshev grid, with an axis of symmetry along the centreline. The mixture fraction obeys the non-dimensional diffusion and advection governing equation, and is specified to be pure fuel in the fuel pipe ($Z = 1$, $0 < |y| < \alpha$) and pure oxidiser in the oxidiser pipe ($Z = 0$, $\alpha < |y| < 1$), with boundary conditions such that there is no diffusion into the walls, ${}^{\partial Z}/_{\partial y}|_{+1,-1} = 0$, and such that the mixture fraction is homogenous far downstream, ${}^{\partial Z}/_{\partial x}|_{10} = 0$. The flame lies on the stoichiometric contour, $Z = Z_{st}$, and is assumed to have an infinite reaction rate. The location of the flame is denoted f^+.



(a) The geometry of the 1D acoustic duct and of the 2D flame field.

(b) 2D steady state Z field.

Figure 5: A 2D mixture fraction field (Z) is used to describe the diffusion flame. The flame lies on the $Z = Z_{st}$ contour, which is shown as a black line in (a) and (b). The heat release from the diffusion flame domain acts at location x_f in a 1D open ended duct.

The heat release from the model is coupled to a simple linear acoustic model of a duct [48, 50, 8], where u and p are the non-dimensional velocity and pressure perturbations in the duct. For the perturbation state, $\underline{x} = [u, p, z]$, the non-dimensional governing equations are shown below, where the subscript f denotes that the value is taken at the flame position, and a bar refers to a steady state quantity. The system has non-dimensional parameters: Peclet number Pe (the ratio of advection to diffusion), the acoustic damping[3] ζ, the stoichiometric mixture fraction Z_{st}, the flame position in the duct x_f, the fuel pipe width α, and the coupling parameter β_T, defined by $\beta_T = 2/(T_{inlet} + T_{adiabatic})$.

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} \tag{10}$$

$$\frac{\partial p}{\partial t} = -\frac{\partial u}{\partial x} - \zeta p + \frac{2\beta_T}{1 - Z_{st}} \delta(x - x_f) \left(- \int \int_0^{y=f^+} \frac{\partial z}{\partial t} \, dy dx + u_f \int_0^{\bar{f}^+} \left(\bar{Z} - Z_{st} \right) dy \right) \tag{11}$$

$$\frac{\partial z}{\partial t} = -\bar{u}_f \frac{\partial z}{\partial x} + \frac{1}{Pe} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) z - u_f \frac{\partial \bar{Z}}{\partial x} - u_f \frac{\partial z}{\partial x} \tag{12}$$

Equation (10) is derived from the acoustic momentum equation [8, 48]. Equation (11) is derived from the acoustic energy equation [8, 48], where the last term corresponds to the heat release from the flame field (similar to Refs. [47, 48]). Equation (12) is derived from the 2D advection-diffusion equation [47, 48], with a constant velocity over the flame field, u_f.

[3]The acoustic damping is defined as $\zeta_j = c_1 j^2 + c_2 \sqrt{j}$, where j is the acoustic mode number [8].

If the first variational state vector is defined as $\underline{X'} = [U', P', Z']$, the first variational equations are:

$$\begin{aligned}
\frac{\partial U'}{\partial t} &= -\frac{\partial P'}{\partial x} \\
\frac{\partial P'}{\partial t} &= -\frac{\partial U'}{\partial x} - \zeta P' \\
&\quad + \frac{2\beta_T}{1 - Z_{st}} \delta\left(x - x_f\right) \left(- \int \int_0^{f^+} \frac{\partial Z'}{\partial t}\, dy dx - \int_{f^+} \frac{\partial z}{\partial t} \left|\underline{\nabla Z}\right|^{-1} Z'\, dl + U'_f \int_0^{\bar{f}^+} \left(\bar{Z} - Z_{st}\right) dy \right) \\
\frac{\partial Z'}{\partial t} &= -\bar{u}_f \frac{\partial Z'}{\partial x} + \frac{1}{Pe} \left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) Z' - U'_f \frac{\partial \bar{Z}}{\partial x} - U'_f \frac{\partial z}{\partial x} - u_f \frac{\partial Z'}{\partial x}
\end{aligned}$$

A more complete description of the model is included in Ref. [49]. The results in this paper are generated with a 30×16 symmetric flame grid and 20 acoustic modes, giving a total system dimension of 475 (the boundary values of the Chebyshev grid are defined by the boundary conditions so are excluded from this number). The model is timemarched in FORTRAN using the standard Runge-Kutta 4 technique with a timestep of 10^{-3}. The results of the model have been compared against those generated with a finer Chebyshev grid and a finer timestep, with only a 1% difference in heat release fluctuation observed.

7. Numerical results

7.1. Fixed point plane

Figure 6 shows the stability of the fixed point solution for the ducted diffusion flame, as two parameters are varied: Peclet number, which changes the ratio of advection to diffusion in the flame, and β_T, which controls the extent to which unsteady combustion perturbs the acoustics (equation (11)). The parameters that are held fixed are $Z_{st} = 0.8$, $c_1 = 0.0247$, $c_2 = 0.018$, $\alpha = 0.35$, $x_f = 0.25$, ${}^{ML}/_{H} = 1$. The damping coefficients are typical for a laboratory scale combustor, and $Z_{st} = 0.8$ corresponds to diluted methane fuel and pure oxygen [p.94, 51].

By construction, the fixed point solution for the ducted diffusion flame model is when the state vector is zero, because the state vector contains only perturbation quantities relative to mean values. The Hopf bifurcation marks the boundary between linearly stable and linearly unstable operating conditions. It is a strong function of Pe and β_T. The bistable region has both a stable fixed point and a stable limit cycle. The extent of the bistable region cannot be determined by finding the fixed points alone, therefore, because the surface of limit cycles must also be found. The limit cycle surface is shown in the next section.

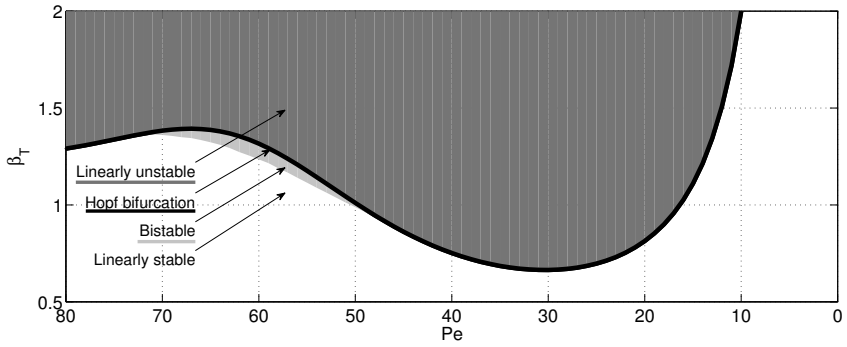


Figure 6: Stability of the fixed point solution, as a function of Pe and β_T. Note that the Pe axis is reversed to match the view in Figure 7.

7.2. Limit cycle surface

Figure 7 shows the limit cycle amplitudes as a function of Pe and β_T, with the same fixed parameters as section 7.1 . Each limit cycle is calculated to a tolerance of $||\underline{x}(0)-\underline{x}(T)|| < 10^{-8}$. The z-axis represents the amplitude of sound generated in the duct by a limit cycle; it is defined as the amplitude of acoustic velocity variation during the trajectory of a limit cycle. The x-y plane of Figure 7 is therefore Figure 6.

The Hopf bifurcation marks the boundary between linearly stable and linearly unstable operating conditions. The limit cycles form a surface which has both subcritical bifurcations for $50 < Pe < 70$ (Fig. 7a) and supercritical bifurcations for $Pe < 50$, $Pe > 70$ (Fig. 7c). Where there is a subcritical bifurcation, there is a stable limit cycle at higher velocity amplitudes. However, this stable limit cycle has velocity amplitude greater than 2 and is not shown in the figure. The bistable operating conditions are those at which the system has both a stable fixed point and a stable limit cycle.

The limit cycles describe the behaviour of the fully coupled system, and are calculated by the continuation method quite cheaply: the Hopf bifurcation line takes roughly 500s and the surface of limit cycles takes 61 CPU hours[4]. The surface is composed of 70 slices and roughly 2500 converged limit cycles, requiring an average of 52 minutes per slice, and 90 seconds per limit cycle. A lower resolution surface can be calculated in less than 10 CPU hours, with 15 slices, coarser spacing between limit cycles, and use of preconditioning. The computation can be easily parallelised because the surface is composed of separate two-dimensional slices.

As described in the introduction, the FDF method is often used to predict limit cycle amplitudes in thermoacoustics. The efficiency of the continuation method, relative to the FDF method, depends on the information that is required. If the flame operating condition is fixed, and the acoustic operating condition is varied, then the FDF method is efficient, because only one FDF evaluation is required. If both the flame and the acoustic operating conditions are varied, however, then the FDF method is inefficient, because the FDF must be re-evaluated at each new flame operating condition. The major advantage of the continuation method over the FDF method is that it is efficient when both the flame and the acoustic operating conditions are varied. In Figure 7, one parameter sets the flame operating condition (Pe), and the other sets the acoustic operating condition (β_T), so one FDF evaluation is required per 2D slice. Using this particular diffusion flame model, one FDF evaluation takes roughly 16 CPU hours [49], so the FDF would require roughly 1120 CPU hours to recreate Figure 7, which is 18 times slower.

The next subsections analyse the results for the ducted 2D diffusion flame model in terms of the physical significance and the efficiency of the numerics. Further physical interpretation of the results is the subject of a separate publication [49].

7.3. Physical significance of the results

The limit cycles and their Floquet multipliers can be examined in order to understand the coupled flame-acoustic interaction. Floquet multipliers describe the evolution of a perturbation around a limit cycle. The largest magnitude Floquet multipliers correspond to the most influential coupled flame-acoustic modes (section 7.3.2).

7.3.1. Limit cycles

Figure 8 shows snapshots of the system during an unstable limit cycle, with Pe of 60, β_T of 1.31 and velocity amplitude 0.62. When the velocity perturbation at the flame is positive, 8(a)-(c), the flame becomes longer and, near the inlet, the regions of high and low z are stretched. When the velocity perturbation at the flame becomes negative, 8(d)-(f), the flame becomes shorter and, near the inlet, new regions of high and low z are formed. In turn, these new regions are then stretched and convected down the flame. For this limit cycle, the length of the flame only varies by $\Delta x_c \approx 0.4$ during the cycle, but slight wrinkling of the flame surface can be seen as the regions of high and low z are convected down the flame. The limit cycle has an almost symmetric form during the first and second halves of the limit cycle, which demonstrates that the nonlinearity in this flame model has weak response at even harmonics.

[4]All computations run with an Intel i7-2600 3.4GHz processor.

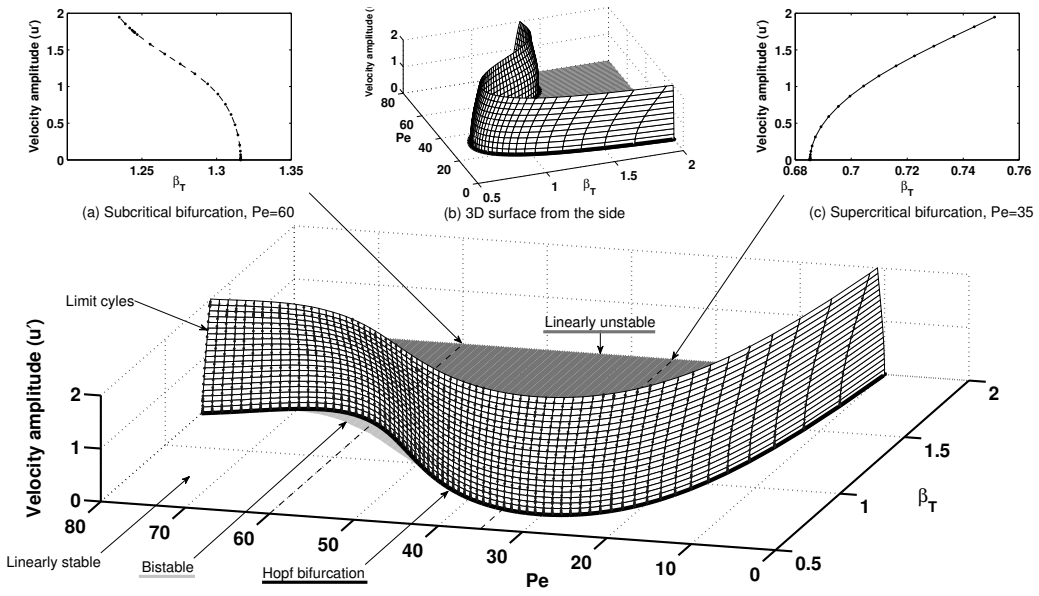


Figure 7: Continuation results for the ducted diffusion flame, as a function of Pe and β_T. The thick dark line is the Hopf bifurcation, which is the same as the linear stability limit. The linearly unstable region is shaded dark gray and the bistable region is shaded light gray (the bistable region is the vertical shadow of the subcritical bifurcation). The limit cycles are shown as dark dots on a white surface. The surface is defined by slices taken at 70 Pe numbers. The z-coordinate of each limit cycle is the amplitude of the acoustic velocity variation over its trajectory. The surface exhibits both subcritical bifurcations (a) and supercritical bifurcations (c), where stable limit cycles are shown with a solid line and unstable limit cycles with a dashed line. Further physical analysis of these results is found in Ref. [49].

7.3.2. Floquet multipliers

Figure 9 shows the Floquet multipliers for an unstable limit cycle ($Pe = 60$, $\beta_T = 1.31$). The majority of the multipliers are located near the origin and correspond to the motions of the system that are quickly dissipated. The rest of the multipliers correspond to the bulk motions of the system. The bulk motions are not quickly dissipated and therefore they dominate in the long time limit. The GMRES algorithm converges efficiently to limit cycles because of this simple distinction between the bulk motions and the dissipated motions. This will be analysed further in section 7.4.3.

Figure 10 shows the Floquet modes corresponding to the leading Floquet multipliers, which are located on the 'arm' extending to the right in Figure 9. These are the influential bulk motions of the system. In Figure 10 (a)-(g), the modes have large components in the $1^{\text{st}} - 7^{\text{th}}$ acoustic modes respectively, and affect the z field around the flame, primarily near the inlet. The Floquet modes corresponding to the 2^{nd} and 6^{th} acoustic modes, Figure 10 (b) and (f), have no component in the flame field. This is because the flame is located at $x_f = 0.25$, which is a velocity node for these modes.

The first Floquet multiplier has no imaginary component because it represents monotonic growth away from this unstable limit cycle. The others have imaginary components because they oscillate (as well as grow or decay) around the unstable limit cycle. The first Floquet mode has a large component in the fundamental acoustic mode and has the same period as the limit cycle. The mode shape is similar to that of the unstable limit cycle (Figure 8). A perturbation in the direction of the unstable Floquet mode will make the amplitude of the oscillation grow, and the system will reach the stable limit cycle at higher amplitude. Similar results have been shown in a numerical model of a Rijke tube, and presented as a mechanism for triggering [8, 52, 16].

Figure 11 shows a selection of typical Floquet modes corresponding to small Floquet multipliers, which

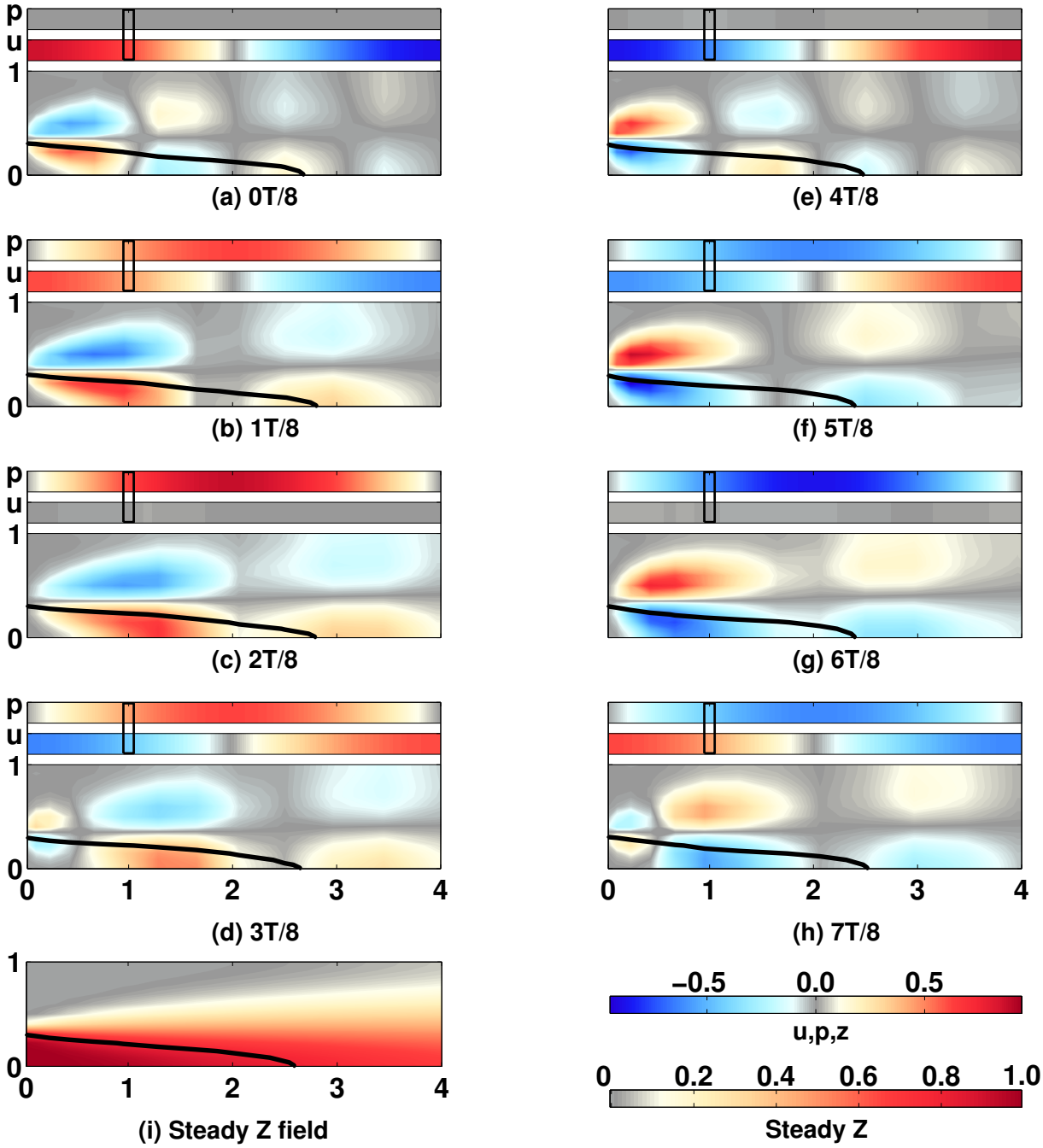


Figure 8: Snapshots of an unstable limit cycle (Pe=60, β_T=1.31), with velocity amplitude 0.62. In each snapshot (a-h), the top two bars show the perturbation pressures and velocities in the 1D duct ($x_a = 0 \rightarrow 1$), and the lower bar shows the 2D perturbation z field and current flame location (black line) in the first part of the flame field ($x_c = 0 \rightarrow 4$). The flame location in the duct is marked by the black box at $x_a = 0.25$. The z values are scaled by a factor of 15 to share the colourbar with the acoustic perturbations. The steady Z field is shown in (i) for comparison.

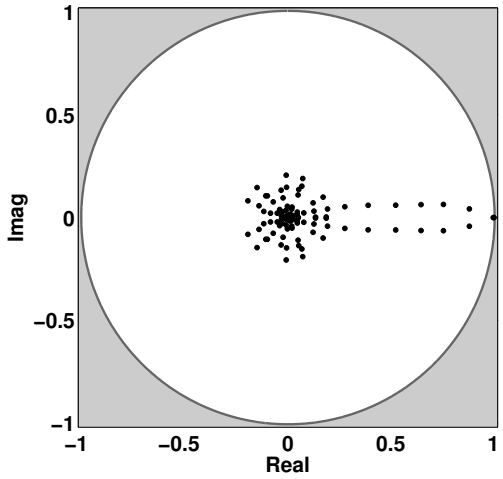


Figure 9: Floquet multipliers for an unstable limit cycle (leading multiplier at $1.000455 + 0i$). Any Floquet multipliers in the grey region correspond to unstable perturbations to the limit cycle.

are located around the origin in Figure 9. These are the quickly dissipated motions of the system. Many of the dissipated modes are dominated by the numerical boundary conditions, which are set to ${}^{\partial z}/_{\partial y}|_{+1,-1} = 0$, ${}^{\partial z}/_{\partial x}|_{10} = 0$. These modes are specific to the Chebyshev discretisation used for the z field. Modes corresponding to these numerical boundary conditions are seen in Figure 11(a,b,c,e). In Figure 11(d) the mode corresponds to the 20^{th} acoustic mode, which is heavily damped. Above the 250^{th} Floquet mode, as in Figure 11(f,g), the modes correspond to small scale features in the z field.

Sections 3.2 and 3.3 introduced two matrix-free approaches for evaluating matrix-vector products: the finite difference approach and the first variational approach. For the limit cycle in Figure 9, Table 1 shows the difference in the Floquet multipliers when evaluated with these two approaches (with $\delta = 10^{-8}$ in equation (6)). The results are almost identical for the value of δ used, which is probably because the ducted diffusion flame model has weak nonlinearities. For a model with stronger nonlinearities, these results may even be qualitatively different.

The first variational approach takes significantly more time to evaluate for this model, however, due to the the line integral term that appears in the first variational equations. This term is evaluated slowly with a Chebyshev discretisation. The results in this paper are therefore generated with the finite difference approach.

7.4. Efficiency of the numerics

7.4.1. Convergence to a limit cycle

Figure 12 shows the rate of convergence to a limit cycle for both the standard shooting and Poincaré shooting formulations. A previously converged limit cycle is used as a starting guess with a change in parameter $\Delta Pe = 1$, which represents convergence from a relatively poor initial guess. The convergence to a limit cycle depends on the system's operating condition, but this is a representative case for the diffusion flame model.

The solution converges steadily to the limit cycle and reaches the required tolerance, $||\underline{x}(0) - \underline{x}(T)|| < 10^{-8}$, in four Newton steps ($\underline{x}(0)^{n+1} = \underline{x}(0)^n + \underline{\Delta x}$). Each Newton step requires the iterative solution of $J\underline{\Delta x} = -\underline{r}$ by GMRES, to a relative convergence tolerance of $\frac{||\underline{r} + J\underline{\Delta x}||}{||\underline{r}||} < 10^{-3}$. In Figure 12, the dots are the actual residuals after each Newton step, and the line is the estimated residual within the solution of each Newton step. The convergence after the first Newton step is relatively poor. Poor convergence is common for Newton based methods when far from a solution. In the last two Newton steps, the convergence is much better, and the improvement in the actual residual matches well with that estimated by GMRES.

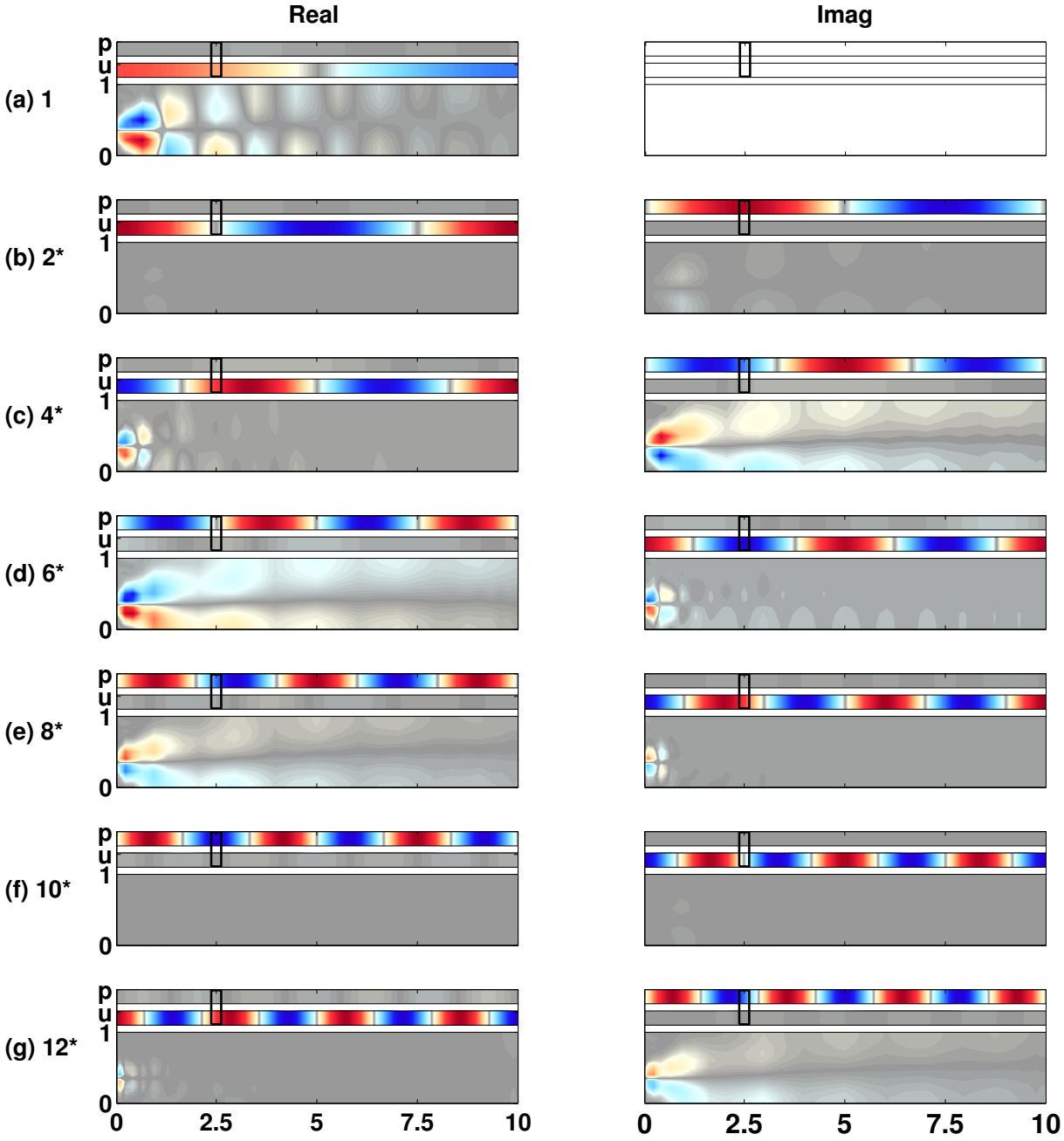


Figure 10: Leading Floquet modes for an unstable limit cycle (Pe=60, β_T=1.31), corresponding to the influential bulk motions of the system. The modes are sorted by the absolute values of the Floquet multipliers, in decreasing order. A star indicates that the mode is one part of a complex conjugate pair. In (a)-(g), the modes have large components in the $1^{\text{st}} - 7^{\text{th}}$ acoustic modes respectively. The z values are scaled by a factor of 35 to share the colourbar with the acoustic perturbations.

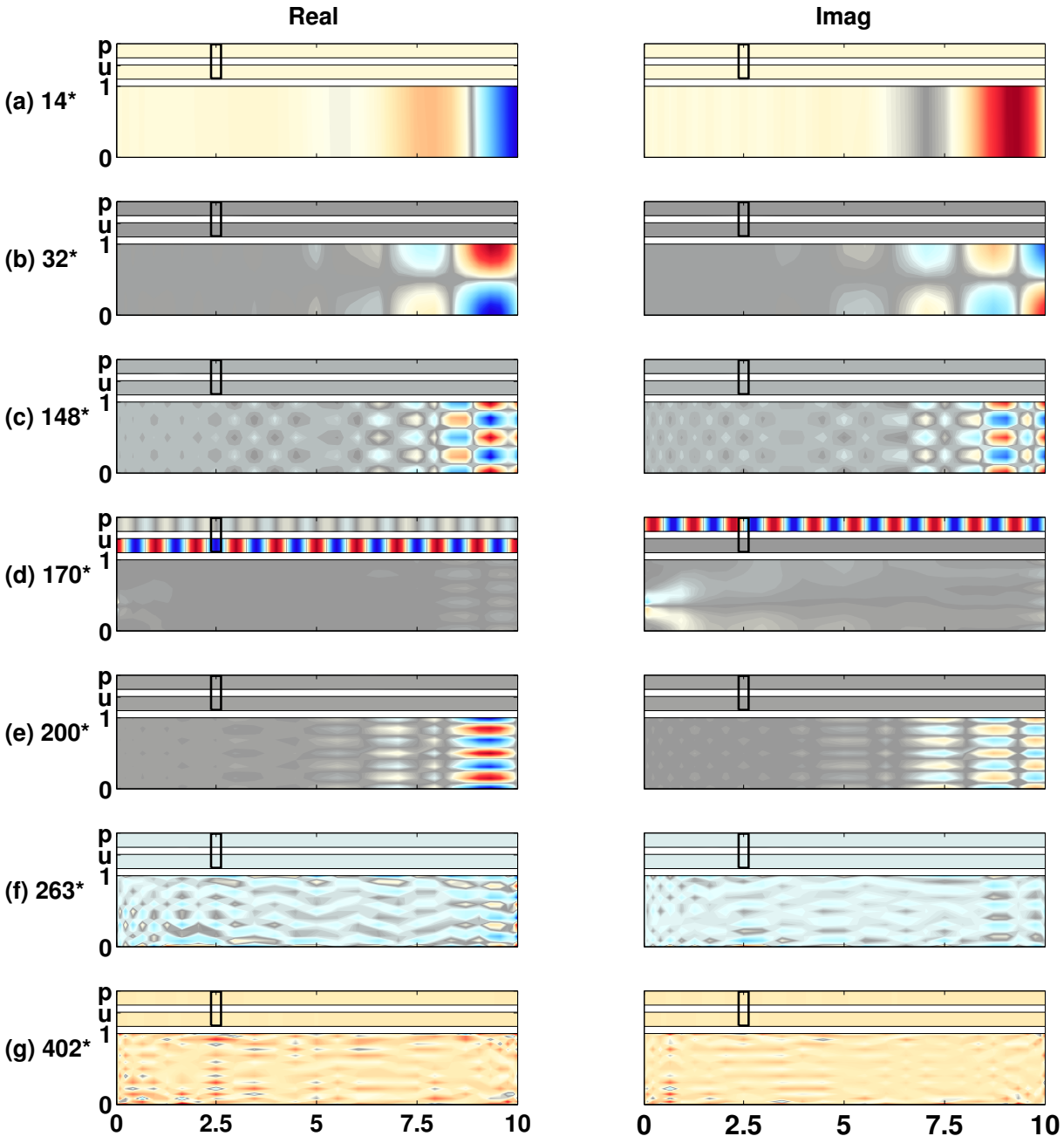
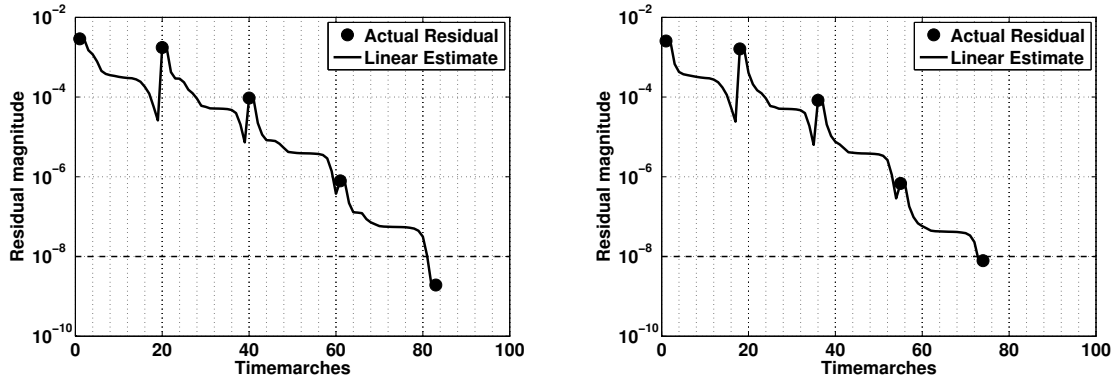


Figure 11: A selection of Floquet modes for an unstable limit cycle (Pe=60, β_T=1.31), corresponding to typical dissipated motions of the system. The modes are sorted by the absolute values of the Floquet multipliers, in decreasing order. A star indicates that the mode is one part of a complex conjugate pair. In (a,b,c,e), the modes are dominated by the numerical boundary conditions at $x_c = 10$ and $y_c = 1$. In (d) the mode corresponds to the 20^{th} acoustic mode, which is heavily damped. Above the 250^{th} Floquet mode, as in (f,g), the modes correspond to small scale features in the z field. The z values are scaled by a factor of 35 to share the colourbar with the acoustic perturbations.

Floquet Multiplier	Finite Difference	First Variational	Discrepancy
1	1.0005e+00	1.0005e+00	1.1363e-06
2*	8.8258e-01 - 4.1732e-02i	8.8258e-01 - 4.1732e-02i	3.1478e-11
4*	7.5645e-01 - 6.4094e-02i	7.5645e-01 - 6.4094e-02i	1.2788e-07
6*	6.4771e-01 - 6.2999e-02i	6.4771e-01 - 6.2999e-02i	2.8708e-15
8*	5.2433e-01 - 6.0199e-02i	5.2433e-01 - 6.0199e-02i	5.7604e-08
167*	4.7382e-05 + 2.7488e-05i	4.7393e-05 + 2.7518e-05i	3.2359e-08
169*	5.1481e-05 - 8.3624e-07i	5.1487e-05 - 8.0574e-07i	3.1135e-08
171*	3.1964e-05 - 3.7150e-05i	3.1964e-05 - 3.7150e-05i	1.3370e-15
173*	1.8760e-05 - 4.2192e-05i	1.8760e-05 - 4.2195e-05i	2.8777e-09
175*	-2.7625e-05 - 2.9150e-05i	-2.7626e-05 - 2.9149e-05i	1.2237e-09
467*	-7.9359e-13 + 6.6930e-13i	1.1102e-16	1.0382e-12
469*	8.4577e-13 + 4.9561e-13i	1.1102e-16	9.8019e-13
471*	2.6668e-13 + 3.5235e-13i	0	4.4189e-13
473*	-9.4147e-14 + 3.7290e-13i	0	3.8460e-13
475*	1.2035e-13 + 9.8341e-14i	0	1.5542e-13

Table 1: A selection of Floquet multipliers as calculated by the finite difference and first variational matrix-vector product approaches. The Floquet multipliers are sorted by their absolute values, in order of decreasing magnitude. The star indicates that the Floquet multiplier is one part of a complex conjugate pair. The discrepancy is the magnitude of the difference between the two eigenvalues.



(a) Standard shooting - no preconditioning

(b) Poincaré shooting - no preconditioning

Figure 12: Convergence to a limit cycle in terms of the residual magnitude, $||\underline{x}(0)-\underline{x}(T)||_2$, and number of timemarches. The dots are the residual values after each Newton step and the line is the the estimated residual within the solution of each Newton step. As the solution converges, the estimated reduction in residual for each Newton step approaches the actual residual reduction achieved.

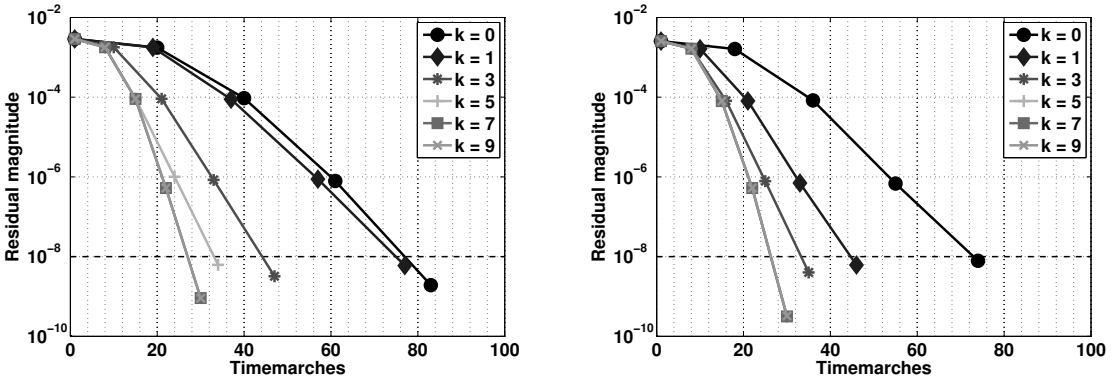
The Poincaré shooting method converges faster than the standard shooting method for two reasons. Firstly, Poincaré shooting does not include the period in the state vector, and therefore the Jacobian has one fewer influential eigenvalue. Secondly, the Jacobian for the standard shooting method has an additional eigenvalue near zero, which raises the condition number of the matrix. It is harder for an iterative algorithm to solve a linear equation if the matrix has a high condition number. The eigenvalue near zero results from the trivial Floquet multiplier of +1 in the direction of $\dot{\underline{x}}(0)$, which is the dimension that is projected away in the Poincaré shooting formulation.

Most importantly, with both methods the residual reduces by a factor of 10^5 using only 80 timemarches,

when there are 475 variables in the system. If a matrix-free method were not used, and the Jacobian matrices were explicitly formed for each Newton step, then four Newton steps would have required $4 \times 475 = 1900$ timemarches. The matrix-free method is therefore 24 times faster, even for this modestly sized test model. In the matrix-free method, the number of timemarches required does not depend on the number of variables. It depends on the number of influential bulk motions. For example, if the resolution of the Chebyshev grid were increased by a factor of two, the number of variables would increase, but the solution would still be found in roughly 80 timemarches, because the increase in resolution adds no bulk motions to the dynamics.

7.4.2. Convergence to a limit cycle with preconditioning

Using the same starting condition as Figure 12, Figure 13 shows the convergence of the residual when preconditioning is used. The Jacobian matrix from the previous converged limit cycle is used to form an invariant subspace preconditioner, with k deflated eigenvalues (section 4.1). The eigenvalues are converged to a precision of 10^{-4}.



(a) Standard shooting - with preconditioning

(b) Poincaré shooting - with preconditioning

Figure 13: Convergence to a limit cycle in terms of the residual magnitude, $||\underline{x}(0) - \underline{x}(T)||_2$, and number of timemarches. Preconditioning is used to increase the convergence of the GMRES solver. Results are plotted for preconditioners with different numbers of deflated eigenvalues (k). Estimated residuals within the solution of the Newton steps are not shown.

As the number of preconditioned eigenvalues increases, the number of timemarches required for convergence steadily decreases. For a preconditioner with 7 converged eigenvalues, the number of timemarches required is greatly reduced, from 83 timemarches to 30. To converge to the 7 eigenvalues and form the preconditioner, however, requires 29 timemarches, so there is only an overall reduction of 24 timemarches. The inverse of the preconditioner must be applied before each of the 30 timemarches, but the additional cost of this application is negligible compared to the cost of timemarching.

The preconditioner would significantly increase the speed of continuation if it were re-used to find several limit cycles, or if it were generated with negligible extra cost. The latter is true if the stability of the limit cycles is calculated post convergence, and the Floquet multipliers are used to form the preconditioner. With the Poincaré shooting method, the Floquet multipliers can be directly used for preconditioning, because the eigenvalues of the Jacobian are simply shifted Floquet multipliers ($eig_J = 1 - eig_M$, section 2.2). With the standard shooting method, however, preconditioning is more complicated because the terms related to the period add another row and column to the borders of the Jacobian matrix, and therefore the eigenvalues of the Jacobian matrix are no longer simply the shifted Floquet multipliers ($eig_J \neq 1 - eig_M$). The same is true for pseudo-arclength continuation methods, which add an additional row and column to the borders of the Jacobian matrix. In these two cases the Floquet Multipliers can be used for preconditioning if bordering methods are used to solve the linear equation. Bordering methods split the Jacobian matrix into a large square part and the resulting small bordering vectors, as shown in equation (3). The bordered form of the

matrix can be used to define a preconditioner [53, 54], or to solve a linear equation in ordered steps [55]. For equation (3) and dummy solution states $\underline{\Delta x}_1, \underline{\Delta x}_2$, a simple example of a bordering method would be to solve: (1) $(I-M)\underline{\Delta x}_1 = -(\underline{x}(0)-\underline{x}(T))$, then (2) $(I-M)\underline{\Delta x}_2 = \underline{b}$, then (3)$(d-\underline{c}\underline{\Delta x}_2)\Delta T = -\theta - \underline{c}\underline{\Delta x}_1$, then (4) $\underline{\Delta x} = \underline{\Delta x}_1 - \underline{\Delta x}_2 \Delta T$. The Floquet multipliers can be used to precondition steps (1) and (2), which are the most computationally expensive. Bordering methods are discussed in depth in several publications [55, 53, 54], and will not be covered further in this paper.

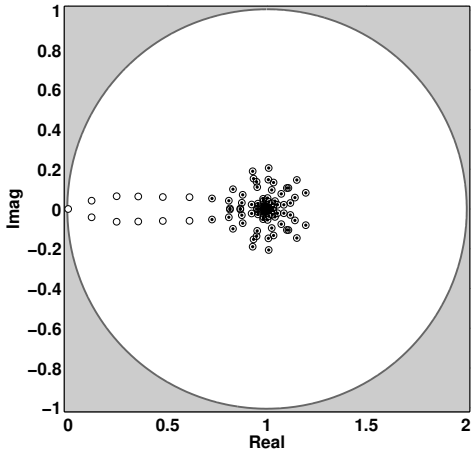


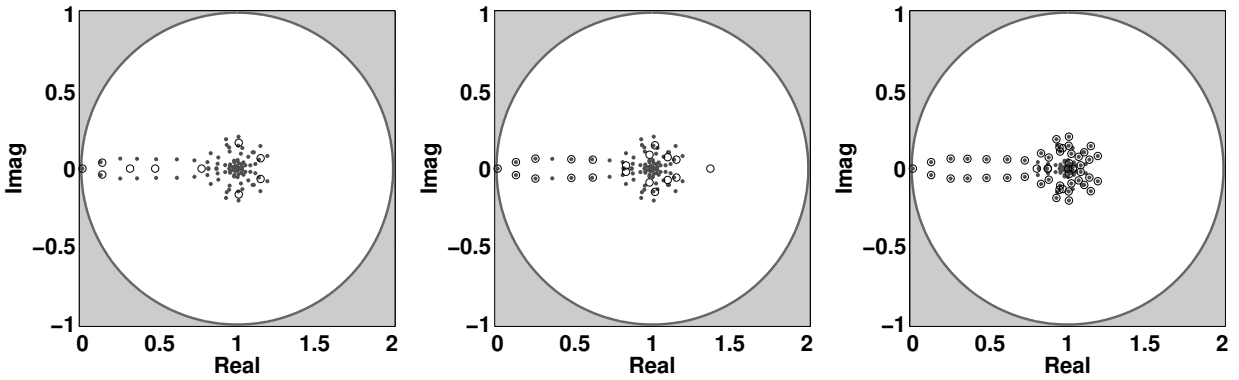
Figure 14: Demonstration of invariant subspace preconditioning, for an eigenspace of the 11 eigenvalues furthest from (+1,0). The eigenvalues of the Jacobian are shown as open circles and the eigenvalues of the preconditioned Jacobian are shown as black dots. The preconditioner takes the eigenvalues furthest from (+1,0), and moves them to (+1,0). Because the Jacobian has an eigenvalue close to zero (-0.000455,0), the preconditioned Jacobian is much better conditioned.

Figure 14 shows the effect of the preconditioner on the Jacobian matrix. The subspace chosen for preconditioning corresponds to the 11 eigenvalues furthest from (+1,0). In Figure 14, the eigenvalues in the subspace were converged to a high tolerance, 10^{-6}, using 40 timemarches. Therefore when the preconditioner is applied, the eigenvalues outside of the subspace are almost unaffected. If there is inaccuracy in the estimated eigenvalues (and eigenvectors) in the subspace, then the eigenvalues outside of the subspace will be altered by the preconditioner. This effect is stronger in more non-normal systems. If the preconditioner moves the eigenvalues outside of the subspace, there is no guarantee that the preconditioning will increase convergence.

Figure 14 also shows that the preconditioner makes the Jacobian better conditioned. The limit cycle is taken from the unstable branch of limit cycles in the bistable region at Pe of 60 (Figure 7). The cycle has one Floquet multiplier that is just unstable, with value $1.000455 + 0i$. The Jacobian therefore has an eigenvalue of $-0.000455 + 0i$, which gives the Jacobian a high condition number of 4638 (based on the ratio of the maximum and minimum singular values). The preconditioned Jacobian has a condition number of 8. A high condition number can also occur when a limit cycle is extremely unstable. High condition numbers result in numerical inaccuracies in iterative processes, and may require a larger Krylov subspace for a solution. It is therefore good to avoid high condition numbers, by using a preconditioner.

7.4.3. Convergence of the GMRES solver

The aim of this section is to demonstrate that the GMRES solver intrinsically uses a 'reduced order model' property when finding limit cycles. In other words, it converges first in the eigenspace corresponding to the bulk motions of the system, and then in the eigenspace corresponding to the quickly dissipated motions. Figure 15 shows the Ritz values and relative residuals as the dimension of the Krylov subspace



(a) Krylov dimension 10
relative residual 0.913

(b) Krylov dimension 20
relative residual 3.45e−05

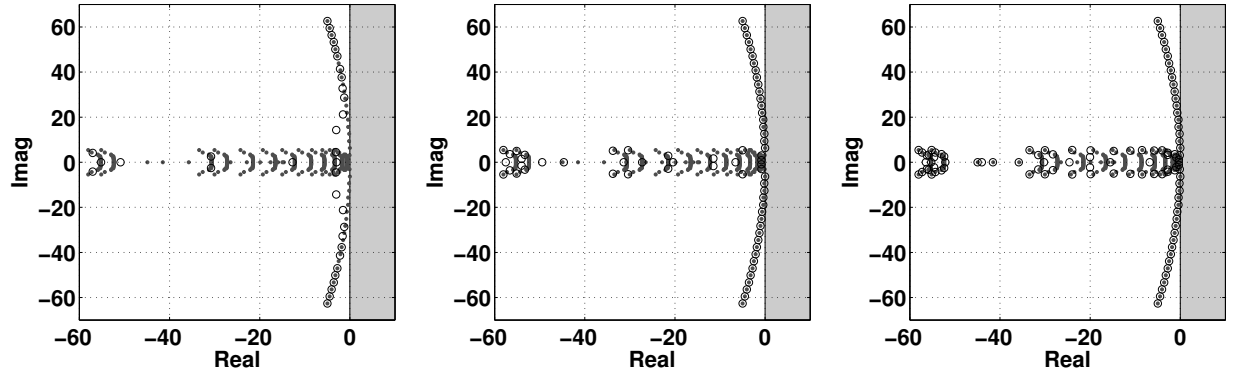
(c) Krylov dimension 50
relative residual 1.32e−29

Figure 15: Convergence of the Ritz values within the GMRES solver and the convergence of the relative residual, as the dimension of the Krylov subspace increases. The dimension of the Krylov subspace is the same as the number of timemarches. The Ritz values (open circles) are the estimated eigenvalues based on the current Krylov subspace. The eigenvalues of the Jacobian are shown as grey dots. The Jacobian matrix used is for the Poincaré shooting formulation, for a converged limit cycle.

(the number of timemarches) increases. The Ritz values are the estimated eigenvalues based on the current Krylov subspace. The relative residual of the solution of $J\underline{\Delta x} = -\underline{r}$ is defined as $\frac{||\underline{r}+J\underline{\Delta x}_k||}{||\underline{r}||}$.

As the dimension of the Krylov subspace increases, the Ritz values converge preferentially to the eigenvalues that are furthest from $(+1, 0)$. These correspond to the bulk fluid motions that are not quickly dissipated in time. This is because Krylov subspaces are formed from a power series, $\underline{v}$, $J\underline{v}$, $J^2\underline{v}, \ldots, J^n\underline{v}$, and therefore the Ritz values converge fastest to the extremal eigenvalues.

The relative residuals decrease rapidly as the dimension of the Krylov subspace increases, even when the Ritz values have not converged to the eigenvalues of the Jacobian. This is because the the GMRES algorithm has already minimised the residual in the space spanned by the eigenvectors, but has not yet converged to the eigenvectors themselves.



(a) Krylov dimension 50
relative residual 0.268

(b) Krylov dimension 100
relative residual 0.122

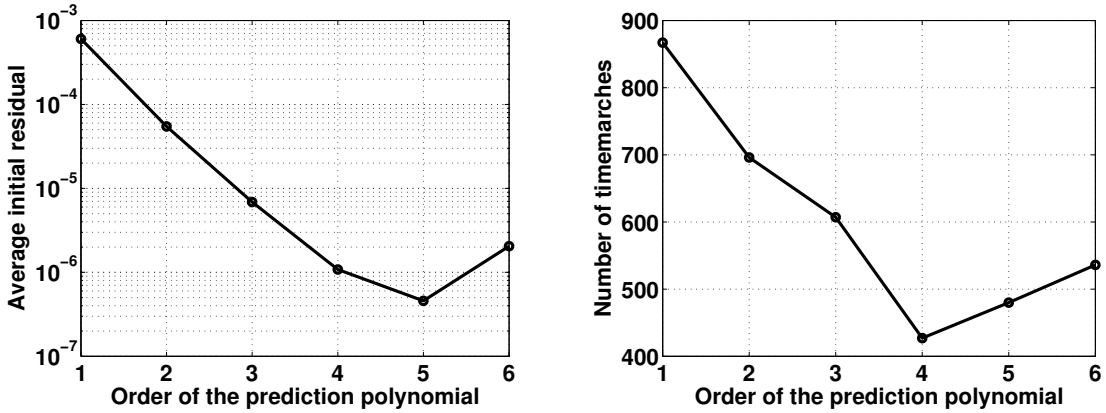
(c) Krylov dimension 150
relative residual 0.0725

Figure 16: Convergence of the GMRES solver for a fixed point (as in Figure 15, which was for a limit cycle). The dimension of the Krylov subspace is the same as the number of gradient evaluations required ($\dot{\underline{x}} = F(\underline{x}, \lambda)$).

Limit cycles can be found easily with GMRES, whereas fixed points often cannot: finding fixed points often requires preconditioning or restarted GMRES. This can be explained by comparing the eigenvalues of the Jacobian matrices. As in Figure 15, Figure 16 shows the convergence of the GMRES solver, but for a fixed point, not a limit cycle. In continuous time formulations, which are used for fixed points, dissipated features have eigenvalues with large negative real parts, with a large spread. The Ritz values converge fastest to the extremal eigenvalues, which for a fixed point correspond to both heavily dissipated and lightly dissipated features. There are a lot more influential eigenvalues than there are in a shooting formulation, and there is no useful subset of eigenvalues that GMRES can prioritise. Therefore, for a fixed point, GMRES requires a much larger Krylov subspace to converge to the same relative residual.

The shooting formulation for limit cycles is therefore particularly suited for solution with GMRES, whereas other continuation formulations may not be.

7.4.4. Effect of higher order prediction



(a) (b)

Figure 17: The effect of the order of the prediction polynomial on: (a) the average initial residual of predicted states, and (b) the total number of timemarches required to find twenty limit cycles.

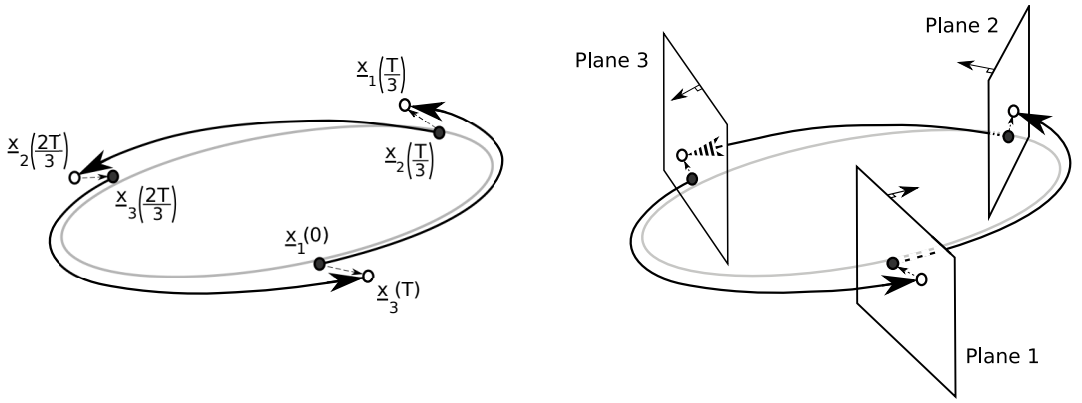
The aim of this section is to demonstrate that higher order prediction methods give lower initial residuals for fixed steplength routines. This is particularly important for finding limit cycles of large systems, because the prediction requires no additional timemarching and is therefore comparatively cheap.

In Figure 17, a section of a supercritical bifurcation diagram ($Pe = 35$, $0.686 < \beta_T < 0.719$) is used as a test case. Twenty limit cycles are found, using a fixed arclength step routine and higher order prediction methods (section 4.3). Figure 17a shows the average initial residual as a function of the order of the prediction polynomial. As the order increases, the average initial residual decreases by a factor of ten per polynomial order, until some instability begins at the fifth order. Figure 17b shows the total number of timemarches against the order of the prediction polynomial. The fourth order polynomial requires half the number of timemarches as a first order polynomial.

The diffusion flame test model has weak nonlinearities, and therefore the solution curve is smooth and the prediction extends well to higher orders. For models with stronger local nonlinearities, instability may occur at lower orders. A robust prediction methodology should detect when the initial residuals are too large, and adjust prediction orders and steplengths accordingly.

8. Multiple shooting

For systems where the timemarching process is very computationally expensive, the real-time required to find limit cycles can be reduced by parallelisation. Parallelisation can be applied in two different ways



(a) Multiple standard shooting with 3 sections. Each section is $T/3$ long, where T is the current period guess.

(b) Multiple Poincaré shooting with 3 sections. The plane definition must be communicated before timemarching is begun. The period is found as a byproduct by summing the times required to timemarch each section.

Figure 18: Multiple shooting methods, in which the limit cycle is split into n sections that are timemarched separately. There are n starting states (black dots), which are iterated to minimise the combined norms of the n residual vectors (dashed arrows).

(which can coexist): first, by splitting the state vector onto different processors and using single shooting methods; second, by splitting the trajectory onto different processors and using multiple shooting methods. The effect of the first parallelisation approach is simple, because the GMRES solver sees the timemarching process as a black box. If parallelising the state vector makes the timemarching process m times faster, then a limit cycle is found m times faster. The effect of the second parallelisation approach is more complicated, however, and will therefore be investigated in this section.

Multiple shooting methods divide the limit cycle into n sections, which are timemarched separately from the n different starting states [32]. In a similar manner to the single shooting methods, n states are found by iteration, where the sum of the residual vector magnitudes is below a predefined tolerance (Figure 18).

The advantage of multiple shooting is that each matrix-vector product, which was previously evaluated by timemarching one period, T, is now evaluated by timemarching n sections by T/n. If separate processors are used for each section, then each matrix-vector product evaluation is therefore n times faster.

The disadvantage of multiple shooting, however, is that the state vector is now nN variables long, and a GMRES solver will require roughly n times more matrix-vector products to converge to the solution [32]. This arises because the eigenspace corresponding to the bulk fluid motions is now n times larger, because the eigenvalues of the monodromy matrix are the n^{th} roots of the Floquet multipliers, as shown in Figure 19.

Without preconditioning, the advantage and the disadvantage of multiple shooting almost cancel each other out [32], although there is some additional speedup for highly unstable limit cycles, because the multiple shooting matrices are better conditioned. This is because as n increases the Floquet multipliers (both unstable and stable) tend to the unit circle when the n^{th} root is taken.

Invariant subspace preconditioning (section 4.1) can be used to offset the disadvantage of increasing the state vector size, by reducing the number of matrix-vector products required per GMRES solve. As n increases, so does the dimension of the invariant subspace that is required to maintain a set speed increase [32], but the overall process, including preconditioner formation, can approach a linear speed increase with n [32].

Figure 20 shows how the speed of continuation increases with a multiple shooting formulation, for varying degrees of preconditioning. A preconditioner is formed and then the multiple Poincaré shooting formulation is used to find 15 limit cycles with a fixed parameter step. The speed increase factor is defined as the

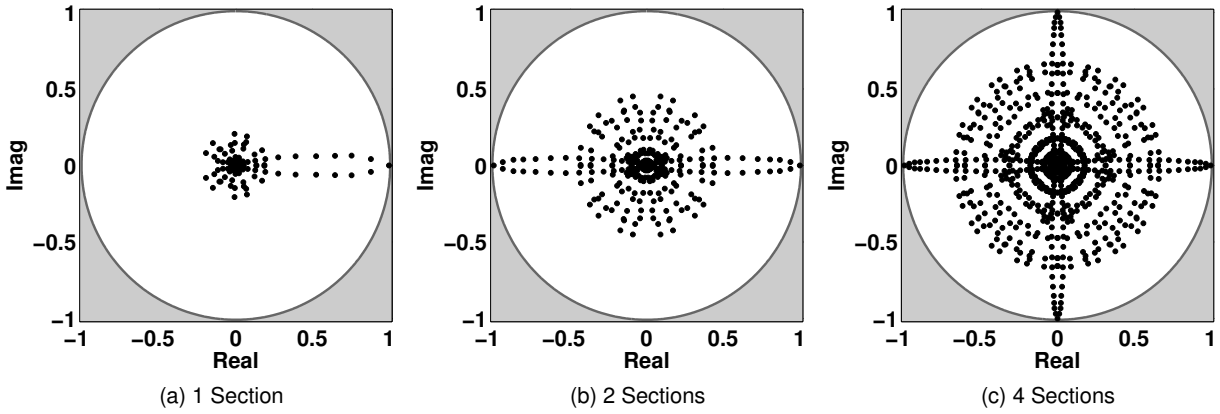


Figure 19: Eigenvalues of the monodromy matrix for the multiple Poincaré shooting formulation, for 1, 2 and 4 sections. For a multiple shooting system with n sections, the eigenvalues are the n^{th} roots of the Floquet multipliers.

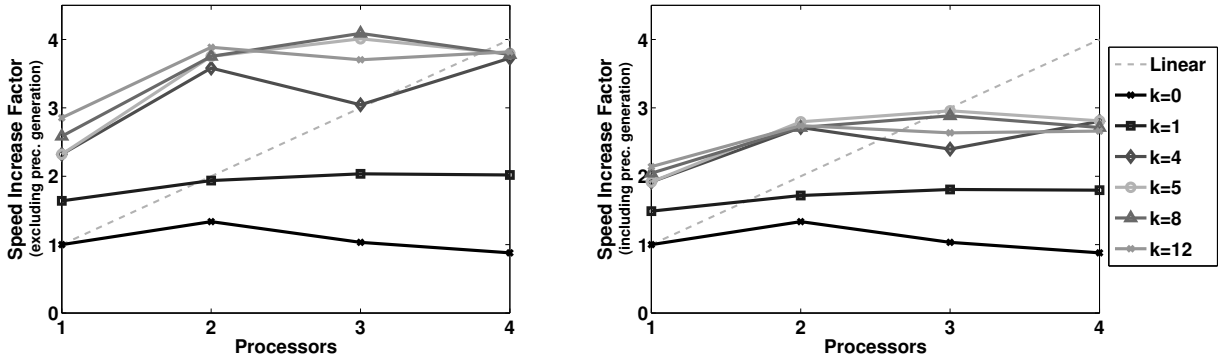


Figure 20: Speed increase for the multiple Poincaré shooting formulation against number of processors, measured over 15 limit cycles with fixed parameter continuation. The speed increase factor is normalised by the speed of the continuation with one processor and no preconditioning. The lines show the effect of preconditioning with $(k \times Processors)$ converged eigenvalues. Speed increase factors are shown when the time for preconditioner formation is excluded (left) and included (right).

real time to find the limit cycles, normalised by the equivalent real time when using one processor and no preconditioned eigenvalues, $SIF = (Real\,time)_{1\,Proc,\,0\,eigs} \,/\, (Real\,time)_{n\,Procs,\,(k\times n)\,eigs}$. The subspace chosen for preconditioning corresponds to the $(k \times n)$ eigenvalues furthest from (+1,0), and the eigenvalue tolerance is 10^{-3}.

Without preconditioning ($k = 0$), there is no speed increase with multiple processors, for the reasons described in the previous paragraphs. With preconditioning, there is a significant speed increase. When $k = 1$, the preconditioner gives a speed increase factor of $1.5 - 2.0$, even though the preconditioner is fast to compute (hence the $k = 1$ lines are almost the same in both halves of Figure 20). As in Figure 19, there is a Floquet multiplier close to (+1,0), which means that the Jacobian matrix has an eigenvalue close to the origin. When $k = 1$ the eigenvalue close to the origin is moved, significantly reducing the condition number of the Jacobian matrix and increasing the rate of convergence (section 7.4.2). When k increases, the speed increase factor also increases, but for this test case there is little difference between the speed for two and four processors. The preconditioners also take longer to form, resulting in greater differences between Figure

20(a) and (b) at high k and high n.

Multiple shooting formulations can therefore reduce the time required to form a bifurcation diagram when used with preconditioning. The overall speed increase depends on the time taken to form the preconditioner and on the number of limit cycles that it can be reused for. When generating most bifurcation diagrams, the stability of the limit cycles is of great importance, and it is therefore likely that the Floquet multipliers will be calculated every few cycles. In this case the preconditioner is generated for negligible extra cost.

9. Conclusions

This paper shows that matrix-free continuation methods can calculate limit cycles efficiently for thermoacoustic systems with $\mathcal{O}(10^3)$ variables. Continuation methods track the limit cycles and bifurcations of the system as parameters vary, in order to find the stability limits of the system over a wide parameter range. The methods operate solely in the time domain, which does not require the single-frequency in single-frequency out assumption used in the FDF method.

Thermoacoustic and fluid systems are dissipative, which makes them particularly suitable for a matrix-free iterative method with GMRES. The iterative method converges quickly to limit cycles by implicitly using a 'reduced order model' property. In other words, GMRES preferentially uses the influential bulk motions of the system, whilst ignoring features that are quickly dissipated in time. When combined with preconditioning and higher order prediction techniques, the iterative method generates bifurcation diagrams with modest computational time. For larger systems, or where timemarching is expensive, multiple shooting may be used to speed up the process of taking matrix-vector products.

A software package is written in C++ to perform matrix-free continuation of generic time dependent systems. A thermoacoustic model of a ducted 2D diffusion flame is used as a test case, with 475 variables. The continuation method converges quickly to limit cycles. The finite difference and first variational matrix-vector products give almost identical convergence for this test case, but this may not be true for models with stronger nonlinearities.

The continuation methods are used to generate a surface of limit cycles for the ducted diffusion flame, as two system parameters vary. Both subcritical and supercritical Hopf bifurcations are found. The mode shapes of the limit cycles are given directly by the continuation method. The mode shapes of any instability to the limit cycles are given by the Floquet multipliers. Examination of the mode shapes gives physical insight into the nature of the coupled flame-acoustic interaction.

When compared to the FDF method, the continuation method is more efficient in finding limit cycles when studying the effect of the flame operating condition. This is because the FDF is valid for all acoustic operating conditions, but only one specific flame operating condition. The FDF must therefore be recalculated at each new flame operating condition. The continuation method, however, is equally fast for changes in acoustic or flame operating conditions. It can therefore be used to study the sensitivity of stability limits to general changes in acoustic or flame operating conditions.

This paper provides a new, quick, and versatile method for finding limit cycles of reasonably sophisticated models of thermoacoustic systems. Furthermore, with increased computing power, it readily scales up to larger models.

References

[1] S. Ducruix, D. Durox, S. Candel, Theoretical and experimental determinations of the transfer function of a laminar premixed flame, Proceedings of the Combustion Institute 28 (1) (2000) 765–773. doi:10.1016/S0082-0784(00)80279-9.
URL http://linkinghub.elsevier.com/retrieve/pii/S0082078400802799

[2] A. P. Dowling, A kinematic model of a ducted flame, Journal of Fluid Mechanics 394 (September 2000) (1999) 51–72. doi:10.1017/S0022112099005686.
URL http://www.journals.cambridge.org/abstract_S0022112099005686

[3] D. Durox, T. Schuller, N. Noiray, S. Candel, Experimental analysis of nonlinear flame transfer functions for different flame geometries, Proceedings of the Combustion Institute 32 (1) (2009) 1391–1398. doi:10.1016/j.proci.2008.06.204.
URL http://linkinghub.elsevier.com/retrieve/pii/S1540748908001818

[4] C. O. Paschereit, B. Schuermans, W. Polifke, O. Mattson, Measurement of Transfer Matrices and Source Terms of Premixed Flames, Journal of Engineering for Gas Turbines and Power 124 (2) (2002) 239. doi:10.1115/1.1383255.
URL http://link.aip.org/link/JETPEZ/v124/i2/p239/s1&Agg=doi
[5] T. Yi, D. A. Santavicca, Flame Transfer Functions for Liquid-Fueled Swirl-Stabilized Turbulent Lean Direct Fuel Injection Combustion, Journal of Engineering for Gas Turbines and Power 132 (2) (2010) 021506. doi:10.1115/1.3157101.
URL http://link.aip.org/link/JETPEZ/v132/i2/p021506/s1&Agg=doi
[6] K. Kim, J. Lee, B. Quay, D. Santavicca, Spatially distributed flame transfer functions for predicting combustion dynamics in lean premixed gas turbine combustors, Combustion and Flame 157 (9) (2010) 1718–1730. doi:10.1016/j.combustflame.2010.04.016.
URL http://linkinghub.elsevier.com/retrieve/pii/S0010218010001203
[7] C. C. Jahnke, F. E. C. Culick, Applicaiton of Dynamical Systems Theory to Nonlinear Combustion Instabilities, Journal of Propulsion and Power 10 (4) (1994) 508–517.
[8] M. P. Juniper, Triggering in the horizontal Rijke tube: non-normality, transient growth and bypass transition, Journal of Fluid Mechanics 667 (2010) 272–308. doi:10.1017/S0022112010004453.
URL http://www.journals.cambridge.org/abstract_S0022112010004453
[9] P. Subramanian, S. Mariappan, R. I. Sujith, P. Wahi, Bifurcation analysis of thermoacoustic instability in a horizontal Rijke tube, International Journal of Spray and Combustion Dynamics 2 (4) (2010) 325–355. doi:10.1260/1756-8277.2.4.325.
[10] L. Kabiraj, R. I. Sujith, P. Wahi, Bifurcations of Self-Excited Ducted Laminar Premixed Flames, Journal of Engineering for Gas Turbines and Power 134 (3) (2012) 031502. doi:10.1115/1.4004402.
URL http://link.aip.org/link/JETPEZ/v134/i3/p031502/s1&Agg=doi
[11] L. Kabiraj, A. Saurabh, P. Wahi, R. I. Sujith, Route to chaos for combustion instability in ducted laminar premixed flames, Chaos (Woodbury, N.Y.) 22 (2) (2012) 023129. doi:10.1063/1.4718725.
URL http://www.ncbi.nlm.nih.gov/pubmed/22757536
[12] J. D. Sterling, Nonlinear Analysis and Modelling of Combustion Instabilities in a Laboratory Combustor, Combustion Science and Technology 89 (1) (1993) 167–179. doi:10.1080/00102209308924107.
[13] T. Lieuwen, Experimental Investigation of Limit-Cycle Oscillations in an Unstable Gas Turbine Combustor, Journal of Propulsion and Power 18 (1) (2002) 61–67.
[14] J. Wicker, W. Greene, S. Kim, Triggering of longitudinal combustion instabilities in rocket motors: Nonlinear combustion response, Journal of Propulsion and Power 12 (6) (1996) 1148–1158.
[15] N. Ananthkrishnan, S. Deo, F. E. C. Culick, Reduced-order modeling and dynamics of nonlinear acoustic waves in a combustion chamber, Combustion Science and Technology 177 (2005) 221–247.
[16] I. C. Waugh, M. P. Juniper, Triggering in a thermoacoustic system with stochastic noise, Journal of Spray and Combustion Dynamics 3 (3).
URL http://multi-science.metapress.com/index/38552K7155JQ2241.pdf
[17] V. S. Burnley, F. E. C. Culick, Influence of Random Excitations on Acoustic Instabilities in Combustion Chambers, AIAA Journal 38 (8).
[18] A. L. Birbaud, D. Durox, S. Candel, Upstream flow dynamics of a laminar premixed conical flame submitted to acoustic modulations, Combustion and Flame 146 (2006) 541–552. doi:10.1016/j.combustflame.2006.05.001.
[19] N. Noiray, D. Durox, T. Schuller, S. Candel, A unified framework for nonlinear combustion instability analysis based on the flame describing function, Journal of Fluid Mechanics 615 (2008) 139–167. doi:10.1017/S0022112008003613.
[20] K. Cliffe, A. Spence, S. Tavener, The numerical analysis of bifurcation problems with application to fluid mechanics, Acta Numerica 9 (00) (2000) 39–131.
URL http://journals.cambridge.org/production/action/cjoGetFulltext?fulltextid=68668
[21] E. Doedel, A. Champneys, T. Fairgrieve, Y. A. Kuznetsov, B. Sandstede, X. Wang, AUTO 2000: Continuation and bifurcation software for ordinary differential equations (with HomCont), Concordia University (2002) 1997–2000.
URL http://www.decf.berkeley.edu/help/apps/auto.pdf
[22] W. Govaerts, Y. Kuznetsov, A. Dhooge, Numerical Continuation of Bifurcations of Limit Cycles in MATLAB, SIAM Journal on Scientific Computing 27 (1) (2005) 231. doi:10.1137/030600746.
URL http://link.aip.org/link/SJOCE3/v27/i1/p231/s1&Agg=doi
[23] K. Engelborghs, T. Luzyanina, D. Roose, Numerical bifurcation analysis of delay differential equations using DDE-BIFTOOL, ACM Transactions on Mathematical Software 28 (1) (2002) 1–21. doi:10.1145/513001.513002.
URL http://portal.acm.org/citation.cfm?doid=513001.513002
[24] C. Kelley, Iterative methods for linear and nonlinear equations, Society for Industrial Mathematics, 1995.
[25] K. Georg, Matrix-free numerical continuation and bifurcation, Numerical Functional Analysis and Optimization 22 (3-4) (2001) 303–320.
[26] J. Sánchez, F. Marques, J. M. Lopez, A Continuation and Bifurcation Technique for Navier-Stokes Flows, Journal of Computational Physics 180 (1) (2002) 78–98. doi:10.1006/jcph.2002.7072.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999102970725
[27] J. Sánchez, M. Net, B. García-Archilla, C. Simó, Newton-Krylov continuation of periodic orbits for Navier-Stokes flows, Journal of Computational Physics 201 (1) (2004) 13–33. doi:10.1016/j.jcp.2004.04.018.
URL http://linkinghub.elsevier.com/retrieve/pii/S0021999104001895
[28] A. G. Salinger, R. B. Lehoucq, R. P. Pawlowski, J. N. Shadid, Computational bifurcation and stability studies of the 8 : 1 thermal cavity problem, International Journal for Numerical Methods in Fluids 1073 (July) (2002) 1059–1073.
[29] A. Salinger, N. Bou-Rabee, RP, LOCA 1.0 Library of continuation algorithms: theory and implementation manual, NM, Technical Report (March).

URL http://prod.sandia.gov/techlib/access-control.cgi/2002/020396.pdf
[30] D. Viswanath, Recurrent motions within plane Couette turbulence, Journal of Fluid Mechanics 580 (2007) 339. doi:10.1017/S0022112007005459.
URL http://www.journals.cambridge.org/abstract_S0022112007005459
[31] G. Chandler, R. Kerswell, Simple invariant solutions embedded in 2D Kolmogorov turbulence, Submitted to the Journal of Fluid MechanicsarXiv:arXiv:1207.4682v1.
URL http://meeting.aps.org/Meeting/DFD12/Event/177556?mathjax=n
[32] J. Sánchez, M. Net, On the Multiple Shooting Continuation of Periodic Orbits By Newton-Krylov Methods, International Journal of Bifurcation and Chaos 20 (01) (2010) 1–19. doi:10.1142/S0218127410025399.
URL http://www.worldscinet.com/ijbc/20/2001/S0218127410025399.html
[33] U. Erdogan, J. Thies, F. Wubs, H. Dijkstra, Determining (seasonal) periodic orbits in global ocean models using continuation methods, Bifurcations in Fluid Dynamics, Barcelona, 2011.
URL congress.cimne.upc.es/bifd2011/Admin/Files/FileAbstract/a195.pdf
[34] R. Lehoucq, D. Sorensen, C. Yang, ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods.
URL http://www.caam.rice.edu/software/ARPACK/UG/ug.html
[35] G. W. Stewart, A Krylov–Schur Algorithm for Large Eigenproblems, SIAM Journal on Matrix Analysis and Applications 23 (3) (2002) 601–614. doi:10.1137/S0895479800371529.
URL http://link.aip.org/link/SJMAEL/v23/i3/p601/s1&Agg=doi
[36] D. Roose, K. Lust, A. Champneys, A. Spence, A Newton-Picard Shooting Method for Computing Periodic Solutions of Large-scale Dynamical Systems, Chaos, Solitons & Fractals 5 (10) (1995) 1913–1925. doi:10.1007/s12194-009-0066-1.
[37] Y. Saad, M. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems., SIAM J. Sci. Stat. Comput. 7 (3) (1986) 856–869.
URL http://www.stat.uchicago.edu/ lekheng/courses/324/saad-schultz.pdf
[38] C. Simó, On the analytical and numerical approximation of invariant manifolds., in: Les Méthodes Modernes de la Mécanique Céleste. Modern Methods in Celestial Mechanics, Vol. 1, 1990, pp. 285–329.
URL http://adsabs.harvard.edu/abs/1990mmmc.conf..285S
[39] C. Barbera, E. Athanassoula, On the calculation of the linear stability parameter of periodic orbits, Astronomy and Astrophysics 336 (1998) 782–785.
URL http://adsabs.harvard.edu/full/1998A%26A...336..782B
[40] M. D. Conner, P. Donescu, L. N. Virgin, On the global convergence characteristics of numerically evaluated Jacobian matrices, Nonlinear Dynamics 10 (2) (1996) 165–174. doi:10.1007/BF00045455.
URL http://www.springerlink.com/index/10.1007/BF00045455
[41] J. Erhel, K. Burrage, B. P, Restarted GMRES preconditioned by deflation, Journal of Computational and Applied Mathematics 69 (1996) 303–318.
[42] J. Baglama, D. Calvetti, G. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM Journal on Scientific Computing 20 (1) (1999) 243–269.
[43] K. Lust, D. Roose, An Adaptive Newton–Picard Algorithm with Subspace Iteration for Computing Periodic Solutions, SIAM Journal on Scientific Computing 19 (4) (1998) 1188. doi:10.1137/S1064827594277673.
URL http://link.aip.org/link/SJOCE3/v19/i4/p1188/s1&Agg=doi
[44] R. P. Pawlowski, J. N. Shadid, J. P. Simonis, H. F. Walker, Globalization Techniques for Newton–Krylov Methods and Applications to the Fully Coupled Solution of the Navier–Stokes Equations, SIAM Review 48 (4) (2006) 700–721. doi:10.1137/S0036144504443511.
URL http://epubs.siam.org/doi/abs/10.1137/S0036144504443511
[45] M. Heroux, R. Bartlett, V. Howle, R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, Others, An overview of the Trilinos project, ACM Transactions on Mathematical Software (TOMS) 31 (3) (2005) 397–423.
URL http://dl.acm.org/citation.cfm?id=1089021
[46] C. G. Baker, U. L. Hetmaniuk, R. B. Lehoucq, H. K. Thornquist, Anasazi software for the numerical solution of large-scale eigenvalue problems, ACM Transactions on Mathematical Software 36 (3) (2009) 1–23. doi:10.1145/1527286.1527287.
URL http://portal.acm.org/citation.cfm?doid=1527286.1527287
[47] M. Tyagi, S. Chakravarthy, Unsteady combustion response of a ducted non-premixed flame and acoustic coupling, Combustion Theory and Modelling 11 (2) (2007) 205–226. doi:10.1080/13647830600733481.
URL http://www.tandfonline.com/doi/abs/10.1080/13647830600733481
[48] K. Balasubramanian, R. I. Sujith, Non-normality and nonlinearity in combustion-acoustic interaction in diffusion flames, Journal of Fluid Mechanics 594 (2008) 29–57.
[49] S. J. Illingworth, I. C. Waugh, M. P. Juniper, Finding thermoacoustic limit cycles for a ducted Burke-Schumann flame, Proceedings of the Combustion Institute 34 (1). doi:10.1016/j.proci.2012.06.017.
URL http://linkinghub.elsevier.com/retrieve/pii/S1540748912001253
[50] K. Balasubramanian, R. I. Sujith, Thermoacoustic instability in a Rijke tube: Non-normality and nonlinearity, Physics of Fluids 20 (2008) 044103.
URL http://link.aip.org/link/?PHFLE6/20/044103/1
[51] T. Poinsot, Theoretical and numerical combustion, 2nd Edition, R T Edwards, 2005.
[52] I. Waugh, M. Geuß, M. Juniper, Triggering , bypass transition and the effect of noise on a linearly stable thermoacoustic system, Proceedings of the Combustion Institute 33 (2) (2011) 2945–2952.
[53] T. Chan, Y. Saad, Iterative methods for solving bordered systems with applications to continuation methods, SIAM

Journal on Scientific and Statistical Computing 6 (2) (1985) 438–451.
URL http://epubs.siam.org/doi/pdf/10.1137/0906031

[54] H. F. Walker, An Adaptation of Krylov Subspace Methods to Path Following Problems, SIAM Journal on Scientific Computing 21 (3) (1999) 1191–1198. doi:10.1137/S1064827597315376.
URL http://epubs.siam.org/doi/abs/10.1137/S1064827597315376

[55] H. B. Keller, The Bordering Algorithm and Path Following Near Singular Points of Higher Nullity, SIAM Journal on Scientific and Statistical Computing 4 (4) (1983) 573–582. doi:10.1137/0904039.
URL http://epubs.siam.org/doi/abs/10.1137/0904039