# METHODS FOR ANALYSIS OF NONLINEAR

# THERMOACOUSTIC SYSTEMS

Iain Waugh

Emmanuel College

Department of Engineering

University of Cambridge

A thesis submitted for the degree of

*Doctor of Philosophy*

March 2013

# Declaration

The work presented in this dissertation was conducted in the Department of Engineering between October 2009 and March 2013. This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. No part of this dissertation has already been, or is being concurrently submitted for any other degree, diploma or qualification.

This dissertation contains approximately 55000 words and 86 figures.

.........................................................

Iain Waugh

18 February 2013

*To my wife, who knows more about thermoacoustics*

*than she ever wanted to.*

# Acknowledgements

# Methods for analysis of nonlinear thermoacoustic systems

## Iain Waugh

**Summary**

This thesis examines the nonlinear behaviour of thermoacoustic systems by using approaches from the field of nonlinear dynamics. The behaviour of a nonlinear system is determined by two things, which are the focus of this thesis: first, by the mechanism that the system transitions from one attractor to another, and second, by the type and form of the attractors in phase space.

In the first part of the thesis, a triggering mechanism is presented for a Rijke tube model, whereby the system transitions from a stable fixed point to a stable limit cycle, via an unstable limit cycle. Using this mechanism, low levels of stochastic noise result in triggering much before the linear stability limit. Stochastic stability maps are introduced to visualise the practical stability of a thermoacoustic system. These theoretical results match well with those from experiments.

In the second part of the thesis, two time domain methods are presented for finding limit cycles in large thermoacoustic systems: matrix-free continuation methods and gradient methods.

Most continuation methods are too computationally expensive for finding limit cycles in large thermoacoustic systems. For dissipative systems, matrix-free continuation methods are shown to converge quickly to limit cycles by preferentially using the influential bulk motions of the system, whilst ignoring the features that are quickly dissipated in time. These matrix-free methods are demonstrated on a model of a ducted 2D diffusion flame and a model of a ducted axisymmetric premixed flame (with G-equation solver). Rich nonlinear behaviour is found: fixed points, sub/supercritical Hopf bifurcations, limit cycles, period-2 limit cycles, fold bifurcations, period-doubling bifurcations and Neimark-Sacker bifurcations. 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.

Gradient methods define a scalar cost function that describes the proximity of a state to a limit cycle. The gradient of the cost function is calculated using adjoint equations and then used to iteratively converge to a limit cycle (or fixed point). The gradient method is demonstrated on a model of a horizontal Rijke tube.

This thesis describes novel nonlinear analysis techniques that can be applied to coupled systems with both advanced acoustic models and advanced flame models. The techniques can characterise the rich nonlinear behaviour of thermoacoustic models with a level of detail that was not previously possible.

# Abstract

This thesis examines the nonlinear behaviour of thermoacoustic systems by using approaches from the field of nonlinear dynamics. The underlying behaviour of a nonlinear system is determined by two things: first, by the type and form of the attractors in phase space, and second, by the mechanism that the system transitions from one attractor to another. For a thermoacoustic system, both of these things must be understood in order to define a safe operating region in parameter space, where no high-amplitude oscillations exist.

Triggering in thermoacoustics is examined in a simple model of a horizontal Rijke tube. A triggering mechanism is presented whereby the system transitions from a stable fixed point to a stable limit cycle, via an unstable limit cycle. The practical stability of the Rijke tube was investigated when the system is forced by stochastic noise. Low levels of noise result in triggering much before the linear stability limit. Stochastic stability maps are introduced to visualise the practical stability of a thermoacoustic system. The triggering mechanism and stochastic dependence of the Rijke tube match extremely well with results from an experimental combustor.

The most common attractors in thermoacoustic systems are fixed points and limit cycles. In order to define the nonlinear behaviour of a thermoacoustic system, it is therefore important to find the regions of parameter space where limit cycles exist. Two methods of finding limit cycles in large thermoacoustic sytems are presented: matrix-free continuation methods and gradient methods.

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 safe operating region is 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.

The matrix-free methods are also demonstrated on a model of a ducted axisymmetric premixed flame, using a kinematic G-equation solver. The methods find limit cycles, period-2 limit cycles, fold bifurcations, period-doubling bifurcations and Neimark-Sacker bifurcations as a function of two parameters: the location of the flame in the duct, and the aspect ratio of the steady flame. The model is seen to display rich nonlinear behaviour and regions of multistability are found.

Gradient methods can also efficiently calculate the limit cycles of large systems. A scalar cost function is defined that describes the proximity of a state to a limit cycle. The gradient of the cost function is used in an optimisation routine to iteratively converge to a limit cycle (or fixed point). The gradient of the cost function is found with a forwards-backwards process: first, the direct equations are marched forwards in time, second, the adjoint equations are marched backwards in time. The adjoint equations are derived by partially differentiating the direct governing equations. The gradient method is demonstrated on a model of a horizontal Rijke tube.

This thesis describes novel nonlinear analysis techniques that can be applied to coupled systems with both advanced acoustic models and advanced flame models. The techniques can characterise the rich nonlinear behaviour of thermoacoustic models with a level of detail that was not previously possible.

# Contents

# Chapter 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. Because the energy density in industrial combustors is so high, the amplitudes of the limit cycles can be so large that the combustor is physically damaged or even destroyed. To avoid damage to the combustors these oscillations must therefore either be avoided or they must be controlled. Although there has been some progress in the field of active control of combustion instabilities, by acoustic feedback [7], jet injection [8] and fuel injection [9], the technology is still far from being applied in systems such as jet engines, largely because there are no actuators suitable for operation in such harsh environments. The most common approach is therefore to avoid thermoacoustic oscillations by defining a safe operating region, where oscillations either do not occur or they remain at sufficiently low amplitude.

Defining a safe operating region, however, is not a trivial problem. Industrial combustion systems often involve complex geometries, turbulent combustion and multi-phase flows. They are therefore strongly non-linear systems that are inherently noisy. Because the systems are strongly nonlinear, the system's behaviour is often a strong function of the parameters that define the operating condition. Industrial systems are generally tested experimentally to find the safe operating region, but this process is extremely expensive. There is therefore good motivation to develop techniques that can estimate the safe operating region of a combustor computationally during the design phase.

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 broadly into those that use linear analysis and those that use nonlinear analysis.

## 1. INTRODUCTION

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 [10, 11] and experimental [12, 13, 14, 15] 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 techniques adapted 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 [16, 17, 18]. 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. Unlike the FTF, this approach cannot be used experimentally.

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, quasiperiodicity, chaos or another fixed point [19, 20, 21, 22, 23]. Linear stability analysis is often used in thermoacoustics to estimate the frequencies of large-amplitude limit cycles [24, 25].

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 [16, 17, 26]. 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 1.1(a) and (b). Both types of bifurcation have been seen in experimental combustors [20, 27].

In the frequency domain, the Flame Describing Function (FDF) is commonly used to estimate limit cycle amplitudes and to characterise the behaviour of flames in computational [28] and experimental [29, 30] studies. 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 main limitation with the FDF is the assumption that the acoustic velocity and pressure signals are sinusoidal. The FDF is nonlinear, but this assumption means that it still has some remnants of a linear framework: it relies on a single-frequency in, single-frequency out relationship. This is because
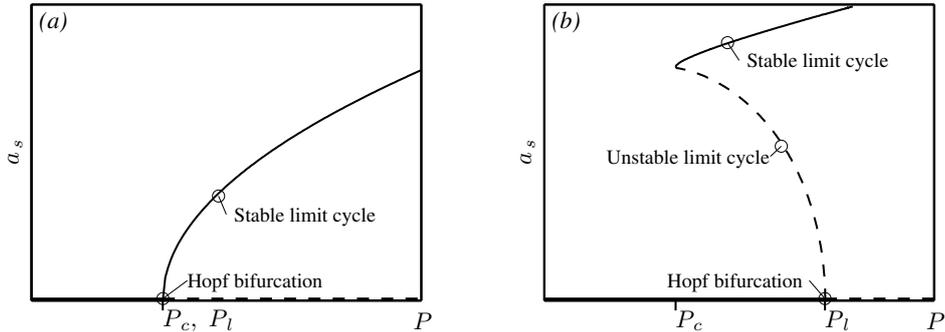
Figure 1.1: Schematic supercritical (a) and subcritical (b) Hopf bifurcations, in terms of the limit cycle amplitude, $a_s$ and a system parameter $P$. The critical parameter, $P_c$, is the parameter value where limit cycles first occur. The linear stability limit, $P_l$, is the same as the Hopf bifurcation 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.

the flame response is discarded at the harmonics of the forcing frequency, because, if $p$ is sinusoidal, over one cycle $\int pq \, \mathrm{d}t = 0$ for the harmonics. In reality, if the heat release signal has higher harmonics then the acoustics will respond at those higher harmonics, and the acoustic velocity and pressure signals are no longer sinusoidal - the original assumption is broken. If the heat release and pressure signals both have higher harmonics (i.e. are periodic but not sinusoidal) then these harmonics contribute to energy growth over a cycle and the FDF result will be inaccurate.

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 [31]. Software packages such as AUTO [32], MATCONT [33] and DDE-BIFTOOL [4] are available for continuation analysis of generic time dependent systems.

Continuation methods have been used in the field of thermoacoustics to calculate limit cycles and sub-/supercritical bifurcations [16, 17, 18, 26, 34], 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. Conventional continuation methods 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. With recent advances in matrix-free iterative methods, which require less computational time and less memory, continuation

methods can now be applied to larger systems such as thermal convection [35].

An alternative method of finding limit cycles in the time domain is to reformulate the problem as the minimisation of a cost functional, where the value of the cost function represents how far an initial condition is from being periodic. For large systems, adjoint equations can be used to find the gradient of the cost functional using a forward-backward procedure: the direct governing equations are marched forward in time, then the adjoint equations are marched backwards in time [36]. This method of gradient evaluation is efficient for large systems because the gradient is evaluated with respect to every variable simultaneously using a single forward-backward timemarch. Nonlinear adjoint equations can require a lot of memory storage, however, because the adjoint equations require the state during the forward timemarch to be saved. This memory requirement can be reduced by using checkpointing [37]. Minimisation using adjoint derived gradients is relatively new in thermoacoustics [17] but is established in the field of hydrodynamics, where it has been used to minimise cost functions in systems such as boundary layers [38], Hiemenz flow [39] and shape optimisation [40, 41].

All of the above methods are used to generate the safe operating region of a system while it is not forced by any external perturbations. In any physical combustion system, however, there will be noise from acoustic and entropic waves [42] and noise arising from the stochastic nature of turbulence and reaction processes. Under the influence of a particular form and amplitude of noise, a stable operating condition may develop large oscillations by two separate mechanisms: first, when the noise affects the system's parameters, the stability of the stable attractor may change from stable to unstable, and second, when the system is simply excited by the noise, the system may be excited enough to reach the basin of attraction of another attractor, such as a limit cycle. This second mechanism is commonly referred to as 'triggering' in thermoacoustics.

Triggering in thermoacoustics requires at least two stable attractors (bistability), one of which is a stable fixed point, the other of which is commonly a stable limit cycle (but does not have to be). It is the process whereby the system transitions from the stable fixed point to the other stable attractor. In some linearly stable thermoacoustic systems, self-sustained oscillations can be triggered by perturbations with amplitudes similar to the background noise level [43, Ch1 §IV], or by exciting the system with a small impulse. The most common examples of impulse instigated triggering are solid and liquid rocket motors, where triggering was initiated using small explosive devices [43] - triggering is therefore termed "pulsed" instability in some early literature [34].

Linear analyses cannot find parameter regions that are bistable[1], and therefore cannot determine whether a particular operating region is susceptible to triggering. Many early linear studies found that a system developed oscillations long before the fixed point became unstable [44]. Premature instability of a linearly stable system is not unique to thermoacoustics, however. It occurs widely in hydrodynamics, and has been studied in areas such as bypass transition to turbulence. The premature instability has been largely attributed to the non-normality of the systems involved [45]. Non-normality occurs when a system's stability operator is not self-adjoint, i.e. $L^+L \neq LL^+$ where $L^+$ is the adjoint of the linear operator L. Non-normality leads to non-orthogonality of a system's eigenmodes. With non-orthogonal eigenmodes, it is possible for a perturbation to transiently grow in amplitude, even in a linear stable operating condition. The degree of this transient growth can be large, and in some cases energy growth factors of $10^6$ occur in hydrodynamic systems [45]. In Ref. [46], the examples in hydrodynamics where linear stability limit predictions worked well, such as Bérnard convection, are those where the linearised system is normal. The examples in hydrodynamics where linear stability limit predictions worked badly, such as planar Couette and Poiseuille flow, are those where the linearised system is non-normal. Non-normality in hydrodynamics has two common sources [47]: convective-type non-normality, where the direct and adjoint global modes have different advection directions; and component-type non-normality, where the direct and adjoint global modes have amplitudes in different velocity components (sometimes referred to as lift-up non-normality).

Thermoacoustic systems have been shown to be non-normal [48, 49] because the eigenmodes are non-orthogonal [50, 51]. It is likely therefore that techniques developed for non-normal analysis will be relevant for thermoacoustic systems [17, 48, 49]. Triggering to self-sustained oscillations in thermoacoustics has been shown to be analagous to bypass transition to turbulence in hydrodynamics [17]. The role of non-normality in switching between stable states is examined in this thesis.

The underlying behaviour of a nonlinear system is determined by two things: first, by the type, form and stability of the attractors in phase space, and second, by the mechanism that the system transitions from one attractor to another. The most commonly reported attractors in thermoacoustics are fixed points and limit cycles[2], which can be found by iterative methods. Once the attractors are known, the mechanism by which a thermoacoustic system transitions

---

[1]Except parameter regions that contain two fixed points.
[2]Although authors in earlier literature may not have expected richer nonlinear behaviour such as quasiperiodicity.

from one attractor to another can then be analysed by examining the trajectories in phase space when the system is forced by noise or given a variety of initial conditions.

This thesis will analyse thermoacoustic systems in this manner. This thesis describes iterative methods that efficiently find limit cycles in large dimensional systems and can find their stability and bifurcations, and this thesis also describes a mechanism by which thermoacoustic systems trigger from a stable fixed point to a stable limit cycle, via an unstable limit cycle.

## 1.1    Thesis scope

The aim and novelties of this thesis are threefold.

First, to provide efficient techniques to find the safe operating region of thermoacoustic systems, with particular emphasis on systems that require large discretisations and where timemarching is computationally expensive. The techniques presented - continuation methods and gradient based methods - are orders of magnitude faster than conventional methods of finding limit cycles when combustion parameters are varied. Furthermore, the techniques use the fully coupled system; the acoustics can have multiple modes, and the technique can account for the inclusion of non-harmonic velocity and heat release fluctuations, unlike the FDF method.

Second, to provide a technique of quantifying the stochastic dependence of the safe operating region on the noise present in the system. The technique developed has been used in experimental systems and shown to match theoretical predictions well.

Third, to demonstrate the techniques above by analysing the stability of three simple thermoacoustic systems, containing a hot wire, a diffusion flame and a premixed flame. The techniques can calculate the limit cycles and bifurcations (fold, period-doubling and Neimark-Sacker) of systems that were previously too large for such computation to be feasible. The techniques can also examine the perturbations around a limit cycle (the Floquet modes), which makes it possible to better understand the coupled flame-acoustic interaction, and to quantify the non-normality around a limit cycle by examining the pseudospectra of the monodromy matrix. The monodromy matrix and Floquet multipliers are defined section 5.2.1.

The continuation methods developed in this thesis deserve special emphasis, because they are arguably the most versatile and accurate tool available for finding and understanding the limit cycles of nonlinear thermoacoustic models. In particular, they can efficiently find the limit cycles of systems with advanced acoustic network models, such as LOTAN [52, 53], coupled with accurate flame models, such as the kinematic G-equation with curvature dependent flame speed (chapter 7). They are also able to detect qualitative changes in system behaviour,

such as bifurcations to period-$2^n$ or quasiperiodic oscillations. Furthermore, by examining the Floquet multipliers, it is then possible to show exactly which coupled flame-acoustic motion is responsible for causing the change in system behaviour. This is a significant advance of the state of the art.

## 1.2 Thesis structure

The thesis is divided into two parts. The first part of the thesis is the result of the first year's work, the second part of the thesis is the result of the latter two years' work.

The first part of the thesis concentrates on triggering in thermoacoustics, and the effect of noise on the practical stability of a thermocoustic system. The thesis begins in chapter 2 with an introduction to the theoretical model of the horizontal Rijke tube, which is used as test model in chapters 3, 4 and 8. The nonlinear behaviour of the Rijke tube is analysed with no forcing from noise. In chapter 3, a bistable operating region of the Rijke tube model is used to present a triggering mechanism, whereby an unstable attractor is used as a temporary stepping stone for small perturbations to develop into large amplitude oscillations. This triggering mechanism is demonstrated with several initial conditions and low amplitude periodic noise. In chapter 4, the concept of practical stability is introduced to define the stochastic nature of stability when a system is excited by random noise. A method of quantifying the stochastic stability of a theoretical or experimental system is presented, and theoretical predictions are compared against experimental results.

The results in chapters 3 and 4 show that the most important feature of a thermoacoustic system is its nonlinearity, rather than its non-normality. This is because it is much more important to know the stable and unstable states of the system, which is governed by the nonlinearity, than it is to know how the system transitions between these states, which is governed by both the non-normality and the nonlinearity.

The second part of the thesis therefore concentrates on efficient methods for computing the limit cycles of large thermoacoustic systems, in particular when timemarching is computationally expensive. Chapter 5 presents a matrix-free continuation method which can efficiently find limit cycles in large thermoacoustic systems. This method is demonstrated on a model of a ducted diffusion flame in chapter 6, and a model of a ducted premixed flame in chapter 7. Chapter 8 then presents an alternative method for finding limit cycles in large thermoacoustic systems, which minimises a cost function by using adjoint equations to find gradient information. This method is demonstrated on the horizontal Rijke tube model.

The thesis finishes with conclusions and suggestions for further work.

# Part I

# Triggering in thermoacoustic systems

# Chapter 2

# The Rijke tube with a hot wire

The horizontal Rijke tube with a hot wire is a simple thermoacoustic system, which is used as a test model in chapters 3, 4 and 8. Figure 2.1 shows a schematic of a horizontal Rijke tube. It consists of an open ended tube with a baseflow imposed through it by an external fan. Air is heated by an electrically powered wire gauze at a location $x_f$ from one end. This simple model contains the interaction between heat release and acoustics, without further complications added by flame dynamics. When conditions are such that an acoustic oscillation exists, a standing wave is set up in the open-ended tube.
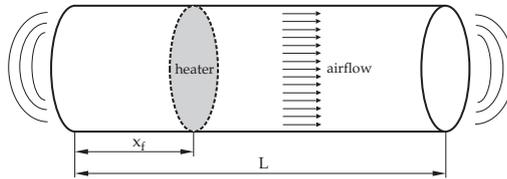


Figure 2.1: Schematic of the horizontal Rijke tube. The baseflow is imposed through the tube by an external fan. The heater is an electrically heated wire gauze.

The Rijke tube is often used in a vertical form, where no external fan is used but a baseflow arises from natural convection. This can also result in acoustic oscillations, but it it makes the model more complicated. To minimise modelling complexities a horizontal Rijke tube is used in this thesis.

## 2.1   Governing equations

The thermoacoustic system examined in this thesis is identical to the simple Rijke tube model studied by Refs. [17, 49], which contain a complete description. In summary, it is a tube of length $L_0$ in which a hot wire gauze is placed $\tilde{x}_f$ from one end. A base flow is imposed through the tube with velocity $u_0$.

## 2. THE RIJKE TUBE WITH A HOT WIRE

Standard linear acoustics are coupled with a heat release model and a damping model to form the governing equations. The relationship between velocity and heat-release at the wire mesh is based on a correlation by Heckl [54]. This is an adaptation of King's Law that includes a nonlinear saturation of heat release and a time delay. The perturbation heat release equation from Heckl's correlation is:

$$\tilde{Q} = \frac{2\tilde{L}_w(\tilde{T}_w - \tilde{T}_0)}{\sqrt{3}} \left(\pi \lambda \tilde{c}_v \tilde{\bar{\rho}} \frac{\tilde{d}_w}{2}\right)^{\frac{1}{2}} \left(\left|\frac{\tilde{u}_0}{3} + \tilde{u}'(t - \tilde{\tau})\right|^{\frac{1}{2}} - \left(\frac{\tilde{u}_0}{3}\right)^{\frac{1}{2}}\right) \tag{2.1}$$

where the physical gas properties in the tube are defined by the constant volume specific heat capacity, $c_v$, the ratio of specific heats, $\gamma$, and the thermal conductivity, $\lambda$, and where $\tau$, the time delay, is estimated in Ref. [55] to be:

$$\tilde{\tau} = 0.2 \frac{\tilde{d}_w}{\tilde{u}_0} \tag{2.2}$$

The damping in the model is assumed to come from two sources: boundary layer losses at the tube walls and sound radiated from the ends of the tube. Derivations from Ref. [56] for each of these terms are added to give a damping coefficient, which is dependent on the frequency, $\omega$, length of the tube, $\tilde{L}$, radius of the tube, $\tilde{R}$, kinematic viscosity, $\tilde{\nu}$, thermal diffusivity, $\tilde{\chi}$, speed of sound, $\tilde{c}_0$, and ratio of specific heats, $\gamma$.

$$\zeta(\omega) = \frac{\tilde{R}^2 \omega^2}{\tilde{c}_0^2} + \frac{\sqrt{2\omega}\tilde{L}}{\tilde{R}\tilde{c}_0}\left(\sqrt{\tilde{\nu}} + \sqrt{\tilde{\chi}}(\gamma - 1)\right) \tag{2.3}$$

The non-dimensional governing equations for momentum and energy are therefore:

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = 0, \tag{2.4}$$

$$\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \zeta(\omega)p - \beta\delta(x - x_f)\left(\left|\frac{1}{3} + u_f(t - \tau)\right|^{\frac{1}{2}} - \left(\frac{1}{3}\right)^{\frac{1}{2}}\right) = 0,$$

where if the subscript 0 represents mean quantities, the heat release parameter, $\beta$, is dependent on the wire length, $L_w$, wire temperature, $T_w$, wire diameter, $d_w$, and tube cross sectional area, $S$:

$$\beta = \frac{1}{p_0\sqrt{u_0}}\frac{(\gamma - 1)}{\gamma}\frac{2L_w(T_w - T_0)}{S\sqrt{3}}\left(\pi\lambda c_v \rho_0 \frac{d_w}{2}\right)^{\frac{1}{2}} \tag{2.5}$$

The system has four control parameters: $\zeta$, which models the damping; $\beta$, which encapsulates all the information about the hot wire, base flow and ambient conditions; $\tau$, which is the time delay between the velocity at the wire and the subsequent heat release and $x_f$, which is

the position of the wire. The heat release parameter, $\beta$, is equivalent to $k/\gamma M$ in Ref. [49].

For the system examined in this thesis, $\partial u/\partial x$ and $p$ are both set to zero at the ends of the tube. These boundary conditions are enforced by choosing an appropriate basis set:

$$u(x,t) = \sum_{j=1}^{N} \eta_j(t)\cos(j\pi x), \tag{2.6}$$

$$p(x,t) = -\sum_{j=1}^{N} \left(\frac{\dot{\eta}_j(t)}{j\pi}\right)\sin(j\pi x), \tag{2.7}$$

In this Galerkin discretisation, all the basis functions (often referred to as Galerkin modes) are orthogonal. It is important to point out that these Galerkin modes are not, in general, the eigenmodes of the system. They are merely the basis set into which $u$ and $p$ are decomposed. The governing equations then reduce to two ordinary differential equations for each mode, labelled $j$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta_j - j\pi\left(\frac{\dot{\eta}_j}{j\pi}\right) = 0, \tag{2.8}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right) + j\pi\eta_j + \zeta_j\left(\frac{\dot{\eta}_j}{j\pi}\right) + 2\beta\sin(j\pi x_f)\left(\left|\frac{1}{3}+u_f(t-\tau)\right|^{\frac{1}{2}} - \left(\frac{1}{3}\right)^{\frac{1}{2}}\right) = 0, \tag{2.9}$$

where:

$$u_f(t-\tau) = \sum_{k=1}^{N}\eta_k(t-\tau)\cos(k\pi x_f). \tag{2.10}$$

and if the frequency of each Galerkin mode is assumed to remain at a constant multiple of the fundamental mode, $\omega = \omega_1 j$, then:

$$\zeta_j = \frac{\tilde{R}^2\omega_1^2}{\tilde{c}_0^2}j^2 + \frac{\sqrt{2\omega_1}L}{\tilde{R}\tilde{c}_0}\left(\sqrt{\tilde{\nu}} + \sqrt{\tilde{\chi}}\,(\gamma-1)\right)j^{\frac{1}{2}}, \tag{2.11}$$

The state of the system is given by the amplitudes of the Galerkin modes that represent velocity, $\eta_j$, and those that represent pressure, $\dot{\eta}_j/j\pi$. These are given the notation $\mathbf{u} \equiv (\eta_1, \ldots, \eta_N)^T$ and $\mathbf{p} \equiv (\dot{\eta}_1/\pi, \ldots, \dot{\eta}_N/N\pi)^T$. The state vector of the discretised system is the column vector $\mathbf{x} \equiv (\mathbf{u}; \mathbf{p})$. The most convenient measure of the size of the perturbations is the acoustic energy per unit volume:

$$E = \frac{1}{2}u^2 + \frac{1}{2}p^2 = \frac{1}{2}\mathbf{x}^H\mathbf{x} = \frac{1}{2}||\mathbf{x}||^2, \tag{2.12}$$

where $||\cdot||$ represents the 2-norm.

Equations (2.8 - 2.9) can be linearised into the form $\mathrm{d}\mathbf{x}/\mathrm{d}t = \mathbf{L}\mathbf{x}$, from which it is found that $\mathbf{L}$ is non-normal [49]. (A non-normal operator satisfies $\mathbf{L}^H\mathbf{L} \neq \mathbf{L}\mathbf{L}^H$, where $H$ denotes the

Hermitian transpose.) Non-normality gives rise to linear transient growth, which is discussed in chapters 3 and 4.

## 2.2    Nonlinear analysis of the system

Combustion systems have been shown to display limit cycle, period-$2^n$, chaotic and quasi-periodic behaviour [19, 20, 21], which are well studied phenomena in the field of nonlinear dynamics. Using analysis techniques developed in nonlinear dynamics the underlying behaviour of the Rijke tube system can be defined. In particular, continuation analysis can be used to iteratively find the fixed points (steady states) and limit cycles of the system, as a function of system parameters. An open source continuation tool for delay-differential equations, DDE-BIFTOOL [4], is used in this section to calculate bifurcation diagrams for the Rijke tube model.

### 2.2.1    Bifurcation diagrams
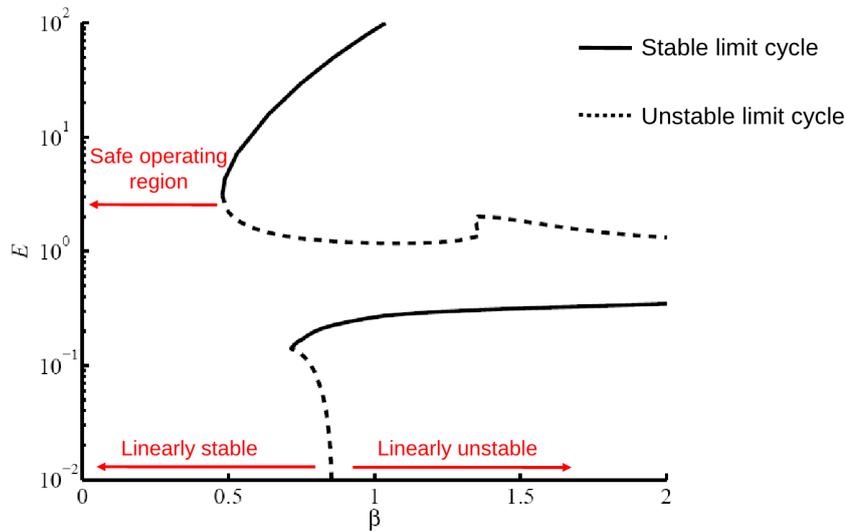


Figure 2.2: Limit cycle amplitudes of the Rijke tube model in terms of the minimum acoustic energy of the limit cycle (logarithmic), as a function of the non-dimensional heat release parameter ($\beta$). The linearly stable region of the fixed point, and the safe operating region are annotated.

The bifurcation diagram in Figure 2.2 shows the minimum acoustic energy of the limit cycles of the Rijke tube model as the heat release parameter $\beta$ is varied. The parameters that are held fixed are $\zeta_j = 0.05j^2 + 0.01j^{1/2}$, $x_f = 0.3$ and $\tau = 0.02$, and 20 Galerkin modes are used. The Floquet multipliers of the limit cycles are calculated to determine whether they are

stable (solid lines) or unstable (dashed lines). Figure 2.2 is similar to the bifurcation diagrams in Ref. [43, Ch1 Fig. 1.17] and Ref. [30].

Even though the system is linearly stable for $0 \geq \beta \geq 0.866$, it can support a high amplitude self-sustained limit cycle for $\beta \geq 0.478$ and a separate low amplitude limit cycle for $\beta \geq 0.722$. The Hopf bifurcation at $\beta = 0.866$ is sub-critical, which is typical of combustion systems that exhibit triggering or "pulsed" instabilities [26]. The importance of this subcritical bifurcation for triggering is covered in chapters 3 and 4.

There are two sets of unstable and stable limit cycles in Figure 2.2. Figure 2.2 has a logarithmic energy scale because there is a large difference in amplitude between these two sets of oscillations. The higher amplitude set of limit cycles occurs due to the nonlinearity of the modulus term in the heat release equation. These limit cycles are used in the next chapter for demonstrative purposes because they act in qualitatively the same way as the lower amplitude limit cycles, but it should be noted that they are probably non-physical. It is important to note that the lines in Figure 2.2 are projections of the $N$-dimensional trajectories of the limit cycles, and not fixed energy boundaries of the system.
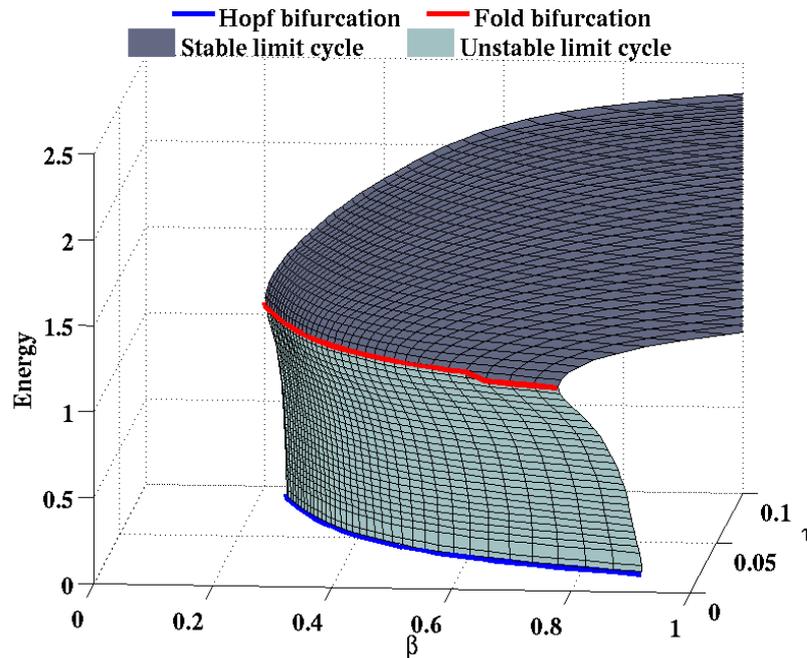


Figure 2.3: Stability surface for varying non-dimensional wire temperature ($\beta$) and time delay ($\tau$), obtained with continuation methods

When multiple parameters are varied simultaneously, stability surfaces can be obtained with continuation methods. Figure 2.3 shows the time delay dependence of the Hopf bifurcation (blue) and the fold bifurcation (red). The Hopf bifurcation is the linear stability limit of the

system; the fold bifurcation is where the limit cycles transition from stable to unstable. For the Rijke tube model, the fold bifurcation also defines the boundary of the safe operating region, where no limit cycles exist. An increase in time delay moves the Hopf bifurcation to a lower wire temperature, and also results in a smaller bistable region. The energy of oscillations also increases with the time delay. Further examples of bifurcation surfaces for the Rijke tube are detailed in Ref. [18].
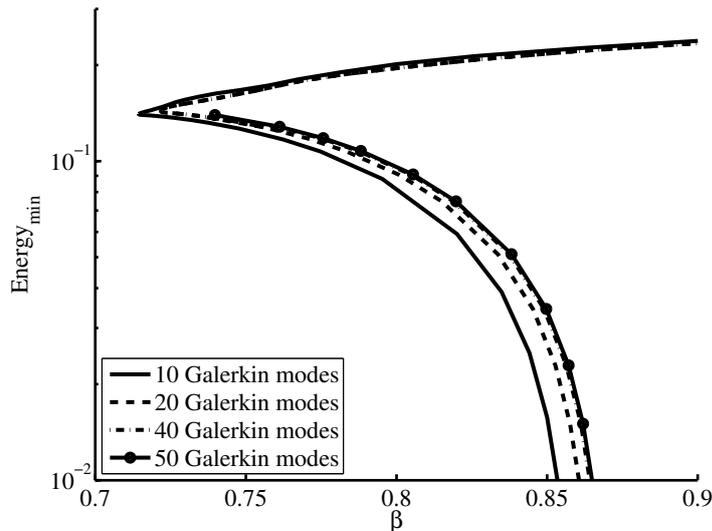


Figure 2.4: Robustness of the subcritical bifurcation to changes in the number of discretisation modes.

The results in Figure 2.2 are presented for a 20 Galerkin mode discretisation, but have been calculated for 10, 20, 40 and 50 modes and are shown in Figure 2.4. The differences between the 20 and 50 mode results are negligible, so as a compromise between accuracy and speed 20 modes are used for the rest of this thesis.

## 2.3 Limitations to the model

It is important to consider some of the limitations to the Rijke tube model and what effect they may have on accuracy. First, the model contains only linear acoustics and therefore there will be some inaccuracy when limit cycles have large amplitudes. Second, the model also takes no account of the temperature variation of the flow along the pipe. A temperature jump across the wire would introduce several changes: the acoustic modes would not be sinusoidal, the mode frequencies would no longer be exactly harmonic, gas properties would vary, there would be less heat release when there is reverse flow over the hot wire and the non-normality of the system may decrease [51]. Third, the damping model is simplistic in that only discrete frequencies are

damped and that reflection coefficients are not used. Damping is not the key element of study for this model, however, and the simplistic damping model should not alter the fundamental nature of the triggering mechanisms investigated in chapters 3 and 4. Fourth, the heat release model is simplistic in that it is based on an empirical fit of experimental data that mimics King's Law. The empirical fit is designed to match the fraction of mean flow where nonlinear saturation begins, but is not likely to give accurate heat release values when the velocity is significantly less than the mean flow [57]. Fifth, the time delay is constant. In reality, the time delay is both frequency and amplitude dependent, and not even constant across the duct, because the mean flow will have a parabolic profile.

Lastly, the Galerkin discretisation method does not allow travelling waves to enter and leave the tube. It is likely that the baseflow entering the tube will be noisy, and these perturbations will travel downstream. Although it would be more accurate to include travelling waves when noise is considered in chapter 4, the noise induced triggering mechanisms identified in chapter 4 will be unchanged.

# Chapter 3

# Analogy between triggering in thermoacoustics and bypass transition in hydrodynamics

In some linearly stable thermoacoustic systems, self-sustained oscillations can be triggered by perturbations with amplitudes similar to the background noise level [43, Ch1 §IV]. This is known as triggering. In some linearly-stable hydrodynamic systems, turbulence can be triggered by perturbations with a similarly small amplitude [45]. This is known as bypass transition to turbulence. While it is clear that non-linear effects are important in thermoacoustics, it has only recently been shown that non-normal effects are also influential [48, 49]. An analogy between triggering in thermoacoustics and bypass transition in hydrodynamics has been suggested by Lieuwen [43, Ch1 §IV] and recently demonstrated by Juniper [17].

This chapter investigates this analogy further by examining the mechanism by which small perturbations and low amplitudes of periodic noise can cause the system to trigger from a stable fixed point to a stable limit cycle. The content of this chapter was originally published in Ref. [1], with co-authors Matthias Geuß and Matthew Juniper.

## 3.1   Bypass transition mechanism

In hydrodynamics, bypass transition can be divided into five stages [45]. The first stage is initiation of small perturbations to the flow. The second stage is linear amplification of these perturbations due to non-normal growth. The third stage is non-linear saturation towards a new unstable quasi-steady periodic state. The fourth stage is growth of secondary instabilities on top of this periodic base flow. The fifth stage is breakdown to turbulence, where non-

linearities and/or symmetry-breaking instabilities excite an increasing number of scales in the flow.

The second, third and fourth stages can also be considered in terms of dynamical systems [58, 59, 60]. A boundary in state space is identified between trajectories that decay to a laminar solution and trajectories that evolve to a turbulent solution. This is known as the 'edge of chaos'. Trajectories from certain regions of state space are attracted towards this boundary. Those that are attracted from lower energy states exhibit the non-normal transient growth identified in stage 2. The boundary itself corresponds to the unstable periodic state identified in stage 3. In hydrodynamics, this boundary contains several heteroclinic saddle points and at least one local relative attractor, each corresponding to a periodic travelling wave solution [60]. The saddle points have at least two unstable eigenvalues in the boundary. This means that trajectories enter along their stable manifolds and exit along their unstable manifolds without leaving the boundary. The state therefore wanders from the vicinity of one travelling wave solution to the vicinity of another until it reaches a local relative attractor. The local relative attractor (an unstable attractor), has just one unstable eigenvalue so trajectories enter from all directions along the boundary but exit perpendicular to the boundary. Whether this trajectory then evolves towards a laminar solution or towards a turbulent solution, corresponding to stage 5 above, depends on the direction in which it exits the boundary. This, in turn, depends very sensitively on its initial position in phase space at stage 1.

In thermoacoustics, the analogous boundary is between trajectories that decay to zero, which are analogous to the laminar solution, and trajectories that evolve to high amplitude self-sustained oscillations, which are analogous to the turbulent solution. Recent studies, such as [30], have shown that a trajectory starting from this boundary can evolve to the high amplitude state. If the analogy is appropriate, however, trajectories from other regions of state space will be attracted towards this boundary and then be repelled either to a high or a low amplitude state. Thermoacoustic systems, like many hydrodynamic systems, are non-normal [49, 50, 51], meaning that this attraction can be from states with lower energy than the boundary. Thermoacoustic systems, unlike many hydrodynamic systems, have nonlinearities that do not conserve energy, which is another reason that this attraction can be from states with lower energy than the boundary.

It has been shown that background noise and noise from combustion processes is present in the limit cycles of combustors [27], and it is thus likely that some background noise and noise from combustion processes exists before the system is triggered to self-sustained oscillations. Furthermore, it has been shown theoretically that such noise can instigate triggering to these

'pulsed' oscillations [42]. This chapter will analyse whether this triggering process uses the unstable attractors in a similar way to bypass transition to turbulence.

The first aim of this chapter is to explore the analogy between triggering in thermoacoustics and bypass transition in hydrodynamics. The second aim is, in light of this link, to investigate the effect that different types of noise have on triggering in a simple thermoacoustic system.

## 3.2   Unstable attractors in the Rijke tube
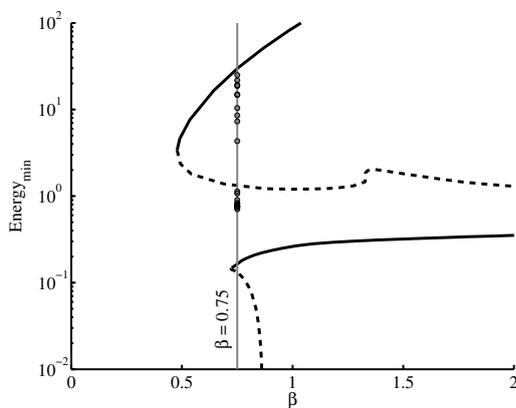


Figure 3.1: Bifurcation diagram for the 20-Galerkin mode system with $\tau$ of 0.02. The black points indicate the starting points for the time marching routine, taken at $\beta = 0.75$.

From an initial state, the system converges to either a stable limit cycle or the zero fixed point. If the initial state is close to a stable solution, the system will be attracted towards it. If the initial state is far from a stable solution, however, the system acts more interestingly. The aim of this section is to show that the unstable limit cycles can act as attractors during the transient phase, before repelling the trajectory towards one of the stable solutions. Such manifolds are *unstable attractors*, and are found in physical systems such as pulse-coupled oscillators [61]. These manifolds have a stable eigenvalue in at least one dimension (often more) and an unstable eigenvalue in at least one other dimension.

To demonstrate this, twenty different initial states are evolved forward in time for $\beta = 0.75$ using a $4^{th}$ order Runge-Kutta scheme to solve (2.8 - 2.9). These initial states have been chosen because they are all initially attracted towards the higher unstable limit cycle. For each initial state, the total initial acoustic energy is shown as a dot on Fig. 3.1 and the initial energy distribution between the Galerkin modes is shown in Fig. 3.2. Initial states 1 to 10 have less energy than the minimum energy on the higher unstable limit cycle. Initial states 11 to 20 have more energy than this.
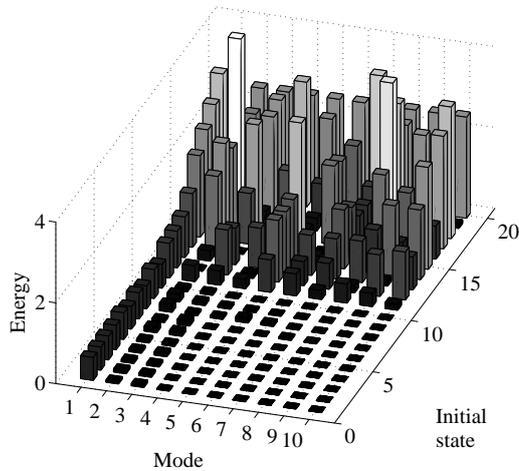
Figure 3.2: Energy distribution between the modes of the initial states (modes 11-20 have zero energy). Initial states 1 to 10 have low energy, initial states 11 to 20 have high energy.
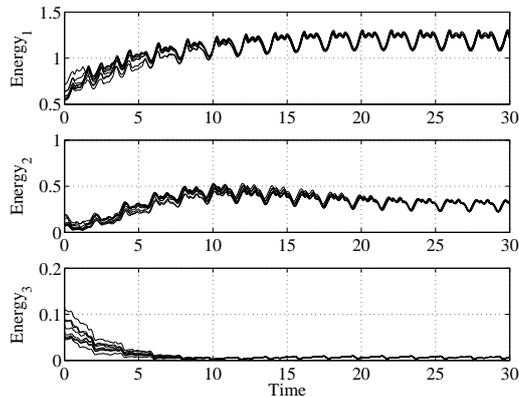


Figure 3.3: Time evolution from the ten low energy initial states shown in Figure 3.1. The acoustic energy of the first 3 Galerkin modes is plotted. Higher Galerkin modes exhibit a similar decay to the third mode.

For the ten initial states with low energy, the evolution of the first three Galerkin modes is plotted in Fig. 3.3. They initially evolve towards the same periodic trajectory in state space, which is the higher unstable limit cycle. The energies of the first two modes grow transiently towards this while those of the third (and higher) modes decay. Although not shown in Fig. 3.3, the trajectories then continue alongside the unstable limit cycle for several hundred time units before either decaying to the lower stable limit cycle or growing to the higher stable limit cycle [17]. It is worth noting that most of the energy in these initial states is in the first three Galerkin modes. A full analysis of the transient growth process and a non-linear adjoint looping algorithm for finding the lowest initial energy that can evolve to the higher unstable limit cycle can be found in Ref. [17].

For the ten initial states with high energy, the evolution of the first three Galerkin modes
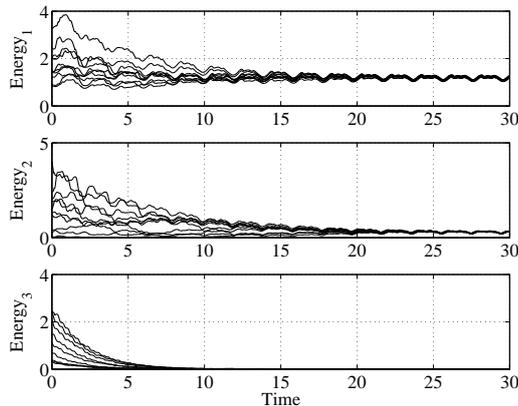
Figure 3.4: Time evolution from the ten high energy initial states shown in Figure 3.1. The acoustic energy of the first 3 Galerkin modes is plotted. Higher Galerkin modes exhibit a similar decay to the third mode.

is plotted in Fig. 3.4. They also evolve towards the higher unstable limit cycle and, from there, either to the lower stable limit cycle or to the higher stable limit cycle. These initial states, however, lose a large proportion of their energy during the transient period. This can partly be explained by the damping model, $\zeta_j = 0.05j^2 + 0.01j^{1/2}$, which damps higher modes more than lower modes. It is worth noting, however, that most of the initial states 11 to 20 start with more energy in the first three modes than initial states 1 to 10. Thus growth or decay from an initial state cannot be determined simply with an energy threshold condition.

Figs 3.1 to 3.4 demonstrate that the higher unstable limit cycle is an unstable attractor whose basin of attraction spans a wide energy range. Trajectories that are attracted towards this limit cycle are ultimately repelled either towards the higher stable limit cycle or the lower stable limit cycle. Although not shown in this chapter, the same is true of the lower unstable limit cycle [17]. The basins of attraction of both unstable limit cycles include regions with lower energy than the point with lowest energy on the respective unstable limit cycle. This low energy region reaches the unstable limit cycle via transient growth, which is a feature of non-normal systems, and nonlinear processes that do not conserve energy, which is a feature of thermoacoustic systems but not hydrodynamic systems. This process of attraction followed by repulsion is analogous to bypass transition to turbulence in hydrodynamics. Specifically, it corresponds to stages three and four in Ref. [45] and the unstable attractors are simpler versions of the 'edge of chaos' described in Refs. [58, 59, 60].

During the evolution, each Galerkin mode oscillates at its own frequency, which varies in time and is not necessarily its natural frequency. The frequency and growth rate of the first Galerkin mode, which has the highest energy, are calculated from the data in Figs. 3.3 and 3.4, and their subsequent evolution with time. They are plotted as functions of each other in

Fig. 3.5 for initial states 1 to 10 (upper figure) and initial states 11 to 20 (lower figure). The limit cycles have zero growth rate. The low energy stable limit cycle (LES) lies to the right of Fig. 3.5 and the high energy stable limit cycle (HES) lies to the left. The unstable limit cycle, to which the initial states all converge, lies in the middle. Initial states 1 to 10 have positive growth rates towards this point while initial states 11 to 20 have negative growth rates towards this point.
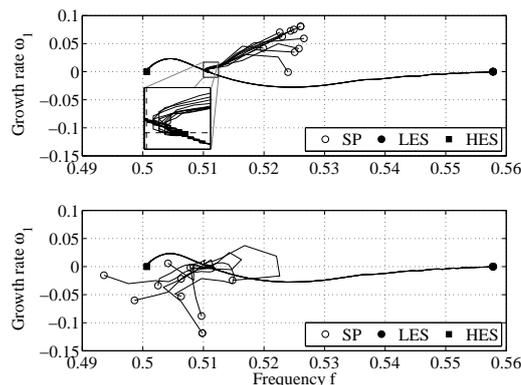


Figure 3.5: Evolution of the frequency and growth rate of the $1^{st}$ Galerkin mode from starting points (white circles) to either the high stable limit cycle (black squares) or the low stable limit cycle (black circles). Top frame: low energy initial states. Bottom frame: high energy initial states.

The phase portraits in Fig. 3.5 represent measurable outputs and can be compared with previous theoretical and experimental results. In a comprehensive theoretical and experimental study, Ref. [30, Figs. 8,9] found trajectories between the unstable limit cycle and the stable limit cycles that are very similar to those found in this report. Ref. [30], however, considered only the fundamental mode and assumes higher harmonics have negligible effects on system stability. Transient growth or transient decay towards the unstable limit cycle cannot be modelled in a single mode system [17], which means that Ref. [30] could capture the transition away from the unstable limit cycle but could not capture the preceding transient behaviour that is the subject of this chapter.

## 3.3 Noise transitions via unstable attractors

The previous section assumes that the system is noiseless and deals with triggering to sustained oscillations from a given initial state. This is useful because it gives the maximum initial energy below which all states decay to the zero solution (or to the lower stable limit cycle) in a noiseless system. This section examines the effect of forcing the system with low levels of noise, firstly as bursts of noise, secondly as continuous noise.

Noise can be divided conceptually into four types: additive noise, where a small forcing is added continually to the system; parametric noise, where coefficients in the governing equations vary; multiplicative noise, where noise amplitude is proportional to the current state of the system [62]; and modal noise, where energy is redistributed between the Galerkin modes, without any overall change in energy.

Multiplicative and parametric noise have been shown to move the stability boundaries of a combusting system [42, 62, 63]. Additive noise, however, excites the system but has no effect on the location of the stability boundaries. This section deals with additive noise, and aims to show that the excitation from the additive noise can be large enough to reach a non-linear stability boundary, in this case the basin of attraction of an unstable attractor, as seen in §3.2. In this way, the stability boundaries are not moved, but are reached. For each noise strength, therefore, there will be an effective stability boundary, that will be different from the normal stability boundaries if triggering can occur.

At certain parameter values and forcing levels, the evolution of the system is very sensitive to the initial conditions and the forcing. This means that simulations with stochastic forcing become less clear to analyse. In order to make simulations repeatable and comparable with each other, the characteristics of the noise are determined in advance by specifying amplitudes and relative phases of the forcing of the first 10 Galerkin modes. Above the $10^{th}$ Galerkin mode, the response of the system to forcing is very weak and so is not used here to form the noise profiles. This forcing signal is periodic, with period equal to that of the natural frequency of the first Galerkin mode. Three types of noise are considered: white noise, in which each mode is forced equally; pink noise, in which the lower modes are forced more than the higher modes; and blue noise, in which the higher modes are forced more than the lower modes. The noise profiles used in this section are not optimal, in that the phases chosen will not exhibit triggering with the minimum noise energy possible. However, the noise profiles chosen in this section are representative of the trends between white, pink and blue noise.

### 3.3.1 Bursts of noise

Small bursts of noise in the system are examined in order to discover whether certain noise profiles are more successful than others at triggering sustained oscillations. As shown in §3.2, certain initial states are more successful at triggering sustained oscillations in a noiseless system. It seems logical to suppose, therefore, that the same will be true of bursts of noise.

The first noise profile is white noise, in which every Galerkin mode has equal amplitude, Fig. 3.6. The evolution of acoustic energy is shown in Fig. 3.7, in which this noise is added for
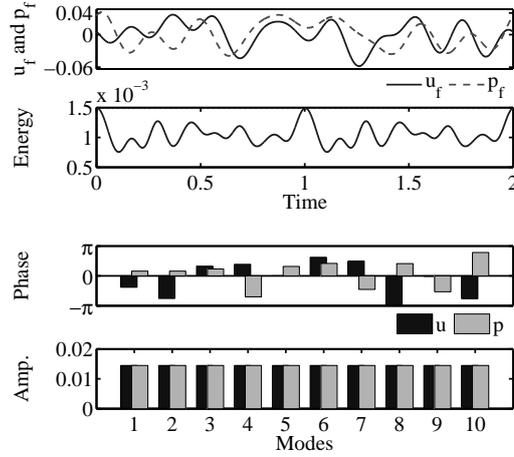
Figure 3.6: White noise profile in terms of the forcing signal's amplitude (top time series) and energy (bottom time series). Each Galerkin mode is forced with equal amplitude (bottom bar chart) but randomly-distributed phase (top bar chart). The noise level is quantified by the maximum energy of the signal, which is $1.49 \times 10^{-3}$.
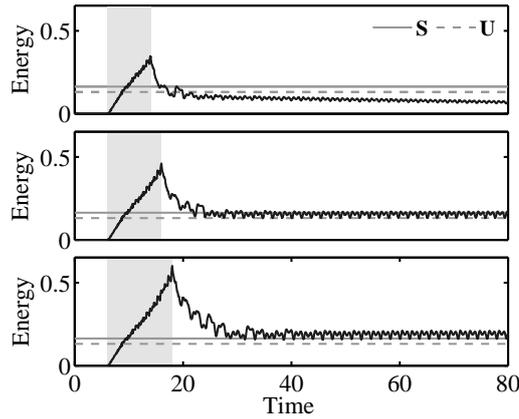


Figure 3.7: Evolution of the energy of the system with bursts of white noise for 4, 5 and 6 periods (shaded), showing attraction to the stable (S) and unstable (U) limit cycles.

4, 5 and 6 periods. Each time unit corresponds to $L_0/c_0$ seconds, where $L_0$ is the tube length and $c_0$ is the speed of sound, so these bursts would be of order 10 ms in a typical Rijke tube.

The three plots in Fig. 3.7 capture the three types of behaviour that are expected. After forcing for 4 periods (top plot), the system is not yet in the basin of attraction of the unstable limit cycle (dashed line) and subsequently decays to the zero solution. After forcing for 5 periods, the system reaches the basin of attraction of the unstable limit cycle, is attracted towards this solution and, from there, grows to the stable limit cycle (solid line). After forcing for 6 periods, the system decays to the stable limit cycle without passing via the unstable limit cycle.

The second noise profile is shown in Fig. 3.8. This is a blue noise profile, in which the

26
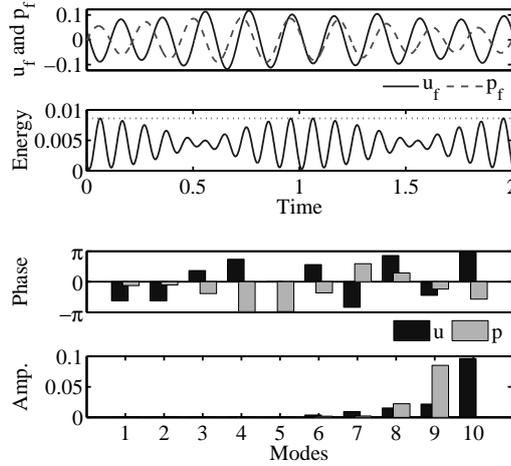
Figure 3.8: Blue noise with energy $8.6 \times 10^{-3}$ (as for Fig. 3.6)
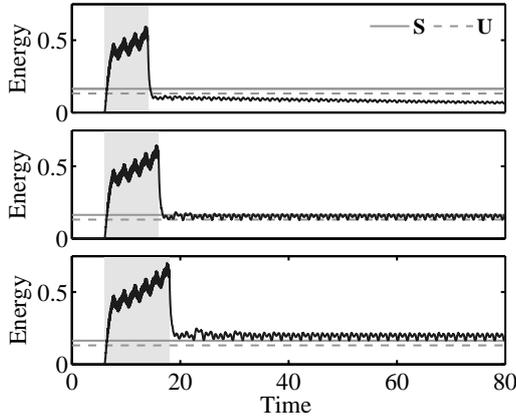


Figure 3.9: As for Fig. 3.7 but for blue noise.

higher Galerkin modes are forced with higher amplitude than the lower Galerkin modes. The maximum energy of the forcing signal is $8.6 \times 10^{-3}$, which is 5.8 times greater than that of the white noise profile. The evolution of acoustic energy is shown in Fig. 3.9, in which this noise is added for 4, 5 and 6 periods.

The three plots in Fig. 3.9 capture the same three types of behaviour seen for the white noise. The blue noise, however, requires much higher energy than the white noise to trigger self-sustained oscillations. This is expected from the results in §3.2, in which it was shown that a system reaches sustained oscillations from a lower initial energy when that energy is mainly in the lower Galerkin modes. Although not shown here, similar simulations with pink noise show that, as expected, less energy ($6.9 \times 10^{-4}$) is required to trigger the system to self-sustained oscillations than is required for white noise.

As well as triggering the system from the zero solution to the lower stable limit cycle, noise can trigger the system from the lower stable limit cycle to the upper stable limit cycle. An
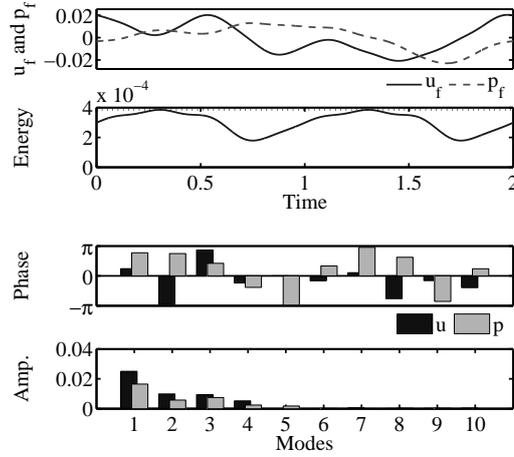
Figure 3.10: Pink noise with energy $3.85 \times 10^{-4}$ (as for Fig. 3.6)
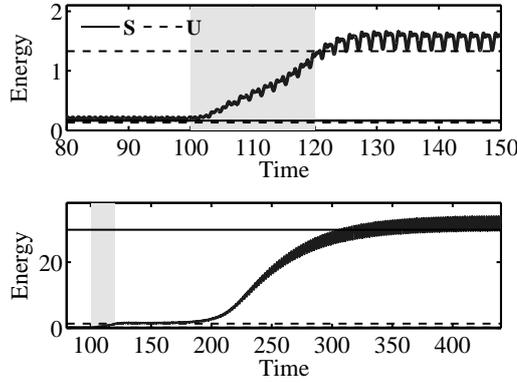


Figure 3.11: Energy evolution of the system with a 10 period burst of pink noise (shaded), showing transition from the lower stable limit cycle (S) to the higher stable limit cycle (S) via the higher unstable limit cycle (U).

example of this is shown for the pink noise profile in Fig. 3.10 with maximum energy $3.85 \times 10^{-4}$, applied for 10 periods. This corresponds to velocity and pressure perturbations of less than $\pm 2.2\%$ at the wire. The evolution of acoustic energy is shown in Fig. 3.11. The top plot shows the jump from the lower stable limit cycle (solid line) to the higher unstable limit cycle (higher dashed line) during the time when the noise is applied (shaded). The bottom plot shows the subsequent evolution from the higher unstable limit cycle to the higher stable limit cycle in the absence of noise. As expected from §3.2, it takes very little energy for pink noise to trigger the system to the higher stable limit cycle.

In summary, pink noise is more effective at triggering to the stable limit cycles than white noise, which in turn is more effective than blue noise. Noise can cause triggering from the zero solution to the stable limit cycles, as well as from the lower to the higher stable limit cycle. Very little noise is required, particularly if it leaves the system in a similar state to the 10 lower

energy points in §3.2, which will grow transiently to the unstable limit cycle. In more complex thermoacoustic systems, where the degree of non-normality and transient growth is higher [48], the latter effect will be more pronounced. Rapid triggering to high energy oscillations through the action of low energy noise has been seen in experimental combustors, such as in Lieuwen [27, Fig. 15].

It is worth noting that additive noise can also reduce the system's acoustic energy, particularly when added to the higher Galerkin modes. This is as expected from the 10 higher energy points in §3.2.

## 3.4 Conclusions

This chapter explores the analogy between triggering in thermoacoustics and bypass transition in hydrodynamics, which was demonstrated by Juniper [17]. The behaviour of a simple thermoacoustic system [17, 49] is compared qualitatively with that of simplified models of hydrodynamic systems [45, 58, 59, 60]. In the thermoacoustic system, it is shown that initial states over a wide energy range evolve first towards an unstable attractor and then towards a stable solution, which is either a limit cycle or a fixed point. Some of these states have lower energy than the lowest energy on the unstable attractor and make use of non-normal transient growth to reach it, which is analogous to bypass transition in hydrodynamics. These initial states have higher amplitudes at low frequencies.

In light of this analogy, this chapter then explores the effect that different types of noise have on triggering. Three types of noise are considered: pink noise (higher amplitudes at low frequencies),white noise (similar amplitudes at all frequencies) and blue noise (higher amplitudes at high frequencies). The noise is applied both as short bursts and continuously. In both cases, pink noise is more effective than white noise, which is more effective than blue noise, at causing triggering to a higher amplitude limit cycle. Indeed, blue noise can even inhibit triggering. These results concur with the results in the first part of the chapter. The noise signature of flames has been shown to be pink in nature [64], so these results are pertinent for ducted flame systems.

The triggering mechanism presented in this chapter has since been demonstrated to occur for a diffusion flame in an acoustic duct, in a detailed experimental study by Jegadeesan and Sujith [65, 66]. Small amplitudes of noise were shown to cause triggering to a high amplitude limit cycle via an unstable limit cycle. The similarity of these experimental results is discussed further at the end of the next chapter.

# Chapter 4

# Noise induced triggering

In the previous chapter, the Rijke tube model was analysed and a mechanism of triggering was proposed with an analogy to bypass transition in hydrodynamics. It was shown that unstable attractors exist in thermoacoustic systems, and that in the presence of low amplitude periodic noise the system evolves via the unstable attractors to reach higher amplitude periodic states.

All physical thermoacoustic systems contain noise, so even if the behaviour of a thermoacoustic system has been characterised without noise, it is important to think about how the system would behave when perturbed by noise. The concept of practical stability is introduced in this chapter to describe the stability of a system when perturbed by noise.

In the first half of this chapter, the effect of random noise on the system is examined, and a method is presented for quantifying noise transitions with a stochastic description. Practical stability results are shown for the hot wire Rijke tube model, which demonstrate that random noise causes triggering before the system becomes linearly unstable. Furthermore, the point at which triggering occurs is shown to be stochastic and dependent on the strength of the noise. These results are compared against those from an experimental combustor at the end of the chapter. Good qualitative agreement is seen.

In the second half of this chapter, the effect of different types of noise is explored. Noise can be characterised in several different ways, such as additive, parametric and multiplicative noise. To give a complete description of the effect of noise on the hot wire Rijke tube model, several different types of noise are examined. Each shows qualitatively similar behaviour, in that triggering occurs before the system becomes linearly unstable.

The content of this chapter was originally published in Ref. [2], co-authored with Matthew Juniper.
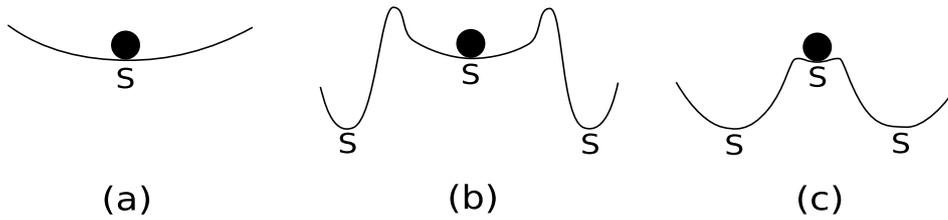
Figure 4.1: All three systems are linearly stable about the central point but, in (b) and (c), finite perturbations can excite the system to the other stable states. The system in (b) is more practically stable than in (c) because the central basin of attraction is larger and more attracting.

## 4.1 Introduction

Emphasis in thermoacoustics has often been placed on the importance of linear stability, but in this chapter we wish to emphasise the importance of practical stability. Linear analysis is valid for infinitesimal perturbations from the linearisation point, and linear stability implies that all infinitesimal perturbations will eventually decay. If the system has only one stable state, then linear stability further implies that all finite perturbations will eventually decay. If the system has more than one stable state, however, then linear stability can no longer imply that all finite perturbations will eventually decay. If a finite perturbation is large enough for the system to reach the basin of attraction of another stable state, then the original stable state is unstable to a perturbation of that size.

Practical stability is a measure of the stability of a state in the presence of noise [67], and has been studied in the context of chaotic attractors and shown to be important in their control [68, 69]. The difference between linear stability and practical stability is highlighted in Figure 4.1. All three scenarios are linearly stable about the central point. When no other stable solutions exist, Figure 4.1a, linear stability implies stability both to infinitesimal and finite perturbations. In Figure 4.1b, three stable states exist and a large enough perturbation can excite the system to either of the other stable states. In Figure 4.1c, the basin of attraction around the current stable state is much smaller and shallower, so a much smaller perturbation is required to reach either of the other stable states. Figure 4.1b is thus more practically stable than Figure 4.1c.

To define practical stability [67] we must first define what we consider to be 'stable', and then measure when this stability criterion is obeyed, in terms of the initial system condition, $x_0$, and perturbations in time $p(x, t)$. In the context of triggering, we will define stability here to be when the system remains in the basin of attraction of the steady state (fixed point at 0). The basin of attraction of the fixed point is termed $Q$, where $Q \subset \mathbb{R}^N$. We define the perturbations to be less than some magnitude: $||p(x, t)|| \leq \delta$ , $\delta \geq 0$, and we define the initial

conditions to be bounded: $\forall x_0 \in Q_0 \subset Q$. If we have a dynamical system:

$$\frac{\partial x}{\partial t} = F(x, t) \tag{4.1}$$

With a permanently acting perturbation this becomes [67]:

$$\frac{\partial x}{\partial t} = F(x, t) + p(x, t) \tag{4.2}$$

Then the practical stability in a given time period is defined by Yang [67] to be:

"The solution $x = 0$ to Eq. (4.1) is said to be practically stable with respect to $\delta$, $Q$ and $Q_0$ in finite time period $[t_0, T]$, if for every perturbation $p(x, t)$ with $||p(x, t)|| \leq \delta$ the solution $x(t_0, x_0, t)$ to Eq. (4.2) remains in $Q$ for $t \in [t_0, T]$ and $x_0 \in Q_0$."

This definition is complete, but it is difficult to compute basins of attraction for large nonlinear dynamical systems, and even more difficult to compute the action of the system under all possible perturbations $||p(x, t)|| \leq \delta$. In this chapter we aim to examine the practical stability of a thermoacoustic system under stochastic forcing, with perturbations, $p(t)$, that are independent of the current system state. Varying the mean amplitude of the perturbations will show how sensitive the practical stability is to background noise. The practical stability of the Rijke tube model to initial conditions has already been investigated by Juniper [17].

All physical thermoacoustic systems are subject to random noise, whether from incoming flow fields, combustion processes or mechanical vibration. Various types of noise have been shown to destabilise systems that are linearly stable when noiseless [42, 62, 63, 70]. Noise resulting from entropy and vorticity has also been shown to play a part in the process [42]. Non-normality has been highlighted as a key factor in the amplification of noise [71], and the susceptibility of a system to to be pushed out of the basin of attraction of the stable fixed point.

Stochastic delay differential equations have been studied, but mainly with linear analyses [70]. Ref. [70] showed that, for linear delay differential equations, additive noise has no effect on stability boundaries. Other studies examine the stochastic effects of parametric and multiplicative noise on a delay differential equation with a pitchfork bifurcation and show the resultant changes in stability boundaries [72]. Ref. [72] uses the result of [70] as evidence that additive noise will have no effect on stability boundaries. This chapter aims to demonstrate that, in nonlinear systems, one of the key roles of noise is to force the system from one attracting state to another, not necessarily to change the stability boundaries.

When stochastic forcing is present, probability density functions (PDFs) have been used to describe the response of thermoacoustic systems in both experiments and theory [27, 42, 62, 73, 74]. These papers show that a single limit cycle creates a bimodal PDF, and that the addition of other stable solutions creates additional peaks. These papers did not, however, map these PDFs over a sub- or supercritical bifurcation as in other disciplines [72, 75].

The aim of this chapter is to investigate the effect that noise has on triggering in a simple thermoacoustic system. The concept of a stochastic stability map is introduced in order to visualise the practical stability of a system under stochastic forcing. This can be used to define the region of practical stability of a thermoacoustic system for a given noise level. The chapter ends with comparison of the stochastic stability maps of the Rijke tube model and an experimental combustor. Good qualitative agreement is shown in both the stochastic stability boundaries and the proposed mechanism of triggering.

## 4.2   Noise induced triggering

Chapter 3.2 assumes that the system is noiseless and deals with triggering to sustained oscillations from a given initial state. This is useful because it gives the maximum initial energy below which all states decay to the fixed point at zero (or to the lower stable limit cycle) in a noiseless system. This section examines the effect of forcing the system with low levels of noise.

The bifurcation diagram in Figure 4.2 is shown for the noiseless system. In the bistable region, where both a stable fixed point at zero and a stable limit cycle exist, the system can trigger from the former to the latter if perturbed strongly enough and this can occur via the
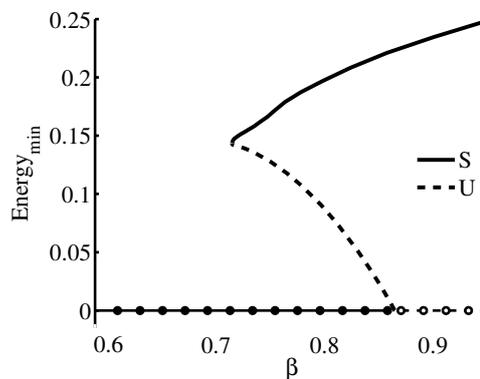


Figure 4.2: Bifurcation diagram for the 20-Galerkin mode system with $\tau = 0.02$. Dotted lines are steady states (fixed points), undotted lines are limit cycles. A solid line represents a stable solution, and a dashed line represents an unstable solution.

unstable limit cycle. Refs. [1, 17] show that the basin of attraction of the unstable limit cycle contains states with lower energy than the minimum energy on the limit cycle, due to non-normal and nonlinear growth around the unstable limit cycle. Therefore low amplitude noise may be enough to cause triggering via this mechanism.

In this section, we consider triggering to the stable limit cycle in Figure 4.2. The practical stability of the system will be examined when stochastic noise is present. Low levels of noise will be seen to initiate triggering and cause the system to be practically unstable in some operating conditions. Multiple random noise signals will be used to examine the practical stability of the system in a probabilistic manner.

### 4.2.1   Noise definition

Noise can be divided conceptually into four types: additive noise, where a small forcing is added continually to the system; parametric noise, where coefficients in the governing equations vary; multiplicative noise, where noise amplitude is proportional to the current state of the system [62]; and modal noise, where energy is redistributed between the Galerkin modes without any overall change in energy.

Noise is added to the heat release term in equation (2.9) to give equation (4.3), where $u_N$ is the noise term.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right) + j\pi\eta_j + \zeta_j\left(\frac{\dot{\eta}_j}{j\pi}\right) + 2\beta\sin(j\pi x_f)\left(\left|\frac{1}{3} + u_f(t-\tau) + u_N(t-\tau)\right|^{\frac{1}{2}} - \left(\frac{1}{3}\right)^{\frac{1}{2}}\right) = 0,$$
(4.3)

This simple noise model examines the response of the heat release to base flow noise at the wire. (General base flow noise propagating through the tube would require the tube boundary conditions to be relaxed, which is beyond the focus of this study.) The noise profile is independent of the current state of the system when added to $u_f$, but the action of the noise is not purely additive or independent of the system state because it must combine with $u_f$ within the modulus and square root. It is a mixture of multiplicative and additive noise. A purely additive noise signal will be analysed in section 4.3.1 and shown to produce similar results.

The noise signal is generated by defining an amplitude distribution in the frequency domain and assigning random phases to each discretised frequency. An inverse Fourier transform is applied to the noise frequencies to form the noise signal in the time domain. This method allows us to control the noise spectrum, and results in a signal that is random and non-periodic over the time domain. We have found that results for this simple noise model are qualitatively
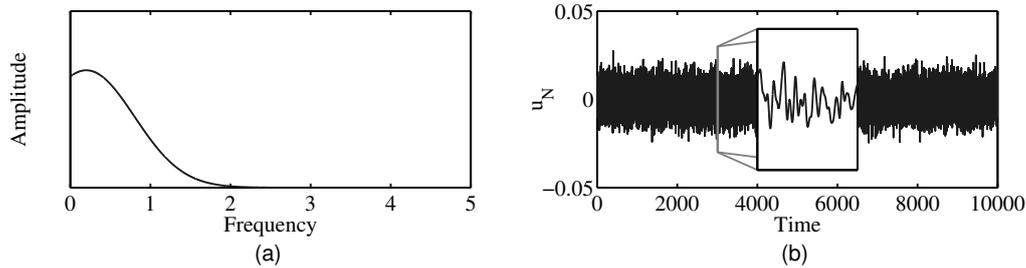
Figure 4.3: Pink noise profile in (a) the frequency domain and (b) the time domain, with a strength of 0.5%. The enlarged portion of the trace in (b) is 20 time units long.

similar to those using noise generated by the rigorously stochastic Ornstein-Uhlenbeck process [76].

The Rijke tube model used in this thesis was shown in chapter 3 to respond more strongly to low frequency noise [1]. Noise that has higher amplitudes at low frequencies, herein referred to as pink noise, will be used for the remainder of this chapter. A sample pink noise spectrum and time domain trace are shown in Figure 4.3. The spectrum is chosen to be a smooth function that has higher amplitude at lower frequencies. Higher frequency noise, which has lower correlation time, also induces triggering but requires a higher noise strength.

The noise profile is not periodic so the maximum amplitude is not a good measure of the noise strength. The noise strength in this chapter is quantified by the averaged absolute value of the noise signal, $u_N$, normalised by $u_0$. A root mean square measure could also have been used.

It was shown in §3.2 that the unstable limit cycles act as unstable attractors, and have a basin of attraction which contains states of higher and lower energy than that of the limit cycle itself. It seems logical to suppose, therefore, that if the system can reach these basins of attraction from the non-normal and non-linear growth of noise, then triggering may occur. The self-sustained oscillations that result from this process have sometimes been referred to as pulsed oscillations [26, 42]. In this section, we consider triggering to the lower energy branch of the bifurcation diagram in Figure 4.2.

### 4.2.2 Triggering mechanism

When sufficient noise is added, the model triggers to self-sustained oscillations as expected. If the strength of a particular noise signal is increased to the point where it first triggers then the system evolves via the unstable limit cycle. This is shown in Figure 4.4a and 4.4b, where a fractional increase in noise strength results in triggering. If the noise strength is increased
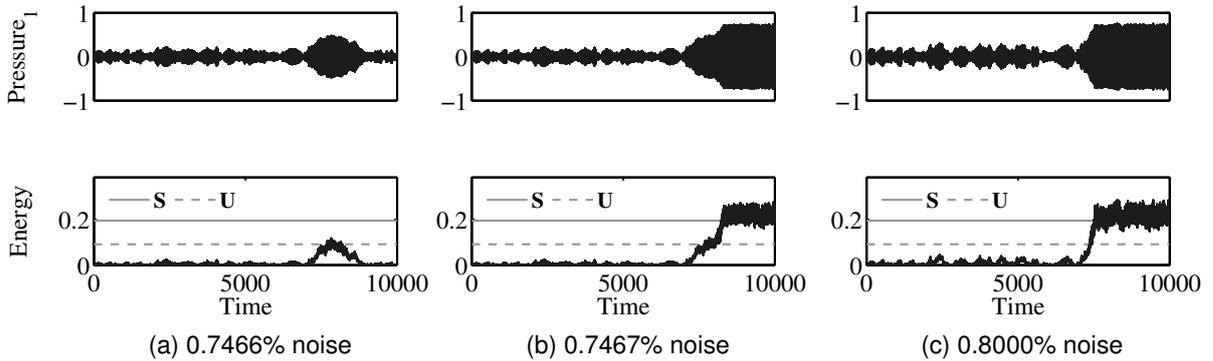
Figure 4.4: The system behaviour with $\beta = 0.8$ for a noise profile with amplitudes of (a) 0.7466%, (b) 0.7467%, (c) 0.8000%. The role of the unstable limit cycle in the triggering process is evident in (a) and (b), where a small increase in noise results in triggering. With the higher noise amplitude in (c), however, the system triggers directly to the stable limit cycle. Pressure traces in this chapter are shown for the fundamental mode only, which contains most of the perturbation energy.

further then the system evolves directly to the stable limit cycle. This is shown in Figure 4.4c and in Ref. [1].

At the limit of triggering, the trajectory of the system is affected by the proximity of the unstable limit cycle to the stable fixed point at zero. In Figure 4.6 we plot the system response to noise at the limit of triggering for the three values of $\beta$ identified in Figure 4.5. In Figure 4.6c, only weak noise is required to reach the unstable limit cycle and the system stays near the unstable limit cycle for many cycles. In Figure 4.6b, stronger noise is required to reach the unstable limit cycle and the system stays near the unstable limit cycle for few cycles.

A conceptually different response is shown in Figure 4.6a, where the energy of the unstable limit cycle is close to that of the stable limit cycle. (Energy alone is not enough to quantify
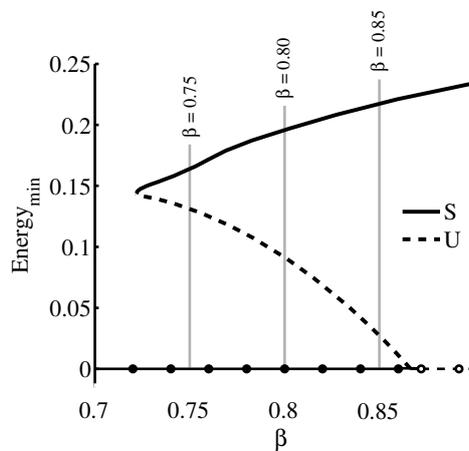


Figure 4.5: Bifurcation diagram showing $\beta$ values considered in Figure 4.6.
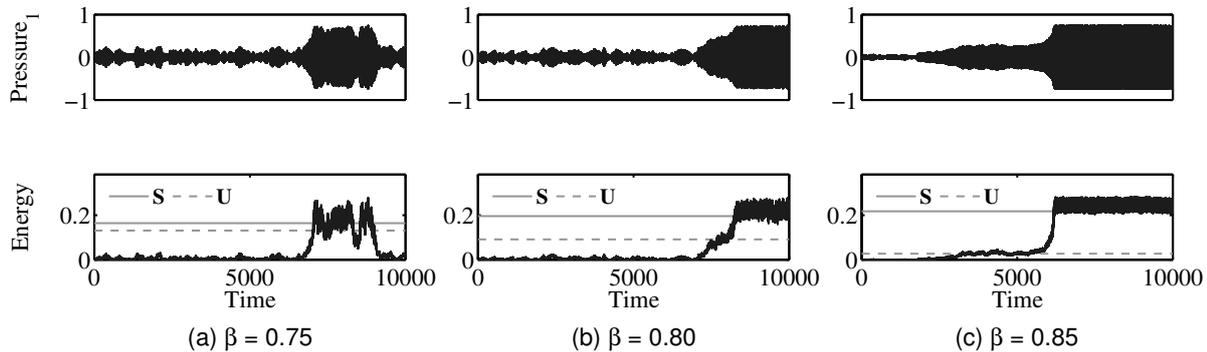
Figure 4.6: Triggering at three different values of $\beta$, as illustrated by Figure 4.5. Noise is applied with strength (a) 1.2%, (b) 0.73% and (c) 0.25%.
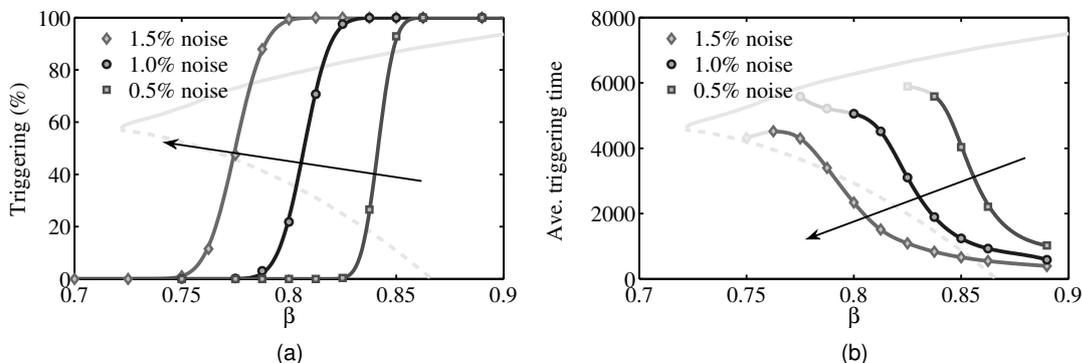


Figure 4.7: Results from 4000 runs of 10000 time units with randomly generated pink noise. The plots show (a) the percentage of triggering incidences and (b) the average time to triggering, for noise of strengths 1.5%, 1% and 0.5%. Fainter points in (b) have larger errors due to their low triggering rate. Arrows show the direction of increasing noise.

the proximity of two trajectories in $N$-dimensional state space, but both trajectories are rich in the lower Galerkin modes and are close in state space.) The strong noise in this simulation quickly triggers the system to the stable limit cycle and is also strong enough to dislodge the system back to the fixed point at zero. The ability of the noise to link the basin of attractions of the stable limit cycle and the fixed point at zero is examined further in §4.2.3.

The pressure trace of Figure 4.4b is similar to that seen in the experiments of Ref. [27, fig. 15b], in which the oscillations "first jump up rapidly, then stay relatively constant, and then smoothly increase" . These experiments could indicate the presence of an unstable attractor in a real combustor.

The region of practical instability for a particular noise strength can be visualised by examining the trends of repeated runs. Here 4000 evolutions of 10000 time steps are calculated for each value of $\beta$ and noise strength, and the trends are plotted in Figure 4.7. The percentage of runs that results in triggering is shown in Figure 4.7a for three different noise levels. For

each noise level, there is a sharp transition between the $\beta$ values that do and do not result in triggering. The points are fitted with functions of the form $y = 50\,\text{erf}\,(a\,(x - \mu)) + 50$, where $a$ and $\mu$ are scaling and shifting constants. As noise strength increases, this transition occurs at lower values of $\beta$, further away from the Hopf bifurcation. This shows qualitatively the same result as Ref. [75, Fig. 10], where the most probable amplitudes and average amplitudes are plotted for an electronic oscillator with increasing noise. Above the linear stability limit ($\beta > 0.866$), all runs reach the limit cycle. This is due to linear amplification of perturbations followed by nonlinear saturation to the limit cycle, rather than triggering.

It is also important to understand whether triggering is immediate or delayed. The trend of average time to triggering is plotted in Figure 4.7b. We define triggering to have occurred after the system has spent several cycles at the amplitude of the limit cycle. For each noise level, the average time to triggering decreases as the system nears and enters the linearly unstable region. An increase in noise strength results in a drop in average time to triggering at each $\beta$ location. Above the linear stability limit ($\beta > 0.866$), a finite time is still required for the system to linearly amplify the perturbations and saturate to the limit cycle.

### 4.2.3 Stochastic stability maps

As demonstrated in §4.2.2, a system with a subcritical bifurcation can be triggered with low amplitude noise. In the presence of noise, therefore, the linear stability limit in a bistable region can be misleading. To analyse the practical stability of the system with stochastic noise, a probabilistic approach must be used. The concept of a stochastic stability map is introduced in this section.

Figure 4.7 shows how likely a system is to trigger as a function of noise and $\beta$ but gives no other information. For a given energy-time trace, such as Figure 4.8, the time spent by the system at each energy level can be quantified by dividing the energy signal into quantisation
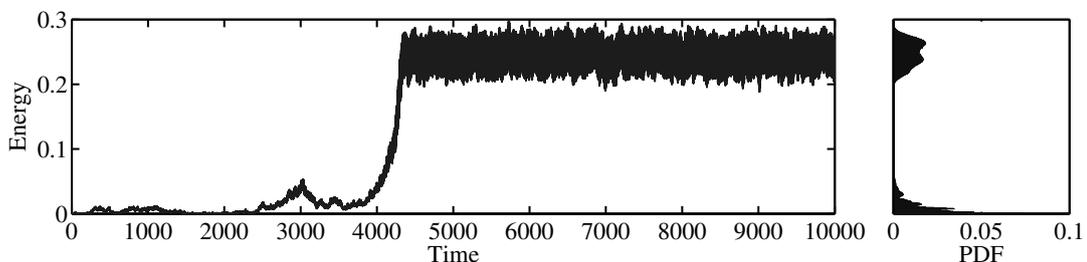


Figure 4.8: An energy-time trace is taken for $\beta = 0.85$, with noise of 0.5%. The energy of the system is divided into quantisation levels, and a normalised PDF created from the energy-trace. PDFs from multiple time traces can be averaged for a set noise strength and $\beta$, to obtain a smoother curve which is then used as a 2D slice in Figure 4.9.
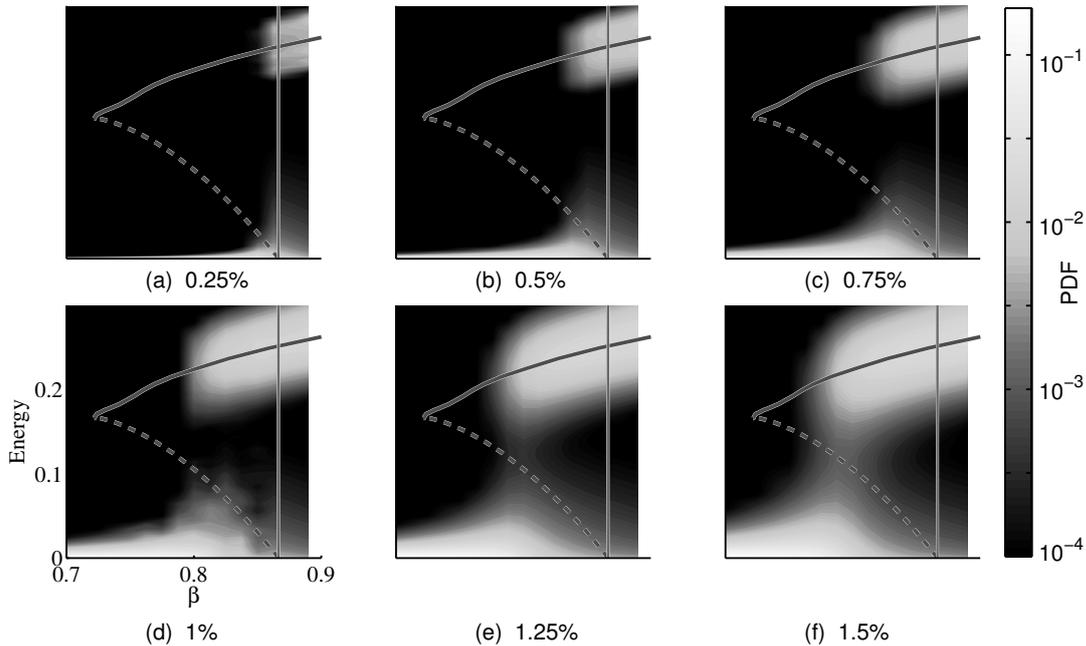
Figure 4.9: Probability Density Functions (PDFs) of the system energy with increasing velocity noise strength, against the bifurcation diagram of the noiseless system. The bifurcation diagram for the noiseless system is plotted for the mean energy of the limit cycles. The PDFs are generated with 4000 runs of 10000 time units for each $\beta$ and noise strength combination. The vertical line marks the onset of linear instability. A floor of $10^{-4}$ is applied to the PDF data for plotting.

levels and calculating a probability density function (PDF). In this figure, the increase in the PDF around energy of 0.2-0.3 shows that the system has triggered.

If enough time series of a set length are examined for a particular $\beta$ and noise strength, but with randomly generated noise signals, then a smooth average PDF is created. The PDF depends on the length of the time window, particularly for time windows that are of the same order as the average triggering time. This sensitivity is particularly evident around the linear stability limit, where triggering is very likely to occur for a given noise level. If enough time series of a long enough length are used, however, then the PDFs become a useful quantity to define a stable operating region.

For a fixed noise level, this process can be repeated at different $\beta$ values and the 2D slices combined to form a surface. We refer to this as a stochastic stability map, shown in Figure 4.9 for 6 levels of increasing noise, where the greyscale represents the PDF. The bifurcation diagram of mean oscillation energy and the linear stability limit (vertical line) are superimposed onto each map.

In the absence of noise, the system would only reach self-sustained oscillations beyond the linear stability limit. In Figure 4.9a only a small amount of noise is applied. The stochastic stability map shows that the system will reach self-sustained oscillations beyond the linear
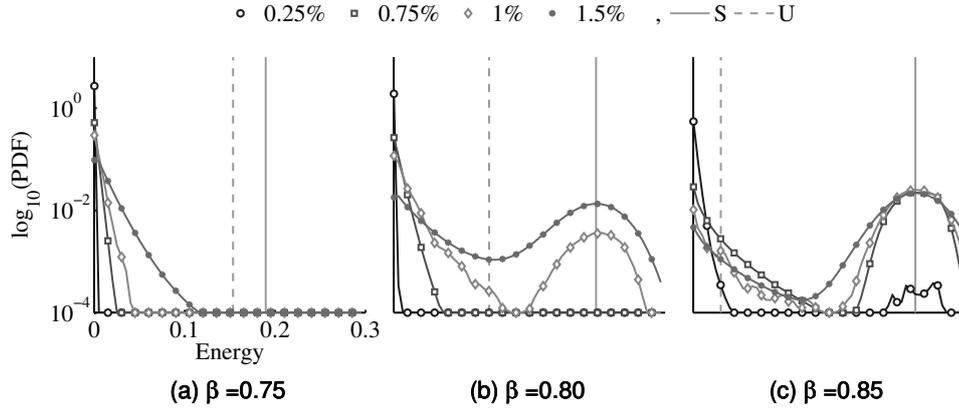
Figure 4.10: Probability Density Functions (PDFs) of the system energy at three values of $\beta$, as shown in Figure 4.5, with increasing noise. The results are plotted for 4000 runs of 10000 time units for each $\beta$ and noise combination. A floor of $10^{-4}$ is applied to the PDF data.

stability limit, but also shows the system is likely to trigger slightly before it. As the strength of noise is increased in Figure 4.9b-c, this region of triggering extends to lower values of $\beta$. There is a sharp transition between $\beta$ values that do and do not trigger, which is also seen in Figure 4.7. The system starts from rest so, even beyond the linear stability limit, some time is required for the system to reach the stable limit cycle. Consequently, the PDF value near the fixed point at zero in Figure 4.9 and the average time to triggering in Figure 4.7b are non-zero.

Figure 4.9b-c correspond to the cases in Figure 4.6b and c, where the noise level is just high enough to cause triggering, but not large enough to dislodge the system back to the fixed point at zero. This is seen in the PDF diagrams, in which the system lies in two distinct regions. One is between the fixed point at zero and the unstable limit cycle, the other is around the stable limit cycle. This shows that, once the unstable limit cycle is reached, the system quickly reaches the stable limit cycle and is unlikely to return. The PDFs in this region are qualitatively the same the theoretical results from the combustion chamber model of [42].

In Figure 4.9d-f a bridge develops between these two regions around $\beta = 0.8$. This bridge corresponds to energy traces such as in Figure 4.6a. Here the noise is strong enough to trigger the system from the fixed point at zero to the stable limit cycle and also strong enough to dislodge the system back from the stable limit cycle to the fixed point at zero. This forms bursts of high energy oscillations as seen in Figure 4.6a. At the same noise level, the bridge is much less pronounced at higher $\beta$, because the unstable limit cycle is further from the stable limit cycle, and therefore the system is less easily dislodged from the stable limit cycle.

The formation of the bridge can be be seen clearly in Figure 4.10. This plots 2D slices through the stochastic stability maps at three locations on the bifurcation diagram, as shown in Figure 4.5. The PDFs of four different noise levels are compared at these three locations.

When the energy of the unstable limit cycle is high, as in Figure 4.10a, none of the noise strengths result in triggering. As the noise increases, the spread around the fixed point at zero increases. In Figure 4.10b, this spread comes close enough to the unstable limit cycle that the system can trigger to the stable limit cycle. In Figure 4.10c, as the noise increases, the probability of triggering and the spread around both stable solutions increase.

These results are qualitatively the same as those of Ref. [77], who studied a non-normal model of magnetic energy in a turbulent dynamo, which exhibits a subcritical bifurcation. By including multiplicative noise in the model, their system triggers from the zero fixed point to the basin of attraction of a second fixed point. This lowers the PDF around the zero fixed point and creates a new peak around the second fixed point. If their noise intensity is increased further, their system can be dislodged back into the basin of attraction of the zero fixed point. The same process can also dislodge their system into the basin of attraction of a third fixed point.

System responses such as in Figure 4.6a, where the system displays quiet periods between periods with self-sustained oscillations, were also predicted using multiplicative noise in the model of [62, Fig. 5b]. It was assumed that this was due to the hysteresis region of a subcritical bifurcation. This may be true in the case of multiplicative noise, which moves the stability boundaries of the system, but is qualitatively different from the mechanism seen in this section. Here the bimodal PDF results from noise dislodging the system from the stable limit cycle to the fixed point at zero.

Ref. [71] neatly highlights the importance of non-normality in triggering by considering a model of transition to turbulence. The probability of leaving the basin of attraction of the fixed point at zero is plotted against a non-normality parameter for different strengths of noise [71, Fig. 1]. This probability increases with the noise strength, and increases strongly with the degree of non-normality. In combustors, the degree of non-normality and transient growth is large [48]. The Rijke tube model exhibits transient energy growth with a factor of around 1.4, whereas more complex thermoacoustic models can have much larger factors [78]. This will result in a stronger magnification of background noise and triggering with lower noise strengths.

## 4.3 Effect of different noise types

The noise term in equation (4.3) models the effect of velocity noise at the wire. The noise model in this section is expanded to include parametric and multiplicative noise. Parametric noise moves the stability boundaries relative to the system's current state. This can temporarily
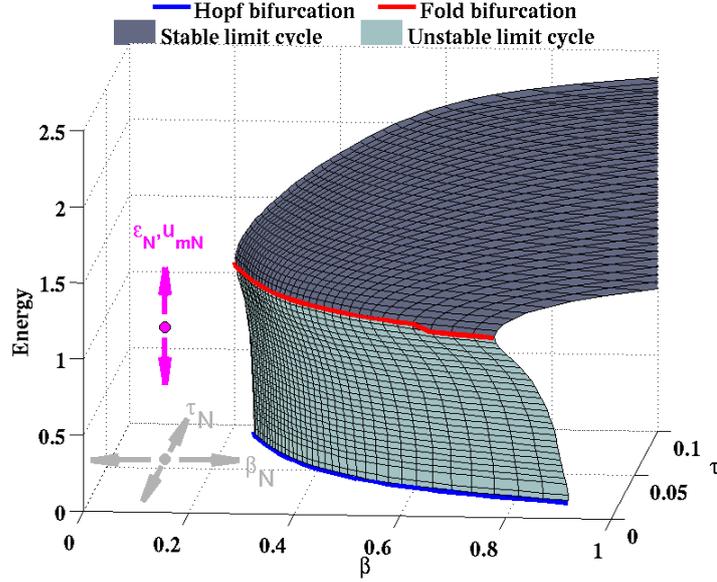
Figure 4.11: Physical effect of different noise types on the state of the system (pink circle). Additive and multiplicative noise excite the system's energy ($\epsilon_N$ and $u_{mN}$), whereas parametric noise moves the stability boundaries relative to the state of the system ($\beta_N$ and $\tau_N$).

make the system linearly unstable, or temporarily push the system into the basin of attraction of the unstable limit cycle.

The new noise model is defined in equation (4.4), where $\zeta_{jN}(t), \beta_N(t)$ and $\tau_N(t)$ are the new parametric noise terms, $u_{mN}(t - \tau)$ is the multiplicative velocity noise term and $\epsilon_N(t)$ is the additive noise term.

$$
\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right) + j\pi\eta_j + \zeta_j(1 + \zeta_{jN}(t))\left(\frac{\dot{\eta}_j}{j\pi}\right) + 2\beta(1 + \beta_N(t))\sin(j\pi x_f) \times \ldots
$$
$$
\left(\left|\frac{1}{3} + u_f(t - (\tau + \tau_N(t))) \times (1 + u_{mN}(t - \tau))\right|^{\frac{1}{2}} - \left(\frac{1}{3}\right)^{\frac{1}{2}}\right) = \epsilon_N(t), \qquad (4.4)
$$

The physical effect of the different noise types is summarised in Figure 4.11, superimposed on the stability surface shown in Figure 2.3. The effect of the additive and velocity multiplicative noise ($\epsilon_N$ and $u_{mN}$) is to excite the energy of the system, but the effect of the parametric noise, here $\beta_N$ and $\tau_N$, is to move the system relative to the location of the stability boundaries. A bifurcation surface similar to that of Figure 2.3 is formed when varying $\zeta$ instead of $\tau$ for the y-axis. Damping noise ($\zeta_{jN}$) therefore has a similar effect to time delay noise ($\tau_N$).

The noise terms are generated in the same manner as Figure 4.3, but in this section the noise amplitude is measured in terms of the standard deviation, as a percentage of the unperturbed value. For the additive noise term, which has no mean value, a reference value of 1 is used.

### 4.3.1 Additive Noise

It has been shown with linear analyses that pure additive noise does not change stability boundaries [70]. In a nonlinear system, however, there may be multiple stable states. Additive noise does not change the stability boundaries of the system, but it may excite the system into the basin of attraction of another stable state. This section aims to show that purely additive noise will affect the practical stability of the system in a similar way to the velocity noise used in section 4.2. A purely additive noise signal is defined, $\epsilon(t)$, with the same low frequency spectrum as $u_N(t)$ in section 4.2.1. The strength of the noise is quantified by the averaged absolute value of the noise trace, normalised by one half of the pressure non-dimensionalisation quantity, $p_0 \gamma M$.

$$\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right) + j\pi\eta_j + \zeta_j\left(\frac{\dot{\eta}_j}{j\pi}\right) + 2\beta\sin(j\pi x_f) \times \left(\left|\frac{1}{3} + u_f(t-\tau)\right|^{\frac{1}{2}} - \left(\frac{1}{3}\right)^{\frac{1}{2}}\right) = \epsilon_{jN}(t), \quad (4.5)$$

The stochastic stability map for the system with additive noise is shown in Figure 4.12. The trends are qualitatively similar to those obtained for the velocity noise. The stochastic stability map shows that the system will definitely reach self-sustained oscillations beyond the
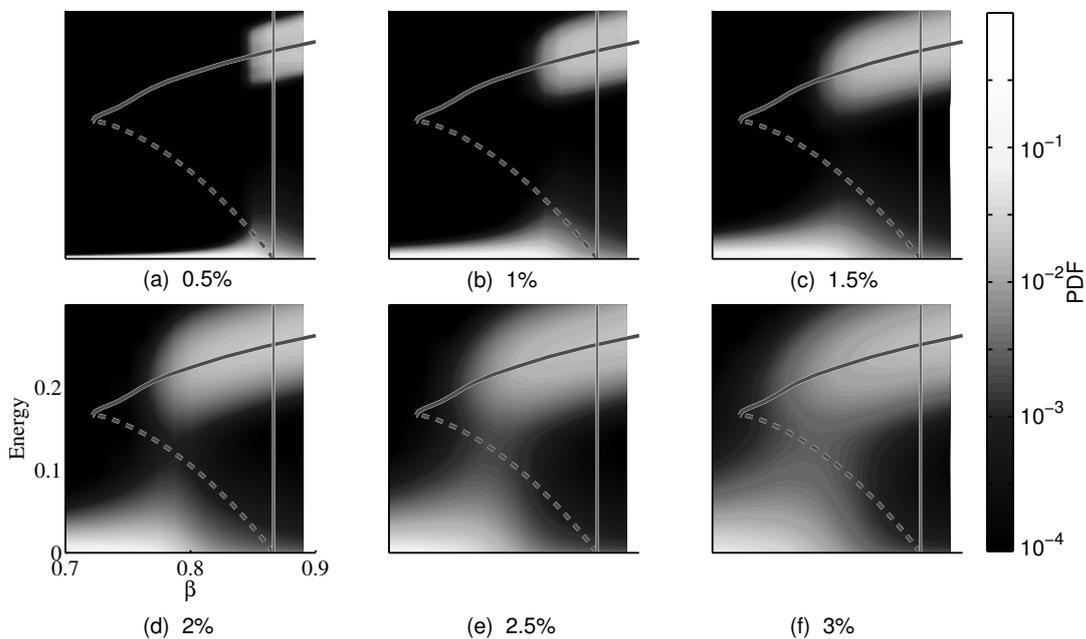


Figure 4.12: Probability Density Functions (PDFs) of system energy with increasing additive noise strength, against the bifurcation diagram of the noiseless system. The bifurcation diagram for the noiseless system is plotted for the mean energy of the limit cycles. The PDFs are generated with 4000 runs of 10000 time units for each $\beta$ and noise strength combination. The vertical line marks the onset of linear instability. A floor of $10^{-4}$ is applied to the PDF data for plotting.
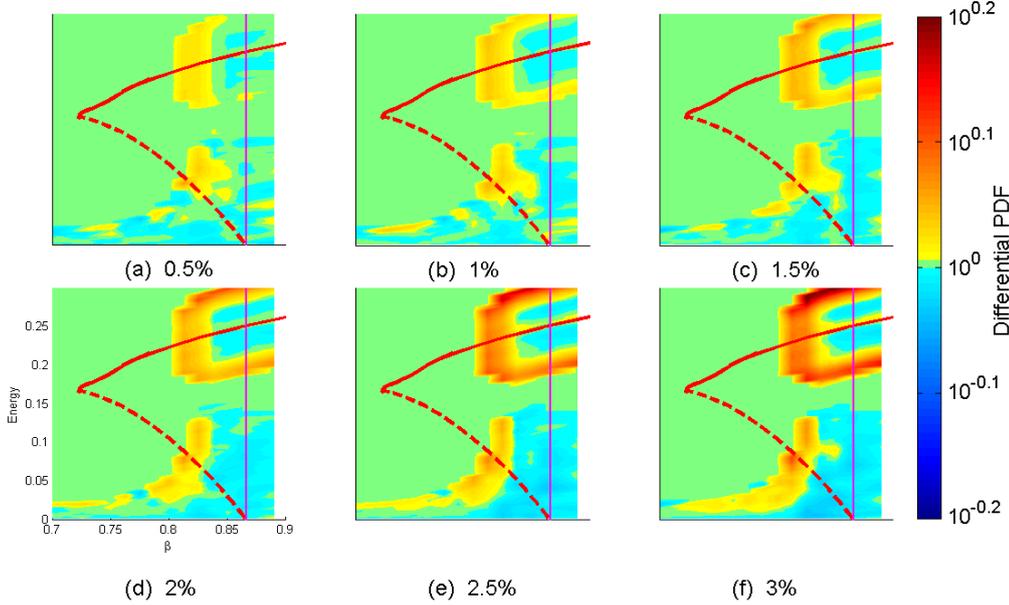
Figure 4.13: Differential Probability Density Functions (PDFs) of system energy with increasing wire temperature noise strength ($\beta_N$), relative to the PDFs of the system under additive noise of amplitude 1% alone, against the bifurcation diagram of the noiseless system. The bifurcation diagram for the noiseless system is plotted for the mean energy of the limit cycles. The PDFs are generated with 4000 runs of 10000 time units for each $\beta$ and noise strength combination. The magenta line marks the onset of linear instability. A floor of $10^{-4}$ is applied to the PDF data.

linear stability limit, but also shows that the system is likely to trigger slightly before it. As the strength of noise is increased in Figure 4.12 (b-c), this region of triggering extends to lower values of $\beta$. In Figure 4.12 (d-f), a bridge again develops between the two stable regions around $\beta = 0.8$.

## 4.3.2 Parametric noise

### 4.3.2.1 Wire temperature

Figure 4.13 shows a differential stochastic stability map for for the wire temperature noise ($\beta_N$). The multiplicative and parametric noise terms are applied to the system at the same time as applying 1% additive noise (Figure 4.12b). The noise results are therefore presented as differential PDFs, which have the logarithmic value:

$$\text{Differential PDF} = \log_{10} \frac{PDF_\beta}{PDF_{add\ 1\%}} \tag{4.6}$$

As defined by equation (4.6), a plotted value of $10^{0.2}$ means that the system is $10^{0.2}$ (1.6) times more likely to exist at that particular energy level than with 1% additive noise alone.
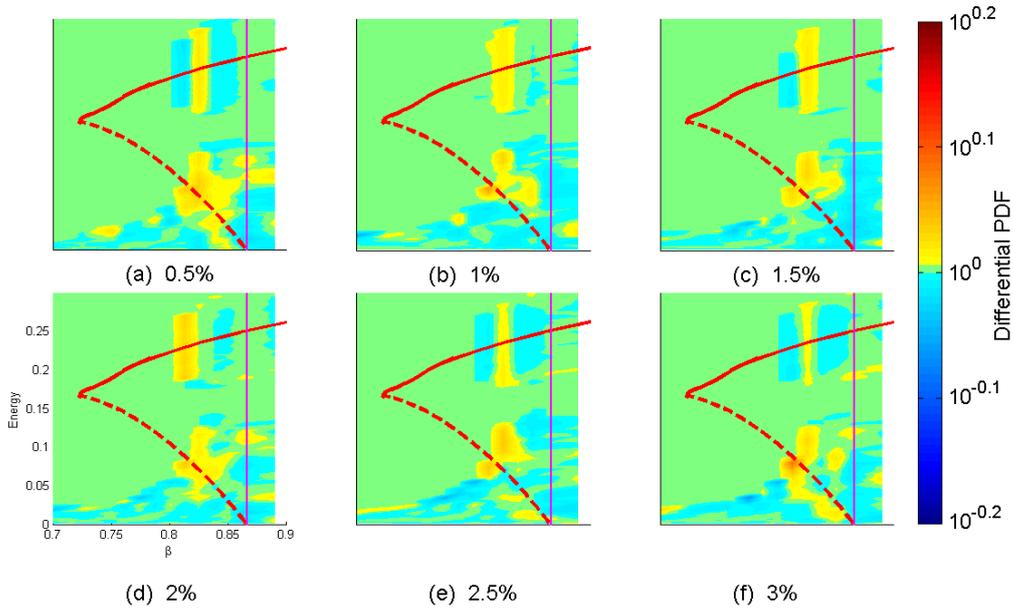
Figure 4.14: Differential Probability Density Functions (PDFs) of system energy with increasing time delay ($\tau$) noise strength, relative to the PDFs of the system under additive noise of amplitude 1% alone, against the bifurcation diagram of the noiseless system. Similar to Figure 4.13 but for time delay noise.

As the amplitude of wire temperature noise increases so does the likelihood of triggering, because the differential stochastic stability map is positive (orange) at the low $\beta$ end of the stable limit cycle[1]. This positive (orange) region forms a '**C**' shape around a negative (blue) region at higher $\beta$. This means that the probability distribution around the stable limit cycle at higher $\beta$ (e.g. 0.85), has become shallower and wider. This indicates that the system is varying in amplitude more around the stable limit cycle. A positive region is also present around the unstable attractor, suggesting that the system is more likely to be excited and attracted towards it. A large negative region also develops at low energies near the Hopf point. This is because the $\beta$ noise at this location sometimes temporarily pushes the system into a linearly unstable region, increasing the chance of triggering.

#### 4.3.2.2 Time delay

Figure 4.14 shows the equivalent results for time delay noise ($\tau_N$). They are slightly more complicated to analyse than the $\beta$ noise results, because the magnitudes of the differential PDFs are lower, but important characteristics of parametric noise can still be recognised.

---

[1]An error in the code resulted in the non-dimensional wire temperature noise, ($\beta_N$), being normalised by $\beta^2$ instead of $\beta$, thus the actual noise strength for $\beta_N$ in this chapter is 0.7-0.9 times smaller than the presented value. This does not change the qualitative trends observed

The differential PDFs are defined relative to the PDF surface with 1% additive noise alone (Figure 4.12(b)). Figure 4.12(b) shows that the probability of triggering is most sensitive to changes in $\beta$ when $\beta \approx 0.825$. This is because the probability of triggering has a similar error function profile to that of Figure 4.7a, the gradient of which is fitted by a Gaussian distribution which represents the sensitivity to triggering in terms of the parameter $\beta$. The peak of the Gaussian distribution is at $\beta \approx 0.825$. In Figure 4.14, at $\beta \approx 0.825$, the time delay noise increases the probability of reaching the unstable attractor and triggering.

A physical explanation of this can be visualised using the limit cycle surface in Figure 4.11, if the current system state (pink dot) were underneath the overhang of the subcritical bifurcation. For a given additive noise level, the probability of triggering is most sensitive to changes in $\beta$ when the system is excited just enough to be attracted towards the unstable attractor. This can be thought of in Figure 4.11 as when the system (pink dot) bounces just high enough to reach the unstable limit cycle surface (or more accurately a point just below the surface because the surface is attractive). If the time delay noise temporarily increases $\tau$, then the unstable attractor exists at a lower value of $\beta$, because the limit cycle surface is curved in the $\beta$-$\tau$ plane. The system may therefore have enough energy to be attracted to the unstable limit cycle surface at $\tau + d\tau$ when it did not have enough energy to be attracted at $\tau$. The time delay noise therefore has the largest effect at the most sensitive region for triggering.

For $\beta > 0.825$, the probability of triggering is actually impaired by the time delay noise, because the system can be shifted to $\tau - d\tau$, where the unstable limit cycle surface exists at lower $\beta$ values.

### 4.3.2.3  Damping

The results for damping noise ($\zeta_N$) are plotted in Figure 4.15. They are similar in nature to those for the time delay noise in Figure 4.14. The parametric noise has the largest effect where the probability of triggering is most sensitive.

### 4.3.3  Multiplicative noise

Figure 4.16 shows the equivalent results for multiplicative velocity noise ($u_{mN}$). In Figure 4.16(a), the multiplicative noise seems to inhibit triggering. As the noise intensity increases in Figure 4.16(c-f), however, the probability of triggering strongly increases at lower $\beta$ values. A very negative region also develops at low energies near the Hopf point, in a similar manner to the results for wire temperature noise. The spread of the system state around the limit cycle is also increased due to the multiplicative noise.
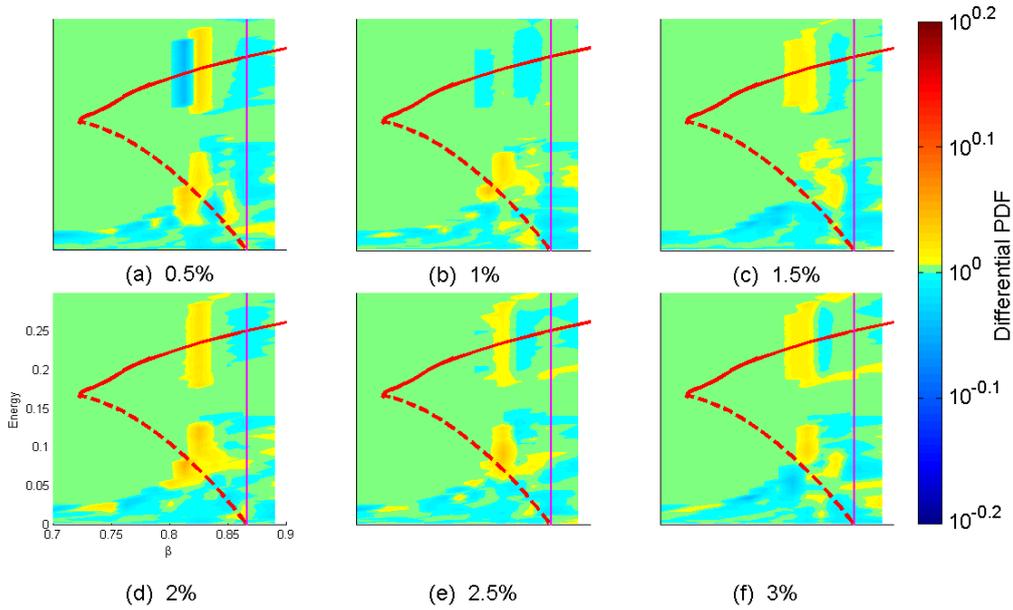
Figure 4.15: Differential Probability Density Functions (PDFs) of system energy with increasing damping ($\zeta$) noise strength, relative to the PDFs of the system under additive noise of amplitude 1% alone, against the bifurcation diagram of the noiseless system. Similar to Figure 4.13 but for damping noise.
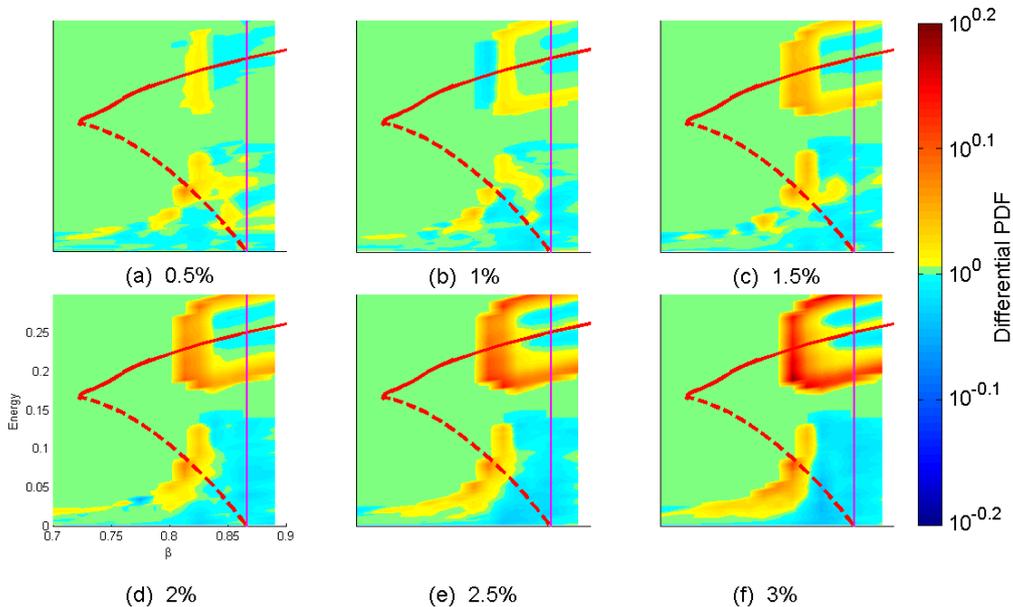


Figure 4.16: Differential Probability Density Functions (PDFs) of system energy with increasing multiplicative velocity ($u_{mN}$) noise strength, relative to the PDFs of the system under additive noise of amplitude 1% alone, against the bifurcation diagram of the noiseless system. Similar to Figure 4.13 but for multiplicative noise.
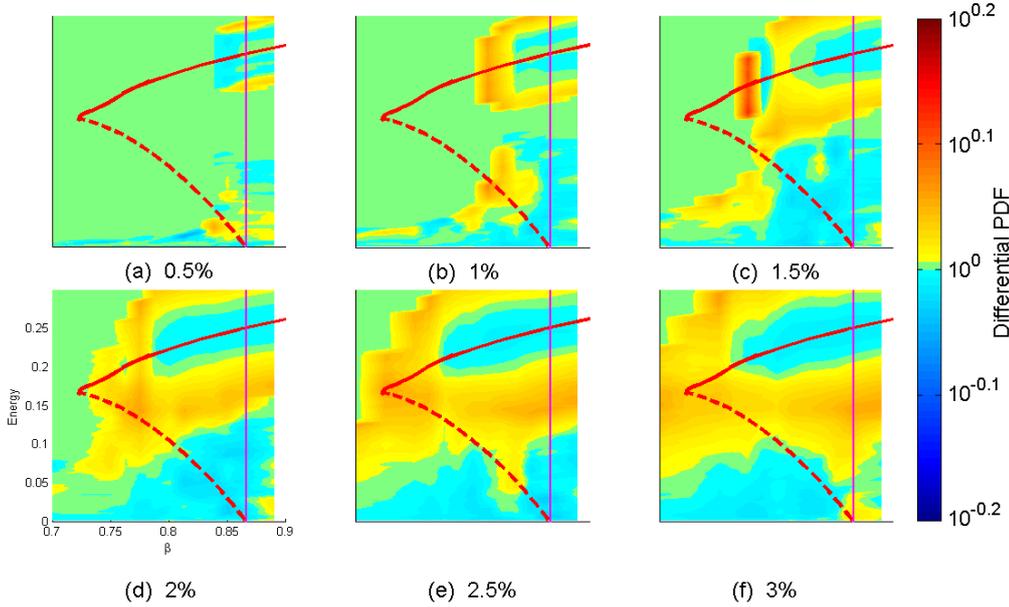
Figure 4.17: Differential Probability Density Functions (PDFs) of system energy with increasing noise strength of the combined noise signals, relative to the PDFs of the system under additive noise of the same amplitude, against the bifurcation diagram of the noiseless system. Similar to Figure 4.13 but for all noise signals at the same amplitudes, compared to the additive noise results for the equivalent amplitude.

### 4.3.4 Combined noise

In this section the additive, parametric and multiplicative noise terms are all applied simultaneously, each with the same noise strength. The differential PDF in each subfigure is compared to the equivalent noise strength with additive noise alone, not just to the 1% additive noise results as defined before in equation (4.6). Each of the six subfigures of Figure 4.17 is therefore compared with the equivalent subfigure in Figure 4.12.

The effect of combining the noise types can be seen to increase the probability of triggering, and reduce the probability of the system remaining around the stable fixed point, $E \approx 0$. As noise intensity increases there is a large increase in probability of the system transitioning between the stable limit cycle and the zero fixed point, indicated by the positive differential PDF in the bridge region.

## 4.4 Comparison against experimental combustor data

The triggering mechanism in this chapter was proposed for a simple theoretical thermoacoustic model, consisting of a simplified hot wire heat source and 1D duct. After the publication in 2011 of this mechanism and the methods of Section 4.2 [2], a thorough experimental study of
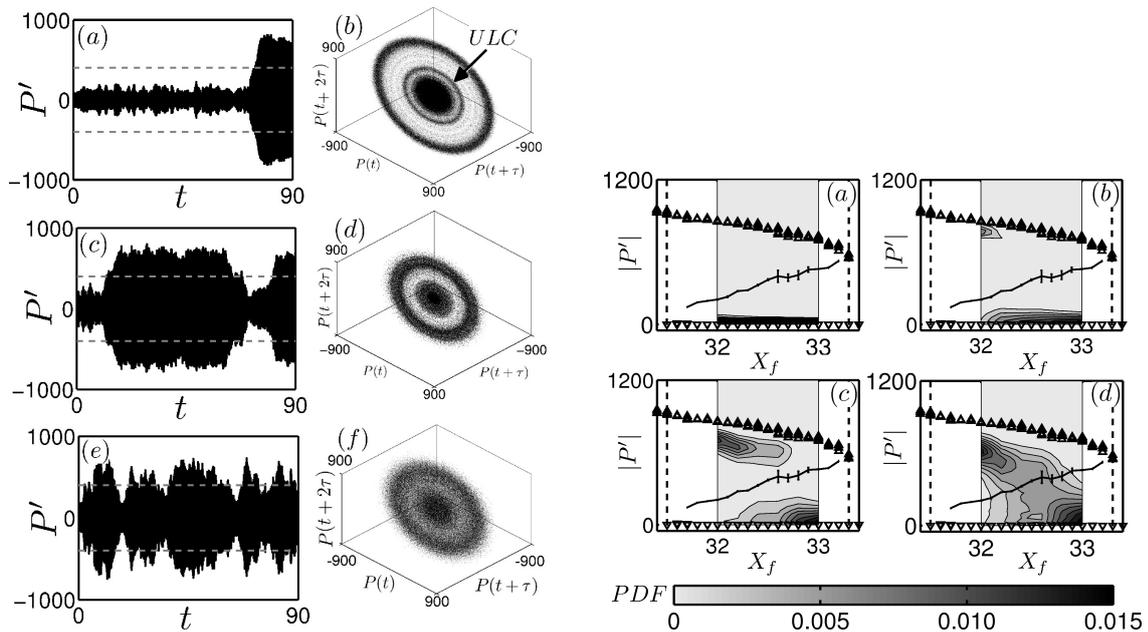
(a) The evolution of acoustic pressure and their corresponding phase portrait at $x_f = 0.36$ for noise levels of 38.8 Pa (a and b), 57.2 Pa (c and d) and 100 Pa (e and f). The time delay chosen for reconstructing the phase space is $\tau = 13$.

(b) Stochastic stability map for system for noise levels (a) 18.6 Pa, (b) 38.8 Pa, (c) 57.2 Pa and (d) 100 Pa.

Figure 4.18: Time traces and stochastic stability map for the experimental system of Jegadeesan and Sujith, under forcing with different amplitudes of quasi-white noise. The system undergoes the same triggering mechanism as the theoretical Rijke tube model; for low amplitudes of noise the stable limit cycle is reached via the unstable limit cycle (a(a), b(b-c)), for large amplitudes of noise the system changes between the stable limit cycle and stable fixed point (a(c)) and therefore the bridging phenomenon is seen in the stochastic stability map (b(d)). Both figures are reproduced from Ref. [65] with kind permission of the authors.

these concepts was completed in 2012 by Jegadeesan and Sujith [65, 66] at IIT Madras. With their kind permission, their results are reproduced in this section to demonstrate the similarity between the results from the simple Rijke tube model and the results from an experimental combustor.

The experimental combustor of Jegadeesan and Sujith consists of a non-premixed flame (diluted methane/oxygen) in a cylindrical duct. The flame can be moved axially in the duct and this position is the parameter used in the bifurcation diagrams. The flame is driven by acoustic noise from a loudspeaker located in a plenum at the base of the fuel tube, and therefore the noise appears as a fluctuation in the fuel flow rate. Pressure measurements, chemiluminescence measurements and high speed photography are used to examine the dynamics of the flame.

The flame was observed to have a bistable region caused by a subcritical bifurcation, when the flame is located one third of the way along the duct. Harmonic forcing was used to estimate

the amplitude of the unstable limit cycle in this bistable region, referred to as the 'triggering amplitude' in some literature. At some operating conditions, noise induced triggering was probable even when the amplitude of the noise was less than 5% of the amplitude of the stable limit cycle.

Figure 4.18 shows sample time traces for the system when forced with increasing amplitudes of noise in the bistable operating region. The system undergoes the same triggering mechanism as the theoretical Rijke tube model; for low amplitudes of noise the stable limit cycle is reached via the unstable limit cycle (Figure 4.18 a(a), b(b-c)), for large amplitudes of noise the system changes between the stable limit cycle and stable fixed point (Figure 4.18 a(c)) and therefore the bridging phenomenon is seen in the stochastic stability map (Figure 4.18 b(d)).

## 4.5 Conclusions

Triggering is seen to be strongly dependent on the strength of additive noise and the parameter $\beta$, and the system is shown to be unstable below the linear stability limit when forced with low amplitude stochastic noise. As noise strength increases, the system becomes unstable further from the Hopf point, increasing the region where triggering may occur. As noise strength increases further it becomes possible for the system to unsettle itself from the stable limit cycle and return to the stable fixed point, via the unstable limit cycle. This results in short bursts of high energy oscillations.

Stochastic stability maps are introduced as a method of visualising the stability of a thermoacoustic system when forced by stochastic noise. Stochastic stability maps quantify the probability of the state of the system lying in a particular region, and highlight the bimodal PDFs in operating regions with potential for triggering. The stable oscillation amplitudes when the system is forced by low amplitude noise agree well with those predicted by the noiseless bifurcation diagram. If the level of noise in a real combustion system, such as a gas turbine, is known, then a stochastic stability map could be used to predict a region of safe operation. Alternatively, multiple experimental results could be used in a stochastic stability map to visualise the stability of the system under operating conditions. The stochastic stability maps and triggering mechanism of the Rijke tube model are seen to have a very similar form to those of an experimental combustor.

Parametric and multiplicative noise are also shown to affect the probability of the system triggering. Non-dimensional wire-temperature noise ($\beta_N$) and multiplicative velocity noise ($u_{mN}$) are seen to have a larger effect than parametric noise in the time delay ($\tau_N$) and damping ($\zeta_{jN}$). Noise terms that excite the energy of the system, such as the velocity additive

noise in section 4.2.2 and the external noise in section 4.3.1, are demonstrated to be the most dangerous for triggering.

In a real gas turbine combustor there will always be a combination of additive, parametric and multiplicative noise, from sources such as: additive noise from incoming flow fields, turbulence in the compressor and vibration of the chassis, parametric noise from turbulence and vibration in the fuel pipelines, multiplicative noise from the boundary layers and flow-combustion interaction in the combustion chamber. As such, the combustor is continuously forced by non-negligible noise sources and stability of the engine should be measured in terms of practical stability rather than linear stability.

This chapter demonstrates that small amplitudes of noise can cause thermoacoustic systems to transition between different attractors. In the case of the hot wire Rijke tube model there are only two types of attractors: fixed points and limit cycles. In thermoacoustic systems with flame models, such as that presented in chapter 7, there are often many other types of attractors, such as period-$2^n$, quasiperiodic or chaotic oscillations, and there may be many stable attractors and many unstable attractors that exist at the same operating condition [79]. In these cases noise induced transitions become more complicated, because the system may evolve via more than one unstable attractor before a stable attractor is reached. The stochastic methods of this chapter are not applied to the more complicated thermoacoustic models in the latter half of this thesis, but qualitatively similar results are expected.

# Part II

# Methods for finding limit cycles in large thermoacoustic systems

# Chapter 5

# Matrix-free methods and continuation analysis

In chapters 3 and 4 a triggering mechanism was presented, whereby a thermoacoustic system transitions from a stable fixed point to a stable limit cycle, via an intermediate unstable limit cycle. The eventual behaviour of the system is governed entirely by the location and form of the attractors involved. It is therefore useful to have efficient methods of finding the fixed points and limit cycles in a thermoacoustic system.

In the Rijke tube model, which has $\mathcal{O}(10)$ variables, the fixed point and limit cycle attractors were found using a continuation method suitable for delay differential equations [4]. Continuation methods have been used in the field of thermoacoustics to calculate limit cycles and sub-/supercritical bifurcations [16, 17, 18, 26], but only for small systems with $\mathcal{O}(10)$ variables. To investigate more complex thermoacoustic systems that include 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. For models of that size, most continuation methods are practical for finding fixed points, but are impractical for finding limit cycles. This chapter presents an efficient method for finding limit cycles of large thermoacoustic systems.

The content of this chapter is published in Ref. [80], co-authored with Simon Illingworth and Matthew Juniper.

## 5.1 Introduction

Continuation methods rely on the solution of a series of linear equations to find limit cycles by iteration. The exact solution of these linear equations, however, 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 [81, 82]. Matrix-free methods have recently been used to find cusp bifurcations [83] and limit cycles in thermal convection [35], with $\mathcal{O}(10^3)$ variables. Continuation of steady states has been performed for thermal convection [84] using the matrix-free algorithms of the LOCA package [5], of which some subroutines are used in this chapter. Limit cycles have also been extracted from turbulent Couette flow [85] with $\mathcal{O}(10^5)$ variables, and turbulent pipe flow [86] 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 [87] and ocean modelling [88], 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 [89] or Krylov-Schur [90] 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.

The aim of this chapter 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. By finding separate Hopf bifurcations it is possible to follow separate branches of limit cycles for systems that have multiple unstable frequencies. The chapter begins by introducing two shooting methods and by introducing the iterative techniques used to converge to the limit cycles. The chapter 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 demonstrated on a model of a ducted diffusion flame in chapter 6, and a ducted premixed flame in chapter 8.

## 5.2   Shooting methods

Continuation methods examine nonlinear systems whose evolution is governed by:

$$\frac{\mathrm{d}\underline{x}(t)}{\mathrm{d}t} = F(\underline{x}(t), \lambda), \quad \underline{x}(t) \in \mathbb{R}^N \tag{5.1}$$

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

Limit cycles satisfy:

$$\underline{x}(0) = \underline{x}(T), \quad \left\{ T \in \mathbb{R}^+ | T \neq 0 \right\}, \tag{5.2}$$

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 $\underline{x}(-\tau \leq t \leq 0) = \underline{x}(T - \tau \leq t \leq T)$.

Delays are often used in low order thermoacoustic systems to model a convective time delay, because these systems use simple polynomial expressions for the heat release at the flame [17, 18]. Polynomial expressions have no 'memory' because the heat release value is only dependent on the current velocity, so therefore a time delay in the velocity values must be used to model a convective delay. When a flame surface is modelled explicitly and has a space-dependent velocity field, however, then no time delay is required because the space-dependence of the velocity field is defined by the 'memory' of the convective process. Because this chapter is describing a continuation method that is suitable for studying flame-acoustic interaction, where the flame field is modelled explicitly, systems without delays will be considered for the rest of this chapter.

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, $||\underline{x}(T) - \underline{x}(0)|| < \epsilon, \quad \{\epsilon \in \mathbb{R}^+ | \epsilon \neq 0\}$, where $\epsilon$ is small.

It should be noted that it is possible to perform the shooting methods of this chapter with delay differential equations, with a small modification. First, the methods must iterate several ($n$) states, $\underline{x}(-n\tau/n), \ldots \underline{x}(-i\tau/n), \ldots \underline{x}(0), \ 0 \leq i \leq n$, rather than just iterating $\underline{x}(0)$. Second, to begin timemarching at time 0, a polynomial fit must be applied to these $n$ states to describe the system evolution between $-\tau < t < 0$, to use in the equations for evolution of the system between $0 < t < \tau$. Third, the residual of each of the $n$ states must be minimised, $||\underline{x}(T - i\tau/n) - \underline{x}(-i\tau/n)|| < \epsilon, \ 0 \leq i \leq n$, until a predefined level of convergence is reached.

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

---

[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.
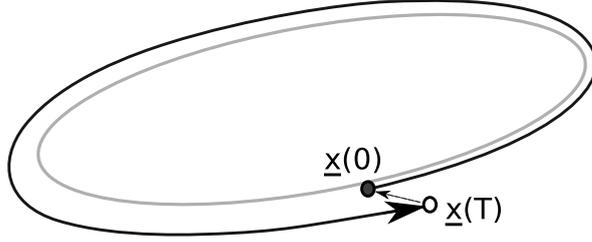
Figure 5.1: Standard shooting method to find a limit cycle. Given a current guess for a state on a limit cycle, $\underline{x}(0)$, we timemarch forward $T$ time units to $\underline{x}(T)$, where $T$ is our guess for the period. We then iterate our starting guess, $\underline{x}(0)$, to minimise the length of the residual vector, $\underline{x}(0) - \underline{x}(T)$, (dashed arrow).

## 5.2.1  Standard shooting

The standard shooting method used in this chapter 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. 5.1), 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 $[\underline{x}(0) - \underline{x}(T), \theta]$, where $\theta$ is a phase condition described later. 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 (5.3) shows the linear equation for the $n^{th}$ iteration, where $i$ and $j$ are the row and column indices of the matrix [91]. It has the standard form for multi-dimensional Newton iteration, $J\underline{\Delta x} = -\underline{r}$ .

$$\overbrace{\begin{bmatrix} \begin{array}{c|c} I - M & \underline{b} \\ \hline \underline{c} & d \end{array} \end{bmatrix}}^{(N+1) \times (N+1)} \overbrace{\begin{bmatrix} \underline{\Delta x} \\ \Delta T \end{bmatrix}}^{(N+1) \times 1} = - \overbrace{\begin{bmatrix} (\underline{x}(0) - \underline{x}(T))^n \\ \theta^n \end{bmatrix}}^{(N+1) \times 1} \tag{5.3}$$

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

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

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 ($\theta$) 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 5.3). 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 5.3, the characteristics of the system are contained in the monodromy matrix $M$, which relates a change in $\underline{x}(0)$ to a change in $\underline{x}(T)$. The eigenvalues of the monodromy matrix are called Floquet multipliers. In a dissipative system, such as in thermoacoustics, most of the Floquet multipliers 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 Floquet multipliers 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 (§6.2.3.2).

### 5.2.2 Poincaré shooting

In the Poincaré shooting method, the solution is constrained to lie on a hyperplane that is perpendicular (or nearly perpendicular) to the limit cycle [35]. 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 5.2). 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
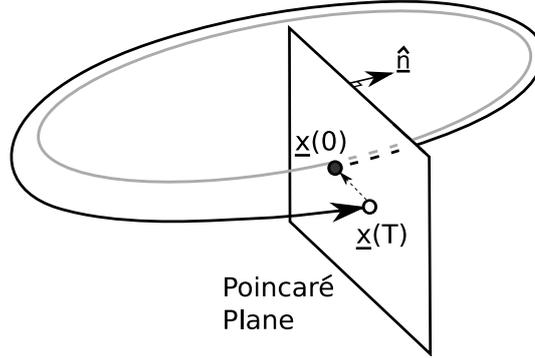
Figure 5.2: Poincaré shooting method to find a limit cycle. The starting state, $\underline{x}(0)$, and end state, $\underline{x}(T)$, are constrained to lie on a hyperplane perpendicular to the limit cycle. The system is timemarched until it crosses the Poincaré plane again, in the same direction as $\hat{n}$. The period $T$ is found as a byproduct.

parametrisation. One method of parametrisation is shown in Figure 5.3 for a 3 dimensional case [35]. In Figure 5.3a, 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 [35], 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 5.3b, 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 a $(N-1)$D 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 $R_k$ 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:

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

The iterative procedure is modified from equation (5.3), and now involves minimising the residual, $\bar{r}(\underline{x}(0), \lambda) = \bar{x}(0) - \bar{x}(T)$, with the following steps :

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



(b) 2D diagram of parametrisation of a plane.
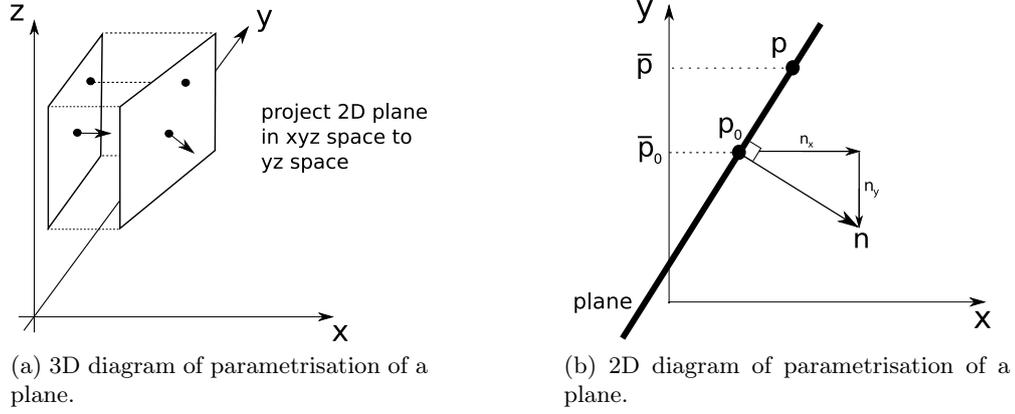
Figure 5.3: Parametrisation method, which takes an $(N-1)$ dimensional hyperplane with $N$ variables, and projects it onto a space with only $(N-1)$ variables. This is a quick method of reducing the number of variables, so that the Poincaré shooting method can iterate in $(N-1)$ dimensions.

$$
\overset{\text{(N-1)}\times\text{(N-1)}}{\begin{bmatrix} I - M \end{bmatrix}} \overset{\text{(N-1)}\times 1}{\begin{bmatrix} \underline{\Delta\bar{x}} \end{bmatrix}} = - \overset{\text{(N-1)}\times 1}{\begin{bmatrix} (\bar{x}(0) - \bar{x}(T))^n \end{bmatrix}} \tag{5.4}
$$

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

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 (5.3)), 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 5.4.1 and 6.2.4.1). 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 6.1, 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.

## 5.3  Matrix-free methods

As stated in section 5.2.1, 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.

### 5.3.1  GMRES

The matrix-free method used in this chapter to solve $J\underline{\Delta x} = -\underline{r}$ is the Generalised Minimal Residual method (GMRES) [92]. 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 [93], 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.

### 5.3.2    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 [82]. 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 (5.3), and can be written as:

$$J_{ij} = \frac{\partial\left(\underline{x}_i(0) - \underline{x}_i(T)\right)}{\partial \underline{x}_j(0)} \tag{5.5}$$

The matrix-vector product for arbitrary vector $\underline{v}$ can therefore be approximated by equation (5.6), where $\delta$ is small:

$$J\underline{v} = \underline{v} - \frac{A(\underline{x}(0) + \delta\underline{v}) - A(\underline{x}(0))}{\delta} + \mathcal{O}(\delta) \tag{5.6}$$

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 ($\theta$) components of the Jacobian matrix are also partial differentials ($\underline{b}, \underline{c}, d$ in equation (5.4)), 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 [5]. 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.

### 5.3.3    First variational matrix-vector products

The matrix-vector product can also be calculated by taking the first variational equations, sometimes referred to as the tangent linear equations, and integrating them in time. This approach is used in the Navier-Stokes continuation of Sanchez [35, 87]. The first variational

equations linearise about a trajectory, rather than a point. As in equation (5.1), 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:

$$J\underline{v} = \underline{v} - \underline{X}'(T) \tag{5.7}$$

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 [93]:

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

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 [94, 95]. 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 6.2.3.2, for the test model of a 2D diffusion flame in an acoustic duct.

## 5.4  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.

### 5.4.1 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, $JP^{-1}(P\underline{\Delta x}) = -\underline{r}$. 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, multi-grid 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 [87, 96, 97] (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 5.2.1). 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 (5.3)). 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 5.2.1). The eigenvalues can be found in a matrix-free fashion with the Arnoldi algorithm or the Krylov-Schur algorithm of Stewart [90].

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.

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

From the construction of the partial-Schur decomposition, the inverse must exist, $J^{-1}U_k = U_k S_k^{-1}$, 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, $JV_k = V_k H_k$, where $H_k$ is a $k$ dimensional upper Hessenberg matrix, and further taking a Schur decomposition, $H_k Q_k = Q_k S_k$, giving $JU_k = U_k S_k$ where $U_k = V_k Q_k$. 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.

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

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 [91, 98]. 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 [35].

### 5.4.2 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[1]. 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 chapter, the adaptive step size routine from the LOCA framework is used [5]. 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 [5]:

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

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

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

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 [35].

### 5.4.3 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

---

[1]In the literature, there are several globalisation strategies that can be used in tandem with Newton methods to increase the zone of convergence [99].

less chance of divergence. Alternatively, larger steps can be taken for the same magnitude of initial residual, reducing the number of points required 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, however, the prediction will break down around fold bifurcations, because the polynomial is single valued. For this reason, the arclength will be used in this chapter.

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, $S_k, S_{k-1}, \ldots S_1$, and the $i^{th}$ variable from the last $k$ solutions, $\underline{x}_{i,k}, \underline{x}_{i,k-1}, \ldots, \underline{x}_{i,1}$ to find the polynomial coefficients for that variable, $\underline{p}_{i,n}, \underline{p}_{i,n-1}, \ldots, \underline{p}_{i,1}$. We form a linear equation for each variable in the state:

$$
\overset{k \, \text{x} \, (n+1)}{\begin{bmatrix} S_k^n & \cdots & S_k^2 & S_k & 1 \\ S_{k-1}^n & \cdots & S_{k-1}^2 & S_{k-1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ & & & & \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ S_1^n & \cdots & S_1^2 & S_1 & 1 \end{bmatrix}} \overset{(n+1) \, \text{x} \, 1}{\begin{bmatrix} \underline{p}_{i,n} \\ \vdots \\ \underline{p}_{i,2} \\ \underline{p}_{i,1} \\ \underline{p}_{i,0} \end{bmatrix}} = \overset{k \, \text{x} \, 1}{\begin{bmatrix} \underline{x}_{i,k} \\ \underline{x}_{i,k-1} \\ \vdots \\ \\ \vdots \\ \underline{x}_{i,1} \end{bmatrix}}
\tag{5.8}
$$

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 (5.8). 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 $\sigma$ is the standard deviation of $S$. The order of the prediction technique that is most effective will be problem specific (§5.4.3).

## 5.5   Software implementation

The methods above are implemented in a continuation code in C++, as a supplementary package to the Trilinos framework written by Sandia National Labs [6]. The package uses the GMRES solver from the NOX package, the parallelisable vectors from the Epetra package, some subroutines from the LOCA package [5] and the Krylov-Schur eigensolver from the Anasazi package [100]. 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.

To analyse a thermoacoustic model in the continuation software, the parameters of the system to be varied must be specified, and routines must be supplied to return $d\underline{x}/dt$ given $\underline{x}$, to initiate the system, to supply a phase condition and to print solutions once converged. Timemarching can then be achieved using fixed-timestep Runge-Kutta 4 or variable-timestep Runge-Kutta-Fehlberg 45 routines, or through a user specified routine.

The software package is organised so the user need only specify the model specific code; all the analysis specific options are then specified in a top-level list to minimise the code required to analyse a new model. An object-oriented approach is used to switch simply between fixed point, standard shooting, Poincaré shooting, multiple shooting and Hopf bifurcation tracking algorithms. Similarly, there are options for fixed parameter or pseudo-arclength continuation, adaptive stepsizing, higher order prediction and invariant subspace preconditioning. Automated routines are included in the software package to find a Hopf bifurcation and transition to shooting on a branch of limit cycles, or to transfer between single and multiple shooting. The framework can accept state vectors that are distributed across parallel machines, and in the case of multiple shooting, sections of the limit cycle that are distributed on parallel machines.

A full description of the Jacobian and residual formation for the different continuation algorithms supported by the software is given in section 5.5.1. The Hopf bifurcation tracking routine is basic and requires a full matrix decomposition. It is effective for systems with $\mathcal{O}(10^2)$ , but would not be suitable for much larger systems. Matrix-free methods for finding

Hopf bifurcations are available in the literature [101, 102] and as part of the LOCA package [103].

Tables 5.1, 5.2 and 5.3 detail the computational formulation for the continuation of steady states and shooting of limit cycles. Switching between continuation types is achieved in the code by changing one input string, to minimise the amount of code that has to be re-written whilst generating bifurcation diagrams. All bordering of the state is done automatically, i.e. changes in the length of the state to incorporate extra variables such as $T$ and $\lambda$. Although Poincaré shooting does not explicitly require the period in the state, it is included for ease of switching between shooting types. The part of the Jacobian matrix multiplying the period acts as the identity, and the residual is zero, which prevents the period from participating in the Newton solution. This method is also used for the projection variable in the Poincaré shooting method, and the parameter variable in fixed parameter continuation. The computational formulations shown are for the finite difference matrix-free approach; for the first variational matrix-free approach slightly different formulations would be required for the Hopf and Poincaré shooting algorithms.

### 5.5.1 Computational formulation

In Tables 5.1-5.3 the following notation is used:

- $\frac{dx}{dt} = F(\underline{x}, \lambda)$

- $\theta(\underline{x})$ is the phase condition

- $\underline{\Lambda}$ is the current normalised solution tangent, found with a secant method on the last two solution vectors, with weighting factors for the parameter.

- $\underline{x}_b$ if the bordered state

- $\Delta S$ is the current pseudo-arclength step

- $\sigma$ is the real part of the eigenpair with the highest growth rate. This is found with a full eigendecomposition of the Jacobian matrix, which would not be suitable for larger systems.

- A superscript of $T$ denotes a transpose vector.

- $T_p$ represents the time taken for the trajectory to cross the plane.

| Continuation Type | State | Fixed Parameter | | Pseudo-Arclength | | M-F |
|---|---|---|---|---|---|---|
| | | Residual | Jacobian | Residual | Jacobian | |
| Fixed Point | $\begin{bmatrix} \underline{x} \\ \lambda \end{bmatrix} {\scriptstyle (N+1)}$ | $\begin{bmatrix} \frac{d\underline{x}}{dt} \\ 0 \end{bmatrix} {\scriptstyle (N+1)}$ | $J = \begin{bmatrix} M & 0 \\ 0 & 1 \end{bmatrix}$ $M_{ij} = \frac{\partial F_i(\underline{x},\lambda)}{\partial \underline{x}_j}$ | $\begin{bmatrix} \frac{d\underline{x}}{dt} \\ \underline{x}_b \cdot \underline{\Lambda} - \Delta S \end{bmatrix} {\scriptstyle (N+1)}$ | $J = \begin{bmatrix} M & \underline{b} \\ \underline{\Lambda}_x^T & \underline{\Lambda}_\lambda \end{bmatrix}$ $M_{ij} = \frac{\partial F_i(\underline{x},\lambda)}{\partial \underline{x}_j}$ $b_i = \frac{\partial F_i(\underline{x},\lambda)}{\partial \lambda}$ | Yes |
| Hopf Bifurcation | $\begin{bmatrix} \underline{x} \\ \lambda_1 \\ \lambda_2 \end{bmatrix} {\scriptstyle (N+2)}$ | $\begin{bmatrix} \frac{d\underline{x}}{dt} \\ \sigma \\ 0 \end{bmatrix} {\scriptstyle (N+2)}$ | $J = \begin{bmatrix} M & \underline{b} & 0 \\ \underline{c}^T & d & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $M_{ij} = \frac{\partial F_i(\underline{x},\lambda)}{\partial \underline{x}_j}$ $b_i = \frac{\partial F_i(\underline{x},\lambda)}{\partial \lambda_1}$ $\underline{c}_j^T = \frac{\partial \sigma}{\partial \underline{x}_j}$ $d = \frac{\partial \sigma}{\partial \lambda_1}$ | $\begin{bmatrix} \frac{d\underline{x}}{dt} \\ \sigma \\ \underline{x}_b \cdot \underline{\Lambda} - \Delta S \end{bmatrix} {\scriptstyle (N+2)}$ | $J = \begin{bmatrix} M & \underline{b} & \underline{e} \\ \underline{c}^T & d & f \\ \underline{\Lambda}_x^T & \underline{\Lambda}_{\lambda_1} & \underline{\Lambda}_{\lambda_2} \end{bmatrix}$ $M_{ij} = \frac{\partial F_i(\underline{x},\lambda)}{\partial \underline{x}_j}$ $b_i = \frac{\partial F_i(\underline{x},\lambda)}{\partial \lambda_1}$ $\underline{c}_j^T = \frac{\partial \sigma}{\partial \underline{x}_j}$ $d = \frac{\partial \sigma}{\partial \lambda_1}$ $\underline{e}_i = \frac{\partial F_i(\underline{x},\lambda)}{\partial \lambda_2}$ $f = \frac{\partial \sigma}{\partial \lambda_2}$ | No |

Table 5.1: Summary of the continuation types included in the continuation code. M-F denotes whether the algorithm is matrix-free.

| Continuation Type | State | Fixed Parameter | | Pseudo-Arclength | | M-F |
|---|---|---|---|---|---|---|
| | | Residual | Jacobian | Residual | Jacobian | |
| Standard Shooting | $\begin{bmatrix} x(0) \\ T \\ \lambda \end{bmatrix}^{(N+2)}$ | $\begin{bmatrix} x(0) - x(T) \\ \theta(x(0)) \\ 0 \end{bmatrix}^{(N+2)}$ | $J = \begin{bmatrix} I - M & b & 0 \\ c^T & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $M_{ij} = \dfrac{\partial x_i(T)}{\partial x_j(0)}$ $b_i = -\dfrac{\partial x_i(T)}{\partial T}$ $c_j^T = \dfrac{\partial \theta(x(0))}{\partial x_j(0)}$ | $\begin{bmatrix} x(0) - x(T) \\ \theta(x(0)) \\ x_b \cdot \Lambda - \Delta S \end{bmatrix}^{(N+2)}$ | $J = \begin{bmatrix} I - M & b & d \\ c^T & 0 & 0 \\ \Lambda_x^T & \Lambda_T & \Lambda_\lambda \end{bmatrix}$ $M_{ij} = \dfrac{\partial x_i(T)}{\partial x_j(0)}$ $b_i = -\dfrac{\partial x_i(T)}{\partial T}$ $c_j^T = \dfrac{\partial \theta(x(0))}{\partial x_j(0)}$ $d_i = -\dfrac{\partial x_i(T)}{\partial \lambda}$ | Yes |
| Poincaré Shooting (k is the index of the parametrisation variable) | $\begin{bmatrix} \bar{x}_1(0) \\ \cdots \\ \bar{x}_{k-1}(0) \\ 0 \\ \bar{x}_{k+1}(0) \\ \cdots \\ \bar{x}_N(0) \\ T \\ \lambda \end{bmatrix}^{(N+2)}$ | $\begin{bmatrix} \bar{x}_1(0) - \bar{x}_1(T_p) \\ \cdots \\ \bar{x}_{k-1}(0) - \bar{x}_{k-1}(T_p) \\ 0 \\ \bar{x}_{k+1}(0) - \bar{x}_{k+1}(T_p) \\ \cdots \\ \bar{x}_N(0) - \bar{x}_N(T_p) \\ 0 \end{bmatrix}^{(N+2)}$ | $J = \begin{bmatrix} I - M & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ $M_{ij} = \dfrac{\partial \bar{x}_i(T_p)}{\partial \bar{x}_j(0)}, \; i,j \neq k$ $M_{ik} = 0$ $M_{kj} = 0$ | $\begin{bmatrix} \bar{x}_1(0) - \bar{x}_1(T_p) \\ \cdots \\ \bar{x}_{k-1}(0) - \bar{x}_{k-1}(T_p) \\ 0 \\ \bar{x}_{k+1}(0) - \bar{x}_{k+1}(T_p) \\ \cdots \\ \bar{x}_N(0) - \bar{x}_N(T_p) \\ 0 \\ x_b \cdot \Lambda - \Delta S \end{bmatrix}$ | $J = \begin{bmatrix} I - M & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \Lambda_\lambda \end{bmatrix}$ $M_{ij} = \dfrac{\partial \bar{x}_i(T_p)}{\partial \bar{x}_j(0)}, \; i,j \neq k$ $M_{ik} = 0$ $M_{kj} = 0$ | Yes |

Table 5.2: Summary of the continuation types included in the continuation code. M-F denotes whether the algorithm is matrix-free.

| Continuation Type | State | Fixed Parameter | | | M-F |
|---|---|---|---|---|---|
| | | Residual | Jacobian | | |
| Multiple Standard Shooting ($n$ sections) | $$\begin{bmatrix} (nN+2) \\ \underline{x}^1(0) \\ \underline{x}^2(\tfrac{T}{n}) \\ \vdots \\ \underline{x}^n(\tfrac{(n-1)T}{n}) \\ T \\ \lambda \end{bmatrix}$$ | $$\begin{bmatrix} (nN+2) \\ \underline{x}^1(0)-\underline{x}^n(T) \\ \underline{x}^2(\tfrac{T}{n})-\underline{x}^1(\tfrac{T}{n}) \\ \vdots \\ \underline{x}^n(\tfrac{(n-1)T}{n})-\underline{x}^{n-1}(\tfrac{(n-1)T}{n}) \\ \theta(\underline{x}^1(0)) \\ 0 \end{bmatrix}$$ | $$J = \begin{bmatrix} I & 0 & \cdots & 0 & -M^n & \underline{b}^n & 0 \\ -M^1 & I & \ddots & & 0 & \underline{b}^1 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots & \cdots & \\ \vdots & & \ddots & 0 & \cdots & & \\ 0 & \cdots & 0 & -M^{n-1} & I & \underline{b}^{n-1} & 0 \\ \underline{c}^T & 0 & \cdots & \cdots & 0 & 0 & 0 \\ & & & & & 0 & 1 \end{bmatrix}$$ $$M_{ij}^k = \frac{\partial \underline{x}_i^k(\tfrac{T}{n})}{\partial \underline{x}_j^k(\tfrac{(k-1)T}{n})}, \qquad \underline{b}_i^k = -\frac{\partial \underline{x}_i^k(\tfrac{kT}{n})}{\partial T}$$ $$\underline{c}_j^T = \frac{\partial \theta(\underline{x}^1(0))}{\partial \underline{x}_j^1(0)}$$ | | Yes |

| | | Pseudo-Arclength | | | M-F |
|---|---|---|---|---|---|
| | | Residual | Jacobian | | |
| | | $$\begin{bmatrix} (nN+2) \\ \underline{x}^1(0)-\underline{x}^n(T) \\ \underline{x}^2(\tfrac{T}{n})-\underline{x}^1(\tfrac{T}{n}) \\ \vdots \\ \underline{x}^n(\tfrac{(n-1)T}{n})-\underline{x}^{n-1}(\tfrac{(n-1)T}{n}) \\ \theta(\underline{x}^1(0)) \\ \underline{x}_b \cdot \underline{\Delta} - \Delta S \end{bmatrix}$$ | $$J = \begin{bmatrix} I & 0 & \cdots & 0 & -M^n & \underline{b}^n & \underline{d}^n \\ -M^1 & I & \ddots & & 0 & \underline{b}^1 & \underline{d}^1 \\ 0 & \ddots & \ddots & \ddots & \vdots & \cdots & \\ \vdots & & \ddots & 0 & \cdots & & \\ 0 & \cdots & 0 & -M^{n-1} & I & \underline{b}^{n-1} & \underline{d}^{n-1} \\ \underline{c}^T & 0 & \cdots & \cdots & 0 & \Delta_T^T & 0 \\ \Delta_{\underline{x}^1}^T & \Delta_{\underline{x}^2}^T & \cdots & \Delta_{\underline{x}^n}^T & \Delta_T^T & \Delta_\lambda^T \end{bmatrix}$$ $$M_{ij}^k = \frac{\partial \underline{x}_i^k(\tfrac{T}{n})}{\partial \underline{x}_j^k(\tfrac{(k-1)T}{n})}, \qquad \underline{b}_i^k = -\frac{\partial \underline{x}_i^k(\tfrac{kT}{n})}{\partial T}, \qquad \underline{d}_i^k = -\frac{\partial \underline{x}_i^k(\tfrac{kT}{n})}{\partial \lambda}$$ $$\underline{c}_j^T = \frac{\partial \theta(\underline{x}^1(0))}{\partial \underline{x}_j^1(0)}$$ | | Yes |

Table 5.3: Summary of the continuation types included in the continuation code. M-F denotes whether the algorithm is matrix-free.

## 5.6 Conclusions

Continuation methods can track limit cycles as system parameters vary, and are therefore an important tool for calculating the safe operating region of a thermoacoustic system. Conventional continuation methods 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. They have therefore only been used to analyse a few reduced order models of thermoacoustic systems.

Matrix-free continuation methods use matrix-free iterative solvers to solve the underlying linear algebra inexactly. The matrix-free solvers do not require the monodromy matrix to be explicitely formed. They only require the action of the monodromy matrix upon a vector. For large systems, this represents a huge saving in computational time, because forming the monodromy matrix requires timemarching $N$ periods, where $N$ is the number of variables. A suitable matrix-free iterative solver is the Generalised Minimal Residual method (GMRES).

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 (see section 6.2.4.3).

A software package is written in C++ to perform matrix-free continuation of generic time dependent systems. The software package can find fixed points, Hopf bifurcations[1] and limit cycles. The continuation methods are demonstrated on a ducted diffusion flame model in chapter 6 and a ducted premixed flame model in chapter 7.

---

[1]Hopf bifurcations are not found using a matrix-free technique.

# Chapter 6

# Continuation analysis of a ducted diffusion flame

In the last chapter, a matrix-free method of finding limit cycles was presented. In this chapter, these methods will be applied to a model of a ducted diffusion flame to demonstrate the efficiency of the numerics. A bifurcation surface of limit cycles is generated over a parameter range to locally describe the 'safe operating region' of the flame model. Once the surface of limit cycles is found, the limit cycles and their Floquet multipliers can be used to examine the flame-acoustic interaction.

The content of this chapter is the subject of two papers: Ref. [80] and Ref. [3] (section 6.4). Both papers are co-authored with Matthew Juniper and also with Simon Illingworth, who developed the ducted diffusion flame model and generated the FDF results in section 6.4.

## 6.1   Ducted 2D diffusion flame model

The ducted diffusion flame model consists of a Burke-Schumann flame in a 2D flame domain coupled to a 1D acoustic duct. Earlier versions of this model were discretised using a Galerkin method in the flame domain [48, 104]. This version was discretised by Illingworth using a spectral method in the flame domain [3]. This improves its numerical accuracy, which is important for the application of continuation methods. In the 2D flame domain, 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

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

(b) 2D steady state $Z$ field.

Figure 6.1: 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.

far downstream, $\partial Z/\partial x|_{10} = 0$. The flame lies on the stoichiometric contour, $Z = Z_{st}$, and is assumed to be mixing controlled. The location of the flame is denoted $f^+$.

The heat release from the model is coupled to a simple linear acoustic model of a duct [17, 48, 49], 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[1] $\zeta$, the stoichiometric mixture fraction $Z_{st}$, the flame position in the duct $x_f$, the acoustic velocity at the flame $u_f$, the fuel pipe width $\alpha$, and the coupling parameter $\beta_T$, defined by $\beta_T = 2/(T_{inlet} + T_{adiabatic})$.

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} \tag{6.1}$$

$$\frac{\partial p}{\partial t} = -\frac{\partial u}{\partial x} - \zeta p + \frac{2\beta_T}{1 - Z_{st}} \delta\left(x - x_f\right) \times$$

$$\left(-\int \int_0^{y=f^+} \frac{\partial z}{\partial t} \, dy dx + u_f \int_0^{\bar{f}^+} \left(\bar{Z} - Z_{st}\right) dy\right) \tag{6.2}$$

$$\frac{\partial z}{\partial t} = -\bar{u}_f \frac{\partial z}{\partial x} + \frac{1}{Pe}\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right) z - u_f \frac{\partial \bar{Z}}{\partial x} - u_f \frac{\partial z}{\partial x} \tag{6.3}$$

Equation (6.1) is derived from the acoustic momentum equation [17, 48]. Equation (6.2) is derived from the acoustic energy equation [17, 48], where the last term corresponds to the heat release from the flame field (similar to Ref. [48, 104]). Equation (6.3) is derived from the 2D advection-diffusion equation [48, 104], with a constant velocity over the flame field, $u_f$.

---

[1]The acoustic damping is defined as $\zeta_j = c_1 j^2 + c_2\sqrt{j}$, where $j$ is the acoustic mode number [17].

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

$$\frac{\partial U'}{\partial t} = -\frac{\partial P'}{\partial x}$$

$$\frac{\partial P'}{\partial t} = -\frac{\partial U'}{\partial x} - \zeta P' + \frac{2\beta_T}{1 - Z_{st}} \delta(x - x_f) \times$$
$$\left( -\int \int_0^{f^+} \frac{\partial Z'}{\partial t} \, dy dx - \int_{f^+} \frac{\partial z}{\partial t} |\nabla Z|^{-1} Z' \, dl + U_f' \int_0^{\bar{f}^+} (\bar{Z} - Z_{st}) \, dy \right)$$

$$\frac{\partial Z'}{\partial t} = -\bar{u}_f \frac{\partial Z'}{\partial x} + \frac{1}{Pe} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) Z' - U_f' \frac{\partial \bar{Z}}{\partial x} - U_f' \frac{\partial z}{\partial x} - u_f \frac{\partial Z'}{\partial x}$$

A more complete description of the model is included in Ref. [3]. The results in this chapter are generated with a $30 \times 16$ 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.

## 6.2 Numerical results

### 6.2.1 Fixed point plane

Figure 6.2 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 $\beta_T$, which controls the extent to which unsteady combustion perturbs the acoustics (equation (6.2)). 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, 105].

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 $\beta_T$. 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.
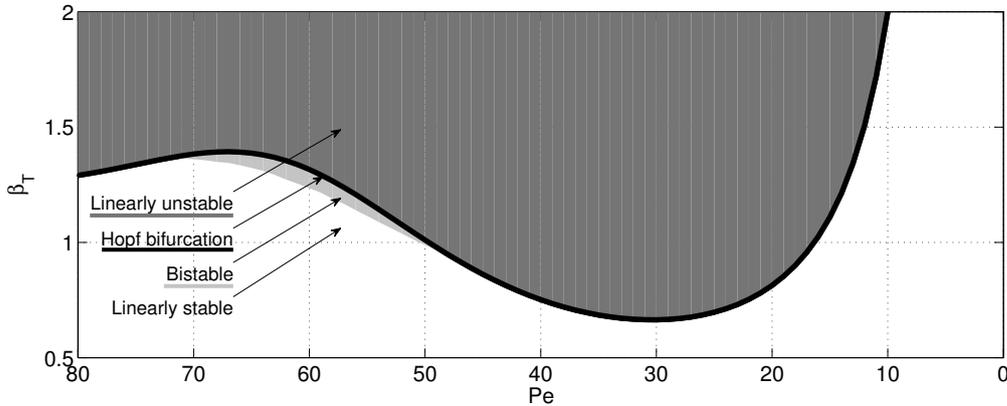
Figure 6.2: Stability of the fixed point solution, as a function of $Pe$ and $\beta_T$. Note that the $Pe$ axis is reversed to match the view in Figure 6.3.

### 6.2.2  Limit cycle surface

Figure 6.3 shows the limit cycle amplitudes as a function of $Pe$ and $\beta_T$, with the same fixed parameters as section 6.2.1. 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 6.3 is therefore Figure 6.2.

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. 6.3a) and supercritical bifurcations for $Pe < 50$, $Pe > 70$ (Fig. 6.3c). 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 surface of limit cycles is smooth and steep because the model is only weakly nonlinear. The weakly nonlinear behaviour of the model is discussed further in the conclusions of this chapter. A model with stronger nonlinearities would display a much more distorted surface (see Chapter 7).

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[1]. 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

---

[1]All computations run with an Intel i7-2600 3.4GHz processor.
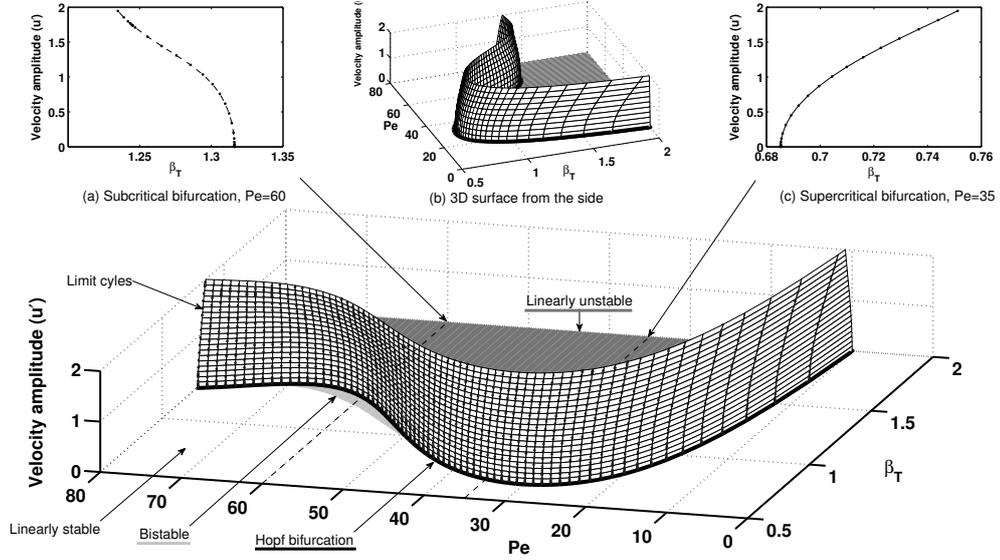
Figure 6.3: Continuation results for the ducted diffusion flame, as a function of $Pe$ and $\beta_T$. The thick dark line is the Hopf bifurcation, which is the same as the linear stability limit. The linearly unstable region is shaded dark gray and the bistable region is shaded light gray (the bistable 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). The limit cycles are unstable in (a) and stable in (c). Further physical analysis of these results is found in Ref. [3].

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.

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 [3].

### 6.2.3 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 6.2.3.2).

#### 6.2.3.1 Limit cycles

Figure 6.4 shows snapshots of the system during an unstable limit cycle, with Pe of 60, $\beta_T$ of 1.31 and velocity amplitude 0.62. When the velocity perturbation at the flame is positive, 6.4(a)-(c), the flame becomes longer and, near the inlet, the regions of high and low $z$ are
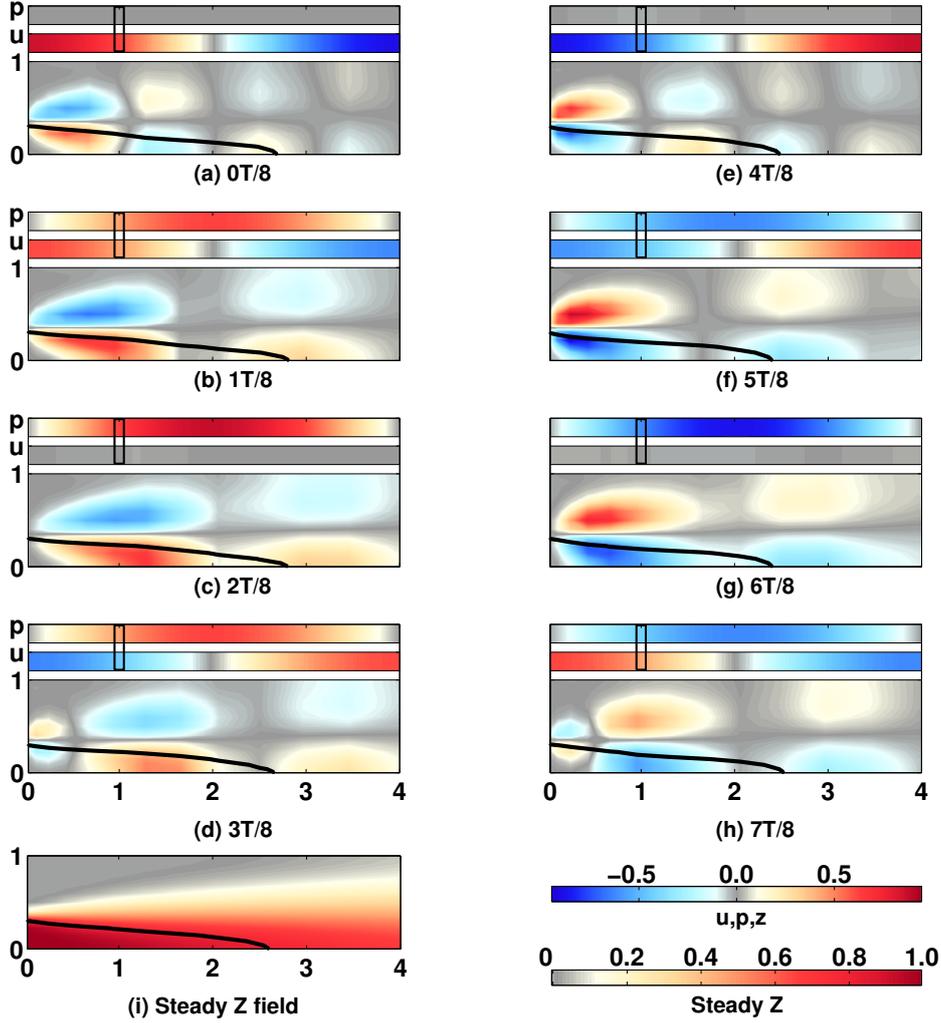
Figure 6.4: Snapshots of an unstable limit cycle (Pe=60, $\beta_T$=1.31), with velocity amplitude 0.62. In each snapshot (a-h), the top two bars show the perturbation pressures and velocities in the 1D duct ($x_a = 0 \rightarrow 1$), and the lower bar shows the 2D perturbation $z$ field and current flame 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.

stretched. When the velocity perturbation at the flame becomes negative, 6.4(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.
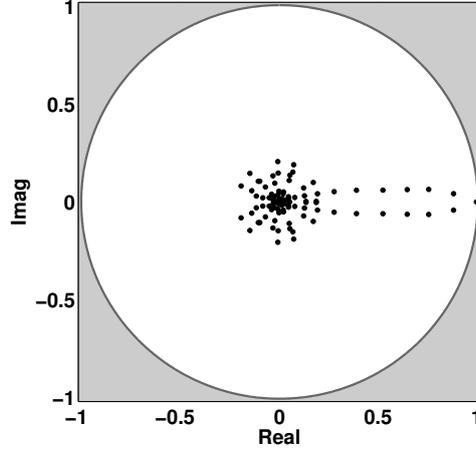
Figure 6.5: Floquet multipliers for an unstable limit cycle (leading multiplier at $1.000455 + 0i$). Any Floquet multipliers in the grey region correspond to unstable perturbations to the limit cycle.

### 6.2.3.2 Floquet multipliers

Figure 6.5 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 6.2.4.3.

Figure 6.6 shows the Floquet modes corresponding to the leading Floquet multipliers, which are located on the 'arm' extending to the right in Figure 6.5. These are the influential bulk motions of the system. In Figure 6.6 (a)-(g), the modes have large components in the $1^{st} - 7^{th}$ 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 6.6 (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.
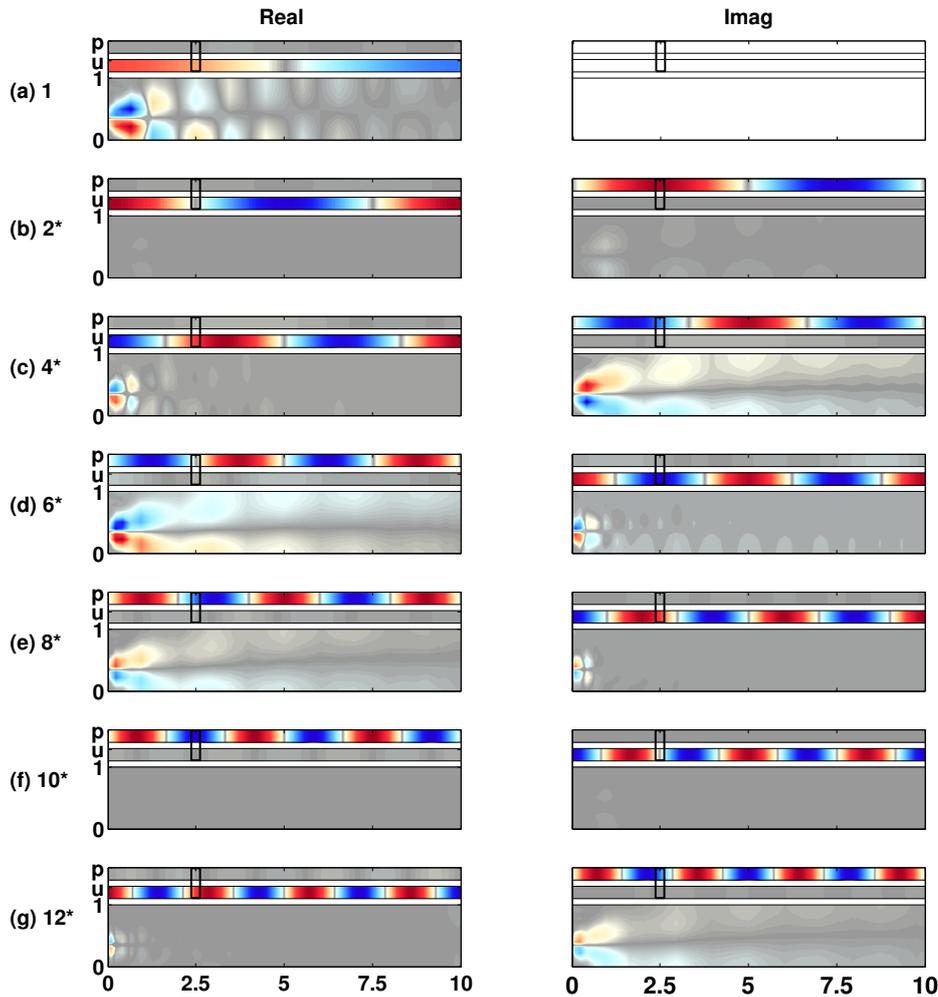
Figure 6.6: Leading Floquet modes for an unstable limit cycle (Pe=60, $\beta_T = 1.31$), corresponding to the influential bulk motions of the system. The modes are sorted by the absolute values of the Floquet multipliers, in decreasing order. A star indicates that the mode is one part of a complex conjugate pair. In (a)-(g), the modes have large components in the $1^{st} - 7^{th}$ acoustic modes respectively. The $z$ values are scaled by a factor of 35 to share the colourbar with the acoustic perturbations.

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 6.4). 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 [1, 2, 17].
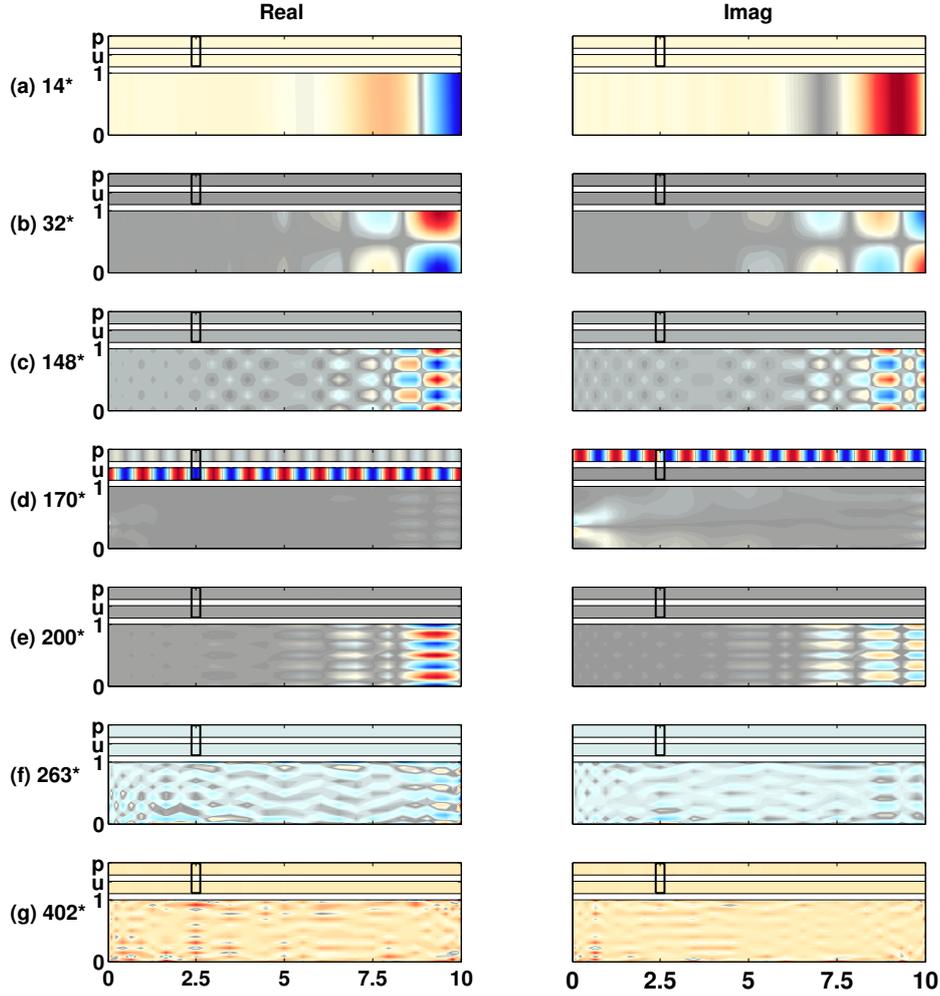
Figure 6.7: A selection of Floquet modes for an unstable limit cycle (Pe=60, $\beta_T = 1.31$), corresponding to typical dissipated motions of the system. The modes are sorted by the absolute values of the Floquet multipliers, in decreasing order. A star indicates that the mode is one part of a complex 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 $20^{\text{th}}$ acoustic mode, which is heavily damped. Above the $250^{\text{th}}$ 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.

Figure 6.7 shows a selection of typical Floquet modes corresponding to small Floquet multipliers, which are located around the origin in Figure 6.5. 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 6.7(a,b,c,e). In Figure 6.7(d) the mode corresponds to the $20^{\text{th}}$ acoustic mode, which is heavily damped. Above the $250^{\text{th}}$ Floquet mode, as in Figure 6.7(f,g), the modes correspond to small scale features in the $z$ field.

Sections 5.3.2 and 5.3.3 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 6.5, Table 6.1 shows the difference in the Floquet multipliers when evaluated with these two approaches (with $\delta = 10^{-8}$ in Eq. 5.6). The results are almost identical for the value of $\delta$ 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 chapter are therefore generated with the finite difference approach.

| 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 6.1: 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.

## 6.2.4 Efficiency of the numerics

### 6.2.4.1 Convergence to a limit cycle

Figure 6.8 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.

(a) Standard shooting - no preconditioning  (b) Poincaré shooting - no preconditioning
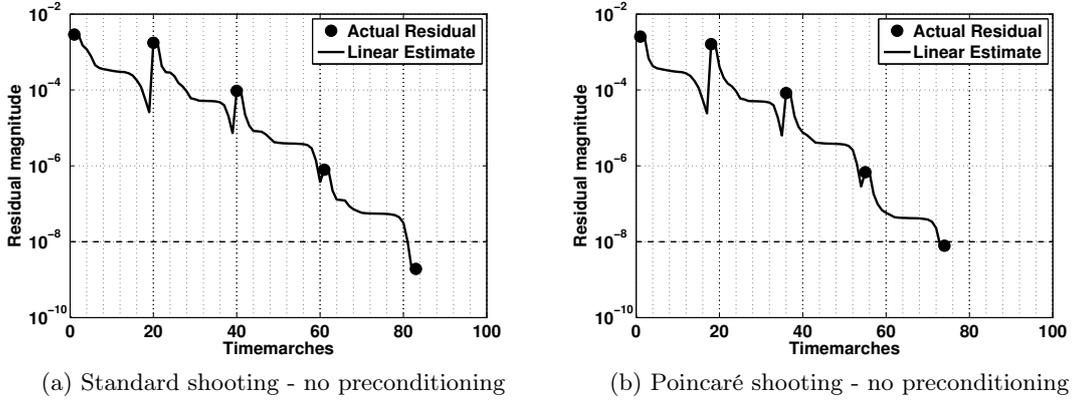
Figure 6.8: Convergence to a limit cycle in terms of the residual magnitude, $||\underline{x}(0)-\underline{x}(T)||_2$, and number of timemarches. The dots are the residual values after each Newton step and the line is the the estimated residual within the solution of each Newton step. As the solution converges, the estimated reduction in residual for each Newton step approaches the actual residual reduction achieved.

The 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 6.8, 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.

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 $\underline{\dot{x}}(0)$, 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.

### 6.2.4.2   Convergence to a limit cycle with preconditioning

Using the same starting condition as Figure 6.8, Figure 6.9 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 5.4.1). The eigenvalues are converged to a precision of $10^{-4}$.



(a) Standard shooting - with preconditioning       (b) Poincaré shooting - with preconditioning
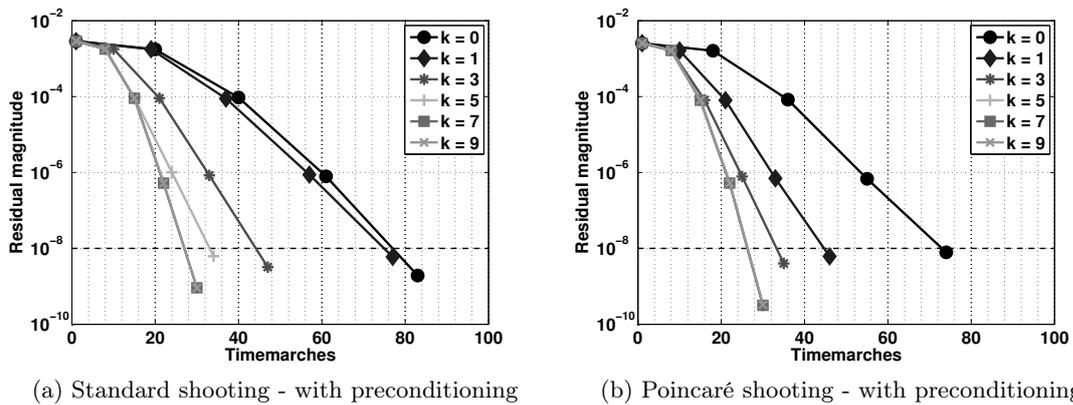
Figure 6.9: Convergence to a limit cycle in terms of the residual magnitude, $||\underline{x}(0)-\underline{x}(T)||_2$, and number of timemarches. Preconditioning is used to increase the convergence of the GMRES solver. Results are plotted for preconditioners with different numbers of deflated eigenvalues (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 5.2.2). 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 (5.3). The bordered form of the matrix can be used to define a preconditioner [106, 107], or to solve a linear equation in ordered steps [108]. For equation (5.3) 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 the literature and will not be covered further in this thesis.

Figure 6.10 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 Fig-
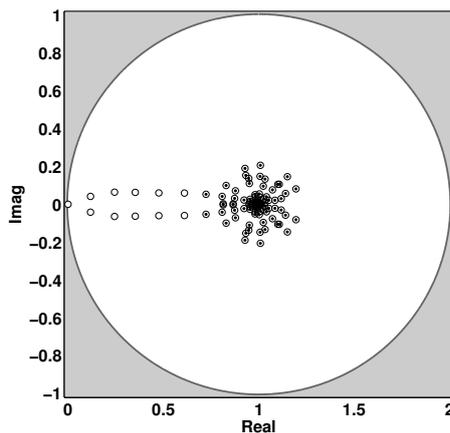


Figure 6.10: Demonstration of invariant subspace preconditioning, for an eigenspace of the 11 eigenvalues furthest from (+1,0). The eigenvalues of the Jacobian are shown as open circles and the eigenvalues of the preconditioned Jacobian are shown as black dots. The preconditioner takes the eigenvalues 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.

ure 6.10, 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 6.10 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 6.3). 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.

### 6.2.4.3 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 6.11 shows the Ritz values and relative residuals as the dimension of the Krylov subspace (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 $\frac{||\underline{r}+J\underline{\Delta x_k}||}{||\underline{r}||}$.

(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
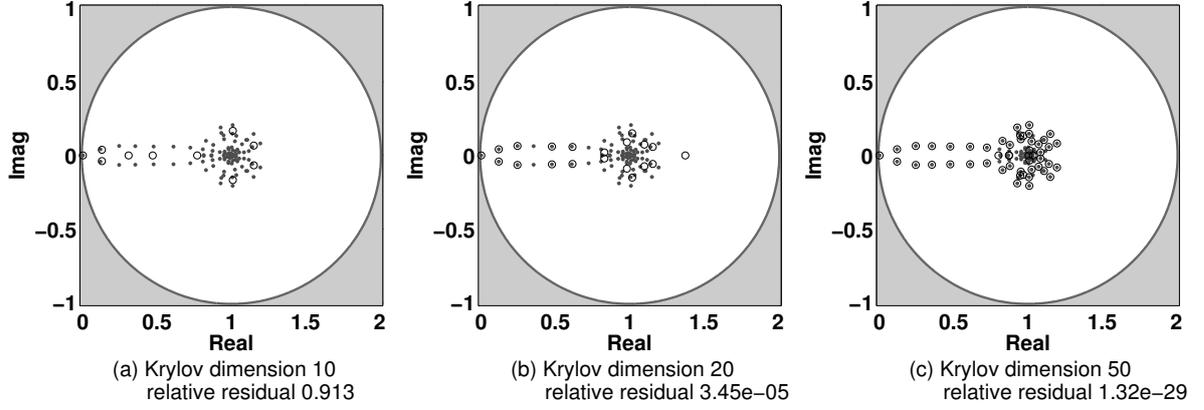
Figure 6.11: Convergence of the Ritz values within the GMRES solver and the convergence of the relative residual, as the dimension of the Krylov subspace increases. The dimension of the Krylov subspace is the same as the number of timemarches. The Ritz values (open circles) are the estimated eigenvalues 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.

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, $\underline{v}$, $J\underline{v}$, $J^2\underline{v}, \dots, J^n\underline{v}$, 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
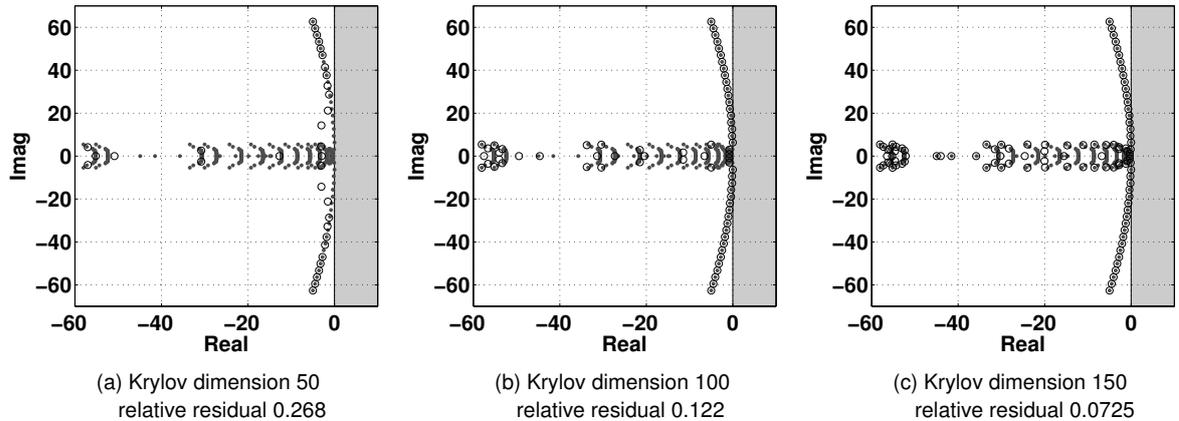
Figure 6.12: Convergence of the GMRES solver for a fixed point (as in Figure 6.11, which was for a limit cycle). The dimension of the Krylov subspace is the same as the number of gradient evaluations required ($\underline{\dot{x}} = F(\underline{x}, \lambda)$).

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 6.11, Figure 6.12 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.

### 6.2.4.4   Effect of higher order prediction
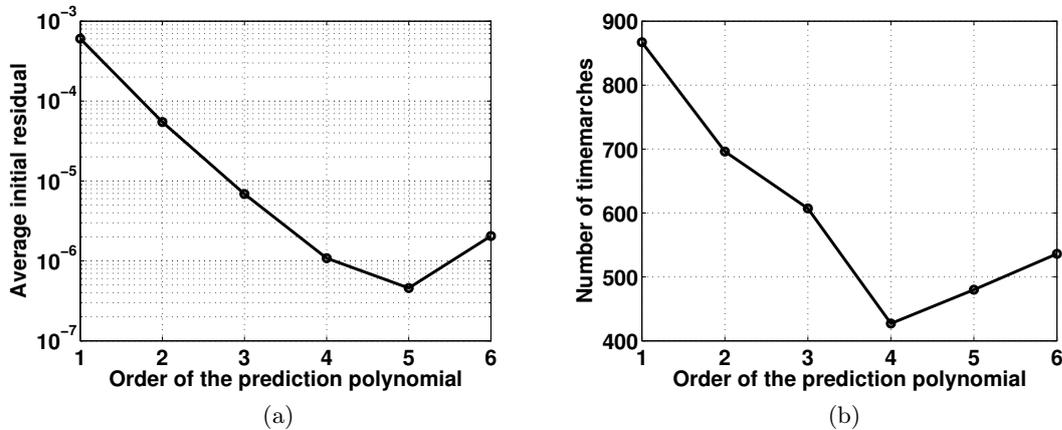


Figure 6.13: The effect of the order of the prediction polynomial on: (a) the average initial residual of predicted states, and (b) the total number of timemarches required to find twenty limit cycles.

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 6.13, 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 5.4.3). Figure 6.13a 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 6.13b 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.

## 6.3 Multiple shooting



(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.
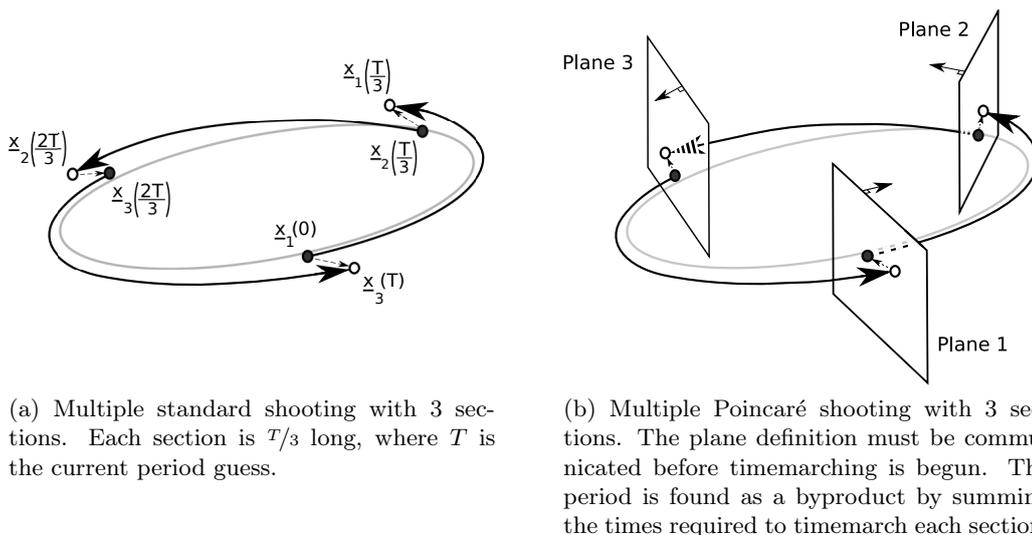
Figure 6.14: Multiple shooting methods, in which the limit cycle is split into $n$ sections that are timemarched separately. There are $n$ starting states (black dots), which are iterated to minimise the combined norms of the $n$ residual vectors (dashed arrows).

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 (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 [87]. 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 6.14).

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 [87]. 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 6.15.



(a) 1 Section          (b) 2 Sections          (c) 4 Sections
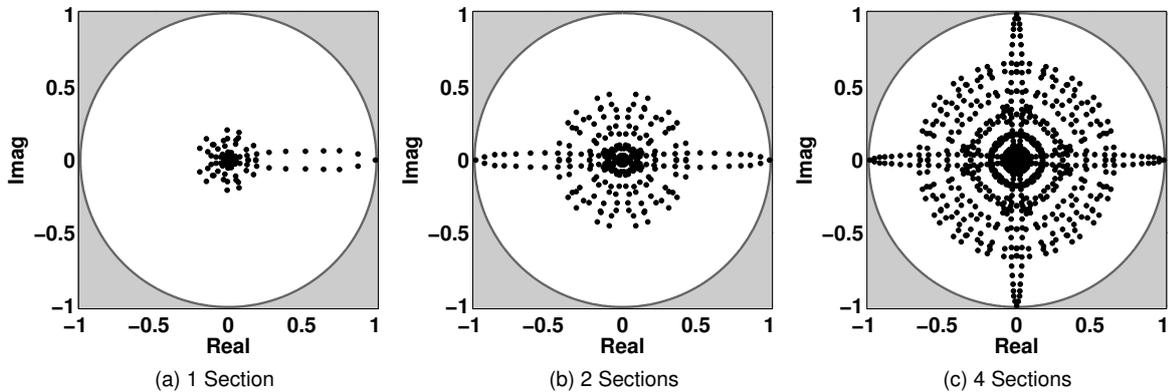
Figure 6.15: Eigenvalues of the monodromy matrix for the multiple Poincaré shooting formulation, for 1, 2 and 4 sections. For a multiple shooting system with $n$ sections, the eigenvalues are the $n^{th}$ roots of the Floquet multipliers.

Without preconditioning, the advantage and the disadvantage of multiple shooting almost cancel each other out [87], 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 5.4.1) 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 [87], but the overall process, including preconditioner formation, can approach a linear speed increase with $n$ [87].
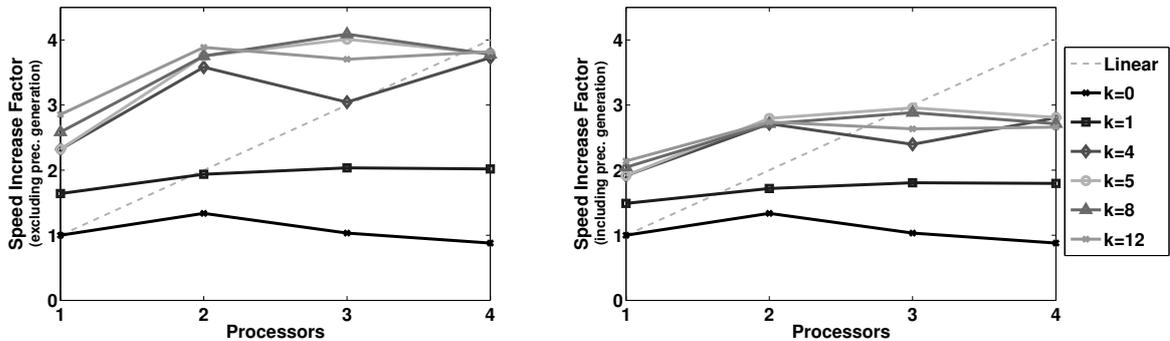


Figure 6.16: Speed increase for the multiple Poincaré shooting formulation against number of processors, measured over 15 limit cycles with fixed parameter continuation. The speed increase factor is normalised by the speed of the continuation with one processor and no preconditioning. The lines show 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).

Figure 6.16 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 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 6.16). As in Figure 6.15, 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 6.2.4.2). 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 6.16(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.

## 6.4 Comparison between the continuation method and the flame describing function

In this section, the limit cycles found by the continuation method are compared with the limit cycles predicted by the Flame Describing Function (FDF). The FDF uses a gain and a phase to describe the relationship between an acoustic perturbation and a heat release perturbation. The gain and phase are both dependent on the frequency and amplitude of the acoustic perturbation. The main limitation with the FDF is the assumption that the acoustic velocity and pressure signals are sinusoidal. In reality, if the heat release signal has higher harmonics of the forcing frequency then the acoustics will respond at those higher harmonics, and the acoustic velocity and pressure signals are no longer sinusoidal - the original assumption is broken. This assumption can therefore lead to inaccuracy in the predicted limit cycle amplitudes if the flame responds strongly at frequencies other than the forcing frequency.

Figure 6.17 shows a comparison between the limit cycle amplitudes predicted by the FDF method and those calculated by the continuation method [3]. The results from the FDF method, calculated by Illingworth, match excellently with those calculated by the continuation method. This is because the flame responds weakly at frequencies other than the forcing frequency, and therefore the approximations made in forming the FDF - namely that higher harmonics of the flame's response may be neglected - are valid for the diffusion flame model under investigation [3].

Figure 6.17: Comparison between limit cycle amplitudes predicted by the FDF method (dots) and the continuation method (lines), for (a) the subcritical case ($Pe = 60$) and for (b) the supercritical case ($Pe = 35$). These two cases are highlighted in Figure 6.3(a,c).

The FDF method was also used by Illingworth to demonstrate that the diffusion flame model under investigation has particularly weak nonlinearities, because the gain and phase of the FDF were only weakly amplitude dependent [3]. This explains the steep increase in amplitude of the limit cycle surface around the Hopf bifurcation in Figure 6.3, and it explains why the stable limit cycles in the bistable region have such large amplitudes (not shown in Figure 6.3). The weak amplitude dependence probably results from assuming a uniform velocity in the flame field.

When generating a bifurcation surface the computational 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 the bifurcation surface of Figure 6.3, for example, one parameter sets the flame operating condition ($Pe$), and the other sets the acoustic operating condition ($\beta_T$), so one FDF evaluation is required per 2D slice. Using this particular diffusion flame model, one FDF evaluation takes roughly 16 CPU hours [3], so the FDF would require roughly 1120 CPU hours to recreate Figure 6.3, which is 18 times slower.

## 6.5 Non-normality in the ducted diffusion flame model

The importance of non-normality in causing triggering was described in chapters 3 and 4. In particular, non-normality of the linear operator around a fixed point leads to transient growth of small perturbations. If the perturbations grow enough, the system may then be attracted towards an unstable attractor. In the Rijke tube model, this mechanism was shown to lead to triggering to a stable limit cycle. Furthermore, the non-normality of the monodromy matrix of the unstable limit cycle has been shown to be influential in defining the magnitude of the smallest perturbations that lead to triggering [109].

In this section the non-normality of the diffusion flame model is examined. The fixed point and unstable limit cycle are taken from the bistable operating condition at Pe of 60, $\beta_T$ of 1.31 .

### 6.5.1 Fixed point

Figure 6.18 shows the pseudospectra (red) and numerical range (blue) of the stable fixed point in the bistable region at Pe of 60, $\beta_T$ of 1.31. The fixed point is stable because every eigenvalue has a negative real part. Some transient energy growth can be expected at small times, however, because the numerical range extends into the unstable half-plane (shaded grey). The pseudospectra show that the diffusion flame system is only slightly non-normal, because the contours form almost regular circles around most of the eigenvalues and the contours of the pseudospectra do not extend far into the unstable half-plane. This is in contrast with the results of Balasubramian et al. [48], where the pseudospectra of the same diffusion model do extend far into the unstable half-plane. Figure 6.18 suggests that the degree of transient growth reported by Balasubramian et al. [48] - which has been heavily repeated in the literature - is incorrect, which has been demonstrated since by the detailed study of Magri [110].

(a)                                    (b)
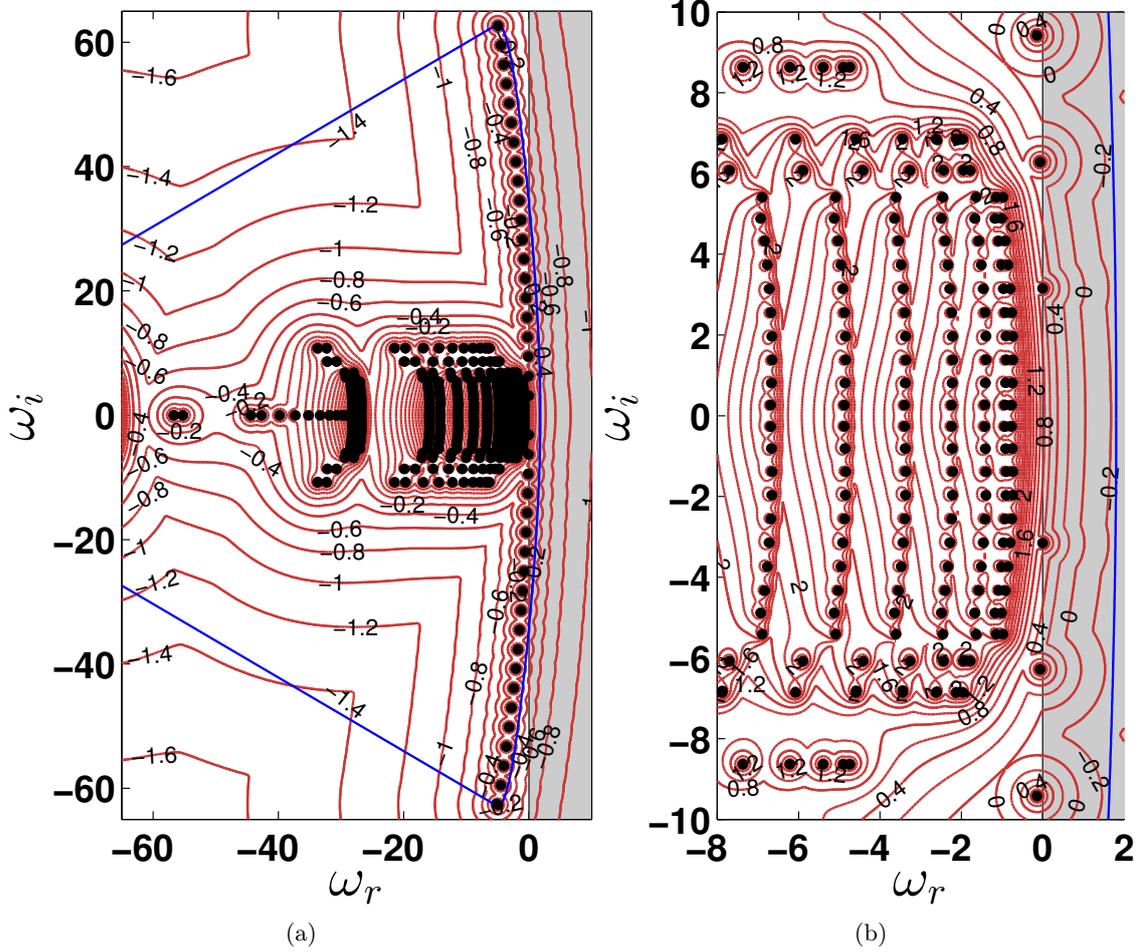
Figure 6.18: Pseudospectra (red) and numerical range (blue) of the Jacobian for the stable fixed point at Pe of 60, $\beta_T$ of 1.31, with a zoomed section around the real axis in (b).

### 6.5.2 Monodromy matrix

Figure 6.19 shows the psuedospectra (red) and numerical range (blue) of the monodromy matrix for the unstable limit cycle at Pe of 60, $\beta_T$ of 1.31 and velocity amplitude of 0.62. The psuedospectra show that the monodromy matrix is only slightly non-normal, similar to the results for the fixed point in the previous section. It is therefore suggested that the diffusion flame model is both weakly non-normal and weakly non-linear. It is likely that if some of the unphysical assumptions in the model were relaxed, the model would become both more strongly non-linear and more strongly non-normal. Some of the unphysical assumptions that could be relaxed are: the uniform temperature in the acoustic duct, the uniform velocity in the $z$ field and the compact flame assumption. In hydrodynamics, non-normality is often a result

97
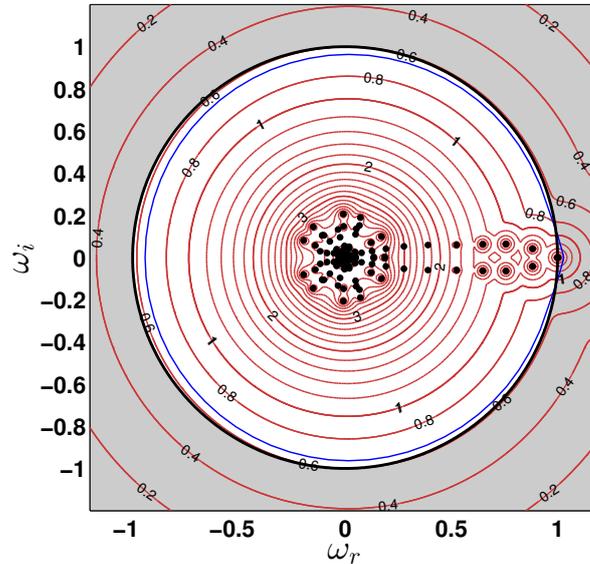
Figure 6.19: Pseudospectra (red) and numerical range (blue) of the monodromy matrix for the unstable limit cycle at Pe of 60, $\beta_\mathrm{T}$ of 1.31 and velocity amplitude of 0.62.

of strong convection in the flow, which would be introduced into this model if the velocity in the $z$ field were space-dependent.

## 6.6 Conclusions

This chapter 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 and use the fully coupled system. Continuation methods therefore do not assume that the velocity fluctuations and flame response are harmonic, whereas the FDF approach does, because it operates in the frequency domain.

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 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 the safe operating region to general changes in acoustic or flame operating conditions.

The ducted diffusion flame model is shown to be only weakly non-normal, around both fixed points and limit cycles. This disagrees with the result in Ref. [48], where the ducted diffusion flame was suggested to be strongly non-normal around fixed points.

# Chapter 7

# Continuation analysis of a ducted premixed flame

Continuation methods were used in chapter 6 to analyse a model of a ducted diffusion flame. The methods were shown to be efficient at calculating limit cycles whilst varying system parameters, and these limit cycles were plotted as a bifurcation surface. The diffusion flame model was found to be only weakly nonlinear, however, so the bifurcation surface was relatively smooth, and the model did not exhibit period-2, quasiperiodic or chaotic behaviour.

In this chapter continuation methods are used to analyse a model of a ducted premixed flame. The heat release of premixed flames is generally a more nonlinear function of velocity than the heat release of diffusion flames. Premixed flames have been shown to exhibit periodic, period-2, quasiperiodic and chaotic behaviour in both experiments [20, 21, 22, 23] and numerical simulations [28, 79, 111]. Bifurcation surfaces are generated over a parameter range and the limit cycles analysed as in chapter 6.

The work in this chapter was undertaken in collaboration with Santosh Hemchandra (IISc Bangalore)[112], who wrote the original version of the LSGEN2D code, and has used it to analyse the response of premixed flames to acoustic forcing [113] and to equivalence ratio fluctuations [114]. The work was also undertaken in collaboration with Karthik Kashinath (CUED), who constructed the velocity field models for the current version of the LSGEN2D code, and has studied the nonlinear behaviour of premixed flames in detail [28, 111]. In a co-authored paper Kashinath has also used the LSGEN2D code developed in this chapter to show the importance of unstable attractors in mode switching [79].

## 7.1 Model description

The ducted premixed flame model consists of a 2D flame domain coupled to a 1D acoustic duct. The 2D flame domain uses the G-equation and a level set method to advect the flame shape. The flame domain can represent either 2D or axisymmetric flame configurations. The model of the acoustic duct is the same as that used in the diffusion flame model of chapter 6 and the Rijke tube model of chapter 2.

### 7.1.1 Acoustic field

Similar to the Rijke tube model in Chapter 2, the dimensional acoustic equations are:

$$\tilde{\rho}_0 \frac{\partial \tilde{u}}{\partial \tilde{t}} + \frac{\partial \tilde{p}}{\partial \tilde{x}} = 0$$

$$\frac{\partial \tilde{p}}{\partial \tilde{t}} + \gamma \tilde{p}_0 \frac{\partial \tilde{u}}{\partial \tilde{x}} + \zeta \frac{\tilde{c}_0}{\tilde{L}_0} \tilde{p} - (\gamma - 1) \dot{\tilde{Q}} \delta (\tilde{x} - \tilde{x}_f) = 0$$

Using the non-dimensionalisations $u = \frac{\tilde{u}}{\tilde{u}_0}, p = \frac{\tilde{p}}{\gamma M \tilde{p}_0}, x = \frac{\tilde{x}}{\tilde{L}_0}, t = \frac{\tilde{t}\tilde{c}_0}{\tilde{L}_0}$, the equations become:

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = 0$$

$$\frac{\partial p}{\partial t} + \frac{\partial u}{\partial \tilde{x}} + \zeta p - \beta_T \dot{Q} \delta (x - x_f) = 0$$

where if $\alpha$ is the ratio of burner width to duct width, and $\dot{Q}$ is the heat release rate, then the heat release parameter, $\beta_T$, is:

$$\beta_T = \frac{(\gamma - 1) \dot{\tilde{Q}}_0 \alpha}{\gamma \tilde{p}_0 \tilde{u}_0} \quad \text{and} \quad \dot{Q} = \frac{\dot{\tilde{Q}}}{\dot{\tilde{Q}}_0}$$

### 7.1.2 Flame field

Premixed flames propagate in a direction that is normal to the local flame surface. Simulations of laminar premixed flames must be able to calculate both the location of the flame surface and the normal vector to the flame surface. This is commonly achieved in one of two distinct ways: flame tracking, where the location of the flame surface is discretised by a spline or by discrete points along its length; or the G-equation, where the flame surface is defined as the zero contour of a scalar field and the flame normals are defined by the local gradient of the scalar field. The latter approach is used in this chapter because it has several advantages, most notably that it can can cope with flame flashback, pinch-off and the formation of cusps. A

boundary condition is defined in section 7.2.2 that allows the attachment point of the flame to either flash back into the burner tube, or move across the burner lip when the outwards perturbation velocity is larger than the flame speed.
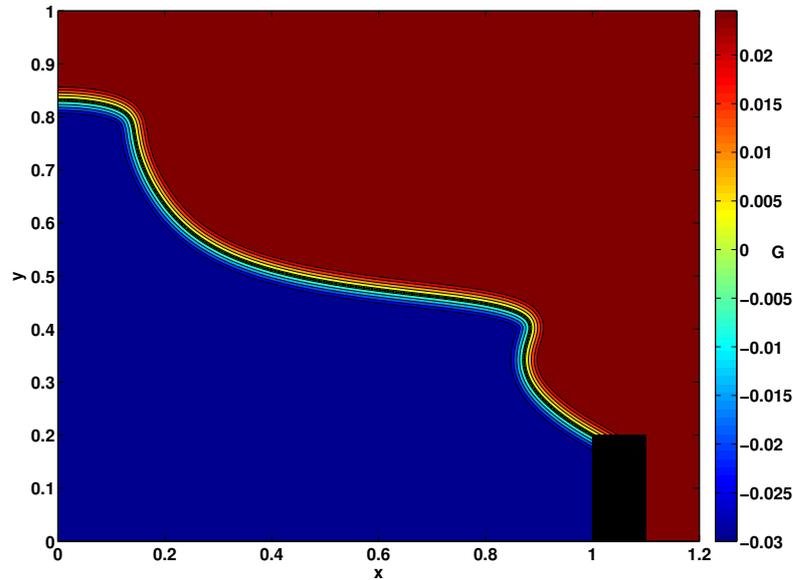


Figure 7.1: An example of the G-field for an axisymmetric flame with centreline at $x = 0$. The burner body is shown as a thick black rectangle, and the flame surface as a thin black line at the zero contour of G. The colours show the value of the scalar G, which are a signed distance function when local to the flame, and are set to fixed signed values when far from the flame.

Figure 7.1 shows the scalar field $G(x, y)$. The G-field is defined by two constraints: first, that the flame surface is the contour $G = 0$, and second, that the value of G is a signed distance function from the $G = 0$ contour. A signed distance function has a gradient of magnitude 1, which ensures that the gradient of the G-field is equal to the normal vector to the flame. It is important to note that the scalar G has no intrinsic physical property. The G-equation is an example of a level set method, which track surfaces using fixed grids.

It is only necessary to measure the G-field locally around the flame surface (Figure 7.1), because the flame propagation is governed only by quantities at the flame surface: the normal vector, the curvature and the local velocity field. To evaluate the gradients at the flame surface, a stencil of several grid cells either side of the flame surface is used with a fifth order Weighted Essentially Non Oscillatory (WENO) procedure [115, 116, 117]. It is therefore only necessary for the G-field to be a signed distance function in a thin tube around the flame surface, because only information in that tube is required to evolve the flame surface in time.

The evolution of the G-field in time is given by:

$$\frac{\partial G}{\partial t^*} + \underline{U}^* \cdot \underline{\nabla G} - s_L \left| \underline{\nabla G} \right| \quad = \quad 0 \tag{7.1}$$

where $s_L$ is the flame speed, and $\underline{U}^*$ is the velocity vector in the flame field, and an asterisk represents a dimensional quantity in the flame domain.

Equation (7.1) can be rearranged, using the substitution $\left| \underline{\nabla G} \right| = \frac{\nabla G \cdot \nabla G}{\left| \underline{\nabla G} \right|}$, to give:

$$\frac{\partial G}{\partial t^*} + \underline{U}^* \cdot \underline{\nabla G} - s_L \frac{\nabla G \cdot \nabla G}{\left| \underline{\nabla G} \right|} \quad = \quad 0$$

$$\frac{\partial G}{\partial t^*} + \left( \underline{U}^* - s_L \frac{\underline{\nabla G}}{\left| \underline{\nabla G} \right|} \right) \cdot \underline{\nabla G} \quad = \quad 0$$

$$\frac{\partial G}{\partial t^*} + \left( \underline{U}^* - s_L \underline{\hat{n}} \right) \cdot \underline{\nabla G} \quad = \quad 0$$

where $\underline{\hat{n}}$ is the unit normal to the flame surface, which is the given by the gradient of the G-field.

In two dimensions, the velocity field is split into $x^*$ and $y^*$ components, $\underline{U} = [U, V]$. The G-equation in two dimensions is therefore:

$$\frac{\partial G}{\partial t^*} + U \frac{\partial G}{\partial x^*} + V \frac{\partial G}{\partial y^*} - s_L \sqrt{\left( \frac{\partial G}{\partial x^*} \right)^2 + \left( \frac{\partial G}{\partial y^*} \right)^2} = 0$$

### 7.1.3 Heat release

The heat release rate in 2D is:

$$\dot{Q} = \int_D \rho s_L \left( \phi \right) h_R \left( \phi \right) \left| \underline{\nabla G} \right| \delta \left( G \right) dx^* dy^*$$

where the $\delta(G)$ function is a two dimensional numerical integral over the zero contour of the G field (section 7.2.3).

When the equivalence ratio is fixed across the domain, the heat release rate in 2D is given by:

$$\dot{Q}_{2D} = \rho s_{L0} \left( \phi \right) h_R \left( \phi \right) \int_D \left( 1 + M_\kappa \kappa_{2D} \right) \left| \underline{\nabla G} \right| \delta \left( G \right) dx^* dy^*$$

where $M_\kappa$ is the Markstein length and $\kappa_{2D}$ is the two dimensional curvature (section 7.2.7)

When the equivalence ratio is fixed across the domain, and the domain is axisymmetric, the heat release rate is given by:

$$\dot{Q}_{axi} = \rho s_{L0} \left( \phi \right) h_R \left( \phi \right) \int_D 2\pi r \left( 1 + M_\kappa \kappa_{axi} \right) \left| \underline{\nabla G} \right| \delta \left( G \right) dr^* dz^* \tag{7.2}$$

Strain effects are not included in the flame speed because it is not appropriate with the current velocity model (section 7.2.6).

## 7.2 Numerical formulation

The LSGEN2D code was originally written by Santosh Hemchandra (IISc Bangalore) and used to study the response of premixed flames to acoustic forcing [113] and to equivalence ratio fluctuations [114]. The same G-equation solver has also been used by Preetham to study the effects of velocity models and flame strain on flame dynamics [118]. The flame model was later modified by Karthik Kashinath (CUED) to include a simple velocity model and coupling to 1D duct acoustics [28, 79, 111]

Continuation methods require the numerical model to have very low numerical noise levels, especially when matrix-vector products are evaluated using a finite difference procedure. Several of the routines in the LSGEN2D code needed to be upgraded in order to reduce the noise to a suitable level for continuation to work. The most notable modifications were, in order of increasing importance: a cut-off function in the advection equation, rotating boundary conditions, WENO derived curvature effects in the flame speed, a higher order delta function to integrate the heat release rate, and a modified reinitialisation equation. Together these techniques have reduced the noise in the code by at least four orders of magnitude. Some of the more important details of the numerical formulation are described in this subsection.

### 7.2.1 Domain geometry

The domain is two dimensional and uses a Cartesian grid. The WENO gradient of a cell value uses weighted finite difference stencils, which require the values from three cells upwind and three cells downwind. To allow the cells at the edge of the domain to take gradients using two-sided stencils, a border of ghost cells is added around the 2D grid. A vector of boundary conditions is used to describe the location of the lip of any burner walls within the domain. At each of these burner lips, a rotating boundary condition is added (section 7.2.2). The local level set band around the flame surface is split into two regions as defined by Ref. [119]: the tube, which immediately surrounds the flame surface, and the halo, which immediately surrounds the tube. The tube and the halo are treated differently in the advection equation (see section 7.2.5).
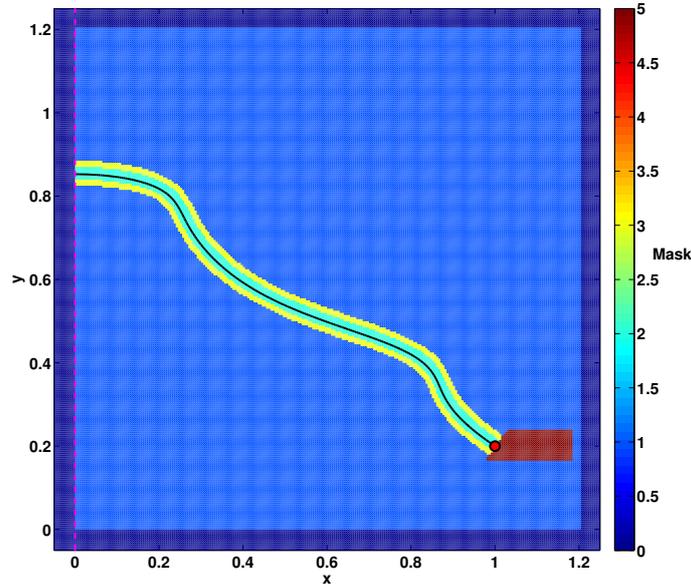
Figure 7.2: Mask to distinguish between the important regions in the flame domain: (0) ghost cells, (1) unused domain cells, (2) tube cells, (3) halo cells, (5+) boundary condition region (section 7.2.2). The black line shows the flame contour, the red circle shows the location of the burner lip and the magenta dashed line shows the centreline.

A mask is defined to distinguish between each of these separate regions in the domain. Figure 7.2 shows this mask for a typical axisymmetric configuration. The mask is updated every timestep to ensure that the flame remains in the centre of the tube, and to ensure that the boundary condition is in the correct orientation (section 7.2.2).

To define the geometry of the flame it is necessary to provide three things: first, the location of the centreline, second, a flag to define whether the flame is 2D or axisymmetric, and third, a vector of boundary condition information. The boundary condition information must include three things: the $(x, y)$ location of the burner lip, a flag to define whether it is the left or right hand side of the burner (important for flashback), and the width of the burner (important for bulging). The geometry information provided to the code allows many different configurations to be analysed, some examples of which are: one half of a 2D symmetric/axisymmetric flame, both halves of a 2D flame, a 2D/axisymmetric dump combustor, multiple flames side by side.

## 7.2.2 Rotating boundary conditions

At the burner lip, premixed flames exist in one of four different forms. They are either anchored to the burner lip, flashing back into the burner, bulging out onto the top of the burner, or lifted off the burner completely. The LSGEN2D code can model the first three forms but cannot model lift-off. A boundary condition must be defined at the burner that allows the flame to

progress smoothly between the attached, flashback and bulging forms. Physically, flashback occurs when the flame speed is higher than the vertical velocity at the attachment point and the flame surface is locally horizontal. Bulging occurs when the horizontal velocity at the burner lip is outwards (away from the centreline) and the flame surface is locally vertical.
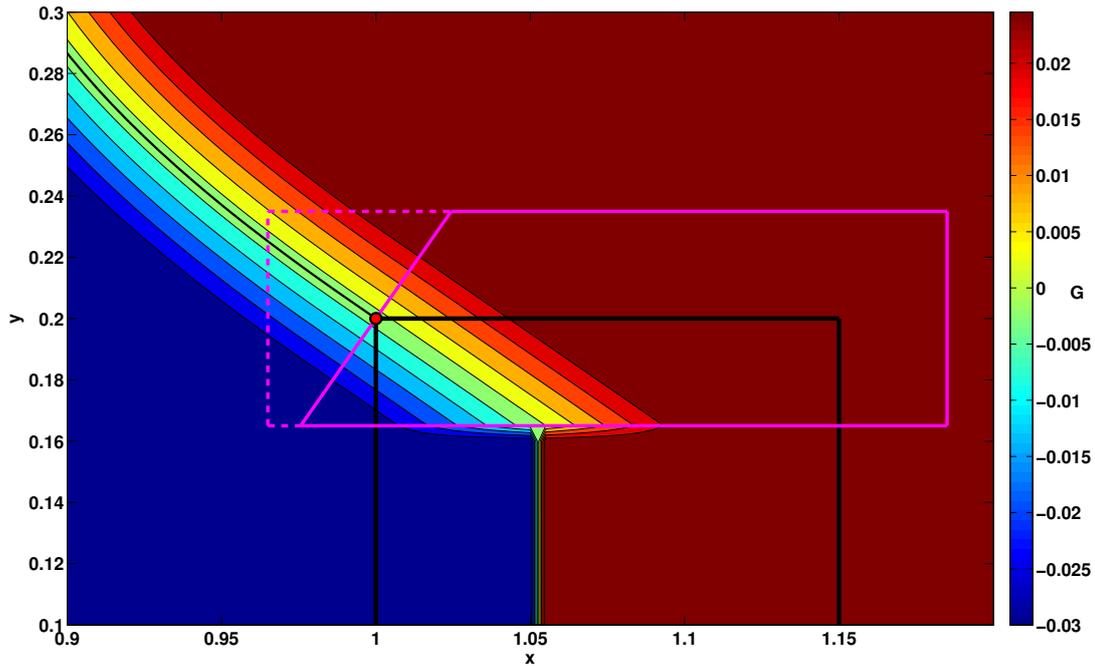


Figure 7.3: Rotated boundary condition when the flame is attached to the burner lip. The thick black lines show the walls of the burner, which has a finite width to allow the flame to bulge on the top of the burner. The red circle shows the attachment point. The dashed magenta rectangle shows all cells that are considered for the boundary condition in the attached and bulging cases. The boundary condition domain is surrounded by a solid magenta line. In this region the G-field is extended from the values at the attachment point.

A rotating boundary condition is applied to maintain a constant gradient of the G-field at the burner lip. Figure 7.3 shows the rotating boundary condition when the flame is attached to the burner lip. All cells in the dashed magenta rectangle are examined to see whether they lie in the flame domain or the boundary condition domain. The boundary condition domain is shown by the solid magenta line (and the BC region in Figure 7.2). Whether a cell lies in the flame domain or the boundary condition domain is decided by the normal vector at the attachment point, $\hat{\underline{n}}_{att}$, and a simple cross product relationship, where $\sigma = 1$ for the right hand side of the burner and $\sigma = -1$ for the left hand side:

$$\text{Flame domain if: } \sigma \begin{bmatrix} x - x_{att} \\ y - y_{att} \end{bmatrix} \times \underline{\hat{n}}_{att} \quad \leq \quad 0$$

$$\text{Boundary condition domain if: } \sigma \begin{bmatrix} x - x_{att} \\ y - y_{att} \end{bmatrix} \times \underline{\hat{n}}_{att} \quad > \quad 0$$

Figures 7.4a and 7.4b show the rotating boundary condition during flashback and bulging. When the flame is flashing back or bulging the attachment moves along the burner surface, but the G-field remains smooth at the attachment point. If multiple boundary conditions are included in the domain then they act independently; one boundary condition might be flashing back whilst another might still be attached.
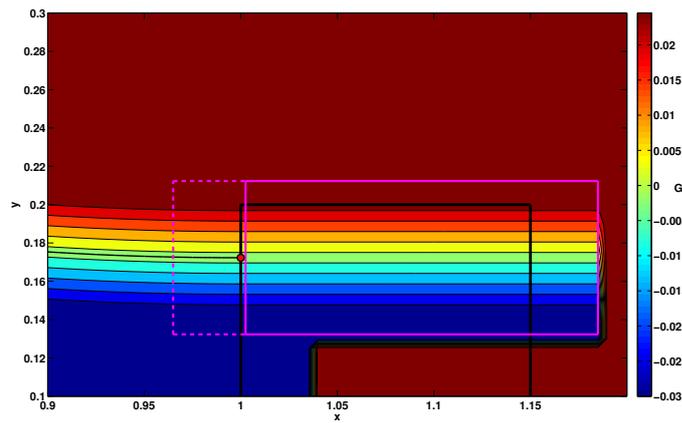


(a) Flashback
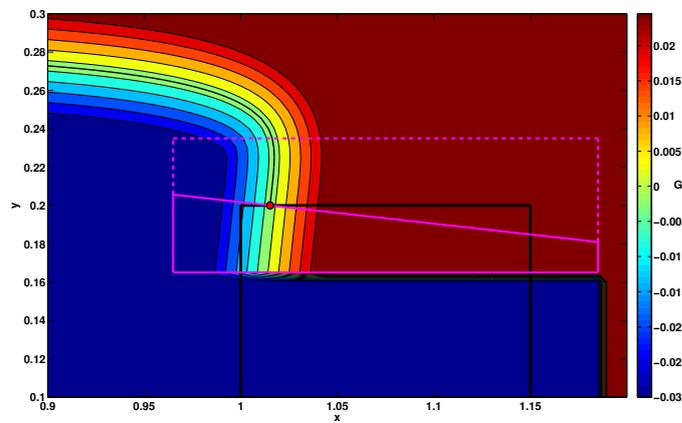


(b) Bulging

Figure 7.4: Rotating boundary condition during (a) flashback and (b) bulging.

### 7.2.3 Delta function

The delta function, $\delta(G)$, is used to integrate the flame area and get the heat release rate. The delta function in this chapter uses the numerical formulation of Wen [120]. The delta function first finds each cell that contains the flame contour, and uses cubic polynomial interpolation to find the two locations where the flame surface crosses the cell boundaries (Figure 7.5, red dots). An intermediate point between the two crossing locations is then found by two dimensional cubic polynomial interpolation (Figure 7.5, blue dots) and then Simpson's rule is used to integrate the length of flame surface within the cell.
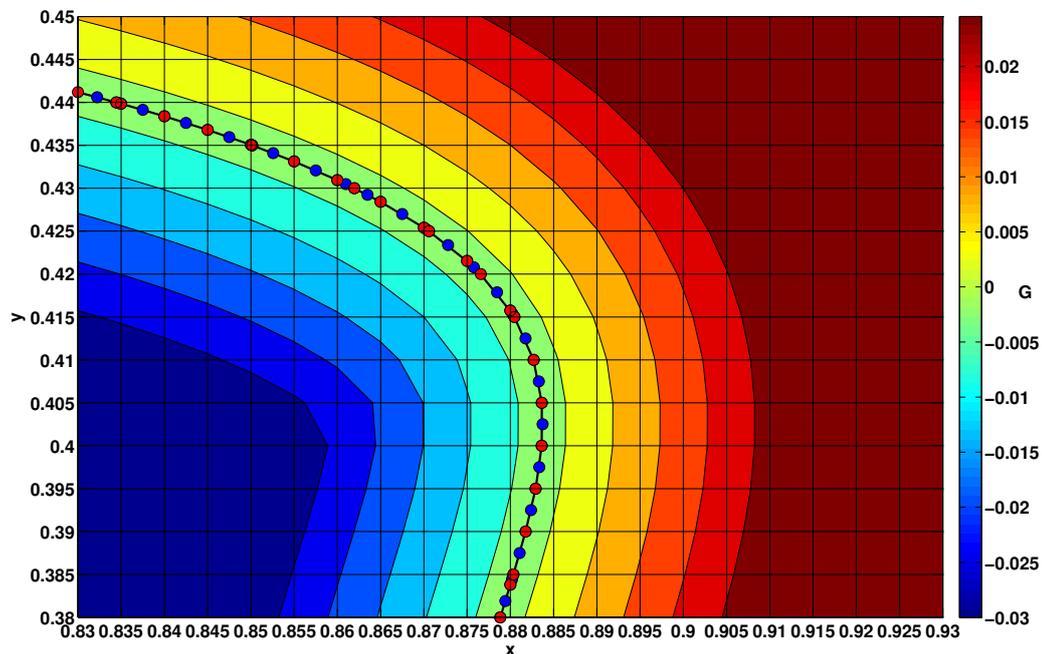


Figure 7.5: Delta function to integrate the heat release. The locations where the flame surface crosses cell boundaries (red dots) are found by polynomial interpolation. An additional location is found between the cell crossings (blue dots) by two dimensional polynomial interpolation.

### 7.2.4 Reinitialisation equation

The gradient of the G-field must be equal to the normal vector to the flame surface, which can only be achieved if the G-field is a signed distance function from the flame surface. After each timestep the flame surface has moved, however, so it is necessary to reset the G-field to be contours from the new flame surface location. This process is known as reinitialisation.

Reinitialisation can be achieved in one of two ways: first, by calculating the signed distance from the location of interest to every point on the flame, and then choosing the distance with the lowest absolute value; second, by solving the Eikonal equation ($|\nabla G| = 1$) using a pseudo-

time dependent PDE. The PDE has a steady solution with gradient of magnitude 1, and a time dependent factor that carries information hyperbolically away from the flame surface. The G-field is reinitialised by marching the PDE forwards in pseudo-time towards its steady state. The PDE approach is efficient when accurate G contours are only required in a narrow band around the flame surface, and is therefore the approach used in this chapter.

A reinitialisation method should create a contour field without altering the location of the zero contour. Early versions of the LSGEN2D code used the reinitialisation method of Peng [119], which was found to move the zero contour significantly, in particular by rounding off cusps. This movement of the zero contour resulted in large amounts of noise, because both the advection and reinitialisation routines continually changed the flame shape. Each timestep the reinitialisation equation rounded off the cusps slightly, then the advection equation let the cusps sharpen again and the cycle continued. Because this process happened every timestep, significant noise was generated.

In this chapter the HCR-2 reinitialisation method of Hartmann [121, 122] is used with fifth order WENO gradients. Hartmann's method modifies the method of Peng [119] to include an additional forcing term in the PDE that minimises the movement of the zero contour. This modification reduced the noise level in the timemarching code by over three orders of magnitude, with only a small extra computational cost. The reinitialisation equation is:

$$\frac{\partial G}{\partial t_r} = -S(G)\left(|\nabla G| - 1\right) + F$$

where $t_r$ is the reinitialisation timescale, $S(G)$ is a smoothed sign function $S(G) = G/\sqrt{G^2 + \epsilon^2}$, $\epsilon \ll 1$, and $F$ is the forcing function [121]. This forcing function is applied only to the cells that directly surround the flame surface. In terms of the timescale $t_r$, the information about the zero contour propagates outwards with speed 1. A Courant number of 0.5 is used for the reinitialisation equation in this chapter.

### 7.2.5 Advection equation

The advection equation uses fifth order WENO gradients and a Runge-Kutta 3 Total Variation Diminishing (TVD) timemarching scheme. As suggested by Peng [119, eq.9], a cut-off function is applied across the tube and halo. The cut-off function is exactly one within the tube and exactly zero at the edge of the halo, with a smooth function between the two. This means that the cells within the tube advect fully, whereas those in the halo advect only partially. This reduces instability at the Runge-Kutta pseudo timesteps, because the field is not reinitialised

after each pseudo timestep. The rotated boundary condition is applied after each pseudo-timestep.

### 7.2.6 Velocity field

A simple velocity field is applied to the flame domain. Experiments on premixed flames have shown that acoustic perturbations cause travelling waves to advect down the flame [29]. The travelling waves typically advect more slowly than the mean flow. In this chapter, the 1D advection equation propagates the acoustic disturbance at the burner downstream [28, 113, 118], at a speed of $u_0/K$, where $K$ is a parameter set typically between 1.0 and 1.5 [123]. The velocity field is therefore simply the time history of the acoustic perturbation at the burner lip; the further from the burner lip, the further back in time that the acoustic disturbance was generated. The transverse velocity is determined by satisfying continuity.

The velocity model produces flame shapes that look similar to those seen in experiments using a relatively simple formulation. This velocity model gives rise to some non-physical behaviour, however, such as that the mean heat release rate decreases as the forcing amplitude increases. It should stay the same because the average amount of fuel entering the flame is constant, and all fuel must be burnt. This defect was detected late in the project but, because it does not affect the successful application of the continuation algorithms, the model was retained. A more advanced velocity model, such as the Euler method used by Preetham [118], would be better for more detailed studies.

The value of $K$ has been demonstrated in numerical studies to have a strong effect on the nonlinear behaviour of premixed flames [28]. In particular, subcritical Hopf bifurcations are more prevalent at high values of $K$, and there are more parameter regions that have multiple limit cycles at different amplitudes and frequencies.

### 7.2.7 Curvature dependent flame speed

Much of the noise in the early LSGEN2D code was generated at the flame cusps and at the centreline. This is because sharp cusps cause a discontinuity in the contour field which increases error in gradient calculations. Adding curvature dependence to the flame speed causes sharp cusps to become more rounded, and causes the flame to become horizontal at the centreline. This more closely represents the physics of experimental flames. For a conical flame which has been scaled by $\beta$ in the $x$ direction, the flame speed is dependent on the flame aspect ratio, $\beta$, and is given by:

$$s_L = \frac{u_0}{\sqrt{1 + \beta^2}} \left(1 + M_\kappa \kappa\right)$$

where $M_\kappa$ is a nondimensionalised Markstein length and $\kappa$ is the signed curvature, which has a different formula in the 2D and axisymmetric cases. Strain effects are not included in the flame speed because it is not appropriate with the simple velocity field model.

Without curvature effects, the cusps formed by the LSGEN2D code are unphysically sharp. Curvature effects are therefore used in this chapter to both reduce the noise and make the flame shapes more physically realistic. Because curvature is a second order quantity, however, adding curvature effects requires a significant drop in timestep to ensure that the CFL condition is met, and therefore timemarching is significantly slower.

## 7.3 Adaptation for matrix-free continuation methods

To perform continuation methods, a suitable state vector must be defined. The state vector is a snapshot of the system at an instant in time. For continuation methods, the state vector must obey three principles. First, the state vector must contain all the information required to describe the system at that instant in time. It should be possible to restart the simulation exactly from the state vector alone - time traces of any variable or derived quantity should show no sign of a restart having occurred. Second, the state vector should contain no information that is not required to describe the state of the system - each variable in the state vector must contain some independent information. Otherwise, if a dependent quantity is included in the state vector then the state vector may become inconsistent when perturbed by the continuation algorithm. For example, it would not be appropriate to include a mean flame speed if the mean equivalence ratio were also included. Third, the state vector must have a suitable form for perturbation. This last principle is subtle and often problem specific.

If matrix-vector products are obtained by the first variational equations, this last principle can be achieved solely by weighting the variables in the state vector. The variables should be weighted such that a perturbation of a fixed size affects the system to a similar degree, for example: a perturbation of size $10^{-2}$ on a pressure value of $10^5$Pa would have almost no effect on system behaviour, but the same perturbation on a mixture fraction value might have a significant effect on system behaviour - in this case the pressure value should be scaled down. Note that this weighting methodology is not the same as simply scaling each variable to have a similar magnitude, although this often has a similar effect. The weighting is important for two reasons: first, so that the Jacobian matrices are well conditioned and therefore easier for

iterative methods to solve, and second, so that a 2-norm of the state vector is appropriate to indicate convergence of the iterative methods.

If matrix-vector products are obtained by finite difference approximations, however, then weighting alone is no longer enough to ensure that the state vector is suitable for perturbation. Finite difference matrix-vector products for limit cycles were defined in chapter 5, equation (5.6), to be $J\underline{v} = \underline{v} - (A(\underline{x}(0) + \delta\underline{v}) - A(\underline{x}(0)))/\delta + \mathcal{O}(\delta)$, where $\underline{v}$ is an arbitrary vector and $A(\underline{x})$ represents timemarching forward $T$ time units from state $\underline{x}$. The accuracy of the matrix-vector product is dependent on both $\delta$ and the level of noise in the timemarching process. The value of $\delta$ should be small for an accurate matrix-vector product (to be approximately linear), but must also be large enough that the effect of the perturbation is not lost in the timemarching noise. A noisy timemarching process therefore means that a larger value of $\delta$ must be used, which in the case of the premixed flame model, limits which discretisations can be used to define the flame shape. This is discussed further in the next section.

### 7.3.1 Discretisation of the flame shape

In the premixed flame model of this chapter, the state vector must include the state of the acoustic field, the state of the velocity field and the state of the G-field. In the previous section, it was established that variables that do not contain any independent information should not be included in the state vector, because it leads to inconsistencies when the state vector is perturbed. The G-field is a good example of this point. The G-field is described entirely by the location of the flame surface. The signed distance function is a one to one mapping: each flame shape defines a unique G-field, and each G-field defines a unique flame shape.

It would be inappropriate to include the entire G-field in the state vector because when the state vector is perturbed the G-field will no longer be a signed distance function. This is because each value of G in the 2D field is not independent - it is defined by the location of the flame surface - and therefore it cannot be varied independently. A perturbation to the state vector must result in a new G-field that obeys a signed distance function to the flame surface.

The only information in the G-field is the shape of the flame surface. The state vector must therefore include a discretised version of this flame shape. The flame shape is a 2D curve, which may have multiple cusps, and may have multiple separate sections (during pinch-off). The discretisation used to describe the flame shape must satisfy four conditions. First, the discretisation must define a unique 2D shape to a suitable level of accuracy. Second, the size of the discretisation must not change when the flame surface is perturbed slightly, because the state vector must not change size during the solution of each linear equation (equation (5.3)). If the state vector changes size when perturbed, because the flame surface is slightly longer

or shorter, then the Jacobian matrix would change dimension during the iterative solution of the linear equation, which invalidates the linear equation that is being solved. Third, the discretisation must be suitable for comparing two flame shapes; as much as is possible, the discretisation should allow easy comparison between the same parts of a curve. This is not the case with arclength based discretisations (see later). Fourth, when a random perturbation of size $\delta$ is added to the discretisation to take a matrix-vector product, the flame shape must remain smooth. If the discretisation were simply $(x, y)$ points along the flame surface, for example, with a fine spacing to accurately describe the flame shape, then it is possible for the perturbation to create ripples on the flame surface with wavelength of double the point spacing. The LSGEN2D code can only accurately timemarch features that are several grid cells in size, so sub grid cell size ripples would cause extra noise and make matrix-vector products less accurate.

Several approaches have been considered for discretising the flame shape, which can be divided roughly into arclength based approaches and fixed location based approaches. A few of these approaches are discussed in this subsection, and evaluated against the four criteria in the previous paragraph. The discretisations are generally inappropriate during pinch-off or flashback, but that does not mean that the continuation methods cannot find limit cycles that contain pinch-off/flashback. Pinch-off and flashback generally occur for only a fraction of the limit cycle, therefore a limit cycle can be found if the phase of the cycle is fixed such that there is no pinch-off or flashback in the starting state, $\underline{x}(0)$. This is because the continuation methods see the timemarching process as a black box that maps an input state, $\underline{x}(0)$, to an output state, $\underline{x}(T)$, and therefore the pinch-off phenomena is within the black box and does not affect the continuation methods.

Arclength based approaches discretise the flame shape by storing quantities as a function of cumulative arclength, $s$. Because the length of the flame will change but the size of the discretisation must remain constant, it is better to store quantities in terms of normalised arclength, $\hat{s} = \frac{s}{s_{TOTAL}}$, and to store the total arclength, $s_{TOTAL}$, separately. One arclength approach would be to define two functions, $x = f_1(\hat{s})$ and $y = f_2(\hat{s})$. Another arclength approach would be to define the curvature of the flame shape, $\kappa = f(\hat{s})$, and the angle of the flame shape at the burner lip, $\phi_0$. By integrating the curvature function twice, it is possible to recreate the 2D flame shape [124]. Arclength based approaches have the significant advantage that functions of arclength are always single valued. Arclength based approaches have two main disadvantages, however: first, the process of calculating arclength (and curvature) generates errors which affect matrix-vector product accuracy; and second, it is not easy to compare the same parts of a curve when using normalised arclength. This latter disadvantage is important

with iterative methods. For example, imagine comparing two flame shapes that are identical near the burner but different near the centreline. Even where the flames are identical, the values of the discretisation are different, because the length of the two flames is different and therefore the discretisation points have slid along the flame surface. This makes it harder for the iterative algorithms to reduce the residual between two flame shapes, because they cannot operate solely on a small section of the flame; to reduce the difference between flame shapes on a small section of the flame requires all the values in the discretisation to be changed.

The simplest fixed location approach would be to evaluate the curve as a function $y = f(x)$, by storing $y$ values of the flame surface at fixed $x$ locations. This is simple but does not allow the function $f$ to be multi-valued, and therefore cannot discretise cusps. This discretisation has been used before in flame tracking models for linear stability analysis of premixed flames [10, 125], where amplitudes are small and sharp cusps do not occur. An equivalent discretisation using $y$ values at set fractions of the flame height, $x = f(y)$, would be unable to discretise the flame during flashback and would often be ill-conditioned, because the flame shape is often flat near the centreline when forced by the acoustics.
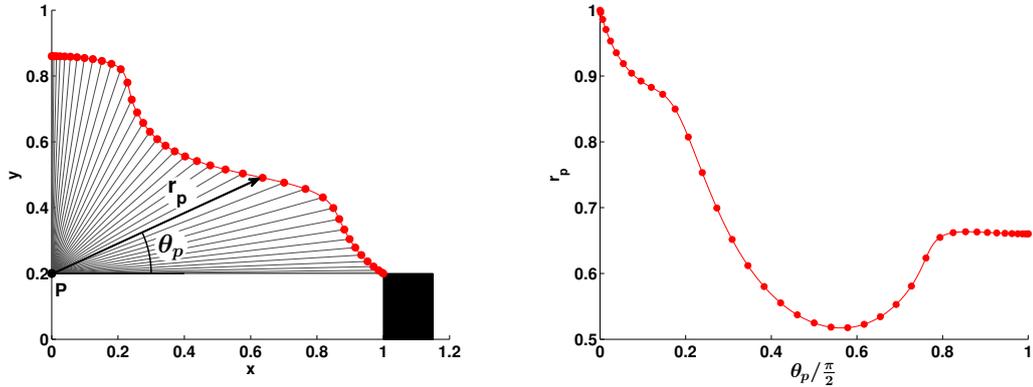


Figure 7.6: Discretisation method based on polar co-ordinates. A reference point, $p$, is placed at the centre point level with the burner lip. The flame shape is described by the radius from the reference point, $\underline{r}_p = f(\underline{\theta}_p)$, with a Chebyshev spacing in $\theta_p$.

A fixed location approach based on polar co-ordinates is chosen to generate the results in the latter part of this chapter. The discretisation is shown schematically in Figure 7.6. A reference point (P) is placed at the centreline at the level of the burner lip. The flame shape is then defined in polar co-ordinates from this point, with Chebyshev spacing in $\theta_p$ between 0 and $\frac{\pi}{2}$, to form a function $r_p = f(\theta_p)$. The subscript p is used to denote that this is the radius from the reference point, not the radius from the burner centreline. The Chebyshev discretisation ensures high accuracy and a smooth function over $\theta_p$. The discretisation is not perfect: it will fail if cusps are sharp and near the centreline, because $\underline{r}_p$ will no longer be a

single valued function of $\theta_p$. The discretisation also clusters points near the start and end of the curve, which may lead to sub grid size ripples when a large number of points is used. For the parameter regions studied in the results section, with a suitable choice of phase condition this discretisation was able to capture the flame shapes without failing.

Figure 7.7 shows a summary of the discretisation process, and shows how the GMRES solver of the continuation methods interacts with the LSGEN2D G-equation solver. The LSGEN2D timemarching routine is written in C, but the continuation routines and interface routines are written in MATLAB. The continuation routines and interface routines require negligible computational time compared to the timemarching routine.

## 7.4 Results

The matrix-free continuation methods of chapter 5 were used to generate a bifurcation surface of limit cycles as two parameters are varied: the flame location in the duct, $x_f$, and the aspect ratio of the flame, $\beta = \sqrt{\left(\frac{u_0}{s_L}\right)^2 - 1}$. The parameters that are held fixed are: $\phi = 1.0$, $\alpha = 0.7$, $K = 1.5$, $M_\kappa = 0.04$, $L_0 = 1$, $\rho_0 = 1.16^{kg}/m^3$, $p_0 = 10^5 \text{Pa}$, $\gamma = 1.4$, and acoustic damping factors $c_1 = 0.012$ and $c_2 = 0.024$[1] . The G-field is discretised on a $401 \times 401$ grid with spacing 0.005 and tube and halo regions both 3 grid cells wide. A timestep of $1.5 \times 10^{-4}$ is used with 14 reinitialisation steps per timestep, with reinitialisation Courant number of 0.5. Twenty Galerkin modes are used.

The aim of this section is to demonstrate that the continuation algorithms can efficiently map the nonlinear dynamics of a thermoacoustic system with a premixed flame model, by finding limit cycles and bifurcations to the limit cycles. The results demonstrate this well. Unfortunately, a small error was found in the code just before submission of the thesis: the $(1 + M_\kappa \kappa_{axi})$ term was not included in the heat release integral (equation (7.2)). The period-doubling and Neimark-Sacker bifurcations shift slightly in parameter space, but the qualitative behaviour remains unchanged.

Because the focus of this chapter is on the implementation of the continuation methods, rather than the quantitative location of bifurcations, these results are presented anyway.

---

[1]It should be noted that there is a slight inconsistency with these damping values - which are dependent on the duct dimensions, $R_0, L_0$ - and the flame parameters, which set a physical flame size through the flame speed, and the ratio $R_f/R_0$ through $\alpha$. For a consistent set of parameters, $c_1$ should be set slightly lower, and $c_2$ set slightly higher. Because the damping model in the acoustics is already basic, and the results of this chapter are not used for quantitative comparisons this discrepancy was considered acceptable.
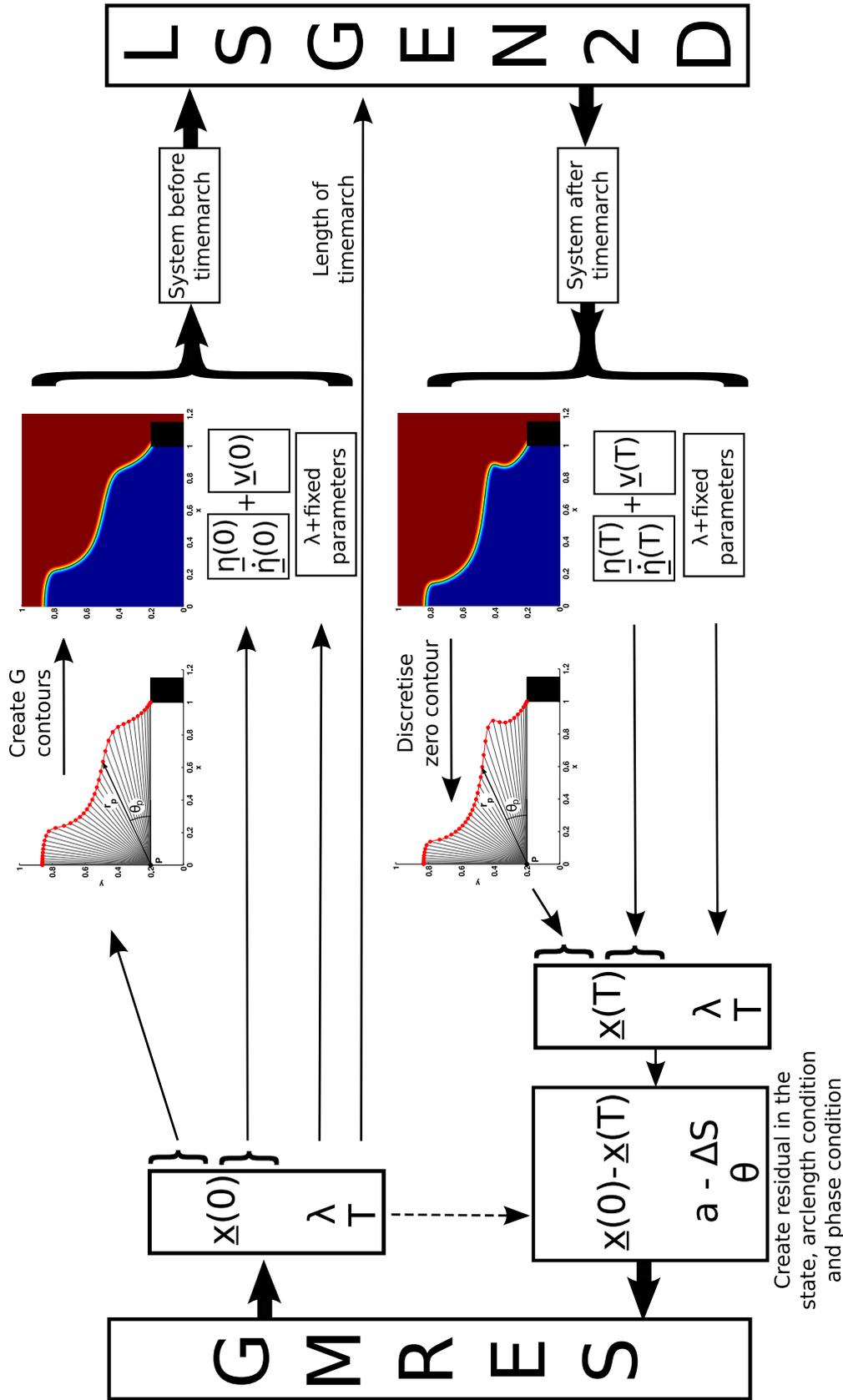
Figure 7.7: Schematic of the interface between the GMRES solver and the LSGEN2D G-equation code. The GMRES solver gives a new state vector, $\underline{x}(0)$, which contains values for: the polar co-ordinate discretisation of the flame shape, the acoustic Galerkin modes, $\left[\underline{\eta}(0); \underline{\dot{\eta}}(0)\right]$, the vertical velocity field, $\underline{v}(0)$, the current guess for the period, $T$, and a continuation parameter, $\lambda$, if arclength continuation is used. To form the input file for LSGEN2D, the G-contours must be created from the flame shape, and fixed parameters must be added. The LSGEN2D code then timemarches for $T$ time units. The zero contour of the G-field at time $T$ must then be discretised to form the state vector $\underline{x}(T)$. The residual vector is then created and fed back into GMRES.

### 7.4.1 Bifurcation surfaces

Figure 7.8 shows the bifurcation surface of period-1 limit cycles, whose frequency is close to that of the fundamental acoustic mode. The surface has regions with unstable limit cycles (dashed gray lines) and regions with stable limit cycles (solid black lines), whose boundaries are defined by the location of the period-doubling bifurcation (red line) and the Neimark-Sacker bifurcation (magenta line). The surface is shown only for realistic flame aspect ratios in the region $2 < \beta < 6$, and when the flame is in the first half of the acoustic duct, $x_f < 0.5$.
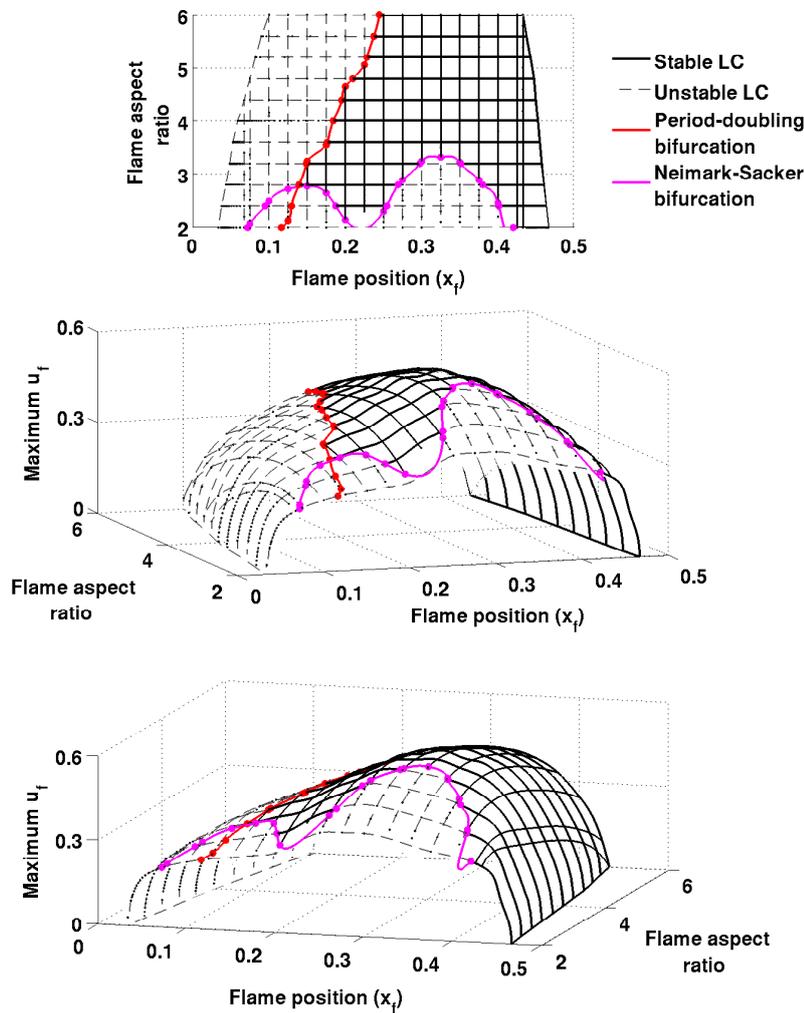


Figure 7.8: Bifurcation surface of period-1 limit cycles, as two parameters are varied: the flame location in the duct, $x_f$, and the flame aspect ratio, $\beta$. The surface is composed of over 600 limit cycles each converged to $||\underline{x}(0) - \underline{x}(T)|| < 5 \times 10^{-4}$. The $z$-axis is the maximum acoustic velocity at the flame. The surface has regions with unstable limit cycles (dashed gray lines) and regions with stable limit cycles (solid black lines), whose boundaries are defined by the location of the period-doubling bifurcation (red line) and the Neimark-Sacker bifurcation (magenta line). Subfigure (a) shows the surface from above, and subfigures (b) and (c) show the same 3D surface from two different views.

Figure 7.9 shows the bifurcation surface of period-2 limit cycles, whose frequency is close to half that of the fundamental acoustic mode. The surface shows the maximum velocity value at the flame during the limit cycle - this is one of several surfaces that show the period-2 cycle (see Figure 7.10). The surface has regions with unstable limit cycles (dashed blue lines and gray shading) and regions with stable limit cycles (solid blue lines), whose boundaries are defined by the location of the period-doubling bifurcation (red line) and the fold bifurcation (green line). The period-doubling bifurcation is the same as that on the period-1 surface (Figure 7.8). The surface is shown for the same parameter range as Figure 7.8. The fold bifurcation exists at high $\beta$ but its location is unresolved.
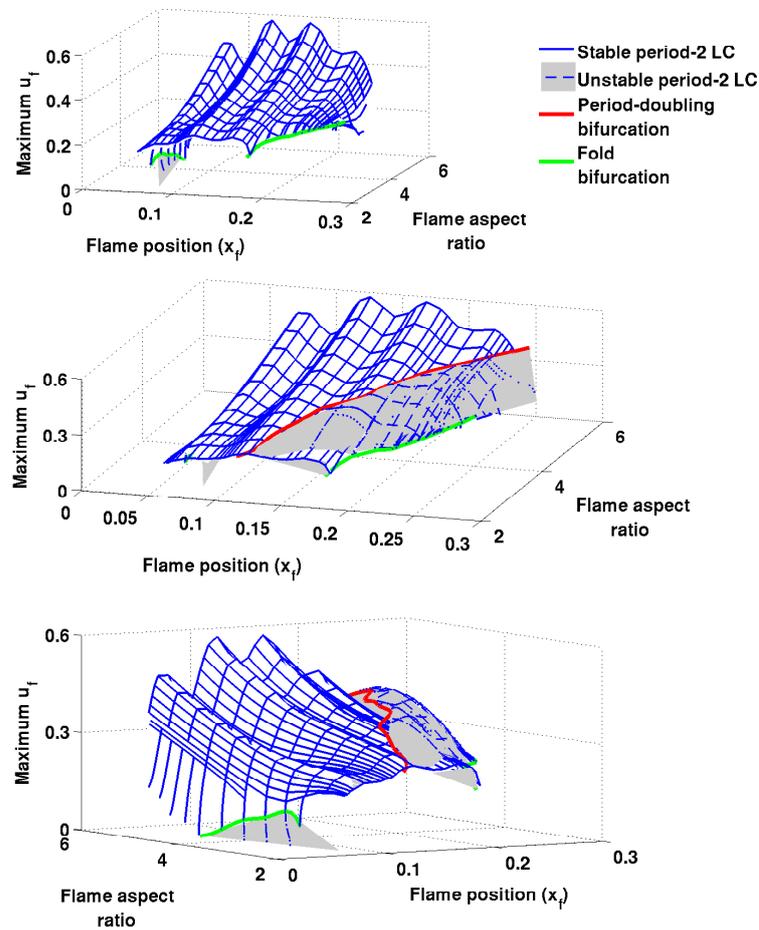


Figure 7.9: Bifurcation surface of period-2 limit cycles, as two parameters are varied: the flame location in the duct, $x_f$, and the flame aspect ratio, $\beta$. The surface is composed of over 1200 limit cycles each converged to $||\underline{x}(0) - \underline{x}(T)|| < 5 \times 10^{-4}$. The surface has regions with unstable limit cycles (dashed blue lines and gray shading) and regions with stable limit cycles (solid blue lines), whose boundaries are defined by the location of the fold bifurcation (green line) and the location of the period-doubling bifurcation (red line) - which is the same as that in Figure 7.8. Subfigure (a) shows does not show the unstable limit cycle surface; subfigures (b) and (c) show both the stable and unstable limit cycle surfaces from two different views.

# 7. CONTINUATION ANALYSIS OF A DUCTED PREMIXED FLAME

Figures 7.8 and 7.9 only show the amplitude of the velocity fluctuation during the limit cycles. For the period-2 cycles, more information can be gained by plotting both the peaks and the troughs of the time series during the limit cycle. This is how the experimental results of Kabiraj [20] and the computational results of Kashinath [79] are presented. Figure 7.10 shows a 2D slice of the combined period-1 and period-2 bifurcation surfaces, taken at $\beta = 4$, with the $y$-axis showing both the peaks and the troughs of the limit cycles.
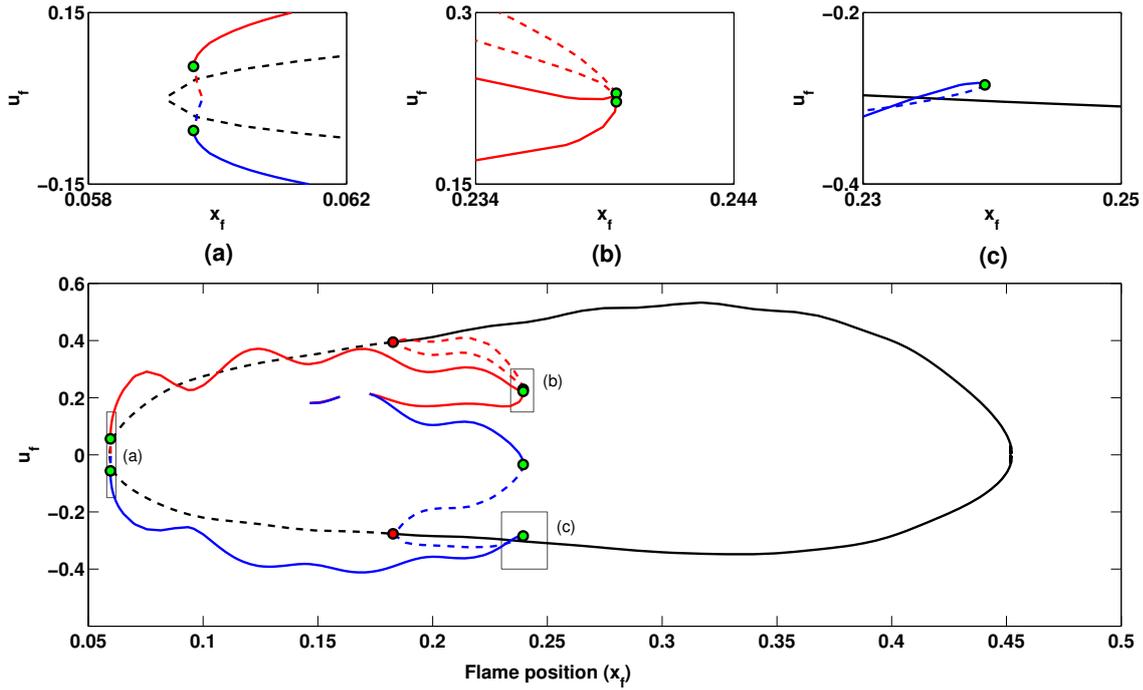


Figure 7.10: 2D slice of the combined period-1 and period-2 bifurcation surfaces, taken at $\beta = 4$. The $y$-axis plots the peaks and troughs of the velocity time series: period-1 peaks and troughs are shown in black, period-2 peaks are shown in red and period-2 troughs in blue. A solid line represents a stable limit cycle; a dashed line represents an unstable limit cycle. The period-doubling bifurcation is shown as a red dot; the fold bifurcations are shown as green dots.

The period-1 cycles (black) have only one peak and one trough, and these are not symmetric about zero - the velocity peak has a larger magnitude than the velocity trough. The period-2 peaks (red) and period-2 troughs (blue) form a more complicated shape. At some locations the period-2 cycles have two peaks and two troughs ($0.15 < x_f < 0.16$, $0.17 < x_f$), and at other locations the period two cycles have only one peak and one trough ($x_f < 0.15$, $0.16 < x_f < 0.17$). This difference occurs because the period-2 cycles are composed of two frequencies, whose relative magnitudes change along the period-2 branch (see next section). As the period-2 branch approaches the period-doubling bifurcation, the two peaks and two troughs close together. The period-doubling bifurcation is subcritical: the period-2 cycles emerging from it

are unstable, and overlap with the stable period-1 limit cycle. There is therefore a region of bistability between $0.182 < x_f < 0.240$, where four attractors exist: a stable period-1 limit cycle, a stable period-2 limit cycle, an unstable period-2 limit cycle, and an unstable fixed point.

It is important to note that there are only two fold bifurcations on Figure 7.10 - the four green dots at $x_f = 0.240$ are not four separate fold bifurcations, they are a single fold bifurcation acting simultaneously on the four separate traces. The same is true of the two green dots at $x_f = 0.06$.

The period-2 peaks and troughs on Figure 7.10 oscillate with a wavelength of $\Delta x_f = 0.1$, which matches the wavelength of the twentieth Galerkin mode (only twenty were considered). This is probably an indication that a larger number of Galerkin modes are required in the discretisation. Including more Galerkin modes would probably smooth the peaks and troughs, but would probably not qualitatively change the behaviour.

### 7.4.2 Limit cycles

Once the limit cycles have been found by the continuation methods, their form can then be analysed. At an operating condition of $x_f = 0.195$ and flame aspect ratio 4 (Figure 7.10), there are three limit cycles: a stable period-1 limit cycle, an unstable period-2 limit cycle and a stable period-2 limit cycle. All three of these cycles have comparable velocity amplitudes at the flame. In this section, the form of and spectra of these three limit cycles are compared.

Figure 7.11a shows snapshots of the flame during the stable period-1 limit cycle. The flame shapes are qualitatively similar to those seen in experimental axisymmetric flames [29]. In particular, it was found that including curvature effects was important if flame shapes are to be compared with experimental ones. Without curvature the cusps become too sharp, especially at the centreline.

Figure 7.11b shows the time traces and spectra of the acoustic velocity and pressure at the flame, and the heat release of the flame. The heat release time trace is not sinusoidal because it contains a significant amount of higher harmonics. This will always be true when there are cusps on the flame surface.

(a)                    (b)
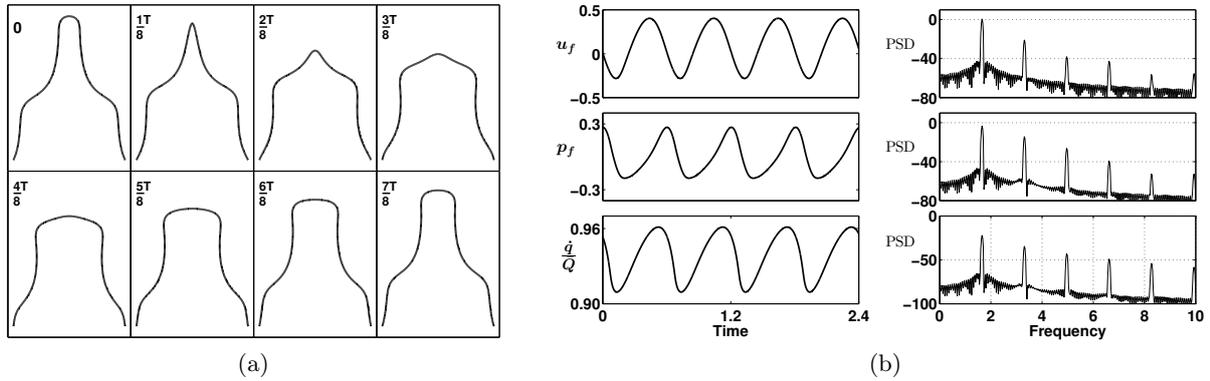
Figure 7.11: Snapshots of the flame surface (a) and time traces and spectra (b) for the stable period-1 limit cycle at $x_f = 0.195$, with steady state flame aspect ratio of 4.

Near the period-doubling bifurcation, the period-2 cycles have a particular form where the cusps alternate position during the first and second halves of the cycle. Figure 7.12 shows the equivalent of Figure 7.11, but for the unstable period-2 limit cycle. The black lines show the flame shape during the first half of the cycle and the gray lines show the flame shape during the second half of the cycle. Because this is a period-2 limit cycle, a peak has appeared on the spectra at 0.9, which is half the frequency of the fundamental acoustic mode.



(a)                    (b)
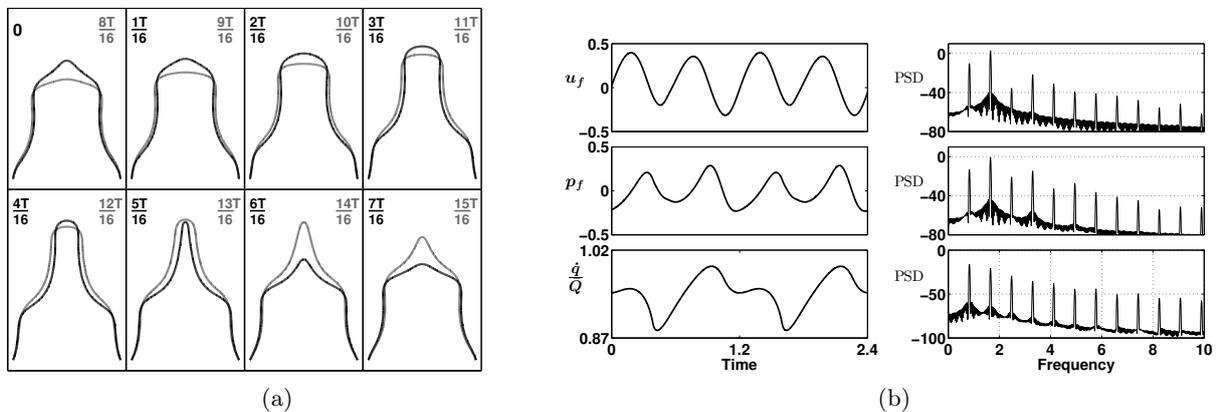
Figure 7.12: Snapshots of the flame surface (a) and time traces and spectra (b) for the unstable period-2 cycle at $x_f = 0.195$, with steady state flame aspect ratio of 4. The time scale is the same as Fig. 7.11.

When further from the period-doubling bifurcation, however, it is more difficult to identify that the cycle is part of the period-2 branch, and not just a limit cycle with half the frequency of the fundamental acoustic mode - the peaks of the time trace lose the characteristic one-up, one-down structure that is commonly associated with period-2 cycles. In other words, the half-frequency starts to dominate, and the flame shape does not alternate between the two
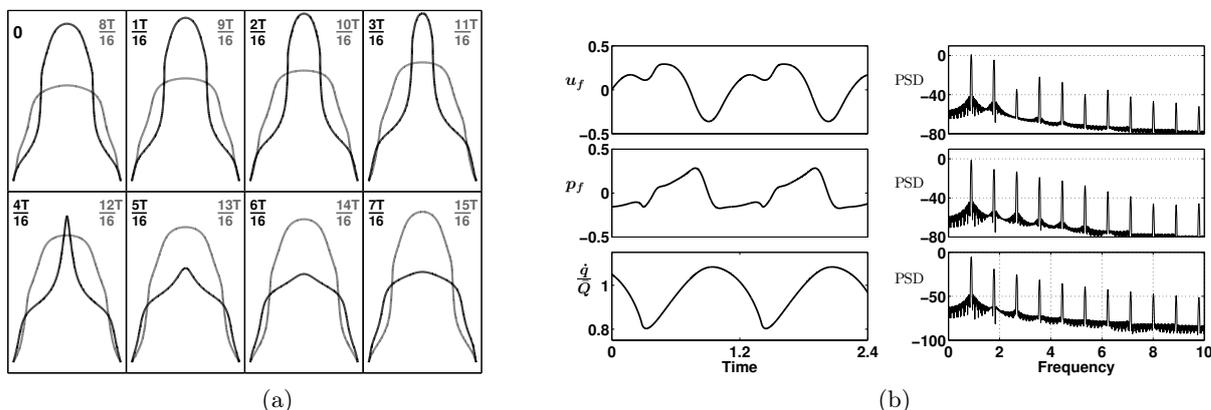
Figure 7.13: Snapshots of the flame surface (a) and time traces and spectra (b) for the stable period-2 cycle at $x_f = 0.195$, with steady state flame aspect ratio of 4. The time scale is the same as Fig. 7.11.

halves of the cycle. Figure 7.13b shows an example of this, for the stable period-2 cycle.

The qualitative change in the form of the cycle along the period-2 branch can be seen using phase portraits. Figure 7.14 shows the phase portraits of the converged period-2 cycles at different locations along the period-2 branch of Figure 7.10, starting from near the Hopf bifurcation (Figure 7.14(a)) and moving along the branch towards towards the period-doubling bifurcation (Figure 7.14(t)). Because this branch of period-2 cycles is a strong function of two frequencies, it will not be captured well by frequency domain methods such as the FDF, because they consider each frequency independently.

### 7.4.3 Floquet multipliers

The Floquet multipliers describe the stability of a limit cycle to infinitesimal perturbations, as was discussed in section 6.2.3.2 for the model of the ducted diffusion flame. Bifurcations to the limit cycle occur when a Floquet multiplier, or complex pair of Floquet multipliers, cross the unit circle. The location of the crossing point defines the type of bifurcation. The ducted premixed flame model shows several bifurcations of limit cycles: fold bifurcations, period-doubling bifurcations and Neimark-Sacker bifurcations.

#### 7.4.3.1 Fold bifurcation

A fold (LPC) bifurcation occurs when a Floquet multiplier crosses the unit circle at $+1$. The bifurcation changes the stability of a branch of limit cycles from stable to unstable. A fold bifurcation is observed on the period-2 branch of Figure 7.10 at $x_f = 0.239$. Figure 7.15 shows the Floquet multipliers of the period-2 cycles either side of the fold bifurcation in Figure 7.10, which clearly show the Floquet multiplier crossing $+1$. Together, the fold bifurcation
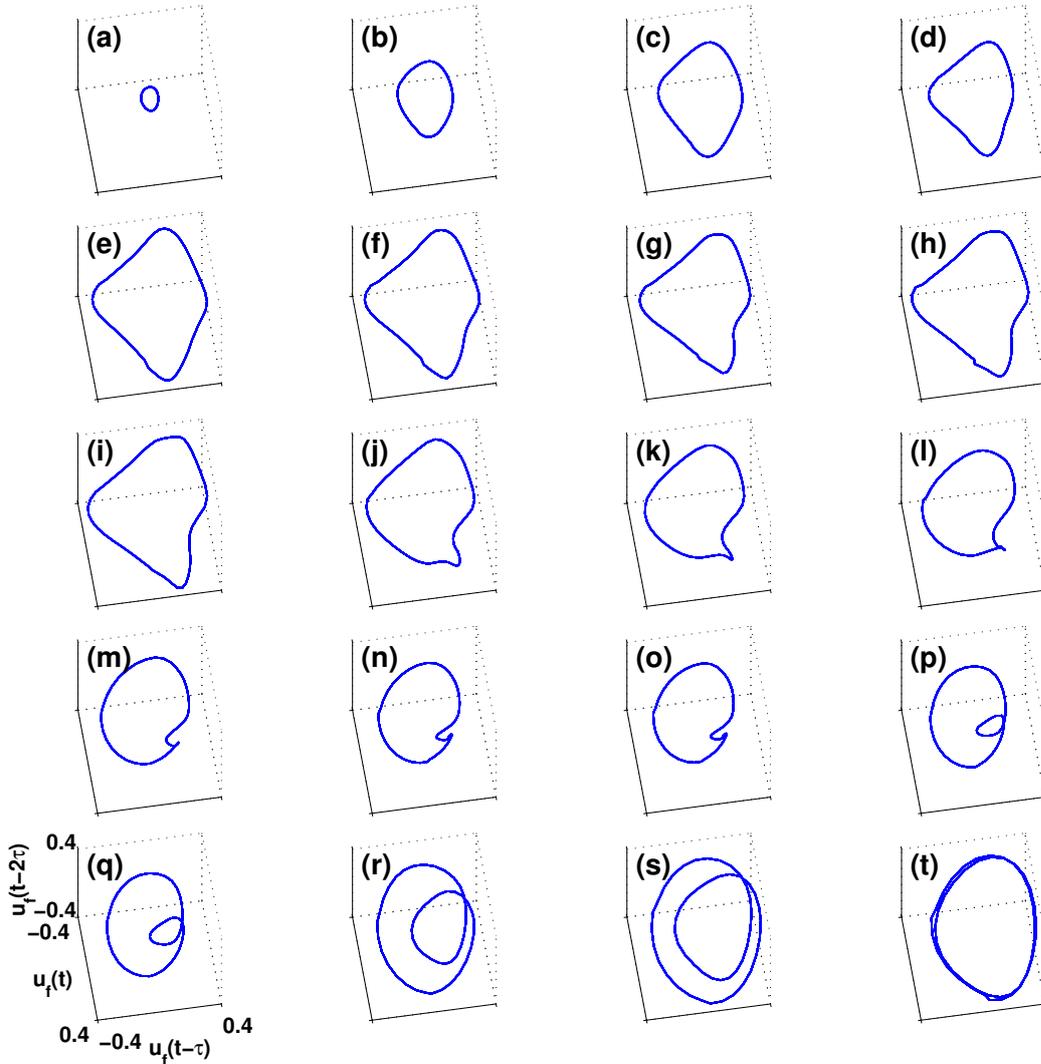
Figure 7.14: Phase portraits of the period-2 limit cycles at different locations along the period-2 branch of Figure 7.10. Near the Hopf bifurcation (a), the cycle is nearly sinusoidal with a frequency of half the fundamental acoustic mode. The cycle becomes less sinusoidal as $x_f$ increases (b-j), because the response at the fundamental acoustic mode increases relative to the response at the half frequency. As the cycle moves closer to the period-doubling bifurcation (k-t) the phase portraits develop the familiar double loop form of a period-2 cycle, because the response of the fundamental acoustic mode is much greater than that of the half frequency. The fold bifurcation occurs between (q) and (r); the period-2 cycles (a-q) are stable, the period-2 cycles (r-t) are unstable.
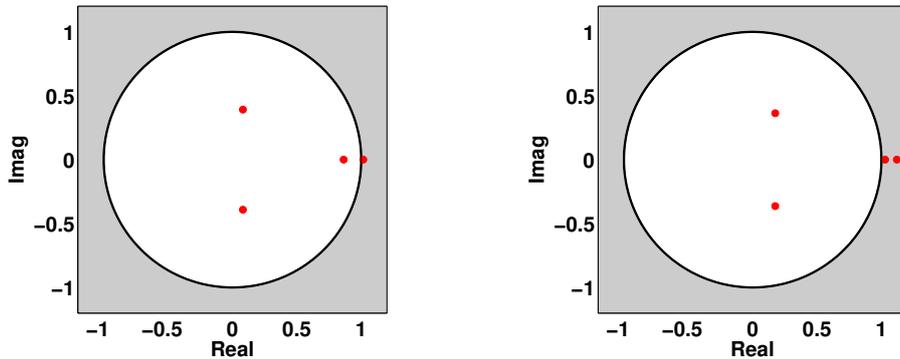
Figure 7.15: Floquets multipliers either side of the fold bifurcation, at $(x_f, max(u_f))$ values of $(0.239, 0.224)$ (left) and $(0.239, 0.244)$ (right). The fold bifurcation is caused by the Floquet multiplier crossing $+1$. The four largest Floquet multipliers are converged to $10^{-2}$ accuracy with the Arnoldi algorithm. The Floquet multiplier at $(+1,0)$ is the trivial one which defines a limit cycle.

at $x_f = 0.239$ and the subcritical period-doubling bifurcation at $x_f = 0.182$ create a bistable region, because between $0.182 < x_f < 0.239$ there are both stable period-1 limit cycles and stable period-2 limit cycles. Mode switching is possible in this parameter region.

### 7.4.3.2 Period-doubling bifurcation

A period-doubling (flip) bifurcation occurs when a Floquet multiplier crosses the unit circle at -1. The bifurcation creates a branch of period-2 limit cycles, and the period-1 limit cycle becomes unstable. Figure 7.16 shows the Floquet multipliers at a limit cycle either side of the period-doubling bifurcation in Figure 7.10, which clearly show the Floquet multiplier crossing -1.
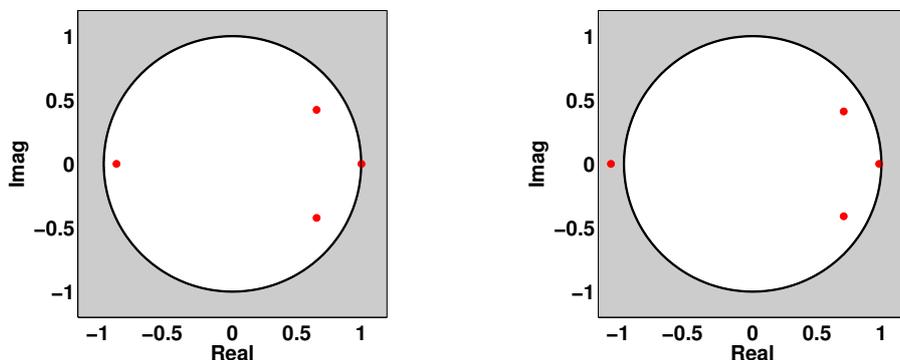


Figure 7.16: Floquets multipliers either side of the period-doubling bifurcation, at $x_f$ values of 0.193 (left) and 0.175 (right). The period-doubling bifurcation is caused by the Floquet multiplier crossing -1. The four largest Floquet multipliers are converged to $10^{-2}$ accuracy with the Arnoldi algorithm. The Floquet multiplier at $(+1,0)$ is the trivial one which defines a limit cycle.

A period-doubling bifurcation can also be seen from a time series (Figure 7.17). Starting at the unstable limit cycle just after the period-doubling bifurcation, the system grows exponentially away from the limit cycle, with the peaks forming a characteristic one-up, one-down pattern. Because the limit cycle was converged to a high tolerance, and because the unstable Floquet multiplier is only just outside the unit circle, the system requires a long time to reach the period-2 limit cycle.
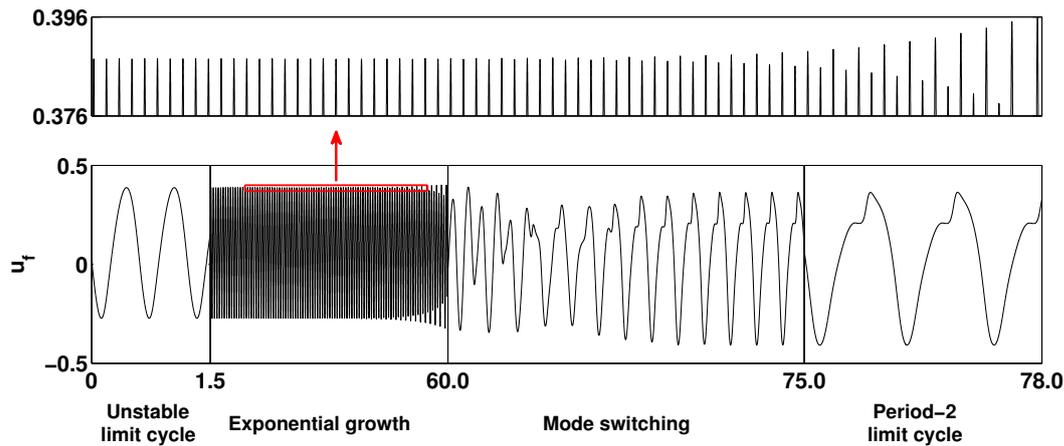


Figure 7.17: Time series of the system growing exponentially away from an unstable period-1 limit cycle just after a period-doubling bifurcation. The peaks form a characteristic one-up, one-down pattern. The first and the last boxes have the same scale, to show that the period-2 limit cycle has roughly twice the period of the unstable period-1 limit cycle.

The mode switching section of the time series in Figure 7.17 also shows that the period-doubling bifurcation is subcritical. Because the unstable limit cycle is only just beyond the period-doubling bifurcation, then if the bifurcation were supercritical, the system would quickly reach a stable period-2 limit cycle with a relatively low magnitude in the half frequency. In other words, the stable period-2 cycle would be very similar to the unstable period-1 cycle, with only a small one-up, one-down variation in the peaks and troughs.

At the period-doubling bifurcation, the Floquet mode that corresponds to the Floquet multiplier at -1 shows which coupled motion of the system is responsible for the bifurcation. This Floquet mode is shown schematically in Figure 7.18. At the frequency of the fundamental acoustic mode, the flame has two cusps on its surface. At half of this frequency, there should be only one cusp. The Floquet mode is a coupled motion with: (1) a flapping motion of the flame surface, where the tip and base of the flame move outwards and the middle of the flame moves inwards, coupled with (2) a variation in the velocity field every other cycle, and (3) a reduction in acoustic pressure in the duct and an increase in acoustic velocity in the duct before the flame.
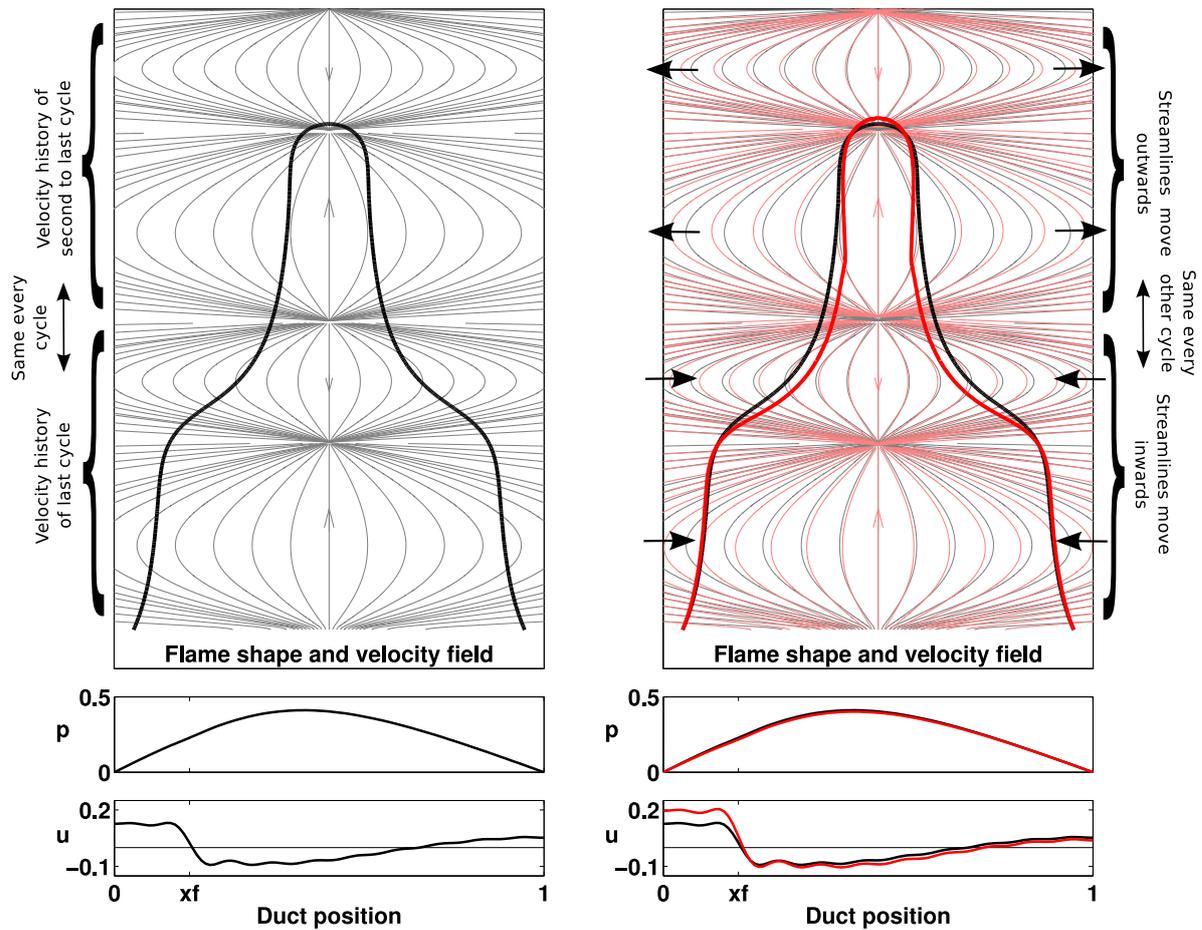
Figure 7.18: Floquet mode for the Floquet multiplier that causes the period-doubling bifurcation. The left hand image shows the flame shape and streamlines of the velocity field at a state on the limit cycle at the period-doubling bifurcation (aspect ratio 4). With the $y$-axis limits chosen, the velocity field in the lower half of the domain is repeated in the upper half of the domain - the velocity field is the history of the acoustic perturbation over the last two cycles; the last cycle corresponds to the lower half of the domain, the second to last cycle corresponds to the upper half of the domain. The right hand image shows the same as the left image (gray), superimposed with the state when perturbed a small amount in the direction of the Floquet mode that causes the period-doubling bifurcation (red). The coupled motion responsible for the period-doubling bifurcation is therefore: (1) a flapping motion of the flame surface, where the tip and base of the flame move outwards and the middle of the flame moves inwards, coupled with (2) a variation in the velocity field every other cycle, and (3) a reduction in acoustic pressure in the duct and an increase in acoustic velocity in the duct before the flame. The flame shape perturbation is scaled by a factor of two for clarity.

### 7.4.3.3 Neimark-Sacker bifurcation

A Neimark-Sacker (torus) bifurcation occurs when a complex pair of Floquet multipliers crosses the unit circle. The bifurcation creates a branch of quasiperiodic oscillations. A quasiperiodic oscillation has two incommensurate frequencies, and therefore forms a torus in phase space. Figure 7.19 shows the Floquet multipliers at a limit cycle either side of the Neimark-Sacker
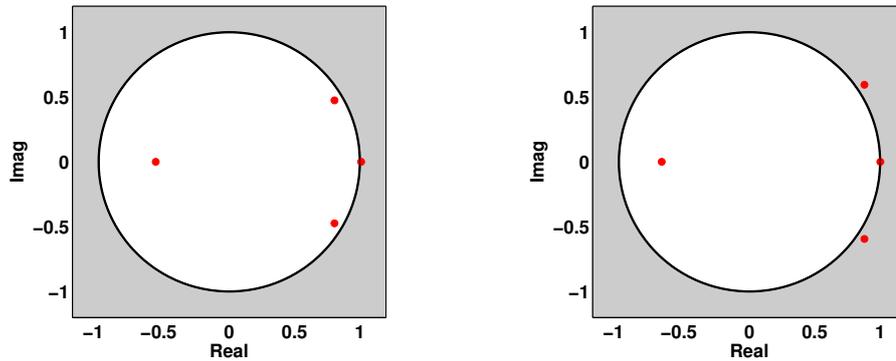
Figure 7.19: Floquets multipliers either side of the Neimark-Sacker bifurcation on Figure 7.8, at flame aspect ratio of 2.4 and $x_f$ values of 0.2 (left) and 0.175 (right). The Neimark-Sacker bifurcation is caused by the pair of Floquet multipliers crossing the unit circle. The four largest Floquet multipliers are converged to $10^{-2}$ accuracy with the Arnoldi algorithm. The Floquet multiplier at $(+1,0)$ is the trivial one which defines a limit cycle.

bifurcation, which clearly show the pair of Floquet multipliers crossing the unit circle.

A Neimark-Sacker bifurcation can also be seen from a time series. Starting at the unstable limit cycle just after the Neimark-Sacker bifurcation, the system grows exponentially away from the limit cycle, with the peaks oscillating at a second frequency. The ratio between these two frequencies is given by the argument of the complex eigenvalue pair, divided by $2\pi$. For the pair of Floquet multipliers in Figure 7.19, the ratio between the two frequencies is 10.57 (to 2d.p.). This can be seen in the time series of Figure 7.20, where the peaks oscillate with a period 10.57 times that of the limit cycle. It is important to note that if this number were rational, then the oscillation would be frequency locked and not quasiperiodic.
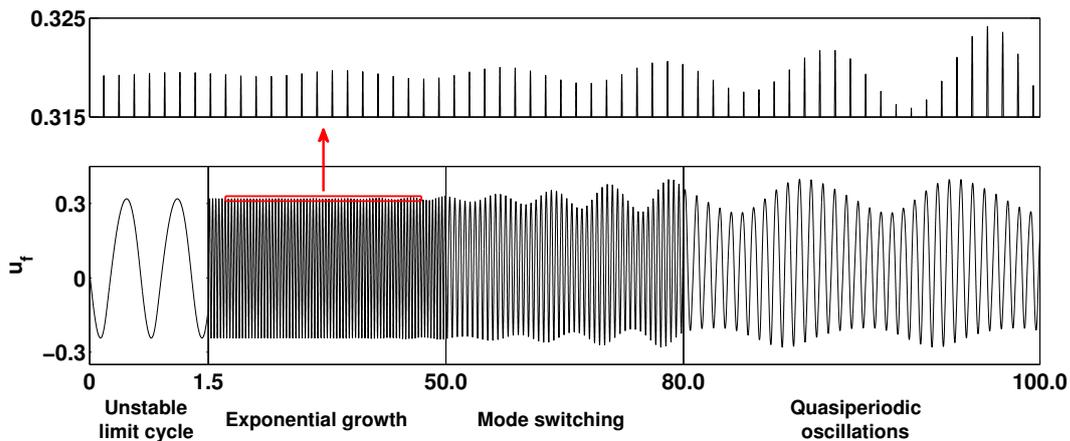


Figure 7.20: Time series of the system growing exponentially away from an unstable limit cycle just after a Neimark-Sacker bifurcation. The peaks oscillate at a second frequency that is defined by the argument of the pair of Floquet multipliers that cross the unit circle.
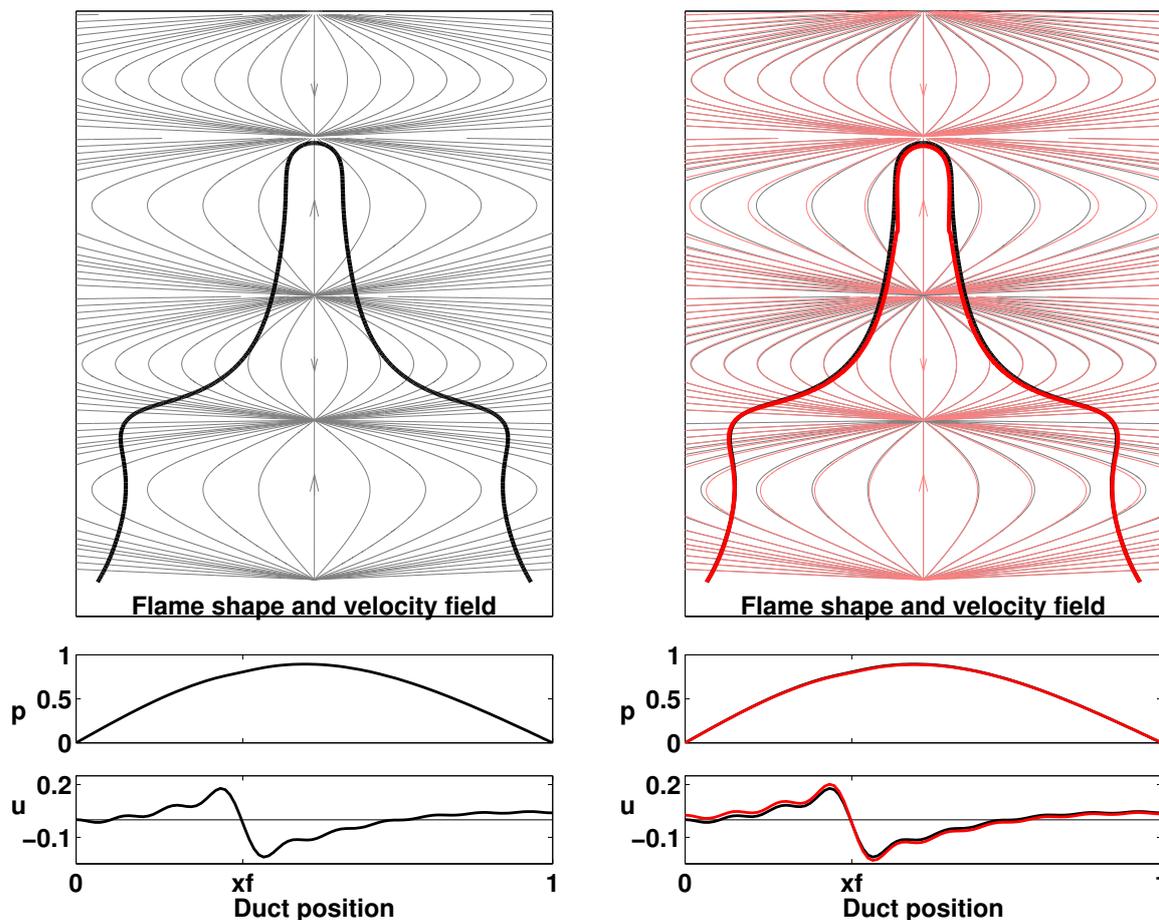
Figure 7.21: Real part of the Floquet mode for the complex Floquet multiplier pair that causes the Neimark-Sacker bifurcation. The coupled motion responsible for the Neimark-Sacker bifurcation is therefore: (1) a bulging motion of the flame surface, where the base of the flame move outwards and the tip of the flame moves inwards, coupled with (2) a variation in the velocity field with a period much greater than that of the limit cycle, and (3) a reduction in acoustic pressure in the duct and an increase in acoustic velocity in the duct before the flame. The changes in the velocity field are much harder to understand visually than the changes for the period-doubling bifurcation in Figure 7.18, because the velocity field changes slowly over 10.57 cycles.

At the Neimark-Sacker bifurcation, the Floquet mode that corresponds to the complex Floquet multiplier pair at the unit circle shows which coupled motion of the system is responsible for the bifurcation. Figure 7.21 shows the real part of this pair of Floquet multipliers. The Floquet mode is a coupled motion with: (1) a bulging motion of the flame surface, where the base of the flame moves outwards and the tip of the flame moves inwards, coupled with (2) a variation in the velocity field with a period much greater than that of the limit cycle, and (3) a reduction in acoustic pressure in the duct and an increase in acoustic velocity in the duct before the flame. The perturbation in the velocity field is much less clear to analyse than that

129

of the period-doubling bifurcation in Figure 7.18, because it does not affect the velocity history from previous cycles in a symmetric way (whereas the period-doubling bifurcation did). In the Neimark-Sacker case, the perturbation to the velocity history is changing both the peaks and troughs of the previous cycles, and the size of these changes varies at a frequency lower than that of the original limit cycle.

### 7.4.4 Numerical efficiency

Chapter 6 showed that the convergence of the GMRES solver is not dependent on the number of variables in the state vector. Instead, it is dependent on the number of bulk motions of the system.

Figure 7.22 shows the convergence to an unstable limit cycle ($x_f = 0.175$, $\beta = 2$) as a function of the number of periods that are timemarched. The continuation methods converge steadily to the limit cycle from a relatively poor initial guess, converging by a factor of 1000 in only 26 timemarches. Convergence tests have managed to reach a tolerance of $||\underline{x}(0)-\underline{x}(T)|| < 2 \times 10^{-6}$, which is a relative residual of $\frac{||\underline{x}(0)-\underline{x}(T)||}{||\underline{x}(0)||} < 2 \times 10^{-7}$. A tolerance of $||\underline{x}(0)-\underline{x}(T)|| < 5 \times 10^{-4}$ is chosen in this chapter to signify that a limit cycle has been found, however. At this level of tolerance, the difference between the flame shapes at $\underline{x}(0)$ and $\underline{x}(T)$ is everywhere less than 1% of the size of a grid cell. This tolerance is considered to be low enough to ensure that a limit cycle has been found, but high enough to ensure that computing bifurcation surfaces is fast.

The continuation methods converge by timemarching only a few cycles, but the real-time to create a bifurcation surface is governed by the speed of timemarching one cycle. The LSGEN2D
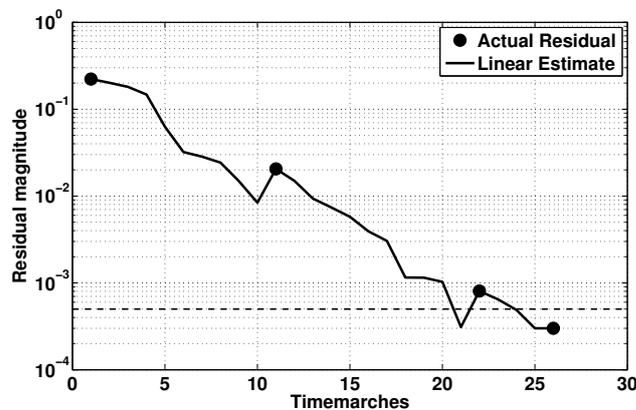


Figure 7.22: Typical convergence to a limit cycle in terms of the residual magnitude, $||\underline{x}(0)-\underline{x}(T)||_2$, and number of timemarches. The dots are the residual values after each Newton step and the line is the estimated residual within the solution of each Newton step.

code used in this chapter is several hundred times slower at timemarching than the diffusion flame code used in chapter 6. Despite the LSGEN2D code being more computationally expensive to timemarch, only around 14000 CPU hours were required to generate the bifurcation surfaces in section 7.4.1 and to analyse the stability of the limit cycles: this is equivalent to 80 CPU cores running for one week. Because the surface is composed of several 2D slices, the process of forming a surface is easily parallelisable. For relatively little computational cost, therefore, the continuation methods can characterise the nonlinear behaviour of the coupled system over a wide parameter range.

It should be noted that the convergence is not as fast as it could be. There are several immediate areas that would be worth further investigation: (i) Discretisations - the Chebyshev polar coordinate discretisation is not ideal, because the point spacing is too small near the edges of the flame, and it does work not for arbitrary flame shapes, such as in pinch-off or flashback. The Chebyshev discretisation will also be poorly conditioned - or will fail - when there is a sharp cusp near the centreline. For the Chebyshev discretisation, an immediate extension would be to use a mapping function to increase the point spacing near the edges of the flame. (ii) Accuracy of the reinitialisation routine - the reinitialisation routine is responsible for most of the timemarching noise. If the timemarching noise were reduced then the finite difference matrix-vector products would be more accurate. (iii) Smoothing - the flame shape will always be smooth after timemarching one cycle, but the output from the linear solver will not be perfectly smooth. A smoothing function could be applied after each GMRES solution to ensure that the next guess for a state on the limit cycle has a smooth flame shape. (iv) Dynamic phase condition - the phase condition could be altered for each limit cycle to ensure that there are no sharp cusps near the centreline, which would ensure that the Chebyshev polar coordinate discretisation is well conditioned.

## 7.5 Conclusions

Matrix-free continuation techniques were applied to a model of a ducted premixed flame. The flame model uses the kinematic G-equation, with a local level set solver. The premixed flame model has many attributes of equivalent experimental systems: the flame is axisymmetric, the flame speed is dependent on the curvature, the flame has sharp cusps, and the flame is capable of pinch-off, flashback and bulging at the burner lip. The ducted premixed flame model has been shown previously by Kashinath to exhibit limit cycle, period-$2^n$, quasiperiodic and chaotic behaviour [79], and to have many parameter regions that are multistable. These results show qualitatively the same phenomena observed in experiments by Kabiraj [20].

# 7. CONTINUATION ANALYSIS OF A DUCTED PREMIXED FLAME

The continuation techniques are used to efficiently find a surface of stable and unstable limit cycles, as two system parameters vary. The continuation methods also explicitly find period-doubling and Neimark-Sacker bifurcations, by examining the Floquet multipliers of the limit cycles. A separate surface of period-2 limit cycles was found emerging from a subcritical period-doubling bifurcation. This is the first computational thermoacoustic study where period-doubling and Neimark-Sacker bifurcations have been found. The Floquet modes are examined at the bifurcations to show the coupled flame-acoustic motions that are responsible for the qualitative changes in behaviour. The continuation methods can find unstable limit cycles easily, whereas many other techniques cannot. This is important because the unstable limit cycles are crucial for mode switching [2, 79] and for separating the basins of attraction of different attractors.

The monodromy matrix was used to define the Floquet multipliers of a limit cycle. The monodromy matrix for the premixed flame was not analysed in detail in this chapter, but it contains extra information that would be worthy of further investigation. First, the pseudospectra of the monodromy matrix - with a physically based norm - defines the transient growth around the limit cycle (section 6.5.2) which is important for mode switching and triggering. Because the diffusion flame of chapter 6 showed lower transient growth than that reported in the literature, it would be interesting to compare the transient growth for the premixed flame. Second, the acoustic-acoustic section of the monodromy matrix shows how much interaction there is between the acoustic modes - how much energy is transferred between modes during one cycle. The degree of coupling between the modes would help understand the physical nature of thermoacoustics.

The LSGEN2D code is able to capture the dynamics of a premixed flame under acoustic forcing, and the continuation methods are able to calculate any limit cycles and bifurcations when the flame model is coupled to an acoustic model. The results from the continuation methods could be compared with experiments in one of two ways: first, by comparing flame shapes and heat release responses; second, by comparing the self-excited behaviour and the bifurcation diagrams. With improvements to the velocity model and the acoustic model (see further work section), the method described in this chapter could be an effective means of predicting and analysing the nonlinear behaviour seen in experimental premixed flame systems.

# Chapter 8

# Gradient methods

After the work on the influence of noise (part I), I decided to focus on finding limit cycles of large thermoacoustic systems. I developed two different approaches in parallel because, at that time, it was not clear whether either or both would work. As it turns out, they both work and they have complementary features, which will benefit future research in this area. The method developed in this chapter would have been harder to implement for the large thermoacoustic systems in chapters 6 and 7, so the matrix-free techniques were used instead. Nevertheless, in this chapter, I describe these gradient methods because current developments in the field of algorithmic differentiation will make them easier to implement in the future.

Chapter 5 described an efficient method for finding limit cycles in large thermoacoustic systems. This method required the inexact solution of a series of linear equations, each of which has the standard form for multidimensional Newton iteration, $J\underline{\Delta x} = -\underline{r}$ . Newton iteration converges quadratically when started close to a solution (proportional to the second derivative). When started far from a solution, however, Newton iteration may diverge. The matrix-free methods of chapter 5 are therefore only useful when a good estimate of a limit cycle is known. In this chapter an alternative method of finding limit cycles is presented, which should be more stable when converging from poor initial guesses.

## 8.1 Introduction

When attempting to find a limit cycle in a large thermoacoustic system, a good estimate of a limit cycle may already be known. This estimated limit cycle may come from a known Hopf bifurcation, or from timemarching data that shows a feature that is nearly periodic. Stable fixed points and stable limit cycles can be found by timemarching. Unstable fixed points and unstable limit cycles are harder to find by timemarching, however, because the trajectories

diverge[1]. By extracting near-periodic features from timemarching data, and then attempting to converge to a limit cycle, it has been possible to find unstable limit cycles in a model of 2D turbulence [86]. This process is time-consuming, however, because it requires a lot of timemarching, and many near-periodic features are found that do not correspond to a nearby limit cycle.

In this chapter a method of finding limit cycles is presented which should be more stable when converging from poor initial guesses. This method formulates the problem as the minimisation of a scalar cost function. The simplest cost function, $\mathcal{J}$, is the length of the residual vector, $\mathcal{J} = \underline{r} \cdot \underline{r} = (\underline{x}(T) - \underline{x}(0)) \cdot (\underline{x}(T) - \underline{x}(0))$ [126, 127]. Gradient methods use the direction of the gradient of the cost function, $\mathrm{d}\mathcal{J}/\mathrm{d}\underline{x}(0)$, to iterate towards a nonlinear minimum. The gradient of the cost function, $\mathrm{d}\mathcal{J}/\mathrm{d}\underline{x}(0)$, defines the direction in which the cost function increases. The value of the cost function can therefore be reduced by heading in the opposite direction to the gradient. This is the foundation of all gradient based optimisation methods.

It is important to have an efficient method of finding the gradient of the cost function, because the gradient vector contains the derivative of the cost function with respect to each of the system variables, $\mathrm{d}\mathcal{J}/\mathrm{d}\underline{x}(0)$. In this chapter adjoint equations will be used to calculate the gradient of the cost function. As will be shown later in this chapter, for any system dimension, $N$, the adjoint equations provide the gradient of the cost function in two timemarches, one forwards and one backwards. Therefore when $N$ is large, adjoint methods are far more efficient at finding gradients than finite difference methods, which separately vary each variable in the state vector to evaluate $\mathrm{d}\mathcal{J}/\mathrm{d}\underline{x}_i(0)$. As $N$ increases, however, adjoint methods require large amounts of memory and elaborate checkpointing routines. As will be discussed later in this chapter, this memory requirement can be reduced by using multiple shooting methods.

Adjoint equations were first used to find limit cycles by Nakhla [126] and then later by Tadi [127], who independently derived the form of the standard shooting method described in the next section. Tadi additionally introduced higher order terms into the cost function that match the time derivatives at $\underline{x}(0)$ and $\underline{x}(T)$. Higher order terms of a different form will be described later in this chapter. Neither of these references, however, considered a Poincaré shooting method, which is derived in this chapter. Multiple shooting methods have also been implemented by Lan [128] and Boghosian [129].

This chapter begins with general derivations for the standard shooting and Poincaré shooting methods. The chapter then discusses modification of the cost function by normalisation and by inclusion of higher order terms. A general form of the second order adjoint is then

---

[1]The unstable fixed points and unstable limit cycles are unstable attractors, so trajectories initially converge towards them from directions that are stable. The trajectories will eventually diverge, however, in a direction that is unstable.

derived for the standard shooting method. A general form of the multiple shooting method is then derived to provide a gradient method equivalent to the multiple shooting method of chapter 6. The numerical methods are then demonstrated on the horizontal Rijke tube model.

### 8.1.1  Definition of inner products

Various inner products are used throughout this chapter. They are defined here as integrals in time and integrals over some measure of space ($z$):

$$\left\{\underline{f},\underline{g}\right\}_{t_1}^{t_2} \equiv \int_{t_1}^{t_2} \underline{f}(t) \cdot \underline{g}(t) \, \mathrm{d}t$$

$$\left\langle \underline{f},\underline{g} \right\rangle \equiv \underline{f} \cdot \underline{g}$$

$$\left[\underline{f},\underline{g}\right]_{t_1}^{t_2} \equiv \int_{t_1}^{t_2} \int \underline{f}(z,t) \cdot \underline{g}(z,t) \, dz \, dt$$

## 8.2  Standard shooting method

As in chapter 5, this derivation is for nonlinear systems whose evolution is governed by:

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

where $\underline{x}$ is the current state of the system and $\lambda$ are parameters. In the rest of this section, the parameter dependence of the system will be dropped from the notation for clarity. Limit cycles satisfy:

$$\underline{x}(0) = \underline{x}(T), \quad \left\{T \in \mathbb{R}^+ | T \neq 0\right\}, \tag{8.1}$$

For finding limit cycles, the cost function is defined as:

$$\mathcal{J} = ||\underline{x}(T) - \underline{x}_0||^2 = (\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0) \tag{8.2}$$

where $\underline{x}_0$ is the current guess for a state on the limit cycle.

The Lagrange equation is:

$$\mathcal{L} \equiv \mathcal{J} - \mathcal{C}$$

where $\mathcal{C}$ are the constraints. For finding limit cycles, there are two constraints: first, that the system evolves with $\frac{\mathrm{d}x}{\mathrm{d}t} = F(\underline{x})$ for all time; and second, that the current guess for a state on the limit cycle, $\underline{x}_0$, is the state at time $t = 0$, $\underline{x}_0 = \underline{x}(0)$. These constraints are functions of time and space but the Lagrangian is a scalar, therefore inner products of the constraints are

taken. If these constraints are put into the format $\mathcal{C} = 0$ (for reasons explained later), and inner products are taken, then the Lagrangian equation becomes:

$$\mathcal{L} \equiv \mathcal{J} - \int_0^T \underline{\theta} \cdot \left( \frac{\mathrm{d}\underline{x}}{\mathrm{d}t} - F(\underline{x}) \right) \mathrm{d}t - \underline{G} \cdot (\underline{x}(0) - \underline{x}_0) \tag{8.3}$$

where $\underline{\theta}$ and $\underline{G}$ are Lagrange multipliers. The adjoint variables, $\underline{\theta}$, are the Lagrange multipliers of the direct governing equations. They are time dependent. In the literature a dagger is often used to denote the adjoint state ($\underline{\theta} = \underline{x}^\dagger$), but in this chapter $\underline{\theta}$ is used for clarity.

With the inner product definitions of section 8.1.1, equation (8.3) can be rewritten as:

$$\mathcal{L} \equiv \mathcal{J} - \left[ \underline{\theta}, \frac{\mathrm{d}\underline{x}}{\mathrm{d}t} - F(\underline{x}) \right]_0^T - \langle \underline{G}, \underline{x}(0) - \underline{x}_0 \rangle \tag{8.4}$$

The Lagrangian ($\mathcal{L}$) is therefore a function of the starting state, the direct variables and the adjoint variables, $\mathcal{L} = f(\underline{x}_0, \underline{x}(\underline{x}_0, t), \underline{\theta}(\underline{x}_0, \underline{x}, t))$. Gradient methods require the gradient of the cost function with respect to the starting state, $\mathrm{d}\mathcal{J}/\mathrm{d}\underline{x}(0)$. Each component of the gradient contains two terms:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} = \frac{\partial \mathcal{J}}{\partial \underline{x}_{0k}} + \sum_i \frac{\partial \mathcal{J}}{\partial \underline{x}_i(\underline{x}_0, t)} \frac{\partial \underline{x}_i(\underline{x}_0, t)}{\partial \underline{x}_{0k}}$$

The first term is simple to evaluate but the second term is not, because the $\frac{\partial \underline{x}_i(\underline{x}_0,t)}{\partial \underline{x}_{0k}}$ term is the result of a nonlinear integration in time and therefore cannot be expressed analytically. Adjoint equations use the Lagrangian equation, and a mathematical trick, to find the gradient of the cost function without having to explicitly calculate this unknown term.

Using the chain rule, differentiating the Lagrangian with respect to one variable in the initial state gives the following, where the subscript $i$ is used to denote the $i^{th}$ variable in the state vector (but the current guess, $\underline{x}_0 \neq \underline{x}_i|_{i=0}$) :

$$\begin{aligned}
\mathcal{J} &= \mathcal{L} + \mathcal{C} \\
\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} &= \frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}} + \sum_i \frac{\partial \mathcal{L}}{\partial \underline{x}_i(\underline{x}_0, t)} \frac{\partial \underline{x}_i(\underline{x}_0, t)}{\partial \underline{x}_{0k}} + \sum_i \frac{\partial \mathcal{L}}{\partial \underline{\theta}_i(\underline{x}_0, \underline{x}, t)} \frac{\partial \underline{\theta}_i(\underline{x}_0, \underline{x}, t)}{\partial \underline{x}_{0k}} \cdots \\
&\quad + \frac{\partial \mathcal{C}}{\partial \underline{x}_{0k}} + \sum_i \frac{\partial \mathcal{C}}{\partial \underline{x}_i(\underline{x}_0, t)} \frac{\partial \underline{x}_i(\underline{x}_0, t)}{\partial \underline{x}_{0k}} + \sum_i \frac{\partial \mathcal{C}}{\partial \underline{\theta}_i(\underline{x}_0, \underline{x}, t)} \frac{\partial \underline{\theta}_i(\underline{x}_0, \underline{x}, t)}{\partial \underline{x}_{0k}}
\end{aligned} \tag{8.5}$$

Because the system is nonlinear, the $\frac{\partial \underline{x}_i(\underline{x}_0,t)}{\partial \underline{x}_{0k}}$ and $\frac{\partial \underline{\theta}_i(\underline{x}_0,t)}{\partial \underline{x}_{0k}}$ terms are unknown. If it were possible to fix $\frac{\partial \mathcal{L}}{\partial \underline{x}_i(\underline{x}_0,t)} = \frac{\partial \mathcal{L}}{\partial \underline{\theta}_i(\underline{x}_0,\underline{x},t)} = 0$, and construct all of the constraints so that $\mathcal{C} = 0$, however, then these unknown terms are multiplied by zero, and equation (8.5) would reduce

to:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} = \frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}} \tag{8.6}$$

By the construction of equation (8.4), $\frac{\partial \mathcal{L}}{\partial \underline{\theta}_i(\underline{x}_0, \underline{x}, t)}$ is always zero because the governing equation constraint is zero, $\frac{\mathrm{d}x}{\mathrm{d}t} - F(\underline{x}) = 0$. As will be seen in the following derivation, setting $\frac{\partial \mathcal{L}}{\partial \underline{x}_i(\underline{x}_0, t)} = 0$ defines particular boundary conditions for the adjoint variables, and also defines how they evolve in time. These are the adjoint equations.

The first step in the derivation of the adjoint equations is to integrate the second term of equation (8.4) by parts, which gives two extra terms. These terms will later be used to define the boundary conditions for the adjoint state:

$$\mathcal{L} = \mathcal{J} - \left[\frac{\mathrm{d}\underline{\theta}}{\mathrm{d}t}, -\underline{x}\right]_0^T - [\underline{\theta}, -F(\underline{x})]_0^T - \langle \underline{\theta}(T), \underline{x}(T) \rangle + \langle \underline{\theta}(0), \underline{x}(0) \rangle - \langle \underline{G}, \underline{x}(0) - \underline{x}_0 \rangle$$

The second step in the derivation is to differentiate the Lagrangian with respect to one of the direct variables:

$$\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_k}, \delta \underline{x}_k\right]_0^T = \langle 2(\underline{x}_k(T) - \underline{x}_{k0}), \delta \underline{x}_k(T) \rangle + \left[\frac{\mathrm{d}\underline{\theta}_k}{\mathrm{d}t}, \delta \underline{x}_k\right]_0^T + \left[\underline{\theta}, \frac{\partial F(\underline{x})}{\partial \underline{x}_k} \delta \underline{x}_k\right]_0^T \ldots$$
$$- \langle \underline{\theta}_k(T), \delta \underline{x}_k(T) \rangle + \langle \underline{\theta}_k(0), \delta \underline{x}_k(0) \rangle - \langle \underline{G}_k, \delta \underline{x}_k(0) \rangle \tag{8.7}$$

The adjoint formulation must set $\frac{\partial \mathcal{L}}{\partial \underline{x}(\underline{x}_0, t)} = 0$, to agree with equation (8.6). Summing components of $\delta \underline{x}_k$, and setting them equal to zero gives an evolution equation for each of the adjoint variables, $\underline{\theta}_k$:

$$\frac{\mathrm{d}\underline{\theta}_k}{\mathrm{d}t} + \sum_i \underline{\theta}_i \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k} = 0 \tag{8.8}$$

Summing the components at time $t = T$ and setting them equal to zero gives the boundary condition:

$$\underline{\theta}_k(T) = 2(\underline{x}_k(T) - \underline{x}_{0k}) \tag{8.9}$$

and the equivalent at time $t = 0$ gives:

$$\underline{\theta}_k(0) - \underline{G}_k = 0$$

The third step in the derivation is to differentiate the Lagrangian with respect to one of the variables in the starting state:

$$\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}}, \delta \underline{x}_{0k}\right]_0^T = \langle -2(\underline{x}_k(T) - \underline{x}_{0k}), \delta \underline{x}_{0k} \rangle + \langle \underline{G}_k, \delta \underline{x}_{0k} \rangle \tag{8.10}$$

From equation (8.6), the gradient of the cost function is equal to $\frac{\partial \mathcal{L}}{\partial \underline{x}_0}$:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} = \frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}} = \underline{G}_k - 2(\underline{x}_k(T) - \underline{x}_{0k}) = \underline{\theta}_k(0) - 2(\underline{x}_k(T) - \underline{x}_{0k}) \tag{8.11}$$

Therefore the gradient of the cost function is:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_0} = \underline{\theta}(0) - 2(\underline{x}(T) - \underline{x}_0) \tag{8.12}$$

The equation for the gradient in equation (8.12) requires the difference between the start and end of the loop, based on the current guess for $\underline{x}_0$ and $T$, and requires the adjoint state at time $0$, $\underline{\theta}_k(0)$. The only known adjoint state is at $t = T$ from equation (8.9), but the time evolution of the adjoint state is defined by equation (8.8). The evaluation of the gradient therefore requires two separate timemarches: first, the direct equations must be marched forward in time from $\underline{x}(0)$ to $\underline{x}(T)$, second, the adjoint equations must be marched backwards in time from $\underline{\theta}(T)$ to $\underline{\theta}(0)$.

If the direct governing equation is nonlinear then the adjoint equations at time $t$ require knowledge of $\underline{x}(t)$, because $\sum_k \sum_i \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k} = f(\underline{x})$. Conversely, if the governing equation is linear then the adjoint equations at time $t$ do not require knowledge of $\underline{x}(t)$, because $\sum_k \sum_i \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k}$ is a constant. When finding limit cycles in nonlinear systems, the values of $\underline{x}(t)$ must therefore be stored in memory or recalculated during the backwards timemarch. For large systems checkpointing algorithms are used, which are effective but significantly increase the complexity and memory requirements of the calculations.

To find limit cycles with a standard shooting formulation, the value of $\mathrm{d}\mathcal{J}/\mathrm{d}T$ is also required.

$$\mathcal{J} = ||\underline{x}(T) - \underline{x}_0||^2 = (\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0)$$

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T} = \frac{\partial \mathcal{J}}{\partial \underline{x}(T)} \cdot \frac{\partial \underline{x}(T)}{\partial T}$$

where:

$$\begin{aligned}
\frac{\partial \underline{x}(T)}{\partial T} &= \frac{\partial}{\partial T} \int_0^T \underline{\dot{x}} dt \\
&= \frac{\int_0^{T+\epsilon} \underline{\dot{x}} dt - \int_0^T \underline{\dot{x}} dt}{\epsilon} \\
&= \underline{\dot{x}}(T) = F(\underline{x}(T))
\end{aligned}$$

Therefore the gradient of the cost function with respect to time is [126]:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T} = 2\left(\underline{x}(T) - \underline{x}_0\right) \cdot \underline{\dot{x}}(T)$$

When the combined gradient, $\left[\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_0}; \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T}\right]$, is passed to an optimisation algorithm, poor convergence (or instability) may occur if $\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T}$ is a different order of magnitude to the other components of the gradient. This can be mitigated by weighting the initial conditions, $\underline{x}_0$. A vector of weights, rather than a single weight, is recommended when the elements of $\underline{x}_0$ are of different magnitudes [127]. When a Poincaré plane formulation is used, the gradient with respect to $T$ is not required (section 8.3).

## 8.2.1 Adjoint looping algorithm

1. Given a starting state $\underline{x}_0$ and guess for the period $T$, integrate forward to $t = T$ with the direct equations:

$$\underline{x}(0) = \underline{x}_0$$

$$\frac{\mathrm{d}\underline{x}}{\mathrm{d}t} = F(\underline{x})$$

2. Initialize the adjoint variables at $t = T$ with:

$$\underline{\theta}_k(T) = 2(\underline{x}_k(T) - \underline{x}_{0k})$$

3. Integrate backward in time to $t = 0$ with the adjoint equations:

$$\frac{\mathrm{d}\underline{\theta}_k}{\mathrm{d}t} + \sum_i \underline{\theta}_i \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k} = 0$$

4. Calculate the gradient information at $t = 0$ with:

$$\begin{aligned}
\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} &= \underline{\theta}_k(0) - 2(\underline{x}_k(T) - \underline{x}_{0k}) \\
\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T} &= 2\left(\underline{x}(T) - \underline{x}_0\right) \cdot \underline{\dot{x}}(T)
\end{aligned}$$

5. Pass the gradient information to an optimisation algorithm, which calculates a new initial starting state $\underline{x}_0$ and a new guess for the period $T$. Optimisation algorithms will be discussed later in the chapter.

6. Return to step 1 until $\mathcal{J} < tol$, where $tol$ is some positive convergence value, normally some factor of $||\underline{x}_0||$.

### 8.2.1.1 Phase condition

The cost function, $\mathcal{J} = (\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0)$, is zero at every point on the limit cycle, therefore every point on the limit cycle is a minimum. For some minimisation algorithms to converge efficiently, a single minimum point is required. This can be achieved by adding a phase condition to the cost function. The simplest phase condition fixes the value of a particular variable that is known (*a priori*) to change during a limit cycle. In a simple thermoacoustic system, with time varying amplitudes of acoustic modes, one might fix the instantaneous amplitude of the fundamental velocity or pressure mode to be zero.

The cost function with this phase condition is:

$$\mathcal{J} = (\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0) + \alpha \left( \underline{x}_{0ph} - \theta \right)^2 \tag{8.13}$$

where $ph$ is the index of the variable that is fixed to be equal to some constant, $\theta$, which the variable is known (*a priori*) to take during the limit cycle. The scalar $\alpha$ is a weighting between the periodicity and phase terms. The phase condition adds a term to the gradient for the fixed variable:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{ph0}} = \underline{\theta}_{ph}(0) - 2(\underline{x}_{ph}(T) - \underline{x}_{0ph}) + 2\alpha \left( \underline{x}_{0ph} - \theta \right)$$

## 8.3 Poincaré shooting method

The adjoint looping algorithm for a Poincaré shooting method is similar to that derived above for the standard shooting method. With a Poincaré shooting method (section 5.2.2), the direct timemarching is stopped at time $T_p$ when the hyperplane is reached, therefore the adjoint state is initialised at time $T_p$, $\underline{\theta}_k(T_p) = 2(\underline{x}_k(T_p) - \underline{x}_{0k})$. The adjoint equations are then marched backwards in time from $\underline{\theta}(T_p)$ to $\underline{\theta}(0)$. Extra terms must be included when evaluating $\mathrm{d}\mathcal{J}/\mathrm{d}\underline{x}_0$, however, because $T_p = f(\underline{x}_0)$:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} = \frac{\partial \mathcal{J}}{\partial \underline{x}(T_p)} \cdot \frac{\partial \underline{x}(T_p)}{\partial T_p} \frac{\partial T_p}{\partial \underline{x}_{0k}} + \frac{\partial \mathcal{J}}{\partial \underline{x}_{0k}} \tag{8.14}$$

The exact value of $\partial T_p / \partial \underline{x}_{0k}$ depends on the how the hyperplane is defined in the Poincaré shooting method. If the plane is defined by the starting state, $\underline{x}_0$, and the normal is defined by $\underline{\dot{x}}_0$, then the normal also changes with $\underline{x}_0$, whereas if the hyperplane is defined by a state from a previous converged solution, $\underline{x}_0^*$, and the normal is defined by $\underline{\dot{x}}_0^*$, then the normal does not also change with $\underline{x}_0$. When the hyperplane is defined by $\underline{x}_0$, the Poincaré shooting formulation converges well if $\underline{x}_0$ is close to a limit cycle, but may not converge if $\underline{x}_0$ is far from a limit cycle, because the trajectory may not recross the plane defined by $\underline{\dot{x}}_0$.

### 8.3.1 Hyperplane defined by the starting state

In this section, the first part of the derivation assumes a hyperplane that varies with $\underline{x}_0$. The second part of the derivation simplifies the first part to derive the formulation with a hyperplane that is fixed by $\underline{x}_0^*$. The crossing of the hyperplane is defined by:

$$\dot{\underline{x}}_0 \cdot (\underline{x}(T_p) - \underline{x}_0) = 0 \qquad \text{or} \qquad \dot{\underline{x}}_0^* \cdot (\underline{x}(T_p) - \underline{x}_0^*) = 0 \tag{8.15}$$

When the hyperplane varies with $\underline{x}_0$, equation (8.15) must still be obeyed when starting from $\underline{x}_0 + \delta\underline{x}_{0k}$:

$$\left( F(\underline{x}_0) + \frac{\partial F(\underline{x}_0)}{\partial \underline{x}_{0k}} \delta\underline{x}_{0k} \right) \cdot \tag{8.16}$$
$$\left( \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}} \delta\underline{x}_{0k} + F(\underline{x}(T_p)) \frac{\partial T_p}{\partial \underline{x}_{0k}} \delta\underline{x}_{0k} - \frac{d\underline{x}_0}{d\underline{x}_{0k}} \delta\underline{x}_{0k} + (\underline{x}(T_p) - \underline{x}_0) \right) = 0$$

Because the normal to the hyperplane is perpendicular to the residual in the hyperplane, $F(\underline{x}_0) \cdot (\underline{x}(T_p) - \underline{x}_0) = 0$, then using only first order terms in $\delta\underline{x}_{0k}$, equation (8.16) reduces to:

$$F(\underline{x}_0) \cdot \left( \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}} + F(\underline{x}(T_p)) \frac{\partial T_p}{\partial \underline{x}_{0k}} - \frac{d\underline{x}_0}{d\underline{x}_{0k}} \right) + \frac{\partial F(\underline{x}_0)}{\partial \underline{x}_{0k}} \cdot (\underline{x}(T_p) - \underline{x}_0) = 0 \tag{8.17}$$

The $F(\underline{x}_0) \cdot \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}}$ term is the dot product of $F(\underline{x}_0)$ with the $k^{th}$ column of the monodromy matrix, or alternatively, it is the $k^{th}$ row of the matrix-vector product $M^T F(\underline{x}_0)$. This matrix-vector product can be evaluated by setting $\underline{\theta}_p(T_p) = F(\underline{x}_0)$ and integrating the adjoint equations backwards in time, to get the matrix-vector product, $M^T F(\underline{x}_0) = \underline{\theta}_p(0)$.

Rearranging equation (8.17) expresses how the crossing time depends on a variable in the starting state:

$$\frac{\partial T_p}{\partial \underline{x}_{0k}} = \frac{F(\underline{x}_0) \cdot \frac{d\underline{x}_0}{d\underline{x}_{0k}} - F(\underline{x}_0) \cdot \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}} - \frac{\partial F(\underline{x}_0)}{\partial \underline{x}_{0k}} \cdot (\underline{x}(T_p) - \underline{x}_0)}{F(\underline{x}_0) \cdot F(\underline{x}(T_p))}$$

Expanding equation (8.14) to include the terms derived in equation (8.11), the gradient becomes:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} = \underline{\theta}_k(0) - 2(\underline{x}_k(T_p) - \underline{x}_{0k}) + 2(\underline{x}(T_p) - \underline{x}_0) \cdot \dot{\underline{x}}(T_p) \frac{\partial T_p}{\partial \underline{x}_{0k}}$$

The $\frac{\partial T_p}{\partial \underline{x}_{0k}}$ term requires the evaluation of $F(\underline{x}_0) \cdot \frac{\partial \underline{x}(T)}{\partial \underline{x}_{0k}}$, which can be calculated by integrating the adjoint equation backwards in time. It is not necessary to integrate the adjoint

equations separately to find $\underline{\theta}_k(0)$ and $F(\underline{x}_0) \cdot \frac{\partial \underline{x}(T)}{\partial \underline{x}_{0k}}$, however. The two values can be evaluated in one backwards timemarch, if the adjoint boundary condition $\underline{\theta}(T_p)$ is modified:

$$
\begin{aligned}
\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} &= \underline{\theta}_k(0) - 2(\underline{x}_k(T_p) - \underline{x}_{0k}) + \left( \frac{2(\underline{x}(T_p) - \underline{x}_0) \cdot \underline{\dot{x}}(T_p)}{F(\underline{x}_0) \cdot \underline{\dot{x}}(T_p)} \right) \times \\
&\quad \left( F(\underline{x}_0) \cdot \frac{d\underline{x}_0}{d\underline{x}_{0k}} - F(\underline{x}_0) \cdot \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}} - \frac{\partial F(\underline{x}_0)}{\partial \underline{x}_{0k}} \cdot (\underline{x}(T_p) - \underline{x}_0) \right) \\
&= \underline{\theta}_k(0) - 2(\underline{x}_k(T_p) - \underline{x}_{0k}) - \left( \frac{2(\underline{x}(T_p) - \underline{x}_0) \cdot \underline{\dot{x}}(T_p)}{F(\underline{x}_0) \cdot \underline{\dot{x}}(T_p)} \right) F(\underline{x}_0) \cdot \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}} \dots \\
&\quad + \left( \frac{2(\underline{x}(T_p) - \underline{x}_0) \cdot \underline{\dot{x}}(T_p)}{F(\underline{x}_0) \cdot \underline{\dot{x}}(T_p)} \right) \left( F_k(\underline{x}_0) - \frac{\partial F(\underline{x}_0)}{\partial \underline{x}_{0k}} \cdot (\underline{x}(T_p) - \underline{x}_0) \right) \\
&= \underline{\theta}_{ck}(0) - 2(\underline{x}_k(T_p) - \underline{x}_{0k}) \dots \\
&\quad + \left( \frac{2(\underline{x}(T_p) - \underline{x}_0) \cdot \underline{\dot{x}}(T_p)}{F(\underline{x}_0) \cdot \underline{\dot{x}}(T_p)} \right) \left( F_k(\underline{x}_0) - \frac{\partial F(\underline{x}_0)}{\partial \underline{x}_{0k}} \cdot (\underline{x}(T_p) - \underline{x}_0) \right) \quad (8.18)
\end{aligned}
$$

where the combined adjoint boundary condition is:

$$
\underline{\theta}_{ck}(T_p) = 2(\underline{x}_k(T_p) - \underline{x}_{0k}) - \left( \frac{2(\underline{x}(T_p) - \underline{x}_0) \cdot \underline{\dot{x}}(T_p)}{F(\underline{x}_0) \cdot \underline{\dot{x}}(T_p)} \right) F_k(\underline{x}_0) \quad (8.19)
$$

To summarise, when using the Poincaré shooting method, and a formulation where the hyperplane is defined by $\underline{x}_0$, there are three differences with the standard shooting adjoint formulation. First, the forward timemarch must stop exactly at $T_p$, where the trajectory crosses the hyperplane. Iteration may be required to find $T_p$ exactly. Second, the adjoint boundary condition at time $T_p$ in equation (8.19) has one more term than equation (8.9). Third, the gradient evaluation in equation (8.18) has one more term than equation (8.12). These extra terms arise because the crossing time is dependent on the initial condition, and the hyperplane is dependent on the initial condition.

### 8.3.2 Hyperplane defined by a fixed state

When the hyperplane is defined by a fixed state, $\underline{x}^*$, the $+\frac{\partial F(\underline{x}_0)}{\partial \underline{x}_{0k}} \delta \underline{x}_{0k}$ and $-\frac{d\underline{x}_0}{d\underline{x}_{0k}} \delta \underline{x}_{0k}$ terms from equation (8.17) have no equivalent, and the equation for crossing the hyperplane becomes:

$$
F(\underline{x}_0^*) \cdot \left( \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}} \delta \underline{x}_{0k} + F(\underline{x}(T_p)) \frac{\partial T_p}{\partial \underline{x}_{0k}} \delta \underline{x}_{0k} + (\underline{x}(T_p) - \underline{x}_0^*) \right) = 0 \quad (8.20)
$$

The last term is zero because the residual on the plane, $(\underline{x}(T_p) - \underline{x}_0^*)$, is perpendicular to the plane normal, $F(\underline{x}_0^*)$. Therefore:

$$
F(\underline{x}_0^*) \cdot \left( \frac{\partial \underline{x}(T_p)}{\partial \underline{x}_{0k}} + F(\underline{x}(T_p)) \frac{\partial T_p}{\partial \underline{x}_{0k}} \right) = 0
$$

$$\therefore \frac{\partial T_p}{\partial \underline{x}_{0k}} = \frac{-F(\underline{x}_0^*) \cdot \frac{\partial \underline{x}(T)}{\partial \underline{x}_{0k}}}{F(\underline{x}_0^*) \cdot F(\underline{x}(T_p))}$$

As in equation (8.19), the $F(\underline{x}_0^*) \cdot \frac{\partial \underline{x}(T)}{\partial \underline{x}_{0k}}$ term can be found by timemarching the adjoint equations backwards, and it can be incorporated into the adjoint boundary condition as before:

$$
\begin{aligned}
\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} &= \underline{\theta}_k(0) - 2(\underline{x}_k(T_p) - \underline{x}_{0k}) + 2(\underline{x}(T_p) - \underline{x}_0) \cdot \underline{\dot{x}}(T_p) \frac{-F(\underline{x}_0^*) \cdot \frac{\partial \underline{x}(T)}{\partial \underline{x}_{0k}}}{F(\underline{x}_0^*) \cdot F(\underline{x}(T_p))} \\
&= \underline{\theta}_{ck}(0) - 2(\underline{x}_k(T_p) - \underline{x}_{0k})
\end{aligned}
$$

where the combined adjoint boundary condition is:

$$\underline{\theta}_{ck}(T_p) = 2(\underline{x}_k(T_p) - \underline{x}_{0k}) - \left( \frac{2(\underline{x}(T_p) - \underline{x}_0) \cdot \underline{\dot{x}}(T_p)}{F(\underline{x}_0^*) \cdot F(\underline{x}(T_p))} \right) F_k(\underline{x}_0^*)$$

To summarise, when using the Poincaré shooting method, and a formulation where the hyperplane is fixed by $\underline{x}_0^*$, there are two differences with the standard shooting adjoint formulation. First, the forward timemarch must stop exactly at $T_p$, where the trajectory crosses the hyperplane. Iteration may be required to find $T_p$ exactly. Second, the adjoint boundary condition at time $T_p$ in equation (8.19) has one more term than equation (8.9). This extra terms arises because the crossing time is dependent on the initial condition.

For good convergence of the optimisation routine with a fixed Poincaré plane, it is important that the initial starting state lies on the starting Poincaré plane, and that the gradient is projected onto the plane each iteration:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_0} \bigg|_{PLANE} = \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_0} - \hat{\underline{n}} \left( \hat{\underline{n}} \cdot \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_0} \right)$$

## 8.4 Normalised cost function

The cost function specified in section 8.2, $\mathcal{J} = (\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0)$, tends to zero at a limit cycle or fixed point. In an oscillatory system it also tends to zero, however, when the amplitude of an oscillation tends to zero, simply because the system is acting at a smaller scale rather than because the system is nearer a periodic state. To make the cost function tend to zero only near limit cycles or fixed points, a normalised cost function can be defined:

$$\mathcal{J}_{norm} = \frac{(\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0)}{\underline{x}_0 \cdot \underline{x}_0 + \epsilon}$$

where $\epsilon$ is a small positive number that prevents the cost function becoming infinite at $\underline{x}_0 = 0$.

The normalised cost function changes both the adjoint boundary condition at $t = T$ and the gradient evaluation condition:

$$
\begin{aligned}
\underline{\theta}_k(T) &= \frac{2(\underline{x}_k(T) - \underline{x}_{0k})}{\underline{x}_0 \cdot \underline{x}_0 + \epsilon} \\
\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} &= \underline{\theta}_k(0) - \frac{2(\underline{x}_k(T) - \underline{x}_{0k})}{\underline{x}_0 \cdot \underline{x}_0 + \epsilon} - \frac{2\underline{x}_{0k}(\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0)}{(\underline{x}_0 \cdot \underline{x}_0 + \epsilon)^2} \\
\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T} &= \frac{2(\underline{x}(T) - \underline{x}_0) \cdot \underline{\dot{x}}(T)}{\underline{x}_0 \cdot \underline{x}_0 + \epsilon}
\end{aligned}
$$

## 8.5 Higher order cost functions

The convergence of gradient methods depends on the shape of the cost function about the local minimum; a smoother cost function space will aid convergence. A higher order cost function can be created by using additional limit cycle constraints. The simplest higher order cost function includes time differentials [127], because a limit cycle requires $\underline{x}(0) = \underline{x}(T)$ but also $\underline{\dot{x}}(0) = \underline{\dot{x}}(T)$, $\underline{\ddot{x}}(0) = \underline{\ddot{x}}(T)$ etc. . The higher order cost function has additional terms:

$$
\mathcal{J} = (\underline{x}(T) - \underline{x}_0) \cdot (\underline{x}(T) - \underline{x}_0) + \alpha_1 (\underline{\dot{x}}(T) - \underline{\dot{x}}_0) \cdot (\underline{\dot{x}}(T) - \underline{\dot{x}}_0) + \ldots
$$

A higher order cost function of this type may lead to numerical inaccuracy because the time differentials of a limit cycle may increase in magnitude as more differentials are taken. Some weighting factors, $\alpha_n \in \mathbb{R}^+$, are therefore required to improve the numerical conditioning of the problem. These weighting factors may be based on a characteristic timescale of the system, $\alpha_n = \Delta t^n$. In practice, even with weighting factors, numerical instability may arise at higher orders near the minimum if there are errors in the gradient evaluation. The adjoint formulation for this cost function is almost identical to the derivation in section 8.2, but with extra terms in the adjoint boundary condition and gradient evaluation equations.

Figure 8.1a shows that an alternative method is to define the cost function over a path, where the cost function is the sum of the norms of $n$ residual vectors, each spaced $dt$ apart. The cost function has the form of equation (8.21) . Including multiple states is similar to including time differentials, but with fewer complications arising from different scales.

$$
\begin{aligned}
\mathcal{J} &= (\underline{x}(T) - \underline{x}(0)) \cdot (\underline{x}(T) - \underline{x}(0)) \ldots \\
&\quad + (\underline{x}(T + dt) - \underline{x}(dt)) \cdot (\underline{x}(T + dt) - \underline{x}(dt)) \\
&\quad + \ldots + (\underline{x}(T + ndt) - \underline{x}(ndt)) \cdot (\underline{x}(T + ndt) - \underline{x}(ndt)) \quad (8.21)
\end{aligned}
$$

Figure 8.1b shows the equivalent cost function for the Poincaré shooting method, where

(a) The cost function is the sum of norms of residual vectors (blue).

(b) The cost function is sum of norms of residual vectors on Poincaré hyperplanes (blue).
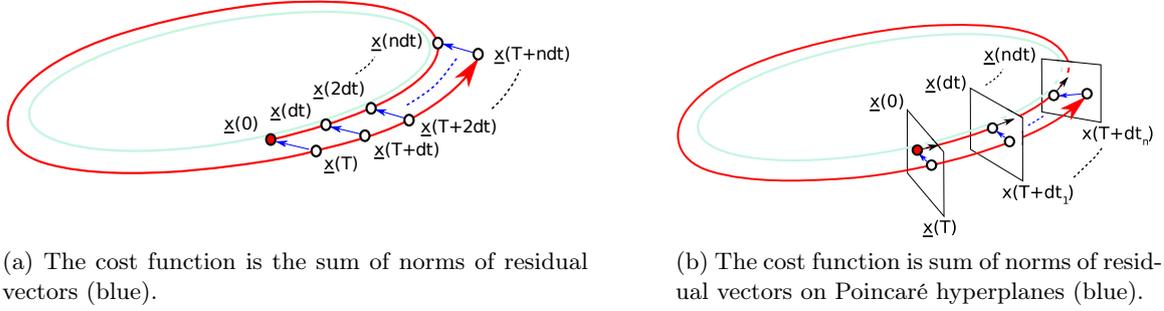
Figure 8.1: Higher order cost functions over a path, with (a) the standard shooting method and (b)the Poincaré shooting method. For the Poincaré shooting method, note that $ndt \neq dt_n$.

the cost function is the sum of the norms of $n$ residual vectors on hyperplanes defined every $dt$ time units. It is important to note that with the Poincaré shooting formulation the $n^{th}$ residual vector is not $\underline{x}(T + ndt) - \underline{x}(0 + ndt)$, because the time between crossing the hyperplanes will not be the same for different parts of the trajectory. A similar higher order cost function can be applied to a multiple shooting formulation.

The derivation of the first order adjoint equations (FOA) for a cost function over a path is identical to that in section 8.2, except that there are $2n$ discontinuities in the adjoint variables, at times $t = dt, 2dt, \ldots ndt$ and $t = T + dt, T + 2dt, \ldots T + ndt$, due to the extra terms of $\partial \mathcal{J}/\partial \underline{x}$. The FOA equations were originally:

$$FOA = \begin{cases} \underline{\theta}(T) = 2\left(\underline{x}(T) - \underline{x}_0\right), & \text{Boundary condition} \\ \frac{d\underline{\theta}_k}{dt} + \sum_i \underline{\theta}_i \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k} = 0, & 0 < t < T \\ \frac{d\mathcal{J}}{d\underline{x}_0} = \underline{\theta}(0) - 2(\underline{x}(T) - \underline{x}_0), & \text{Gradient evaluation} \end{cases}$$

Incorporating the higher order cost function in equation (8.21) adds the additional discontinuities in the adjoint variable:

$$FOA_{HO} = \begin{cases} \underline{\theta}(T + ndt) = 2\left(\underline{x}(T + ndt) - \underline{x}(ndt)\right), & \text{Boundary condition} \\ \underline{\theta}(t) = \underline{\theta}(t) + 2\left(\underline{x}(T + kdt) - \underline{x}(kdt)\right) & t = T + kdt, \{k \in \mathbb{Z}^+ \,|\, [0, n)\} \\ \frac{d\underline{\theta}_k}{dt} + \sum_i \underline{\theta}_i \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k} = 0, & 0 < t < T + ndt \\ \underline{\theta}(t) = \underline{\theta}(t) - 2\left(\underline{x}(T + kdt) - \underline{x}(kdt)\right) & t = kdt, \{k \in \mathbb{Z}^+ \,|\, (0, n]\} \\ \frac{d\mathcal{J}}{d\underline{x}_0} = \underline{\theta}(0) - 2(\underline{x}(T) - \underline{x}_0), & \text{Gradient evaluation} \end{cases}$$

The changes to an existing adjoint code are minor, so the higher order cost functions can be easily retrofitted to existing codes.

## 8.6   Optimisation algorithms

The convergence of gradient methods is very dependent on the choice of optimisation algorithm. The simplest algorithm to implement is the steepest descent algorithm, where $\underline{x}^{i+1} = \underline{x}^i + \alpha \frac{d\mathcal{J}}{d\underline{x}_0}$. The optimisation heads in the direction of the current gradient, using either a fixed steplength, or a line search until a minimum is found in the gradient direction. The steepest descent algorithm is not efficient for high dimensional systems, however, because the shape of the cost function is often a strong function of the location in $N$-dimensional space. Most importantly, if the cost function has second derivatives that are high in some directions but low in others - which is true when the cost function contours form elongated shapes - then the optimisation trajectory will zig-zag back and forth (see section 8.9.2.1) and the algorithm will converge slowly.

A more suitable optimisation method is the nonlinear conjugate gradient (CG) algorithm, which is much more effective than steepest descent when the cost function contours form elongated shapes. For each new direction, $\underline{d}$, a line search is performed until the gradient is perpendicular to the search direction, or in reality until $\left| \nabla \mathcal{J} \cdot \underline{d} \right| < \epsilon$, $\epsilon \in \mathbb{R}^+$. This can be achieved numerically by the Secant method, or the Newton-Raphson method when the Hessian-direction product, $\nabla^2 \mathcal{J}\underline{d}$, is available to be calculated. A method of calculating the Hessian-direction product is derived in section 8.7, with a second order adjoint formulation. There are several methods for calculating the next search direction in nonlinear CG algorithms, most notably the Fletcher-Reeves and Polak-Ribieré formulae [130]. The Polak-Ribieré formula is less stable, but it converges faster if started near a minimum. Conjugate gradient methods do not recover well from inaccuracies in gradient calculation, so if errors are present then restarting CG with a steepest descent step may be necessary. Preconditioners can increase the convergence of nonlinear CG methods. Near to a minimum, the preconditioner is usually an approximate inverse of the Hessian matrix evaluated at the minimum, $P = \left( \nabla^2 \mathcal{J} \right)^{-1} \Big|_{\underline{x}^*}$. The preconditioner causes the cost function contours to be more hyperspherical in the region around the minimum.

An alternative optimisation approach attempts to minimise the norm of the gradient, which becomes zero at a minima. A general Taylor series about a minima, $\underline{x}^*$, has the form in equation (8.22), which when truncated below $\mathcal{O}\left( \Delta \underline{x}^3 \right)$ is called the quadratic form. The Hessian matrix

is denoted $\nabla^2 \mathcal{J}$.

$$\mathcal{J}(\underline{x}^* + \underline{\Delta x}) = \mathcal{J}(\underline{x}^*) + \underline{\nabla \mathcal{J}} \cdot \underline{\Delta x} + \frac{1}{2}\underline{\Delta x} \cdot \left(\nabla^2 \mathcal{J} \underline{\Delta x}\right) + \mathcal{O}\left(\underline{\Delta x}^3\right) \tag{8.22}$$

Differentiating equation (8.22) by $\underline{\Delta x}$ and truncating terms gives equation (8.23):

$$\nabla^2 \mathcal{J} \underline{\Delta x} = -\underline{\nabla \mathcal{J}} \tag{8.23}$$

Equation (8.23) can be used to converge to the minima using standard multidimensional Newton iteration (see section 5.2). Newton's method is widely used because it converges superlinearly when started close to a solution. When started far from a solution, however, it can diverge. An additional disadvantage is that the solution of the linear equation can be computationally expensive, although matrix-free methods such as GMRES [92] can be used if a matrix-free procedure is available for Hessian-vector product evaluations (see section 5.3). In practice, equation (8.23) is often not solved explicitly, but its action is approximated by $\underline{\Delta x} = -W\underline{\nabla \mathcal{J}}$, where $W \approx \left(\nabla^2 \mathcal{J}\right)^{-1}$ and is symmetric positive definite. The approximate inverse of the Hessian, $W$, is estimated from previous location-gradient pairs and is updated after each new gradient evaluation. Techniques such as this are collectively known as Quasi-Newton methods.

Quasi-Newton methods can require large amounts of memory when $N$ is large, because they must store enough previous location-gradient pairs to approximate the properties of the Hessian matrix. There are several limited memory quasi-Newton methods, such as L-BFGS [131], that truncate the information stored in $W$ as the number of gradient evaluations increases. This truncation process involves extra computations, but does not require additional costly matrix-vector product evaluations.

With no phase condition, the cost function in equation (8.2) does not satisfy the quadratic form as there is a minimum $\mathcal{J} = 0$ around the entire orbit. This is equivalent to the Hessian matrix becoming singular as the search location nears the limit cycle. Whilst the CG algorithm still converges with a singular matrix, the speed of convergence may be reduced. When finding limit cycles with CG algorithms, it is therefore recommended that a phase condition is added to the cost function, or applied implicitly by using the Poincaré shooting method.

Both the steepest descent and CG algorithms are demonstrated on the horizontal Rijke tube model in section 8.9. Because the gradient methods in this chapter are developed specifically to converge to limit cycles from poor initial guesses, quasi-Newton methods are not discussed any further.

## 8.7 Second order adjoint formulation

A second order adjoint (SOA) formulation can be derived in order to evaluate the Hessian-vector product. The Hessian-vector product is useful for several reasons: first, it can be used for Newton-Raphson iteration to speed up line searches; second, it can be used to solve the Newton equation by iteration; and third, it can be used to determine the sensitivity of limit cycles to external perturbations. The sensitivity of limit cycles has been studied for several applications, including oscillator noise in circuitry [132] and weakly coupled oscillators in neuroscience [133]. The general second order adjoint formulation is derived in this section for the standard shooting method.

In the first order adjoint (FOA) derivation of section 8.2, it was shown that for a given starting state, $\underline{x}_0$, the gradient of the cost function, $\nabla \mathcal{J}$, is evaluated by timemarching the direct equations forward, and then timemarching the adjoint equations backwards. The second order adjoint (SOA) will be used in this section to evaluate the Hessian-vector product [134, 135, 136, 137], where the $k^{th}$ element of the product follows $\left(\nabla^2 \mathcal{J} \underline{x}'_0\right)_k = \sum_i \frac{\partial \nabla \mathcal{J}_k}{\partial \underline{x}_{0i}} \underline{x}'_{0i}$. This process requires an extra forwards timemarch and an extra backwards timemarch. The Hessian-vector product could be estimated using finite differences (equation (8.24)), which is less accurate than that found by the SOA method.

$$\nabla^2 \mathcal{J} \underline{x}'_0 = \frac{\nabla \mathcal{J}\big|_{\underline{x}_0 + \delta \underline{x}'_0} - \nabla \mathcal{J}\big|_{\underline{x}_0}}{\delta} + \mathcal{O}(\delta) \tag{8.24}$$

The gradient is found in the FOA derivation by constraining the Lagrangian such that:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}} = \frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}}$$

The Hessian matrix, $\nabla^2 \mathcal{J}_{j,k} = \frac{\partial^2 \mathcal{J}}{\partial \underline{x}_{0k} \partial \underline{x}_{0j}}$ , is found in the SOA derivation by constraining the Lagrangian such that:

$$\frac{\mathrm{d}^2 \mathcal{J}}{\mathrm{d}\underline{x}_{0k} \mathrm{d}\underline{x}_{0j}} = \frac{\partial^2 \mathcal{L}}{\partial \underline{x}_{0k} \partial \underline{x}_{0j}} \tag{8.25}$$

Differentiating the Lagrangian twice, and summing over the columns of the Hessian matrix

($i$) gives an expression for the $k^{th}$ row of the Hessian-vector product:

$$
\begin{aligned}
\mathcal{J} &= \mathcal{L} + \mathcal{C} \\
\sum_i \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{0k}\mathrm{d}\underline{x}_{0i}}x'_{0i} &= \sum_i x'_{0i}\left(\frac{\partial^2\mathcal{L}}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}} + \frac{\partial^2\mathcal{L}}{\partial\underline{x}(\underline{x}_0,t)\partial\underline{x}_{0i}}\cdot\frac{\partial\underline{x}(\underline{x}_0,t)}{\partial\underline{x}_{0k}}\cdots\right. \\
&\quad + \frac{\partial\mathcal{L}}{\partial\underline{x}(\underline{x}_0,t)}\cdot\frac{\partial^2\underline{x}(\underline{x}_0,t)}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}} + \frac{\partial^2\mathcal{L}}{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)\partial\underline{x}_{0i}}\cdot\frac{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)}{\partial\underline{x}_{0k}}\cdots \\
&\quad + \frac{\partial\mathcal{L}}{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)}\cdot\frac{\partial^2\underline{\theta}(\underline{x}_0,\underline{x},t)}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}} + \frac{\partial^2\mathcal{C}}{\partial\underline{x}(\underline{x}_0,t)\partial\underline{x}_{0i}}\cdot\frac{\partial\underline{x}(\underline{x}_0,t)}{\partial\underline{x}_{0k}}\cdots \\
&\quad + \frac{\partial\mathcal{C}}{\partial\underline{x}(\underline{x}_0,t)}\cdot\frac{\partial^2\underline{x}(\underline{x}_0,t)}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}} + \frac{\partial^2\mathcal{C}}{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)\partial\underline{x}_{0i}}\cdot\frac{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)}{\partial\underline{x}_{0k}}\cdots \\
&\quad \left.+ \frac{\partial\mathcal{C}}{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)}\cdot\frac{\partial^2\underline{\theta}(\underline{x}_0,\underline{x},t)}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}} + \frac{\partial^2\mathcal{C}}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}}\right)
\end{aligned}
\tag{8.26}
$$

The $\frac{\partial\underline{x}(\underline{x}_0,t)}{\partial\underline{x}_{0k}}$, $\frac{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)}{\partial\underline{x}_{0k}}$, $\frac{\partial^2\underline{x}(\underline{x}_0,t)}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}}$ and $\frac{\partial^2\underline{\theta}(\underline{x}_0,\underline{x},t)}{\partial\underline{x}_{0k}\partial\underline{x}_{0i}}$ terms are unknown because the system is nonlinear. The first order adjoint formulation fixes $\frac{\partial\mathcal{L}}{\partial\underline{x}(\underline{x}_0,t)} = \frac{\partial\mathcal{L}}{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)} = 0$, and constructs all of the constraints so that $\mathcal{C} = 0$ and $\mathcal{C} \neq f(\underline{x}_0,\underline{x}(\underline{x}_0,t),\underline{\theta}(\underline{x}_0,t))$. For the second order adjoint formulation, to reduce equation (8.26) to the form of equation (8.25), it must also fix:

$$
\frac{\partial^2\mathcal{L}}{\partial\underline{x}_k(\underline{x}_0,t)\partial\underline{x}_{0i}} = \frac{\partial^2\mathcal{L}}{\partial\underline{\theta}(\underline{x}_0,\underline{x},t)_k\partial\underline{x}_{0i}} = 0, \quad 1 \leq i,k \leq N
\tag{8.27}
$$

The Hessian-vector product becomes:

$$
\left(\nabla^2\mathcal{J}\underline{x}'_0\right)_k = \sum_i \frac{\partial\nabla\mathcal{J}_k}{\partial\underline{x}_{0i}}x'_{0i} = \sum_i \frac{\partial\left(\frac{\partial\mathcal{L}}{\partial\underline{x}_{0k}}\right)}{\partial\underline{x}_{0i}}x'_{0i}
$$

In the standard shooting method, the direct equations are integrated between limits 0 and $T$, so the FOA Lagrangian formulation is:

$$
\mathcal{L} = \mathcal{J} - \left[\frac{\mathrm{d}\underline{\theta}}{\mathrm{d}t}, -\underline{x}\right]_0^T - [\underline{\theta}, -F(\underline{x})]_0^T - \langle\underline{\theta}(T),\underline{x}(T)\rangle + \langle\underline{\theta}(0),\underline{x}(0)\rangle - \langle\underline{G},\underline{x}(0)-\underline{x}_0\rangle
$$

with differentials:

$$
\begin{aligned}
\left[\frac{\partial\mathcal{L}}{\partial\underline{x}_k},\delta\underline{x}_k\right]_0^T &= \langle 2(\underline{x}_k(T)-\underline{x}_{0k}),\delta\underline{x}_k(T)\rangle + \left[\frac{\mathrm{d}\underline{\theta}_k}{\mathrm{d}t},\delta\underline{x}_k\right]_0^T + \left[\underline{\theta},\frac{\partial F(\underline{x})}{\partial\underline{x}_k}\delta\underline{x}_k\right]_0^T\cdots \\
&\quad - \langle\underline{\theta}_k(T),\delta\underline{x}_k(T)\rangle + \langle\underline{\theta}_k(0),\delta\underline{x}_k(0)\rangle - \langle\underline{G}_k,\delta\underline{x}_k(0)\rangle
\end{aligned}
$$

$$
\begin{aligned}
\left[\frac{\partial\mathcal{L}}{\partial\underline{\theta}_k},\delta\underline{\theta}_k\right]_0^T &= -\left[\delta\underline{\theta}_k,\frac{\mathrm{d}x_k}{\mathrm{d}t}\right]_0^T - [\delta\underline{\theta}_k,-F_k(\underline{x})]_0^T\cdots \\
&\quad - \langle\delta\underline{\theta}_k(T),\underline{x}_k(T)\rangle + \langle\delta\underline{\theta}_k(0),\underline{x}_k(0)\rangle
\end{aligned}
$$

Several steps are now required to enforce the conditions in equation (8.27) and derive the SOA equations. To make the derivation easier, two variables are defined, $\underline{\hat{x}}_{i,k}$ and $\underline{\hat{\theta}}_{k,j}$. They represent time varying elements in the matrices of partial differentials of the direct state and FOA state respectively:

$$\underline{\hat{x}}_{i,k} \equiv \frac{\partial \underline{x}_i}{\partial \underline{x}_{0k}} \quad , \quad \underline{\hat{\theta}}_{k,j} \equiv \frac{\partial \underline{\theta}_k}{\partial \underline{x}_{0j}}$$

First, the differential of the Lagrangian with respect to one of the FOA variables is differentiated again with respect to $\underline{x}_{0j}$, and then forced to equal zero in accordance with the condition in equation (8.27):

$$\left[ \frac{\partial \mathcal{L}}{\partial \underline{\theta}_k}, 1 \right]_0^T = -\left[ 1, \frac{\mathrm{d}\underline{x}_k}{\mathrm{d}t} \right]_0^T - [1, -F_k(\underline{x})]_0^T$$

$$\left[ \frac{\partial \mathcal{L}}{\partial \underline{\theta}_k \partial \underline{x}_{0j}}, \delta\underline{x}_{0j} \right]_0^T = -\left[ \delta\underline{x}_{0j}, \frac{\partial^2 \underline{x}_k}{\partial t \partial \underline{x}_{0j}} \right]_0^T - \left[ \delta\underline{x}_{0j}, -\frac{\partial F_k(\underline{x})}{\partial \underline{x}_{0j}} \right]_0^T$$

$$\therefore \frac{\mathrm{d}\underline{\hat{x}}_{i,k}}{\mathrm{d}t} = \sum_j \frac{\partial F_i(\underline{x})}{\partial \underline{x}_j} \underline{\hat{x}}_{j,k}$$

Second, the differential of the Lagrangian with respect to one of the state variables is differentiated with respect to $\underline{x}_{0j}$, and then forced to equal zero in accordance with the condition in equation (8.27). The notation $\delta_t(T) \equiv \delta(t-T)$ is the Dirac delta at a time $T$, and the notation $\delta_{ij} \equiv 1, i = j, 0, i \neq j$.

$$\left[ \frac{\partial \mathcal{L}}{\partial \underline{x}_k}, 1 \right]_0^T = \langle 2(\underline{x}_k(T) - \underline{x}_{0k}), \delta_t(T) \rangle + \left[ \frac{\mathrm{d}\underline{\theta}_k}{\mathrm{d}t}, 1 \right]_0^T + \left[ \underline{\theta} \cdot \frac{\partial F(\underline{x})}{\partial \underline{x}_k}, 1 \right]_0^T \dots$$
$$- \langle \underline{\theta}_k, \delta_t(T) \rangle + \langle \underline{\theta}_k, \delta_t(0) \rangle - \langle \underline{G}_k, \delta_t(0) \rangle$$

$$\left[ \frac{\partial \mathcal{L}}{\partial \underline{x}_k \partial \underline{x}_{0j}}, \delta\underline{x}_{0j} \right]_0^T = \left\langle \frac{\partial 2(\underline{x}_k(T) - \underline{x}_{0k})}{\partial \underline{x}_{0j}} \delta_t(T), \delta\underline{x}_{0j} \right\rangle + \left[ \frac{\partial^2 \underline{\theta}_k}{\partial t \partial \underline{x}_{0j}}, \delta\underline{x}_{0j} \right]_0^T \dots$$
$$+ \left[ \frac{\partial \underline{\theta}}{\partial \underline{x}_{0j}} \cdot \frac{\partial F(\underline{x})}{\partial \underline{x}_k}, \delta\underline{x}_{0j} \right]_0^T + \left[ \underline{\theta} \cdot \left( \frac{\partial^2 F(\underline{x})}{\partial \underline{x}_k \partial \underline{x}_{0j}} \right), \delta\underline{x}_{0j} \right]_0^T \dots$$
$$- \left\langle \frac{\partial \underline{\theta}_k}{\partial \underline{x}_{0j}} \delta_t(T), \delta\underline{x}_{0j} \right\rangle + \left\langle \frac{\partial \underline{\theta}_k}{\partial \underline{x}_{0j}} \delta_t(0), \delta\underline{x}_{0j} \right\rangle \dots$$
$$- \left\langle \frac{\partial \underline{G}_k}{\partial \underline{x}_{0j}} \delta_t(0), \delta\underline{x}_{0j} \right\rangle \tag{8.28}$$

Rewriting equation (8.28) with the new notation gives:

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_k \partial \underline{x}_{0j}}, \delta \underline{x}_{0j}\right]_0^T = \langle 2(\hat{\underline{x}}_{k,j}(T_p) - \delta_{kj})\delta_t(T), \delta \underline{x}_{0j}\rangle + \left[\frac{\mathrm{d}\hat{\underline{\theta}}_{k,j}}{\mathrm{d}t}, \delta \underline{x}_{0j}\right]_0^T \cdots
$$

$$
+ \left[\hat{\underline{\theta}} \cdot \frac{\partial F(\underline{x})}{\partial \underline{x}_k}, \delta \underline{x}_{0j}\right]_0^T + \left[\underline{\theta} \cdot \left(\sum_m \frac{\partial^2 F(\underline{x})}{\partial \underline{x}_k \partial \underline{x}_m}\frac{\partial \underline{x}_m}{\partial \underline{x}_{0j}}\right), \delta \underline{x}_{0j}\right]_0^T \cdots
$$

$$
- \langle \hat{\underline{\theta}}_{k,j}\delta_t(T), \delta \underline{x}_{0j}\rangle + \langle \hat{\underline{\theta}}_{k,j}\delta_t(0), \delta \underline{x}_{0j}\rangle - \langle \hat{\underline{G}}_{k,j}\delta_t(0), \delta \underline{x}_{0j}\rangle
$$

Equating this to zero gives a boundary condition at time $T$, and a time evolution equation:

$$
\hat{\underline{\theta}}_{k,j}(T) = 2(\hat{\underline{x}}_{k,j}(T) - \delta_{kj})
$$

$$
\frac{\mathrm{d}\hat{\underline{\theta}}_{k,j}}{\mathrm{d}t} + \hat{\underline{\theta}} \cdot \frac{\partial F(\underline{x})}{\partial \underline{x}_k} + \underline{\theta} \cdot \left(\sum_m \frac{\partial^2 F(\underline{x})}{\partial \underline{x}_k \partial \underline{x}_m}\hat{\underline{x}}_{m,j}\right) = 0 \tag{8.29}
$$

Third, to get the Hessian-vector product, an expression for $\frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}\partial \underline{x}_{0j}}$ is required. From the FOA:

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}}, \delta \underline{x}_{0k}\right]_0^T = \langle -2(\underline{x}_k(T) - \underline{x}_{0k}), \delta \underline{x}_{0k}\rangle + \langle \underline{G}_k, \delta \underline{x}_{0k}\rangle
$$

Differentiating this w.r.t $\underline{x}_{0j}$ gives:

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}\partial \underline{x}_{0j}}, \delta \underline{x}_{0j}\right]_0^T = \langle -2(\hat{\underline{x}}_{k,j}(T) - \delta_{kj}), \delta \underline{x}_{0j}\rangle + \langle \hat{\underline{G}}_{k,j}, \delta \underline{x}_{0j}\rangle
$$

The formula for the Hessian-vector product is therefore:

$$
\sum_j \frac{\partial \mathcal{L}}{\partial \underline{x}_{0k}\partial \underline{x}_{0j}}\underline{x}'_{0j} = \sum_j x\left(-2(\hat{\underline{x}}_{k,j}(T) - \delta_{kj}) + \hat{\underline{G}}_{k,j}\right)\underline{x}'_{0j}
$$

$$
= \sum_j \left(-2(\hat{\underline{x}}_{k,j}(T) - \delta_{kj}) + \hat{\underline{\theta}}_{k,j}(0)\right)\underline{x}'_{0j}
$$

$$
= \sum_j \left(-2\hat{\underline{x}}_{k,j}(T) + \hat{\underline{\theta}}_{k,j}(0)\underline{x}'_{0j}\right) + 2\underline{x}'_{0k}
$$

In the formula for the Hessian-vector product each contribution is summed over all $\underline{x}'_{0j}$. Therefore, combined variables can be defined as:

$$
\underline{X}'_k \equiv \sum_j \hat{\underline{x}}_{k,j}\underline{x}'_{0j}
$$

$$
\underline{\theta}'_k \equiv \sum_j \hat{\underline{\theta}}_{k,j}\underline{x}'_{0j}
$$

## 8. GRADIENT METHODS

These are the first variational state, and the first variational adjoint state. The first variational state, $\underline{X}'$, is governed by the first variational equations:

$$\frac{\mathrm{d}\hat{\underline{x}}_{i,k}}{\mathrm{d}t} = \sum_j \frac{\partial F_i(\underline{x})}{\partial \underline{x}_j} \hat{\underline{x}}_{j,k}$$

$$\sum_k \frac{\mathrm{d}\hat{\underline{x}}_{i,k}}{\mathrm{d}t} \underline{x}'_{0k} = \sum_k \sum_j \frac{\partial F_i(\underline{x})}{\partial \underline{x}_j} \hat{\underline{x}}_{j,k} \underline{x}'_{0k}$$

$$\therefore \frac{\mathrm{d}\underline{X}'_i}{\mathrm{d}t} = \sum_j \frac{\partial F_i(\underline{x})}{\partial \underline{x}_j} \underline{X}'_j$$

The first variational equations are equivalent to linearising about the state $\underline{x}$, both in space and time. The first variational equations arise from a perturbation to the state, $\underline{X}'$, in the limit for small $\underline{X}'$:

$$\frac{\mathrm{d}\left(\underline{x}_k + \underline{X}'_k\right)}{\mathrm{d}t} = F_k(\underline{x} + \underline{X}') \approx F_k(\underline{x}) + \sum_i \frac{\partial F_k(\underline{x})}{\partial \underline{x}_i} \underline{X}'_i \approx \frac{\mathrm{d}\underline{x}_k}{\mathrm{d}t} + \frac{\mathrm{d}\underline{X}'_k}{\mathrm{d}t}$$

Summing equation (8.29) over all $\underline{x}'_{0j}$ gives:

$$\sum_j \underline{x}'_{0j} \left( \frac{\mathrm{d}\hat{\underline{\theta}}_{k,j}}{\mathrm{d}t} + \hat{\underline{\theta}} \cdot \frac{\partial F(\underline{x})}{\partial \underline{x}_k} + \underline{\theta} \cdot \left( \sum_m \frac{\partial^2 F(\underline{x})}{\partial \underline{x}_k \partial \underline{x}_m} \hat{\underline{x}}_{m,j} \right) \right) = 0$$

which becomes the second order adjoint equations:

$$\frac{\mathrm{d}\underline{\theta}'_k}{\mathrm{d}t} + \underline{\theta}' \cdot \frac{\partial F(\underline{x})}{\partial \underline{x}_k} + \underline{\theta} \cdot \left( \sum_m \frac{\partial^2 F(\underline{x})}{\partial \underline{x}_k \partial \underline{x}_m} \underline{X}'_m \right) = 0$$

with boundary conditions:

$$\underline{\theta}'_k(T) = 2(\underline{X}'_k(T) - \underline{x}'_{0k})$$

For the standard shooting method we also require the Hessian-vector product terms related to the period, $\sum_i \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T\mathrm{d}\underline{x}_{0i}} \underline{x}'_{0i}$ and $\frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}T^2}$. These are:

$$\sum_i \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T\mathrm{d}\underline{x}_{0i}} \underline{x}'_{0i} = -2\dot{\underline{x}}(T) \cdot \underline{x}'_0$$

$$\frac{\mathrm{d}^2\mathcal{J}}{\mathrm{d}T^2} T' = 2 \left( \dot{\underline{x}}(T) \cdot \dot{\underline{x}}(T) + (\underline{x}(T) - \underline{x}_0) \cdot \ddot{\underline{x}}(T) \right) T'$$

### 8.7.1   Second order adjoint summary

1. Given a starting point $\underline{x}_0$ and guess for the period $T$, first perform the FOA in accordance with the equations in section 8.2.1. The direct and adjoint variables are stored in time, with checkpointing if necessary.

2. The Hessian-vector product is now found, where the vector is $\underline{x}_0'$. The first variational state is initialised with the vector:

$$\underline{X}'(0) = \underline{x}_0'$$

3. The first variational equations are integrated forward to time $T$ using information from the direct variables, $\underline{x}$, calculated in the FOA.

$$\frac{\mathrm{d}\underline{X}_i'}{\mathrm{d}t} = \sum_j \frac{\partial F_i(\underline{x})}{\partial \underline{x}_j} \underline{X}_j'$$

4. The SOA variables are initialised at time $T$ with:

$$\underline{\theta}_k'(T) = 2(\underline{X}_k'(T) - \underline{x}_{0k}')$$

5. The SOA equations are integrated backwards to time 0 using information from the direct variables, $\underline{x}$, and adjoint variables, $\underline{\theta}$, calculated in the FOA.

$$\frac{\mathrm{d}\underline{\theta}_k'}{\mathrm{d}t} + \underline{\theta}' \cdot \frac{\partial F(\underline{x})}{\partial \underline{x}_k} + \underline{\theta} \cdot \left( \sum_m \frac{\partial^2 F(\underline{x})}{\partial \underline{x}_k \partial \underline{x}_m} \underline{X}_m' \right) = 0$$

6. The Hessian-vector product is calculated at time 0 with:

$$\sum_j \frac{\partial \mathcal{L}}{\partial \underline{x}_{0k} \partial \underline{x}_{0j}} \underline{x}_{0j}' \quad = \quad \underline{\theta}_k'(0) - 2 \left( \underline{X}_k'(T) - 2\underline{x}_{0k}' \right)$$

7. The terms relating to the period are also calculated with:

$$\sum_i \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}T \mathrm{d}\underline{x}_{0i}} \underline{x}_{0i}' \quad = \quad -2\underline{\dot{x}}(T) \cdot \underline{x}_0'$$

$$\frac{\mathrm{d}^2 \mathcal{J}}{\mathrm{d}T^2} T' \quad = \quad 2 \left( \underline{\dot{x}}(T) \cdot \underline{\dot{x}}(T) + (\underline{x}(T) - \underline{x}_0) \cdot \underline{\ddot{x}}(T) \right) T'$$

8. The Hessian-vector product can now be used for line searching routines, or Newton methods.

The SOA is not demonstrated in the results section of this chapter, but the derivation is included in this thesis because it is very useful for analysing systems with limit cycles: first, it can greatly increase the efficiency of line search routines used to iterate to a limit cycle; second, it is required to find bifurcations of limit cycles and to find limit cycle sensitivities. If the FOA and SOA are derived and coded by hand, then implementing the SOA is a time consuming process. The SOA can be generated automatically by algorithmic differentiation, however, such as in the dolfin-adjoint extension to the FEniCS finite element package [138, 139]. The potential of algorithmic differentiation is discussed further in chapter 9.

## 8.8 Multiple shooting methods

Multiple shooting methods were introduced in section 6.3. In the multiple shooting method, the limit cycle is split into $n$ segments, which are timemarched separately. This is particularly useful for nonlinear adjoint calculations, which have large memory requirements because the direct state is required in the adjoint equations. Multiple shooting methods also reduce the real time required for a gradient evaluation; if the limit cycle is split into $n$ segments, timemarching each segment is $n$ times faster than timemarching with only one segment.

The multiple shooting method, however, requires minimising a cost function over $nN + 1$ variables, as oppose to $N + 1$ variables for the standard shooting method. To achieve a given level of convergence, the multiple shooting method will therefore require more gradient evaluations than the single shooting method. If the multiple shooting method requires less than $n$ times the number of gradient evaluations that the single shooting method requires, however, then the multiple shooting method will converge faster if sufficient parallel processors are available.

The formulation in this section differs from some multiple shooting methods in the literature. The formulation in Ref. [128] forces the $N$ initial states to form a closed loop, even before a guess for the period is known. Their method then forms a PDE to govern the evolution of this closed loop using a 'Newton descent' strategy, which minimises a cost function that is the integrated value of $\underline{\dot{x}}_p \cdot \underline{t}_p$ around the cycle, where $\underline{\dot{x}}_p$ is the time derivative of a point on the loop and $\underline{t}_p$ is the tangent vector of the loop at that point. This approach is not like the multiple shooting approach in chapter 6, which only attempts to find $N$ initial states on the cycle - but is not interested in the cycle trajectory in between them. The approach of Ref. [128] describes a closed loop, and then distorts the entire closed loop until it matches the cycle. Their method may diverge, however, when an initial guess for a closed loop is not known a

priori. The formulation in Ref. [129] also considers a cost function integrated around the loop, but can also calculate the dependence of the cost function on the period.

The derivation of the adjoint technique for the standard shooting method was given in section 8.2. In the next section, the adjoint formulation is derived for the multiple shooting method with two segments (double-shooting). The adjoint formulation for a general number of segments then follows by analogy.

### 8.8.1 General form of the double shooting method

In this subsection, the limit cycle is split into two segments and the formulation of the double-shooting algorithm is derived. The first segment of the loop will be referred to with subscript $f$ and the second with subscript $s$. Initial conditions are shown with subscript $0$ and $T/2$ respectively. Segment 1 ($f$) is integrated from $t = 0 \rightarrow T/2$ and segment 2 ($s$) from $t = T/2 \rightarrow T$. With two separate segments the cost function minimises the distance between the ends of both segments:

$$\mathcal{J} = \left\langle \underline{x}_s(T) - \underline{x}_{f0} \, , \, \underline{x}_s(T) - \underline{x}_{f0} \right\rangle + \left\langle \underline{x}_f(T/2) - \underline{x}_{sT/2} \, , \, \underline{x}_f(T/2) - \underline{x}_{sT/2} \right\rangle \qquad (8.30)$$

With the loop divided into two segments, the Lagrangian becomes:

$$\mathcal{L} \equiv \mathcal{J} - \left[ \underline{\theta}_f, \frac{\mathrm{d}\underline{x}_f}{\mathrm{d}t} - F(\underline{x}_f) \right]_0^{\frac{T}{2}} - \left[ \underline{\theta}_s, \frac{\mathrm{d}\underline{x}_s}{\mathrm{d}t} - F(\underline{x}_s) \right]_{\frac{T}{2}}^{T} \ldots$$
$$- \left\langle \underline{G}_f, \underline{x}_f(0) - \underline{x}_{f0} \right\rangle - \left\langle \underline{G}_s, \underline{x}_s(T/2) - \underline{x}_{sT/2} \right\rangle \qquad (8.31)$$

Differentiating the Lagrangian with respect to one of the first variables gives:

$$\left[ \frac{\partial \mathcal{L}}{\partial \underline{x}_{fk}}, \delta \underline{x}_{fk} \right] = \left\langle 2(\underline{x}_{fk}(T/2) - \underline{x}_{sT/2k}), \delta \underline{x}_{fk}(T/2) \right\rangle + \left[ \frac{\mathrm{d}\underline{\theta}_{fk}}{\mathrm{d}t}, \delta \underline{x}_{fk} \right]_0^{\frac{T}{2}} \ldots$$
$$+ \left[ \underline{\theta}_{fk}, \frac{\partial F(\underline{x}_f)}{\partial \underline{x}_{fk}} \delta \underline{x}_{fk} \right]_0^{\frac{T}{2}} - \left\langle \theta_{fk}(T/2), \delta \underline{x}_{fk}(T/2) \right\rangle \ldots$$
$$+ \left\langle \theta_{fk}(0), \delta \underline{x}_{fk}(0) \right\rangle - \left\langle G_{fk}, \delta \underline{x}_{fk}(0) \right\rangle \qquad (8.32)$$

## 8. GRADIENT METHODS

Differentiating the Lagrangian with respect to one of the second variables gives:

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{sk}}, \delta \underline{x}_{sk}\right] = \left\langle 2(\underline{x}_{sk}(T) - \underline{x}_{fk0}), \delta x_{sk}(T)\right\rangle + \left[\frac{d\underline{\theta}_{sk}}{dt}, \delta \underline{x}_{sk}\right]_{\frac{T}{2}}^{T} \dots
$$
$$
+ \left[\underline{\theta}_{sk}, \frac{\partial F(\underline{x}_s)}{\partial \underline{x}_{sk}} \delta \underline{x}_{sk}\right]_{\frac{T}{2}}^{T} - \left\langle \theta_{sk}(T), \delta \underline{x}_{sk}(T)\right\rangle \dots
$$
$$
+ \left\langle \theta_{sk}(T/2), \delta \underline{x}_{sk}(T/2)\right\rangle - \left\langle G_{sk}, \delta \underline{x}_{sk}(T/2)\right\rangle \tag{8.33}
$$

Summing the components of $\delta \underline{x}_{fk}$ and equating them to zero gives the adjoint equations and the adjoint boundary conditions. The adjoint equations are identical in form for the two segments. Differentiating with respect to the two starting conditions gives the last two equations that are required to find the gradient:

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{f0k}}, \delta \underline{x}_{f0k}\right] = \left\langle -2(\underline{x}_{sk}(T) - \underline{x}_{fk0}), \delta \underline{x}_{f0k}\right\rangle + \left\langle G_{fk}, \delta \underline{x}_{f0k}\right\rangle \tag{8.34}
$$

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{sT/2k}}, \delta \underline{x}_{sT/2k}\right] = \left\langle -2(\underline{x}_{fk}(T/2) - \underline{x}_{sT/2k}), \delta \underline{x}_{sT/2k}\right\rangle + \left\langle G_{sk}, \delta \underline{x}_{sT/2k}\right\rangle \tag{8.35}
$$

Substituting $\underline{\theta}_{fk}(0) = G_{fk}$ and $\underline{\theta}_{sk}(T/2) = G_{sk}$ into the previous equation gives the gradient equations. The double-shooting method is therefore:

$$
FOA_{DS} = \begin{cases}
\underline{\theta}_f(T/2) = 2(\underline{x}_{fk}(T/2) - \underline{x}_{skT/2}), & \text{Boundary condition}_f \\
\frac{d\underline{\theta}_{fk}}{dt} + \sum_i \underline{\theta}_{fi} \frac{\partial F_i(\underline{x}_f)}{\partial \underline{x}_{fk}} = 0, & 0 \le t \le T/2 \\
\underline{\theta}_{sk}(T) = 2(\underline{x}_{sk}(T) - \underline{x}_{fk0}), & \text{Boundary condition}_s \\
\frac{d\underline{\theta}_{sk}}{dt} + \sum_i \underline{\theta}_{si} \frac{\partial F_i(\underline{x}_s)}{\partial \underline{x}_{sk}} = 0, & T/2 \le t \le T \\
\frac{d\mathcal{J}}{d\underline{x}_{f0}} = \underline{\theta}_f(0) - 2(\underline{x}_s(T) - \underline{x}_{f0}), & \text{Gradient evaluation}_f \\
\frac{d\mathcal{J}}{d\underline{x}_{sT/2}} = \underline{\theta}_s(T/2) - 2(\underline{x}_f(T/2) - \underline{x}_{sT/2}), & \text{Gradient evaluation}_s
\end{cases}
$$

### 8.8.2 General form of the multiple shooting method

The equations derived in the previous section follow a pattern. When dividing the loop into $n$ segments, the state vector for optimisation is of length $nN$, and contains all $n$ starting states $\left[\underline{x}_0, \underline{x}_{\frac{1}{n}}, \underline{x}_{\frac{2}{n}}, \dots, \underline{x}_{\frac{n-1}{n}}\right]$. The cost function becomes:

$$
\mathcal{J} = \left\langle \underline{x}_n(1) - \underline{x}_0, \underline{x}_n(1) - \underline{x}_0\right\rangle + \sum_{j=1}^{n-1} \left\langle \underline{x}_j(j/n) - \underline{x}_{\frac{j}{n}}, \underline{x}_j(j/n) - \underline{x}_{\frac{j}{n}}\right\rangle \tag{8.36}
$$

For each of the $n$ segments the same procedure is used to find the gradient of the cost

156

function. For the $j^{th}$ segment, the process is:

1. Timemarch forward from the initial state, $\underline{x}_{\frac{j-1}{n}}$, from $t = \frac{j-1}{n} \rightarrow \frac{j}{n}$, using the direct equations:

$$\frac{\mathrm{d}\underline{x}}{\mathrm{d}t} = F(\underline{x}) \tag{8.37}$$

2. Initialise the adjoint state, at time $t = \frac{j}{n}$, to be twice the difference between the end of this segment and the start of the next:

$$\underline{\theta}_j(\frac{j}{n}) = 2(\underline{x}_j(\frac{j}{n}) - \underline{x}_{\frac{j}{n}}) \tag{8.38}$$

This is cyclic, so the end of the $n^{th}$ segment is compared with the start of the first.

3. Timemarch backwards with the adjoint equations from time $t = \frac{j}{n} \rightarrow \frac{j-1}{n}$

$$\frac{\mathrm{d}\underline{\theta}_k}{\mathrm{d}t} + \sum_i \underline{\theta}_{ji} \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k} = 0 \tag{8.39}$$

4. Evaluate the gradient of the cost function with respect to the $n^{th}$ initial state:

$$\frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{\frac{j-1}{n}}} = \underline{\theta}_j(\frac{j-1}{n}) - 2(\underline{x}_j(\frac{j}{n}) - \underline{x}_{\frac{j}{n}}) \tag{8.40}$$

5. Assemble the gradients from all segments to form the full gradient vector:

$$\nabla\mathcal{J} = \left[ \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_0}, \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{\frac{1}{n}}}, \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{\frac{2}{n}}}, \ldots, \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{\frac{n-1}{n}}} \right] \tag{8.41}$$

6. This gradient is used in an optimisation algorithm to change the starting states, $\left[\underline{x}_0, \underline{x}_{\frac{1}{n}}, \underline{x}_{\frac{2}{n}}, \ldots, \underline{x}_{\frac{n-1}{n}}\right]$. Steps 1-6 are repeated until a predefined level of convergence is reached.

### 8.8.3 Extensions to the multiple shooting method

In the multiple shooting method above, the adjoint equation for each segment of the limit cycle is initialised with the state at the start of the next segment. This results in poor convergence because the start of the next segment appears fixed. The start of the next segment will move, however, when the gradient information is used to change the initial state. Each segment has no knowledge of how the start of the next segment will move, and therefore the gradient of the cost function changes rapidly as the starting conditions move.

In section 8.5, higher order cost functions were introduced as a method of increasing the convergence of the optimisation procedure. A higher order cost function can be used in the multiple shooting method at each joint between segments. This increases the strength of the matching required between segments and smooths the cost function.

In the next section, the adjoint formulation is derived for the double-shooting method with a higher order cost function. The adjoint formulation for a general number of segments then follows by analogy.

### 8.8.4 General form of the double shooting method with a higher order cost function

In this subsection, the limit cycle is split into two segments, and the adjoint formulation is derived in a similar manner to section 8.8.1, but with additional terms in the cost function to increase the coupling between segments. Each segment of the limit cycle is timemarched further than the $T/2$ needed to close the limit cycle, in order to give an overlap between the segments; segment 1 ($f$) is integrated in time from $t = 0 \rightarrow T/2 + \Delta$ and segment 2 ($s$) from $t = T/2 \rightarrow T + \Delta$, where $0 < \Delta \leq T/2$. In the case of double shooting, when $\Delta = T/2$, it is equivalent to single shooting with two separate segments of the limit cycle and there are no computational benefits. For simplicity, in this section $\Delta$ is taken to be a single value, which means that there is only one extra residual vector per segment, and therefore that only two extra terms are added to the cost function. When multiple extra residual vectors are added (as in Figure 8.1a), the adjoint variable has more discontinuities in time, in a similar manner to the derivation in section 8.5.

The cost function is the distance between the ends of both segments, and the distance between the overlap points:

$$
\begin{aligned}
\mathcal{J} \;=\; & \left\langle \underline{x}_s(T) - \underline{x}_{f0} \,,\, \underline{x}_s(T) - \underline{x}_{f0} \right\rangle \ldots \\
& + \left\langle \underline{x}_s(T + \Delta) - \underline{x}_f(0 + \Delta) \,,\, \underline{x}_s(T + \Delta) - \underline{x}_f(0 + \Delta) \right\rangle \ldots \\
& + \left\langle \underline{x}_f(T/2) - \underline{x}_{sT/2} \,,\, \underline{x}_f(T/2) - \underline{x}_{sT/2} \right\rangle \ldots \\
& + \left\langle \underline{x}_f(T/2 + \Delta) - \underline{x}_s(T/2 + \Delta) \,,\, \underline{x}_f(T/2 + \Delta) - \underline{x}_s(T/2 + \Delta) \right\rangle
\end{aligned}
\tag{8.42}
$$

With the loop divided into two segments, the Lagrangian becomes:

$$
\begin{aligned}
\mathcal{L} \;\equiv\; & \mathcal{J} - \left[ \underline{\theta}_f, \frac{\mathrm{d}\underline{x}_f}{\mathrm{d}t} - F(\underline{x}_f) \right]_0^{\frac{T}{2}+\Delta} - \left[ \underline{\theta}_s, \frac{\mathrm{d}\underline{x}_s}{\mathrm{d}t} - F(\underline{x}_s) \right]_{\frac{T}{2}}^{T+\Delta} \ldots \\
& - \left\langle \underline{G}_f, \underline{x}_f(0) - \underline{x}_{f0} \right\rangle - \left\langle \underline{G}_s, \underline{x}_s(T/2) - \underline{x}_{sT/2} \right\rangle
\end{aligned}
\tag{8.43}
$$

Differentiating the Lagrangian with respect to one of the first variables gives:

$$
\begin{aligned}
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{fk}}, \delta \underline{x}_{fk}\right] =\ & \left\langle 2(\underline{x}_{fk}(T/2) - \underline{x}_{sT/2k}), \delta \underline{x}_{fk}(T/2)\right\rangle \ldots \\
& + \left\langle 2(\underline{x}_{fk}(T/2 + \Delta) - \underline{x}_{sk}(T/2 + \Delta)), \delta x_{fk}(T/2 + \Delta)\right\rangle \ldots \\
& - \left\langle 2(\underline{x}_{sk}(T + \Delta) - \underline{x}_{fk}(0 + \Delta)), \delta x_{fk}(0 + \Delta)\right\rangle \ldots \\
& + \left[\frac{\mathrm{d}\underline{\theta}_{fk}}{\mathrm{d}t}, \delta \underline{x}_{fk}\right]_0^{\frac{T}{2}} + \left[\underline{\theta}_{fk}, \frac{\partial F(X_f)}{\partial \underline{x}_{fk}}\delta \underline{x}_{fk}\right]_0^{\frac{T}{2}} - \left\langle \underline{\theta}_{fk}(T/2), \delta \underline{x}_{fk}(T/2)\right\rangle \ldots \\
& + \left\langle \underline{\theta}_{fk}(0), \delta \underline{x}_{fk}(0)\right\rangle - \left\langle G_{fk}, \delta \underline{x}_{fk}(0)\right\rangle
\end{aligned}
\tag{8.44}
$$

Differentiating the Lagrangian with respect to one of the second variables gives:

$$
\begin{aligned}
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{sk}}, \delta \underline{x}_{sk}\right] =\ & \left\langle 2(\underline{x}_{sk}(T) - \underline{x}_{f0k}), \delta \underline{x}_{sk}(T)\right\rangle \ldots \\
& + \left\langle 2(\underline{x}_{sk}(T + \Delta) - \underline{x}_{fk}(0 + \Delta)), \delta \underline{x}_{sk}(T + \Delta)\right\rangle \ldots \\
& - \left\langle 2(\underline{x}_{fk}(T/2 + \Delta) - \underline{x}_{sk}(T/2 + \Delta)), \delta \underline{x}_{sk}(T/2 + \Delta)\right\rangle \ldots \\
& + \left[\frac{\mathrm{d}\underline{\theta}_{sk}}{\mathrm{d}t}, \delta \underline{x}_{sk}\right]_{\frac{T}{2}}^{T} + \left[\underline{\theta}_{sk}, \frac{\partial F(\underline{x}_s)}{\partial \underline{x}_{sk}}\delta \underline{x}_{sk}\right]_{\frac{T}{2}}^{T} - \left\langle \underline{\theta}_{sk}(T), \delta \underline{x}_{sk}(T)\right\rangle \ldots \\
& + \left\langle \underline{\theta}_{sk}(T/2), \delta \underline{x}_{sk}(T/2)\right\rangle - \left\langle G_{sk}, \delta \underline{x}_{sk}(T/2)\right\rangle
\end{aligned}
\tag{8.45}
$$

Summing components of $\delta \underline{x}_{fk}$ and equating them to zero gives the adjoint equations and the adjoint boundary conditions. The adjoint equations are identical in form for the two segments. There are two more conditions than for the formulation without a higher order cost function, which appear as discontinuities in the adjoint variable at $t = T/2 + \Delta,\ T + \Delta$. Differentiating with respect to the two starting conditions gives the last two equations that are required to find the gradient:

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{f0k}}, \delta \underline{x}_{f0k}\right] = \left\langle -2(\underline{x}_{sk}(T) - \underline{x}_{fk0}), \delta \underline{x}_{f0k}\right\rangle + \left\langle G_{fk}, \delta \underline{x}_{f0k}\right\rangle
\tag{8.46}
$$

$$
\left[\frac{\partial \mathcal{L}}{\partial \underline{x}_{sT/2k}}, \delta \underline{x}_{s1/2k}\right] = \left\langle -2(\underline{x}_{fk}(T/2) - \underline{x}_{sT/2k}), \delta \underline{x}_{sT/2k}\right\rangle + \left\langle G_{sk}, \delta \underline{x}_{s1/2k}\right\rangle
\tag{8.47}
$$

Substituting $\underline{\theta}_{fk}(0) = G_{fk}$ and $\underline{\theta}_{sk}(T/2) = G_{sk}$ into the previous equation gives the gradient

equations. The double-shooting method with a higher order cost function is therefore:

$$FOA_{DS} = \begin{cases} \underline{\theta}_f(T/2 + \Delta) = 2(\underline{x}_f(T/2 + \Delta) - \underline{x}_s(T/2 + \Delta)), & \text{Boundary condition}_f \\ \frac{\mathrm{d}\underline{\theta}_{fk}}{\mathrm{d}t} + \sum_i \underline{\theta}_{fi}\frac{\partial F_i(\underline{x}_f)}{\partial \underline{x}_{fk}} = 0, & 0 \leq t \leq T/2 + \Delta \\ \underline{\theta}_f(T/2) = \underline{\theta}_f(T/2) + 2(\underline{x}_f(T/2) - \underline{x}_s(T/2)), & t = T/2 \\ \underline{\theta}_f(0 + \Delta) = \underline{\theta}_f(0 + \Delta) - 2(\underline{x}_s(T + \Delta) - \underline{x}_f(0 + \Delta)), & t = 0 + \Delta \\ \underline{\theta}_s(T + \Delta) = 2(\underline{x}_s(T + \Delta) - \underline{x}_f(0 + \Delta)), & \text{Boundary condition}_s \\ \frac{\mathrm{d}\underline{\theta}_{sk}}{\mathrm{d}t} + \sum_i \underline{\theta}_{si}\frac{\partial F_i(\underline{x}_s)}{\partial \underline{x}_{sk}} = 0, & T/2 \leq t \leq T + \Delta \\ \underline{\theta}_s(T) = \underline{\theta}_s(T) + 2(\underline{x}_s(T) - \underline{x}_f(0)), & t = T \\ \underline{\theta}_s(T/2 + \Delta) = \underline{\theta}_s(T/2 + \Delta) - 2(\underline{x}_f(T/2 + \Delta) - \underline{x}_s(T/2 + \Delta)), & t = T/2 + \Delta \\ \frac{\mathrm{d}\mathfrak{J}}{\mathrm{d}\underline{x}_{f0}} = \underline{\theta}_f(0) - 2(\underline{x}_s(T) - \underline{x}_{f0}), & \text{Gradient evaluation}_f \\ \frac{\mathrm{d}\mathfrak{J}}{\mathrm{d}\underline{x}_{sT/2}} = \underline{\theta}_s(T/2) - 2(\underline{x}_f(T/2) - \underline{x}_{sT/2}), & \text{Gradient evaluation}_s \end{cases}$$

### 8.8.5 General form of multiple shooting method with higher order cost function

The equations derived in the previous section follow a pattern. When dividing the loop into $n$ segments, the state vector for optimisation is of length $nN$, and contains all $n$ starting states $\left[\underline{x}_0, \underline{x}_{\frac{1}{n}}, \underline{x}_{\frac{2}{n}}, \ldots, \underline{x}_{\frac{n-1}{n}}\right]$. In this derivation, $0 < \Delta < T/n$, but the overlap between segments could be derived to overlap several segments. This would increase the timemarching cost, but may result in a more favourable cost function to solve. The cost function becomes:

$$\begin{aligned} \mathfrak{J} &= \langle \underline{x}_n(T) - \underline{x}_0(0) , \ \underline{x}_n(T) - \underline{x}_0(0) \rangle \ldots \\ &+ \sum_{j=1}^{n-1} \langle \underline{x}_j(j/n) - \underline{x}_{j+1}(j/n) , \ \underline{x}_j(j/n) - \underline{x}_{j+1}(j/n) \rangle \ldots \\ &+ \langle \underline{x}_n(T + \Delta) - \underline{x}_0(0 + \Delta) , \ \underline{x}_n(T + \Delta) - \underline{x}_0(0 + \Delta) \rangle \ldots \\ &+ \sum_{j=1}^{n-1} \langle \underline{x}_j(j/n + \Delta) - \underline{x}_{j+1}(j/n + \Delta) , \ \underline{x}_j(1/n + \Delta) - \underline{x}_{j+1}(j/n + \Delta) \rangle \end{aligned}$$

For each of the $n$ segments the same procedure is used to find the gradient of the cost function. For the j$^{\text{th}}$ segment, the process is:

1. Timemarch forward from the initial state, $\underline{x}_{\frac{j-1}{n}}$, from $t = \frac{j-1}{n} \to j/n + \Delta$, using the direct equations:

$$\frac{\mathrm{d}\underline{x}}{\mathrm{d}t} = F(\underline{x}) \tag{8.48}$$

2. Initialise the adjoint state, at time $t = j/n + \Delta$, to be twice the difference between the

end of the overlap of this segment and the equivalent point in the next segment:

$$\underline{\theta}_j(j/n + \Delta) = 2(\underline{x}_j(j/n + \Delta) - \underline{x}_{(j+1)}(j/n + \Delta)) \tag{8.49}$$

This is cyclic, so the overlap of the $n^{th}$ segment is compared with the start of the first.

3. Timemarch backwards with the adjoint equations from time $t = j/n + \Delta \to j^{-1}/n$

$$\frac{\mathrm{d}\underline{\theta}_k}{\mathrm{d}t} + \sum_i \underline{\theta}_{ji} \frac{\partial F_i(\underline{x})}{\partial \underline{x}_k} = 0 \tag{8.50}$$

4. During the timemarch, discontinuities will have to be added to the adjoint state at times $t = j/n$ and $t = j^{-1}/n + \Delta$.

$$\underline{\theta}_j(j/n) = \quad \underline{\theta}_j(j/n) + 2(\underline{x}_j(j/n) - \underline{x}_{(j+1)}(j/n)), \ t = j/n \tag{8.51}$$

$$\underline{\theta}_j(j^{-1}/n + \Delta) = \quad \underline{\theta}_j(j^{-1}/n + \Delta) \ldots \tag{8.52}$$

$$-2(\underline{x}_{(j-1)}(j^{-1}/n + \Delta) - \underline{x}_j(j^{-1}/n + \Delta)), \ t = j^{-1}/n + \Delta$$

5. Evaluate the gradient of the cost function with respect to the $n^{th}$ initial state:

$$\frac{\mathrm{d}\mathcal{L}}{\mathrm{d}\underline{x}_{\frac{j-1}{n}}} = \underline{\theta}_j(j^{-1}/n) - 2(\underline{x}_j(j/n) - \underline{x}_{\frac{j}{n}}) \tag{8.53}$$

6. Assemble the gradients from all segments to form the full gradient vector:

$$\nabla \mathcal{J} = \left[ \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_0}, \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{\frac{1}{n}}}, \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{\frac{2}{n}}}, \ldots, \frac{\mathrm{d}\mathcal{J}}{\mathrm{d}\underline{x}_{\frac{n-1}{n}}} \right] \tag{8.54}$$

7. This gradient is used in an optimisation algorithm to change the starting states, $\left[\underline{x}_0, \underline{x}_{\frac{1}{n}}, \underline{x}_{\frac{2}{n}}, \ldots, \underline{x}_{\frac{n-1}{n}}\right]$. Steps 1-6 are repeated until some predefined level of convergence is reached.

## 8.9   Nonlinear adjoint of the Rijke tube

The nonlinear adjoint equations for the horizontal Rijke tube model (chapter 2) are derived in Appendix B. The derivation is very similar to that of Juniper [17], but with the cost functional for finding limit cycles, and a smoothing applied to the modulus term in the hot wire heat release ($|x| \approx \sqrt{x^2 + \epsilon^2}$ , $\epsilon \ll 1$). The time evolution equations are calculated by C++ routines, with mexfunction wrappers for use in MATLAB.

### 8.9.1   Gradient information

Figure 8.2 shows the gradient information for a 1 Galerkin mode Rijke tube model, for the standard shooting method (a) and the Poincaré plane method (b). The gradient vectors evaluated by the adjoint method agree numerically with those evaluated with finite difference techniques. Figure 8.2 also shows the normalised cost function contours, $\mathcal{J} = ||\underline{x}(T) - \underline{x}(0)||^2 / (||\underline{x}(0)||^2 + \epsilon)$ with $\epsilon = 5 \times 10^{-4}$, for the single shooting method (c) and the Poincaré plane method (d). The stable limit cycle (outer ring) and unstable limit cycle (inner ring) are not obvious in (a) because the initial guess for the period is poor, but the limit cycles are obvious in (b) because the Poincaré plane formulation does not require a guess for the period.

Figure 8.3 shows the change in cost function contours when a phase condition is applied to the standard shooting method (with optimal period). The phase condition is applied as defined in equation (8.13), with phase condition $\eta_1(0) = 0.45$ and weighting $\alpha = 0.02$. Figure 8.3 illustrates both the advantage and disadvantage of this simple phase condition. The advantage is that there are now only two discrete minima on the limit cycle, one at (0.45,0.67) and one at (0.45,-0.67). The disadvantage is that the phase condition may not be fulfilled by all limit cycles, and if it is not, then the local minima will not coincide with that limit cycle. For example, the unstable limit cycle has $\eta_1(t) < 0.45$, $\forall t$, therefore the phase condition and periodic condition cannot both be satisfied. The minima near the unstable limit cycle therefore moves towards 0.45 and locally $J_{min} > 0$. It is therefore recommended that either the phase condition is chosen so that it applies for all limit cycles (for this case $\eta_1(0) = 0$ would suffice), or that the phase condition is only applied once the current starting state is sufficiently close to a limit cycle.

(a) Gradient information and cost function contours for the standard shooting method ($T = 2$).



(b) Gradient information and cost function contours for the Poincaré shooting method.



(c) Logarithmic normalised cost function surface for the standard shooting method ($T = 2$).



(d) Logarithmic normalised cost function surface for the Poincaré shooting method.
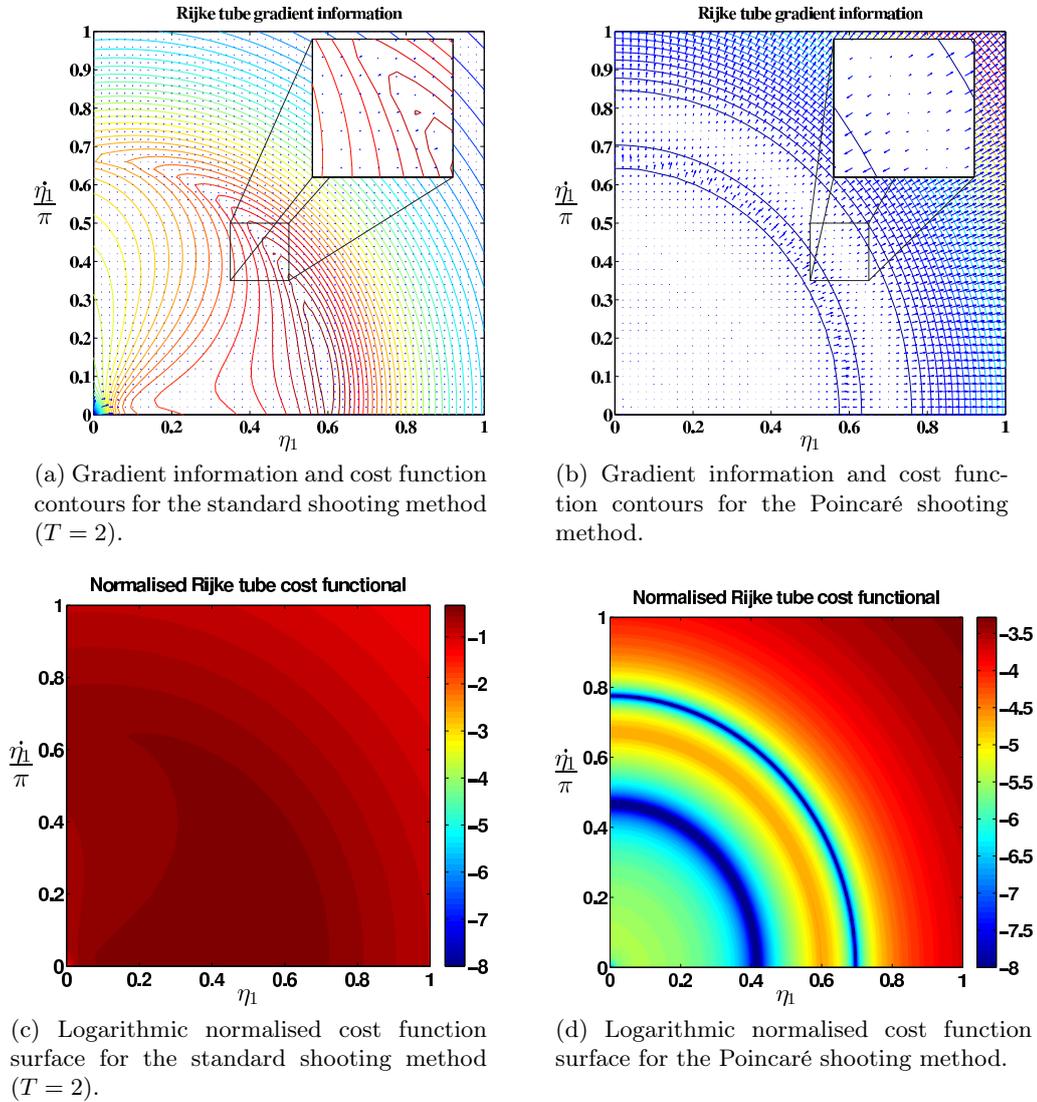
Figure 8.2: Comparison of cost function shape for a single shooting method with $T = 2$ (a,c) and for the Poincaré plane method (b,d). In both cases the gradient vectors are correctly orthogonal to the cost function contours (a,b). With the single shooting method, the location of the limit cycles is not obvious from plot of the cost function (c), because the current guess for the period is 20% too high. With the Poincaré plane formulation in (d), however, the limit cycles and fixed point are obvious; the stable limit cycle is the outer ring and the unstable limit cycle is the inner ring.

(a) Without phase condition.  (b) With phase condition.
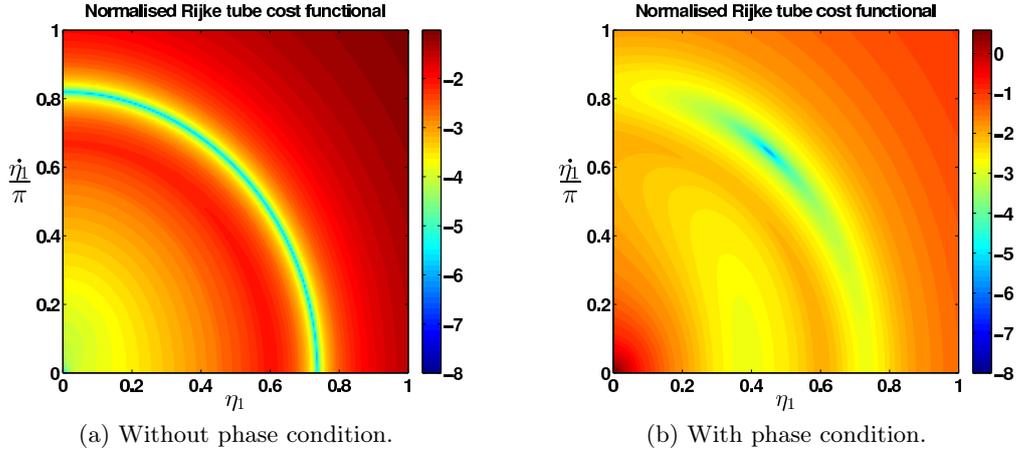
Figure 8.3: Logarithmic normalised cost function contours for the standard shooting method with optimal period ($T = 1.78$), without phase condition (a), and with phase condition (b). The phase condition term in the cost function is $\mathcal{J}_{phase} = \alpha(\eta_1(0)-0.45)^2/(||\underline{x}(0)||^2+\epsilon)$, $\alpha = 0.02$ and $\epsilon = 5\times10^{-4}$.

### 8.9.2 Convergence to a limit cycle

#### 8.9.2.1 Steepest descent algorithm

Figure 8.4 shows convergence to a limit cycle using the steepest descent algorithm for two initial states: (a) where the initial period guess is too low, and (b) where the initial period guess is too high. Both initial states converge to the same limit cycle. The steepest descent algorithm converges similarly when multiple Galerkin modes are included. Figure 8.5 shows the convergence of an initial state to a limit cycle with the steepest descent algorithm and set step length (black) and line search algorithm (pink), using 2 Galerkin modes and the Poincaré shooting method with hyperplane defined by $\eta_1 = 0$. With 2 Galerkin modes the state vector has dimension 4, and the hyperplane therefore has dimension 3. The hyperplane is defined by $\eta_1 = 0$, so that the three dimensional hyperplane has orthogonal axes defined by the remaining variables, $[\eta_2, \frac{\dot{\eta}_1}{\pi}, \frac{\dot{\eta}_2}{2\pi}]$. This hyperplane is not ideal for convergence because it is not perpendicular to the limit cycle. It is clearer to visualise the convergence, however, when the orthogonal axes are defined by $[\eta_2, \frac{\dot{\eta}_1}{\pi}, \frac{\dot{\eta}_2}{2\pi}]$ and not defined by 4D orthogonal vectors projected from an arbitrary plane.

The coloured lines in Figure 8.5 are logarithmic cost function contours, evaluated at slices through the 3D cost function field. A blue colour indicates a low cost function value, a dark red contour indicates a high cost function value. There are two global minima in the figure (both zero), one at the fixed point and one at the limit cycle ($\eta_1 = 0$, $\eta_2 = -0.015$, $\frac{\dot{\eta}_1}{\pi} = 0.83$, $\frac{\dot{\eta}_2}{2\pi} = -0.067$). A local minima can also be seen in the bottom left of (b), when $\frac{\dot{\eta}_1}{\pi} = 0$.

(a) Initial period guess too low ($T_0 = 1.3$)  (b) Initial period guess too high ($T_0 = 2.1$)
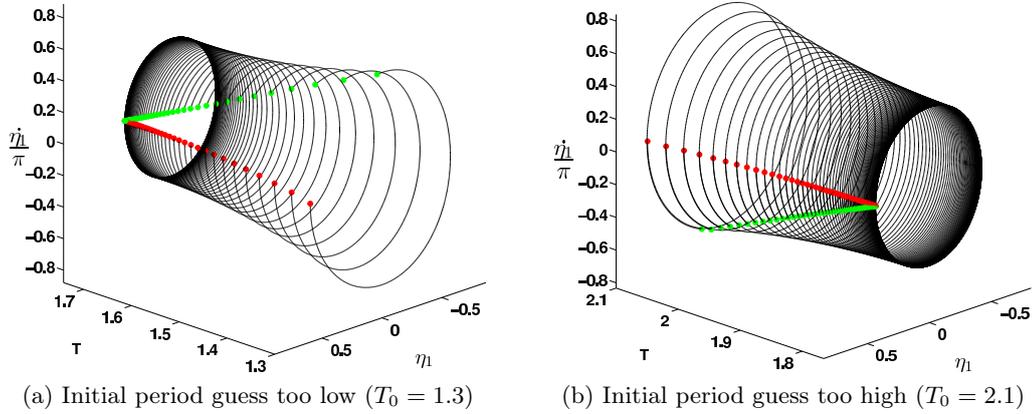
Figure 8.4: Convergence to a limit cycle with 1 Galerkin mode for two cases, with initial period guess too low (a), and initial period guess too high (b). The markers show the initial state (red) and end state (green). Both (a) and (b) converge to the same limit cycle at $T = 1.78$.

From the fixed point the finger of low cost function extends up to $\frac{\dot{\eta}_1}{\pi} \approx 0.5$. This finger is long because the cost function in Figure 8.5 is not normalised.

The black line shows the trajectory of the initial state if a fixed step length is used to generate the next initial state, $\underline{x}^{k+1} = \underline{x}_0^k - \delta \underline{\nabla \mathcal{J}}$, where $\delta$ is small. This is the smoothest trajectory to reach the minima using the gradient information. An efficient optimisation routine will not follow this trajectory, however, because it requires too many gradient evaluations. The pink line shows the trajectory of the initial state if a line minimisation routine is used to generate the next initial state. The minimisation routine finds the value $\alpha$ that minimises $\mathcal{J}(\underline{x}_0^k + \alpha \underline{\nabla \mathcal{J}})$.

The steepest descent algorithm converges relatively inefficiently, because the cost function has a tubular shape (the 3D equivalent of a 2D valley). When the cost function forms a valley shape, steepest descent routines are inefficient because they zig-zag back and forth between the two slopes of the valley (Figure 8.5(a)).

### 8.9.2.2  Conjugate gradient algorithm

Figure 8.6 shows the convergence of an initial state to a limit cycle with the steepest descent algorithm and set step length (black) and conjugate gradient algorithm (pink). Unlike the steepest descent algorithm in Figure 8.5, the conjugate gradient algorithm trajectory does not zig-zag. It is therefore more efficient because it requires fewer gradient evaluations to converge. The conjugate gradient algorithm is efficient for cost functions that form a valley shape, because the trajectory first descends to the valley floor, then it moves along the valley floor.

(a)

(b)

(c)
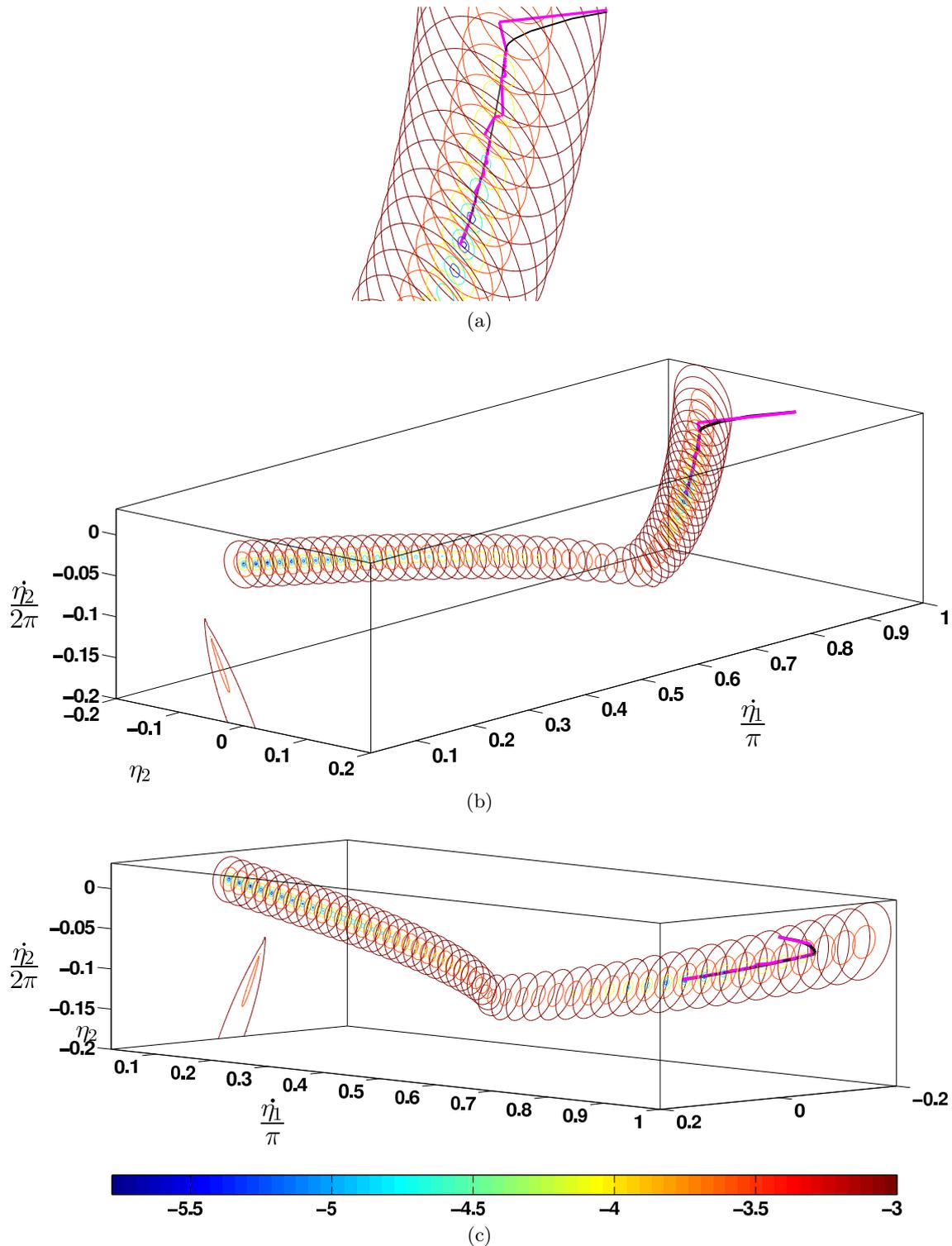
Figure 8.5: Convergence to a limit cycle with the steepest descent algorithm and set step length (black) and line search algorithm (pink), using 2 Galerkin modes and the Poincaré shooting method with hyperplane defined by $\eta_1 = 0$. Figures (a-c) show the same information from different angles. The colours denote logarithmic cost function contours at slices through the 3D cost function field. The initial state is $\underline{x} = [0, 0, 1, 0]$

166
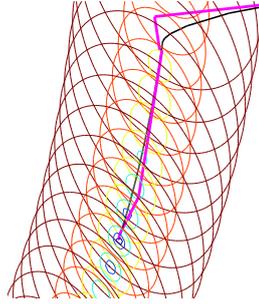
Figure 8.6: Convergence to a limit cycle as in Figure 8.5, but with conjugate gradient algorithm (pink). The conjugate gradient algorithm trajectory does not zig-zag in the same manner as the steepest descent algorithm in Figure 8.5a, so is therefore more efficient.
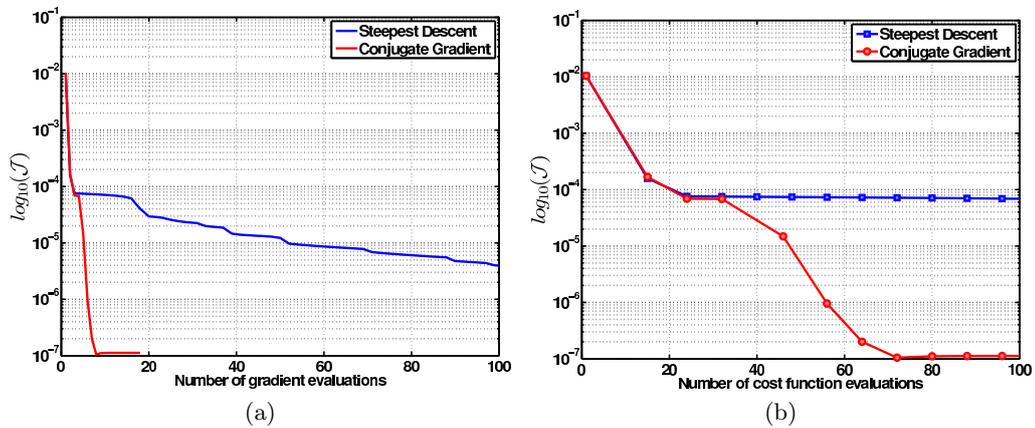


Figure 8.7: Cost function value against number of gradient evaluations (a) and number of cost function evaluations (b), for the steepest descent algorithm (blue) and conjugate gradient algorithm (red). The conjugate gradient algorithm converges much faster than the steepest descent algorithm. The markers in (b) indicate when new gradients are evaluated and a new line search is begun.

For the trajectories shown in Figures 8.5 and 8.6, Figure 8.7(a) shows the cost function convergence as a function of number of gradient evaluations, for both the steepest descent algorithm (with line search) and the conjugate gradient algorithm. The convergence with the first 4 gradient evaluations is almost identical, because the trajectories follow roughly the same path into the tube of low cost function. Once inside the tube, the conjugate gradient algorithm continues to converge quickly, whereas the steepest descent algorithm begins to converge slowly. Figure 8.7(b) shows the the cost function convergence as a function of number of cost function evaluations. For both the steepest descent and conjugate gradient algorithms, there are more cost function evaluations than gradient evaluations because only cost function evaluations are used in the line search routine.

## 8.10 Conclusions

This chapter describes a gradient method for finding limit cycles in large thermoacoustic systems. This method is capable of converging to limit cycles or fixed points (or local minima), even with poor initial guesses. In contrast, the continuation method of chapter 5 can diverge with a poor initial guess, because it is based on a Newton method. Because the gradient method will converge to either fixed points or limit cycles, it is useful for picking out influential features in a nonlinear system. It is much faster at finding these features than brute force approaches, where the system is marched forward in time until it tends to a stable attractor. The gradient method will also find unstable fixed points and unstable limit cycles, which is extremely difficult with timemarching alone. For strongly nonlinear systems, there may be local minima that do not correspond to fixed points or limit cycles. These local minima are simple to identify, however, because their cost function value is not zero.

The convergence to a limit cycle can be improved by adding terms to the cost functional, for example, by adding a normalisation term, by adding time derivatives or by adding additional states to optimise over a path.

The gradient methods are demonstrated on the horizontal Rijke tube. The gradient methods converge quickly to limit cycles and fixed points. The conjugate gradient algorithm converges more quickly than the steepest descent algorithm.

Implementing the adjoint equations from scratch for a new model is a time consuming process, especially when the model is large and elaborate checkpointing routines are required. The gradient methods of this chapter are therefore significantly harder to implement than the continuation methods of Chapter 5. This may not always remain the case, however, because recent developments in the field of algorithmic differentiation [138, 139] can calculate and timemarch the adjoint equations automatically. These algorithmic differentiation routines are implemented in the FEniCS finite element package, which can solve arbitrary PDEs [140]. Implementing the gradient methods of this chapter on a large model using the FEniCS package would be an area worthy of future study.

In summary, the matrix-free methods described in chapter 5 and the gradient methods described in this chapter complement each other: the matrix-free methods converge very fast when close to a limit cycle, but diverge when far from a limit cycle; the gradients methods converge slower when close to a limit cycle, but converge fast when far from a limit cycle. The gradient methods can therefore be used to generate good initial guesses for the matrix-free continuation methods.

For the application in this thesis, however, the matrix-free methods were chosen and the gradient methods in this chapter were not implemented on the more complex thermoacoustic models.

# Chapter 9

# Conclusions

The behaviour of thermoacoustic systems is determined by the attractors in phase space and by the mechanisms through which the system transitions between them. The first half of the thesis concentrated on the mechanism of triggering, whereby a thermoacoustic system transitions from a stable fixed point to a stable limit cycle (or other high amplitude state). The second half of the thesis then concentrated on how to find the attractors of thermoacoustic systems, in particular on methods of finding limit cycles in large thermoacoustic systems.

## 9.1   Overview of work completed

The first part of the thesis examined triggering in thermoacoustics from a dynamical systems perspective. In chapters 3 and 4, unstable limit cycles are identified in the Rijke tube model and are shown to be unstable attractors. These unstable limit cycles exist in bistable regions where the system has undergone a subcritical Hopf bifurcation. A mechanism of triggering is presented, whereby the system transitions from the stable fixed point to the stable limit cycle, via the unstable limit cycle. The unstable limit cycle is an unstable attractor, and because the system is non-normal and the nonlinearities do not conserve energy, perturbations with lower energy than the unstable limit cycle can transition to the stable limit cycle in three steps: first, the perturbations transiently grow in amplitude; second, the system is then attracted to the unstable limit cycle; third, the system is then propelled away from the unstable limit cycle towards the stable limit cycle.

This mechanism allows small perturbations or low amplitude noise to cause triggering to large amplitude limit cycles. In chapter 3, the Rijke tube is forced with low amplitudes of periodic noise and this mechanism is observed. For the Rijke tube, both low frequency noise and high frequency noise can cause triggering, but low frequency noise is more dangerous

because it causes triggering at lower amplitudes. This is unsurprising because the Rijke tube damping model acts like a low pass filter. The noise signature of flames has been shown to have higher amplitudes at low frequencies [64] so these results are pertinent for ducted flame systems.

In chapter 4, the Rijke tube model is forced with low amplitudes of random noise and the stochastic nature of triggering is examined. Triggering is seen to be strongly dependent on the strength of noise and the hot wire temperature. Furthermore, with low amplitudes of noise the system is shown to be practically unstable below the linear stability limit. As noise strength increases, the system becomes practically unstable further from the Hopf point, increasing the region where the fixed point is linearly stable but triggering is likely to occur. As noise strength increases further it becomes possible for the system to unsettle itself from the stable limit cycle and return to the stable fixed point via the unstable limit cycle. This results in short bursts of high energy oscillations.

Stochastic stability maps are introduced as method of visualising the stability of a thermoacoustic system when it is forced by stochastic perturbations. Stochastic stability maps quantify the probability of the state of the system lying in a particular region, and highlight the bimodal PDFs in operating regions with potential for triggering. The stable limit cycle amplitudes under small stochastic perturbations agree well with those predicted by the noiseless bifurcation diagram. If the level of noise in a real combustion system, such as a gas turbine, is known, then a stochastic stability map could be used to predict a region of safe operation. Alternatively, multiple experimental results can be used in a stochastic stability map to visualise the practical stability of the system. Stochastic stability maps have since been created experimentally by Jegadeesan and Sujith [65, 66] for a ducted diffusion flame. Both the triggering mechanism and the predicted stochastic dependence were shown to agree excellently with the theoretical predictions from the Rijke tube model.

Parametric and multiplicative noise are also shown to cause triggering. Non-dimensional wire-temperature noise and multiplicative velocity noise have a larger effect than parametric noise in the time delay and damping. Noise terms that excite the energy of the system, such as the additive noise in section 4.3.1, are demonstrated to be the most dangerous.

The results of chapters 3 and 4 demonstrated that if a system is multistable, sources of noise must be considered to determine which of the stable attractors the system will reach. To examine noise induced transitions or the effect of non-normality, it is first necessary to determine which attractors are present by characterising the nonlinear behaviour of the system. The most common attractors in thermoacoustics are fixed points and limit cycles, so the second

part of the thesis therefore focused on developing efficient numerical methods to find limit cycles in large thermoacoustic systems. Two different methods were developed in parallel: matrix-free continuation methods (chapters 5-7) and gradient methods (chapter 8). Both of these methods are efficient when the system has a large discretisation and timemarching is computationally expensive. It was necessary to develop new methods for analysing nonlinear thermoacoustic systems in the time domain. Existing frequency domain methods, such as the FDF, assume that the acoustics are harmonic, and therefore cannot predict limit cycles that are composed of multiple frequencies, which is often the case for limit cycles, and always the case for period-$2^n$ and quasiperiodic oscillations.

Chapter 5 introduced matrix-free continuation methods, which are a new, quick, and versatile method for finding limit cycles of reasonably sophisticated models of thermoacoustic systems. Furthermore, with increased computing power, they readily scale up to larger models. Continuation methods track the limit cycles and bifurcations of a system as parameters vary, in order to find the safe operating region 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 is likely not true for models with stronger nonlinearities.

The continuation methods are used in chapter 6 to generate a surface of limit cycles for a ducted diffusion flame model, 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 the safe operating region to general changes in acoustic or flame operating conditions.

The continuation methods are also used in chapter 7 to generate a surface of stable and unstable limit cycles for a ducted premixed flame model. Using the Floquet multipliers of the converged limit cycles, the continuation methods showed fold bifurcations, period-doubling bifurcations and Neimark-Sacker bifurcations. A separate surface of period-2 limit cycles was generated, and shown to emerge from the period-doubling bifurcation.

Chapter 8 introduced gradient methods for finding limit cycles of large thermoacoustic systems. The proximity to a limit cycle is formulated as a scalar cost function, and a minimum is found iteratively by using the gradient of the cost function in an optimisation routine. Adjoint equations obtain the gradient of the cost function in two timemarches, one forwards and one backwards. The gradient method efficiently finds a limit cycle or a fixed point (or a local minima), and converges even with a poor initial guess. In contrast, the matrix-free method of chapter 5 is based on Newton iteration, which will diverge with a poor initial guess.

The methods of finding limit cycles that are discussed in this thesis can be ranked in terms of decreasing time for implementation. First, the easiest method to implement is the matrix-free continuation method with finite difference matrix-vector products, because it only requires a timemarching routine for the direct governing equations. This can be used easily with legacy codes. Second, the matrix-free continuation method with first variational matrix-vector products. The matrix-vector products are more accurate, which may be crucial in strongly nonlinear systems, but the first variational equations must be derived, implemented and timemarched in addition to the direct equations. Third, the hardest method to implement is the gradient method for finding limit cycles, because the adjoint equations must be derived, implemented and timemarched in addition to the direct equations. For large systems, elaborate checkpointing routines will also be required. The implementation of the gradient method can be made easier, however, using recent developments in the field of algorithmic differentiation [138, 139].

## 9.2 Summary

This thesis demonstrates that triggering can occur with low levels of background noise, by using unstable attractors as a low energy pathway to reach high energy stable state, which may be periodic, quasiperiodic or chaotic. This thesis therefore highlights the need for thermoacoustic systems to be analysed with nonlinear techniques, because linear techniques cannot calculate the location and form of the unstable attractors, and they cannot calculate the location and form of the final stable state. This thesis therefore describes nonlinear time domain methods for evaluating the safe operating region of coupled thermoacoustic systems. In particular, matrix-free continuation methods are developed and shown to be efficient at calculating limit cycles and period-$2^n$ cycles, and their bifurcations, as a function of system parameters. For computational studies, these time domain methods are both much faster and more accurate than frequency domain equivalents.

## 9.3 Further work

Chapter 7 described the use of continuation methods on a relatively sophisticated model of a laminar premixed flame, albeit with a simple model of the velocity field and a simple model of the acoustic duct. The flame model is extremely versatile - it can model conical flames, gutter flames, axisymmetric dump combustors, and multiple interacting flames side by side. The acoustics in the code are also extremely versatile, due to advances by Illingworth, described next. The LSGEN2D code uses a state space representation of the linear acoustic model, which require two matrices, $A$, $C$, and one vector, $B$, such that:

$$\dot{\underline{x}}_{ac} = A\underline{x}_{ac} + B\dot{q} \quad \text{and} \quad \left[ \begin{array}{c} u_f \\ p_f \end{array} \right] = C\underline{x}_{ac}$$

Illingworth has developed methods of obtaining accurate time domain state space models from frequency domain network models, and in particular, from the LOTAN package [52, 53]. LOTAN is able to provide accurate acoustic models of combustors with complicated geometries, such as ducts with temperature jumps, dump combustors or annular combustors. These acoustic models are used in industry, and are significantly more advanced than the simple Galerkin mode acoustics used in this thesis. Although the LSGEN2D code was modified to use these LOTAN generated acoustic models, there was not time to generate results for this thesis.

Together, the LOTAN generated state space acoustics, the LSGEN2D generated flame dynamics, and the GMRES generated continuation results form a powerful package. The combination of advanced acoustic models, advanced flame models and advanced nonlinear

analysis techniques are a significant improvement on the low order thermoacoustic models in the literature. In particular, they are a significant advance on models that only use a polynomial derived heat release or frequency domain methods to estimate nonlinear behaviour - these cannot capture the rich nonlinear behaviour seen in experimental systems.

This LOTAN/LSGEN2D/GMRES combination is currently being used by Alessandro Orchini to simulate an axisymmetric dump combustor, with the aim of comparing the nonlinear behaviour predicted by the simulations with existing experimental results on the burner of Balachandran [141]. The LOTAN/LSGEN2D/GMRES combination could also be used to compare computational results with the experimental results of the ducted premixed flame of Kabiraj [20, 21]. In particular, there are two aspects of these experimental results that can be compared: first, the behaviour of self-excited experiments and their bifurcation diagrams; second, the flame shapes and heat release signals in forced flame experiments.

The largest weakness in the LSGEN2D flame model is the simplistic velocity model that converts acoustic perturbations at the burner lip to travelling waves that move along the flame surface. An immediate extension to the work in this thesis would be to improve this velocity model by using the Euler equation, which been demonstrated by Preetham [118] in tandem with the G-equation. In particular, an improved velocity model would keep the mean heat release rate of the flame constant and allow the flame speed to depend on flame stretch. The diffusion flame in chapter 6 has very weak nonlinearities, which is unphysical. This improved velocity model could also be used in the diffusion flame model to make the results more physical.

Quasiperiodic oscillations have been recently shown in several experimental thermoacoustic systems [20, 21, 22, 23, 142]. In this thesis, methods of finding Neimark-Sacker bifurcations were presented, which allows regions of quasiperiodicity to be defined. Combining experimental results and modelling results, it is possible to examine the underlying processes in the quasiperiodic regime. A particular example of this is the recent work of Li [143], where a combination of modelling and experiments has helped explain the dynamics of forced flames and jets. Additionally, no methods were presented in this thesis for following branches of quasiperiodic or chaotic oscillations. Another extension would be to develop continuation techniques that can follow quasiperiodic features.

Chapter 8 introduced gradient methods for finding limit cycles in large thermoacoustic systems, which use adjoint equations to calculate gradient information. These methods were not used for the thermoacoustic models of chapters 6 and 7, however, because the matrix-free methods were easier to implement in these cases. Recent developments in the field of algorithmic differentiation by Farrell [138, 139] have made implementation of the adjoint equations

much easier, because they can be derived automatically. These developments not only remove the difficulty of manually implementing adjoint equations and checkpointing routines, they also evaluate the adjoint results faster than if the adjoint equations were implemented by hand. For the finite element package FEniCS [140], which is designed for general PDEs, the dolfin-adjoint package can now automatically calculate first order adjoints and second order adjoints. These algorithms can be easily incorporated into optimisation routines to find limit cycles and their bifurcations using gradient methods. Because the second order adjoint is available, it is also possible to analyse the sensitivity of limit cycles to external perturbations.

It should also be noted that these algorithmic differentiation methods can also automatically calculate the first variational and second variational equations (tangent linear and tangent quadratic equations). As described in chapters 5 and 6, the first variational equations can be used to give accurate matrix-vector products for the continuation methods. The second variational equations can be used to give accurate matrix-vector products when using continuation methods to find bifurcations. If a package of continuation methods were written as a Python wrapper for the FEniCS/dolfin-adjoint packages, then it would be an extremely powerful tool for bifurcation analysis of generic PDEs with finite element discretisations.

# References

[1] 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.

[2] I. C. Waugh, M. P. Juniper, Triggering in a thermoacoustic system with stochastic noise, Journal of Spray and Combustion Dynamics 3 (3) (2011) .

[3] 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) (2012) .

[4] 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.

[5] A. Salinger, N. Bou-Rabee, RP, LOCA 1.0 Library of continuation algorithms: theory and implementation manual, Sandia National Laboratories, Technical Report (March) (2002) .

[6] 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.

[7] A. Morgans, A. Dowling, Model-based control of combustion instabilities, Journal of Sound and Vibration 299 (1-2) (2007) .

[8] H. Murat Altay, D. E. Hudgins, R. L. Speth, A. M. Annaswamy, A. F. Ghoniem, Mitigation of thermoacoustic instability utilizing steady air injection near the flame anchoring zone, Combustion and Flame 157 (4) (2010) 686–700.

[9] J. P. Hathout, A. M. Annaswamy, M. Fleifil, A. F. Ghoniem, Combustion Instability Active Control Using Periodic Fuel Injection, Journal of Propulsion and Power 18 (2) (2002) 390–399.

[10] 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.

[11] A. P. Dowling, A kinematic model of a ducted flame, Journal of Fluid Mechanics 394 (September 2000) (1999) 51–72.

[12] 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.

[13] 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.

# References

[14] 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.

[15] 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.

[16] C. C. Jahnke, F. E. C. Culick, Application of Dynamical Systems Theory to Nonlinear Combustion Instabilities, Journal of Propulsion and Power 10 (4) (1994) 508–517.

[17] M. P. Juniper, Triggering in the horizontal Rijke tube: non-normality, transient growth and bypass transition, Journal of Fluid Mechanics 667 (2010) 272–308.

[18] 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.

[19] J. D. Sterling, Nonlinear Analysis and Modelling of Combustion Instabilities in a Laboratory Combustor, Combustion Science and Technology 89 (1) (1993) 167–179.

[20] 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.

[21] 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.

[22] H. Gotoda, T. Ueda, Transition from periodic to non-periodic motion of a bunsen-type premixed flame tip with burner rotation, Proceedings of the Combustion Institute 29 (2002) 1503–1509.

[23] H. Gotoda, Y. Asano, K. H. Chuah, G. Kushida, Nonlinear analysis on dynamic behavior of buoyancy-induced flame oscillation under swirling flow, International Journal of Heat and Mass Transfer 52 (23-24) (2009) 5423–5432.

[24] S. Stow, A. Dowling, Thermoacoustic oscillations in an annular combustor, in: Proceedings of ASME Turbo Expo, 2001.

[25] S. Stow, A. Dowling, Low-order modelling of thermoacoustic limit cycles, in: Proceedings of ASME Turbo Expo, 2004.

[26] 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.

[27] T. Lieuwen, Experimental Investigation of Limit-Cycle Oscillations in an Unstable Gas Turbine Combustor, Journal of Propulsion and Power 18 (1) (2002) 61–67.

[28] K. Kashinath, S. Hemchandra, M. P. Juniper, Nonlinear Phenomena in Thermoacoustic Systems with Premixed Flames, in: Proceedings of the ASME Turbo Expo, GT2012-68726, 2012.

[29] 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.

[30] 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.

[31] K. Cliffe, A. Spence, S. Tavener, The numerical analysis of bifurcation problems with application to fluid mechanics, Acta Numerica 9 (00) (2000) 39–131.

[32] 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.

[33] W. Govaerts, Y. Kuznetsov, A. Dhooge, Numerical Continuation of Bifurcations of Limit Cycles in MATLAB, SIAM Journal on Scientific Computing 27 (1) (2005) 231.

[34] F. Culick, V. Burnley, G. Swenson, Pulsed instabilities in solid-propellant rockets, Journal of Propulsion and Power 11 (4) (1995) 657–665.

[35] 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.

[36] T. Bewley, Flow control: new challenges for a new renaissance, Progress in Aerospace Sciences 37 (2001) 21–58.

[37] A. Griewank, A. Walther, Algorithm 799: revolve: an implementation of checkpointing for the reverse or adjoint mode of computational differentiation, ACM Transactions on Mathematical Software 26 (1) (2000) 19–45.

[38] P. Corbett, A. Bottaro, Optimal linear growth in swept boundary layers, Journal of Fluid Mechanics 435 (2001) 1–23.

[39] A. Guégan, P. J. Schmid, P. Huerre, Optimal energy growth and optimal control in swept Hiemenz flow, Vol. 566, 2006.

[40] J. Reuther, A. Jameson, Aerodynamic shape optimization of wing and wing-body configurations using control theory, in: 33rd Aerospace Sciences Meeting and Exhibit, AIAA 95-0123, 1995.

[41] A. Jameson, Aerodynamic shape optimization using the adjoint method, Lectures at the Von Karman Institute (2003) .

[42] V. S. Burnley, F. E. C. Culick, Influence of Random Excitations on Acoustic Instabilities in Combustion Chambers, AIAA Journal 38 (8) (2000) .

[43] T. C. Lieuwen, V. Yang, Combustion instabilities in gas turbine engines, AIAA, 2005.

[44] C. E. Mitchell, L. Crocco, W. A. Sirignano, Nonlinear longitudinal instability in rocket motors with concentrated combustion, Combustion Science and Technology 1 (1969) 35–64.

[45] P. Schmid, D. S. Henningson, Stability and transition in shear flows, Springer, 2001.

[46] D. S. Henningson, S. C. Reddy, On the role of linear mechanisms in transition to turbulence, Physics of Fluids 6 (3) (1994) 1396–1398.

[47] D. Sipp, O. Marquet, P. Meliga, A. Barbagallo, Dynamics and control of global instabilities in open-flows: a linearized approach, Applied Mechanics Review 63 (3) (2010) .

[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.

# References

[49] K. Balasubramanian, R. I. Sujith, Thermoacoustic instability in a Rijke tube: Non-normality and non-linearity, Physics of Fluids 20 (2008) 044103.

[50] F. Nicoud, L. Benoit, C. Sensiau, T. Poinsot, Acoustic Modes in Combustors with Complex Impedances and Multidimensional Active Flames, AIAA Journal 45 (2) (2007) 426–441.

[51] D. Zhao, Transient growth of flow disturbances in triggering a Rijke tube combustion instability, Combustion and Flame 159 (6) (2012) 2126–2137.

[52] S. Stow, A. Dowling, A time-domain network model for nonlinear thermoacoustic oscillations, in: Proceedings of ASME Turbo Expo, 2008.

[53] S. R. Stow, A. P. Dowling, A Time-Domain Network Model for Nonlinear Thermoacoustic Oscillations, Journal of Engineering for Gas Turbines and Power 131 (3) (2009) 031502.

[54] M. Heckl, Non-linear acoustic effects in the Rijke tube, Acustica 72 (1990) 6371.

[55] F. R. S. Lighthill, The response of laminar skin friction and heat transfer to fluctuations in the stream velocity, Proceedings of the Royal Society of London 224 (1156) (1954) 1–23.

[56] L. D. Landau, E. M. Lifshitz, Fluid Mechanics, Pergamon Press, 1959.

[57] M. P. Juniper, Transient growth and triggering in the horizontal Rijke tube, International Journal of Spray and Combustion Dynamics 3 (3) (2011) 209–224.

[58] J. Skufca, J. Yorke, B. Eckhardt, Edge of Chaos in a Parallel Shear Flow, Physical Review Letters 96 (17) (2006) 5–8.

[59] T. Schneider, B. Eckhardt, J. Yorke, Turbulence Transition and the Edge of Chaos in Pipe Flow, Physical Review Letters 99 (3) (2007) 1–4.

[60] Y. Duguet, A. P. Willis, R. R. Kerswell, Transition in pipe flow: the saddle structure on the boundary of turbulence, Journal of Fluid Mechanics 613 (2008) 255–274.

[61] P. Ashwin, M. Timme, Unstable attractors: existence and robustness in networks of oscillators with delayed pulse coupling, Nonlinearity 18 (5) (2005) 2035–2060.

[62] P. Clavin, J. S. Kim, F. A. Williams, Turbulence-Induced Noise Effects on High-Frequency Combustion Instabilities, Combustion Science and Technology 96 (1) (1994) 61–84.

[63] T. Lieuwen, A. Banaszuk, Background noise effects on combustor stability, Journal of Propulsion and Power 21 (1) (2005) 25–31.

[64] R. Rajaram, T. Lieuwen, Parametric Studies of Acoustic Radiation From Premixed Flames, Combustion Science and Technology 175 (12) (2003) 2269–2298.

[65] V. Jegadeesan, R. Sujith, Experimental investigation of noise induced triggering in thermoacoustic systems, Proceedings of the Combustion Institute 34 (2) (2013) 3175–3183.

[66] V. Jegadeesan, Experimental Investigation of Noise Induced transitions in Thermoacoustic System, Masters thesis, IIT Madras (2012).

[67] X.-s. Yang, Practical stability in dynamical systems, Chaos, Solitons & Fractals 11 (2000) 1087–1092.

[68] T. Kapitaniak, J. Brindley, Practical Stability of Chaotic Attractors, Chaos, Solitons & Fractals 9 (1) (1998) 43–50.

[69] T. Kapitaniak, J. Brindley, Practical Stability of Synchronized Chaotic Attractors and its Control, Proceedings of the 7th Mediterranean Conference on Control and Automation (1999) 1224–1230.

[70] M. Mackey, I. Nechaeva, Solution moment stability in stochastic differential delay equations, Physical Review E 52 (4) (1995) 3366–3376.

[71] S. Fedotov, I. Bashkirtseva, L. Ryashko, Stochastic analysis of a non-normal dynamical system mimicking a laminar-to-turbulent subcritical transition, Physical Review E 66 (6) (2002) 1–6.

[72] M. Gaudreault, F. Lépine, J. Viñals, Pitchfork and Hopf bifurcation thresholds in stochastic equations with delayed feedback, Physical Review E 80 (6) (2009) 1–8.

[73] T. Lieuwen, Statistical characteristics of pressure oscillations in a premixed combustor, Journal of Sound and Vibration 260 (1) (2003) 3–17.

[74] F. E. C. Culick, M. V. Heitor, J. H. Whitelaw, Unsteady Combustion, Springer, 1996.

[75] R. Berthet, A. Petrossian, S. Residori, B. Roman, S. Fauve, Effect of multiplicative noise on parametric instabilities, Physica D: Nonlinear Phenomena 174 (1-4) (2003) 84–99.

[76] W. Horsthemke, R. Lefever, Noise-induced transitions, Springer-Verlag Berlin, 2006.

[77] S. Fedotov, I. Bashkirtseva, L. Ryashko, Stochastic analysis of subcritical amplification of magnetic energy in a turbulent dynamo, Physica A: Statistical Mechanics and its Applications 342 (3-4) (2004) 491–506.

[78] P. Subramanian, R. I. Sujith, Non-normality and internal flame dynamics in premixed flame-acoustic interaction, Journal of Fluid Mechanics 679 (2011) 315–342.

[79] K. Kashinath, I. C. Waugh, M. P. Juniper, Nonlinear self-excited thermoacoustic oscillations of a ducted premixed flame: bifurcations and routes to chaos, Submitted to Journal of Fluid Mechanics (2013) .

[80] I. Waugh, S. Illingworth, M. Juniper, Matrix-free continuation for bifurcation analysis of large thermoacoustic systems, Journal of Computational Physics 240 (2013) 225–247.

[81] C. Kelley, Iterative methods for linear and nonlinear equations, Society for Industrial Mathematics, 1995.

[82] K. Georg, Matrix-free numerical continuation and bifurcation, Numerical Functional Analysis and Optimization 22 (3-4) (2001) 303–320.

[83] 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.

[84] 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.

[85] D. Viswanath, Recurrent motions within plane Couette turbulence, Journal of Fluid Mechanics 580 (2007) 339.

[86] G. Chandler, R. Kerswell, Simple invariant solutions embedded in 2D Kolmogorov turbulence, Submitted to the Journal of Fluid Mechanics (2012) .

# References

[87] 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.

[88] 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 (2011) .

[89] R. Lehoucq, D. Sorensen, C. Yang, ARPACK users' guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods (1998) .

[90] G. W. Stewart, A Krylov–Schur Algorithm for Large Eigenproblems, SIAM Journal on Matrix Analysis and Applications 23 (3) (2002) 601–614.

[91] 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.

[92] 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.

[93] 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.

[94] C. Barbera, E. Athanassoula, On the calculation of the linear stability parameter of periodic orbits, Astronomy and Astrophysics 336 (1998) 782–785.

[95] 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.

[96] J. Erhel, K. Burrage, B. P, Restarted GMRES preconditioned by deflation, Journal of Computational and Applied Mathematics 69 (1996) 303–318.

[97] J. Baglama, D. Calvetti, G. Golub, L. Reichel, Adaptively preconditioned GMRES algorithms, SIAM Journal on Scientific Computing 20 (1) (1999) 243–269.

[98] 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.

[99] 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.

[100] 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.

[101] W. Govaerts, Y. Kuznetsov, B. Sijnave, Numerical methods for the generalized Hopf bifurcation, SIAM Journal on Numerical Analysis 38 (1) (2000) 329–346.

[102] A. Griewank, G. Reddien, The Calculation of Hopf Points by a Direct Method, IMA Journal of Numerical Analysis 3 (3) (1983) 295–303.

[103] A. G. Salinger, N. M. Bou-rabee, R. P. Pawlowski, E. D. Wilkes, E. A. Burroughs, R. B. Lehoucq, L. A. Romero, Library of Continuation Algorithms : Theory and Implementation Manual, Sandia Report SAND2002-0 (March) (2002) .

[104] M. Tyagi, S. Chakravarthy, R. I. Sujith, Unsteady combustion response of a ducted non-premixed flame and acoustic coupling, Combustion Theory and Modelling 11 (2) (2007) 205–226.

[105] T. Poinsot, Theoretical and numerical combustion, 2nd Edition, R T Edwards, 2005.

[106] 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.

[107] H. F. Walker, An Adaptation of Krylov Subspace Methods to Path Following Problems, SIAM Journal on Scientific Computing 21 (3) (1999) 1191–1198.

[108] 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.

[109] M. P. Juniper, Triggering in thermoacoustics, International Journal of Spray and Combustion Dynamics 4 (3) (2012) 217–238.

[110] L. Magri, K. Balasubramanian, R. I. Sujith, M. P. Juniper, Non-normality in combusion-acoustic interaction in diffusion flames: a critical revision, Submitted to Journal of Fluid Mechanics (2013) .

[111] K. Kashinath, M. P. Juniper, Nonlinear phenomena in thermoacoustics: A comparison betweem single-mode and multi-mode methods, in: Proceedings of the 19th International Conference on Sound and Vibration, 2012.

[112] S. Hemchandra, Dynamics of turbulent premixed flames in acoustic fields, Ph.D. thesis, Georgia Institute of Technology (2009).

[113] S. Hemchandra, Preetham, T. C. Lieuwen, Response of turbulent premixed flames to harmonic acoustic forcing, Proceedings of the Combustion Institute 31 (1) (2007) 1427–1434.

[114] S. Hemchandra, T. Lieuwen, Premixed flame response to equivalence ratio perturbations, Combustion Theory and Modelling 14 (5) (2010) 681–714.

[115] G. Jiang, C. Shu, Efficient implementation of weighted ENO schemes., Journal of Computational Physics 228 (126) (1995) 202–228.

[116] G.-S. Jiang, D. Peng, Weighted ENO Schemes for Hamilton–Jacobi Equations, SIAM Journal on Scientific Computing 21 (6) (2000) 2126–2143.

[117] S. Zhang, C.-W. Shu, A New Smoothness Indicator for the WENO Schemes and Its Effect on the Convergence to Steady State Solutions, Journal of Scientific Computing 31 (1-2) (2006) 273–305.

[118] Preetham, Modeling the response of premixed flames to flow disturbances, Ph.D. thesis, Georgia Institute of Technology (2007).

[119] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-Based Fast Local Level Set Method, Journal of Computational Physics 155 (2) (1999) 410–438.

[120] X. Wen, High order numerical methods to two dimensional delta function integrals in level set methods, Journal of Computational Physics 228 (11) (2009) 4273–4290.

[121] D. Hartmann, M. Meinke, W. Schröder, The constrained reinitialization equation for level set methods, Journal of Computational Physics 229 (5) (2010) 1514–1535.

# References

[122] D. Hartmann, M. Meinke, W. Schröder, Differential equation based constrained reinitialization for level set methods, Journal of Computational Physics 227 (14) (2008) 6821–6845.

[123] K. Kashinath, S. Hemchandra, Nonlinear thermoacoustics of ducted premixed flames : the influence of perturbation convection speed, Submitted to Combustion and Flame (2013) .

[124] J. Miller, Shape Curve Analysis Using Curvature, Ph.D. thesis, University of Glasgow (2009).

[125] T. Schuller, D. Durox, S. Candel, A unified model for the prediction of laminar flame transfer functions, Combustion and Flame 134 (1-2) (2003) 21–34.

[126] M. S. Nakhla, F. H. Branin, Determining the periodic response of nonlinear systems by a gradient method, International Journal of Circuit Theory and Applications 5 (3) (1977) 255–273.

[127] M. Tadi, On computing periodic orbits, Journal of Sound and Vibration 283 (2005) 495–506.

[128] Y. Lan, P. Cvitanović, Variational method for finding periodic orbits in a general flow, Physical Review E 69 (1) (2004) 1–10.

[129] B. M. Boghosian, L. M. Fazendeiro, J. Lätt, H. Tang, P. V. Coveney, New variational principles for locating periodic orbits of differential equations., Philosophical transactions. Series A, Mathematical, physical, and engineering sciences 369 (1944) (2011) 2211–8.

[130] J. R. Schewchuk, An introduction to the conjugate gradient method without the agonizing pain, Carnegie Mellon University (1994) .

[131] R. H. Byrd, J. Nocedal, R. B. Schnabel, Representations of quasi-Newton matrices and their use in limited memory methods, Mathematical Programming 63 (1-3) (1994) 129–156.

[132] F. Traversa, F. Bonani, Frequency-domain evaluation of the adjoint Floquet eigenvectors for oscillator noise characterisation, IET Circuits, Devices & Systems 5 (1) (2011) 46.

[133] M. Schwemmer, T. Lewis, The theory of weakly coupled oscillators, Phase Response Curves in Neuroscience (2012) 1–31.

[134] D. B. Özyurt, P. I. Barton, Large-Scale Dynamic Optimization Using the Directional Second-Order Adjoint Method, Industrial & Engineering Chemistry Research 44 (6) (2005) 1804–1811.

[135] R. Raffard, C. Tomlin, Second order adjoint-based optimization of ordinary and partial differential equations with application to air traffic flow, Proceedings of the American Control Conference (2005) 798–803.

[136] Z. Wang, I. Navon, F. Le Dimet, X. Zou, The second order adjoint analysis: theory and applications, Meteorology and atmospheric physics 50 (1) (1992) 3–20.

[137] Z. Wang, K. Droegemeier, L. White, The adjoint Newton algorithm for large-scale unconstrained optimization in meteorology applications, Computational Optimization and Applications 10 (3) (1998) 283–320.

[138] P. E. Farrell, S. W. Funke, D. A. Ham, A new approach for developing discrete adjoint models, Submitted to ACM Transactions on Mathematical Software (2012) 1–29.

[139] P. E. Farrell, D. A. Ham, S. W. Funke, M. E. Rognes, Automated derivation of the adjoint of high-level transient finite element programs, Under Review (2012) .

[140] A. Logg, K.-A. Mardal, G. N. Wells, Automated Solution of Differential Equations by the Finite Element Method, Vol. 84 of Lecture Notes in Computational Science and Engineering, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[141] R. Balachandran, Experimental Investigation of the Response of Turbulent Premixed Flames to Acoustic Oscillations, Ph.D. thesis, University of Cambridge (2005).

[142] F. Joos, D. Vortmeyer, Self-Excited Oscillations in Combustion Chambers with Premixed Flames and Several Frequencies, Combustion and Flame 65 (1986) 253–262.

[143] L. Li, M. Juniper, Lock-in and quasiperiodicity in hydrodynamically self-excited flames: Experiments and modelling, in: Proceedings of the Combustion Institute 34, 2013, pp. 947–954.

# Appendix A

# Derivation of the first variational equations for the ducted diffusion flame model

The first variational equations of the ducted diffusion flame model are derived in this appendix. The model is described in chapter 6.

## A.1    Governing equations

The governing equations for the acoustic field and mixture fraction field are given as perturbations from the mean values. The governing equation for the mixture fraction field, where $Z = z + \bar{Z}$ is:

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

with boundary conditions $Z = 1$ at the fuel inlet, $Z = 0$ at the oxidiser inlet, $\partial Z/\partial y|_{+1,-1} = 0$, $\partial Z/\partial x|_{x_c} = 0$. The flame lies on the contour $Z = z + \bar{Z} = Z_{st}$, and is assumed to have an infinite reaction rate.

The heat release from the model is coupled to a simple linear acoustic model of a duct.

$$\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = 0$$
$$\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \zeta p = \beta_T \delta \left( x - x_f \right) \dot{q}$$

where $u(x,t)$, $p(x,t)$, $z(x,y,t)$ and $\dot{q}$ is the heat release from the flame.

# A. DERIVATION OF THE FIRST VARIATIONAL EQUATIONS FOR THE DUCTED DIFFUSION FLAME MODEL

## A.2   First variational equations

The first variational equations with respect to the state are derived in this section. First variations with respect to parameter are not derived for clarity (this would increase complexity significantly because $\bar{Z} = f(Pe, \alpha)$).

The diffusion flame model is a time-dependent system with independent variables, and is governed by a set of nonlinear partial differential equations of the form:

$$\frac{\partial \underline{x}_k}{\partial t} = F_k \left( \underline{x} \right)$$

where in this appendix, $\underline{x}$ is the non discretised state.

Once the system is discretised, we have a set of nonlinear ordinary differential equations, where $\underline{x}_D$ is the discretised state:

$$\frac{d\underline{x}_{Dk}}{dt} = F_k \left( \underline{x}_D \right)$$

Once the system is discretised, the first variational equations are an ODE for the discretised perturbation state $\underline{X}'_D$:

$$\frac{d\underline{X}'_{Dk}}{dt} = \sum_j \frac{\partial F_k \left( \underline{x}_D \right)}{\partial \underline{x}_{Dj}} \underline{X}'_{Dj}$$

In this appendix the first variational equations are derived from the original PDE. The continuous first variational equations are of the form:

$$\frac{\partial \underline{X}'_k}{\partial t} = \sum_j \nabla_{\underline{x}_j} F_k \left( \underline{x} \right) \cdot \underline{X}'_j \tag{A.1}$$

where $\nabla_{\underline{x}_1} F_k \left( \underline{x} \right)$ etc. are covariant derivatives. They are defined as:

$$\nabla_{\underline{x}_j} f \left( \underline{x} \right) \cdot \underline{X}'_j = \lim_{\epsilon \to 0} \frac{f \left( \underline{x}_1, \underline{x}_2, \ldots, \underline{x}_j + \epsilon \underline{X}'_j, \ldots, \underline{x}_N, \right) - f \left( \underline{x}_1, \underline{x}_2, \ldots, \underline{x}_j, \ldots, \underline{x}_N, \right)}{\epsilon}$$

For the ducted diffusion flame model, the non-discretised time-dependent state is $\underline{x} = [u, p, z]$, and the evolution is governed by:

$$\frac{\partial u}{\partial t} = -\frac{\partial p}{\partial x} \tag{A.2}$$

$$\frac{\partial p}{\partial t} = -\frac{\partial u}{\partial x} - \zeta p + \beta_T \delta \left( x - x_f \right) \dot{q} \tag{A.3}$$

$$\frac{\partial z}{\partial t} = -\bar{u}_f \frac{\partial z}{\partial x} + \frac{1}{Pe} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) z - u_f \frac{\partial \bar{Z}}{\partial x} - u_f \frac{\partial z}{\partial x} \tag{A.4}$$

The equation for the heat release equation is: (simplified due to symmetry):

$$\dot{q} = \frac{1}{1 - Z_{st}} \left( -\int\int_{Z \geq Z_{st}} \frac{\partial z}{\partial t} \, dy dx + u \int_{\bar{f}^-}^{\bar{f}^+} \left( \bar{Z} - Z_{st} \right) dy \right)$$

$$= \frac{2}{1 - Z_{st}} \left( -\int \int_0^{f^+} \frac{\partial z}{\partial t} \, dy dx + u_f \int_0^{\bar{f}^+} \left( \bar{Z} - Z_{st} \right) dy \right)$$

where $\bar{f}^+$ refers to the positive $y$ location of the steady state flame, and $f^+$ the time-dependent positive $y$ location of the flame. Because the flame surface is defined by $Z = z + \bar{Z} = Z_{st}$, $f^+$ depends on the $z$ field.

The first variational equations describe the evolution of the first variational state $\underline{X}' = [U', P', Z']$.

## A.2.1  Heat release equation

The value of the heat release is a functional, $\dot{q}[u, z]$. The functional derivatives of $\dot{q}$ are required for the first variational pressure equation (section A.2.3). They are :

$$\dot{q}[u, z] = \frac{2}{1 - Z_{st}} \left( -\int\int_0^{f^+} \frac{\partial z}{\partial t} \, dy dx + u_f \int_0^{\bar{f}^+} \left( \bar{Z} - Z_{st} \right) dy \right)$$

$$\frac{\delta \dot{q}[u, z]}{\delta u}(U') = + \frac{2}{1 - Z_{st}} U'_f \int_0^{\bar{f}^+} \left( \bar{Z} - Z_{st} \right) dy$$

$$\frac{\delta \dot{q}[u, z]}{\delta p}(P') = 0$$

$$\frac{\delta \dot{q}[u, z]}{\delta z}(Z') = \lim_{\epsilon \to 0} \frac{\left( -\frac{2}{1 - Z_{st}} \int \int_0^{f^+|_{z + \epsilon Z'}} \frac{\partial(z + \epsilon Z')}{\partial t} \, dy dx \right) - \left( -\frac{2}{1 - Z_{st}} \int \int_0^{f^+} \frac{\partial z}{\partial t} \, dy dx \right)}{\epsilon} \tag{A.5}$$

The differential w.r.t $u$ is constant in time and equal to the steady state heat release. The most difficult differential is that w.r.t. $z$ as both the integrand and the limits of the integral are functions of $z$. Partially differentiating an integral, where both the integrand and limits are functions of a variable has a standard result (Leibniz integration rule ). For example, the Reynolds transport theorum gives for three dimensions:

$$\frac{\partial}{\partial t} \int_{D(t)} F(\underline{x}, t) \, dV = \int_{D(t)} \frac{\partial F(\underline{x}, t)}{\partial t} \, dV + \int_{B(t)} F(\underline{x}, t) \left( \underline{v} \cdot \hat{\underline{n}} \right) dA$$

## A. DERIVATION OF THE FIRST VARIATIONAL EQUATIONS FOR THE DUCTED DIFFUSION FLAME MODEL

This equation gives the formula for the rate of change of a volume integral, with a time dependent domain $(D(t))$. The result is equal to the integral of the rate of change of the function within the domain, plus the rate of expansion of the domain (in terms of volume) multiplied by the function value at the domain boundary $(B(t))$. Considering a point on the boundary, $\underline{B}$, the velocity is just $\underline{v} = \frac{\partial B}{\partial t}$.

Equation (A.5) does not require a differential with respect to time, but it does require a differential with respect to the field $z$. In a similar manner to the Reynolds transport theorum, equation (A.5) therefore becomes:

$$
\begin{aligned}
\frac{\delta \dot{q}\,[u,z]}{\delta z}(Z') &= \lim_{\epsilon \to 0} \frac{\left( -\frac{2}{1-Z_{st}} \int \int_0^{f^+|_{z+\epsilon Z'}} \frac{\partial (z+\epsilon Z')}{\partial t}\, dy dx \right) - \left( -\frac{2}{1-Z_{st}} \int \int_0^{f^+} \frac{\partial z}{\partial t}\, dy dx \right)}{\epsilon} \\
&= -\frac{2}{1-Z_{st}} \left( \int \int_0^{f^+} \frac{\partial Z'}{\partial t}\, dy dx + \int_{f+} \frac{\partial z}{\partial t} \left( (\nabla_z f^+ \cdot Z') \cdot \hat{\underline{n}} \right)\, dl \right) \quad (A.6)
\end{aligned}
$$

The flame location is a function of streamwise distance, $f^+(x)$, so the covariant derivative with respect to the mixture fraction field is required to show how the flame contour moves when a perturbation, $Z'$, is added to the $Z$ field.

The unit normal to the flame contour (evaluated at the flame surface) is:

$$
\hat{\underline{n}} = -\frac{\nabla Z}{|\nabla Z|}
$$

When perturbed in $z$, the flame contour moves in the opposite direction to the gradient of $Z$:

$$
\left( \nabla_z f^+ \cdot Z' \right) = \frac{\hat{\underline{n}}}{|\nabla Z|} Z' \delta_{f+}
$$

where $\delta_{f+}$ represents the spatial location of the flame. In other words, an increase of $\delta z$ moves the flame contour in the direction of $\hat{\underline{n}}$ with magnitude $\delta z\,|\nabla Z|^{-1}$.

Equation (A.6) therefore simplifies to:

$$
\begin{aligned}
\frac{\delta \dot{q}\,[u,z]}{\delta z}(Z') &= -\frac{2}{1-Z_{st}} \left( \int \int_0^{f^+} \frac{\partial Z'}{\partial t}\, dy dx + \int_{f+} \frac{\partial z}{\partial t} \left( \frac{\hat{\underline{n}}}{|\nabla Z|} \cdot \hat{\underline{n}} \right) Z'\, dl \right) \\
&= -\frac{2}{1-Z_{st}} \left( \int \int_0^{f^+} \frac{\partial Z'}{\partial t}\, dy dx + \int_{f+} \frac{\partial z}{\partial t} |\nabla Z|^{-1} Z'\, dl \right)
\end{aligned}
$$

### A.2.2 Velocity equation

$$\frac{\partial u}{\partial t} \quad = \quad -\frac{\partial p}{\partial x} = F_1(u, p, z)$$

$$\nabla_u F_1\left(\underline{x}\right) \cdot U' = \nabla_z F_1\left(\underline{x}\right) \cdot Z' = 0$$

The velocity equation is only affected by the pressure, so the first variational velocity equation is trivial:

$$\frac{\partial U'}{\partial t} = -\frac{\partial P'}{\partial x}$$

### A.2.3 Pressure equation

$$\frac{\partial p}{\partial t} \quad = \quad -\frac{\partial u}{\partial x} - \zeta p + \beta_T \delta\left(x - x_f\right) \dot{q} = F_2(u, p, z)$$

$$
\begin{aligned}
\nabla_u F_2\left(\underline{x}\right) \cdot U' &= -\frac{\partial U'}{\partial x} + \beta_T \delta\left(x - x_f\right) \frac{\delta \dot{q}}{\delta u}(U') \\
&= -\frac{\partial U'}{\partial x} + U' \frac{2\beta_T}{1 - Z_{st}} \delta\left(x - x_f\right) \int_0^{\bar{f}^+} \left(\bar{Z} - Z_{st}\right) dy
\end{aligned}
$$

$$
\begin{aligned}
\nabla_p F_2\left(\underline{x}\right) \cdot P' &= -\zeta P' + \beta_T \delta\left(x - x_f\right) \frac{\delta \dot{q}}{\delta p}(P') \\
&= -\zeta P'
\end{aligned}
$$

$$
\begin{aligned}
\nabla_z F_2\left(\underline{x}\right) \cdot Z' &= \beta_T \delta\left(x - x_f\right) \frac{\delta \dot{q}}{\delta z}(Z') \\
&= -\frac{2\beta_T}{1 - Z_{st}} \delta\left(x - x_f\right) \left( \int \int_0^{f^+} \frac{\partial Z'}{\partial t} \, dy dx + \int_{f^+} \frac{\partial z}{\partial t} \left|\underline{\nabla Z}\right|^{-1} Z' \, dl \right)
\end{aligned}
$$

The first variational pressure equation is therefore:

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

The first part of the heat release term is the linearisation of adding a perturbation to the $Z$ field within the current flame limits. The second part of the heat release term is the linearisation of changing the integral limit. If an infinitessimal perturbation $Z'|_{Z_{st}}$ is added to

# A. DERIVATION OF THE FIRST VARIATIONAL EQUATIONS FOR THE DUCTED DIFFUSION FLAME MODEL

the $Z$ value at the flame location, the flame contour moves a distance $\Delta = \left.\frac{Z'}{|\nabla Z|}\right|_{Z_{st}}$ in the direction of $\hat{n}$, and therefore adds an additional $\left.\frac{\partial z}{\partial t}\right|_{Z_{st}} \Delta$ to the area integral. This additional term is then integrated along the flame contour (i.e. it is a line integral).

### A.2.4 Mixture fraction equation

$$
\frac{\partial z}{\partial t} = -\bar{u}_f \frac{\partial z}{\partial x} + \frac{1}{Pe} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) z - u_f \frac{\partial \bar{Z}}{\partial x} - u_f \frac{\partial z}{\partial x} = F_3(u, p, z)
$$

$$
\nabla_u F_3 \left( \underline{x} \right) \cdot U' = -U'_f \frac{\partial \bar{Z}}{\partial x} - U'_f \frac{\partial z}{\partial x}
$$

$$
\nabla_p F_3 \left( \underline{x} \right) \cdot P' = 0
$$

$$
\nabla_z F_3 \left( \underline{x} \right) \cdot Z' = -\bar{u}_f \frac{\partial Z'}{\partial x} + \frac{1}{Pe} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) Z' - u_f \frac{\partial Z'}{\partial x}
$$

The first variational equation for the mixture fraction is therefore:

$$
\frac{\partial Z'}{\partial t} = -U'_f \frac{\partial \bar{Z}}{\partial x} - U'_f \frac{\partial z}{\partial x} - \bar{u}_f \frac{\partial Z'}{\partial x} + \frac{1}{Pe} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) Z' - u_f \frac{\partial Z'}{\partial x}
$$

## A.3 Summary of the PDEs

With state vector $\underline{x} = [u, p, z]$, the evolution is governed by:

$$
\begin{aligned}
\frac{\partial u}{\partial t} &= -\frac{\partial p}{\partial x} \\
\frac{\partial p}{\partial t} &= -\frac{\partial u}{\partial x} - \zeta p + \frac{2\beta_T}{1 - Z_{st}} \delta \left( x - x_f \right) \times \\
&\qquad \left( -\int \int_0^{f^+} \frac{\partial z}{\partial t} \, dy dx + u_f \int_0^{\bar{f}^+} \left( \bar{Z} - Z_{st} \right) dy \right) \\
\frac{\partial z}{\partial t} &= -\bar{u}_f \frac{\partial z}{\partial x} + \frac{1}{Pe} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) z - u_f \frac{\partial \bar{Z}}{\partial x} - u_f \frac{\partial z}{\partial x}
\end{aligned}
$$

With first variational state vector $\underline{X}' = [U', P', Z']$, the first variational evolution is gov-

erned by:

$$\frac{\partial U'}{\partial t} = -\frac{\partial P'}{\partial x}$$

$$\frac{\partial P'}{\partial t} = -\frac{\partial U'}{\partial x} - \zeta P' + \frac{2\beta_T}{1 - Z_{st}} \delta(x - x_f) \times$$
$$\left( -\int \int_0^{f+} \frac{\partial Z'}{\partial t} \, dy dx - \int_{f+} \frac{\partial z}{\partial t} \left| \underline{\nabla Z} \right|^{-1} Z' \, dl + U_f' \int_0^{\bar{f}+} \left( \bar{Z} - Z_{st} \right) dy \right)$$

$$\frac{\partial Z'}{\partial t} = -\bar{u}_f \frac{\partial Z'}{\partial x} + \frac{1}{Pe} \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} \right) Z' - U_f' \frac{\partial \bar{Z}}{\partial x} - U_f' \frac{\partial z}{\partial x} - u_f \frac{\partial Z'}{\partial x}$$

# Appendix B

# Derivation of the adjoint equations for the Rijke tube

## B.1 Nonlinear adjoint of the Rijke tube equations

In this appendix the adjoint equations of the Rijke tube model are derived, with a linearised time delay. The equations are applied in section 8.9. The derivation is similar to that of Juniper [17], but with the cost functional for finding limit cycles, and a smoothing applied to the modulus term in the hot wire heat release.

### B.1.1 Direct governing equations

The Rijke tube governing equations, with linearised time delay are [17]:

$$
F_1 \equiv \frac{\partial u}{\partial t^*} + \frac{\partial p}{\partial x} = 0, \tag{B.1}
$$

$$
F_2 \equiv \frac{\partial p}{\partial t^*} + \frac{\partial u}{\partial x} + \zeta p - \beta \left( \left| \frac{1}{3} + v_f(t^*) \right|^{\frac{1}{2}} - \left( \frac{1}{3} \right)^{\frac{1}{2}} \right) \delta_D(x - x_f) = 0 \tag{B.2}
$$

$$
F_3 \equiv v(t^*) - \left( u(t^*) - \tau \frac{\partial u(t^*)}{\partial t^*} \right) = 0 \tag{B.3}
$$

The heat release term is now modified to give a smooth transition around $v_f = 0$, which makes the time evolution of the adjoint variables smooth (sudden jumps appear if the modulus sign is kept).

$$
\left| \frac{1}{3} + v_f(t^*) \right|^{\frac{1}{2}} \approx \left( \left( \frac{1}{3} + v_f(t^*) \right)^2 + \epsilon^2 \right)^{\frac{1}{4}} \tag{B.4}
$$

197

## B. DERIVATION OF THE ADJOINT EQUATIONS FOR THE RIJKE TUBE

The parameter $\epsilon$ is constant and must be chosen to be small. In the current formulation $\epsilon = 0.001$.

When searching for limit cycles, these equations are integrated from $t^* = 0$ to $t^* = T$. In the following derivation the time is non-dimensionalised by $T$ such that $t^* = Tt$ and $t \in [t_0, t_1]$, where $t_0 = 0$ and $t_1 = 1$. This makes it easier to express $T$ as a variable in the problem rather than being hard-wired into the integration:

$$
\begin{align}
F_1 &\equiv \frac{1}{T}\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = 0, \tag{B.5}\\
F_2 &\equiv \frac{1}{T}\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \zeta p \ldots \notag\\
&\quad -\beta\left(\left(\left(\frac{1}{3} + v_f(t)\right)^2 + \epsilon^2\right)^{\frac{1}{4}} - \left(\frac{1}{3}\right)^{\frac{1}{2}}\right)\delta_D(x - x_f) = 0 \tag{B.6}\\
F_3 &\equiv v(t) - \left(u(t) - \frac{\tau}{T}\frac{\partial u(t)}{\partial t}\right) = 0 \tag{B.7}
\end{align}
$$

The gradients of the cost function are required with respect to initial values $u(t_0)$ and $p(t_0)$. It helps to separate these initial values from the continuous functions $u(t)$ and $p(t)$. The initial values are denoted $u_0$ and $p_0$, and are constrained to be the starting states by:

$$
\begin{align}
G_1 &\equiv u(t_0) - u_0 = 0 \tag{B.8}\\
G_2 &\equiv p(t_0) - p_0 = 0. \tag{B.9}
\end{align}
$$

### B.1.2 Definition of inner products

The following inner products are defined on $x \in [x_0, x_1]$ and $t \in [t_0, t_1]$, where $x_0 = t_0 = 0$ and $x_1 = t_1 = 1$:

$$
\begin{align}
\langle g, h \rangle &\equiv \int_{x_0}^{x_1} gh \, \mathrm{d}x, \tag{B.10}\\
\{g, h\} &\equiv \int_{t_0}^{t_1} gh \, \mathrm{d}t, \tag{B.11}\\
[g, h] &\equiv \int_{x_0}^{x_1}\int_{t_0}^{t_1} gh \, \mathrm{d}t \, \mathrm{d}x. \tag{B.12}
\end{align}
$$

### B.1.3 Definition of the cost functional

The simplest cost functional for finding limit cycles is defined as:

$$
\mathcal{J} \equiv \langle (u(t_1) - u_0)^2 + (p(t_1) - p_0)^2 \rangle. \tag{B.13}
$$

More complicated cost functionals will include normalisation, phase conditions or higher order terms as described in chapter 8.

### B.1.4   Definition of variations

The functional derivative of $\mathcal{J}$ with respect to $u$, for instance, is defined as:

$$\left[\frac{\partial \mathcal{J}}{\partial u}, \delta u\right] \equiv \lim_{\epsilon \to 0} \int_{x_0}^{x_1} \int_{t_0}^{t_1} \frac{\mathcal{J}(u + \epsilon \delta u) - \mathcal{J}(u)}{\epsilon} \, \mathrm{d}t \, \mathrm{d}x \tag{B.14}$$

### B.1.5   Definition of the Lagrangian functional

The Lagrangian functional is defined as the cost functional minus a set of constraints. The constraints are zero if the direct equations are satisfied. The Lagrangian is:

$$\mathcal{L} \equiv \mathcal{J} - [a, F_1] - [b, F_2] - [c, F_3] - \langle d, G_1 \rangle - \langle e, G_2 \rangle \tag{B.15}$$

The Lagrange multipliers $a, b$ and $c$ can be interpreted as the system's sensitivity to perturbations of the direct governing equations, $F_1$, $F_2$ and $F_3$. The Lagrange multipliers $d$ and $e$ can be interpreted as the system's sensitivity to perturbations of the constraints on the initial state, $G_1$ and $G_2$.

To calculate the gradient of the cost function, the following conditions must be satisfied for general $\delta u$, $\delta p$, and $\delta T$:

$$\left[\frac{\partial \mathcal{L}}{\partial u}, \delta u\right] = 0, \tag{B.16}$$

$$\left[\frac{\partial \mathcal{L}}{\partial p}, \delta p\right] = 0, \tag{B.17}$$

$$\tag{B.18}$$

These will be evaluated by re-arranging (B.15). Applying these two conditions (B.16–B.17) will give the adjoint equations and adjoint boundary conditions, and evaluating $\frac{\partial \mathcal{L}}{\partial u_0}$, $\frac{\partial \mathcal{L}}{\partial p_0}$ and $\frac{\partial \mathcal{L}}{\partial T}$ will give gradient information.

## B. DERIVATION OF THE ADJOINT EQUATIONS FOR THE RIJKE TUBE

First, the contributions to (B.16–B.17) from the cost functional, $\mathcal{J}$, is derived.

$$
\begin{aligned}
\left[\frac{\partial \mathcal{J}}{\partial u}, \delta u\right] &= \lim_{\epsilon \to 0} \int_{x_0}^{x_1} \int_{t_0}^{t_1} \frac{\mathcal{J}(u + \epsilon \delta u) - \mathcal{J}(u)}{\epsilon} \, \mathrm{d}t \mathrm{d}x \\
&= \lim_{\epsilon \to 0} \int_{x_0}^{x_1} \frac{(u(t_1) - u_0 + \epsilon \delta u(t_1))^2 - (u(t_1) - u_0)^2}{\epsilon} \, \mathrm{d}x \\
&= \int_{x_0}^{x_1} 2(u(t_1) - u_0)\delta u(t_1) \, \mathrm{d}x \\
&= \langle \delta u(t_1), 2u(t_1) - 2u_0 \rangle \quad\quad\quad\quad\quad\text{(B.19)}
\end{aligned}
$$

$$
\left[\frac{\partial \mathcal{J}}{\partial p}, \delta p\right] = \langle \delta p(t_1), 2p(t_1) - 2p_0 \rangle \quad\quad\quad\quad\quad\text{(B.20)}
$$

$$
\begin{aligned}
\left\langle \frac{\partial \mathcal{J}}{\partial u_0}, \delta u_0 \right\rangle &= \lim_{\epsilon \to 0} \int_{x_0}^{x_1} \frac{\mathcal{J}(u_0 + \epsilon \delta u_0) - \mathcal{J}(u_0)}{\epsilon} \, \mathrm{d}x \\
&= \lim_{\epsilon \to 0} \int_{x_0}^{x_1} \frac{(u(t_1) - u_0 - \epsilon \delta u_0)^2 - (u(t_1) - u_0)^2}{\epsilon} \, \mathrm{d}x \\
&= \int_{x_0}^{x_1} -2(u(t_1) - u_0)\delta u_0 \, \mathrm{d}x \\
&= \langle \delta u_0, 2u_0 - 2u(t_1) \rangle \quad\quad\quad\quad\quad\text{(B.21)}
\end{aligned}
$$

$$
\left\langle \frac{\partial \mathcal{J}}{\partial p_0}, \delta p_0 \right\rangle = \langle \delta p_0, 2p_0 - 2p(t_1) \rangle \quad\quad\quad\quad\quad\text{(B.22)}
$$

Second, the linear terms in (B.15) are rearranged, leaving the nonlinear term, $\mathcal{K}$, as yet

undefined.

$$
\begin{aligned}
-[a, F_1] &= -\int_0^1 \int_0^1 a \left( \frac{1}{T} \frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} \right) \, \mathrm{d}t \, \mathrm{d}x \\
&= -\int_0^1 \frac{1}{T} \left( a(t_1)u(t_1) - a(t_0)u(t_0) \right) \, \mathrm{d}x + \int_0^1 \int_0^1 u \frac{1}{T} \frac{\partial a}{\partial t} \, \mathrm{d}t \, \mathrm{d}x \ldots \\
&\quad - \int_0^1 \left( a(x_1)p(x_1) - a(x_0)p(x_0) \right) \, \mathrm{d}t + \int_0^1 \int_0^1 p \frac{\partial a}{\partial x} \, \mathrm{d}t \, \mathrm{d}x \\
&= -\frac{1}{T} \langle a(t_1), u(t_1) \rangle + \frac{1}{T} \langle a(t_0), u(t_0) \rangle - \{a(x_1), p(x_1)\} \ldots \\
&\quad + \{a(x_0), p(x_0)\} + \left[ u, \frac{1}{T} \frac{\partial a}{\partial t} \right] + \left[ p, \frac{\partial a}{\partial x} \right]
\end{aligned}
\tag{B.23}
$$

$$
\begin{aligned}
-[b, F_2] &= -\int_0^1 \int_0^1 b \left( \frac{1}{T} \frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \zeta p - \mathcal{K} \right) \, \mathrm{d}t \, \mathrm{d}x \\
&= -\frac{1}{T} \langle b(t_1), p(t_1) \rangle + \frac{1}{T} \langle b(t_0), p(t_0) \rangle - \{b(x_1), u(x_1)\} \ldots \\
&\quad + \{b(x_0), u(x_0)\} + \left[ p, \frac{1}{T} \frac{\partial b}{\partial t} \right] + \left[ u, \frac{\partial b}{\partial x} \right] - [p, \zeta b] + \mathcal{K}
\end{aligned}
\tag{B.24}
$$

$$
\begin{aligned}
-[c, F_3] &= -[v, c] + [u, c] - [\frac{\tau}{T} \frac{\partial u}{\partial t}, c] \\
&= -[v, c] + [u, c] + \{\tau c(x_1), p(x_1)\} - \{\tau c(x_0), p(x_0)\} - \left[ \tau \frac{\partial c}{\partial x}, p \right] \\
&= -[v, c] + [u, c] - \left\langle \frac{\tau}{T} c(t_1), u(t_1) \right\rangle + \left\langle \frac{\tau}{T} c(t_0), u(t_0) \right\rangle \ldots \\
&\quad + \left[ \frac{\tau}{T} \frac{\partial c}{\partial t}, u \right]
\end{aligned}
\tag{B.25}
$$

$$
-\langle d, G_1 \rangle = \langle u_0, d \rangle - \langle u(t_0), d \rangle
\tag{B.26}
$$

$$
-\langle e, G_2 \rangle = \langle p_0, e \rangle - \langle p(t_0), e \rangle
\tag{B.27}
$$

The terms in the curly brackets are equal to zero because $p(x_0) = p(x_1) = b(x_0) = b(x_1) = 0$ due to the acoustic boundary conditions.

The contribution to $\mathcal{L}$ of the nonlinear term, $\mathcal{K}$, comes from $-[b, F_2]$:

$$
\mathcal{K} \equiv - \left[ b, -\beta \delta_D(x - x_f) \left( \left( \left( \frac{1}{3} + v_f \right)^2 + \epsilon^2 \right)^{\frac{1}{4}} - \left( \frac{1}{3} \right)^{\frac{1}{2}} \right) \right].
\tag{B.28}
$$

## B. DERIVATION OF THE ADJOINT EQUATIONS FOR THE RIJKE TUBE

The first variation of $\mathcal{K}$ with respect to variations in $v(x,t)$ is:

$$\left(\left(\frac{1}{3} + (v_f + \epsilon\delta v_f)\right)^2 + \epsilon^2\right)^{\frac{1}{4}} = \frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}} \epsilon\,\delta v_f, \qquad (\text{B.29})$$

where $\delta v_f$ is the variation of $v$ evaluated at the hot wire position and $\delta_D$ represents the Dirac

delta:

$$\delta v_f \equiv \int_0^X \delta v\,\delta_D(x - x_f)\,\mathrm{d}x. \qquad (\text{B.30})$$

For all $v_f$, the first variation of $\mathcal{K}$ is therefore:

$$
\begin{aligned}
\left[\frac{\partial\mathcal{K}}{\partial v}, \delta v\right] &\equiv \lim_{\epsilon \to 0}\int_{x_0}^{x_1}\int_{t_0}^{t_1}\frac{\mathcal{K}(v + \epsilon\delta v) - \mathcal{K}(v)}{\epsilon}\,\mathrm{d}t\,\mathrm{d}x \\
&= -\left[b, -\beta\delta_D(x - x_f)\frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\delta v_f\right] \\
&= \int_{x_0}^{x_1}\int_{t_0}^{t_1}\beta b\delta_D(x - x_f)\frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\delta v_f\,\mathrm{d}t\,\mathrm{d}x \\
&= \int_{t_0}^{t_1}\beta b_f\frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\delta v_f\,\mathrm{d}t \\
&= \int_{t_0}^{t_1}\beta b_f\frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\int_{x_0}^{x_1}\delta v\,\delta_D(x - x_f)\,\mathrm{d}x\,\mathrm{d}t \\
&= \int_{t_0}^{t_1}\int_{x_0}^{x_1}\beta b_f\frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\delta v\,\delta_D(x - x_f)\,\mathrm{d}x\,\mathrm{d}t \\
&= \left[\delta v, \beta b_f\frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\delta_D(x - x_f)\right]. \qquad (\text{B.31})
\end{aligned}
$$

In summary, the linear terms are:

$$-[a, F_1] = -\frac{1}{T}\langle a(t_1), u(t_1)\rangle + \frac{1}{T}\langle a(t_0), u(t_0)\rangle + \left[u, \frac{1}{T}\frac{\partial a}{\partial t}\right] + \left[p, \frac{\partial a}{\partial x}\right]$$

$$-[b, F_2] = -\frac{1}{T}\langle b(t_1), p(t_1)\rangle + \frac{1}{T}\langle b(t_0), p(t_0)\rangle + \left[p, \frac{1}{T}\frac{\partial b}{\partial t}\right] + \left[u, \frac{\partial b}{\partial x}\right]\ldots$$
$$- [p, \zeta b] + \mathcal{K}$$

$$-[c, F_3] = -[v, c] + [u, c] - \left\langle\frac{\tau}{T}c(t_1), u(t_1)\right\rangle + \left\langle\frac{\tau}{T}c(t_0), u(t_0)\right\rangle + \left[\frac{\tau}{T}\frac{\partial c}{\partial t}, u\right]$$

$$-\langle d, G_1\rangle = \langle u_0, d\rangle - \langle u(t_0), d\rangle$$
$$-\langle e, G_2\rangle = \langle p_0, e\rangle - \langle p(t_0), e\rangle$$

In the next step of the derivation, the variations in $\mathcal{L}$ w.r.t. variations in $u, p, v, u_0, p_0$ are summed:

$$\left[\frac{\partial \mathcal{L}}{\partial u}, \delta u\right] = +\left[\frac{1}{T}\frac{\partial a}{\partial t}, \delta u\right] + \left[\frac{\partial b}{\partial x}, \delta u\right] + [c, \delta u] + \left[\frac{\tau}{T}\frac{\partial c}{\partial t}, \delta u\right] \quad \ldots$$
$$-\frac{1}{T}\langle a(t_1), \delta u(t_1)\rangle + \frac{1}{T}\langle a(t_0), \delta u(t_0)\rangle \quad \ldots$$
$$-\langle d, \delta u(t_0)\rangle + \langle 2u(t_1) - 2u_0, \delta u(t_1)\rangle \ldots$$
$$-\left\langle\frac{\tau}{T}c(t_1), \delta u(t_1)\right\rangle + \left\langle\frac{\tau}{T}c(t_0), \delta u(t_0)\right\rangle \tag{B.32}$$

$$\left[\frac{\partial \mathcal{L}}{\partial p}, \delta p\right] = +\left[\frac{1}{T}\frac{\partial b}{\partial t}, \delta p\right] - [\zeta b, \delta p] + \left[\frac{\partial a}{\partial x}, \delta p\right] \quad \ldots \tag{B.33}$$
$$-\frac{1}{T}\langle b(t_1), \delta p(t_1)\rangle + \frac{1}{T}\langle b(t_0), \delta p(t_0)\rangle \quad \ldots$$
$$-\langle e, \delta p(t_0)\rangle + \langle 2p(t_1) - 2p_0, \delta p(t_1)\rangle$$

$$\left[\frac{\partial \mathcal{L}}{\partial v}, \delta v\right] = -[c, \delta v] + \left[\beta b_f \frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\delta_D(x - x_f), \delta v\right] \tag{B.34}$$

$$\left[\frac{\partial \mathcal{L}}{\partial u_0}, \delta u_0\right] = \langle d, \delta u_0\rangle + \langle 2u_0 - 2u(t_1), \delta u_0\rangle \tag{B.35}$$

$$\left[\frac{\partial \mathcal{L}}{\partial p_0}, \delta p_0\right] = \langle e, \delta p_0\rangle + \langle 2p_0 - 2p(t_1), \delta p_0\rangle \tag{B.36}$$

The differentials $\frac{\partial \mathcal{L}}{\partial u}$ and $\frac{\partial \mathcal{L}}{\partial p}$ must equal zero (chapter 8). This yields three relationships:

## B. DERIVATION OF THE ADJOINT EQUATIONS FOR THE RIJKE TUBE

the boundary conditions for the adjoint variables at time $t = t_1$, the adjoint time evolution

equations, and a condition for the adjoint variables at time $t = t_0$.

First, the conditions at $t = t_1$ is:

$$
\begin{aligned}
-\frac{1}{T}a(t_1) - \frac{\tau}{T}c(t_1) + (2u(t_1) - 2u_0) &= 0 \\
-\frac{1}{T}b(t_1) + (2p(t_1) - 2p_0) &= 0.
\end{aligned}
$$

Rearranging these conditions gives the boundary conditions for the adjoint variables:

$$
a(t_1) = -\tau c(t_1) + T(2u(t_1) - 2u_0)
$$

$$
b(t_1) = T(2p(t_1) - 2p_0).
$$

Second, for $t \in [0, 1]$ (non-dimensional time), the following conditions must also be satisfied:

$$
\begin{aligned}
+\frac{1}{T}\frac{\partial a}{\partial t} + \frac{\partial b}{\partial x} + c + \frac{\tau}{T}\frac{\partial c}{\partial t} &= 0 \\
+\frac{1}{T}\frac{\partial b}{\partial t} - \zeta b + \frac{\partial a}{\partial x} &= 0 \\
-c + \beta b_f \frac{1}{2}\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}} \delta_D(x - x_f) &= 0
\end{aligned}
$$

Rearranging these conditions gives the time evolution equations for the adjoint variables:

$$
\begin{aligned}
F_1^+ &= \frac{1}{T}\frac{\partial a}{\partial t} + \frac{\partial b}{\partial x} + c + \frac{\tau}{T}\frac{\partial c}{\partial t} = 0 \\
F_2^+ &= \frac{1}{T}\frac{\partial b}{\partial t} + \frac{\partial a}{\partial x} - \zeta b = 0
\end{aligned}
$$

where

$$\frac{\tau}{T}\frac{\partial c}{\partial t} = \frac{\partial b_f}{\partial t}\beta\frac{\tau}{2T}\left(\frac{1}{3}+v_f\right)\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\delta_D(x-x_f)$$

$$+b_f\beta\frac{\tau}{2T}\frac{\partial v_f}{\partial t}\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\delta_D(x-x_f)$$

$$-\frac{3}{2}b_f\beta\frac{\tau}{2T}\frac{\partial v_f}{\partial t}\left(\frac{1}{3}+v_f\right)^2\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{7}{4}}\delta_D(x-x_f)$$

$$= \beta\frac{\tau}{2T}\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\left(\frac{\partial b_f}{\partial t}\left(\frac{1}{3}+v_f\right)+b_f\frac{\partial v_f}{\partial t}\ldots\right.$$

$$\left.-\frac{3}{2}b_f\frac{\partial v_f}{\partial t}\left(\frac{1}{3}+v_f\right)^2\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-1}\right)\delta_D(x-x_f)$$

The contributions from $c$ and $\frac{\partial c}{\partial t}$ can be added and simplified to give:

$$c+\frac{\tau}{T}\frac{\partial c}{\partial t} = \beta\frac{1}{2}\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\left(\left(b_f+\frac{\tau}{T}\frac{\partial b_f}{\partial t}\right)\left(\frac{1}{3}+v_f\right)+\frac{\tau}{T}b_f\frac{\partial v_f}{\partial t}\ldots\right.$$

$$\left.-\frac{\tau}{T}\frac{3}{2}b_f\frac{\partial v_f}{\partial t}\left(\frac{1}{3}+v_f\right)^2\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-1}\right)\delta_D(x-x_f)$$

Note that the derivative $\frac{\partial v_f}{\partial t}$ is included as $v_f$ varies in time.

Third, the condition at $t=t_0$ is:

$$\frac{1}{T}a(t_0)+\frac{\tau}{T}c(t_0)-d = 0$$

$$\frac{1}{T}b(t_0)-e = 0$$

The gradient information for $(u_0, p_0)$ is given by:

$$\frac{\partial\mathcal{L}}{\partial u_0} = d+2u(t_1)-2u_0 = \frac{1}{T}a(t_0)+\frac{\tau}{T}c(t_0)+2u_0-2u(t_1)$$

$$\frac{\partial\mathcal{L}}{\partial p_0} = e+2p(t_1)-2p_0 = \frac{1}{T}b(t_0)+2p_0-2p(t_1)$$

To converge to a limit cycle using the standard shooting method, the quantity $\partial\mathcal{J}/\partial T$ is also

## B. DERIVATION OF THE ADJOINT EQUATIONS FOR THE RIJKE TUBE

required:

$$
\begin{aligned}
\frac{\partial \mathcal{J}}{\partial T} &= \frac{\partial \mathcal{J}}{\partial u(t_1)} \frac{\partial u(t_1)}{\partial T} + \frac{\partial \mathcal{J}}{\partial p(t_1)} \frac{\partial p(t_1)}{\partial T} \\
&= \left\langle 2 \left( u(t_1) - u_0 \right) \frac{\partial u(t_1)}{\partial T} + 2 \left( p(t_1) - p_0 \right) \frac{\partial p(t_1)}{\partial T} \right\rangle
\end{aligned}
$$

It is easier to find the derivatives with respect to $T$ in the original time frame $t^*$.

$$
u(T) = u_0 + \int_0^T \frac{\partial u}{\partial t^*} dt^* \tag{B.37}
$$

$$
\begin{aligned}
\frac{\partial u(T)}{\partial T} &= \frac{\int_0^{T+\delta T} \frac{\partial u}{\partial t^*} dt^* - \int_0^T \frac{\partial u}{\partial t^*} dt^*}{\delta T} \tag{B.38} \\
&= \frac{\int_T^{T+\delta T} \frac{\partial u}{\partial t^*} dt^*}{\delta T} \\
&= \frac{\partial u(T)}{\partial t^*} = \frac{1}{T} \frac{\partial u(t_1)}{\partial t}
\end{aligned}
$$

The gradient of the cost function with respect to period is therefore:

$$
\frac{\partial \mathcal{J}}{\partial T} = \left\langle 2 \left( u(t_1) - u_0 \right) \frac{1}{T} \frac{\partial u(t_1)}{\partial t} + 2 \left( p(t_1) - p_0 \right) \frac{1}{T} \frac{\partial p(t_1)}{\partial t} \right\rangle
$$

**Summary of the adjoint looping algorithm before discretisation**

The adjoint looping algorithm is as follows:

1. Choose a starting state $(u_0, p_0)$

2. Integrate forward to $t_1$ with the direct equations, storing $u$ and $p$:

$$F_1 \equiv \frac{1}{T}\frac{\partial u}{\partial t} + \frac{\partial p}{\partial x} = 0,$$

$$F_2 \equiv \frac{1}{T}\frac{\partial p}{\partial t} + \frac{\partial u}{\partial x} + \zeta p - \beta \left( \left( \left( \frac{1}{3} + v_f \right)^2 + \epsilon^2 \right)^{\frac{1}{4}} - \left( \frac{1}{3} \right)^{\frac{1}{2}} \right) \delta_D(x - x_f) = 0$$

$$F_3 \equiv v - u + \frac{\tau}{T}\frac{\partial u}{\partial t} = 0$$

3. Initialize the adjoint variables at $t = t_1$ with:

$$a(t_1) = -\tau c(t_1) + T(2u(t_1) - 2u_0)$$

$$b(t_1) = T(2p(t_1) - 2p_0).$$

4. Integrate backwards in time to $t_0$ with the adjoint equations:

$$F_1^+ = \frac{1}{T}\frac{\partial a}{\partial t} + \frac{\partial b}{\partial x} + c + \frac{\tau}{T}\frac{\partial c}{\partial t} = 0$$

$$F_2^+ = \frac{1}{T}\frac{\partial b}{\partial t} + \frac{\partial a}{\partial x} - \zeta b = 0$$

$$c + \frac{\tau}{T}\frac{\partial c}{\partial t} = \beta\frac{1}{2}\left( \left( \frac{1}{3} + v_f \right)^2 + \epsilon^2 \right)^{-\frac{3}{4}}\left( \left( b_f + \frac{\tau}{T}\frac{\partial b_f}{\partial t} \right)\left( \frac{1}{3} + v_f \right) + \frac{\tau}{T}b_f\frac{\partial v_f}{\partial t} \cdots \right.$$

$$\left. -\frac{\tau}{T}\frac{3}{2}b_f\frac{\partial v_f}{\partial t}\left( \frac{1}{3} + v_f \right)^2\left( \left( \frac{1}{3} + v_f \right)^2 + \epsilon^2 \right)^{-1} \right)\delta_D(x - x_f)$$

5. Calculate the gradient information at $t = t_0$ with:

$$\frac{\partial \mathcal{L}}{\partial u_0} = \frac{1}{T}a(t_0) + \frac{\tau}{T}c(t_0) + 2u_0 - 2u(t_1)$$

$$\frac{\partial \mathcal{L}}{\partial p_0} = \frac{1}{T}b(t_0) + 2p_0 - 2p(t_1)$$

$$\frac{\partial \mathcal{J}}{\partial T} = \left\langle 2\left( u(t_1) - u_0 \right)\frac{1}{T}\frac{\partial u(t_1)}{\partial t} + 2\left( p(t_1) - p_0 \right)\frac{1}{T}\frac{\partial p(t_1)}{\partial t} \right\rangle$$

### B.1.6 Discretization of the direct equations

The velocity and pressure perturbations are decomposed into basis functions:

$$u(x,t) = \sum_{j=1}^{N} \eta_j(t) \cos(j\pi x), \tag{B.39}$$

$$p(x,t) = -\sum_{j=1}^{N} \left( \frac{\dot{\eta}_j(t)}{j\pi} \right) \sin(j\pi x), \tag{B.40}$$

These expressions are substituted into the governing equations, which are then multiplied by $\sin(k\pi x)$ and integrated over $x \in [0,1]$ using the relationship:

$$\int_0^1 \sin(j\pi x)\sin(k\pi x) \, \mathrm{d}x = \delta_{j,k}/2$$

(This integration is not necessary for the momentum equation, for which the result is obvious):

1. Momentum equation:

$$\frac{1}{T}\frac{\partial}{\partial t}\sum_{j=1}^{N}\eta_j(t)\cos(j\pi x) - \frac{\partial}{\partial x}\sum_{j=1}^{N}\left(\frac{\dot{\eta}_j(t)}{j\pi}\right)\sin(j\pi x) = 0,$$

$$\Rightarrow \sum_{j=1}^{N}\frac{1}{T}\frac{\mathrm{d}\eta_j}{\mathrm{d}t}\cos(j\pi x) - \sum_{j=1}^{N}\left(\frac{\dot{\eta}_j}{j\pi}\right)j\pi\cos(j\pi x) = 0,$$

$$\Rightarrow \sum_{j=1}^{N}\left(\frac{1}{T}\frac{\mathrm{d}\eta_j}{\mathrm{d}t}\cos(j\pi x) - \left(\frac{\dot{\eta}_j}{j\pi}\right)j\pi\cos(j\pi x)\right) = 0,$$

$$\Rightarrow \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\eta_j - j\pi\left(\frac{\dot{\eta}_j}{j\pi}\right) = 0$$

2. Energy equation:

$$0 = -\frac{1}{T}\frac{\partial}{\partial t}\sum_{j=1}^{N}\left(\frac{\dot{\eta}_j(t)}{j\pi}\right)\sin(j\pi x) + \frac{\partial}{\partial x}\sum_{j=1}^{N}\eta_j(t)\cos(j\pi x)\dots$$

$$-\zeta\sum_{j=1}^{N}\left(\frac{\dot{\eta}_j(t)}{j\pi}\right)\sin(j\pi x)\dots$$

$$-\beta\left(\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{\frac{1}{4}}-\left(\frac{1}{3}\right)^{\frac{1}{2}}\right)\delta_D(x-x_f)$$

$$\Rightarrow 0 = -\sum_{j=1}^{N}\frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right)\sin(j\pi x) - \sum_{j=1}^{N}\eta_j j\pi\sin(j\pi x)\dots$$

$$\dots-\sum_{j=1}^{N}\zeta\left(\frac{\dot{\eta}_j}{j\pi}\right)\sin(j\pi x)\dots$$

$$-\beta\left(\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{\frac{1}{4}}-\left(\frac{1}{3}\right)^{\frac{1}{2}}\right)\delta_D(x-x_f)$$

$$\Rightarrow 0 = \int_{x_0}^{x_1}\left(-\sum_{j=1}^{N}\frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right)\sin(j\pi x) - \sum_{j=1}^{N}\eta_j j\pi\sin(j\pi x)\dots\right.$$

$$-\sum_{j=1}^{N}\zeta\left(\frac{\dot{\eta}_j}{j\pi}\right)\sin(j\pi x)\dots$$

$$\left.-\beta\left(\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{\frac{1}{4}}-\left(\frac{1}{3}\right)^{\frac{1}{2}}\right)\delta_D(x-x_f)\right)\sin(k\pi x)\mathrm{d}x$$

$$\Rightarrow 0 = -\frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_k}{k\pi}\right) - \eta_k k\pi - \zeta\left(\frac{\dot{\eta}_k}{k\pi}\right)\dots$$

$$-2\beta\left(\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{\frac{1}{4}}-\left(\frac{1}{3}\right)^{\frac{1}{2}}\right)\sin(k\pi x_f)$$

$$0 = \Rightarrow \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right) + j\pi\eta_j + \zeta\left(\frac{\dot{\eta}_j}{j\pi}\right)\dots$$

$$+2\beta\left(\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{\frac{1}{4}}-\left(\frac{1}{3}\right)^{\frac{1}{2}}\right)\sin(j\pi x_f)$$

3. $v$-equation (non-dimensional time):

$$v_f = u_f - \frac{\tau}{T}\frac{\partial u_f}{\partial t} = \sum_{j=1}^{N}\left(\eta_j - \frac{\tau}{T}\dot{\eta}_j\right)\cos(j\pi x_f)$$

## B. DERIVATION OF THE ADJOINT EQUATIONS FOR THE RIJKE TUBE

As described in chapter 2, a damping coefficient, $\zeta_j$, is defined for each mode. In summary, the discretized direct equations are:

$$
\begin{aligned}
F_{1G} &\equiv \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\eta_j - j\pi\left(\frac{\dot{\eta}_j}{j\pi}\right) = 0 \\
F_{2G} &\equiv \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\dot{\eta}_j}{j\pi}\right) + j\pi\eta_j + \zeta_j\left(\frac{\dot{\eta}_j}{j\pi}\right)\dots \\
&\quad + 2\beta\left(\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{\frac{1}{4}} - \left(\frac{1}{3}\right)^{\frac{1}{2}}\right)\sin(j\pi x_f) = 0 \\
v_f &= \sum_{j=1}^{N}\left(\eta_j - \frac{\tau j\pi}{T}\left(\frac{\dot{\eta}_j}{j\pi}\right)\right)\cos(j\pi x_f)
\end{aligned}
$$

### B.1.7  Discretization of the adjoint equations

The adjoint equations are:

$$
\begin{aligned}
F_1^+ &= \frac{1}{T}\frac{\partial a}{\partial t} + \frac{\partial b}{\partial x} + c + \frac{\tau}{T}\frac{\partial c}{\partial t} = 0 \\
F_2^+ &= \frac{1}{T}\frac{\partial b}{\partial t} + \frac{\partial a}{\partial x} - \zeta b = 0 \\
c + \frac{\tau}{T}\frac{\partial c}{\partial t} &= \beta\frac{1}{2}\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\left(\left(b_f + \frac{\tau}{T}\frac{\partial b_f}{\partial t}\right)\left(\frac{1}{3}+v_f\right) + \frac{\tau}{T}b_f\frac{\partial v_f}{\partial t}\dots\right. \\
&\quad \left. - \frac{\tau}{T}\frac{3}{2}b_f\frac{\partial v_f}{\partial t}\left(\frac{1}{3}+v_f\right)^2\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-1}\right)\delta_D(x-x_f)
\end{aligned}
$$

Using the substitution $h = c + \frac{\tau}{T}\frac{\partial c}{\partial t}$, the adjoint equations can be simplified to:

$$
\begin{aligned}
F_1^+ &= \frac{1}{T}\frac{\partial a}{\partial t} + \frac{\partial b}{\partial x} + h\delta_D(x-x_f) = 0 \\
F_2^+ &= \frac{1}{T}\frac{\partial b}{\partial t} + \frac{\partial a}{\partial x} - \zeta b = 0 \\
h &= \beta\frac{1}{2}\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\left(\left(b_f + \frac{\tau}{T}\frac{\partial b_f}{\partial t}\right)\left(\frac{1}{3}+v_f\right) + \frac{\tau}{T}b_f\frac{\partial v_f}{\partial t}\dots\right. \\
&\quad \left. - \frac{\tau}{T}\frac{3}{2}b_f\frac{\partial v_f}{\partial t}\left(\frac{1}{3}+v_f\right)^2\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-1}\right)
\end{aligned}
$$

Similar to the discretisation of the direct variables, the adjoint variables are decomposed into basis functions:

$$a(x,t) = \sum_{j=1}^{N} \left( \frac{\xi_j}{j\pi} \right) \cos(j\pi x),$$

$$b(x,t) = -\sum_{j=1}^{N} \nu_j \sin(j\pi x),$$

These are substituted into the adjoint equations.

1. $F_2^+$ equation:

$$\frac{1}{T}\frac{\partial b}{\partial t} + \frac{\partial a}{\partial x} - \zeta b = 0$$

$$-\frac{1}{T}\frac{\partial}{\partial t}\sum_{j=1}^{N} \nu_j \sin(j\pi x) + \frac{\partial}{\partial x}\sum_{j=1}^{N} \left( \frac{\xi_j}{j\pi} \right) \cos(j\pi x) + \zeta \sum_{j=1}^{N} \nu_j \sin(j\pi x) = 0$$

$$-\sum_{j=1}^{N} \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\nu_j \sin(j\pi x) - \sum_{j=1}^{N} \left( \frac{\xi_j}{j\pi} \right) j\pi \sin(j\pi x) + \sum_{j=1}^{N} \zeta \nu_j \sin(j\pi x) = 0$$

$$-\frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\nu_j - \left( \frac{\xi_j}{j\pi} \right) j\pi + \zeta \nu_j = 0$$

2. $F_1^+$:

$$0 = \frac{1}{T}\frac{\partial a}{\partial t} + \frac{\partial b}{\partial x} + h\delta_D(x - x_f)$$

$$\Rightarrow 0 = \frac{1}{T}\frac{\partial}{\partial t}\sum_{j=1}^{N} \left( \frac{\xi_j}{j\pi} \right) \cos(j\pi x) - \frac{\partial}{\partial x}\sum_{j=1}^{N} \nu_j \sin(j\pi x) + h\delta_D(x - x_f)$$

$$\Rightarrow 0 = \sum_{j=1}^{N} \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left( \frac{\xi_j}{j\pi} \right) \cos(j\pi x) - \sum_{j=1}^{N} \nu_j j\pi \cos(j\pi x) + h\delta_D(x - x_f)$$

$$\Rightarrow 0 = \int_0^1 \left( \sum_{j=1}^{N} \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left( \frac{\xi_j}{j\pi} \right) \cos(j\pi x) - \sum_{j=1}^{N} \nu_j j\pi \cos(j\pi x) \dots \right.$$

$$\left. + h\delta_D(x - x_f) \right) \cos(k\pi x) \,\mathrm{d}x$$

$$\Rightarrow 0 = \frac{1}{T}\frac{\mathrm{d}}{\mathrm{d}t}\left( \frac{\xi_k}{k\pi} \right) - \nu_k k\pi + 2h\cos(k\pi x_f)$$

$$\text{(B.41)}$$

where

$$
\begin{aligned}
h \;=\; & \beta \frac{1}{2}\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\left(\left(b_f+\frac{\tau}{T}\frac{\partial b_f}{\partial t}\right)\left(\frac{1}{3}+v_f\right)+\frac{\tau}{T}b_f\frac{\partial v_f}{\partial t}\ldots\right. \\
& \left.-\frac{\tau}{T}\frac{3}{2}b_f\frac{\partial v_f}{\partial t}\left(\frac{1}{3}+v_f\right)^2\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-1}\right)
\end{aligned}
$$

$$
b_f \;=\; -\sum_{j=1}^{N}\nu_j\sin(j\pi x_f)
$$

$$
v_f \;=\; \sum_{j=1}^{N}\left(\eta_j-\frac{\tau j\pi}{T}\left(\frac{\dot\eta_j}{j\pi}\right)\right)\cos(j\pi x_f)
$$

$$
\frac{\partial v_f}{\partial t} \;=\; \sum_{j=1}^{N}\left(j\pi T\left(\frac{\dot\eta_j}{j\pi}\right)-\frac{\tau j\pi}{T}\frac{\partial}{\partial t}\left(\frac{\dot\eta_j}{j\pi}\right)\right)\cos(j\pi x_f)
$$

$$
\frac{\partial b_f}{\partial t} \;=\; T\sum_{j=1}^{N}\left(\frac{\xi_j}{j\pi}\right)j\pi\sin(j\pi x_f)-T\sum_{j=1}^{N}\zeta\nu_j\sin(j\pi x_f)
$$

### B.1.8 Initialization of the adjoint variables at $t=t_1$

In the following equations the shorthand notation $\delta_f=\delta_D(x-x_f)$ will be used.

$$
a(t_1) \;=\; -\tau c(t_1)+T(2u(t_1)-2u_0)
$$

$$
\Rightarrow \sum_{j=1}^{N}\left(\frac{\xi_j}{j\pi}(t_1)\right)\cos(j\pi x) \;=\; -\beta b_f(t_1)\frac{\tau}{2}\left(\frac{1}{3}+v_f\right)\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\delta_f\ldots
$$

$$
+2T\left(\sum_{j=1}^{N}\eta_j(t_1)\cos(j\pi x)-\sum_{j=1}^{N}\eta_{j0}\cos(j\pi x)\right)
$$

$$
\Rightarrow \int_0^1\left(\sum_{j=1}^{N}\left(\frac{\xi_j}{j\pi}(t_1)\right)\cos(j\pi x)\right)\cos(k\pi x)dx = \ldots
$$

$$
\int_0^1\left(-\beta b_f(t_1)\frac{\tau}{2}\left(\frac{1}{3}+v_f\right)\left(\left(\frac{1}{3}+v_f\right)^2+\epsilon^2\right)^{-\frac{3}{4}}\delta_f\right)\cos(k\pi x)dx\ldots
$$

$$
+2T\int_0^1\left(\sum_{j=1}^{N}\eta_j(t_1)\cos(j\pi x)\right)\cos(k\pi x)dx\ldots
$$

$$
-2T\int_0^1\left(\sum_{j=1}^{N}\eta_{j0}\cos(j\pi x)\right)\cos(k\pi x)dx
$$

$$\Rightarrow \quad \frac{\xi_j}{j\pi}(t_1) = -\beta b_f(t_1)\tau\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\cos(j\pi x_f)$$
$$+2T\left(\eta_j(t_1) - \eta_{j0}\right)$$

$$b(t_1) = T(2p(t_1) - 2p_0)$$
$$\Rightarrow \quad -\sum_{j=1}^{N}\nu_j(t_1)\sin(j\pi x) = +2T\left(\sum_{j=1}^{N}\left(\frac{\dot{\eta}_{j0}(t_1)}{j\pi}\right)\sin(j\pi x)\ldots\right.$$
$$\left. -\sum_{j=1}^{N}\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right)\sin(j\pi x)\right)$$
$$\Rightarrow \quad \nu_j(t_1) = +2T\left(\frac{\dot{\eta}_j(t_1)}{j\pi} - \frac{\dot{\eta}_{j0}}{j\pi}\right)$$

## B.1.9  Calculating gradient information

The gradient information at $t = t_0$ is calculated with:

$$\frac{\partial \mathcal{L}}{\partial u_0} = \frac{1}{T}a(t_0) + \frac{\tau}{T}c(t_0) + 2u_0 - 2u(t_1)$$

$$\Rightarrow \quad \frac{\partial \mathcal{L}}{\partial \eta_{j0}} = \frac{1}{2}\left(\frac{1}{T}\left(\frac{\xi_j(t_0)}{j\pi}\right) + \frac{\tau}{T}\beta b_f(t_0)\left(\frac{1}{3} + v_f\right)\left(\left(\frac{1}{3} + v_f\right)^2 + \epsilon^2\right)^{-\frac{3}{4}}\cos(j\pi x_f)\ldots\right.$$
$$\left. +2\eta_{j0} - 2\eta_j(t_1)\right)$$

$$\frac{\partial \mathcal{L}}{\partial p_0} = \frac{1}{T}b(t_0) + 2p_0 - 2p(t_1)$$
$$-\frac{\partial \mathcal{L}}{\partial(\dot{\eta}_{j0}/j\pi)} = \frac{1}{2}\left(-\frac{1}{T}\nu_j(t_0) + 2\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right) - 2\left(\frac{\dot{\eta}_{j0}}{j\pi}\right)\right)$$
$$\frac{\partial \mathcal{L}}{\partial(\dot{\eta}_{j0}/j\pi)} = \frac{1}{2}\left(\frac{1}{T}\nu_j(t_0) - 2\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right) + 2\left(\frac{\dot{\eta}_{j0}}{j\pi}\right)\right)$$

(The minus sign on the LHS in the second line arises because $p$ is $-\sum(\dot{\eta}_j/j\pi)\ldots$) The gradient with respect to period $T$ is:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial T} &= \left\langle 2\left(u(t_1) - u_0\right) \frac{1}{T} \frac{\partial u(t_1)}{\partial t} + 2\left(p(t_1) - p_0\right) \frac{1}{T} \frac{\partial p(t_1)}{\partial t}\right\rangle \\
&= \frac{2}{T} \left\langle \left(\sum_{k=1}^{N} \eta_k(t_1)\cos(k\pi x) - \sum_{j=1}^{N} \eta_{j0}\cos(j\pi x)\right) \sum_{l=1}^{N} l\pi T \left(\frac{\dot{\eta}_l(t_1)}{l\pi}\right)\cos(l\pi x)\ldots \right. \\
&\quad \left. + \left(\sum_{k=1}^{N}\left(\frac{\dot{\eta}_k(t_1)}{k\pi}\right)\sin(k\pi x) - \sum_{j=1}^{N}\left(\frac{\dot{\eta}_{j0}}{j\pi}\right)\sin(j\pi x)\right) \sum_{l=1}^{N} \frac{\partial}{\partial t}\left(\frac{\dot{\eta}_l(t_1)}{l\pi}\right)\sin(l\pi x) \right\rangle \\
&= \frac{1}{T}\sum_{j=1}^{N}\left( (\eta_j(t_1) - \eta_{j0}) j\pi T \left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right) + \left(\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right) - \left(\frac{\dot{\eta}_{j0}}{j\pi}\right)\right)\frac{\partial}{\partial t}\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right)\right)
\end{aligned}
$$

### B.1.10  Discretising the cost function

The cost function is summed over space:

$$
\begin{aligned}
J &= \int_{x_0}^{x_1} \left( (u(t_1) - u_0)^2 + (p(t_1) - p_0)^2 \right) dx \\
&= \int_{x_0}^{x_1} \left( \left(\sum_{j=1}^{N} \eta_j \cos(j\pi x) - \sum_{k=1}^{N} \eta_{k0}\cos(k\pi x)\right)^2 \ldots \right. \\
&\quad \left. + \left(-\sum_{j=1}^{N}\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right)\sin(j\pi x) + \sum_{k=1}^{N}\left(\frac{\dot{\eta}_{k0}}{k\pi}\right)\sin(k\pi x)\right)^2 \right) dx \\
&= \frac{1}{2}\sum_{j=1}^{N}\left( \left(\eta_j(t_1)^2 + \eta_{j0}^2 - 2\eta_j(t_1)\eta_{j0}\right)\ldots \right. \\
&\quad \left. + \left(\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right)^2 + \left(\frac{\dot{\eta}_{j0}}{j\pi}\right)^2 - 2\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right)\left(\frac{\dot{\eta}_{j0}}{j\pi}\right)\right)\right) dx \\
&= \frac{1}{2}\left( (\eta_j(t_1) - \eta_{j0})^2 + \left(\left(\frac{\dot{\eta}_j(t_1)}{j\pi}\right) - \left(\frac{\dot{\eta}_{j0}}{j\pi}\right)\right)^2\right)
\end{aligned}
$$

This is just $\frac{1}{2}$ of the square of the 2-norm of the difference vector.