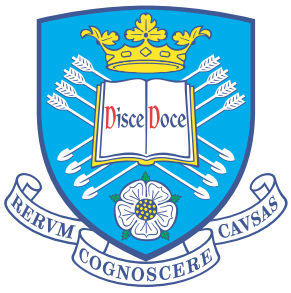
Low-order modelling for feedback control of fluid flows around complex geometries

Oliver James Dellar

Department of Automatic Control and Systems Engineering
The University of Sheffield



A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

February 2017

Dedicated to my parents,
for all their love and support.

Abstract

The majority of goods transportation vehicles' power is consumed in overcoming aerodynamic drag which arises due to vortex shedding over the bluff rear end of the vehicle. As such, a reduction in pressure drag via feedback control could have significant economic and environmental effects on CO_2 emissions, as well as reducing fatigue acting on the body.

The difficulty in designing such controllers lies in obtaining models suited to modern control design methods, which are necessarily of much lesser complexity than typical Computational Fluid Dynamics models, or models derived from immediate spatial discretisation of the equations governing fluid flows. It is with obtaining such low-order models that the work presented in this thesis is concerned.

A computationally efficient modelling approach which is suited to obtaining low-order models of complex geometry fluid flows is described, whereby the system's overall input-output frequency response is built up by connecting together the frequency responses of a large number of computational node subsystems in an efficient manner, exploiting the inherent structure of spatially discretised PDAEs.

In order to choose a suitable formulation of the governing equations – the Navier-Stokes equations – a rigorous analysis of several of the formulations suggested in the literature is presented, whereby the dynamics of different formulations are compared both at the nodal level, and at the full system level for a 2D channel flow (for which a well studied benchmark model exists).

In the penultimate chapter, the work of previous chapters is consolidated by applying the modelling technique to a 2D backward facing step flow. Slot jet actuation on the rear edge is assumed, and two separate output configurations are considered. The resulting models are compared to models obtained in a computational system identification study, which prompts an interesting investigation into the dynamics of the common PISO Computational Fluid Dynamics algorithm.

Acknowledgements

First and foremost, I would like to thank Dr Bryn Jones for supervising me during this project. I am indebted to him for the constant support he's provided throughout, always making himself available to mull ideas over, and the countless hours of useful discussion we've had as a result. I am confident that I would not have gained so much from this project without the input of Bryn.

Secondly, I am grateful to my fellow research students and research staff, from whom I learnt a lot during casual conversation about our work. In particular, I learnt a lot from the members of our small research group; hearing about the work of my colleagues, Paul Towers and Peter Heins, was invaluable in broadening my understanding of fluid dynamics. I also thank Andrew Hills, for helping me countless times with computer related issues when he should have been doing his own work. This project could not have been completed without Andrew's guidance towards all things HPC, and his help troubleshooting LaTeX whilst writing this thesis saved me much frustration.

Last, but certainly not least, I am delighted to have met such a good group of friends during the course of my PhD studies, both from within, and outside the department. Switching off from whatever niggling issue was on my mind after a day of little progress was always helped by socialising with friends, and for that I am grateful.

Contents

List of figures

List of tables

Nomenclature

General

∇	gradient operator
$\nabla\cdot$	divergence operator
∇^2	Laplacian operator
$\mathcal{D}$	differentiation matrix
$\mathfrak{L}$	discrete Laplacian matrix
$\vec{\cdot}$	vector of variable values on computational nodes after spatial discretisation
$\cdot \otimes \cdot$	Kronkecker product
i	$:= \sqrt{-1}$, imaginary unit
I	identity matrix
$\mathcal{I}$	identity operator
$\mathbb{C}$	set of complex numbers
$\mathbb{N}$	set of natural numbers
$\mathbb{R}$	set of real numbers
$\mathbb{R}^+$	set of strictly positive real numbers
$\mathbb{R}_0^+$	set of non-negative real numbers
$\Re(\cdot)$	real part
$\Im(\cdot)$	imaginary part
$\|\cdot\|_p$	p-norm of vector
t	time
f	frequency
ω	angular frequency

$\mathbf{x}$	$:= (\mathsf{x}, \mathsf{y}, \mathsf{z})$, Cartesian coordinates
Ω	spatial domain
$\partial\Omega$	spatial domain boundary
$\mathbf{n}$	unit normal vector
$\boldsymbol{\tau}$	unit tangential vector
$\hat{\cdot}$	Fourier transformed quantity
$\tilde{\cdot}$	Laplace transformed quantity
$\dot{g}(t)$	temporal derivative of $g(t)$
$\bar{\cdot}$	temporally averaged quantity
$\cdot'$	perturbation about mean quantity
$\cdot^\top$	matrix transpose
$\cdot^*$	matrix (or scalar) complex conjugate transpose
$\cdot^{-1}$	matrix inverse
$\cdot^\star$	dimensional quantity
$\mathcal{O}(\cdot)$	big-O notation
$\mathcal{M}$	computational memory
$\kappa_p(\cdot)$	condition number of a matrix
S_{n_p}	computational speedup of parallel computation
	on n_p processing cores
T_s	wall-clock time taken to complete job in serial
T_{n_p}	wall-clock time taken to complete job in parallel
	on n_p processing cores
ϱ	computational mesh density
δ_x, δ_y	computational mesh spacing in x and y directions
n_x, n_y	number of computational atoms in x and y directions
n_{Ω_x}, n_{Ω_y}	number of computational molecules in x and y directions
φ	wave-diffusion system surface height

Fluid dynamics specific

Re	$:= \mathsf{u}_\infty h/\nu$, Reynolds number

St	$:= fh/\mathsf{u}_\infty$, Strouhal number
ν	fluid kinematic viscosity
h	height of bluff body, step, or channel half height
D	cylinder diameter
ρ	fluid density
$\mathbf{u}$	$:= (\mathsf{u}, \mathsf{v}, \mathsf{w})$, velocity field
p	pressure field
u_0	channel centreline velocity
u_∞	free-stream velocity
u_j	fluidic actuator jet velocity
$\mathsf{u}_{j,\mathrm{rms}}$	fluidic actuator root-mean-square jet velocity
c_μ	fluidic actuator momentum coefficient
ℓ_s	fluidic actuator slot width
δ_BL	boundary layer thickness
α	streamwise wavenumber
f_u	lifting function
$\mathscr{E}$	kinetic energy density
$\mathscr{Q}$	energy matrix
$\mathscr{V}$	integration volume
$\mathscr{W}$	matrix of quadrature weights
ϑ	measure of difference between unforced and forced output signal energy
$.^\dagger$, $.^{\dagger\dagger}$	intermediate fields in PISO algorithm

Systems theory specific

A, B, C, D, E	descriptor state-space matrices
n	system state dimension
s	complex frequency
$G(s)$	system transfer function
$G(\mathrm{i}\omega)$	system frequency response

$\mathcal{G}$	vector of system frequency response values
e_{RMS}	root-mean-square-error between frequency response data and fitted transfer function
$\lvert\cdot\rvert$	magnitude
$\angle$	phase
$\mathscr{R}^{p\times q}$	set of real rational transfer function matrices
$\mathscr{H}_\infty$	Hardy space of linear transfer functions with finite infinity-norm
$\lVert\cdot\rVert_\infty$	$:= \sup_{\omega\in\mathbb{R}} \bar{\sigma}\left(G(\mathrm{i}\omega)\right)$, $\mathscr{H}_\infty$-norm
$\bar{\sigma}(\cdot)$	maximum singular value
$\eta(\cdot)$	number of open right-half-plane poles of a system
$\text{wno}\,(\cdot)$	winding number of a scalar transfer function evaluated on the standard Nyquist contour
$\mathbf{y}$, y	system output vector and scalar
$\mathbf{u}$, u	system input vector and scalar
$\mathbf{x}$	system state vector
$\boldsymbol{\xi}$	vector of surrounding subsystem state information
$\mathbf{z}$	vector of state information for surrounding subsystems
$\Lambda_\epsilon(\cdot)$	pseudospectra
$\mathcal{S}$	descriptor to standard state-space system transformation matrix
$\cdot\star\cdot$	Redheffer star product between two systems
$\mathcal{P}(\cdot)$	set of poles of a system
$\mathcal{Z}(\cdot)$	set of zeros of a system
τ	system time constant

Acronyms

2D	**two-d**imensional
3D	**three-d**imensional
ARMAX	**a**uto **r**egressive **m**oving **a**verage with e**x**ogenous inputs
BTCS	**b**ackward **t**ime **c**entred **s**pace
CFD	**c**omputational **f**luid **d**ynamics
DAE	**d**ifferential **a**lgebraic **e**quation
DMD	**d**ynamic **m**ode **d**ecomposition
DNS	**d**irect **n**umerical **s**imulation
ERA	**e**igensystem **r**ealisation **a**lgorithm
HPC	**h**igh **p**erformance **c**omputing
LES	**l**arge **e**ddy **s**imulation
LPV	**l**inear **p**arameter **v**arying
LQG	**l**inear **q**uadratic **G**aussian
LTI	**l**inear **t**ime-**i**nvariant
MBC	**m**odified **b**oundary **c**ondition
MIMO	**m**ulti-**i**nput **m**ulti-**o**utput
MPI	**m**essage **p**assing **i**nterface
ODE	**o**rdinary **d**ifferential **e**quation
PDAE	**p**artial **d**ifferential **a**lgebraic **e**quation
PIMPLE	**p**ressure-**i**mplicit **m**ethod for **p**ressure **l**inked **e**quations
PISO	**p**ressure-**i**mplicit with **s**plitting of **o**perator
PIV	**p**article **i**mage **v**elocimetry
PPE	**p**ressure **P**oisson **e**quation
POD	**p**roper **o**rthogonal **d**ecomposition

RANS	**R**eynolds **a**veraged **N**avier-**S**tokes
RBS	**r**andom **b**inary **s**equence
RMSE	**r**oot-**m**ean-**s**quare-**e**rror
RHSP	**R**ed**h**effer **s**tar **p**roduct
SGS	**s**ub-**g**rid **s**cale
ZNMF	**z**ero-**n**et-**m**ass-**f**lux

Chapter 1

Introduction

1.1 Motivation

The ability to control the flow of a fluid can be of great benefit in a number of situations; from those where turbulent flow is desirable – such as in combustion chambers, in order to increase the mixing of fuel and air – to those where steady, laminar flow is desired – such as that around the hull of a ship, in order to reduce skin-friction drag.

This work is motivated by the need to design feedback control systems to reduce the pressure drag acting on road vehicles, which within the UK accounted for 28% of total annual energy consumption in 2014 [2], and whose CO_2 emissions rose by 26% between 1990 and 2004 [1]. At motorway speeds, aerodynamic drag arising from vortex shedding over the bluff rear end of the vehicle is responsible for up to two thirds of this energy consumption [71, 51, 23, 7], compared to just 20–30% consumed overcoming rolling friction [71]. Figure 1.1 depicts this vortex shedding phenomenon.

The heavy goods transportation industry alone consumes approximately 1.2 billion barrels of oil per year [58], and as such successful drag reduction could result in fuel savings worth billion of dollars annually [35, 15], as well as having significant beneficial effects on CO_2 production.

1.1.1 Classification of bluff body flow control strategies

Different bluff body flow control strategies are divided here into three groups as in Choi *et al.* [22]:

1. Passive control – no additional energy is required to control the flow.
2. Active open-loop control – additional energy is supplied to some actuator(s) which

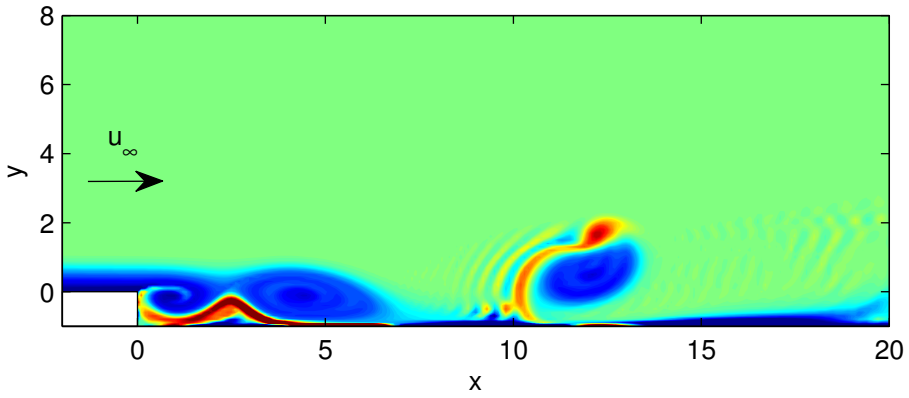


Figure 1.1: Vortex shedding phenomenon behind a bluff body: 2D backward facing step. Free-stream velocity is denoted u_∞.

operates according to some predefined control signal.

3. Active closed-loop (or feedback) control – energy is supplied to some actuator(s) which operates in response to some sensor measurement(s).

When considering bluff body flow control, the term 'passive control' typically refers to geometric modifications or structural additions to the body which affect the nature of the flow in a beneficial manner. An obvious example would be to streamline haulage trucks during their initial design, however geometric modifications are limited as this may reduce storage space significantly [74]. Another example is the addition of the so-called 'boat tail' device to the rear end of a truck [23]. These devices have been shown to reduce the drag acting on bluff body vehicles significantly by pushing the recirculation region downstream of the rear face, however away from the particular operating condition they were designed for, their performance may deteriorate [74]. The same can be said for most passive flow control devices.

Active open-loop bluff body flow control techniques comprise some sort of actuator which operates according to a signal which has been chosen *a priori*. During operation the actuator functions with no information about the system's current state. Actuators used most often in the literature include zero-net-mass-flux (ZNMF) jets (also referred to as synthetic jets), moving surfaces, and plasma actuators [18]. Control signal attributes, such as actuation amplitude and frequency, are often chosen after testing different combinations in experiments in an attempt to characterise how the flow responds to different actuation. This approach has been used successfully in many studies – as will become apparent in Chapter 2 – but it suffers from one major downfall: a lack of robustness. Similarly to the case with passive control devices, whilst an active open-loop controller may perform well in the environment in which it was designed, its performance may deteriorate as the operating conditions change.

Active closed-loop, or feedback control techniques comprise an actuator which operates according to a control signal which is computed in real-time in response to some sensor measurement. This could be, for example, a measurement of the pressure acting on part of the bluff body rear face, or a wall shear stress measurement. Feedback control has the important advantage of being robust to the effects of uncertainty [9], such as those which may arise from unmodelled external disturbances and changing operating conditions, as is prevalent in a practical bluff body flow scenario. In addition, feedback control is more efficient than open-loop control in that the control signals are typically much smaller, and thus the energy required for actuation is far less. As such, the focus of this work is on using feedback control for drag reduction of bluff body flows.

1.1.2 Difficulty in designing feedback controllers for fluid flows

The problem faced when designing feedback controllers for fluid flows is that much of modern control theory requires linear plant models that are of low-order[1]. After non-dimensionalising by appropriate length and velocity scales, $h \in \mathbb{R}^+$ and $\mathsf{u}_\infty \in \mathbb{R}^+$, respectively, the equations governing the flow of an incompressible[2], Newtonian fluid in $d \in \{2, 3\}$ spatial dimensions are given by:

$$\frac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} + \mathbf{u}(\mathbf{x},t)\cdot\nabla\mathbf{u}(\mathbf{x},t) = -\nabla \mathsf{p}(\mathbf{x},t) + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}(\mathbf{x},t),\ \forall\,(\mathbf{x},t) \in \Omega \times [0,t_f]\,, \quad (1.1a)$$

$$\nabla\cdot\mathbf{u}(\mathbf{x},t) = 0,\ \forall\,(\mathbf{x},t) \in \Omega \times [0,t_f]\,, \quad (1.1b)$$

with initial and boundary conditions:

$$\mathbf{u}(\mathbf{x},0) = \mathbf{u}_0(\mathbf{x}),\ \forall\,\mathbf{x} \in \Omega, \quad (1.2a)$$

$$\mathbf{u}(\mathbf{x},t) = \mathbf{u}_{\partial\Omega}(\mathbf{x},t),\ \forall\,(\mathbf{x},t) \in \partial\Omega \times [0,t_f]\,, \quad (1.2b)$$

where $\mathbf{u}(\cdot,\cdot) : \Omega \times [0,t_f] \to \mathbb{R}^d$ is the velocity field, $\mathsf{p}(\cdot,\cdot) : \Omega \times [0,t_f] \to \mathbb{R}$ is the pressure field, $\mathrm{Re} := \mathsf{u}_\infty h/\nu \in \mathbb{R}^+$ is the Reynolds number of the flow – a dimensionless number defined as the ratio of inertial forces to viscous forces [106], $\nu \in \mathbb{R}^+$ is the kinematic viscosity of the fluid, $\Omega \subset \mathbb{R}^d$ is the spatial domain with boundary $\partial\Omega$, $t_f \in \mathbb{R}^+$ is the endpoint of the time interval, $\mathbf{x} \in \Omega$ is a point in the domain, and ∇ is the del operator. Note that in this work $\mathbb{R}^+$ denotes the set of strictly positive real numbers, $\mathbb{R}^+ := \{z \in \mathbb{R} : z > 0\}$, whilst $\mathbb{R}_0^+$ denotes the set of non-negative real numbers, $\mathbb{R}_0^+ := \{z \in \mathbb{R} : z \geq 0\}$.

These equations, separately referred to as the *momentum equation* (1.1a) and the

[1] 'Low-order' in the sense that the state dimension is $\mathcal{O} \lesssim 10^2$.

[2] Flows with velocities up to approximately Mach 0.3 may be considered incompressible – that is, the fluid density ρ remains constant. Thus, these equations are suitable when considering the flow around road vehicles.

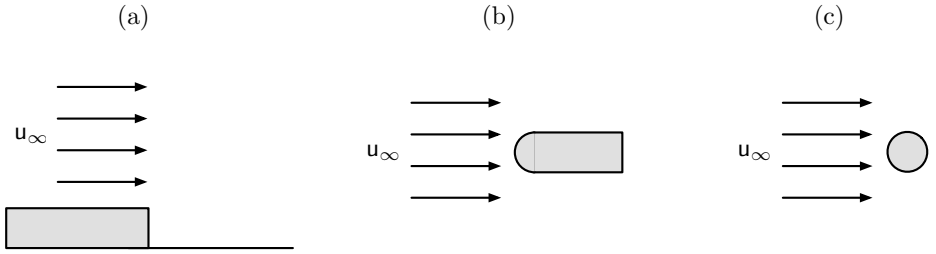


Figure 1.2: 2D complex geometries: (a) backward facing step; (b) D-shaped bluff body; (c) circular cylinder.

continuity equation (1.1b), constitute a set of non-linear partial differential algebraic equations (PDAEs) known as the incompressible Navier-Stokes equations. Due to the algebraic constraint (1.1b) and the infinite-dimensional nature of the equations, obtaining a model suitable for control design is non-trivial.

1.2 Overview

This thesis describes the work undertaken in an effort to develop computationally efficient modelling techniques for obtaining low-order control models of complex geometry fluid flows, such that ultimately feedback controllers can be designed. 'Complex geometries' are defined here as backward facing steps, D-shaped bluff bodies, and cylinder flows, as depicted in Figure 1.2, as opposed to more simple geometries such as plane channel flow.

When designing a control system, if the objective is to suppress perturbations around a mean flow then designing feedback controllers based on linearised approximations gives rise to acceptable closed-loop performance. This has been shown in a number of studies on the control of both complex geometry and plane channel flows (see, e.g. [14, 59, 90, 65, 28, 67, 56, 46]). However, efficiently obtaining low-order models that are accurate in the sense that they retain the most important dynamics of a given flow remains an open question [56]. A typical approach relies upon the spatial discretisation of the linearised Navier-Stokes equations on a computational mesh in order to obtain finite-dimensional state-space models (see, e.g. [3, 11, 12, 46]).

The construction of state-space models of a fluid flow by directly spatially discretising the linearised Navier-Stokes equations is typically restricted to simple geometries such as plane channel flow, whereby the assumption of periodicity in the streamwise (and spanwise for 3D geometries) direction allows the use of Fourier transforms in space, decoupling the flow by spatial wavenumber [14]. For more complex geometries, however, such assumptions are less readily applicable, and hence computational fluid dynamics

(CFD) codes used for flow simulations often use methods based on local interpolants, such as finite-difference, finite-volume, or finite-element discretisation. Whilst these can be used for constructing the state-space matrices describing complex geometry flows, it is difficult in practice, and typically results in systems with extremely large ($\mathcal{O} > 10^6$) state dimension, giving rise to impractically large system matrices. The vast majority of these states, however, have little influence on the actuator-to-sensor response of the system, which means that satisfactory models for feedback control design can have state dimension many orders of magnitude smaller. In the context of bluff body drag reduction, Dahan *et al.* [28] showed that the underlying input-output frequency response of some flows actually resemble those of *second*-order systems.

1.2.1 Thesis layout

The work presented in the remainder of this thesis describes an efficient, computationally tractable approach to obtaining the input-output frequency response of complex geometry fluid flows, such that the low-order nature of the underlying dynamics is exposed and a transfer function can be fitted for control design purposes. It is structured as follows:

Chapter 2 discusses some of the literature relevant to the current study, considering previous works on control of fluid flows around complex geometries.

Chapter 3 discusses in more detail the 2D backward facing step flow which will serve as a case study for which a low-order model will eventually be obtained using the method outlined in Chapter 4. The characteristics and features of a typical backward facing step flow will be presented, before discussing the numerical procedure used to simulate the flow. A linearity check will be performed in order to justify the use of linear systems theory.

Chapter 4 explains the low-order modelling approach used in this work, along with a domain decomposition optimisation technique which significantly reduces its computational complexity. Its efficacy is proven by applying it to the 2D wave-diffusion equation example problem, comparing results to a high-order benchmark model.

Chapter 5 discusses some of the difficulties which arise when spatially discretising the linearised Navier-Stokes equations. The problem of choosing the 'correct' discretisation of the governing equations is addressed by considering both the dynamics of a single computational node, and a full order model of 2D channel flow. This reveals the most suitable discretisation for feedback control design.

In Chapter 6 the frequency response of the 2D backward facing step with ZNMF slot jet actuation and two different output configurations is computed using the mod-

elling approach and insight gained in Chapters 4 and 5. With this frequency response, low-order transfer functions are fitted, and compared to models obtained from a computational fluid dynamics-based system identification study. This reveals significant discrepancy between the two approaches, which is explained by investigating the underlying dynamics of the numerical solver algorithm employed at the computational node level and comparing these to the dynamics of a direct spatial discretisation of the linearised Navier-Stokes equations.

Finally, conclusions and future work are presented in Chapter 7.

1.3 Publications and presentations

Some of the work presented in this thesis is based on the following publications and conference presentations:

- **O. J. Dellar** and B. Ll. Jones. 'Low-order modelling for feedback control of fluid flows around complex geometries', oral presentation at *UKACC PhD Presentation Showcase, London, UK, 23rd October 2014.*
- **O. J. Dellar** and B. Ll. Jones. 'Low-order Modelling for the Feedback Control of Bluff Body Fluid Flows', in proceedings of *USES 2015 - The University of Sheffield Engineering Symposium, Sheffield, UK, 24th June 2015.*
- **O. J. Dellar** and B. Ll. Jones. 'Low-order modelling for feedback control of fluid flows around complex geometries', oral presentation at *APS 68th Annual Division of Fluid Dynamics Meeting, Boston, USA, 22nd-24th November 2015.*
- **O. J. Dellar** and B. Ll. Jones. 'What is the 'correct' discretisation of the linearised Navier-Stokes equations for feedback flow control?', oral presentation at *UKACC PhD Presentation Showcase, London, UK, 17th May 2016.*
- **O. J. Dellar** and B. Ll. Jones. 'Discretising the linearised Navier-Stokes equations: A systems theory approach', in proceedings of *11th UKACC International Conference on Control, Belfast, UK, 30th August-2nd September 2016.*
- **O. J. Dellar** and B. Ll. Jones. 'What is the 'correct' formulation of the linearised Navier-Stokes equations for designing feedback flow control systems?', oral presentation at *APS 69th Annual Division of Fluid Dynamics Meeting, Portland, USA, 20th-22nd November 2016.*
- **O. J. Dellar** and B. Ll. Jones. 'Dynamically correct formulations of the linearised Navier-Stokes equations'. *International Journal for Numerical Methods in Fluids*, 2017.

- **O. J. Dellar** and B. Ll. Jones. 'Low-order modelling for control of spatially distributed systems', under review for *International Journal of Control*.

- **O. J. Dellar** and B. Ll. Jones. 'An investigation into the discrepancy in the dynamics of the PISO algorithm', in preparation.

Chapter 2

Relevant literature

This chapter summarises some of the key flow control techniques and ideas explored in the literature which address the issue of control of fluids around different complex geometries.

2.1 Bluff body and cylinder flow

Bluff body and cylinder flows arise in a range of common situations, and can result in both drag and structural fatigue. The mechanism by which these two detrimental effects occur is the same for both bluff bodies and cylinder flows, and is known as the von Kármán vortex street. The von Kármán vortex street refers to the periodic shedding of vortical structures behind the body. Behind a D-shaped blunt bluff body, two shear layers separate from the body at fixed locations – the upper and lower edges – and roll up to form symmetrical vortical structures with vorticity in opposing directions [27], whilst behind a circular cylinder the boundary layer peels away from the cylinder's surface at a location dependant on the Reynolds number (known as boundary layer separation), before forming similar symmetrical vortical structures [70]. As such, the size of the wake behind a circular cylinder flow is directly influenced by the flow separation location, and can be minimised by ensuring the flow remains attached for as long as possible. For Reynolds numbers greater than a critical value an absolute instability in the wake, which is caused by a Hopf bifurcation in a global mode occuring at this critical Reynolds number, causes the symmetry to break down and vortices to be shed from the upper and lower edges/separation points alternately [52] as depicted in Figure 2.1. This phenomenon results in a (relatively) low pressure region behind the body which is responsible for form drag, and a lift coefficient which alternates between positive and negative values as the vortices are shed, which can result in structural fatigue. The collapse of the Tacoma Narrows Bridge in 1940, for example, was a direct

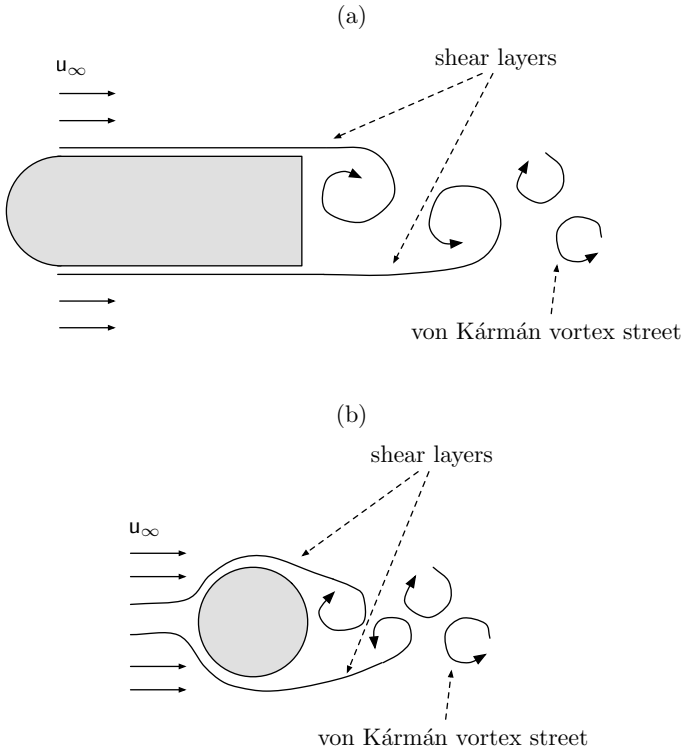


Figure 2.1: Vortex shedding behind: (a) a 2D D-shaped bluff body; (b) a 2D circular cylinder.

consequence of this alternating lift coefficient oscillating at a frequency similar to the resonant frequency of the structure, due to vortex shedding of the wind [99].

For these reasons the control of the wake behind bluff body and cylinder flows has been of interest for a long time.

2.1.1 Bluff body flow

Numerous studies have been carried out in attempts to reduce drag on bluff bodies using open-loop actuation. Parkin *et al.* [74] studied numerically the flow over a 2D D-shaped bluff body at $\mathrm{Re} = 23 \times 10^3$ (where Re is now defined as the Reynolds number based on the body height h and free-stream velocity u_∞). They implemented ZNMF jets at the upper and lower trailing edges, oriented upwards (upper edge), downwards (lower edge), and rearwards (both edges). Periodic forcing was employed at a range of actuation Strouhal numbers (the actuation Strouhal number, $\mathrm{St}_{\mathrm{act}} := f_{\mathrm{act}} h / \mathsf{u}_\infty$, is a dimensionless frequency, based on actuation frequency f_{act} and inflow velocity u_∞), and it was found that drag was reduced most at a $\mathrm{St}_{\mathrm{act}}$ approximately equal to half the natural vortex shedding Strouhal number $\mathrm{St} := fh/\mathsf{u}_\infty$, where f is the shedding frequency.

Parkin *et al.* then carried out a wake analysis using dynamic mode decomposition (DMD) to gain further insight into the physics of the vortex shedding. They found that two modes coexist in all the flow simulations, the superposition of which is able to represent the main flow features, and that by adding energy to these two modes, the dominant of which corresponds to symmetric vortex shedding, they were able to extend the mean recirculation region which in turn reduced drag. The optimal $\mathrm{St}_{\mathrm{act}}$ was found which extended this region the most, and hence increased the base pressure (pressure acting on the rear face) the most, corresponding to drag reduction. They achieved a drag reduction of 20%.

Kim *et al.* [60] considered a 3D D-shaped bluff body in both experiments and numerical simulations. Large eddy simulations (LES) were carried out at $\mathrm{Re} = 4200$, and wind-tunnel experiments were carried out at $\mathrm{Re} = 20 \times 10^3$ and 40×10^3. Unlike Parkin *et al.*, Kim *et al.* implemented steady (time-invariant) slot jets oriented at 45° angles to the horizontal at both the upper and lower edges, but varied the forcing's spatial distribution along the spanwise direction sinusoidally. Two cases were considered: the case where the spatial distribution of forcing on the top and bottom edges was in phase, and the case where it was out of phase. In the LES simulations a base pressure increase of approximately 30% was achieved with in-phase forcing, whilst out of phase forcing had little effect on the flow. Similarly, with in-phase forcing, base pressure increases of around 36% and 18% were achieved in the experimental study for $\mathrm{Re} = 20 \times 10^3$ and 40×10^3, respectively.

Chaligné *et al.* [20] experimentally studied the flow over a 2D D-shaped bluff body in close ground proximity at $\mathrm{Re} \approx 176 \times 10^3$. Locating the body close to the ground better represents the flow physics around a road vehicle. A pulsed jet actuator was implemented at the upper trailing edge acting at 45° to the horizontal, and a study was carried out into what effect varying $\mathrm{St}_{\mathrm{act}}$ had on the base pressure. It was found that the periodic blowing had no beneficial effect on the base pressure, and that at some frequencies it was actually detrimental.

A small overhanging flap was attached at the top of the rear face of the body and this yielded much improved results. Above a certain $\mathrm{St}_{\mathrm{act}}$, significant increase in base pressure was achieved which continued to increase as $\mathrm{St}_{\mathrm{act}}$ was increased, before eventually approaching an asymptotic value.

Pastoor *et al.* [75] studied the turbulent flow around a 3D D-shaped bluff body both theoretically and experimentally, for Re of between 23×10^3 and 70×10^3, with ZNMF actuators at the upper and lower trailing edges oriented at 45° to the horizontal. An analytical model was proposed relating the vorticity of a summation of wake vortices to the velocity field induced due to this vorticity, based on potential theory and Biot-Savart's Law. This model, which also resolves the effects of ZNMF actuation, provided

insight into feedback controller design.

Wind tunnel experiments were carried out with periodic forcing to determine the effects of varying $\mathrm{St}_{\mathrm{act}}$ and the forcing amplitude, and optimal parameters which yielded drag reduction of almost 15% were determined.

A slope-seeking adaptive feedback controller – an extension of extremum-seeking control – was designed which used pressure sensor measurements located on the rear face of the body, with the objective of optimally using the actuators to increase the mean base pressure. This type of control assumes the system can be described by a state-space model representing the input-output dynamics, and a steady-state input-output map. The goal of the controller was to drive the system to a state with a particular slope in the steady-state map. One advantage of this method is that neither the state-space model nor the steady-state map must be known *a priori*. The controller was implemented in an experiment where Re was continuously increased from $\mathrm{Re} = 40.5 \times 10^3$ to 70×10^3, and successfully maintained drag reduction of 15%, demonstrating the robustness of the controller to changes in Re.

Pastoor *et al.* then implemented phase control with the objective of synchronising the upper and lower shear layers, and hence vortex shedding, such that the vortex shedding remained symmetric, therefore extending the recirculation region. This control employed an extended Kalman filter to estimate the phase difference between actuation and vortex shedding based on rear face pressure measurements, and with this determined the actuation signal required to synchronise the upper and lower shear layers. In an experiment at $\mathrm{Re} = 46 \times 10^3$, the phase control also achieved drag reduction of 15%, but only required 56% of the actuation energy required by the slope-seeking controller.

Henning and King [47] considered the flow around a 3D D-shaped bluff body in wind tunnel experiments at $\mathrm{Re} = 40 \times 10^3$, with ZNMF actuators again located at the trailing edges oriented at 45° to the horizontal. Similarly to Pastoor *et al.* [75] they characterised the flow by measuring the mean base pressure coefficient for a range of $\mathrm{St}_{\mathrm{act}}$. They found that with the optimal actuation frequency, drag reduction of 10% could be achieved.

A robust feedback controller was then designed based on an identified linear black-box model which used pressure sensors located on the rear face of the step as the system output, and the momentum coefficient, $c_\mu := \ell_{\mathrm{s}} \mathsf{u}_{j,\mathrm{rms}}^2 / h \mathsf{u}_\infty^2$ (non-dimensional actuation amplitude, where ℓ_{s} is the actuator slot width, and $\mathsf{u}_{j,\mathrm{rms}}$ is the rms jet velocity), as the control input, forcing at constant $\mathrm{St}_{\mathrm{act}}$ (set at the value found to be optimal in the open-loop forcing experiments). In order to identify the linear model, step input experiments were carried out for a range of c_μ. Using this data a family of linear time-

invariant (LTI) transfer function models were deduced, governed by uncertain model parameters. A controller was then designed using loop-shaping techniques in order to satisfy gain and phase margin requirements, and sensitivity specifications. The controller was implemented in wind tunnel experiments and proved to not only just increase the base pressure, but also track a reference signal very well. Robustness of the controller was demonstrated by varying Re between 20×10^3 and 60×10^3, and observing that the controller was still able to track a reference signal.

Dahan [27] studied the flow around a 3D D-shaped bluff body numerically, at Re = 10×10^3. ZNMF actuators were implemented at the upper and lower trailing edges oriented at 45° to the horizontal. Dahan used a system identification approach to modelling the input-output behaviour of the flow (where the output was the pressure coefficient on the body's rear face), forcing the flow periodically at a range of $\mathrm{St}_{\mathrm{act}}$ and plotting both the gain and phase (with respect to actuation) of the system output on Bode plots. This was repeated for a range of different forcing amplitudes in order to determine the significance of the nonlinearity. It was found that the differences in gain and phase recorded for different forcing amplitudes were minimal, justifying Dahan's linear system assumption. After averaging the gain and phase data over the different forcing amplitudes, an 8^{th}-order transfer function model was fitted to the identified frequency response. A $\mathscr{H}_\infty$ loop-shaping robust controller was designed, which when implemented in simulation successfully increased the mean base pressure by 15%, corresponding to drag reduction.

Flinois and Morgans [38] considered numerically the 2D flow around a D-shaped bluff body at Re = 80, with actuation taking the form of horizontal body forcing located just above and below the top and bottom trailing edges, respectively. Two sensor configurations were considered: a vertical velocity measurement at a point in the wake of the body, and a rear face mounted distributed sensor measuring the antisymmetric component of the force acting on the base. The eigensystem realisation algorithm (ERA) was used to obtain linear reduced order models of the flow from open-loop linearised impulse response data, and the authors showed that similar models can be obtained from the nonlinear impulse response. $\mathscr{H}_\infty$ loop-shaping feedback controllers were designed which successfully suppressed vortex shedding in the wake, corresponding to drag reduction.

2.1.2 Cylinder flow

Similarly to bluff body geometries, controlling the wake flow behind a cylinder has been the focus of many studies. Weller *et al.* [104] considered numerically the 2D laminar flow past a square cylinder, at Re = 150. They implemented out of phase suction and blowing actuation located at the upper and lower trailing edges, oriented upwards and

downwards, respectively. Assuming transverse velocity measurements in the cylinder wake, a proportional gain feedback control law was implemented.

A database of flow-field snapshots was recorded for a range of different control signals, with which a set of proper orthogonal decomposition (POD) modes was deduced. The linearised Navier-Stokes equations were projected onto a low-order subspace of modes, yielding a reduced order state-space model of the input-output behaviour. The objective of the controller was to minimise the difference between the steady, unstable flow-field solution and the actuated flow, such that vortex shedding was suppressed and drag was reduced. This was done using an optimisation algorithm to determine the optimal feedback gains for the control law. The controller successfully stabilised the flow when implemented in a CFD simulation, and yielded significant drag reduction.

Akhtar *et al.* [6] studied numerically the 2D flow past a circular cylinder at $\mathrm{Re} = 200$ (where Re is defined here as the Reynolds number based on the cylinder diameter D and free-stream velocity u_∞). Flow-field snapshots were obtained and, similarly to Weller *et al.* [104], POD modes were computed. This was done both without control, and with control that assumed a pair of suction actuators located on the cylinder surface, 75° clockwise and counter-clockwise from the rear. It was found that the first 12 POD modes contained 99% of the system energy, and so were considered sufficient for modelling the system's behaviour.

The Navier-Stokes equations, linearised around the time averaged flow-field, were projected onto these POD modes, and an optimal linear quadratic Gaussian (LQG) controller was designed. This assumed that only the temporal coefficient of the dominant POD mode could be measured. When implemented in a full Navier-Stokes CFD simulation, a reduction in vortex shedding was achieved, resulting in drag reduction.

Mathelin *et al.* [66] considered numerically the 2D flow around a circular cylinder at $\mathrm{Re} \approx 200$. The study assumed the ability to prescribe a wall-normal velocity on the cylinder boundary. CFD simulations with the implementation of a range of different open-loop control signals were carried out, and the data produced was used to determine POD modes. Further simulations were carried out in order to determine the 'optimal trajectory' of the system model in phase space; that is, the phase trajectory which resulted in the minimising of a cost function comprising a sum of the cylinder drag force and the control effort.

Assuming full state feedback, a linear parameter varying (LPV) robust controller was designed based on the reduced order model after it was linearised around a set of points in the optimal phase space trajectory. The controller was shown to be robust to perturbations in Re by implementation in a Navier-Stokes CFD simulation where Re was modelled as a Gaussian stochastic process with a mean value of 200 and a variance

of 3.8. The authors claim that good attenuation of drag was achieved.

Milano *et al.* [69] studied numerically the 2D flow around a circular cylinder at Re = 500 and 1000, and used evolutionary computing approaches to optimising active control parameters for drag reduction. They assumed that the surface of the cylinder was split into 16 equal sized sections whose tangential velocity could be prescribed, simulating a set of 16 actuator belts.

Firstly, for the case where Re = 500, an offline genetic algorithm was used to optimise the choice of actuator belt velocities given a cost function comprising the total drag experienced by the cylinder, and when the control law was implemented in a full Navier-Stokes CFD simulation it quickly suppressed vortex shedding, which resulted in a reduction in drag of approximately 60%.

Next, for the case where Re = 1000, it was assumed that all 16 actuator belts moved with the same velocity; that is, the cylinder could only rotate as a whole. An online optimisation was implemented to deduce the optimal actuation parameters – oscillation amplitude and frequency – and it was found that the algorithm quickly discovered parameters which resulted in the suppression of vortex shedding, and again achieved a decrease in drag of around 60%.

Milovanovic and Aamo [70] studied numerically the 2D flow around a circular cylinder at Re = 60, and used Ginzburg-Landau theory to develop an observer based controller. It was assumed that the mechanism of actuation was cylinder rotation, and that shear measurements were available from the rearmost point of the cylinder. Using data recorded from CFD simulations, parameters for the Ginzburg-Landau equation were determined such that it accurately described the von Kármán vortex shedding phenomenon. This equation was used to deduce an observer to estimate flow velocities downstream of the cylinder, and a full-state feedback control law. When implemented in a Navier-Stokes CFD simulation, the observer estimated states with good accuracy, and the controller fully suppressed vortex shedding, resulting in a decrease in drag.

Illingworth [53] considered numerically the 2D circular cylinder flow. Two blowing and suction actuators were located at $\pm 70^\circ$ from the cylinder's downstream pointing horizontal, and vertical velocity component sensors were placed in the cylinder wake. Using the ERA, low-dimensional linear models were obtained from impulse response simulations for a number of different Reynolds numbers between 45 and 110, and used to design $\mathscr{H}_\infty$ loop-shaping controllers. These controllers were shown to perform well in simulation, both for the case where only a single wake velocity measurement was available, and when multiple measurements were available. The controllers also proved to be robust across a range of Reynolds numbers, but it was highlighted that the maximum stability margin achievable during the $\mathscr{H}_\infty$ design process diminished with

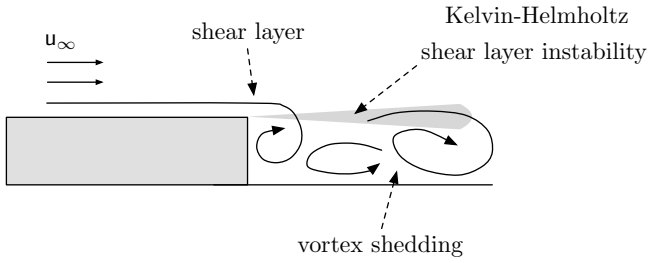


Figure 2.2: Vortex shedding behind 2D backward facing step.

increasing Reynolds number.

2.2 Backward facing step flow

Neglecting the effects of flow underneath a vehicle, backward facing step flows serve as a rudimentary model of the flow behind a bluff body vehicle, such as a lorry. They have been the subject of many past studies, and act as the test case for which a low-order model will be obtained in this work.

When the flow detaches at the edge of a backward facing step, a Kelvin-Helmholtz type instability in the shear layer causes the formation of vortical structures to shed from the body [40] as depicted in Figure 2.2. These vortices result in a low-pressure region in what is known as the 'recirculation zone' aft of the step, and it is this low-pressure region which is responsible for pressure drag. The objective of flushing the recirculation zone further downstream in order to raise the average base pressure, and therefore reduce drag, has been the focus of a number of studies.

Dahan *et al.* [28] studied numerically both 2D and 3D backward facing step flows at Reynolds numbers of $\mathrm{Re} = 2000$ and 2×10^4, respectively, corresponding to laminar and turbulent flow (where Re is defined now as the Reynolds number based on the step height h and free-stream velocity u_∞). Two actuator placements were considered for each case; the first implemented ZNMF actuation immediately before the step trailing edge oriented at 45° to the horizontal, and the second implemented ZNMF actuation at the very bottom of the step rear face oriented rearwards. In a similar fashion to the D-shaped bluff body work, system identification methods were used for both actuator placements in both the 2D and 3D cases, in order to model the input-output behaviour of the flow (where again the output was the base pressure coefficient), forcing the flow periodically at a range of $\mathrm{St}_{\mathrm{act}}$, and plotting the gain and phase of the output on Bode plots. It was again found that the differences in gain and phase observed when actuating with different forcing amplitudes were sufficiently small for the linear systems assumption to hold. For the 3D case, in addition to this harmonic forcing

system identification, the ERA was used with a sum-of-sines excitation signal. This yielded frequency responses that were in excellent agreement.

For the 2D, laminar case, second-order linear feedback controllers were implemented, with resonant frequency equal to that of the dominant vortex shedding frequency, and damping ratios chosen such that the actuation signal did not deviate much from the linear range of forcing amplitudes, and that the sensitivity function was shaped in a desirable fashion. When implemented in a Navier-Stokes CFD simulation both controllers completely suppressed vortex shedding and achieved a significant base pressure increase of approximately 70%, resulting in a large drag reduction.

For the 3D, turbulent case, $\mathscr{H}_\infty$ loop-shaping controllers were designed in order to shape the loop and sensitivity transfer functions. When implemented in a CFD simulation it was found that the controller with actuator located on the step edge outperformed that with the actuator located at the bottom of the rear face, achieving a base pressure increase of approximately 20%, whilst the controller with the actuator located at the bottom of the rear face only achieved an increase of around 10%.

Hervé *et al.* [48] considered numerically a 2D backward facing step flow. They suggested that a data driven system identification approach towards the modelling of the flow avoids some of the disadvantages of model-based approaches, namely that in model-based control design *a priori* assumptions on system noise must be made, whilst a data driven model directly incorporates noise influences.

A flow scenario was studied in which skin-friction measurements were available at a location immediately before the step edge, and on the bottom wall further downstream around the end of the recirculation zone. A single ZNMF actuator was located on the step edge oriented upwards. External forcing (representing noise) was applied upstream of the step, taking a Gaussian shape. For flow with $\mathrm{Re} = 500$, an auto-regressive moving-average with exogenous inputs (ARMAX) model was employed, the construction of which involved running simulations with broadband open-loop forcing being applied by the actuator in the presence of the external forcing noise term, and fitting suitable ARMAX coefficients from the data. A feedforward controller was designed using the ARMAX identified model. To demonstrate the robustness and performance of the controller, it was implemented in a CFD simulation with high noise levels applied, and achieved a reduction of almost 50% of the total turbulent kinetic energy of the flow.

Pastoor *et al.* [76] studied numerically a 2D backward facing step flow at $\mathrm{Re} = 4000$, with a ZNMF actuator located on the step edge oriented at 45° to the horizontal. They deduced vortex models of the flow, with the simplest models comprising just one vortex, and more complex models comprising up to one hundred vortices. The main features

of the backward facing step flow were described well by the vortex models, with a close link between the number of vortices present in the model and the accuracy of the predictions.

A higher order 'vortex blob' model comprising several hundred vortices was deduced, and this proved to describe the vortex shedding phenomenon and recirculation region dynamics well for both the unforced and the open-loop harmonically forced cases.

Gautier and Aider [40] studied experimentally the 3D backward facing step flow at Re = 2070 and 2900, subjected to open-loop harmonic forcing. Unlike most previous studies which located fluid actuators on or near the step edge, Gautier and Aider implemented a pulsed jet upstream of the step edge oriented vertically upwards, with a square wave control signal. They studied the effects of varying the jet amplitude, frequency, and duty cycle on the recirculation zone area aft of the step. For a range of actuation parameters they were able to significantly reduce the area of recirculation.

Gautier and Aider highlight that whilst recirculation region length might be a reasonable criterion to consider in some situations, in others a different criterion, such as the recirculation zone area as was used here, might be more important.

D'Adamo *et al.* [26] considered experimentally a 3D backward facing step flow at Re = 1520 and studied the effect of implementing a dielectric barrier discharge electrohydrodynamic actuator on the body rear face, oriented upwards. The mean reattachment length of the flow was monitored using particle image velocimetry (PIV) in response to varying the actuator frequency, duty cycle, and amplitude, such that optimal actuation parameters could be chosen to minimise this reattachment length. It was observed that maximum reattachment length reduction occurs for a forcing frequency very similar to that of the natural vortex shedding frequency. Actuator amplitude and duty cycle, however, have less effect on reattachment length.

Gautier *et al.* [41] presented a novel genetic programming based approach to controlling the recirculation region of a backward facing step flow in experments at Re = 1350, with the particular objective of minimising the recirculation zone. 2D velocity snapshots were obtained in real-time using PIV and a graphics card, and slotted jets were implemented further upstream of the step edge which functioned in response to a feedback control law. Genetic programming was used in order to optimise the control law with respect to a cost function based on the recirculation area and an actuator penalisation term. They found that after 12 generations, each consisting of 500 individuals, the algorithm had converged to a control law which was able to reduce the recirculation zone by 80%. It was found that this genetic programming based feedback control approach performed similarly to periodic forcing when operating at the design conditions, but that it outperformed periodic forcing when operating conditions were changed by

way of altering the Reynolds number.

2.3 Summary

Much of the work available in the literature concentrates on open-loop control [60, 20, 74, 40, 26] which, whilst has been shown in some studies to perform well for fixed operating conditions, suffers from a lack of robustness, as mentioned in Section 1.1.1.

Approaches using model-based feedback control have also been used, often based on models identified through data driven approaches. These include system identification [47, 28, 38, 53], and model reduction methods such as POD, balanced truncation, and balanced POD [50, 86, 105, 8].

System identification can yield low-order models suitable for controller design, but due to the extremely high state dimension of a typical CFD solver spatial discretisation ($\mathcal{O} > 10^6$), numerical simulation can be very computationally expensive. Forced simulations must be run for a long time in order to resolve the low frequency dynamics, typically on high performance computing (HPC) clusters.

Model reduction, such as projection based methods which project the Navier-Stokes equations onto a low-dimensional subset of modes, may yield models which are close to the high-dimensional system in an open-loop sense, however, may not capture some of the dynamics which arise in a closed-loop setting. This is due to the fact that these methods were developed as model reduction techniques for open-loop simulation models, as opposed to closed-loop control design models [56, 13], and as noted in [59], a model which is sufficiently accurate for control design is not necessarily good enough for numerical simulation. Similarly, a model which is good for numerical simulation is not necessarily appropriate for control design [9]. As such, projection based methods do not provide guarantees that the models produced will be suitable for feedback control design purposes [25]. In addition to this, obtaining the system modes can be difficult in itself due to the size and ill-conditioning of the system matrices involved.

The books by Gad-el-Hak [35] and Aamo and Krstić [3] provide further background on flow control theory in general, and the review papers by Choi *et al.* [22] and Rashidi *et al.* [82] provide more specific summaries of the different vortex shedding suppression techniques which have been studied in the literature.

Chapter 3

Backward facing step flow

This chapter presents a more detailed discussion of the complex geometry flow which will serve as a test case for application of the modelling approach discussed in the remainder of this thesis: the 2D backward facing step flow.

The main features of a backward facing step flow are first discussed, before details of the numerical procedure used for simulation in this work are given. The implementation of actuation is explained, and the use of linear systems theory for modelling the flow is justified.

Note that for the remainder of this thesis since the spatial domain is 2D:

$$\Omega \subset \mathbb{R}^2, \quad \mathbf{x} := (\mathsf{x}, \mathsf{y}) \in \Omega, \quad \mathbf{u} := (\mathsf{u}, \mathsf{v}) \in \mathbb{R}^2, \tag{3.1}$$

where quantites in the governing equations (1.1) have been non-dimensionalised by the step height h and free-stream velocity u_∞, i.e.:

$$t = \frac{t^\star \mathsf{u}_\infty}{h}, \quad \mathbf{x} = \frac{\mathbf{x}^\star}{h}, \quad \mathbf{u} = \frac{\mathbf{u}^\star}{\mathsf{u}_\infty}, \quad \mathsf{p} = \frac{\mathsf{p}^\star}{\rho \mathsf{u}_\infty^2}, \quad \mathrm{Re} = \frac{\mathsf{u}_\infty h}{\nu}, \tag{3.2}$$

where $\rho \in \mathbb{R}^+$ is the fluid density and the superscript * denotes a dimensional quantity.

3.1 Flow features

The main features present in a backward facing step flow are depicted in Figure 3.1. The boundary layer of the flow at the point of separation is denoted δ_{BL}, and is equal to the height above the boundary at which $\mathsf{u} = 0.99\mathsf{u}_\infty$. Separation occurs at a fixed location – the rear edge of the step – after which a Kelvin-Helmholtz type instability develops in the shear layer due to the high velocity gradients. This Kelvin-Helmholtz

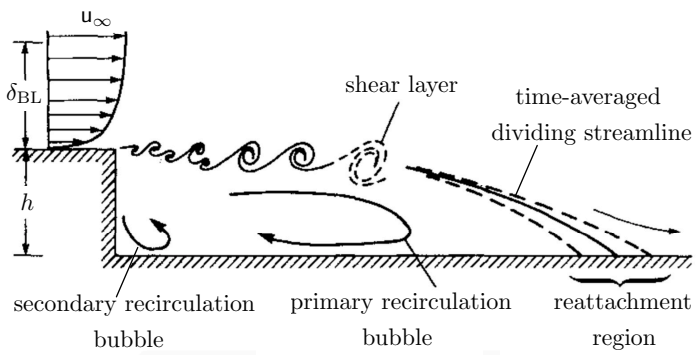


Figure 3.1: Main features of a backward facing step flow, adapted from Driver *et al.* [34].

boundary	velocity	pressure
$\partial\Omega_{\text{in}}$	$(\mathsf{u},\mathsf{v}) = (\mathsf{u}_\infty, 0)$	$\partial\mathsf{p}/\partial\mathbf{n} = 0$
$\partial\Omega_{\text{out}}$	$\partial\mathsf{u}/\partial\mathbf{n} = 0$	$\mathsf{p} = 0$
$\partial\Omega_{\text{bottom}}$	$(\mathsf{u},\mathsf{v}) = (0,0)$ (no-slip)	$\partial\mathsf{p}/\partial\mathbf{n} = 0$
$\partial\Omega_{\text{top}}$	$\mathsf{v} = \partial\mathsf{u}/\partial\mathbf{n} = 0$ (free-slip)	$\partial\mathsf{p}/\partial\mathbf{n} = 0$

Table 3.1: Backward facing step numerical simulation boundary conditions.

instability then causes vortex shedding behind the step which results in a relatively low pressure region. Two main recirculation bubbles are present: the larger *primary recirculation bubble*, within which there is significant flow reversal, and the smaller *secondary recirculation bubble*. After the primary recirculation bubble the flow reattaches to the boundary and starts to develop a boundary layer again.

3.2 Numerical simulation

In this work, a 2D flow was considered at Re = 2000 (based on the step height h and free-stream velocity u_∞) and simulated numerically using LES. The computational domain depicted in Figure 3.2 was used, where the origin of the Cartesian coordinate system is located at the step edge. The boundary conditions implemented were as described by Table 3.1, where $\mathbf{n} \in \mathbb{R}^2$ denotes the boundary-normal unit vector.

A uniform velocity profile was prescribed at the inlet, and a large inlet length was used to ensure that a reasonably thick Blasius boundary layer ($\delta_{\text{BL}} \approx h$) had developed once the step edge was reached.

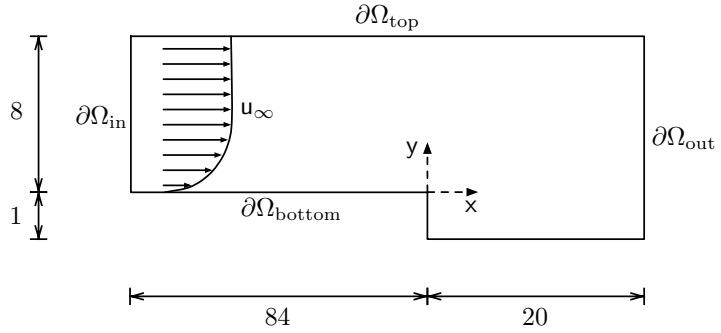


Figure 3.2: Backward facing step flow computational domain with boundaries $\partial\Omega_{(\cdot)}$.

3.2.1 Background on computational fluid dynamics

CFD is the branch of fluid dynamics concerned with the simulation of fluid flows. Given only a few analytical solutions to the equations governing fluid dynamics (1.1) exist, which are typically for very simple geometries, CFD techniques approximate solutions for more complex geometries numerically.

Traditional simulation approaches are typically divided into the following three categories[1]:

1. Direct numerical simulation (DNS).
2. Large eddy simulation (LES).
3. Reynolds averaged Navier-Stokes (RANS).

These categories differ in the way in which they model turbulence. DNS is conceptually the 'simplest' approach to simulating a fluid flow, in that it directy solves the governing equations (1.1) on a sufficiently fine computational mesh that all eddy length scales are resolved – i.e. it does not use a 'model' for turbulence. Whilst this is the most accurate method of simulation, it is by far the most computationally expensive approach, and despite its existence for many decades now, even with today's high performance computing power only simulations at low Reynolds numbers are computationally tractable [30].

The LES approach is to numerically solve a spatially filtered set of governing equations to resolve the large eddy scales in the flow, and use a model to approximate the effects of the smaller scales [30, 19]. This is acceptable as often it is the larger scales

[1]In addition to these traditional approaches there exists a separate class of methods developed more recently called Lattice Boltzmann methods which, rather than numerically solving the Navier-Stokes equations, simulate fluid flows using the discrete Boltzmann equation and particle collision models based on kinetic theory. These methods are not considered in this work, but the interested reader is referred to the review paper by Aidun and Clausen [5] for details.

which dominate momentum transfer [30]. The spatial filter is usually simply the computational mesh, and for this reason the model is called the sub-grid scale (SGS) model. Whilst the computational cost of LES simulations is still significant, it is considerably cheaper than DNS, and as such can be used for simulating complex geometry flows.

Finally, the least computationally expensive approach is RANS simulation. RANS simulations are based on the equations which result from Reynolds-decomposition – that is, decomposing the flow quantities into a time-averaged (or ensemble averaged) component and a fluctuating component [37]. Upon averaging, terms are produced which must be modelled using a turbulence model. The RANS approach is typically employed when time-averaged properties are of interest, such as the mean force acting on a surface.

In this work, LES simulations were employed as they achieve a good tradeoff between computational cost and sufficient resolution of spatial and temporal scales of the flow.

3.2.2 OpenFOAM

The open-source C++ based finite volume CFD code OpenFOAM [103] was used for the numerical simulations carried out in this work. More specifically, the pressure-implicit method for pressure linked equations (PIMPLE)[2] algorithm solver was used in conjunction with a k-equation eddy-viscosity model for the sub-grid scale model [37] (where $k \in \mathbb{R}^+$ is the sub-grid scale kinetic energy), with kinetic energy wall functions. The solver uses a second-order accurate central differencing scheme in space, and a backward Euler differencing scheme in time – the backward time centered space method (BTCS). This scheme is unconditionally stable [49], but an adjustable time-step was used in order to keep the Courant number small [77] such that temporal scales were well resolved. The recent paper by Robertson *et al.* [85] presents a validation study of OpenFOAM's reliability in simulating incompressible bluff body flows, largely based on time-averaged statistical data. The authors conclude that it is indeed reliable for such problems.

3.2.3 Mesh generation

A structured, graded mesh with increased cell density in the region aft of the step (as depicted in Figure 3.3) was used for all numerical simulations. Before deciding on a suitable mesh, simulations were run using six different meshes of increasing cell density

[2]The PIMPLE algorithm is a variation of the standard pressure-implicit with splitting of operator (PISO) algorithm first described by Issa [54] which incorporates outer correction loops, and allows the use of a larger timestep.

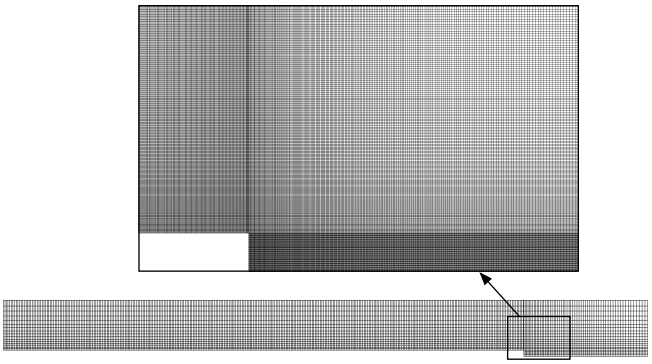


Figure 3.3: Final backward facing step computational mesh with approximately 1 in 50 cells shown for clarity.

mesh ID	n_c
MESH 1	6×10^3
MESH 2	25×10^3
MESH 3	119×10^3
MESH 4	217×10^3
MESH 5	309×10^3
MESH 6	529×10^3

Table 3.2: Total number of computational cells n_c for increasingly fine backward facing step computational meshes.

in order to ensure convergence. The total number of computational cells in each mesh are given in Table 3.2.

Several characteristics of the flow were considered in order to ensure the simulations' convergence upon mesh refinement. Firstly, time-averaged velocity profiles $\bar{\mathsf{u}}(\mathsf{y})$ at several locations in the wake of the step were considered (note that $\bar{\cdot}$ denotes temporal averaging of a quantity). These are shown in Figure 3.4, and it is clear that the time-averaged flowfield has converged by MESH 4, capturing the flow reversal present in the recirculation region.

The magnitude spectra of a pressure measurement on the bottom wall at $(\mathsf{x}, \mathsf{y}) = (0.1, -1)$ were computed and are plotted in Figure 3.5. Note that the spectra obtained for MESH 1 and MESH 2 are not plotted, as these were too coarse to capture any vortex shedding. The spectra obtained for the higher fidelity meshes converged to show the largest peak at $\mathrm{St} = \mathrm{St}_{\mathrm{shed}} = 0.064$ (see Table 3.3), with smaller peaks at harmonics of this frequency, i.e. at $2\mathrm{St}_{\mathrm{shed}}$, $3\mathrm{St}_{\mathrm{shed}}$, etc. The largest peak corresponds to the dominant shedding mode, and is in excellent agreement with Dahan *et al.* [28].

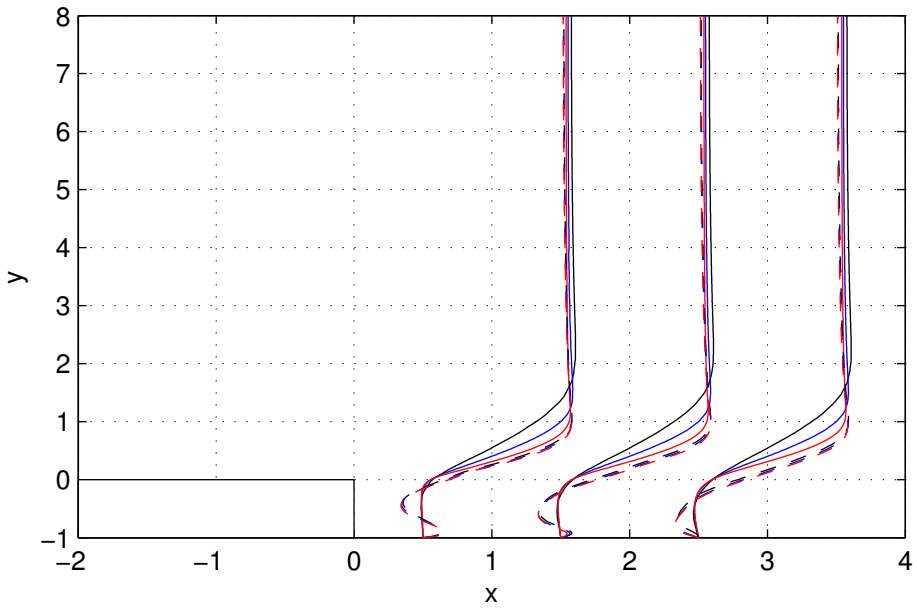


Figure 3.4: Time averaged velocity profiles $\bar{u}(y)$ at $x = 0.5, 1.5$, and 2.5: Mesh 1 (−), Mesh 2 (−), Mesh 3 (−), Mesh 4 (−−), Mesh 5 (−−), Mesh 6 (−−).

mesh ID	shedding frequency St_{shed}
Mesh 1	n/a
Mesh 2	n/a
Mesh 3	0.070
Mesh 4	0.063
Mesh 5	0.064
Mesh 6	0.064

Table 3.3: Shedding frequency St_{shed} observed for different computational meshes.

It was found that the reattachment length – the point at which the time-averaged wall shear stress is equal to zero – converged to $x_r = 6.3$, which is also in good agreement with Dahan *et al.* [28].

Figure 3.6 shows the time-averaged flowfield. The low pressure region responsible for pressure drag immediately aft of the step is evident in Figure 3.6(c). Figure 3.7 displays time-averaged streamlines which clearly depict the characteristic primary and secondary recirculation bubbles discussed in Section 3.1.

For the remainder of the work in this thesis, Mesh 5 was selected for numerical simulations.

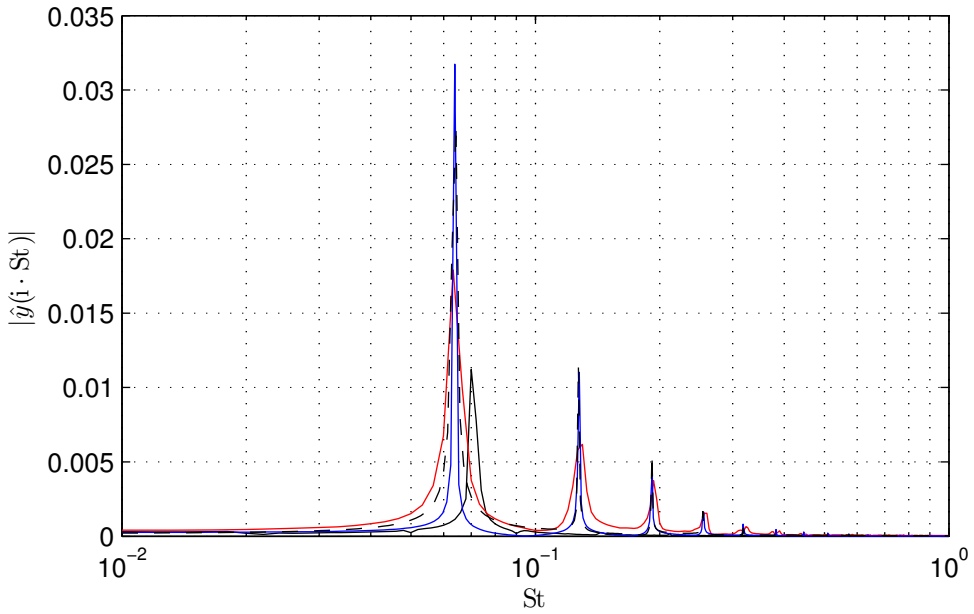


Figure 3.5: Magnitude spectra of $y(t) = \mathsf{p}(0.1, -1, t)$: Mesh 3 (−), Mesh 4 (−), Mesh 5 (−), Mesh 6 (−−).

3.2.4 Parallelisation

OpenFOAM uses the message passing interface (MPI) system for parallel computation. The spatial domain is decomposed into n_p subdomains, and at each computational timestep the calculation for each subdomain is computed on one of the n_p processing cores allocated to the job. Between timesteps, MPI is used for communication between cores.

The benefit of running a simulation in parallel is typically measured by the job's speedup $S_{n_\mathrm{p}} \in \mathbb{R}^+$ [37], where linear speedup, $S_{n_\mathrm{p}} = n_\mathrm{p}$, is the ideal case[3]. Speedup is defined as:

$$S_{n_\mathrm{p}} = \frac{T_\mathrm{s}}{T_{n_\mathrm{p}}}, \tag{3.3}$$

where T_s is the wall-clock time taken to complete a particular job in serial (one processing core), and T_{n_p} is the wall-clock time taken to complete the same job using n_p processing cores.

The University of Sheffield's HPC cluster Iceberg was used for all backward facing step flow simulations presented in this work, and the speedup achievable was studied by running simulations of a fixed number of non-dimensional time units on varying numbers

[3] 'Superlinear speedup' ($S_{n_\mathrm{p}} > n_\mathrm{p}$) is occasionally observed and is usually the result of a particularly efficient use of cache memory in the parallel processing environment, however this is atypical [4].

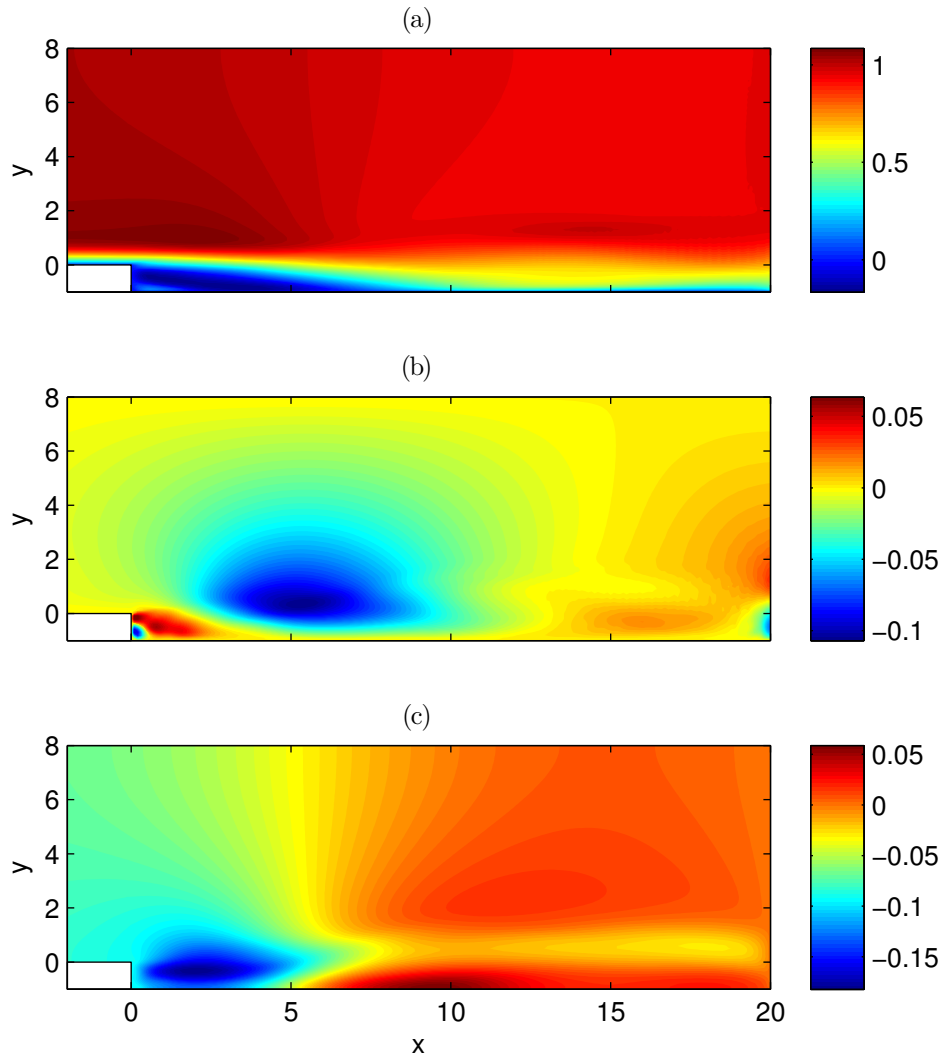


Figure 3.6: Backward facing step time-averaged flow: (a) $\bar{\mathsf{u}}(\mathbf{x})$; (b) $\bar{\mathsf{v}}(\mathbf{x})$; (c) $\bar{\mathsf{p}}(\mathbf{x})$.

of cores. Results are depicted in Figure 3.8. Reasonable speedup (albeit less than linear) was achieved up to 32 processing cores – the maximum number of cores allowed per job on the ICEBERG cluster. It should be noted that the the achievable speedup is dependent on the particular problem and solution algorithms used. Robertson *et al.* [85], for example, achieved much greater speedup up to 192 procesing cores when performing OPENFOAM RANS simulations of bluff body flows.

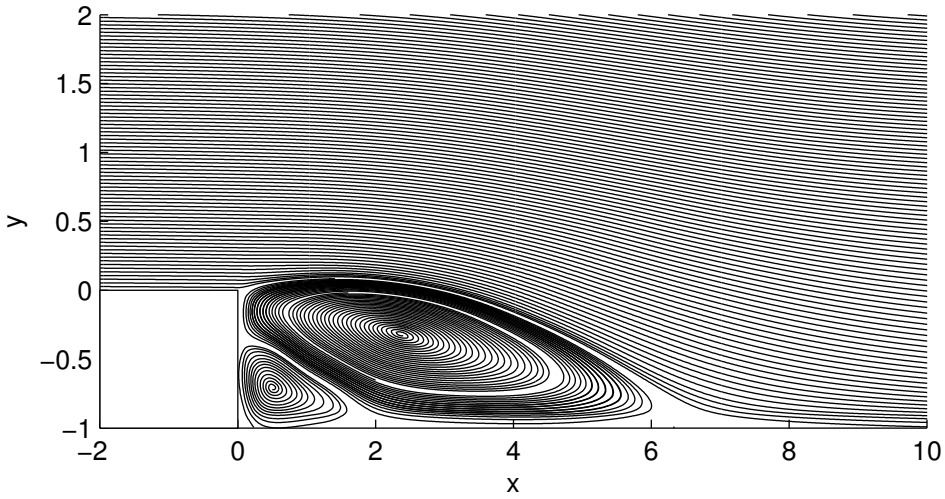


Figure 3.7: Backward facing step time-averaged streamlines.

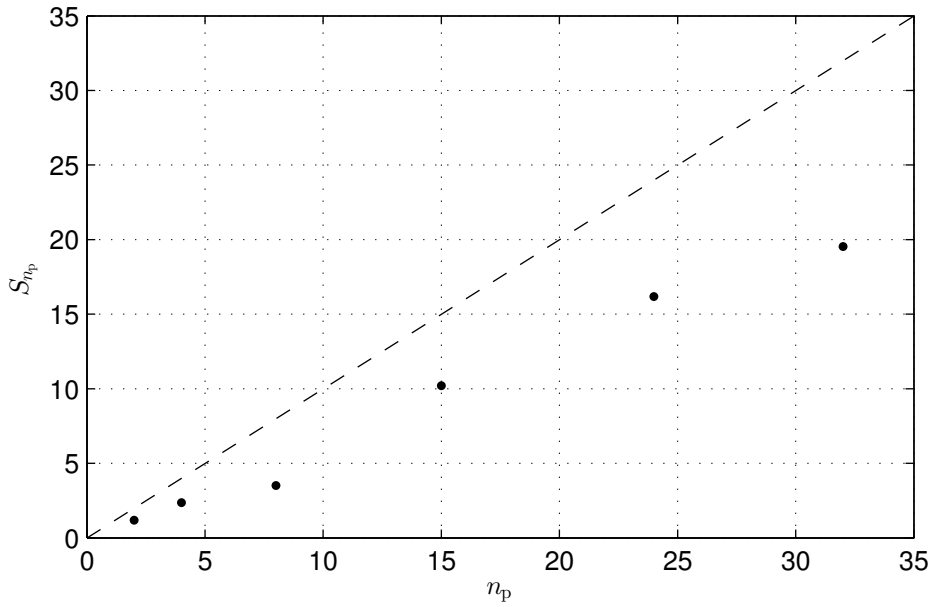


Figure 3.8: Speedup S_{n_p} achieved when backward facing step simulation was run on n_p processing cores of ICEBERG. Dashed line represents linear (ideal) speedup.

3.3 Actuation

As became evident in Chapter 2, a number of different actuator types have been implemented in previous flow control studies. The most common was ZNMF actuation, and this has been used in the present work. For a thorough discussion of different actuator options for flow control, the reader is referred to the review paper by Cattafesta and Sheplak [18].

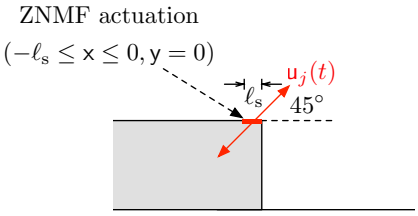


Figure 3.9: Backward facing step ZNMF actuator location.

Actuation, of slot width $\ell_s = 0.03$, was located on top of the step immediately before the edge at an angle of 45° to the horizontal, as depicted in Figure 3.9. In numerical simulations the actuation was modelled as a simple modified boundary condition (MBC) [79] with a top-hat spatial profile. As such, for the actuator slot region boundary $\partial\Omega_{\text{slot}} := \{\mathbf{x} \in \Omega : -\ell_s \leq x \leq 0, y = 0\}$, the velocity is given by:

$$\mathbf{u}(\mathbf{x}, t) = \begin{bmatrix} u_{\text{act}}(\mathbf{x}, t) \\ v_{\text{act}}(\mathbf{x}, t) \end{bmatrix} = \begin{bmatrix} \frac{1}{\sqrt{2}} \\ \frac{1}{\sqrt{2}} \end{bmatrix} u_j(t), \ \forall \ (\mathbf{x}, t) \in \partial\Omega_{\text{slot}} \times [0, t_f] \,, \tag{3.4}$$

where $u_j(t)$ is the actuator velocity.

Whilst other actuator slot spatial profiles could be employed, it was found in Kral *et al* [61] that modelling ZNMF actuation with a top-hat spatial profile yielded simulation results which best agreed with experiments when compared to several other profiles.

3.4 Linear system assumption

Whilst many linear control approaches to bluff body flow control have successfully been used in previous studies (many of which were discussed in Chapter 2), in order to fully justify the use of such methods in the present work, an output was defined and the linearity of the response to harmonic forcing (sinusoidal signals) at a number of forcing frequencies and amplitudes was analysed.

The output $y(t)$ was defined as the vertical velocity component at a point in the wake:

$$y(t) = v(1, -0.5, t), \tag{3.5}$$

and the actuation velocity followed a sinusoidal signal:

$$u_j(t) = A_j \sin\left(2\pi \cdot \text{St}_{\text{act}} \cdot t\right), \tag{3.6}$$

where A_j is the actuation amplitude, and St_{act} is the forcing frequency.

Simulations for a range of forcing frequencies $\mathrm{St}_{\mathrm{act}} \in \{0.01, 0.05, 0.1, 0.2, 0.4, 0.6, 1\}$ and amplitudes $A_j \in \{0.01, 0.1, 0.2, 0.3, 0.4\}$ were run, and the output spectra were computed. Figures 3.10–3.12 show these spectra.

It is immediately clear that at low amplitudes the system does indeed behave linearly. For forcing amplitude $A_j = 0.01$ the spectra were largely unchanged by the forcing, and the natural vortex shedding continued unperturbed. However, as the amplitude is increased, the effect of the nonlinearity becomes evident, and the forcing alters the output spectra not only just at the forcing frequency, but across many other frequencies too. For forcing amplitudes greater than $A_j = 0.2$, the actuation starts to dominate the flow, and, depending on forcing frequency, the natural vortex shedding at $\mathrm{St} = 0.064$ is no longer always present. This suggests that as the magnitude of control signals increases, the baseline flow – around which linear models capture the dynamics of flow perturbations – changes significantly. As such, careful consideration into what baseline flow is used to construct models around is important.

A scalar measure $\vartheta \in \mathbb{R}_0^+$ is defined here to quantify the difference in energy between forced and unforced output signals:

$$\vartheta := \frac{\left| \int_{-\infty}^{\infty} \left|\hat{y}_{\mathrm{unforced}}(\mathrm{i} \cdot \mathrm{St})\right|^2 \mathrm{dSt} - \int_{-\infty}^{\infty} \left|\check{\hat{y}}_{\mathrm{forced}}(\mathrm{i} \cdot \mathrm{St})\right|^2 \mathrm{dSt} \right|}{\int_{-\infty}^{\infty} \left|\hat{y}_{\mathrm{unforced}}(\mathrm{i} \cdot \mathrm{St})\right|^2 \mathrm{dSt}}, \tag{3.7}$$

where $\check{\hat{y}}_{\mathrm{forced}}(\mathrm{i} \cdot \mathrm{St})$ denotes the forced system's output spectra after removing the peak corresponding to input forcing – i.e. removing the energy content attributed to the input forcing. Therefore, $\vartheta = 0$ means that the energy content of the forced system output signal is the same as that of the unforced system output signal, suggesting that the system behaves perfectly linearly. Conversely, $\vartheta \gg 0$ suggests a larger difference in energy, due to the nonlinearity moving energy from the forcing frequency to other frequencies. Figure 3.13 depicts the value of ϑ computed for each of the test cases.

As one would expect from analysing Figures 3.10–3.12, ϑ is smallest for the smaller forcing amplitudes, and increases as the forcing amplitudes are increased. This confirms that the system behaves in a linear fashion for small control signals. Since in feedback control the required signals are typically small, the linear control approach employed in this work is justified.

3.5 Summary

The main features of the backward facing step flow, which serves as the complex geometry flow test case in this work, have been presented in this chapter.

Details of the numerical simulation procedure employed in this work have been discussed, including details of the solution algorithm used, a study of the convergence of properties of the solution upon mesh refinement, and aspects of parallelisation of the simulations.

The numerical implementation of ZNMF actuation has been explained, and the assumption that linear systems theory is valid for describing the flow dynamics was justified.

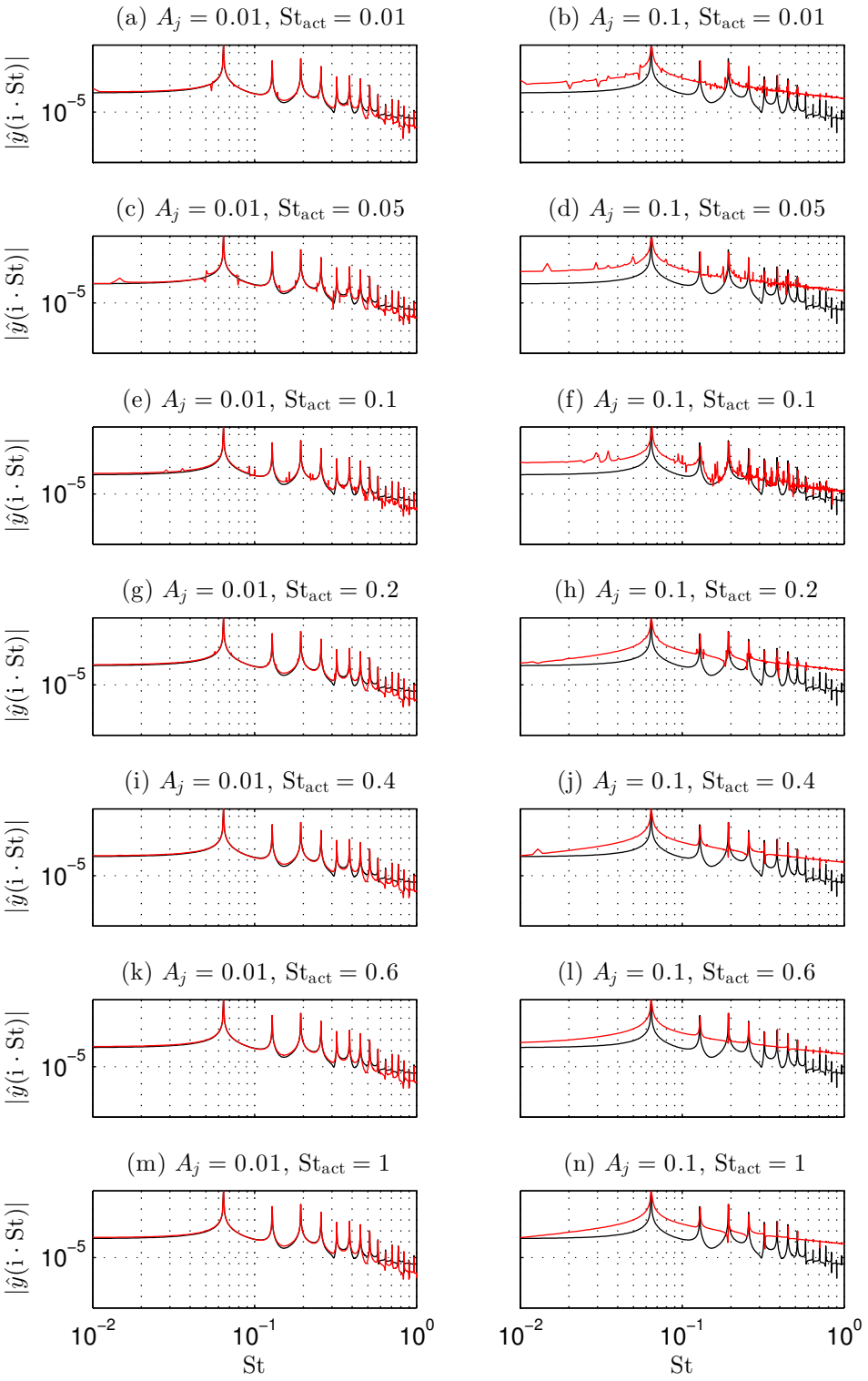


Figure 3.10: Output magnitude spectra $|\hat{y}(\mathrm{i} \cdot \mathrm{St})|$ of harmonically forced flow (−) with various forcing frequencies $\mathrm{St_{act}}$ and amplitudes A_j. Unforced flow (−) is shown for comparison.

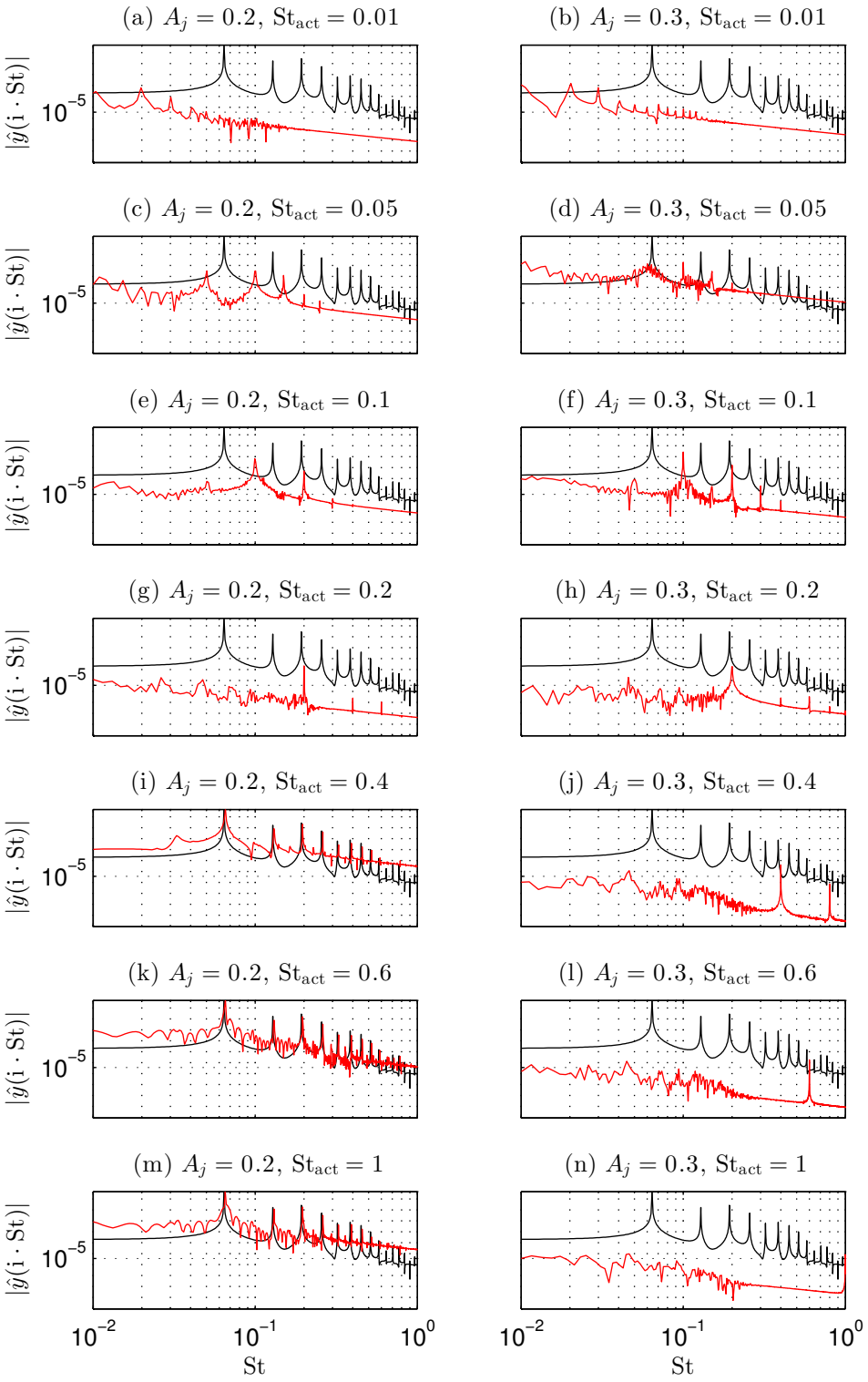


Figure 3.11: Output magnitude spectra $|\hat{y}(\mathrm{i} \cdot \mathrm{St})|$ of harmonically forced flow (−) with various forcing frequencies $\mathrm{St}_{\mathrm{act}}$ and amplitudes A_j. Unforced flow (−) is shown for comparison.

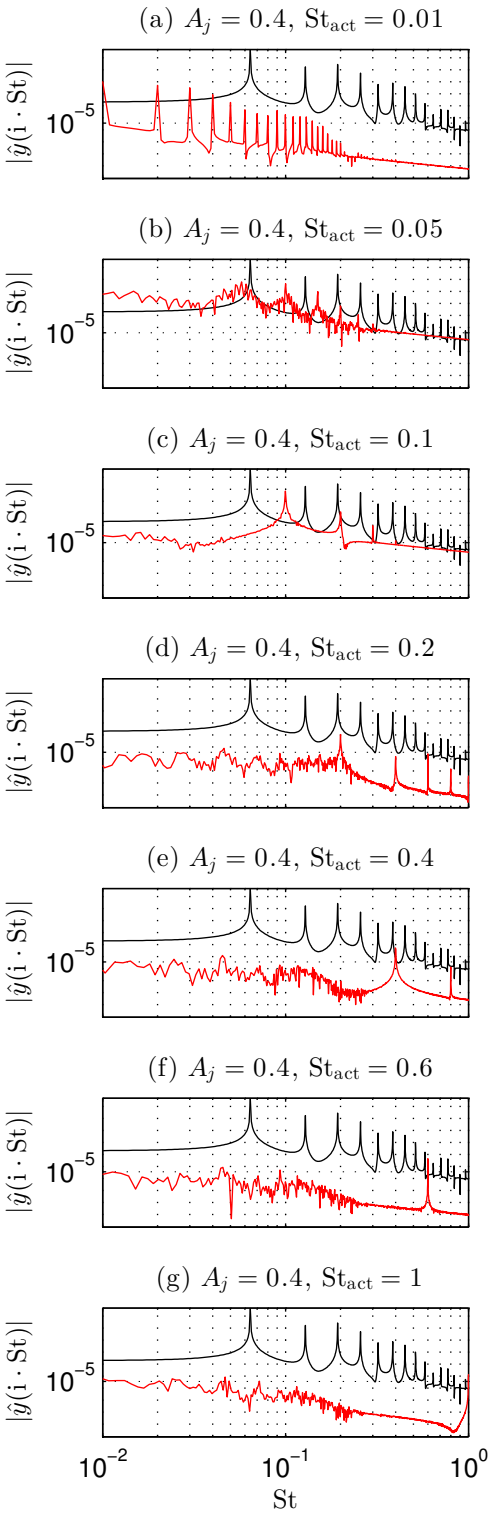


Figure 3.12: Output magnitude spectra $|\hat{y}(\mathrm{i} \cdot \mathrm{St})|$ of harmonically forced flow (−) with various forcing frequencies $\mathrm{St}_{\mathrm{act}}$ and amplitudes A_j. Unforced flow (−) is shown for comparison.

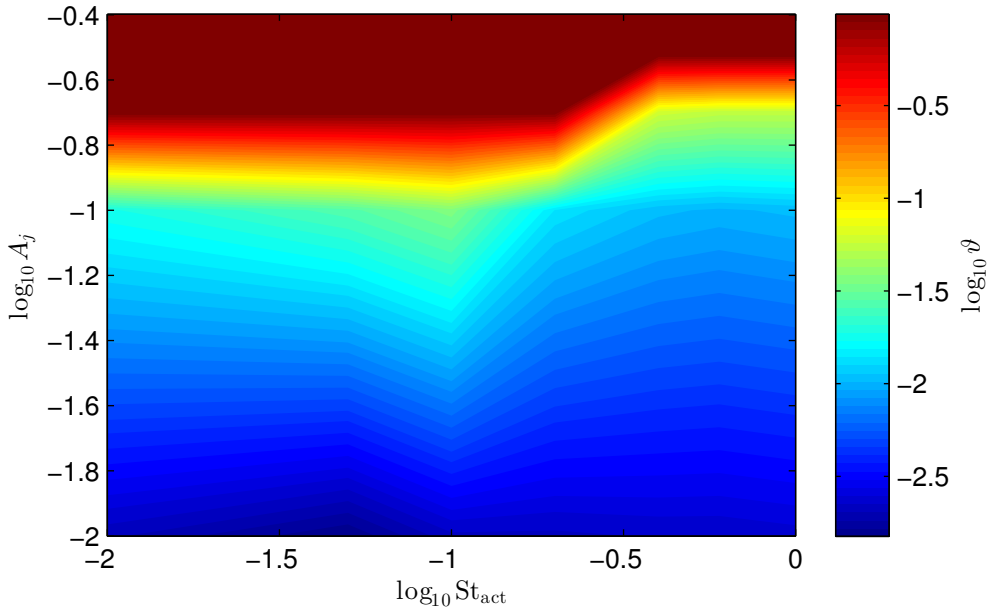


Figure 3.13: Contours of $\log_{10} \vartheta$ for different forcing frequencies and amplitudes.

Chapter 4

Low-order modelling of PDAEs

This chapter introduces the low-order modelling technique which is employed in this work. The methodology – which is a generalisation of the modelling approach used by Baramov *et al.* [10, 11], who used a similar technique to obtain models of a 2D channel flow – avoids some of the difficulties associated with other approaches (as discussed in Chapter 2), chiefly that expensive and time-consuming simulations and/or experiments required by system identification procedures are unnecessary, and that large, ill-conditioned system matrices are never explicitly constructed. The methodology can readily be applied to complex geometry flows in order to obtain input-output frequency response.

The modelling technique is presented, along with a spatial domain decomposition optimisation which greatly reduces the computational complexity of the method. The method's efficacy is demonstrated by application to a wave-diffusion equation example for which a benchmark model can easily be constructed. Results are presented which confirm that this modelling approach produces identical frequency response data to the benchmark model, at a lessened computational cost, and with more desirable computational properties.

4.1 Modelling technique overview

The aim of this modelling approach is to obtain low-dimensional models of PDAE systems which are suitable for feedback control, whilst avoiding the necessity to construct, store, or invert extremely large state-space matrices, or run numerous computationally expensive simulations. The overall frequency response of a system is built up by connecting together the frequency responses of a large number of computational node subsystems in an efficient manner by exploiting the inherent structure of spatially

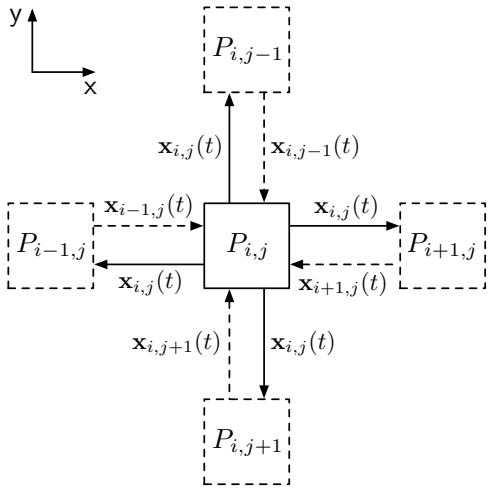


Figure 4.1: Interconnection of neighbouring subsystems after spatial discretisation of a linear PDAE.

discretised PDAEs. This exposes what can be very low-order behaviour, despite the high-dimensionality of the original plant [28].

In addition, the approach lends itself to using the gap metric bound construction theory presented in [55], which, by designing a suitable $\mathscr{H}_\infty$ robust controller, can guarantee closed-loop stability of the infinite-dimensional plant.

4.1.1 Spatial discretisation and formulation of nodal subsystems

If the underlying dynamics of the system are governed by a nonlinear PDAE, the first step is to linearise the PDAE around the desired operating condition (assuming the use of linear control is justifiable) and discretise in space using centred finite-differences, yielding simple interconnections between adjacent nodes. For every node on the computational mesh, a low-dimensional descriptor[1] state-space model can be deduced, and treated as an individual subsystem $P_{i,j}$ whose inputs and outputs correspond to the flow of state information from and to its neighbouring nodes, respectively (with additional control inputs and measured outputs in the case of nodes representing actuation or sensing). This is depicted in Figure 4.1 for the two spatial dimension case, where i and j are node indices in the x and y directions, respectively.

[1]For PDE (as opposed to PDAE) systems these will be standard state-space (as opposed to descriptor) systems.

Each of these subsystems has a descriptor state-space representation of the form:

$$E_{i,j}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_{i,j}(t) = A_{i,j}\mathbf{x}_{i,j}(t) + B_{i,j}\boldsymbol{\xi}_{i,j}(t) + B_{\mathbf{u}_{i,j}}\mathbf{u}_{i,j}(t), \tag{4.1a}$$

$$\mathbf{z}_{i,j}(t) = C_{i,j}\mathbf{x}_{i,j}(t), \tag{4.1b}$$

$$\mathbf{y}_{i,j}(t) = C_{\mathbf{y}_{i,j}}\mathbf{x}_{i,j}(t), \tag{4.1c}$$

where, for a PDAE in $d \in \{1,2,3\}$ spatial dimensions, $\mathbf{x}_{i,j}(t) \in \mathbb{R}^n$ is the node's state vector, $\boldsymbol{\xi}_{i,j}(t) \in \mathbb{R}^{2dn}$ is a vector of state information flowing in from neighbouring nodes – in the two spatial dimension case for example:

$$\boldsymbol{\xi}_{i,j}(t) = \begin{bmatrix}\mathbf{x}_{i,j-1}^\top(t) & \mathbf{x}_{i+1,j}^\top(t) & \mathbf{x}_{i,j+1}^\top(t) & \mathbf{x}_{i-1,j}^\top(t)\end{bmatrix}^\top \in \mathbb{R}^{4n},$$

$\mathbf{u}_{i,j}(t) \in \mathbb{R}^q$ is a vector of control inputs, $\mathbf{y}_{i,j}(t) \in \mathbb{R}^p$ is a vector of measured outputs, $\mathbf{z}_{i,j}(t) \in \mathbb{R}^{2dn}$ is the output of the node's state information to neighbouring nodes, $E_{i,j}, A_{i,j} \in \mathbb{R}^{n\times n}$ and $B_{i,j} \in \mathbb{R}^{n\times 2dn}$ are matrices which arise in the spatial discretisation,

$$C_{i,j} := \begin{bmatrix} I & I & \cdots & I\end{bmatrix}^\top \in \mathbb{R}^{2dn\times d},$$

and $B_{\mathbf{u}_{i,j}} \in \mathbb{R}^{n\times q}$ and $C_{\mathbf{y}_{i,j}} \in \mathbb{R}^{p\times n}$ are matrices which describe how control inputs affect the states, and how measured outputs are defined as linear combinations of the states, respectively. Note that for many systems of interest the majority of computational nodes' subsystems do not have control inputs or measured outputs, only nodes corresponding to actuation or sensing locations will possess these inputs and outputs.

Taking Laplace transforms of (4.1), and prescribing initial conditions $\mathbf{x}_{i,j}(0) = 0$ yields:

$$\begin{bmatrix}\tilde{\mathbf{z}}_{i,j}(s)\\ \tilde{\mathbf{y}}_{i,j}(s)\end{bmatrix} = \underbrace{\begin{bmatrix}C_{i,j}\\ C_{\mathbf{y}_{i,j}}\end{bmatrix}(sE_{i,j} - A_{i,j})^{-1}\begin{bmatrix}B_{i,j} & B_{\mathbf{u}_{i,j}}\end{bmatrix}}_{P_{i,j}(s)}\begin{bmatrix}\tilde{\boldsymbol{\xi}}_{i,j}(s)\\ \tilde{\mathbf{u}}_{i,j}(s)\end{bmatrix}, \tag{4.2}$$

where $P_{i,j}(s) \in \mathscr{R}^{(2dn+p)\times(2dn+q)}$ is a real-rational transfer function matrix, $\tilde{\cdot}$ denotes a Laplace transformed quantity, and $s \in \mathbb{C}$.

The subsystem (4.2) is defined as a computational 'atom'. For each and every atom on the domain, the frequency response $P_{i,j}(\mathrm{i}\omega) \in \mathbb{C}^{(2dn+p)\times(2dn+q)}$ for a particular frequency $\omega \in \mathbb{R}$ is computed by evaluating $P_{i,j}(s)$ with $s = \mathrm{i}\omega$, where $\mathrm{i} := \sqrt{-1}$. Whilst matrix inversion is a costly operation (in big-O notation, $\mathcal{O}\left(n^3\right)$ flops[2]), evaluating an

[2]Floating point operations, defined as a single addition, subtraction, multiplication or division between two floating point numbers [42].

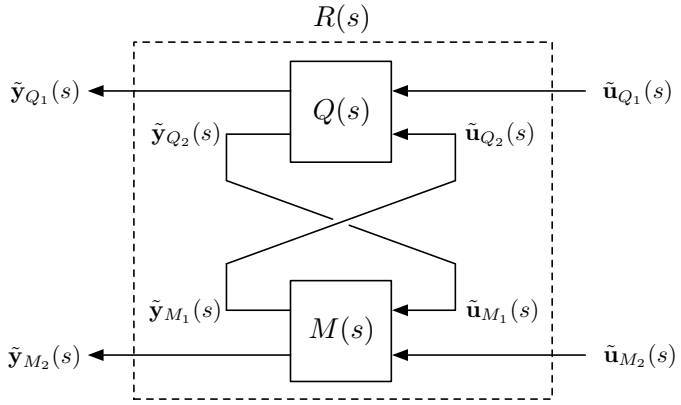


Figure 4.2: Redheffer star product between two systems.

atom's frequency response is cheap as the most expensive operation is the inversion of $(\mathrm{i}\omega E_{i,j} - A_{i,j})$, which for fluids problems is small (4×4 or smaller).

Neighbouring atoms' frequency responses can be connected together using the Redheffer star product (RHSP) operation, also known as the generalised linear fractional transformation [33].

4.1.2 Redheffer star product

The RHSP describes the interconnection of two multi-input multi-output (MIMO) systems. The block diagram of the RHSP between two systems $Q(s)$ and $M(s)$ is depicted in Figure 4.2. Assuming $Q(s)$ and $M(s)$ are partitioned (compatibly) as such:

$$\begin{bmatrix} \tilde{\mathbf{y}}_{Q_1}(s) \\ \tilde{\mathbf{y}}_{Q_2}(s) \end{bmatrix} = \underbrace{\begin{bmatrix} Q_{11}(s) & Q_{12}(s) \\ Q_{21}(s) & Q_{22}(s) \end{bmatrix}}_{Q(s)} \begin{bmatrix} \tilde{\mathbf{u}}_{Q_1}(s) \\ \tilde{\mathbf{u}}_{Q_2}(s) \end{bmatrix}, \tag{4.3a}$$

$$\begin{bmatrix} \tilde{\mathbf{y}}_{M_1}(s) \\ \tilde{\mathbf{y}}_{M_2}(s) \end{bmatrix} = \underbrace{\begin{bmatrix} M_{11}(s) & M_{12}(s) \\ M_{21}(s) & M_{22}(s) \end{bmatrix}}_{M(s)} \begin{bmatrix} \tilde{\mathbf{u}}_{M_1}(s) \\ \tilde{\mathbf{u}}_{M_2}(s) \end{bmatrix}, \tag{4.3b}$$

where $Q(s) \in \mathscr{R}^{(a_{Q_1}+a_{Q_2})\times(b_{Q_1}+b_{Q_2})}$, $M(s) \in \mathscr{R}^{(a_{M_1}+a_{M_2})\times(b_{M_1}+b_{M_2})}$, $a_{Q_2} = b_{M_1}$, and $a_{M_1} = b_{Q_2}$, the Redheffer star product $\cdot \star \cdot : \mathscr{R}^{(a_{Q_1}+a_{Q_2})\times(b_{Q_1}+b_{Q_2})} \times$

$\mathscr{R}^{(a_{M_1}+a_{M_2})\times(b_{M_1}+b_{M_2})} \to \mathscr{R}^{(a_{Q_1}+a_{M_2})\times(b_{Q_1}+b_{M_2})}$ is defined as [39]:

$$\begin{aligned} R(s) &= Q(s) \star M(s) \\ &:= \begin{bmatrix} Q_{11} + Q_{12}M_{11}\left(I - Q_{22}M_{11}\right)^{-1}Q_{21} & Q_{12}M_{11}\left(I - Q_{22}M_{11}\right)^{-1}Q_{22}M_{12} + Q_{12}M_{12} \\ M_{21}\left(I - Q_{22}M_{11}\right)^{-1}Q_{21} & M_{21}\left(I - Q_{22}M_{11}\right)^{-1}Q_{22}M_{12} + M_{22} \end{bmatrix}, \end{aligned} \tag{4.4}$$

which yields the overall interconnection:

$$\begin{bmatrix} \tilde{\mathbf{y}}_{Q_1}(s) \\ \tilde{\mathbf{y}}_{M_2}(s) \end{bmatrix} = R(s) \begin{bmatrix} \tilde{\mathbf{u}}_{Q_1}(s) \\ \tilde{\mathbf{u}}_{M_2}(s) \end{bmatrix}. \tag{4.5}$$

In order to compute the frequency response from $\begin{bmatrix} \tilde{\mathbf{u}}^*_{Q_1}(s) & \tilde{\mathbf{u}}^*_{M_2}(s) \end{bmatrix}^*$ to $\begin{bmatrix} \tilde{\mathbf{y}}^*_{Q_1}(s) & \tilde{\mathbf{y}}^*_{M_2}(s) \end{bmatrix}^*$, $Q(s)$ and $M(s)$ are simply evaluated on the imaginary axis $s = \mathrm{i}\omega$.

The overall frequency response of the PDAE system $G(\mathrm{i}\omega)$, from control inputs $\tilde{\mathbf{u}}(s)$ to measured outputs $\tilde{\mathbf{y}}(s)$, is thus obtained by 'chaining' the individual atoms' frequency responses together with the RHSP:

$$G(\mathrm{i}\omega) = P_{1,1}(\mathrm{i}\omega) \star P_{1,2}(\mathrm{i}\omega) \star \cdots \star P_{i,j}(\mathrm{i}\omega) \star \cdots \star P_{n_i,n_j}(\mathrm{i}\omega). \tag{4.6}$$

The ordering of the chaining is depicted in Figure 4.3. As such, this modelling approach is referred to as the *RHSP-based construction method* in this work.

This procedure is completed for a number of frequencies of interest, before a low-order transfer function is fitted to the frequency response using, for example, least squares regression. By performing these steps a number of times on successively finer computational meshes, an upper bound on the ν-gap [97, 98] between the finite-dimensional approximation and the original infinite-dimensional plant may be obtained [55], such that $\mathscr{H}_\infty$ robust controllers can be designed which guarantee closed-loop stability. Note that since the frequency response obtained at one frequency is completely independent of that obtained at other frequencies, the problem is *embarrassingly parallel*. This can be exploited by performing the computations on a HPC cluster, dedicating a single processing core to each frequency point of interest.

4.1.3 Computational cost

Here, (linear) PDAE systems in two spatial dimensions are considered, although analogous results could be deduced for the three spatial dimension case. Whilst the modelling approach described above is well suited to complex geometries, in the following

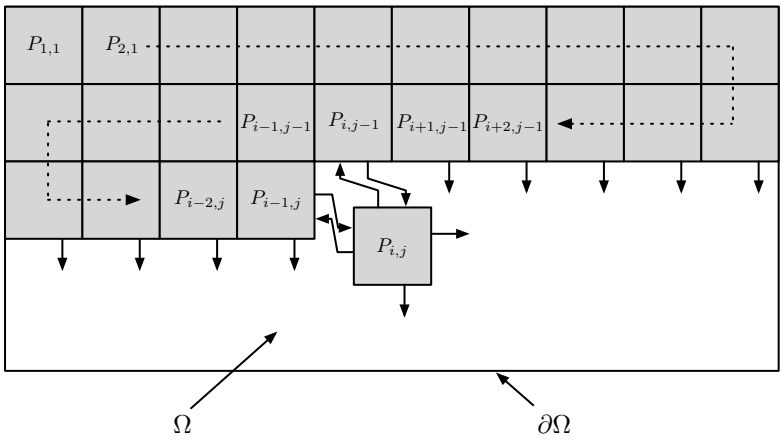


Figure 4.3: Connecting computational atoms together to obtain overall PDAE system frequency response.

matrix operation	big-O cost (flops)
addition, $A + B$, where $A, B \in \mathbb{C}^{a\times b}$	$\mathcal{O}(ab)$
multiplication, AB, where $A \in \mathbb{C}^{a\times b}, B \in \mathbb{C}^{b\times c}$	$\mathcal{O}(abc)$
inversion, A^{-1}, where $A \in \mathbb{C}^{a\times a}$	$\mathcal{O}\left(a^3\right)$

Table 4.1: Computational cost of matrix operations.

a rectangular domain $\Omega := [-\ell_{\mathsf{x}}/2, \ell_{\mathsf{x}}/2] \times [-\ell_{\mathsf{y}}/2, \ell_{\mathsf{y}}/2] \subset \mathbb{R}^2$ is assumed with uniform computational mesh density $\varrho \in \mathbb{N}$ (nodes per unit length) in both directions, resulting in $n_{\mathsf{x}} \in \mathbb{N}$ and $n_{\mathsf{y}} \in \mathbb{N}$ computational nodes in the x and y directions, respectively.

Noting the cost of typical matrix operations as presented in Table 4.1, summing all the operations required to construct all $n_{\text{total}} = n_{\mathsf{x}} n_{\mathsf{y}}$ atoms and evaluate (4.6) yields the overall computational cost $\mathcal{C} \in \mathbb{R}^+$:

$$\mathcal{C} = \mathcal{O}\left(n_{\mathsf{x}}^3 n_{\mathsf{y}} \left(1+n\right) n^2 + 2n_{\mathsf{x}}^2 n_{\mathsf{y}} \left(1+6n\right) n^2 + 52 n_{\mathsf{x}} n_{\mathsf{y}} n^3\right), \tag{4.7}$$

where $n \in \mathbb{N}$ is the state dimension of each subsystem.

Since $n_{\mathsf{x}} = \varrho \ell_{\mathsf{x}}$ and $n_{\mathsf{y}} = \varrho \ell_{\mathsf{y}}$, for given ϱ, ℓ_{x}, and ℓ_{y}, (4.7) can be written:

$$\mathcal{C} = c_1 \ell_{\mathsf{x}}^3 \ell_{\mathsf{y}} \left(1+n\right) n^2 \varrho^4 + 2c_2 \ell_{\mathsf{x}}^2 \ell_{\mathsf{y}} \left(1+6n\right) n^2 \varrho^3 + 52 c_3 \ell_{\mathsf{x}} \ell_{\mathsf{y}} n^3 \varrho^2, \tag{4.8}$$

where $c_1, ..., c_3 \in \mathbb{R}^+$ are unknown constants. Hence the cost is $\mathcal{O}\left(\varrho^4\right)$ to leading order.

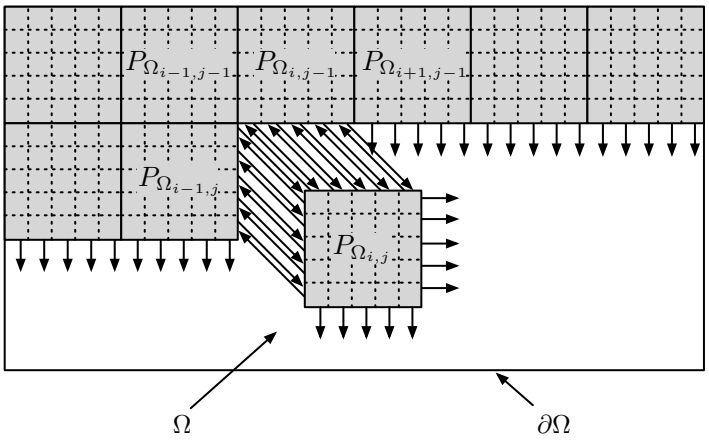


Figure 4.4: Connecting computational atoms together via domain decomposition to obtain overall PDAE system frequency response.

4.2 Optimisation by domain decomposition

The complexity of the above modelling approach can be reduced further by employing a domain decomposition optimisation. The computational mesh is first split into $n_\Omega = n_{\Omega_\mathsf{x}} n_{\Omega_\mathsf{y}} \in \mathbb{N}$ subdomains, defined here as computational 'molecules', with $n_{\Omega_\mathsf{x}} \in \mathbb{N}$ and $n_{\Omega_\mathsf{y}} \in \mathbb{N}$ molecules in the x and y directions, respectively. For each molecule, the individual atom frequency responses within the molecule are connected together using the RHSP resulting in the molecule frequency response:

$$P_{\Omega_{i,j}}(\mathrm{i}\omega) = P_{1,1}(\mathrm{i}\omega) \star P_{2,1}(\mathrm{i}\omega) \star \cdots \star P_{n_i,n_j}(\mathrm{i}\omega). \tag{4.9}$$

The overall PDAE system frequency response $G(\mathrm{i}\omega)$ is then obtained by connecting each of the molecules together:

$$G(\mathrm{i}\omega) = P_{\Omega_{1,1}}(\mathrm{i}\omega) \star P_{\Omega_{2,1}}(\mathrm{i}\omega) \star \cdots \star P_{\Omega_{n_{\Omega_\mathsf{x}},n_{\Omega_\mathsf{y}}}}(\mathrm{i}\omega). \tag{4.10}$$

This is depicted in Figure 4.4.

For the two spatial dimension case, the overall computational complexity of the modelling approach can be reduced from $\mathcal{O}\left(\varrho^4\right)$ to $\mathcal{O}\left(\varrho^2\right)$ by employing the domain decomposition optimisation. The following Lemma establishes existence and uniqueness of values of n_{Ω_x} and n_{Ω_y} which minimise the computational cost:

Lemma 1. *For a given mesh density ϱ, there exist unique values $n_{\Omega_\mathsf{x}} = n_{\Omega_x,\mathrm{opt}}$ and $n_{\Omega_\mathsf{y}} = n_{\Omega_y,\mathrm{opt}}$ which minimise the overall cost of evaluating* (4.9) *for each molecule, and finally evaluating* (4.10)*.*

Proof. Summing the cost of all the computational operations required to evaluate (4.9)

for each molecule, and finally evaluate (4.10) yields the overall computational cost:

$$\begin{aligned}\mathcal{C} = \mathcal{O}\Bigg(& n_{\mathsf{x}}^3 n_{\Omega_{\mathsf{x}}}^{-2} n_{\mathsf{y}}(1+2n)n^2 + 2n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} n_{\mathsf{y}}(1+6n)n^2 + 32 n_{\mathsf{x}} n_{\mathsf{y}} n^3 + 4 n_{\mathsf{x}} n_{\mathsf{y}} n^2 \\ & + 8 n_{\mathsf{x}} n_{\mathsf{y}} n + 16 \left(n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} + n_{\mathsf{y}} n_{\Omega_{\mathsf{y}}}^{-1} \right)^3 n_{\Omega_{\mathsf{x}}} n_{\Omega_{\mathsf{y}}} n^3 \\ & + 5 n_{\Omega_{\mathsf{x}}} n_{\Omega_{\mathsf{y}}} \left(n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} + n_{\mathsf{y}} n_{\Omega_{\mathsf{y}}}^{-1} \right)^2 \left(n_{\mathsf{x}} - n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} \right) n^3 \\ & + n_{\Omega_{\mathsf{x}}} n_{\Omega_{\mathsf{y}}} \left(n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} + n_{\mathsf{y}} n_{\Omega_{\mathsf{y}}}^{-1} \right)^2 n^2 + n_{\Omega_{\mathsf{x}}} n_{\Omega_{\mathsf{y}}} \left(n_{\mathsf{x}} - n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} \right)^2 n^2 \\ & + n_{\Omega_{\mathsf{x}}} n_{\Omega_{\mathsf{y}}} \left(n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} + n_{\mathsf{y}} n_{\Omega_{\mathsf{y}}}^{-1} \right) \left(n_{\mathsf{x}} - n_{\mathsf{x}} n_{\Omega_{\mathsf{x}}}^{-1} \right) n^2 \Bigg).\end{aligned} \tag{4.11}$$

The optimal values of $n_{\Omega_{\mathsf{x}}}$ and $n_{\Omega_{\mathsf{y}}}$ which achieve the complexity reduction discussed above are those which satisfy:

$$\begin{aligned}\frac{\partial \mathcal{C}}{\partial n_{\Omega_{\mathsf{x}}}} = 0 = & -2c_1 \ell_{\mathsf{x}}^3 \ell_{\mathsf{y}} (1+n) n^2 n_{\Omega_{\mathsf{x}}}^{-3} \varrho^4 - 48 c_6 \ell_{\mathsf{x}}^2 \ell_{\mathsf{y}} n^3 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^3 \\ & - 32 c_6 \ell_{\mathsf{x}}^3 n_{\Omega_{\mathsf{y}}} n^3 n_{\Omega_{\mathsf{x}}}^{-3} \varrho^3 + 16 c_6 \ell_{\mathsf{y}}^3 n^3 n_{\Omega_{\mathsf{y}}}^{-2} \varrho^3 + 10 c_7 \ell_{\mathsf{x}}^2 \ell_{\mathsf{y}} n^3 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^3 \\ & - 5 c_7 \ell_{\mathsf{x}}^3 n_{\Omega_{\mathsf{y}}} n^3 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^3 + 5 c_7 \ell_{\mathsf{x}} \ell_{\mathsf{y}}^2 n^3 n_{\Omega_{\mathsf{y}}}^{-1} \varrho^3 - c_8 n_{\Omega_{\mathsf{y}}} \ell_{\mathsf{x}}^2 n^2 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^2 \\ & + 10 c_7 \ell_{\mathsf{x}}^3 n_{\Omega_{\mathsf{y}}} n^3 n_{\Omega_{\mathsf{x}}}^{-3} \varrho^3 + c_8 \ell_{\mathsf{y}}^2 n^2 n_{\Omega_{\mathsf{y}}}^{-1} \varrho^2 + c_9 \ell_{\mathsf{x}}^2 n_{\Omega_{\mathsf{y}}} n^2 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^2 \\ & + c_9 \ell_{\mathsf{x}} \ell_{\mathsf{y}} n^2 \varrho^2 - c_{10} \ell_{\mathsf{x}}^2 n_{\Omega_{\mathsf{y}}} n^2 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^2 + c_{10} \ell_{\mathsf{x}}^2 n_{\Omega_{\mathsf{y}}} n^2 \varrho^2 \\ & - 2 c_2 \ell_{\mathsf{x}} \ell_{\mathsf{y}} (1+6n) n^2 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^2,\end{aligned} \tag{4.12}$$

and

$$\begin{aligned}\frac{\partial \mathcal{C}}{\partial n_{\Omega_{\mathsf{y}}}} = 0 = & -48 c_6 \ell_{\mathsf{x}} \ell_{\mathsf{y}}^2 n^3 n_{\Omega_{\mathsf{y}}}^{-2} \varrho^3 + 16 c_6 \ell_{\mathsf{x}}^3 n^3 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^3 + 5 c_7 \ell_{\mathsf{x}}^3 n^3 n_{\Omega_{\mathsf{x}}}^{-1} \varrho^3 \\ & - 32 c_6 \ell_{\mathsf{y}}^3 n_{\Omega_{\mathsf{x}}} n^3 n_{\Omega_{\mathsf{y}}}^{-3} \varrho^3 + 5 c_7 \ell_{\mathsf{x}} \ell_{\mathsf{y}}^2 n^3 n_{\Omega_{\mathsf{y}}}^{-2} \varrho^3 + c_8 \ell_{\mathsf{x}}^2 n^2 n_{\Omega_{\mathsf{x}}}^{-1} \varrho^2 \\ & - 5 c_7 \ell_{\mathsf{x}} \ell_{\mathsf{y}}^2 n_{\Omega_{\mathsf{x}}} n^3 n_{\Omega_{\mathsf{y}}}^{-2} \varrho^3 - 5 c_7 \ell_{\mathsf{x}}^3 n^3 n_{\Omega_{\mathsf{x}}}^{-2} \varrho^3 - c_8 n_{\Omega_{\mathsf{x}}} \ell_{\mathsf{y}}^2 n^2 n_{\Omega_{\mathsf{y}}}^{-2} \varrho^2 \\ & + c_9 \ell_{\mathsf{x}}^2 n^2 \varrho^2 - c_9 \ell_{\mathsf{x}}^2 n^2 n_{\Omega_{\mathsf{x}}}^{-1} \varrho^2 + c_{10} \ell_{\mathsf{x}}^2 n^2 n_{\Omega_{\mathsf{x}}}^{-1} \varrho^2 \\ & - 2 c_{10} \ell_{\mathsf{x}}^2 n^2 \varrho^2 + c_{10} \ell_{\mathsf{x}}^2 n_{\Omega_{\mathsf{x}}} n^2 \varrho^2,\end{aligned} \tag{4.13}$$

which can be written:

$$a n_{\Omega_{\mathsf{x}}}^3 + b n_{\Omega_{\mathsf{x}}} + c = 0, \tag{4.14}$$

and

$$d n_{\Omega_{\mathsf{y}}}^3 + e n_{\Omega_{\mathsf{y}}} + f = 0, \tag{4.15}$$

respectively, where:

$$\begin{aligned} a :=\ & 16c_6\ell_{\mathrm{y}}^3 n^3 n_{\Omega_{\mathrm{y}}}^{-2}\varrho^3 + 5c_7\ell_{\mathrm{x}}\ell_{\mathrm{y}}^2 n^3 n_{\Omega_{\mathrm{y}}}^{-1}\varrho^3 + c_8\ell_{\mathrm{y}}^2 n^2 n_{\Omega_{\mathrm{y}}}^{-1}\varrho^2 + c_9\ell_{\mathrm{x}}\ell_{\mathrm{y}} n^2\varrho^2 \\ & + c_{10}\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2\varrho^2, \end{aligned} \tag{4.16a}$$

$$\begin{aligned} b :=\ & 10c_7\ell_{\mathrm{x}}^2\ell_{\mathrm{y}} n^3\varrho^3 - 48c_6\ell_{\mathrm{x}}^2\ell_{\mathrm{y}} n^3\varrho^3 - 5c_7\ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3\varrho^3 - c_8 n_{\Omega_{\mathrm{y}}}\ell_{\mathrm{x}}^2 n^2\varrho^2 \\ & + c_9\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2\varrho^2 - c_{10}\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2\varrho^2 - 2c_2\ell_{\mathrm{x}}\ell_{\mathrm{y}}\left(1+6n\right) n^2\varrho^2, \end{aligned} \tag{4.16b}$$

$$c := 10c_7\ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3\varrho^3 - 32c_6\ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3\varrho^3, \tag{4.16c}$$

and:

$$\begin{aligned} d :=\ & 16c_6\ell_{\mathrm{x}}^3 n^3 n_{\Omega_{\mathrm{x}}}^{-2}\varrho^3 + 5c_7\ell_{\mathrm{x}}^3 n^3 n_{\Omega_{\mathrm{x}}}^{-1}\varrho^3 - 5c_7\ell_{\mathrm{x}}^3 n^3 n_{\Omega_{\mathrm{x}}}^{-2}\varrho^3 + c_8\ell_{\mathrm{x}}^2 n^2 n_{\Omega_{\mathrm{x}}}^{-1}\varrho^2 \\ & + c_9\ell_{\mathrm{x}}^2 n^2\varrho^2 - c_9\ell_{\mathrm{x}}^2 n^2 n_{\Omega_{\mathrm{x}}}^{-1}\varrho^2 - 2c_{10}\ell_{\mathrm{x}}^2 n^2\varrho^2 + c_{10}\ell_{\mathrm{x}}^2 n^2 n_{\Omega_{\mathrm{x}}}^{-1}\varrho^2 + c_{10}\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{x}}} n^2\varrho^2, \end{aligned} \tag{4.17a}$$

$$e := 5c_7\ell_{\mathrm{x}}\ell_{\mathrm{y}}^2 n^3\varrho^3 - 48c_6\ell_{\mathrm{x}}\ell_{\mathrm{y}}^2 n^3\varrho^3 - 5c_7\ell_{\mathrm{x}}\ell_{\mathrm{y}}^2 n_{\Omega_{\mathrm{x}}} n^3\varrho^3 - c_8 n_{\Omega_{\mathrm{x}}}\ell_{\mathrm{y}}^2 n^2\varrho^2, \tag{4.17b}$$

$$f := -32c_6\ell_{\mathrm{y}}^3 n_{\Omega_{\mathrm{x}}} n^3\varrho^3. \tag{4.17c}$$

For $\varrho > 0$, polynomials of the form (4.14) and (4.15) have a single global minimum, and hence the cost (4.11) is convex. ■

The proposed reduction in cost is proven in the following theorem:

Theorem 1. *For the optimal choice of $n_{\Omega_{\mathrm{x}}} = n_{\Omega_{\mathrm{x}},\mathrm{opt}}$ and $n_{\Omega_{\mathrm{y}}} = n_{\Omega_{\mathrm{y}},\mathrm{opt}}$, the cost of obtaining the overall frequency response using* (4.9) *and* (4.10) *reduces to $\mathcal{O}\left(\varrho^2\right)$.*

Proof. The cost of evaluating (4.9) for each molecule, and finally evaluating (4.10) is given by the expression (4.11). Since $n_{\mathrm{x}} = \varrho\ell_{\mathrm{x}}$ and $n_{\mathrm{y}} = \varrho\ell_{\mathrm{y}}$, for given ϱ, ℓ_{x} and ℓ_{y}, (4.11) can be written as a polynomial in ϱ:

$$\begin{aligned} \mathcal{C} =\ & c_1\ell_{\mathrm{x}}^3\ell_{\mathrm{y}}\left(1+n\right) n^2 n_{\Omega_{\mathrm{x}}}^{-2}\varrho^4 + \Big(48c_6\ell_{\mathrm{x}}^2\ell_{\mathrm{y}} n^3 n_{\Omega_{\mathrm{x}}}^{-1} + 48c_6\ell_{\mathrm{x}}\ell_{\mathrm{y}}^2 n^3 n_{\Omega_{\mathrm{y}}}^{-1} \\ & + 16c_6\ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3 n_{\Omega_{\mathrm{x}}}^{-2} + 10c_7\ell_{\mathrm{x}}^2\ell_{\mathrm{y}} n^3 + 16c_6\ell_{\mathrm{y}}^3 n_{\Omega_{\mathrm{x}}} n^3 n_{\Omega_{\mathrm{y}}}^{-2} - 10c_7\ell_{\mathrm{x}}^2\ell_{\mathrm{y}} n^3 n_{\Omega_{\mathrm{x}}}^{-1} \\ & - 5c_7\ell_{\mathrm{x}}\ell_{\mathrm{y}}^2 n^3 n_{\Omega_{\mathrm{y}}}^{-1} + 5c_7\ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3 n_{\Omega_{\mathrm{x}}}^{-1} + 5c_7\ell_{\mathrm{x}}\ell_{\mathrm{y}}^2 n_{\Omega_{\mathrm{x}}} n^3 n_{\Omega_{\mathrm{y}}}^{-1} - 5c_7\ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3 n_{\Omega_{\mathrm{x}}}^{-2}\Big)\varrho^3 \\ & + \Big(c_8 n_{\Omega_{\mathrm{y}}}\ell_{\mathrm{x}}^2 n^2 n_{\Omega_{\mathrm{x}}}^{-1} + 2c_8\ell_{\mathrm{x}}\ell_{\mathrm{y}} n^2 + c_8 n_{\Omega_{\mathrm{x}}}\ell_{\mathrm{y}}^2 n^2 n_{\Omega_{\mathrm{y}}}^{-1} + c_9\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 - c_9\ell_{\mathrm{x}}\ell_{\mathrm{y}} n^2 \\ & - c_9\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 n_{\Omega_{\mathrm{x}}}^{-1} + c_9\ell_{\mathrm{x}}\ell_{\mathrm{y}} n_{\Omega_{\mathrm{x}}} n^2 + c_{10}\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 n_{\Omega_{\mathrm{x}}}^{-1} - 2c_{10}\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 + 32c_3\ell_{\mathrm{x}}\ell_{\mathrm{y}} n^3 \\ & + c_{10}\ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{x}}} n_{\Omega_{\mathrm{y}}} n^2 + 2c_2\ell_{\mathrm{x}}\ell_{\mathrm{y}}\left(1+6n\right) n^2 n_{\Omega_{\mathrm{x}}}^{-1} + 4c_4\ell_{\mathrm{x}}\ell_{\mathrm{y}} n^2 + 8c_5\ell_{\mathrm{x}}\ell_{\mathrm{y}} n\Big)\varrho^2, \end{aligned} \tag{4.18}$$

which is of the form:

$$\mathcal{C} = \varrho^2\left(a\varrho^2 + b\varrho + c\right), \tag{4.19}$$

subject to the constraint that $\mathcal{C} \geq 0 \; \forall \; \varrho \geq 0$, where:

$$a := c_1 \ell_{\mathrm{x}}^3 \ell_{\mathrm{y}} \left(1+n\right) n^2 n_{\Omega_{\mathrm{x}}}^{-2}, \tag{4.20a}$$

$$\begin{aligned} b := {} & 48 c_6 \ell_{\mathrm{x}}^2 \ell_{\mathrm{y}} n^3 n_{\Omega_{\mathrm{x}}}^{-1} + 48 c_6 \ell_{\mathrm{x}} \ell_{\mathrm{y}}^2 n^3 n_{\Omega_{\mathrm{y}}}^{-1} + 16 c_6 \ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3 n_{\Omega_{\mathrm{x}}}^{-2} + 10 c_7 \ell_{\mathrm{x}}^2 \ell_{\mathrm{y}} n^3 \\ & + 16 c_6 \ell_{\mathrm{y}}^3 n_{\Omega_{\mathrm{x}}} n^3 n_{\Omega_{\mathrm{y}}}^{-2} - 10 c_7 \ell_{\mathrm{x}}^2 \ell_{\mathrm{y}} n^3 n_{\Omega_{\mathrm{x}}}^{-1} - 5 c_7 \ell_{\mathrm{x}} \ell_{\mathrm{y}}^2 n^3 n_{\Omega_{\mathrm{y}}}^{-1} \\ & + 5 c_7 \ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3 n_{\Omega_{\mathrm{x}}}^{-1} + 5 c_7 \ell_{\mathrm{x}} \ell_{\mathrm{y}}^2 n_{\Omega_{\mathrm{x}}} n^3 n_{\Omega_{\mathrm{y}}}^{-1} - 5 c_7 \ell_{\mathrm{x}}^3 n_{\Omega_{\mathrm{y}}} n^3 n_{\Omega_{\mathrm{x}}}^{-2}, \end{aligned} \tag{4.20b}$$

$$\begin{aligned} c := {} & c_8 n_{\Omega_{\mathrm{y}}} \ell_{\mathrm{x}}^2 n^2 n_{\Omega_{\mathrm{x}}}^{-1} + 2 c_8 \ell_{\mathrm{x}} \ell_{\mathrm{y}} n^2 + c_8 n_{\Omega_{\mathrm{x}}} \ell_{\mathrm{y}}^2 n^2 n_{\Omega_{\mathrm{y}}}^{-1} + c_9 \ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 - c_9 \ell_{\mathrm{x}} \ell_{\mathrm{y}} n^2 \\ & - c_9 \ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 n_{\Omega_{\mathrm{x}}}^{-1} + c_9 \ell_{\mathrm{x}} \ell_{\mathrm{y}} n_{\Omega_{\mathrm{x}}} n^2 + c_{10} \ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 n_{\Omega_{\mathrm{x}}}^{-1} - 2 c_{10} \ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{y}}} n^2 \\ & + 32 c_3 \ell_{\mathrm{x}} \ell_{\mathrm{y}} n^3 + c_{10} \ell_{\mathrm{x}}^2 n_{\Omega_{\mathrm{x}}} n_{\Omega_{\mathrm{y}}} n^2 + 2 c_2 \ell_{\mathrm{x}} \ell_{\mathrm{y}} \left(1+6n\right) n^2 n_{\Omega_{\mathrm{x}}}^{-1} + 4 c_4 \ell_{\mathrm{x}} \ell_{\mathrm{y}} n^2 \\ & + 8 c_5 \ell_{\mathrm{x}} \ell_{\mathrm{y}} n, \end{aligned} \tag{4.20c}$$

and $c_1, ..., c_{10} \in \mathbb{R}^+$ are unknown constants.

The minimum cost is achieved when:

$$\frac{\partial \mathcal{C}}{\partial \varrho} = \varrho \left(4a\varrho^2 + 3b\varrho + 2c\right) = 0, \tag{4.21}$$

which has solutions:

$$\varrho_1 = -\frac{3b}{8a} + \frac{\sqrt{9b^2 - 32ac}}{8a}, \tag{4.22a}$$

$$\varrho_2 = -\frac{3b}{8a} - \frac{\sqrt{9b^2 - 32ac}}{8a}, \tag{4.22b}$$

$$\varrho_3 = 0. \tag{4.22c}$$

Since:

$$\frac{\partial^2 \mathcal{C}}{\partial \varrho^2} = 12a\varrho^2 + 6b\varrho + 2c, \tag{4.23}$$

and $\mathcal{C}$ must be increasing when $\varrho = 0$,

$$\begin{aligned} \frac{\partial^2 \mathcal{C}(0)}{\partial \varrho^2} &= 2c > 0 \\ \therefore c &> 0. \end{aligned} \tag{4.24}$$

Given $a, c > 0$, for the non-trivial roots of (4.21) to be physical, i.e. $\varrho_1, \varrho_2 > 0$, the following must be true:

$$b < 0. \tag{4.25}$$

Analysis of (4.23), evaluated at the non-trivial roots, shows that the local minimum lies at $\varrho_{\mathrm{opt}} = \varrho_1$.

Substituting ϱ_{opt} into (4.19) yields:

$$\begin{aligned}\mathcal{C}\left(\varrho_{\text{opt}}\right) &= \frac{9b^2c}{32a^2} - \frac{c^2}{4a} - \frac{27b^4}{512a^3} + \frac{9b^3\sqrt{9b^2-32ac}}{512a^3} - \frac{2bc\sqrt{9b^2-32ac}}{32a^2} \\ &= c\varrho_{\text{opt}}^2 + \frac{9b^3}{64a^2}\varrho_{\text{opt}} + \frac{c}{4}\left(\frac{c}{a} + \frac{b\sqrt{9b^2-32ac}}{8a^2}\right) \\ &= \check{a}\varrho_{\text{opt}}^2 + \check{b}\varrho_{\text{opt}} + \check{c}, \end{aligned} \tag{4.26}$$

where:

$$\check{a} := c, \tag{4.27a}$$

$$\check{b} := \frac{9b^3}{64a^2}, \tag{4.27b}$$

$$\check{c} := \frac{c}{4}\left(\frac{c}{a} + \frac{b\sqrt{9b^2-32ac}}{8a^2}\right). \tag{4.27c}$$

Therefore for $\varrho = \varrho_{\text{opt}}$, the cost (4.19) reduces to (4.26), which is $\mathcal{O}\left(\varrho^2\right)$ to leading order. ■

4.3 Application to 2D wave-diffusion equation

In order to demonstrate the efficacy of the approach outlined above, it is applied to a 2D wave-diffusion equation example, and compared to a full-order state-space representation obtained by directly spatially discretising the governing equation.

The governing equation reads:

$$\frac{\partial^2\varphi(\mathbf{x},t)}{\partial t^2} = c^2\left(1 + k\frac{\partial}{\partial t}\right)\varphi(\mathbf{x},t),\ \forall\,(\mathbf{x},t) \in \Omega \times [0,t_f]\,, \tag{4.28}$$

where $\varphi(\cdot,\cdot) : \Omega \times [0,t_f] \to \mathbb{R}$ is the height of the surface, $c,k \in \mathbb{R}^+$ are constants which dictate the wave propogation speed and rate of diffusion, respectively, $\Omega := [-2,2]\times[-1,1] \subset \mathbb{R}^2$ is a rectangular spatial domain with boundary $\partial\Omega$, $t_f \in \mathbb{R}^+$ is the endpoint of the time inverval, and $\mathbf{x} \in \Omega$ is a point in the domain. Periodic boundary conditions are assumed on all boundaries, i.e.:

$$\varphi(2,\mathsf{y},t) = \varphi(-2,\mathsf{y},t),\ \forall\,(\mathsf{y},t) \in [-1,1]\times[0,t_f]\,, \tag{4.29a}$$

$$\varphi(\mathsf{x},1,t) = \varphi(\mathsf{x},-1,t),\ \forall\,(\mathsf{x},t) \in [-2,2]\times[0,t_f]\,, \tag{4.29b}$$

actuation takes the form of direct control of the surface height at $(\mathsf{x}, \mathsf{y}) = (1.5, 0)$:

$$u(t) = \varphi(1.5, 0, t), \tag{4.30}$$

and sensing is a measurement of the surface height in the centre of the domain:

$$y(t) = \varphi(0, 0, t). \tag{4.31}$$

4.3.1 Full-order model

In order to derive a full-order model, the spatial domain was first discretised on a uniform computational mesh with mesh density $\varrho \in \mathbb{N}$ in both the x and y directions. As such, there were $n_{\mathsf{x}} = 4(\varrho - 1) + 1$ and $n_{\mathsf{y}} = 2(\varrho - 1) + 1$ computational nodes in the x and y directions, respectively, yielding a total of $n_{\text{total}} = n_{\mathsf{x}} n_{\mathsf{y}}$ nodes.

Second-order spatial derivatives in both directions were implemented using second-order accurate centred finite-differences. Finite-difference matrices were constructed as:

$$\mathcal{D}_{2,\mathsf{x}} := \frac{1}{\delta_{\mathsf{x}}^2} \begin{bmatrix} -2 & 1 & 0 & & \cdots & & 0 \\ 1 & -2 & 1 & & & & \\ 0 & 1 & -2 & 1 & & & \vdots \\ & & & \ddots & & & \\ \vdots & & & 1 & -2 & 1 & 0 \\ & & & & 1 & -2 & 1 \\ 0 & & \cdots & & 0 & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n_{\mathsf{x}} \times n_{\mathsf{x}}}, \tag{4.32a}$$

$$\mathcal{D}_{2,\mathsf{y}} := \frac{1}{\delta_{\mathsf{y}}^2} \begin{bmatrix} -2 & 1 & 0 & & \cdots & & 0 \\ 1 & -2 & 1 & & & & \\ 0 & 1 & -2 & 1 & & & \vdots \\ & & & \ddots & & & \\ \vdots & & & 1 & -2 & 1 & 0 \\ & & & & 1 & -2 & 1 \\ 0 & & \cdots & & 0 & 1 & -2 \end{bmatrix} \in \mathbb{R}^{n_{\mathsf{y}} \times n_{\mathsf{y}}}, \tag{4.32b}$$

where $\delta_{\mathsf{x}} = \delta_{\mathsf{y}} = \delta$ is the (uniform) mesh spacing in both the x and y directions. Periodic boundary conditions were implemented by simply setting the $(1, n_{\mathsf{x}})$ and $(n_{\mathsf{x}}, 1)$ elements of $\mathcal{D}_{2,\mathsf{x}}$ equal to one, and similarly for $\mathcal{D}_{2,\mathsf{y}}$; i.e. making the matrices circulant [42].

The Laplacian operator was approximated as [94]:

$$\nabla^2 \approx \mathfrak{L} := I_{n_\mathsf{x}} \otimes \mathcal{D}_{2,\mathsf{y}} + \mathcal{D}_{2,\mathsf{x}} \otimes I_{n_\mathsf{y}} \in \mathbb{R}^{n_{\text{total}} \times n_{\text{total}}}, \tag{4.33}$$

where $\cdot \otimes \cdot : \mathbb{R}^{n_1 \times n_1} \times \mathbb{R}^{n_3 \times n_4} \to \mathbb{R}^{n_1 n_3 \times n_2 n_4}$ is the Kronecker product.

The overall system can be represented by the following descriptor state-space system:

$$\underbrace{\begin{bmatrix} E_{11} & 0 \\ 0 & I_{n_{\text{total}}} \end{bmatrix}}_{E} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \vec{\varphi}(t) \\ \dot{\vec{\varphi}}(t) \end{bmatrix} = \underbrace{\begin{bmatrix} A_{11} & A_{12} \\ c^2\mathfrak{L} & c^2 k \mathfrak{L} \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} \vec{\varphi}(t) \\ \dot{\vec{\varphi}}(t) \end{bmatrix}}_{\mathbf{x}(t)} + Bu(t), \tag{4.34a}$$

$$y(t) = C\mathbf{x}(t), \tag{4.34b}$$

where $\vec{\varphi}(t) \in \mathbb{R}^{n_{\text{total}}}$ is a vector of the values of $\varphi(\mathbf{x}, t)$ at the computational nodes, $\mathbf{x}(t) \in \mathbb{R}^{2n_{\text{total}}}$ is the state vector, $E_{11}, A_{12} \in \mathbb{R}^{n_{\text{total}} \times n_{\text{total}}}$ are identity matrices, except that the element on the diagonal corresponding to the $\varphi(t)$ state in the actuation location has been set equal to 0, $A_{11} \in \mathbb{R}^{n_{\text{total}} \times n_{\text{total}}}$ is a matrix of zeros except for the element on the diagonal corresponding to the $\varphi(t)$ state in the actuation location, which is set equal to 1, $E, A \in \mathbb{R}^{2n_{\text{total}} \times 2n_{\text{total}}}$, $B \in \mathbb{R}^{2n_{\text{total}} \times 1}$ is a matrix of zeros except for the row corresponding to the $\varphi(t)$ state in the actuation location, which is set equal to -1, and $C \in \mathbb{R}^{1 \times 2n_{\text{total}}}$ is a matrix of zeros except for the column corresponding to the $\varphi(t)$ state at the sensor location, which is set equal to 1.

Whilst this full-order model is a dynamically correct representation of the wave-diffusion system, the system matrices quickly become large and ill-conditioned as the computational mesh density is increased. The frequency response of (4.34) can be computed for frequencies of interest $\omega \in \mathbb{R}$ as:

$$G(\mathrm{i}\omega) = C\left(\mathrm{i}\omega E - A\right)^{-1} B. \tag{4.35}$$

Inverting the matrix $(\mathrm{i}\omega E - A) \in \mathbb{C}^{n_{\text{total}} n \times n_{\text{total}} n}$ here is particularly costly as $n_{\text{total}} = n_\mathsf{x} \times n_\mathsf{y} = \varrho\ell_\mathsf{x} \times \varrho\ell_\mathsf{y}$, and so the cost of this operation is $\mathcal{O}\left(\varrho^6\right)$. Whilst this is tractable for the simple wave-diffusion equation example on a relatively coarse computational mesh, and serves as a means of providing a benchmark model, it would be infeasible for a typical complex geometry fluid dynamics modelling problem.

4.3.2 Obtaining frequency response using RHSP-based construction method

Discretising (4.28) on the same compututational mesh as the full-order model, again using second-order accurate centred finite-differences yields the following state-space representation for each individual computational atom $P_{i,j}$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\varphi_{i,j}(t)\\ \dot{\varphi}_{i,j}(t)\end{bmatrix} = \underbrace{\begin{bmatrix}0 & 1\\ -\frac{4c^2}{\delta^2} & -\frac{4c^2k}{\delta^2}\end{bmatrix}}_{A_{i,j}}\underbrace{\begin{bmatrix}\varphi_{i,j}(t)\\ \dot{\varphi}_{i,j}(t)\end{bmatrix}}_{\mathbf{x}_{i,j}(t)} + \underbrace{\begin{bmatrix}\check{B}_{i,j} & \check{B}_{i,j} & \check{B}_{i,j} & \check{B}_{i,j}\end{bmatrix}}_{B_{i,j}}\boldsymbol{\xi}_{i,j}(t), \tag{4.36a}$$

$$\mathbf{z}_{i,j}(t) = \underbrace{\begin{bmatrix}I & I & I & I\end{bmatrix}^{\top}}_{C_{i,j}}\mathbf{x}_{i,j}(t), \tag{4.36b}$$

where $\varphi_{i,j}(t) \in \mathbb{R}$ is the value of $\varphi(\mathbf{x}, t)$ at the location of the subsystem, $\boldsymbol{\xi}_{i,j}(t) \in \mathbb{R}^8$ and $\mathbf{z}_{i,j}(t) \in \mathbb{R}^8$ are defined as in Section 4.1.1,

$$\check{B}_{i,j} := \begin{bmatrix}0 & 0\\ \frac{c^2}{\delta^2} & \frac{c^2k}{\delta^2}\end{bmatrix},$$

$A_{i,j} \in \mathbb{R}^{2\times 2}$, $B_{i,j} \in \mathbb{R}^{2\times 8}$, and $C_{i,j} \in \mathbb{R}^{8\times 2}$. The atom has corresponding frequency response:

$$P_{i,j}(\mathrm{i}\omega) = C_{i,j}\left(\mathrm{i}\omega I - A_{i,j}\right)^{-1} B_{i,j} \in \mathbb{C}^{8\times 8}. \tag{4.37}$$

The atoms describing boundary nodes, the node with actuation, and the node with sensing are altered as required.

The overall system frequency response $G(\mathrm{i}\omega)$ was obtained using the RHSP-based construction method that was outlined above, both with and without the use of domain decomposition optimisation. The results are presented in Figure 4.5 for mesh density $\varrho = 25$ and physical parameters chosen arbitrarily as $c = 0.5$ and $k = 0.1$. The frequency response obtained from the full-order model is plotted for comparison.

The RHSP-based construction method produces identical frequency response data to those obtained from the full-order model, both with and without the domain decomposition optimisation, hence confirming the modelling approach's validity.

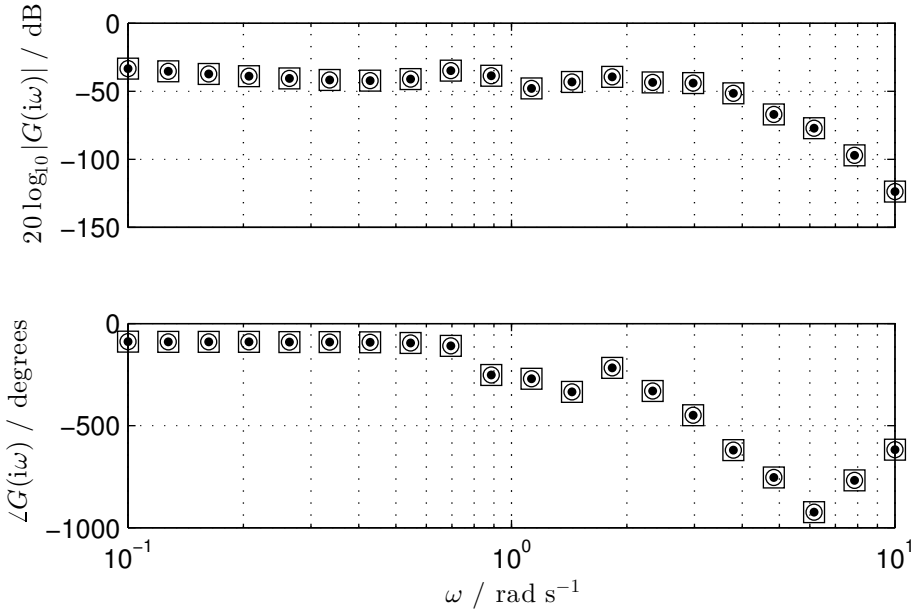


Figure 4.5: Frequency response of 2D wave-diffusion system: from full-order system (•), using RHSP-based construction method without domain decomposition optimisation (◯), using RHSP-based construction method with domain decomposition optimisation (□).

4.3.3 Computational aspects of RHSP-based construction method

The modelling exercise described above was repeated for a single frequency on increasingly fine computational meshes in order to analyse some computational aspects of the RHSP-based construction method. Firstly, Figure 4.6 shows the (log of) wall-clock time T (seconds) required to compute $G(\mathrm{i}\omega)$ for a single frequency[3] as a function of $\log_{10}\varrho$. It should be noted that the frequency response could not be computed from a full-order state-space system model for mesh resolutions as high as those considered for the RHSP-based construction method as the memory requirements quickly became too high.

Since the computational cost is quantified in big-O notation as $\mathcal{O}\left(\varrho^b\right)$ for some $b \in \mathbb{R}^+$, it is fair to assume that for large ϱ:

$$T = a\varrho^b,$$
$$\therefore\ \log_{10} T = \log_{10} a + b\log_{10}\varrho,$$

where $a \in \mathbb{R}^+$ is some unknown constant.

The gradients of the plots in Figure 4.6 confirm that the computational cost of

[3]All computations were carried out using IEEE standard 754 double-precision floating point arithmetic on a 3.40GHz Intel Core i7 (quad core) machine with relative machine precision $\varepsilon = 2.2 \times 10^{-16}$.

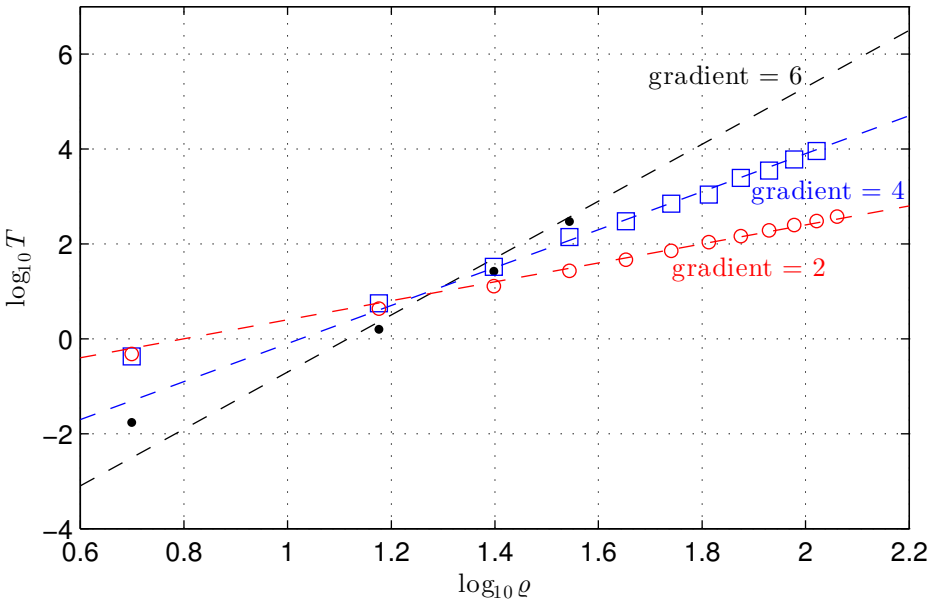


Figure 4.6: log of wall-clock time T (seconds) required to obtain $G(\mathrm{i}\omega)$ for a single frequency as a function of $\log_{10} \varrho$: from full-order state-space system (•), using RHSP-based construction method without domain decomposition optimisation (□), using RHSP-based construction method with domain decomposition optimisation (◯).

obtaining the frequency response of a two spatial dimension PDAE system from the full-order state-space system, via the RHSP-based construction method, and via the RHSP-based construction method with domain decomposition optimisation are $\mathcal{O}\left(\varrho^6\right)$, $\mathcal{O}\left(\varrho^4\right)$, and $\mathcal{O}\left(\varrho^2\right)$, respectively, as predicted in Sections 4.1.3–4.3.1.

Other factors which should be considered when analysing numerical modelling methods are the memory requirements of the approach, and the numerical conditioning of the algorithms used within the method. These factors were analysed by considering the memory $\mathcal{M} \in \mathbb{N}$ (bytes) required to store the largest matrix which was constructed during computation, and the condition number $\kappa_p \in \mathbb{R}^+$ of the most ill-conditioned matrix which required inversion during computation, where the condition number of a matrix A is defined as [42]:

$$\kappa_p(A) := \|A\|_p \cdot \left\|A^{-1}\right\|_p, \tag{4.38}$$

for some appropriate norm $\|\cdot\|_p$, and indicates how likely numerical errors are when computing A^{-1}. The values of $\mathcal{M}$ and κ_2 are plotted for increasing values of ϱ in Figures 4.7(a) and 4.7(b), respectively. Both $\mathcal{M}$ and κ_2 are orders of magnitude smaller for all ϱ considered when using the RHSP-based construction approach with domain decomposition optimisation than computing the frequency response from the full order state-space system.

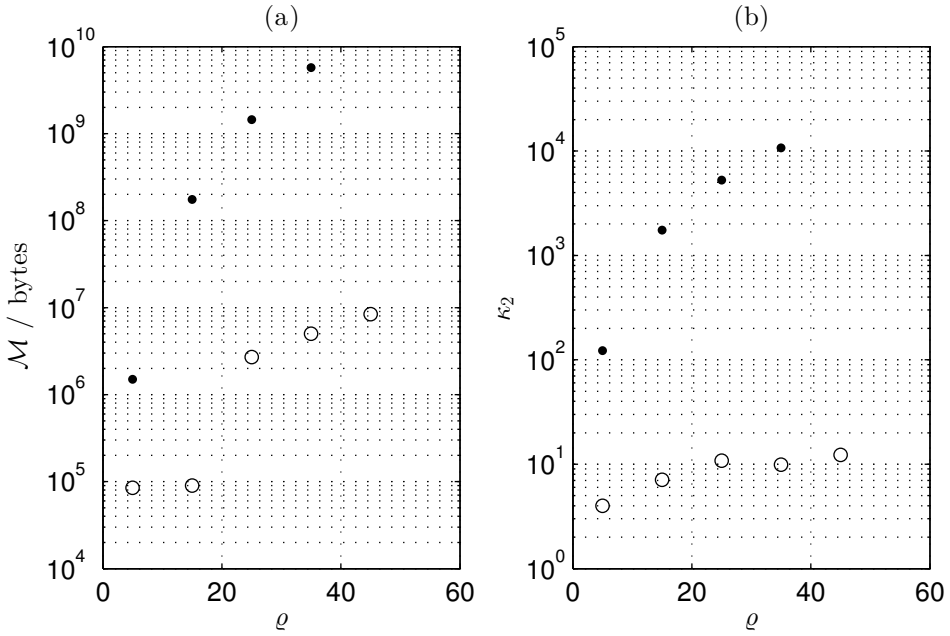


Figure 4.7: (a) memory $\mathcal{M}$ required to store largest matrix constructed during computation of frequency response; (b) 2-norm condition number κ_2 of most ill-conditioned matrix which required inversion during computation of frequency response: from full-order state-space system (•), using RHSP-based construction method with domain decomposition optimisation (◯).

Of course, one could improve on the performance of the simplest approach (directly computing the frequency response from the full-order state-space system) by using sparse matrices and methods [31]. Numerical tests revealed a cost of $\sim \mathcal{O}\left(\varrho^{2.4}\right)$, but this has to be balanced by the difficulty in forming the full-scale system matrices in the first place, and still suffers from large condition numbers of system matrices.

4.4 Summary

In this chapter, the low-order modelling approach used in this work has been presented. Step-by-step details of obtaining PDAE system input-output frequency response were given, including a description of the important numerical operation – the Redheffer star product – which lies at the heart of the method.

For the two spatial dimension case, it was shown that the computational cost of the method is $\mathcal{O}\left(\varrho^4\right)$ (whilst obtaining the same frequency response by directly computing $C(\mathrm{i}\omega E - A)^{-1}B$ of the equivalent full-order state-space system costs $\mathcal{O}\left(\varrho^6\right)$). By introducing a domain decomposition optimisation, however, it was proven that by choosing the optimal number of subdomains in each spatial direction, the cost reduces to $\mathcal{O}\left(\varrho^2\right)$.

The validity of the modelling approach was demonstrated by application to the 2D wave-diffusion equation, and the resulting frequency response was in exact agreement with that obtained from a full-order state-space model. Results were presented which showed the computational complexity scaled as predicted, and that the modelling methodology used in this work was better conditioned and required less memory than a traditional approach.

Chapter 5

Discretising the Navier-Stokes equations

Whilst the previous chapter demonstrated that the RHSP-based construction method correctly produces the frequency response of systems governed by linear PDAEs, the correct choice of formulation and discretisation of the Navier-Stokes equations is not straightforward. This chapter considers some of the issues associated with the governing equations, and investigates the effect on system dynamics of basing plant models on different formulations of the linearised equations.

Firstly, the dynamics of an individual computational node subsystem formed by spatial discretisation of the linearised governing equations in both *primitive variables* (momentum and continuity equations) and *pressure Poisson equation* (PPE) formulations are considered. This reveals fundamental numerical differences at the nodal level, whose effects on the linearised system dynamics at the full system level are then exemplified by considering the corresponding formulations of a 2D channel flow, subjected to a variety of different boundary conditions.

5.1 Issues with the Navier-Stokes equations

As was discussed in Chapter 4, for the purpose of the RHSP-based construction method, second-order accurate centred finite-difference discretisation of the governing equations is attractive as it results in simple connections between neighbouring subsystems. It is well known, however, that directly discretising the Navier-Stokes equations in their standard primitive variables formulation (1.1) on a co-located mesh using finite-differences can lead to the so-called 'checkerboard instability', whereby non-physical sawtooth shaped pressure fields can satisfy the discretised equations [37, 77, 24, 36]. Figure 5.1

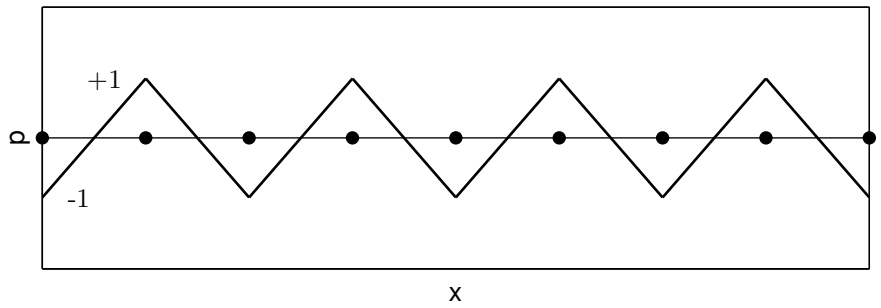


Figure 5.1: Example of checkerboard instability; non-physical pressure field satisfying discretised incompressible Navier-Stokes equations (discretised using centred finite-differences), since $\partial \mathsf{p}/\partial \mathsf{x} = 0$ at all computational nodes (black dots).

shows an example of such a pressure field.

The checkerboard instability issue is a result of the fact that after any spatial discretisation the incompressible Navier-Stokes equations constitute a system of differential algebraic equations (DAEs) of 'higher index' [101], with only the pressure gradient appearing in the equations, and not pressure itself. The nonlinear system is of differentiation index two [16], or strangeness index one [101, 62], since at least part of the equations must be differentiated a minimum of two times with respect to the time variable in order to obtain a system of ordinary differential equations (ODEs). One way of circumventing the checkerboard instability issue is by reformulating the set of equations as a strangeness free system [101]. This can be achieved by reformulating the system in, for example, velocity and vorticity variables, which results in a system of ODEs (of differentiation index zero [101]), or supplementing the original equations with a PPE which results in a strangeness free DAE system (of differentiation index one).

Whilst supplementing the original system of equations with the PPE avoids the checkerboard instability, it poses the additional problem of choosing a correct pressure boundary condition. For the case of the nonlinear incompressible Navier-Stokes equations, there seems to be some disagreement in the literature regarding this matter [44, 43, 87, 84, 91, 100], and so several different boundary conditions were considered in this work, and their effects on system dynamics were studied.

Another way of circumventing the issue of checkerboard instability, which avoids the need to reformulate the governing equations as a strangeness free system, is by discretising the original formulation of the equations on a staggered mesh, such as the 'marker and cell' method [45]. This method does, however, come with its own difficulties. Namely that one typically obtains discretisations where pressure nodes do not lie on the domain boundaries, which is inconvenient from a control design perspective if pressure on a surface is the measured signal, as may be the case for bluff-body drag

reduction.

For the purpose of simulation, the various formulations of the Navier-Stokes equations have received much attention in the literature, however for the purpose of feedback control design this is not the case. In particular, for simulation purposes the equations can be formulated in such a way that a 'pseudopressure' which requires only simple boundary conditions is used rather than actual thermodynamic pressure, for example in fractional step/projection methods [93, 84]. This allows efficient simulation by first computing an intermediate velocity field at each timestep which does not, in general, satisfy the divergence-free requirement of the governing equations, before correcting this velocity field using the pseudopressure, yielding a divergence-free field [84]. In the context of simulation, the validity of numerical schemes is typically assessed by comparing certain time averaged properties, such as mean velocity profiles, to benchmark data from other simulations or experiments (see, e.g. [21, 80, 81, 28, 74]). On the other hand, models derived for control design must correctly capture the continuous time dynamics of both velocity and pressure fields in a single set of differential algebraic equations, and in particular must exhibit the correct frequency response in the frequency band of interest.

5.2 Linearised Navier-Stokes equations

Since this work assumes that the use of linear systems theory is justified, the linearised incompressible Navier-Stokes equations were considered. By decomposing each flow variable into a temporally averaged mean part and a time dependent fluctuating part as such:

$$\begin{bmatrix} \mathbf{u}(\mathbf{x},t) \\ \mathsf{p}(\mathbf{x},t) \end{bmatrix} := \underbrace{\begin{bmatrix} \bar{\mathbf{u}}(\mathbf{x}) \\ \bar{\mathsf{p}}(\mathbf{x}) \end{bmatrix}}_{\text{mean part}} + \underbrace{\begin{bmatrix} \mathbf{u}'(\mathbf{x},t) \\ \mathsf{p}'(\mathbf{x},t) \end{bmatrix}}_{\text{fluctuating part}}, \tag{5.1}$$

substituting these into the original nonlinear equations (1.1), and finally neglecting second-order in fluctuation terms, one obtains the linearised Navier-Stokes equations:

$$\begin{aligned} \frac{\partial \mathbf{u}'(\mathbf{x},t)}{\partial t} + \mathbf{u}'(\mathbf{x},t)\cdot\nabla\bar{\mathbf{u}}(\mathbf{x}) + \bar{\mathbf{u}}(\mathbf{x})\cdot\nabla\mathbf{u}'(\mathbf{x},t) = \\ -\nabla\mathsf{p}'(\mathbf{x},t) + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}'(\mathbf{x},t),\ \forall\,(\mathbf{x},t)\in\Omega\times[0,t_f]\,, \end{aligned} \tag{5.2a}$$

$$\nabla\cdot\mathbf{u}'(\mathbf{x},t) = 0,\ \forall\,(\mathbf{x},t)\in\Omega\times[0,t_f]\,. \tag{5.2b}$$

The remainder of this chapter considers 2D flows, i.e. $\Omega\subset\mathbb{R}^2$, $\mathbf{x} := (\mathsf{x},\mathsf{y})\in\Omega$, $\mathbf{u}' := (\mathsf{u}',\mathsf{v}')\in\mathbb{R}^2$, and $\bar{\mathbf{u}} := (\bar{\mathsf{u}},\bar{\mathsf{v}})\in\mathbb{R}^2$, but the theory readily extends to the 3D case.

Two formulations of the linearised Navier-Stokes equations were considered in this work. The first of these, the primitive variables formulation, is given by the momentum (5.2a) and continuity (5.2b) equations [84]. The presence of the algebraic constraint in the form of the continuity equation makes this a set of PDAEs. As was discussed in Section 5.1, this PDAE system of differentiation index two can be reformulated as a strangeness-free system by introducing the PPE. The linearised PPE is obtained by substituting (5.2b) into the divergence of (5.2a) [77], yielding:

$$\begin{aligned}\nabla^2 \mathsf{p}'(\mathbf{x},t) = &-\nabla\cdot\left(\mathbf{u}'(\mathbf{x},t)\cdot\nabla\bar{\mathbf{u}}(\mathbf{x})\right)\\ &-\nabla\cdot\left(\bar{\mathbf{u}}(\mathbf{x})\cdot\nabla\mathbf{u}'(\mathbf{x},t)\right),\ \forall\,(\mathbf{x},t)\in\Omega\times[0,t_f]\,.\end{aligned} \tag{5.2c}$$

The system of equations comprising (5.2a) and (5.2c) is known as the PPE formulation [84] and represents the second formulation employed in this work.

Boundary and initial conditions for the primitive variables formulation are given by:

$$\bar{\mathbf{u}}(\mathbf{x}) = \bar{\mathbf{u}}_{\partial\Omega}(\mathbf{x}),\ \forall\,\mathbf{x}\in\partial\Omega, \tag{5.3a}$$

$$\mathbf{u}'(\mathbf{x},t) = \mathbf{u}'_{\partial\Omega}(\mathbf{x},t),\ \forall\,(\mathbf{x},t)\in\partial\Omega\times[0,t_f]\,, \tag{5.3b}$$

$$\mathbf{u}'(\mathbf{x},0) = \mathbf{u}'_0(\mathbf{x}),\ \forall\,\mathbf{x}\in\Omega. \tag{5.3c}$$

Discussion of the boundary conditions for the PPE formulation is deferred until Section 5.4.

5.3 Individual computational node subsystem

Several formulations of the linearised Navier-Stokes equations were considered when obtaining individual computational node subsystems: the primitive variables formulation was discretised on both a co-located mesh and a staggered mesh, and the PPE formulation was discretised on a co-located mesh only. The co-located mesh, in which both velocity and pressure values are stored on the cell vertices, is depicted in Figure 5.2(a). The staggered mesh is shown in Figure 5.2(b), in which the velocity components u and v are stored on the vertical and horizontal cell boundaries, respectively, whilst the pressure is stored in the cell centre.

State-space models of the following form are considered:

$$E\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = A\mathbf{x}(t) + B\boldsymbol{\xi}(t), \tag{5.4a}$$

$$\mathbf{z}(t) = C\mathbf{x}(t), \tag{5.4b}$$

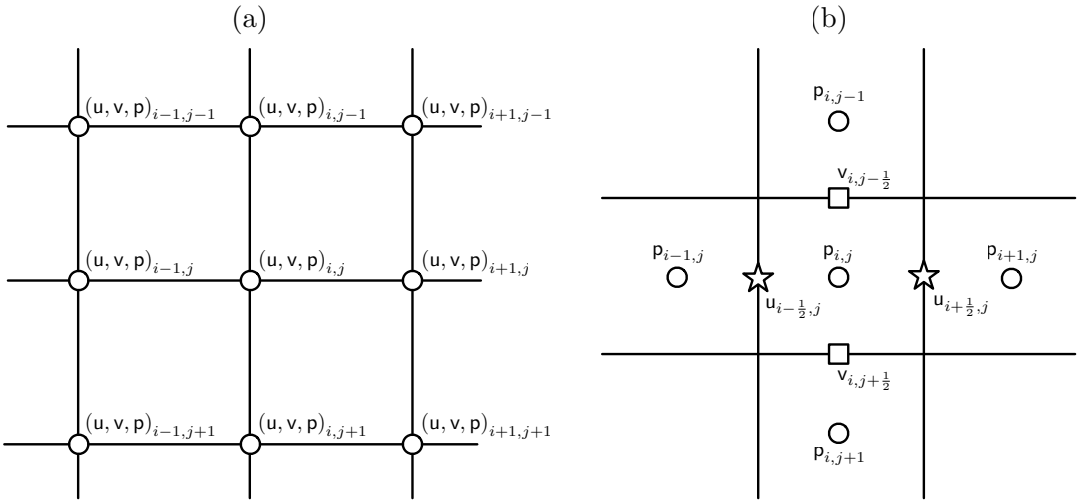


Figure 5.2: Computational meshes: (a) co-located computational mesh; (b) staggered mesh.

where $\mathbf{x}(t) \in \mathbb{R}^n$, $\mathbf{z}(t) \in \mathbb{R}^p$, $\boldsymbol{\xi}(t) \in \mathbb{R}^q$, $E, A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times q}$, and $C \in \mathbb{R}^{p\times n}$, with corresponding transfer functions:

$$G(s) = C\left(sE - A\right)^{-1} B \in \mathscr{R}^{p\times q}. \tag{5.5}$$

The poles of a linear dynamical system determine its asymptotic behaviour and are defined as:

$$\mathcal{P}\left(G(s)\right) := \left\{s \in \mathbb{C} : \det\left(sE - A\right) = 0\right\}. \tag{5.6}$$

The zeros are the values of $s \in \mathbb{C}$ at which the system output drops to zero despite the input and states being non-zero [32, 92]. The set of zeros is thus defined as:

$$\mathcal{Z}\left(G(s)\right) := \left\{s \in \mathbb{C} : \mathbf{y}(t) = 0,\ \boldsymbol{\xi}(t), \mathbf{x}(t) \neq 0\right\}. \tag{5.7}$$

The presence and locations of the zeros are dependent on sensor and actuation arrangement, and can impose fundamental limitations on the performance achievable through feedback control [92]. For single-input single-output systems, the poles and zeros are simply equal to the roots of the denominator and numerator of the transfer function, respectively.

The different formulations of the governing equations were discretised in space using second-order accurate centred finite-differences on computational meshes with uniform grid spacing δ in both the x and y directions.

For the case of co-located mesh discretisations, $\mathsf{u}'_{i,j}(t)$, $\mathsf{v}'_{i,j}(t)$, and $\mathsf{p}'_{i,j}(t)$ are grouped

as a single subsystem, and are referred to as the $(i,j)^{\text{th}}$ node, whilst for staggered mesh discretisations, $\mathsf{u}'_{i+1/2,j}(t)$, $\mathsf{v}'_{i,j+1/2}(t)$, and $\mathsf{p}'_{i,j}(t)$ are grouped together as a single subsystem, and are also referred to as the $(i,j)^{\text{th}}$ node (where the indices of the pressure node are used to label the subsystem).

The dynamics of the $(i,j)^{\text{th}}$ computational node subsystem can therefore be described by the following state-space model:

$$E_{i,j}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}_{i,j}(t) = A_{i,j}\mathbf{x}_{i,j}(t) + B_{i,j}\boldsymbol{\xi}_{i,j}(t), \tag{5.8a}$$

$$\mathbf{z}_{i,j}(t) = C_{i,j}\mathbf{x}_{i,j}(t), \tag{5.8b}$$

where $\mathbf{x}_{i,j} := \begin{bmatrix}\mathsf{u}'_{i,j} & \mathsf{v}'_{i,j} & \mathsf{p}'_{i,j}\end{bmatrix}^\top \in \mathbb{R}^3$ is the subsystem's state vector in the co-located case, and $\mathbf{x}_{i,j} := \begin{bmatrix}\mathsf{u}'_{i+1/2,j} & \mathsf{v}'_{i,j+1/2} & \mathsf{p}'_{i,j}\end{bmatrix}^\top \in \mathbb{R}^3$ is the subsystem's state vector in the staggered case. The vector of state values flowing in from the neighbouring subsystems is $\boldsymbol{\xi}_{i,j} := \begin{bmatrix}\mathbf{x}_{i,j-1}^\top & \mathbf{x}_{i+1,j}^\top & \mathbf{x}_{i,j+1}^\top & \mathbf{x}_{i-1,j}^\top\end{bmatrix}^\top \in \mathbb{R}^{12}$, and $\mathbf{z}_{i,j} \in \mathbb{R}^{12}$ is the subsystem's output. The matrices $E_{i,j}, A_{i,j} \in \mathbb{R}^{3\times 3}$, $B_{i,j} \in \mathbb{R}^{3\times 12}$ and $C_{i,j} \in \mathbb{R}^{12\times 3}$, which arise from the spatial discretisation of the governing equations, have different structure depending on the particular formulation/discretisation.

The structure of the state-space matrices describing a single computational node subsystem are presented for three different cases.

5.3.1 Case 1: primitive variables on co-located mesh

Upon discretising the standard primitive variables formulation using finite-differences on a co-located mesh, the $E_{i,j}$ and $A_{i,j}$ matrices were found to have the following structure:

$$E_{i,j} := \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix},\ A_{i,j} := \begin{bmatrix}a_{11} & a_{12} & 0\\ a_{21} & a_{22} & 0\\ 0 & 0 & 0\end{bmatrix}, \tag{5.9}$$

where:

$$a_{11} := -0.5\delta^{-1}\left(\bar{\mathsf{u}}_{i+1,j} - \bar{\mathsf{u}}_{i-1,j}\right) - 4\delta^{-2}\mathrm{Re}^{-1}, \tag{5.10a}$$

$$a_{12} := -0.5\delta^{-1}\left(\bar{\mathsf{u}}_{i,j+1} - \bar{\mathsf{u}}_{i,j-1}\right), \tag{5.10b}$$

$$a_{21} := -0.5\delta^{-1}\left(\bar{\mathsf{v}}_{i+1,j} - \bar{\mathsf{v}}_{i-1,j}\right), \tag{5.10c}$$

$$a_{22} := -0.5\delta^{-1}\left(\bar{\mathsf{v}}_{i,j+1} - \bar{\mathsf{v}}_{i,j-1}\right) - 4\delta^{-2}\mathrm{Re}^{-1}. \tag{5.10d}$$

When attempting to compute the node's transfer function as in (5.5), it is immediately clear that $(sE_{i,j} - A_{i,j})$ is singular (rank deficient) for all $s \in \mathbb{C}$, and so the resolvent $(sE_{i,j} - A_{i,j})^{-1}$ and transfer function $G_{i,j}(s)$ do not exist. This lack of uniqueness is the source of the checkerboard instability.

5.3.2 Case 2: primitive variables on staggered mesh

Discretising the same primitive variables formulation on a staggered mesh yields $E_{i,j}$ and $A_{i,j}$ matrices with the following structure:

$$E_{i,j} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; A_{i,j} := \begin{bmatrix} a_{11} & 0 & a_{13} \\ 0 & a_{22} & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix}, \tag{5.11}$$

where:

$$a_{11} := -0.5\delta^{-1}\left(\bar{\mathrm{u}}_{i+3/2,j} - \bar{\mathrm{u}}_{i-1/2,j}\right) - 4\delta^{-2}\mathrm{Re}^{-1}, \tag{5.12a}$$

$$a_{13} := \delta^{-1}, \tag{5.12b}$$

$$a_{22} := -0.5\delta^{-1}\left(\bar{\mathrm{v}}_{i,j-1/2} - \bar{\mathrm{v}}_{i,j+3/2}\right) - 4\delta^{-2}\mathrm{Re}^{-1}, \tag{5.12c}$$

$$a_{23} := -\delta^{-1}, \tag{5.12d}$$

$$a_{31} := 1, \tag{5.12e}$$

$$a_{32} := -1. \tag{5.12f}$$

In this case, the matrix pair $(E_{i,j}, A_{i,j})$ is regular (in that there exists $s \in \mathbb{C}$ such that $(sE_{i,j} - A_{i,j})$ is non-singular [29]), and so the resolvent $(sE_{i,j} - A_{i,j})^{-1}$, and hence transfer function $G_{i,j}(s)$, exist and are uniquely defined.

5.3.3 Case 3: PPE on co-located mesh

Finally, discretising the PPE formulation on a co-located mesh yields $E_{i,j}$ and $A_{i,j}$ matrices with the following structure:

$$E_{i,j} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \; A_{i,j} := \begin{bmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & a_{33} \end{bmatrix}, \tag{5.13}$$

where:

$$a_{11} := -0.5\delta^{-1}\left(\bar{\mathsf{u}}_{i+1,j} - \bar{\mathsf{u}}_{i-1,j}\right) - 4\delta^{-2}\mathrm{Re}^{-1}, \tag{5.14a}$$

$$a_{12} := -0.5\delta^{-1}\left(\bar{\mathsf{u}}_{i,j+1} - \bar{\mathsf{u}}_{i,j-1}\right), \tag{5.14b}$$

$$a_{21} := -0.5\delta^{-1}\left(\bar{\mathsf{v}}_{i+1,j} - \bar{\mathsf{v}}_{i-1,j}\right), \tag{5.14c}$$

$$a_{22} := -0.5\delta^{-1}\left(\bar{\mathsf{v}}_{i,j+1} - \bar{\mathsf{v}}_{i,j-1}\right) - 4\delta^{-2}\mathrm{Re}^{-1}, \tag{5.14d}$$

$$\begin{aligned} a_{31} := {} & \delta^{-1}\left(\bar{\mathsf{u}}_{i+1,j} - 2\bar{\mathsf{u}}_{i,j} + \bar{\mathsf{u}}_{i-1,j}\right) \\ & + 0.25\delta^{-2}\left(\bar{\mathsf{v}}_{i+1,j+1} - \bar{\mathsf{v}}_{i+1,j-1} - \bar{\mathsf{v}}_{i-1,j+1} + \bar{\mathsf{v}}_{i-1,j-1}\right), \end{aligned} \tag{5.14e}$$

$$\begin{aligned} a_{32} := {} & \delta^{-2}\left(\bar{\mathsf{v}}_{i,j+1} - 2\bar{\mathsf{v}}_{i,j} + \bar{\mathsf{v}}_{i,j-1}\right) \\ & + 0.25\delta^{-2}\left(\bar{\mathsf{u}}_{i+1,j+1} - \bar{\mathsf{u}}_{i+1,j-1} - \bar{\mathsf{u}}_{i-1,j+1} + \bar{\mathsf{u}}_{i-1,j-1}\right), \end{aligned} \tag{5.14f}$$

$$a_{33} := -4\delta^{-2}. \tag{5.14g}$$

This again yields a regular matrix pair $(E_{i,j}, A_{i,j})$, resulting in the existence and uniqueness of both the resolvent $(sE_{i,j} - A_{i,j})^{-1}$, and transfer function $G_{i,j}(s)$.

The existence of subsystem transfer functions for the cases of the primitive variables formulation discretised on a staggered mesh and the PPE formulation discretised on a co-located mesh shows that these formulations are well posed, whilst the contrary applies for the case of the primitive variables formulation discretised on a co-located mesh.

5.4 PPE pressure boundary conditions

Although the PPE formulation discretised on a co-located mesh is well posed, the main difficulty with such a formulation lies in the correct choice of pressure boundary conditions. Several different pressure boundary conditions present in the literature are now considered.

The first, and simplest pressure boundary condition considered is the Neumann type condition:

$$\frac{\partial \mathsf{p}'(\mathbf{x}, t)}{\partial \mathbf{n}} = 0, \ \forall \ (\mathbf{x}, t) \in \partial\Omega \times [0, t_f], \tag{5.15}$$

where $\mathbf{n} \in \mathbb{R}^2$ is the normal outward pointing unit vector. As stated in [84], the wall-normal gradient of the pressure at the boundary in incompressible flow is, in general, non-zero, however if treated with care it can be used as part of fractional step methods [93] for the purposes of simulation.

For the case of the incompressible nonlinear Navier-Stokes equations, it was suggested by Gresho and Sani [44] that the correct pressure boundary condition for a PPE

formulation is a Neumann type condition obtained by projecting the momentum equation onto the boundary in the normal direction. Applying this idea to the linearised equations yields the boundary condition:

$$\begin{aligned}\mathbf{n}\cdot\nabla\mathsf{p}'(\mathbf{x},t) = \mathbf{n}\cdot\Bigg(&\frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}'(\mathbf{x},t) - \mathbf{u}'(\mathbf{x},t)\cdot\nabla\bar{\mathbf{u}}(\mathbf{x},t)\\ &-\bar{\mathbf{u}}(\mathbf{x},t)\cdot\nabla\mathbf{u}'(\mathbf{x},t) - \frac{\partial}{\partial t}\mathbf{u}'(\mathbf{x},t)\Bigg),\ \forall\,(\mathbf{x},t)\in\partial\Omega\times[0,t_f]\,.\end{aligned} \tag{5.16}$$

For the common case of a no-slip boundary this condition reduces to:

$$\mathbf{n}\cdot\nabla\mathsf{p}'(\mathbf{x},t) = \mathbf{n}\cdot\frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}'(\mathbf{x},t),\ \forall\,(\mathbf{x},t)\in\partial\Omega\times[0,t_f]\,. \tag{5.17}$$

Whilst this seems a more logical approach to deducing a valid pressure boundary condition than that discussed above, Rempfer [83, 84] points out that using the PPE in conjunction with this condition yields a system of equations where the divergence free condition (5.2b) appears nowhere. For the nonlinear case, Rempfer [84] shows that substituting the PPE into the divergence of the momentum equation yields a heat equation. For the case of the linearised equations, this amounts to substituting (5.2c) into the divergence of (5.2a), yielding:

$$\frac{\partial}{\partial t}\Theta(\mathbf{x},t) = \frac{1}{\mathrm{Re}}\nabla^2\Theta(\mathbf{x},t),\ \forall\,(\mathbf{x},t)\in\Omega\times[0,t_f]\,, \tag{5.18}$$

where $\Theta(\mathbf{x},t) := \nabla\cdot\mathbf{u}'(\mathbf{x},t)$ is the divergence of the fluctuating part of the velocity field. This shows that the combination of (5.2a) and (5.2c) does not enforce the incompressibility of the velocity field, it just specifies that the divergence of the velocity field satisfies a heat equation, and is a harmonic funtion in the steady-state case. According to the extremum theorems for harmonic functions [78], the extremal values of harmonic functions are reached on the boundary $\partial\Omega$, which led to Rempfer's theorem regarding the nonlinear case [84, Thm.1], stating that the solution to the incompressible Navier-Stokes equations in primitive variables formulation is equivalent to the solution of the PPE formulation if and only if the boundary conditions for the PPE are chosen such that $\nabla\cdot\mathbf{u}(\mathbf{x},t) = 0,\ \forall\,(\mathbf{x},t)\in\partial\Omega\times[0,t_f]$. This was described by Rempfer [83, 84] as an 'indirect boundary condition', since the boundary condition for one variable, in this case pressure, must be chosen such that some condition for another variable, in this case velocity, is met on the boundary.

In more recent work, Shirokoff and Rosales [91] took a new approach to choosing not only the PPE pressure boundary condition, but also the corresponding velocity boundary conditions for a no-slip boundary. Rather than enforcing all d components of the no-slip boundary condition on the nonlinear momentum equation (where $d\in\{2,3\}$ is the number of spatial dimensions), only the $d-1$ tangential components are enforced,

and the set of boundary conditions for the momentum equation is completed with the divergence free condition (5.2b). In practice, this results in specifying that the $d-1$ tangential velocity components are equal to zero on the boundary, and that the normal gradient of the normal velocity component is also equal to zero, hence satisfying (5.2b). For the linearised equations this becomes:

$$\boldsymbol{\tau} \cdot \mathbf{u}'(\mathbf{x}, t) = 0, \ \forall \ (\mathbf{x}, t) \in \partial\Omega \times [0, t_f] \,, \tag{5.19a}$$

$$\mathbf{n} \cdot \left(\nabla \left(\mathbf{n} \cdot \mathbf{u}'(\mathbf{x}, t) \right) \right) = 0, \ \forall \ (\mathbf{x}, t) \in \partial\Omega \times [0, t_f] \,, \tag{5.19b}$$

where $\boldsymbol{\tau} \in \mathbb{R}^2$ is the wall-tangential unit vector. As for the pressure boundary condition, the momentum equation is projected in the normal direction onto the boundary yielding the Neumann type condition (5.15).

5.5 Investigating dynamics of 2D channel flow

Analysis of the structure of the system matrices describing a single computational node subsystem revealed, from a systems theory perspective, the unsuitability of the standard primitive variables formulation on a co-located mesh discretised using finite-differences. However, such analysis revealed little concerning the suitability for feedback control system design, based on models obtained from different formulations employing various boundary conditions. In order to gain insight into this, the dynamics of a full fluid flow system must be considered.

In the following, the benchmark example of a fully developed flow between two flat plates, otherwise known as channel flow or plane Poiseuille flow is considered. This is one of very few fluid flows for which an analytical solution to the Navier-Stokes equations is known, and for which the flow dynamics have been studied extensively (see, e.g. [72, 17, 89, 88]).

After non-dimensionalising length scales by the channel half-height h, velocity scales by the maximum laminar centreline velocity $\bar{\mathsf{u}}_0$, and pressure by $\rho\bar{\mathsf{u}}_0^2$, the upper walls of the channel are located at $\mathsf{y} = \pm 1$, and the streamwise (x) channel length is $l = 2\pi$, as depicted in Figure 5.3. It can be shown that the the baseline velocity is independent of x, and is equal to $\bar{\mathsf{u}}(\mathsf{y}) = 1 - \mathsf{y}^2$. Hence, the linearised Navier-Stokes equations in

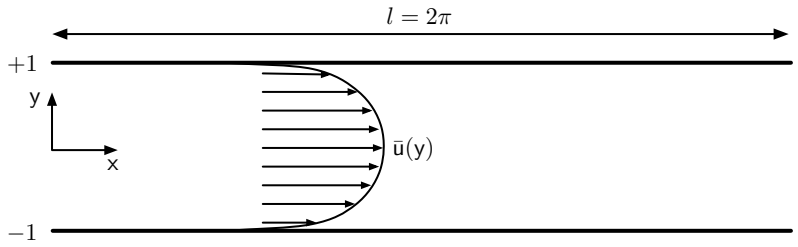


Figure 5.3: Schematic of 2D channel flow geometry.

primitive variables formulation reduce to:

$$\frac{\partial \mathsf{u}'(\mathbf{x},t)}{\partial t} + \left(1-\mathsf{y}^2\right)\frac{\partial \mathsf{u}'(\mathbf{x},t)}{\partial \mathsf{x}} - 2\mathsf{y}\mathsf{v}'(\mathbf{x},t) = -\frac{\partial \mathsf{p}'(\mathbf{x},t)}{\partial \mathsf{x}} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \mathsf{u}'(\mathbf{x},t)}{\partial \mathsf{x}^2} + \frac{\partial^2 \mathsf{u}'(\mathbf{x},t)}{\partial \mathsf{y}^2}\right), \tag{5.20a}$$

$$\frac{\partial \mathsf{v}'(\mathbf{x},t)}{\partial t} + \left(1-\mathsf{y}^2\right)\frac{\partial \mathsf{v}'(\mathbf{x},t)}{\partial \mathsf{x}} = -\frac{\partial \mathsf{p}'(\mathbf{x},t)}{\partial \mathsf{y}} + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2 \mathsf{v}'(\mathbf{x},t)}{\partial \mathsf{x}^2} + \frac{\partial^2 \mathsf{v}'(\mathbf{x},t)}{\partial \mathsf{y}^2}\right), \tag{5.20b}$$

$$\frac{\partial \mathsf{u}'(\mathbf{x},t)}{\partial \mathsf{x}} + \frac{\partial \mathsf{v}'(\mathbf{x},t)}{\partial \mathsf{y}} = 0, \tag{5.20c}$$

where the Reynolds number is defined as $\mathrm{Re} := \bar{\mathsf{u}}_0 h/\nu$.

The PPE (5.2c) reduces to:

$$\frac{\partial^2 \mathsf{p}'(\mathbf{x},t)}{\partial \mathsf{x}^2} + \frac{\partial^2 \mathsf{p}'(\mathbf{x},t)}{\partial \mathsf{y}^2} = 4\mathsf{y}\frac{\partial \mathsf{v}'(\mathbf{x},t)}{\partial \mathsf{x}}. \tag{5.21}$$

No-slip boundary conditions were assumed on the upper and lower walls, $\mathsf{u}'(\mathsf{x},\pm 1,t) = \mathsf{v}'(\mathsf{x},\pm 1,t) = 0$, and boundary conditions in the streamwise direction were assumed periodic, $\mathsf{u}'(0,\mathsf{y},t) = \mathsf{u}'(l,\mathsf{y},t)$, $\mathsf{v}'(0,\mathsf{y},t) = \mathsf{v}'(l,\mathsf{y},t)$, and $\mathsf{p}'(0,\mathsf{y},t) = \mathsf{p}'(l,\mathsf{y},t)$.

With these equations, finite-dimensional state-space models were constructed which describe the dynamics of the fluctuation variables about the baseline flow.

5.5.1 Orr-Sommerfeld state-space model

Firstly, the well known Orr-Sommerfeld formulation [89, 14, 68] was used to obtain a benchmark state-space model of the flow, against which other formulations could be compared.

Substituting (5.20c) along with the divergence of (5.20a) and (5.20b) into the Laplacian of (5.20b) yields the following 4^{th} order PDE, where the pressure perturbation

variable $\mathsf{p}'(\mathsf{x},t)$, has been analytically removed:

$$\nabla^2\frac{\partial}{\partial t}\mathsf{v}'(\mathsf{x},t)+\left(1-\mathsf{y}^2\right)\frac{\partial}{\partial \mathsf{x}}\nabla^2\mathsf{v}'(\mathsf{x},t)+2\frac{\partial}{\partial \mathsf{x}}\mathsf{v}'(\mathsf{x},t)=\frac{1}{\mathrm{Re}}\nabla^4\mathsf{v}'(\mathsf{x},t). \tag{5.22}$$

Exploiting the periodicity in the streamwise direction, Fourier transforms are taken in x. This allows the y direction perturbation velocity to be written:

$$\mathsf{v}'(\mathsf{x},t):=\hat{\mathsf{v}}(\mathsf{y},t)e^{\mathrm{i}\alpha\mathsf{x}}, \tag{5.23}$$

where $\hat{\mathsf{v}}(\mathsf{y},t)\in\mathbb{C}$ is the Fourier transformed wall-normal velocity perturbation at an individual spatial wavenumber $\alpha\in\mathbb{R}$. Spatial derivatives with respect to x in (5.22) simply become multiplication by $\mathrm{i}\alpha$, and (5.22) can be written:

$$\mathcal{E}\frac{\partial}{\partial t}\hat{\mathsf{v}}(\mathsf{y},t)=\mathcal{A}\hat{\mathsf{v}}(\mathsf{y},t), \tag{5.24}$$

where the system operators are defined as:

$$\mathcal{E}:=\mathrm{Re}\left(\frac{\partial^2}{\partial \mathsf{y}^2}-\alpha^2\right), \tag{5.25a}$$

$$\mathcal{A}:=\frac{\partial^4}{\partial \mathsf{y}^4}-2\alpha^2\frac{\partial^2}{\partial \mathsf{y}^2}+\alpha^4-\mathrm{i}\alpha\mathrm{Re}\left(\left(1-\mathsf{y}^2\right)\left(\frac{\partial^2}{\partial \mathsf{y}^2}-\alpha^2\right)+2\right). \tag{5.25b}$$

As is standard for channel flow geometries, discretisation in the wall-normal direction was performed using Chebyshev interpolants. As such, (5.24) was discretised upon $n_{\mathsf{y}}\in\mathbb{N}$ Chebyshev collocation points and the wall-normal differential operators were approximated by differentiation matrices [102], yielding the following finite-dimensional descriptor state equation:

$$E\frac{\mathrm{d}}{\mathrm{d}t}\vec{\hat{\mathsf{v}}}(t)=A\vec{\hat{\mathsf{v}}}(t), \tag{5.26}$$

where $E,A\in\mathbb{C}^{n_{\mathsf{y}}\times n_{\mathsf{y}}}$ are discrete versions of $\mathcal{E}$ and $\mathcal{A}$, respectively, and $\vec{\hat{\mathsf{v}}}(t)\in\mathbb{C}^{n_{\mathsf{y}}}$ is the state vector comprising the values of $\hat{\mathsf{v}}(\mathsf{y},t)$ evaluated on the collocation points. Finally, premultiplying (5.26) by E^{-1} yields the following state equation:

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{\hat{\mathsf{v}}}(t)=A_{\mathrm{OS}}\vec{\hat{\mathsf{v}}}(t), \tag{5.27}$$

where $A_{\mathrm{OS}}:=E^{-1}A$ is the system dynamics matrix.

The poles of the Orr-Sommerfeld system correspond to the eigenvalues of A_{OS}, and are dependent on the Reynolds number and streamwise wavenumber. For the case $(\mathrm{Re},\alpha)=(5772.22,1.02)$, the poles of the system are plotted in Figure 5.4(a) and are in agreement with those originally calculated in [73]. For clarity of exposition, the three main branches of the spectrum are labelled A, S and P, as in [64]. The pole with

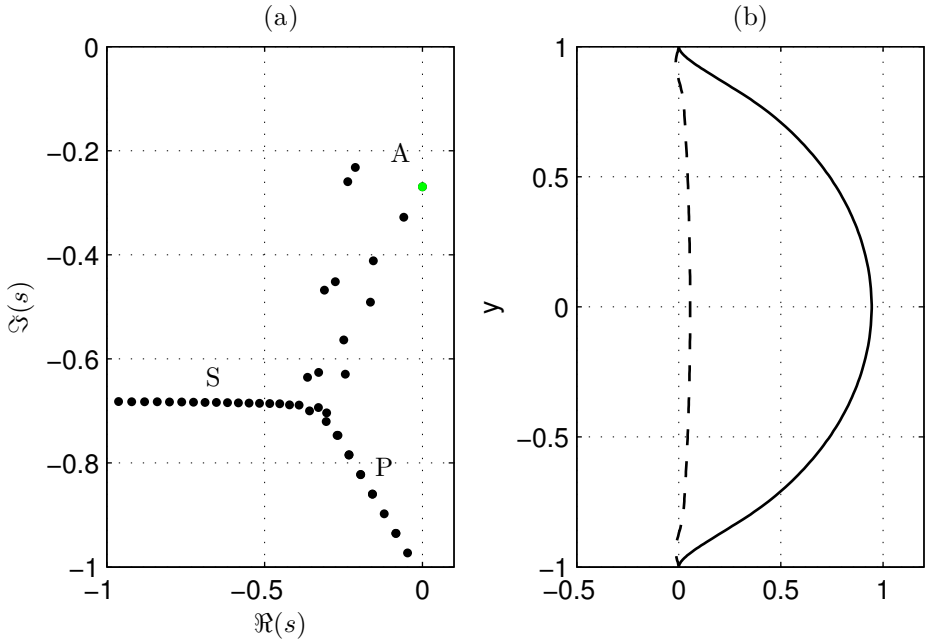


Figure 5.4: Orr-Sommerfeld system: (a) poles (•) and least stable pole (•); (b) eigenfunction corresponding to least stable mode of Orr-Sommerfeld system: $\Re(\hat{\mathsf{v}}(\mathsf{y}))$ $(-)$, $\Im(\hat{\mathsf{v}}(\mathsf{y}))$ $(--)$.

greatest real part (corresponding to the least stable mode) is shown in green, along with its mode shape (eigenfunction) in Figure 5.4(b).

5.5.2 Primitive variables and PPE formulation state-space models

State-space models of 2D channel flow based on both primitive variables and PPE formulations were then constructed. Starting with (5.20) and (5.21), once again the periodicity in the streamwise direction was exploited and Fourier transforms were taken in x, such that the variables can be written:

$$\mathsf{u}'(\mathsf{x},t) := \hat{\mathsf{u}}(\mathsf{y},t)e^{\mathrm{i}\alpha\mathsf{x}}, \quad \mathsf{v}'(\mathsf{x},t) := \hat{\mathsf{v}}(\mathsf{y},t)e^{\mathrm{i}\alpha\mathsf{x}}, \quad \mathsf{p}'(\mathsf{x},t) := \hat{\mathsf{p}}(\mathsf{y},t)e^{\mathrm{i}\alpha\mathsf{x}}. \tag{5.28}$$

Evaluating the derivatives with respect to x in (5.20) and (5.21) yields two systems of PDAEs. The first of these is the following primitive variables formulation:

$$\underbrace{\begin{bmatrix}\mathcal{I} & 0 & 0\\ 0 & \mathcal{I} & 0\\ 0 & 0 & 0\end{bmatrix}}_{\mathcal{E}_{\mathrm{Pr}} \in \mathbb{R}^{3\times 3}} \frac{\partial}{\partial t} \underbrace{\begin{bmatrix}\hat{\mathsf{u}}(\mathsf{y},t)\\ \hat{\mathsf{v}}(\mathsf{y},t)\\ \hat{\mathsf{p}}(\mathsf{y},t)\end{bmatrix}}_{\boldsymbol{\chi}(\mathsf{y},t) \in \mathbb{C}^3} = \underbrace{\begin{bmatrix}\mathcal{A}_{11} & \mathcal{A}_{12} & \mathcal{A}_{13}\\ 0 & \mathcal{A}_{22} & \mathcal{A}_{23}\\ \mathcal{A}_{31} & \mathcal{A}_{32} & 0\end{bmatrix}}_{\mathcal{A}_{\mathrm{Pr}} \in \mathbb{C}^{3\times 3}} \begin{bmatrix}\hat{\mathsf{u}}(\mathsf{y},t)\\ \hat{\mathsf{v}}(\mathsf{y},t)\\ \hat{\mathsf{p}}(\mathsf{y},t)\end{bmatrix}, \tag{5.29}$$

where the system operators in $\mathcal{A}_{\mathrm{Pr}}$ are defined as:

$$\mathcal{A}_{11} = A_{22} := -\mathrm{i}\alpha\left(1-\mathsf{y}^2\right) + \frac{1}{\mathrm{Re}}\left(\frac{\partial^2}{\partial \mathsf{y}^2} - \alpha^2\right), \tag{5.30a}$$

$$\mathcal{A}_{12} := 2\mathsf{y}, \tag{5.30b}$$

$$\mathcal{A}_{13} := -\mathrm{i}\alpha, \tag{5.30c}$$

$$\mathcal{A}_{23} := -\frac{\partial}{\partial \mathsf{y}}, \tag{5.30d}$$

$$\mathcal{A}_{31} := \mathrm{i}\alpha, \tag{5.30e}$$

$$\mathcal{A}_{32} := \frac{\partial}{\partial \mathsf{y}}, \tag{5.30f}$$

and $\mathcal{I}$ is the identity operator.

The second system of PDAEs is the following PPE formulation:

$$\underbrace{\begin{bmatrix}\mathcal{I} & 0 & 0\\ 0 & \mathcal{I} & 0\\ 0 & 0 & 0\end{bmatrix}}_{\mathcal{E}_{\mathrm{PPE}} \in \mathbb{R}^{3\times 3}} \frac{\partial}{\partial t} \underbrace{\begin{bmatrix}\hat{\mathsf{u}}(\mathsf{y},t)\\ \hat{\mathsf{v}}(\mathsf{y},t)\\ \hat{\mathsf{p}}(\mathsf{y},t)\end{bmatrix}}_{\boldsymbol{\chi}(\mathsf{y},t) \in \mathbb{C}^3} = \underbrace{\begin{bmatrix}\mathcal{A}_{11} & \mathcal{A}_{12} & \mathcal{A}_{13}\\ 0 & \mathcal{A}_{22} & \mathcal{A}_{23}\\ 0 & \mathcal{A}_{32} & \mathcal{A}_{33}\end{bmatrix}}_{\mathcal{A}_{\mathrm{PPE}} \in \mathbb{C}^{3\times 3}} \begin{bmatrix}\hat{\mathsf{u}}(\mathsf{y},t)\\ \hat{\mathsf{v}}(\mathsf{y},t)\\ \hat{\mathsf{p}}(\mathsf{y},t)\end{bmatrix}, \tag{5.31}$$

where the system operators $\mathcal{A}_{11}$, $\mathcal{A}_{12}$, $\mathcal{A}_{13}$, $\mathcal{A}_{21}$, and $\mathcal{A}_{22}$ in $\mathcal{A}_{\mathrm{PPE}}$ are the same as those in $\mathcal{A}_{\mathrm{Pr}}$, but $\mathcal{A}_{32}$ and $\mathcal{A}_{33}$ are defined as:

$$\mathcal{A}_{32} := -4\mathrm{i}\alpha\mathsf{y}, \tag{5.32a}$$

$$\mathcal{A}_{33} := \frac{\partial^2}{\partial \mathsf{y}^2} - \alpha^2. \tag{5.32b}$$

The wall-normal derivatives in (5.29) and (5.31) were then approximated either using Chebyshev methods, or using finite-differences on co-located or staggered meshes. In the PPE case different pressure boundary conditions were then enforced in order to see how these differences affected the models. This yields finite-dimensional descriptor state-space systems of the form:

$$E_{\mathrm{Pr}}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = A_{\mathrm{Pr}}\mathbf{x}(t), \tag{5.33}$$

and:

$$E_{\mathrm{PPE}}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = A_{\mathrm{PPE}}\mathbf{x}(t), \tag{5.34}$$

where $\mathbf{x}(t) := \begin{bmatrix}\vec{\hat{\mathsf{u}}}^*(t) & \vec{\hat{\mathsf{v}}}^*(t) & \vec{\hat{\mathsf{p}}}^*(t)\end{bmatrix}^* \in \mathbb{C}^{n_{\mathsf{u}}+n_{\mathsf{v}}+n_{\mathsf{p}}}$ is the state vector comprising the values of Fourier tansformed (in x) flow variables at the discretisation points. The

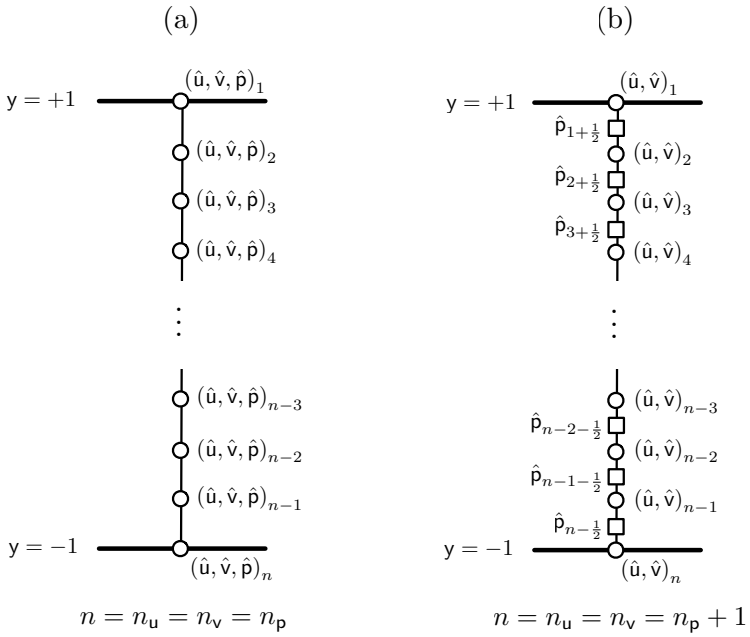


Figure 5.5: Channel flow computational meshes (in y direction): (a) co-located computational mesh; (b) staggered mesh.

discrete versions of $\mathcal{E}_{\text{Pr}}$ and $\mathcal{A}_{\text{Pr}}$ are $E_{\text{Pr}}, A_{\text{Pr}} \in \mathbb{C}^{(n_u+n_v+n_p)\times(n_u+n_v+n_p)}$, respectively, and $E_{\text{PPE}}, A_{\text{PPE}} \in \mathbb{C}^{(n_u+n_v+n_p)\times(n_u+n_v+n_p)}$ are the discrete versions of $\mathcal{E}_{\text{PPE}}$ and $\mathcal{A}_{\text{PPE}}$. In the co-located mesh case, $n_u = n_v = n_p$, where $n_u, n_v, n_p \in \mathbb{N}$ are the respective number of $\hat{u}$, $\hat{v}$, and $\hat{p}$ discretisation points. However in the staggered mesh case, $n_u = n_v = n_p + 1$.

5.5.3 Channel flow formulations

A number of formulations of the linearised incompressible Navier-Stokes equations were used to deduce descriptor state-space models of 2D channel flow of the form (5.33) and (5.34). In the case of primitive variables formulations, both co-located and staggered mesh arrangements were considered. It should be noted that due to the use of Fourier transforms in the x direction, the problem is simplified to essentially a problem in one spatial dimension, with the corresponding meshes shown in Figure 5.5. For the PPE formulation case a number of different pressure boundary conditions were enforced. Table 5.1 summarises the different models constructed. The Orr-Sommerfeld model (5.27) is denoted [OS], whilst [Pr1]–[Pr3] are models based on the primitive variables formulation (5.33), discretised in different fashions according to their identifiers in Table 5.1.

Models [PPE1]–[PPE4] are based on the PPE formulation (5.34) all discretised on co-located meshes. [PPE1] implements the simple $\partial\hat{p}/\partial y = 0$ Neumann boundary

Model identifier	Primitive variables or PPE	co-located or staggered mesh	discretisation technique in y direction	$\mathsf{y} = \pm 1$ boundary conditions
[OS]	-	co-located	Chebyshev	$\hat{\mathsf{v}} = \frac{\partial \hat{\mathsf{v}}}{\partial \mathsf{y}} = 0$
[Pr1]	primitive	co-located	Chebyshev	$\hat{\mathsf{u}} = \hat{\mathsf{v}} = 0$
[Pr2]	primitive	co-located	finite-difference	$\hat{\mathsf{u}} = \hat{\mathsf{v}} = 0$
[Pr3]	primitive	staggered	finite-difference	$\hat{\mathsf{u}} = \hat{\mathsf{v}} = 0$
[PPE1]	PPE	co-located	Chebyshev	$\hat{\mathsf{u}} = \hat{\mathsf{v}} = \frac{\partial \hat{\mathsf{p}}}{\partial \mathsf{y}} = 0$
[PPE2]	PPE	co-located	Chebyshev	$\hat{\mathsf{u}} = \hat{\mathsf{v}} = \nabla \cdot \hat{\mathbf{u}} = 0$
[PPE3]	PPE	co-located	Chebyshev	$\hat{\mathsf{u}} = \frac{\partial \hat{\mathsf{v}}}{\partial \mathsf{y}} = \frac{\partial \hat{\mathsf{p}}}{\partial \mathsf{y}} - \frac{1}{\mathrm{Re}} \frac{\partial^2 \hat{\mathsf{v}}}{\partial \mathsf{y}^2} = 0$
[PPE4]	PPE	co-located	finite-difference	$\hat{\mathsf{u}} = \frac{\partial \hat{\mathsf{v}}}{\partial \mathsf{y}} = \frac{\partial \hat{\mathsf{p}}}{\partial \mathsf{y}} - \frac{1}{\mathrm{Re}} \frac{\partial^2 \hat{\mathsf{v}}}{\partial \mathsf{y}^2} = 0$

Table 5.1: Summary of different 2D channel flow models.

condition. [PPE2] implements the divergence free velocity boundary condition in an attempt to satisfy the conditions stated by Rempfer [84]. Finally, [PPE3] (Chebyshev) and [PPE4] (finite-differences) implement the Neumann pressure boundary condition discussed in both [44] and [91], with the additional Neumann condition on the normal component of the velocity in order to satisfy the divergence free condition on the boundary, as suggested in [91].

Imposing boundary conditions in models of the form (5.33) and (5.34) is simply a matter of altering the rows of the E and A matrices corresponding to the boundary node states. In the following, the dynamics of each formulation are compared in a variety of ways. Firstly, the asymptotic behaviour of the systems is studied through comparison of the system poles and corresponding eigenfunctions. Secondly, differences in transient behaviour are explored through comparison of energy weighted pseudospectra. The various frequency responses of the systems for a particular system input and output are computed and compared, and finally the ν-gaps between different model formulations and the benchmark Orr-Sommerfeld model are computed in order to quantify the differences in the models from a closed-loop dynamics perspective.

5.5.4 System poles and eigenfunctions

Figure 5.6 shows the poles of all three primitive variables models, [Pr1]–[Pr3], and both the $\hat{\mathsf{v}}$ and $\hat{\mathsf{p}}$ parts of the eigenfunction corresponding to the least stable mode. The poles of all three formulations lie in the same locations as those of [OS]. In all three cases the $\hat{\mathsf{v}}$ parts of the eigenfunction agree well with that of [OS]. The $\hat{\mathsf{p}}$ parts of the eigenfunctions of [Pr1] and [Pr3] are in agreement, however whilst the general shape of the $\hat{\mathsf{p}}$ part of [Pr2] is similar, a non-physical sawtooth shape is clearly evident.

With respect to [Pr2], the use Fourier transforms in the x direction alters the structure of the A matrix in (5.9) in such a way as to ensure that its transfer function exists, and so the eigenfunctions are unique (to within an arbitrary scaling). However, the use of centred finite-differences in the wall-normal direction contributes to a loss of coupling between the pressure and velocity variables, hence giving rise to the non-physical pressure mode shape.

Figure 5.7 shows the poles of the four PPE formulations, [PPE1]–[PPE4], and both the $\hat{\mathsf{v}}$ and $\hat{\mathsf{p}}$ parts of the eigenfunction corresponding to the least stable mode. Whilst the poles of all four systems lie in three characteristic branches, the exact pole locations do not completely agree with those of [OS]. The poles of [PPE1] in the S branch agree well, however in the A branch there is significant difference, not least in the failure to agree in the location of the least stable pole. There are also extra spurious poles in the P branch. The failure of this formulation to accurately match the dynamics of [OS] is unsurprising, since as was mentioned in Section 5.4, the Neumann condition (5.15) is unphysical and its use is primarily for the separate purpose of time-marching, as part of a predictor-corrector scheme.

In the cases of [PPE2]–[PPE4], whilst the A branches contain all the poles present in that of [OS], the branch is contaminated by additional spurious poles. Similarly to [PPE1], additional spurious poles exist in the P branch. Despite the differences in pole locations in [PPE2]–[PPE4], the $\hat{\mathsf{v}}$ parts of the eigenfunctions agree well with that of [OS], and all three $\hat{\mathsf{p}}$ parts match. The same cannot be said for [PPE1], however, since neither the $\hat{\mathsf{v}}$ nor $\hat{\mathsf{p}}$ parts agree exactly with the other models.

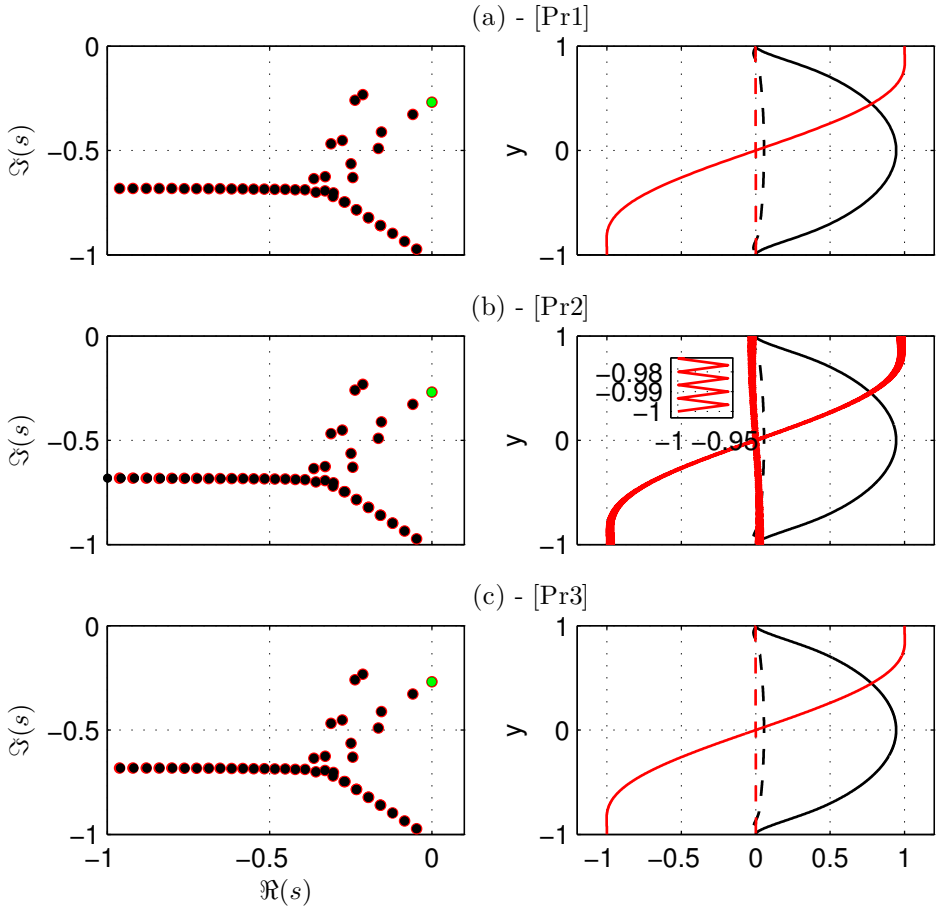


Figure 5.6: Left: poles of [OS] (•), poles of [Pr(·)] (•), and least stable pole (•); Right: $\Re(\hat{v}(y))$ (−), $\Im(\hat{v}(y))$ (−−), $\Re(\hat{p}(y))$ (−), $\Im(\hat{p}(y))$ (−−), where $\hat{v}(y)$ and $\hat{p}(y)$ are parts of the eigenfunction corresponding to the least stable pole. (a) [Pr1]; (b) [Pr2] - inset (right) is close-up of part of $\Re(\hat{p}(y))$; (c) [Pr3].

5.5.5 System psuedospectra

Whilst the poles of a system dictate its asymptotic behaviour, the transient response is influenced by the non-orthogonality of the system eigenfunctions, as is well understood in the area of non-modal stability theory [17, 96, 88]. The psuedospectra Λ_ϵ, of a system provide a means of visualising this non-orthogonality in the complex plane. For a standard state-space system ($E = I$ in (5.4)), pseudospectra are defined as [94, 95]:

$$\Lambda_\epsilon := \left\{ s \in \mathbb{C} : \left\| C(sI - A)^{-1} B \right\|_2 \geq \epsilon^{-1} \in \mathbb{R}^+ \right\}, \tag{5.35}$$

where C and B represent weights required to convert a measure of energy to the standard $\mathscr{L}_2$–norm [88]. This is accomplished by first defining the kinetic energy density of

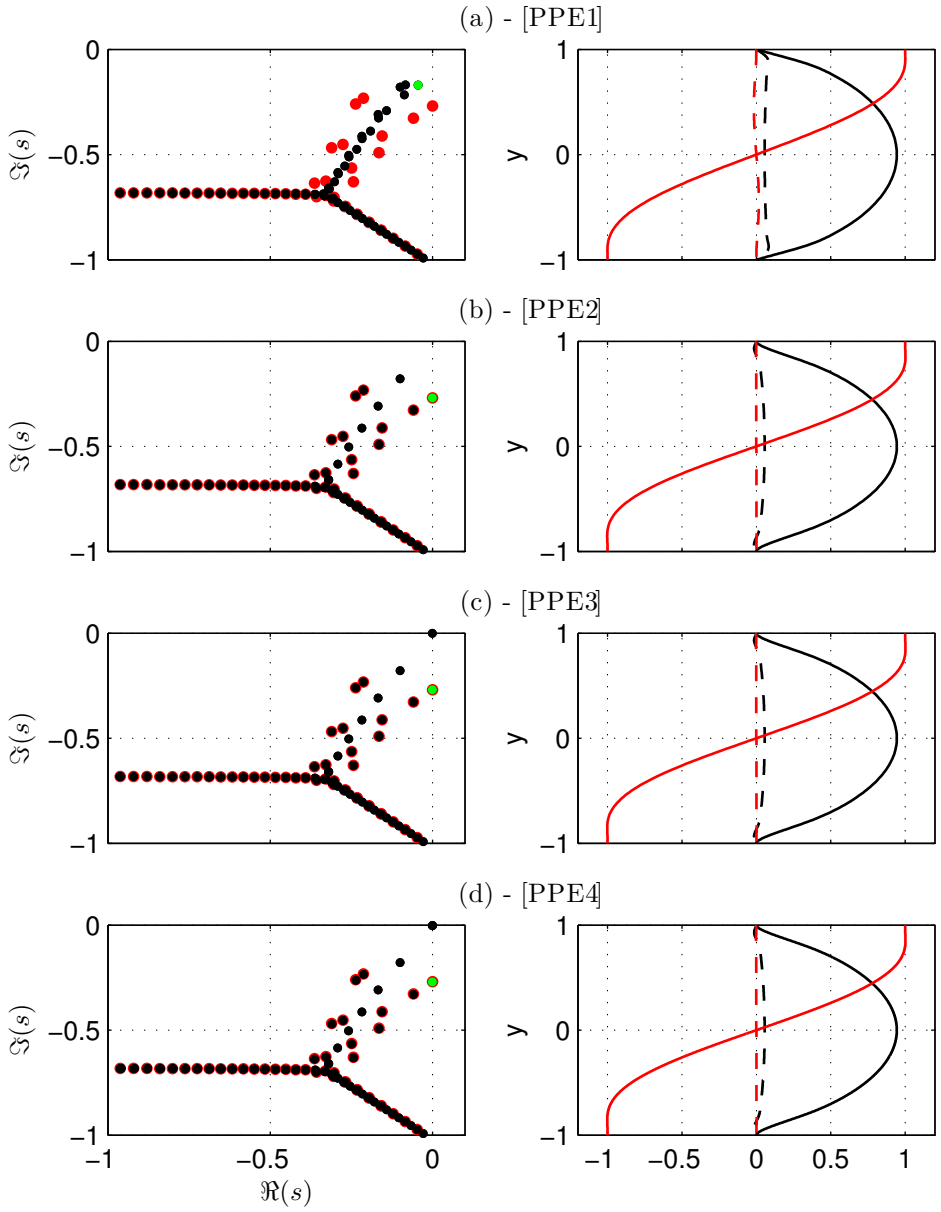


Figure 5.7: Left: poles of [OS] (●), poles of [PPE(·)] (●), and least stable pole (●); Right: $\Re(\hat{v}(y))$ (−), $\Im(\hat{v}(y))$ (−−), $\Re(\hat{p}(y))$ (−), $\Im(\hat{p}(y))$ (−−), where $\hat{v}(y)$ and $\hat{p}(y)$ are parts of the eigenfunction corresponding to the least stable pole. (a) [PPE1]; (b) [PPE2]; (c) [PPE3]; (d) [PPE4].

the flow perturbations $\mathscr{E}(t) \in \mathbb{R}_0^+$, as in [17], as:

$$\mathscr{E}(t) := \frac{1}{2\mathscr{V}} \int_{-1}^{1} \int_{0}^{\frac{2\pi}{\alpha}} u'^2(x,t) + v'^2(x,t) dx dy, \tag{5.36}$$

where $\mathscr{V} = 4\pi/\alpha$ is the integration volume. In terms of Fourier transformed velocities, this reduces to:

$$\mathscr{E}(t) = \frac{1}{8}\int_{-1}^{1} \hat{\mathsf{u}}^*(\mathsf{y},t)\hat{\mathsf{u}}(\mathsf{y},t) + \hat{\mathsf{v}}^*(\mathsf{y},t)\hat{\mathsf{v}}(\mathsf{y},t)\mathrm{d}\mathsf{y}. \tag{5.37}$$

This integral is evaluated numerically, and can be written

$$\mathscr{E}(t) = \begin{bmatrix}\vec{\hat{\mathsf{u}}}^*(t) & \vec{\hat{\mathsf{v}}}^*(t) & \vec{\hat{\mathsf{p}}}^*(t)\end{bmatrix} \underbrace{\begin{bmatrix}\frac{1}{8}\mathscr{W}_{\hat{\mathsf{u}}} & 0 & 0\\ 0 & \frac{1}{8}\mathscr{W}_{\hat{\mathsf{v}}} & 0\\ 0 & 0 & 0\end{bmatrix}}_{\mathscr{Q}} \begin{bmatrix}\vec{\hat{\mathsf{u}}}(t)\\ \vec{\hat{\mathsf{v}}}(t)\\ \vec{\hat{\mathsf{p}}}(t)\end{bmatrix} \tag{5.38a}$$

$$= \mathbf{x}^*(t)\mathscr{Q}\mathbf{x}(t), \tag{5.38b}$$

where $\mathscr{W}_{\hat{\mathsf{u}}} \in \mathbb{R}^{n_{\mathsf{u}}\times n_{\mathsf{u}}}$ and $\mathscr{W}_{\hat{\mathsf{v}}} \in \mathbb{R}^{n_{\mathsf{v}}\times n_{\mathsf{v}}}$ are diagonal matrices of quadrature weights.

For the case of [OS], $\vec{\hat{\mathsf{u}}}(t)$ does not appear in the state vector, but can be derived from the continuity equation as follows:

$$\vec{\hat{\mathsf{u}}}(t) = \frac{\mathrm{i}}{\alpha}\mathcal{D}_{\mathsf{y}}\vec{\hat{\mathsf{v}}}(t), \tag{5.39}$$

where $\mathcal{D}_{\mathsf{y}} \in \mathbb{R}^{n_{\mathsf{y}}\times n_{\mathsf{y}}}$ is the first order Chebyshev differentiation matrix. For [OS] (5.38) can then be written:

$$\mathscr{E}(t) = \vec{\hat{\mathsf{v}}}^*(t)\underbrace{\left(\frac{1}{8\alpha^2}\mathcal{D}_{\mathsf{y}}^{\top}\mathscr{W}_{\hat{\mathsf{v}}}\mathcal{D}_{\mathsf{y}} + \frac{1}{8}\mathscr{W}_{\hat{\mathsf{v}}}\right)}_{\mathscr{Q}_{\mathrm{OS}}}\vec{\hat{\mathsf{v}}}(t), \tag{5.40}$$

where $\mathscr{Q}_{\mathrm{OS}}$ can be factored as $\mathscr{Q}_{\mathrm{OS}} = C_{\mathrm{OS}}^{\top}C_{\mathrm{OS}}$. Hence, the weighting matrices for computing the pseudospectra in (5.35) for [OS] are $B = C_{\mathrm{OS}}^{-1}$ and $C = C_{\mathrm{OS}}$.

To compute the pseudospectra of the primitive variables and PPE formulations, it is convenient to convert from descriptor to standard state-space systems. This is accomplished using the numerical algorithm outlined in [57], yielding systems of the form:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \check{A}_{\mathrm{Pr}}\mathbf{x}(t), \tag{5.41}$$

and:

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = \check{A}_{\mathrm{PPE}}\mathbf{x}(t). \tag{5.42}$$

In doing so, state transformation matrices $\mathcal{S}_{\mathrm{Pr}}$ and $\mathcal{S}_{\mathrm{PPE}}$ are computed, and are subse-

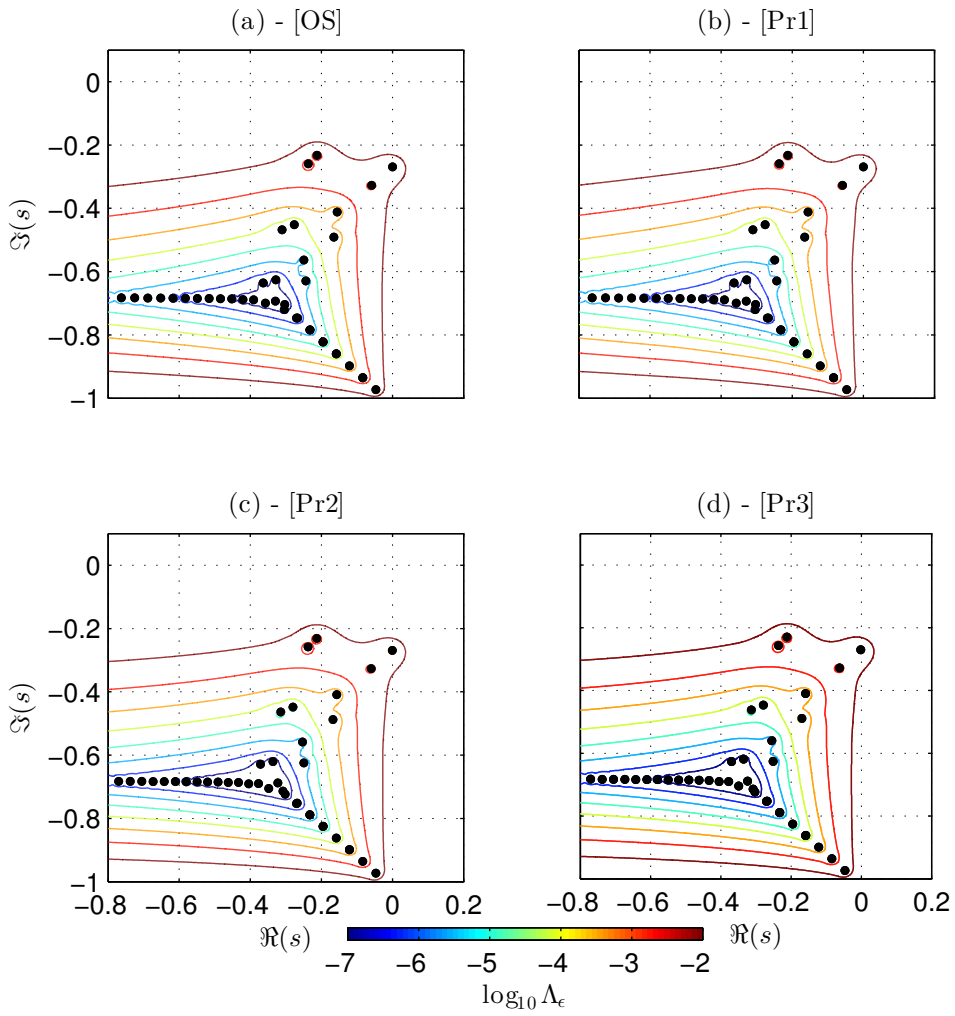


Figure 5.8: Energy weighted pseudospectra contour plots: (a) [OS]; (b) [Pr1]; (c) [Pr2]; (d) [Pr3].

quently used to redefine $\mathscr{Q}$ in (5.38) for use with (5.41) and (5.42) accordingly:

$$\mathscr{Q}_{\mathrm{Pr}} = \mathcal{S}_{\mathrm{Pr}}^{\top}\mathscr{Q}\mathcal{S}_{\mathrm{Pr}}, \tag{5.43}$$

and

$$\mathscr{Q}_{\mathrm{PPE}} = \mathcal{S}_{\mathrm{PPE}}^{\top}\mathscr{Q}\mathcal{S}_{\mathrm{PPE}}. \tag{5.44}$$

From (5.43) and (5.44), the matrices C_{Pr} and C_{PPE} are computed and employed in a similar fashion to C_{OS} in forming appropriate energy weighting matrices.

Having constructed input and output matrices B and C, such that the models are appropriately energy weighted, pseudospectra were computed and are displayed as contour plots in Figures 5.8 and 5.9. Figure 5.8 shows that all three primitive

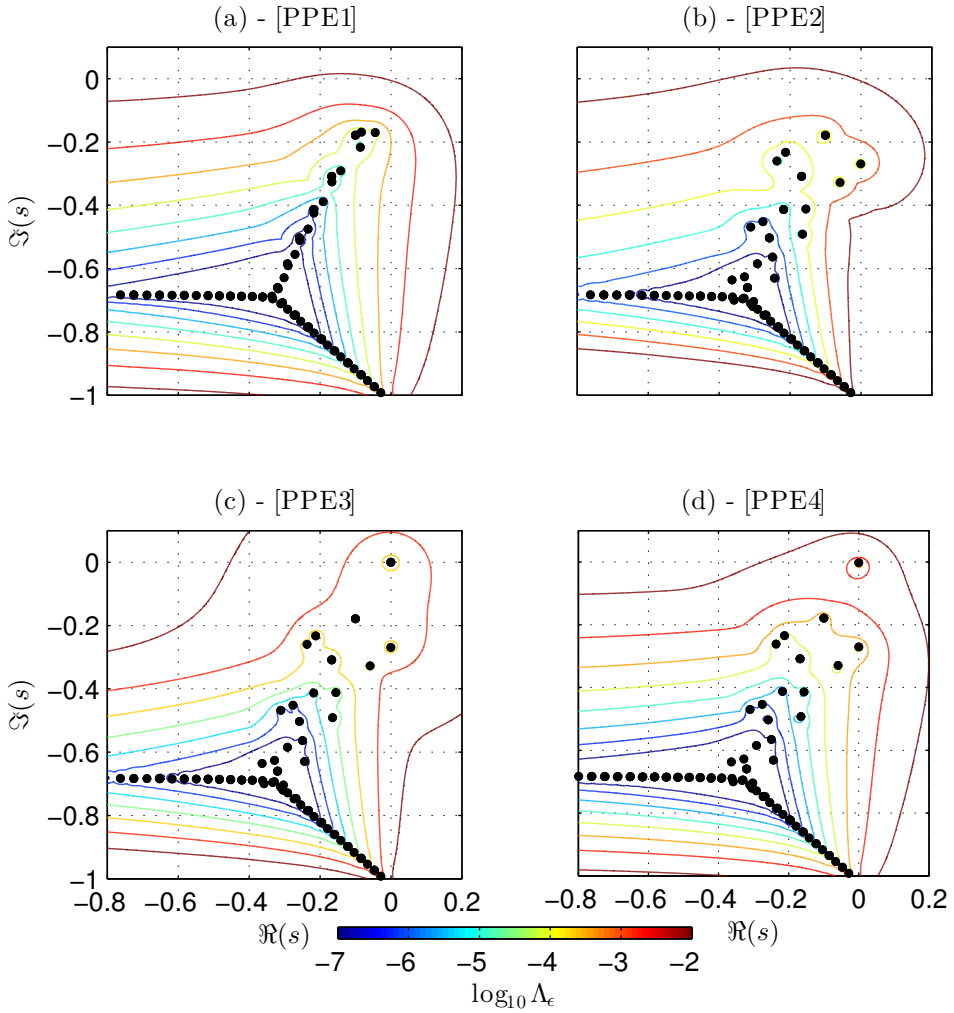


Figure 5.9: Energy weighted pseudospectra contour plots: (a) [PPE1]; (b) [PPE2]; (c) [PPE3]; (d) [PPE4].

variables models, [Pr1]–[Pr3], have the same pseudospectra as the Orr-Sommerfeld model [OS], implying similar transient behaviour. On the other hand, Figure 5.9 shows that all four PPE formulations [PPE1]–[PPE4] exhibit pseudospectra that not only differ markedly from [OS], but also differ from one another. This is understandable in the cases of [PPE1] and [PPE2], owing to their differing pole locations. However, [PPE3] and [PPE4] share the same set of poles and only differ with respect to method of spatial discretisation, and so one would expect their pseudospectra to converge upon grid refinement. This is not the case however, owing to numerical ill-conditioning of the system matrices, particularly with respect to [PPE3]. For example, and with respect to (5.35), the condition number of $\left(sI - \check{A}\right)$ for [PPE3] with $s = 0.1$ (a region where significant discrepancy exists) is $\mathcal{O}\left(10^7\right)$ greater than that for [PPE4].

In summary, the poles, eigenfunctions, and pseudospectra reveal significant differences in the dynamics of different system formulations, pointing to significantly different

open-loop behaviour. However, the question remains, what effect do these differences have when using the system formulations for feedback control design? At this point it is stressed again that models that are suitable for simulation are not necessarily suitable for control, and vice-versa [59, 9, 56]. In order to gain insight into this, actuation and sensing were defined for the 2D channel flow in order to determine the system zeros, and compare the frequency responses from actuation to sensing.

5.5.6 Applying actuation and sensing

In what follows, the choice of system actuation and measurement is somewhat arbitrary but serves to illustrate some of the subtleties of model selection for the purposes of feedback control design. As in many previous studies (see, e.g. [68, 56, 46]), actuation in the form of wall transpiration was assumed, in this case at the upper ($\mathsf{y} = +1$) wall. The actuator was modelled as a first-order system:

$$\frac{\partial}{\partial t}\hat{\mathsf{v}}(+1,t) = -\frac{1}{\tau}\hat{\mathsf{v}}(+1,t) + \frac{1}{\tau}u(t), \tag{5.45}$$

where $u(t) \in \mathbb{R}$ is the control input signal, and a time constant $\tau = 1$ was assumed for convenience. Including wall transpiration in each of the primitive variables and PPE formulations simply amounts to altering the rows, corresponding to the $\hat{\mathsf{v}}(+1,t)$ state, of the system matrices A and E, and defining an input matrix $B \in \mathbb{R}^{n_\mathsf{u}+n_\mathsf{v}+n_\mathsf{p}}$. For the Orr-Sommerfeld formulation, however, owing to the 4^{th}-order spatial derivatives present in (5.22) a lifting procedure based on that presented in [68] was used to incorporate the effects of actuation. The first step to employing this lifting procedure was to write the Fourier transformed wall-normal velocity as:

$$\hat{\mathsf{v}}(\mathsf{y},t) := \underbrace{\hat{\mathsf{v}}_\mathrm{h}(\mathsf{y},t)}_{\text{homogeneous part}} + \underbrace{\hat{\mathsf{v}}_{+1}(t)f_\mathrm{u}(\mathsf{y})}_{\text{inhomogeneous part}}, \tag{5.46}$$

where $\hat{\mathsf{v}}_\mathrm{h}(\mathsf{y},t)$ is the Fourier transformed wall-normal velocity for $\mathsf{y} \in [-1,1)$, $\hat{\mathsf{v}}_{+1}(t) := \hat{\mathsf{v}}(+1,t)$, and $f_\mathrm{u}(\mathsf{y})$ is a *lifting function*. A lifting function is any polynomial satisfying the boundary conditions:

$$f_\mathrm{u}(+1) = 1, \tag{5.47a}$$

$$f_\mathrm{u}(-1) = \frac{\mathrm{d}f_\mathrm{u}(\pm 1)}{\mathrm{d}\mathsf{y}} = 0. \tag{5.47b}$$

In this work, the following lifting function was chosen:

$$f_\mathrm{u}(\mathsf{y}) = \frac{1}{4}\left(2\mathsf{y}^4 - \mathsf{y}^3 - 4\mathsf{y}^2 + 3\mathsf{y} + 4\right). \tag{5.48}$$

From (5.24) and (5.46) one then obtains the expression:

$$\mathcal{E}_{\mathrm{h}}\frac{\partial}{\partial t}\hat{\mathsf{v}}_{\mathrm{h}}(\mathsf{y},t)+\mathcal{E}_{+1}f_{\mathrm{u}}(\mathsf{y})\frac{\mathrm{d}}{\mathrm{d}t}\hat{\mathsf{v}}_{+1}(t)=\mathcal{A}_{\mathrm{h}}\hat{\mathsf{v}}_{\mathrm{h}}(\mathsf{y},t)+\mathcal{A}_{+1}f_{\mathrm{u}}(\mathsf{y})\hat{\mathsf{v}}_{+1}(t), \tag{5.49}$$

where $\mathcal{E}_{\mathrm{h}}$ and $\mathcal{E}_{+1}$ are, respectively, the homogeneous Dirichlet and inhomogeneous versions of $\mathcal{E}$, and $\mathcal{A}_{\mathrm{h}}$ and $\mathcal{A}_{+1}$ are, respectively, the homogeneous Dirichlet and inhomogeneous versions of $\mathcal{A}$.

Inserting (5.45) yields:

$$\begin{aligned}\mathcal{E}_{\mathrm{h}}\frac{\partial}{\partial t}\hat{\mathsf{v}}_{\mathrm{h}}(\mathsf{y},t)-\frac{1}{\tau}\mathcal{E}_{+1}f_{\mathrm{u}}(\mathsf{y})\hat{\mathsf{v}}_{+1}(t)+\frac{1}{\tau}\mathcal{E}_{+1}f_{\mathrm{u}}(\mathsf{y})u(t)&\\ =\mathcal{A}_{\mathrm{h}}\hat{\mathsf{v}}_{\mathrm{h}}(\mathsf{y},t)+\mathcal{A}_{+1}f_{\mathrm{u}}(\mathsf{y})\hat{\mathsf{v}}_{+1}(t),&\end{aligned} \tag{5.50}$$

and therefore:

$$\begin{aligned}\frac{\partial}{\partial t}\hat{\mathsf{v}}_{\mathrm{h}}(\mathsf{y},t)=&\underbrace{\mathcal{E}_{\mathrm{h}}^{-1}\mathcal{A}_{\mathrm{h}}}_{\check{\Xi}_{11}}\hat{\mathsf{v}}_{\mathrm{h}}(\mathsf{y},t)\\ &+\underbrace{\mathcal{E}_{\mathrm{h}}^{-1}\left(\mathcal{A}_{+1}f_{\mathrm{u}}(\mathsf{y})+\frac{1}{\tau}\mathcal{E}_{+1}f_{\mathrm{u}}(\mathsf{y})\right)}_{\check{\Xi}_{12}}\hat{\mathsf{v}}_{+1}(t)\underbrace{-\frac{1}{\tau}\mathcal{E}_{\mathrm{h}}^{-1}\mathcal{E}_{+1}f_{\mathrm{u}}(\mathsf{y})}_{\check{\Gamma}_1}u(t).\end{aligned} \tag{5.51}$$

After spatially discretising in y using Chebyshev interpolants, this can be written in the following matrix form, which serves as the final model state equation:

$$\underbrace{\frac{\mathrm{d}}{\mathrm{d}t}\begin{bmatrix}\vec{\hat{\mathsf{v}}}_{\mathrm{h}}(t)\\ \hat{\mathsf{v}}_{+1}(t)\end{bmatrix}}_{\mathbf{x}(t)}=\underbrace{\begin{bmatrix}\Xi_{11}&\Xi_{12}\\ 0&-\frac{1}{\tau}\end{bmatrix}}_{A_{\mathrm{OS}}}\begin{bmatrix}\vec{\hat{\mathsf{v}}}_{\mathrm{h}}(t)\\ \hat{\mathsf{v}}_{+1}(t)\end{bmatrix}+\underbrace{\begin{bmatrix}\Gamma_1\\ 0\end{bmatrix}}_{B_{\mathrm{OS}}}u(t), \tag{5.52}$$

where $\Xi_{11}\in\mathbb{C}^{n_{\mathsf{y}}\times n_{\mathsf{y}}}$, $\Xi_{12}\in\mathbb{C}^{n_{\mathsf{y}}\times 1}$, and $\Gamma_1\in\mathbb{C}^{n_{\mathsf{y}}}$ are spatially discretised versions of $\check{\Xi}_{11}$, $\check{\Xi}_{12}$, and $\check{\Gamma}_1$, respectively. The state vector is denoted $\mathbf{x}(t)\in\mathbb{C}^{n_{\mathsf{y}}+1}$.

The measured output $y(t)\in\mathbb{C}$ was chosen to be the Fourier transformed pressure at the lower wall:

$$y(t)=\hat{\mathsf{p}}(-1,t). \tag{5.53}$$

For the primitive variables and PPE formulations discretised on a co-located mesh, the corresponding output matrices $C\in\mathbb{C}^{1\times(n_{\mathsf{u}}+n_{\mathsf{v}}+n_{\mathsf{p}})}$ are simply all zeros except for a one in the element corresponding to the $\hat{\mathsf{p}}(-1,t)$ state.

For staggered mesh discretisations, since pressure nodes do not sit on the domain boundaries, linear extrapolation was used to approximate $\hat{\mathsf{p}}(-1,t)$ based on the values

of $\hat{\mathsf{p}}(\mathsf{y},t)$ and $\frac{\partial}{\partial \mathsf{y}}\hat{\mathsf{p}}(\mathsf{y},t)$ at the closest node to the wall. With respect to Figure 5.5(b), this approximation is as follows:

$$\hat{\mathsf{p}}(-1,t) \approx \frac{3}{2}\hat{\mathsf{p}}_{n-\frac{1}{2}}(t) - \frac{1}{2}\hat{\mathsf{p}}_{n-1-\frac{1}{2}}(t). \tag{5.54}$$

Therefore, the output matrices $C \in \mathbb{C}^{1\times(n_{\mathsf{u}}+n_{\mathsf{v}}+n_{\mathsf{p}})}$ are simply all zeros except for the elements corresponding to the $\hat{\mathsf{p}}\left(-1+\frac{\delta}{2},t\right)$ and $\hat{\mathsf{p}}\left(-1+\frac{3\delta}{2},t\right)$ states.

Finally, for the Orr-Sommerfeld system, pressure must be written as a function of $\hat{\mathsf{v}}(\mathsf{y},t)$, since it does not appear in the state vector. Considering the Fourier transformed in x version of (5.20a) which reads as:

$$\begin{aligned}\frac{\partial}{\partial t}\hat{\mathsf{u}}(\mathsf{y},t) + \mathrm{i}\alpha\left(1-\mathsf{y}^2\right)\hat{\mathsf{u}}(\mathsf{y},t) - 2\mathsf{y}\hat{\mathsf{v}}(\mathsf{y},t) = \\ -\mathrm{i}\alpha\hat{\mathsf{p}}(\mathsf{y},t) + \frac{1}{\mathrm{Re}}\left(-\alpha^2 + \frac{\partial^2}{\partial \mathsf{y}^2}\right)\hat{\mathsf{u}}(\mathsf{y},t),\end{aligned} \tag{5.55}$$

and using the fact that $\hat{\mathsf{u}}(-1,t) = \hat{\mathsf{v}}(-1,t) = 0$ due to the no-slip boundary condition, this reduces to:

$$\hat{\mathsf{p}}(-1,t) = \frac{1}{\mathrm{i}\alpha}\frac{1}{\mathrm{Re}}\frac{\partial^2}{\partial \mathsf{y}^2}\hat{\mathsf{u}}(-1,t). \tag{5.56}$$

Finally, using the continuity equation (5.20c), this can be written:

$$\hat{\mathsf{p}}(-1,t) = \frac{1}{\alpha^2}\frac{1}{\mathrm{Re}}\frac{\partial^3}{\partial \mathsf{y}^3}\hat{\mathsf{v}}(-1,t). \tag{5.57}$$

Hence, the output matrix $C \in \mathbb{C}^{1\times(n_{\mathsf{y}}+1)}$ is defined as:

$$C = \frac{1}{\alpha^2}\frac{1}{\mathrm{Re}}\begin{bmatrix}\mathcal{D}_{\mathsf{yyy}_{[n_{\mathsf{y}},1:n_{\mathsf{y}}]}} & \frac{\mathrm{d}^3 f_{\mathrm{u}}(-1)}{\mathrm{dy}^3}\end{bmatrix}, \tag{5.58}$$

where $\mathcal{D}_{\mathsf{yyy}_{[n_{\mathsf{y}},1:n_{\mathsf{y}}]}}$ denotes the bottom row of the third order Chebyshev differentiation matrix.

The overall single-input single-output (SISO) systems then have the following forms

$$\frac{\mathrm{d}}{\mathrm{d}t}\vec{\hat{\mathsf{v}}}(t) = A_{\mathrm{OS}}\vec{\hat{\mathsf{v}}}(t) + B_{\mathrm{OS}}u(t), \tag{5.59a}$$

$$y(t) = C_{\mathrm{OS}}\vec{\hat{\mathsf{v}}}(t), \tag{5.59b}$$

$$E_{\mathrm{Pr}}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = A_{\mathrm{Pr}}\mathbf{x}(t) + B_{\mathrm{Pr}}u(t), \tag{5.60a}$$

$$y(t) = C_{\mathrm{Pr}}\mathbf{x}(t), \tag{5.60b}$$

$$E_{\mathrm{PPE}}\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{x}(t) = A_{\mathrm{PPE}}\mathbf{x}(t) + B_{\mathrm{PPE}}u(t), \tag{5.61a}$$

$$y(t) = C_{\mathrm{PPE}}\mathbf{x}(t), \tag{5.61b}$$

with respective transfer functions

$$G_{\mathrm{OS}}(s) = C_{\mathrm{OS}}\left(sI - A_{\mathrm{OS}}\right)^{-1}B_{\mathrm{OS}}, \tag{5.62}$$

$$G_{\mathrm{Pr}}(s) = C_{\mathrm{Pr}}\left(sE_{\mathrm{Pr}} - A_{\mathrm{Pr}}\right)^{-1}B_{\mathrm{Pr}}, \tag{5.63}$$

$$G_{\mathrm{PPE}}(s) = C_{\mathrm{PPE}}\left(sE_{\mathrm{PPE}} - A_{\mathrm{PPE}}\right)^{-1}B_{\mathrm{PPE}}. \tag{5.64}$$

Poles and zeros

The poles and zeros of all model formulations are plotted in Figures 5.10 and 5.11. One should note the addition of a pole located at $s = -1/\tau$ in all cases owing to the presence of actuator dynamics (5.45). As can be seen in Figure 5.10, the zeros are in the same locations for the Orr-Sommerfeld model [OS], and primitive variables formulations [Pr1] and [Pr3], suggesting that these three models will exhibit the same input-output behaviour. However, whilst the majority of the zeros of [Pr2] match those of [OS], the single right half plane zero (located at approximately $s = 0.03 - 1.1\mathrm{i}$) does not match that of [OS]. This immediately suggests that their input-output behaviour differs.

As can be seen in Figure 5.11, the zeros of [PPE1] and [PPE2] do not agree with those of [OS]. One would therefore expect different input-output behaviour from the two systems. It should be noted that when spatially discretising the system models upon successively finer computational meshes, the matrix $(sE_{\mathrm{PPE}} - A_{\mathrm{PPE}})$ of [PPE2] becomes increasingly ill-conditioned, and the computed zeros fail to converge. This was not the case for any other formulation considered, suggesting that directly enforcing the divergence free velocity condition (5.20c) yields inherently ill-conditioned system matrices.

In the cases of [PPE3] and [PPE4], the zeros lie largely in the same locations as

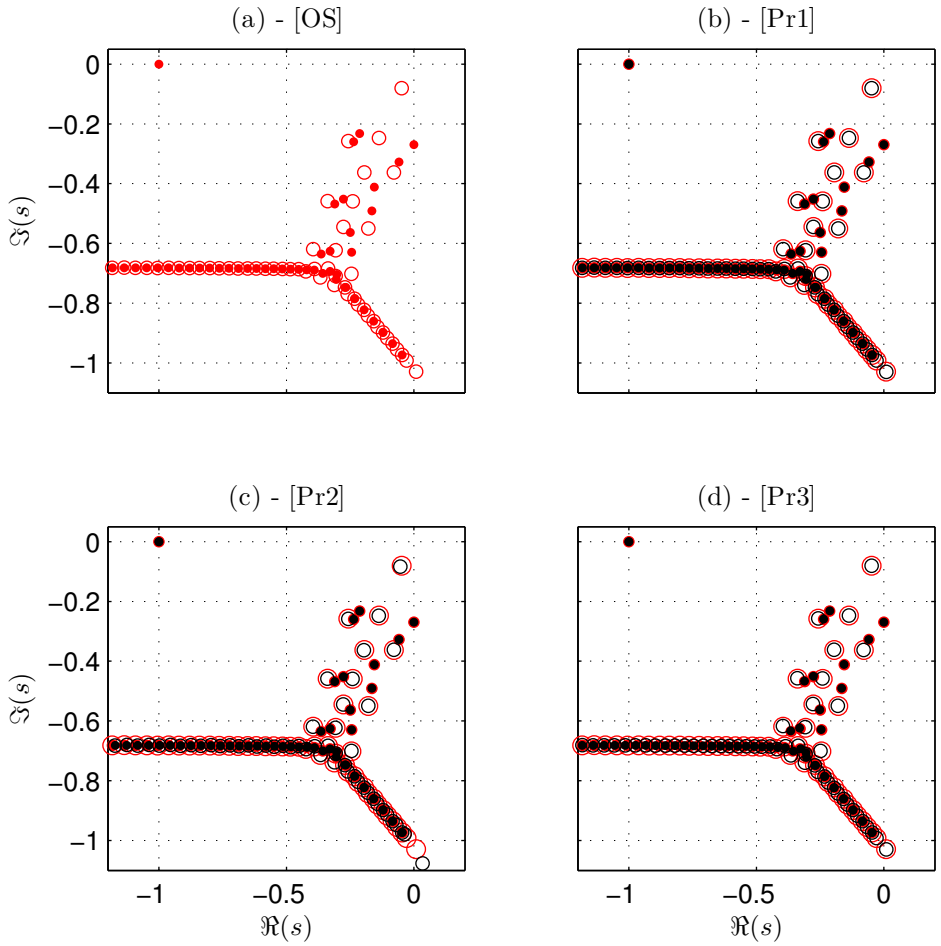


Figure 5.10: Poles (•) and zeros (◯) of primitive variables formulations with poles (•) and zeros (◯) of [OS] shown for comparison: (a) [OS]; (b) [Pr1]; (c) [Pr2]; (d) [Pr3].

those of [OS], but with additional zeros lying in the same locations as the spurious poles (discussed in Section 5.5.4). This results in pole/zero cancellation of the spurious poles in the corresponding transfer functions and so the overall effect is that one obtains a transfer function the same as that of [OS] after computing a minimal realisation [92]. This suggests, therefore, that the two systems will exhibit the same input-output behaviour.

Frequency response

Another way of comparing input-output behaviour between systems is by comparing their frequency responses. Bode plots for each model are shown in Figures 5.12 and 5.13, where the frequency response of the Orr-Sommerfeld formulation [OS] is also plotted for comparison. As can be seen in Figure 5.12, the frequency responses of [Pr1] and [Pr3]

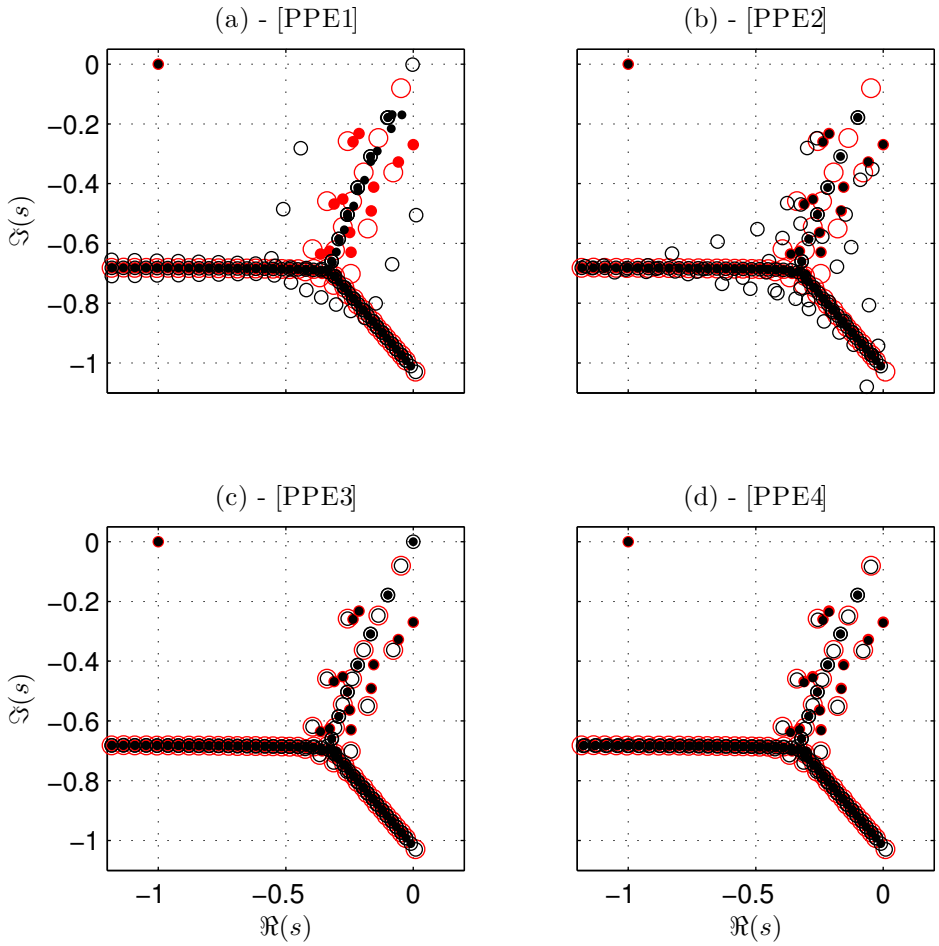


Figure 5.11: Poles (•) and zeros (◯) of PPE formulations with poles (•) and zeros (◯) of [OS] shown for comparison: (a) [PPE1]; (b) [PPE2]; (c) [PPE3]; (d) [PPE4].

match that of [OS] exactly, whilst there is an offset at low frequencies for [Pr2] due to the different location of the right half plane zero. This again exposes the problems associated with using finite-difference discretisation on a co-located mesh, and also confirms that such a formulation and discretisation does not yield a model suitable for controller design as the frequency response is incorrect. The use of a staggered mesh with finite-differences does indeed solve this problem. Figure 5.13 shows that both [PPE3] and [PPE4] have frequency response identical to that of [OS], suggesting that the PPE formulation along with pressure boundary conditions based on those discussed in [91] can produce models suitable for controller design, even when using finite-difference discretisation on a co-located mesh.

The frequency response of [PPE1] shows very little agreement with that of [OS], suggesting that the PPE formulation in conjunction with the simple Neumann pressure boundary condition (5.15) does not yield a model that is at all representative of

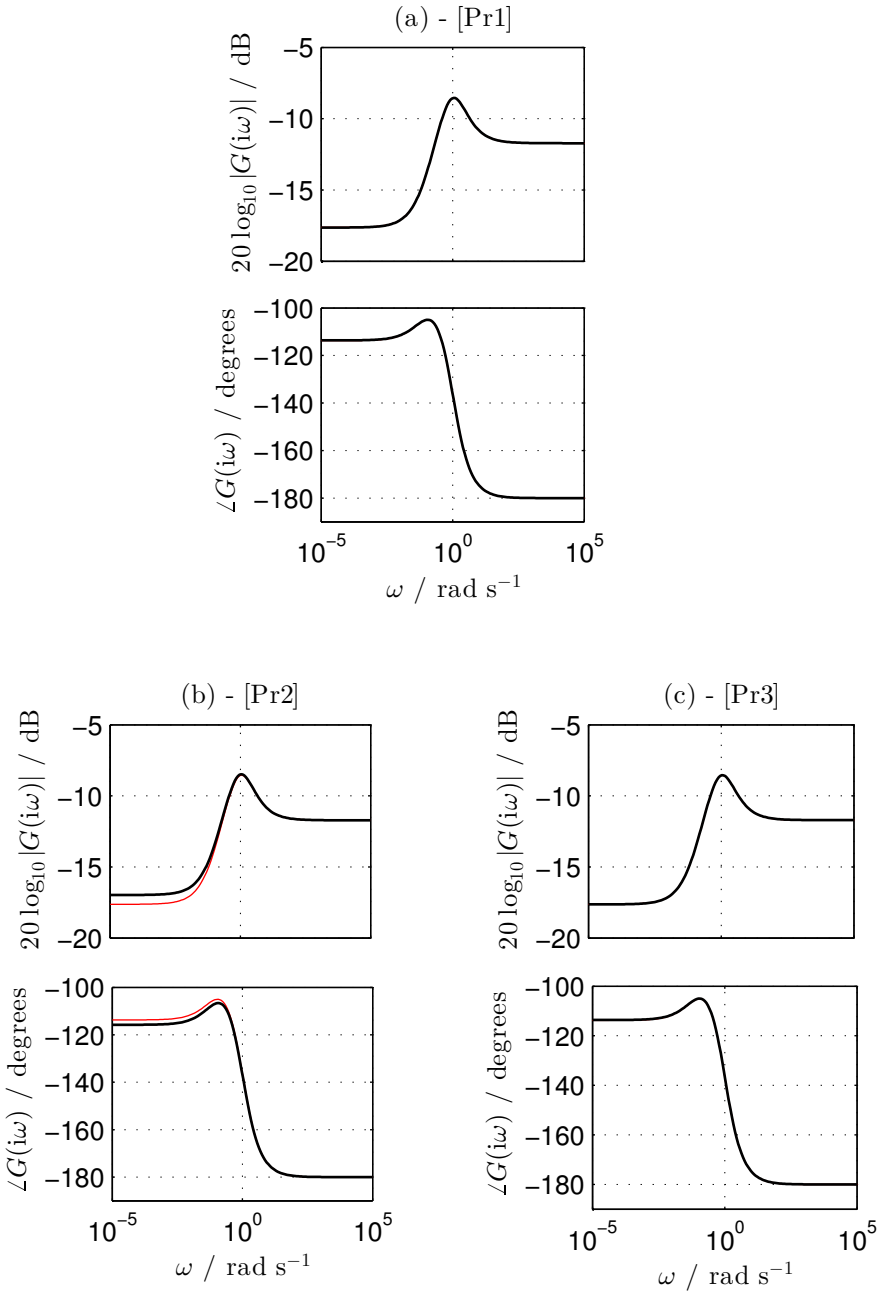


Figure 5.12: Bode plots of primitive variables formulation systems (−) with frequency response of $G_{\mathrm{OS}}(s)$ shown for comparison (−): (a) [Pr1]; (b) [Pr2]; (c) [Pr3]. Note that in (a) and (c) the red line is not visible as the frequency responses agree exactly.

the actual system dynamics. Finally, whilst the frequency response of [PPE2] follows the general shape of that of [OS], it is unphysical owing to the occurrence of spurious poles and zeros. This suggests that simply enforcing the divergence free velocity con-

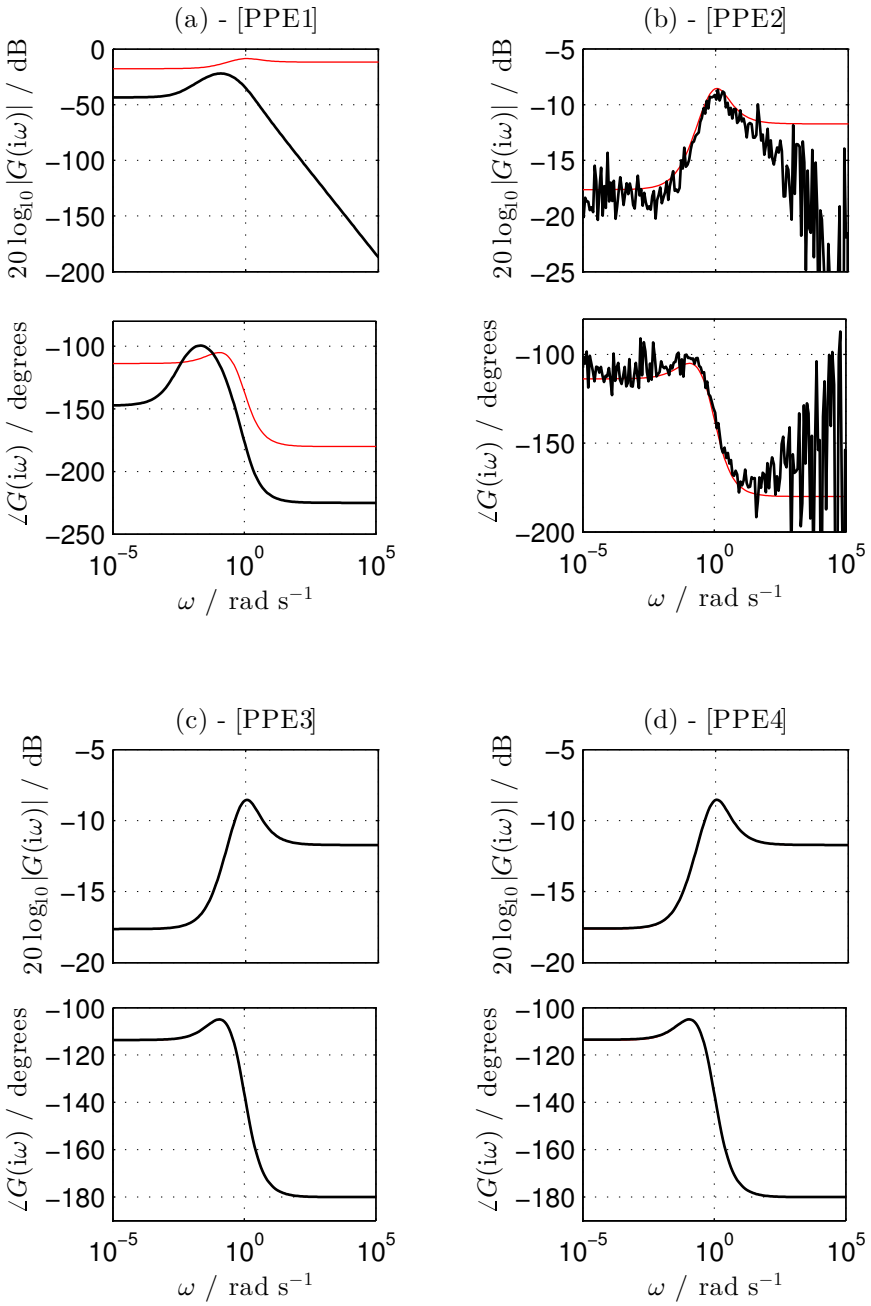


Figure 5.13: Bode plots of PPE formulation systems (−) with frequency response of $G_{\mathrm{OS}}(s)$ shown for comparison (−): (a) [PPE1]; (b) [PPE2]; (c) [PPE3]; (d) [PPE4]. Note that in (c) and (d) the red line is not visible as the frequency responses agree exactly.

dition (5.20c) on the boundary does not necessarily result in a suitable control model.

[Pr1]	[Pr2]	[Pr3]	[PP1]	[PPE2]	[PPE3]	[PPE4]
0.0002	0.0112	0.0005	0.3404	0.2596	0.0002	0.0007

Table 5.2: ν-gap between different 2D channel flow model formulations and the benchmark Orr-Sommerfeld model.

ν-gap metric

A final way of comparing the suitability of the different models for feedback control design is by considering the ν-gap metric between each of the different formulations and the benchmark Orr-Sommerfeld model. The ν-gap metric $\delta_\nu(\cdot,\cdot) : \mathscr{R}^{m\times n} \times \mathscr{R}^{m\times n} \to [0,1]$ is a measure of the 'distance' between two systems in a closed-loop sense, where a value of 0 indicates identical closed-loop behaviour, and a value of 1 indicates the opposite [98]. In terms of frequency response, the ν-gap between two systems $G_1(s)$ and $G_2(s)$ is defined as [97]:

$$\delta_\nu\left(G_1(s),G_2(s)\right) := \begin{cases} \left\|\Upsilon\left(G_1(s),G_2(s)\right)\right\|_\infty & \text{if } \det\left(I+G_2^*(\mathrm{i}\omega)G_1(\mathrm{i}\omega)\right)\neq 0,\ \forall\omega\in\mathbb{R} \\ & \text{and } \operatorname{wno}\det\left(I+G_2^*(s)G_1(s)\right) \\ & \qquad +\eta\left(G_1(s)\right)-\eta\left(G_2(s)\right)=0, \\ 1 & \text{otherwise,} \end{cases} \tag{5.65}$$

where $\Upsilon(\cdot,\cdot) : \mathscr{R}^{m\times n} \times \mathscr{R}^{m\times n} \to \mathscr{R}^{m\times n}$ is defined as:

$$\begin{aligned}\Upsilon\left(G_1(s),G_2(s)\right) = {} & \left(I+G_2(\mathrm{i}\omega)G_2^*(\mathrm{i}\omega)\right)^{-1/2} \\ & \times\left(G_2(\mathrm{i}\omega)-G_1(\mathrm{i}\omega)\right)\left(I+G_1^*(\mathrm{i}\omega)G_1(\mathrm{i}\omega)\right)^{-1/2},\end{aligned} \tag{5.66}$$

the $\mathscr{H}_\infty$-norm is denoted $\|\cdot\|_\infty$, $\eta\left(G(s)\right) \in \mathbb{N}$ denotes the number of open right-half-plane poles of $G(s)$, and $\operatorname{wno}\left(g(s)\right) \in \mathbb{N}$ denotes the winding number of the scalar transfer frunction $g(s)$ evaluated on the standard Nyquist contour.

The ν-gaps between each model formulation and the benchmark Orr-Sommerfeld model were computed, and results are presented in Table 5.2.

For models [Pr1], [Pr3], [PPE3], and [PPE4], the ν-gaps were very close to 0, and were significantly larger for [Pr2], [PPE1], and [PPE2]. It is interesting to note that whilst the frequency response of [Pr2] is broadly similar to that of the Orr-Sommerfeld model, the offset at low frequencies results in a ν-gap that is orders of magnitude greater than that of the models which agreed exactly. This is due to the differing location of the right half plane zero, and models arising from spatial discretisations that produce such spurious right half plane zeros are not suitable for feedback control design. This

confirms our previous analyses.

5.6 Summary

In this chapter, a number of different formulations of the linearised Navier-Stokes equations have been compared to one another in terms of their dynamic response, with a view towards identifying formulations that are suitable for designing feedback controllers.

Models of a 2D channel flow were constructed, and the dynamics and input-output behaviour compared to a benchmark Orr-Sommerfeld model. This showed that when using centred finite-differences, the only formulation which was dynamically identical to the Orr-Sommerfeld model was the primitive variables formulation discretised on a staggered mesh. This yielded the same poles, zeros, psuedospectra, and frequency response, and is thus a recommended choice as the basis for designing feedback controllers, particularly for flows around complex geometries.

On the other hand, none of the models based on PPE formulations yielded the same underlying dynamics, which was clearly exposed by their differing poles and pseudospectra. However, in some cases the frequency response obtained upon defining an input and output to the system agreed with that of the Orr-Sommerfeld model, since the zeros caused pole/zero cancellation of spurious poles. This implies that some PPE-based formulations may be suitable for feedback control design. However, as the zero locations are dependent on how the inputs and outputs are defined, there is no guarantee that such pole/zero cancellation would occur for different sensing and actuation configurations, and so considerable care is advised.

Chapter 6

Modelling backward facing step flow

In this chapter, the RHSP-based construction method presented in Chapter 4 is used, along with insight gained in Chapter 5, to construct the frequency response of the 2D backward facing step flow that was introduced in Chapter 3.

Details of the spatial domain used for the model, how actuation and sensing was defined, and the specific formulation and spatial discretisation of the governing equations used are discussed, before the frequency response obtained using the RHSP-based construction method is presented, and low-order transfer functions are fitted.

Results are compared with those obtained using a CFD-based system identification study, which reveals discrepancies. These discrepancies are explained by investigation of the underlying dynamics of the CFD solver algorithm employed.

6.1 Formulating the model

6.1.1 Model spatial domain

The computational domain used here was much the same as that used for simulation purposes (see Figure 3.2), but with a shorter inlet length. Much of the inlet was discarded since, due to the convective nature of the flow, it was assumed that ZNMF actuation located on the step rear edge would have no effect on the upstream flow. As such, including this region in the model would add unnecessary computational cost. The domain used is depicted in Figure 6.1.

Boundary conditions for the fluctuating parts of flow variables were based on those

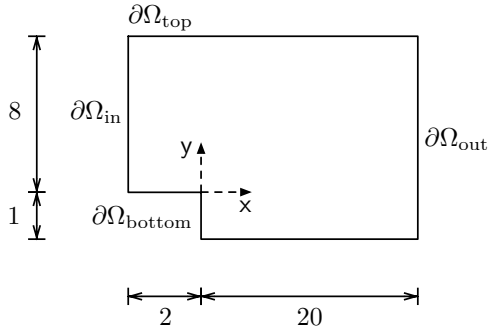


Figure 6.1: Backward facing step flow RHSP-based construction method computational domain with boundaries $\partial\Omega_{(\cdot)}$.

boundary	velocity
$\partial\Omega_{\text{in}}$	$(\mathsf{u}', \mathsf{v}') = (0, 0)$
$\partial\Omega_{\text{out}}$	$\partial\mathsf{u}'/\partial\mathbf{n} = 0$
$\partial\Omega_{\text{bottom}}$	$(\mathsf{u}', \mathsf{v}') = (0, 0)$ (no-slip)
$\partial\Omega_{\text{top}}$	$\mathsf{v}' = \partial\mathsf{u}'/\partial\mathbf{n} = 0$ (free-slip)

Table 6.1: Backward facing step flow RHSP-based construction method fluctuating variables boundary conditions.

used for simulations, and are described in Table 6.1.

6.1.2 Individual computational node subsystem

Informed by the analysis presented in Chapter 5, the primitive variables formulation of the linearised Navier-Stokes equations discretised on a staggered mesh was chosen for modelling the backward facing step flow. Upon spatially discretising the linearised momentum (5.2a) and continuity (5.2b) equations using centred finite-differences, the following descriptor state-space representation was obtained for each individual computational node:

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{E_{i,j}} \frac{\mathrm{d}}{\mathrm{d}t} \begin{bmatrix} \mathsf{u}'_{i+1/2,j}(t) \\ \mathsf{v}'_{i,j+1/2}(t) \\ \mathsf{p}'_{i,j}(t) \end{bmatrix} = A_{i,j} \underbrace{\begin{bmatrix} \mathsf{u}'_{i+1/2,j}(t) \\ \mathsf{v}'_{i,j+1/2}(t) \\ \mathsf{p}'_{i,j}(t) \end{bmatrix}}_{\mathbf{x}_{i,j}(t)} + B_{i,j}\boldsymbol{\xi}_{i,j}(t), \tag{6.1a}$$

$$\mathbf{z}_{i,j}(t) = \underbrace{\begin{bmatrix} I & I & I & I \end{bmatrix}^{\top}}_{C_{i,j}} \mathbf{x}_{i,j}(t), \tag{6.1b}$$

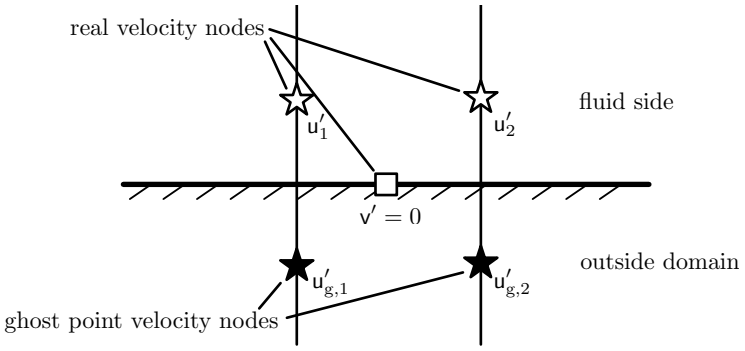


Figure 6.2: Location of ghost points on computational domain boundary.

where $\boldsymbol{\xi}_{i,j}(t) \in \mathbb{R}^{12}$and $\mathbf{z}_{i,j}(t) \in \mathbb{R}^{12}$ are defined as in Section 4.1.1, the state vector $\mathbf{x}_{i,j}(t) \in \mathbb{R}^3$ is defined as in Section 5.3, $A_{i,j} \in \mathbb{R}^{3\times3}$, $C_{i,j} \in \mathbb{R}^{12\times3}$, and:

$$B_{i,j} := \begin{bmatrix} B_{1_{i,j}} & B_{2_{i,j}} & B_{3_{i,j}} & B_{4_{i,j}} \end{bmatrix} \in \mathbb{R}^{3\times12}. \tag{6.2}$$

The elements of $A_{i,j}$ and $B_{i,j}$ are defined in Appendix A. The mean flowfield $(\bar{\mathbf{u}}(\mathbf{x}), \bar{\mathsf{p}}(\mathbf{x}))$ was obtained by time-averaging an unforced LES simulation of the flow, however this could be obtained at a lesser computational cost using RANS simulations.

As in the case of the 2D wave-diffusion equation, the subsystems describing boundary nodes were altered as required in order to enforce the boundary conditions. Due to the use of a staggered mesh arrangement, the use of *ghost points* – ficticious computational nodes outside the computational domain – was required for some boundaries in order to enforce velocity boundary conditions [45]. With respect to the example boundary in Figure 6.2, for a no-slip boundary the ghost point velocities must be set such that $\mathsf{u}'_{g,1} = -\mathsf{u}'_1$, $\mathsf{u}'_{g,2} = -\mathsf{u}'_2$, etc, whilst for a free-slip boundary they must be set such that $\mathsf{u}'_{g,1} = +\mathsf{u}'_1$, $\mathsf{u}'_{g,2} = +\mathsf{u}'_2$, etc. Analogous conditions can easily be derived for vertical boundaries.

6.1.3 Actuation and sensing

Actuation was defined as the same ZNMF slot jet used in Section 3.3, and implemented in the model by altering the matrices of the subsystems corresponding to the actuation slot. Appropriate $B_{\mathbf{u}_{i,j}}$ matrices were defined as required.

Two different sensing configurations were considered in this work; a pressure fluctuation measurement in the centre of the step's rear face:

$$y_1(t) = \mathsf{p}'(0, -0.5, t), \tag{6.3a}$$

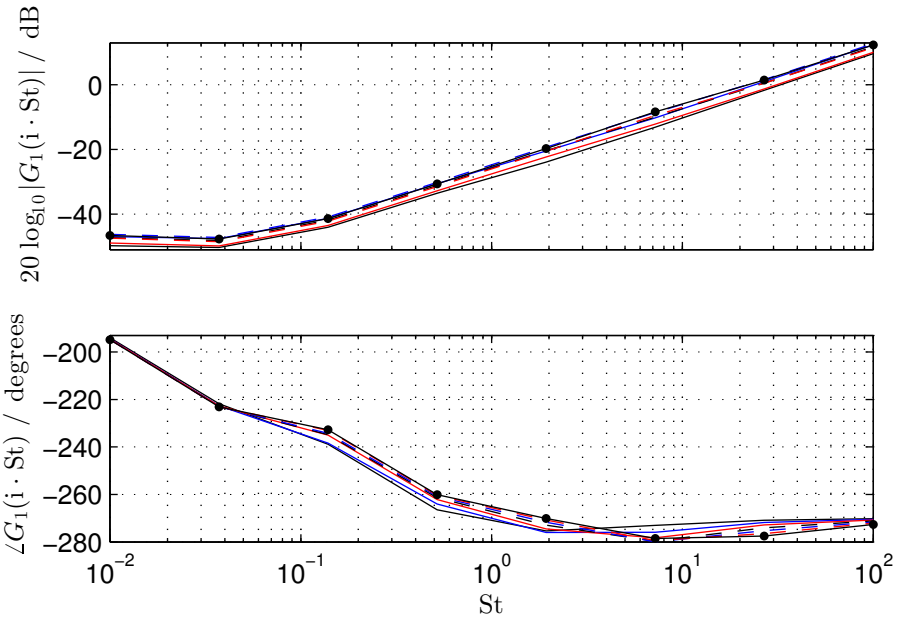


Figure 6.3: Frequency response $G_1(\mathrm{i}\cdot\mathrm{St})$ of backward facing step flow with output $y_1(t)$: $\varrho = 41$ (−), $\varrho = 61$ (−), $\varrho = 81$ (−), $\varrho = 101$ (−−), $\varrho = 121$ (−−), $\varrho = 141$ (−−), $\varrho = 161$ (− • −).

and a wall-normal velocity fluctuation measurement in the wake:

$$y_2(t) = \mathsf{v}'(1, -0.5, t). \tag{6.3b}$$

These outputs were defined by constructing appropriate $C_{\mathbf{y}_{i,j}}$ matrices for the relevant subsystems. It should be noted that since a staggered mesh was used for spatial discretisation, pressure nodes did not lie on the step face boundary. As such, the pressure measurement was estimated using linear extrapolation based on the nearest two pressure nodes, similar to that employed in Section 5.5.6

6.2 Results

6.2.1 Frequency response

Frequency responses for each sensing configuration were computed using the RHSP-based construction method using the ICEBERG HPC cluster in order to parallelise over frequency points. This was carried out for a number of successively finer uniform computational meshes (with mesh density ϱ) in order to ensure convergence. The frequency responses obtained for $y_1(t)$ and $y_2(t)$ are depicted in Figures 6.3 and 6.4, respectively. The vector of frequency response values evaluated at n_{St} frequency points

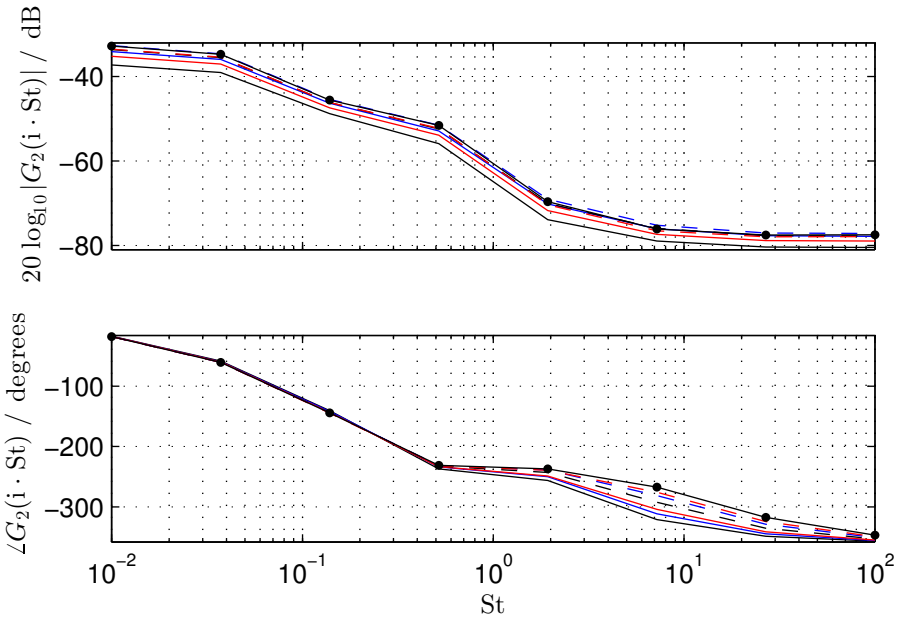


Figure 6.4: Frequency response $G_2(\mathrm{i}\cdot\mathrm{St})$ of backward facing step flow with output $y_2(t)$: $\varrho = 41$ (−), $\varrho = 61$ (−), $\varrho = 81$ (−), $\varrho = 101$ (−−), $\varrho = 121$ (−−), $\varrho = 141$ (−−), $\varrho = 161$ (− • −).

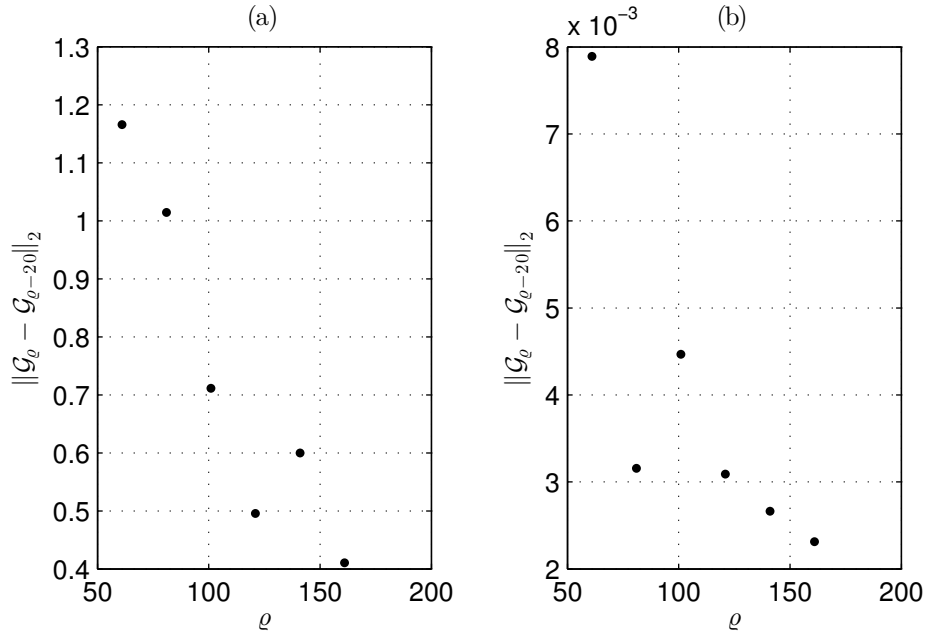


Figure 6.5: 2-norm of $\mathcal{G}_\varrho - \mathcal{G}_{\varrho-20}$ for successively finer meshes (with mesh density ϱ): (a) frequency response $G_1(\mathrm{i}\cdot\mathrm{St})$ for output $y_1(t)$; (b) frequency response $G_2(\mathrm{i}\cdot\mathrm{St})$ for output $y_2(t)$.

for a mesh density ϱ is denoted $\mathcal{G}_\varrho \in \mathbb{C}^{n_{\mathrm{St}}}$, i.e:

$$\mathcal{G}_\varrho := \begin{bmatrix} G(\mathrm{i}\cdot\mathrm{St}_1) & G(\mathrm{i}\cdot\mathrm{St}_2) & \cdots & G(\mathrm{i}\cdot\mathrm{St}_{n_{\mathrm{St}}}) \end{bmatrix}^\top. \tag{6.4}$$

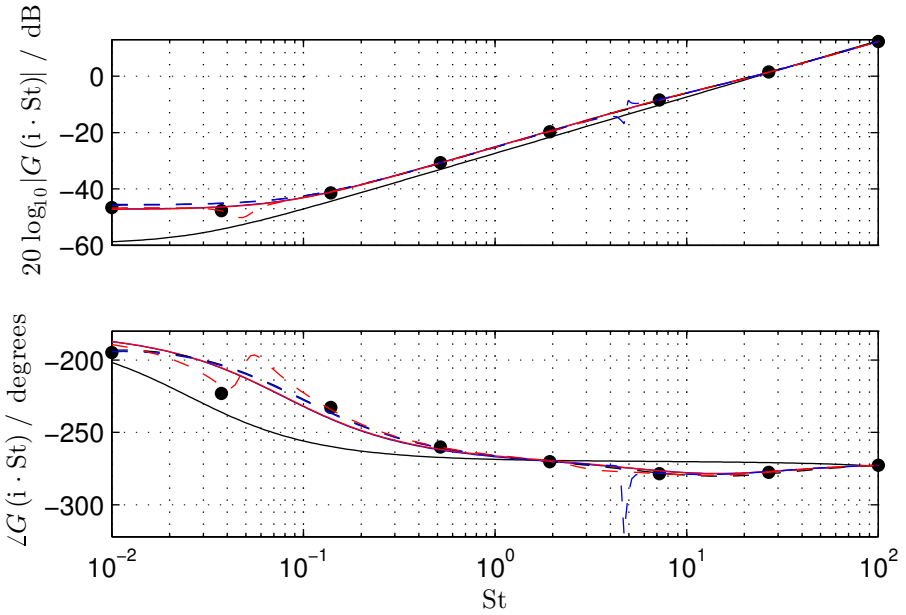


Figure 6.6: Fitting n^{th}-order transfer function $G(s)$ to frequency response data $G_1(\mathrm{i}\cdot\mathrm{St})$ obtained with RHSP-based construction method: original data (•), $n = 1$ fit (−), $n = 2$ fit (−), $n = 3$ fit (−), $n = 4$ fit (−−), $n = 5$ fit (−−), $n = 6$ fit (−−).

The convergence of $G_1(\mathrm{i}\cdot\mathrm{St})$ and $G_2(\mathrm{i}\cdot\mathrm{St})$ is demonstrated in Figure 6.5(a) and 6.5(b), respectively, which shows the 2-norm of $\mathcal{G}_\varrho - \mathcal{G}_{\varrho-20}$ decreasing for successively finer meshes. Based on Figures 6.3, 6.4, and 6.5, the frequency responses were deemed to have converged sufficiently at $\varrho = 161$.

6.2.2 Fitting transfer functions

Transfer functions $G(s)$ were fitted to the frequency response data obtained for the highest mesh density considered ($\varrho = 161$) using least squares regression (implemented in MATLAB's `fitfrd()` routine). The frequency responses of fitted transfer functions of increasing order n are shown in Figures 6.6 and 6.7.

The fit error was quantified here using the root-mean-square-error (RMSE):

$$e_{\text{RMS}} = \sqrt{\frac{1}{n_{\text{St}}}\sum_{k=1}^{n_{\text{St}}} |G_{\text{RHSP}}(\mathrm{i}\cdot\mathrm{St}_k) - G_{\text{fit}}(\mathrm{i}\cdot\mathrm{St}_k)|^2}, \tag{6.5}$$

where $G_{\text{RHSP}}(\mathrm{i}\cdot\mathrm{St}_k)$ are the frequency response values obtained via the RHSP-based construction method, and $G_{\text{fit}}(\mathrm{i}\cdot\mathrm{St}_k)$ are the frequency response values of the fitted transfer function evaluated at the same n_{St} frequencies. The errors of fitted models of increasing order are shown in Figure 6.8.

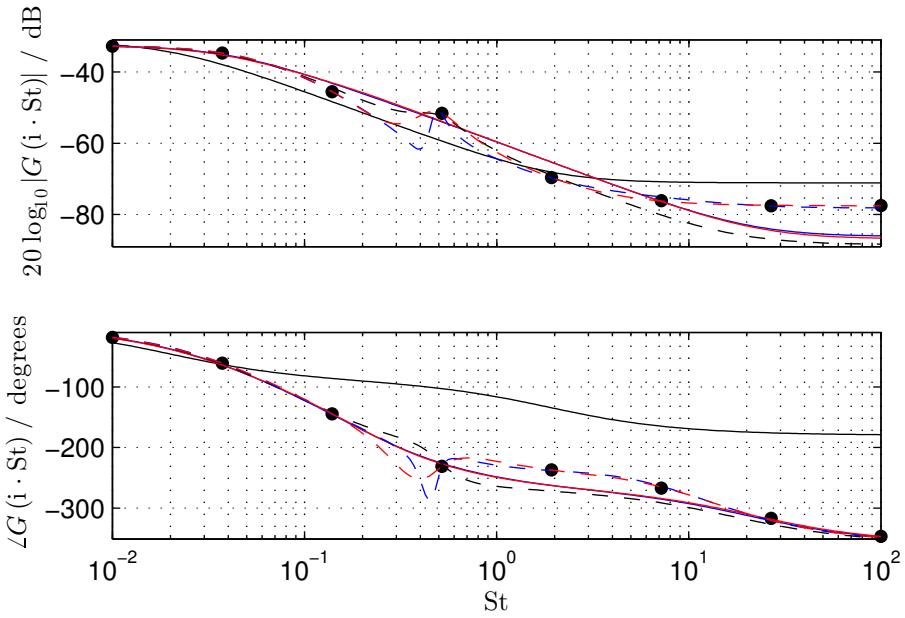


Figure 6.7: Fitting n^{th}-order transfer function $G(s)$ to frequency response data $G_2(\mathrm{i}\cdot\mathrm{St})$ obtained with RHSP-based construction method: original data (•), $n = 1$ fit (−), $n = 2$ fit (−), $n = 3$ fit (−), $n = 4$ fit (− −), $n = 5$ fit (− −), $n = 6$ fit (− −).

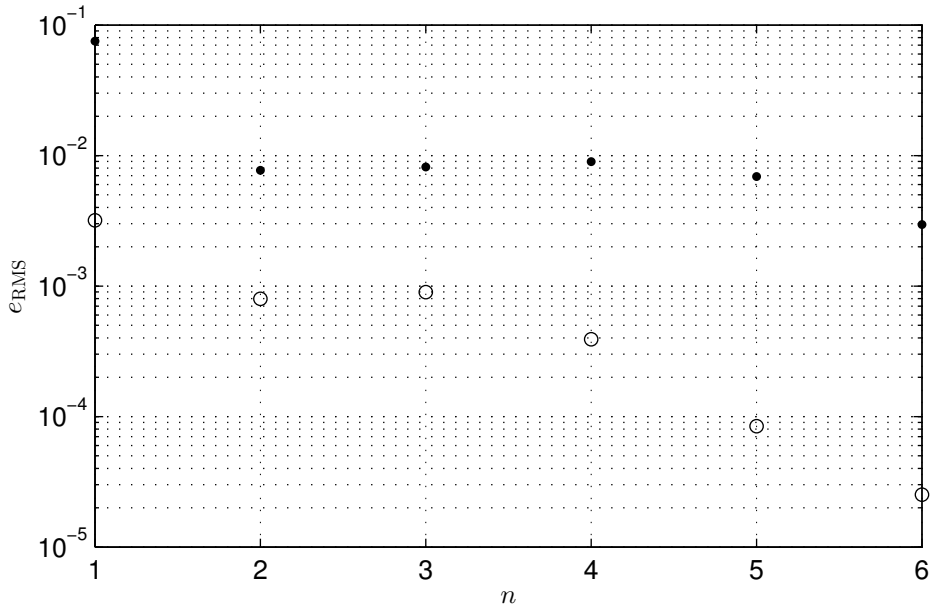


Figure 6.8: e_{RMS} of n^{th}-order transfer functions fitted to frequency response data: $G_1(\mathrm{i}\cdot\mathrm{St})$ (•), $G_2(\mathrm{i}\cdot\mathrm{St})$ (◯).

Based on Figures 6.6, 6.7, and 6.8, it was decided that 4^{th}-order transfer functions could be used to represent both $G_1(\mathrm{i}\cdot\mathrm{St})$ and $G_2(\mathrm{i}\cdot\mathrm{St})$ with sufficient accuracy, whilst

avoiding over-fitting. These transfer functions are given by:

$$G_1(s) = \frac{-6.4 \times 10^2 s^4 - 4.5 \times 10^3 s^3 + 1.1 \times 10^5 s^2 - 1.1 \times 10^4 s - 24}{s^4 - 1.6 \times 10^4 s^3 - 2.7 \times 10^4 s^2 + 2.1 \times 10^6 s + 1.9 \times 10^3}, \tag{6.6}$$

and:

$$G_2(s) = \frac{3.8 \times 10^{-5} s^4 - 6.4 \times 10^{-4} s^3 - 2.2 \times 10^{-4} s^2 - 6.3 \times 10^{-5} s + 3.3 \times 10^{-5}}{s^4 + 4.8 \times 10^{-1} s^3 + 3.1 \times 10^{-1} s^2 + 4.3 \times 10^{-2} s + 1.4 \times 10^{-3}}, \tag{6.7}$$

respectively.

6.2.3 Comparison with system identification

The models obtained using the RHSP-based construction method were compared with models obtained from a CFD-based linear system identification study. System identification was carried out using the subspace identification methods presented in [63] in order to identify n^{th}-order transfer functions (implemented in MATLAB's `n4sid()` routine).

A number of open-loop forced simulations were run with varying excitation signals, including random binary sequences (RBS) and sums-of-sines, with maximum amplitudes corresponding to the region for which, in Section 3.4, it was shown the flow could be modelled as a linear system. For each experiment, the excitation signal was designed such that for the first few hundred time units there was no actuation in order for the subspace identification algorithm to model the natural vortex shedding as an external disturbance to the linear system.

The subspace identification method allows the analysis of the system's Hankel singular values during the identification procedure in order to inform the choice of model order. For both output cases $y_1(t)$ and $y_2(t)$ this suggested a 3^{rd}-order model. The frequency responses of the models identified for $G_1(s)$ and $G_2(s)$ by system identification (from a number of experiments with different excitation signals) are shown in Figures 6.9 and 6.10, respectively. It should be noted that since the simulation data was sampled with a sampling period of 0.05 non-dimensional time units, due to the Nyquist sampling theorem one can only expect to correctly identify frequency content up to $\text{St} = 10$. This is likely the cause of the discrepency between different experiments past $\text{St} \approx 20$ in Figure 6.10.

For each of the outputs, the models identified in different experiments agreed well, however there is significant discrepancy between these identified models and the models identified using the RHSP-based construction method. To rule out this discrepancy being associated solely with the PIMPLE algorithm, the same experiments were car-

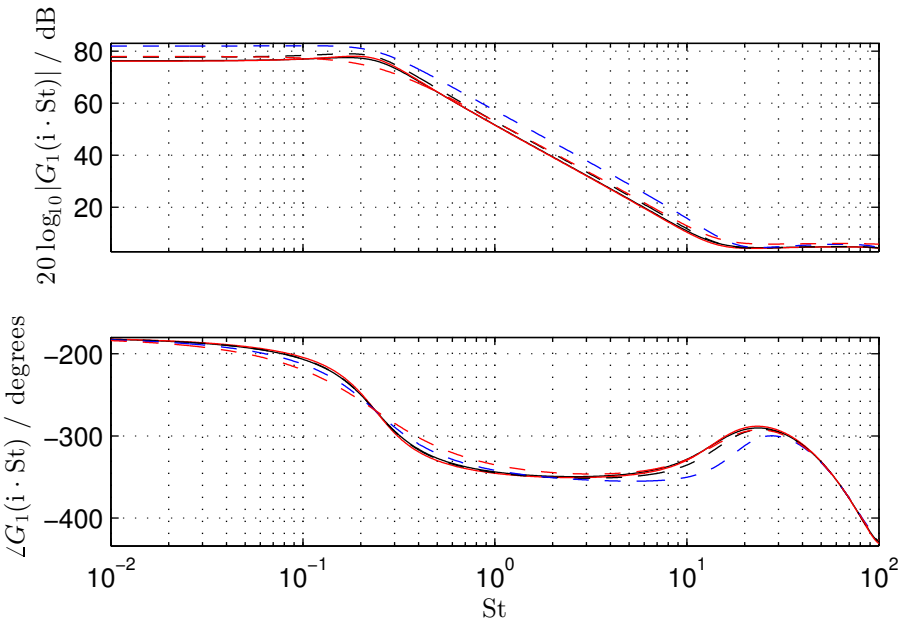


Figure 6.9: Frequency response of transfer function $G_1(s)$ identified using system identification for output $y_1(t)$. Different line styles represent experiments with different excitation signals.

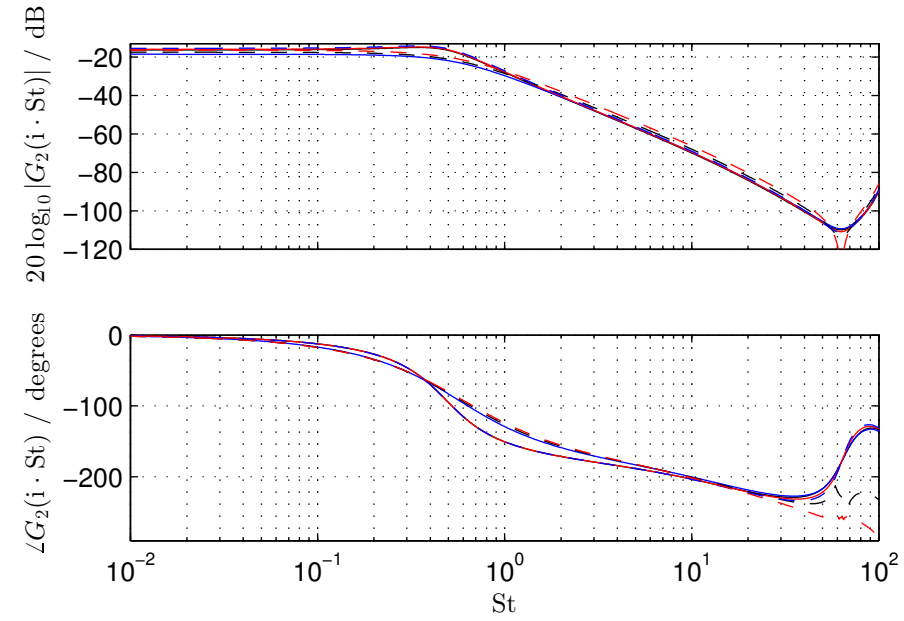


Figure 6.10: Frequency response of transfer function $G_2(s)$ identified using system identification for output $y_2(t)$. Different line styles represent experiments with different excitation signals.

ried out using OpenFOAM's standard PISO algorithm solver with a very small fixed timestep. This, however, yielded almost identical results to the PIMPLE algorithm.

The results obtained using the RHSP-based construction method and system iden-

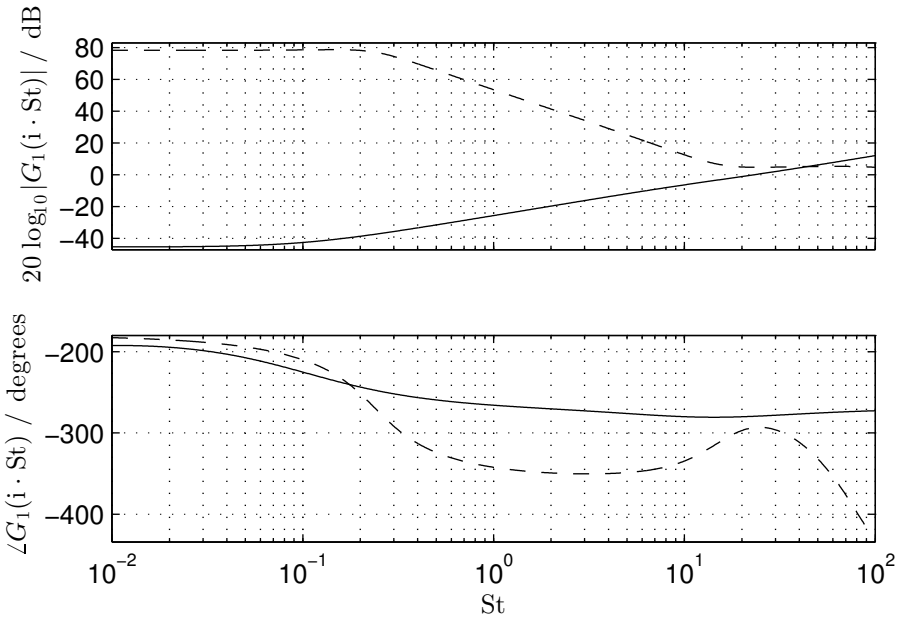


Figure 6.11: Comparison of frequency responses of $G_1(s)$: model obtained using RHSP-based construction method (−), average of models obtained using system identification experiments (−−).

tification method are compared in Figures 6.11 and 6.12, for outputs $y_1(t)$ and $y_2(t)$, respectively.

For output $y_1(t)$, the difference in models is most substantial. Whilst the model obtained using the RHSP-based construction method exhibits high-pass dynamics, the model obtained using system identification displays low-pass dynamics, indicating considerably different behaviour.

In the case of output $y_2(t)$, the differences are less severe, but are existent nonethe-less. Both models exhibit low-pass behaviour, however are quantitatively different.

These significant differences prompted an investigation into the underyling dynamics of the PISO algorithm solver, which is presented in the following section.

6.3 PISO algorithm dynamics

The PISO algorithm, originally described by Issa [54], consists of three steps within each timestep: an initial predictor step, followed by two corrector steps. In order to obtain a discrete-time state-space representation of the PISO algorithm with which the dynamics could be compared to those of the computational atom used in the RHSP-based construction method, the linearised version of the algorithm was considered. After backward Euler temporal discretisation (with timestep Δt), the three steps can

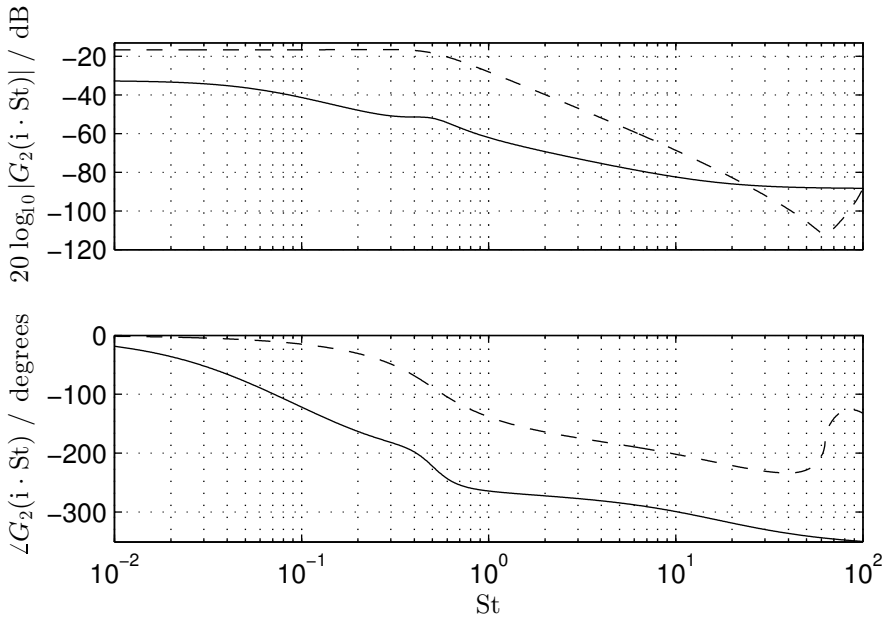


Figure 6.12: Comparison of frequency responses of $G_2(s)$: model obtained using RHSP-based construction method (−), average of models obtained using system identification experiments (−−).

be summarised as below, where the starting point is the k^{th} timestep, the fluctuating velocity and pressure fields are denoted $\mathbf{u}'^k$ and p'^k, respectively, and the superscripts † and †† denote the values of intermediate fields. Note that the velocity and pressure fields' dependence on spatial location has been dropped for notational brevity.

(1) *Predictor* step: The pressure field at timestep k is used to predict an intermediate fluctuation velocity field $\mathbf{u}'^\dagger$ by solving the following equation:

$$\frac{1}{\Delta t}\left(\mathbf{u}'^\dagger - \mathbf{u}'^k\right) = -\mathbf{u}'^\dagger\cdot\nabla\bar{\mathbf{u}} - \bar{\mathbf{u}}\cdot\nabla\mathbf{u}'^\dagger + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}'^\dagger - \nabla\mathsf{p}'^k. \tag{6.8}$$

This new velocity field will not, in general, satisfy the continuity equation.

(2) *Corrector* step 1: New fluctuation velocity and pressure fields $\mathbf{u}'^{\dagger\dagger}$ and $\mathsf{p}'^\dagger$ are sought such that the velocity field does satisfy the continuity equation. The new pressure field is first computed by solving the following PPE:

$$\nabla^2\mathsf{p}'^\dagger = -\nabla\cdot\left(\mathbf{u}'^\dagger\cdot\nabla\bar{\mathbf{u}}\right) - \nabla\cdot\left(\bar{\mathbf{u}}\cdot\nabla\mathbf{u}'^\dagger\right) + \frac{1}{\mathrm{Re}}\nabla^2\left(\nabla\cdot\mathbf{u}'^\dagger\right) + \frac{1}{\Delta t}\nabla\cdot\mathbf{u}'^k. \tag{6.9}$$

The new velocity field is then computed by solving the corrector momentum equation:

$$\frac{1}{\Delta t}\left(\mathbf{u}'^{\dagger\dagger} - \mathbf{u}'^k\right) = -\mathbf{u}'^\dagger\cdot\nabla\bar{\mathbf{u}} - \bar{\mathbf{u}}\cdot\nabla\mathbf{u}'^\dagger + \frac{1}{\mathrm{Re}}\nabla^2\mathbf{u}'^\dagger - \nabla\mathsf{p}'^\dagger. \tag{6.10}$$

(3) *Corrector* step 2: The final velocity and pressure fields $\mathbf{u}'^{k+1}$ and p'^{k+1} are then computed by first solving the second PPE:

$$\begin{aligned}\nabla^2 \mathsf{p}'^{k+1} = -\nabla \cdot \left(\mathbf{u}'^{\dagger\dagger} \cdot \nabla \bar{\mathbf{u}}\right) - \nabla \cdot \left(\bar{\mathbf{u}} \cdot \nabla \mathbf{u}'^{\dagger\dagger}\right) \\ + \frac{1}{\mathrm{Re}} \nabla^2 \left(\nabla \cdot \mathbf{u}'^{\dagger\dagger}\right) + \frac{1}{\Delta t} \nabla \cdot \mathbf{u}'^{k},\end{aligned} \tag{6.11}$$

before solving the final predictor momentum equation:

$$\frac{1}{\Delta t}\left(\mathbf{u}'^{k+1} - \mathbf{u}'^{k}\right) = -\mathbf{u}'^{\dagger\dagger} \cdot \nabla \bar{\mathbf{u}} - \bar{\mathbf{u}} \cdot \nabla \mathbf{u}'^{\dagger\dagger} + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}'^{\dagger\dagger} - \nabla \mathsf{p}'^{k+1}. \tag{6.12}$$

In order to construct a discrete-time state-space model of a single computational atom, (6.8)–(6.12) were discretised, using centred finite-differences, on a staggered mesh with uniform mesh spacing $\delta_{\mathsf{x}} = \delta_{\mathsf{y}} = \delta$. For the $(i,j)^{\text{th}}$ node, neglecting effects of surrounding nodes, these steps can be written as a sequence of matrix transformations as such:

(1) *Predictor* step:
The initial predictor step can be expressed as:

$$\begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{p}'^{k}_{i,j} \\ \mathsf{u}'^{\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger}_{i,j+1/2} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ \frac{1}{a_1 \Delta t} & 0 & \frac{1}{a_1 \delta} \\ 0 & \frac{1}{a_2 \Delta t} & -\frac{1}{a_2 \delta} \end{bmatrix}}_{A_{\mathrm{P}} \in \mathbb{R}^{5\times 3}} \begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{p}'^{k}_{i,j} \end{bmatrix}, \tag{6.13}$$

where:

$$a_1 := \frac{1}{\Delta t} + \frac{1}{2\delta}\left(\bar{\mathsf{u}}_{i+3/2,j} - \bar{\mathsf{u}}_{i-1/2,j}\right) + \frac{4}{\delta^2 \mathrm{Re}}, \tag{6.14a}$$

$$a_2 := \frac{1}{\Delta t} + \frac{1}{2\delta}\left(\bar{\mathsf{v}}_{i,j-1/2} - \bar{\mathsf{v}}_{i,j+3/2}\right) + \frac{4}{\delta^2 \mathrm{Re}}. \tag{6.14b}$$

(2) *Corrector* step 1:
Solution of the first PPE can be expressed as:

$$\begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{u}'^{\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger}_{i,j+1/2} \\ \mathsf{p}'^{\dagger}_{i,j} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ -\frac{\delta}{4\Delta t} & \frac{\delta}{4\Delta t} & 0 & a_3 & a_4 \end{bmatrix}}_{A_{\mathrm{C1p}} \in \mathbb{R}^{5\times 5}} \begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{p}'^{k}_{i,j} \\ \mathsf{u}'^{\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger}_{i,j+1/2} \end{bmatrix}, \tag{6.15}$$

where:

$$a_3 := \frac{1}{2}\left(\bar{\mathsf{u}}_{i+1/2,j} - \bar{\mathsf{u}}_{i-1/2,j}\right) - \frac{1}{16}\left(\bar{\mathsf{u}}_{i+1/2,j} + \bar{\mathsf{u}}_{i-1/2,j}\right), \tag{6.16a}$$

$$a_4 := \frac{1}{2}\left(\bar{\mathsf{v}}_{i,j-1/2} - \bar{\mathsf{v}}_{i,j+1/2}\right) - \frac{1}{16}\left(\bar{\mathsf{v}}_{i,j+1/2} + \bar{\mathsf{v}}_{i,j-1/2}\right), \tag{6.16b}$$

and solution of the first corrector momentum equation can be expressed as:

$$\begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{u}'^{\dagger\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger\dagger}_{i,j+1/2} \\ \mathsf{p}'^{\dagger}_{i,j} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & a_5 & 0 & \frac{\Delta t}{\delta} \\ 0 & 1 & 0 & a_6 & -\frac{\Delta t}{\delta} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{\mathrm{C1m}} \in \mathbb{R}^{5\times 5}} \begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{u}'^{\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger}_{i,j+1/2} \\ \mathsf{p}'^{\dagger}_{i,j} \end{bmatrix}, \tag{6.17}$$

where:

$$a_5 := -\frac{\Delta t}{2\delta}\left(\bar{\mathsf{u}}_{i+3/2,j} - \bar{\mathsf{u}}_{i-1/2,j}\right) - \frac{4\Delta t}{\delta^2 \mathrm{Re}}, \tag{6.18a}$$

$$a_6 := -\frac{\Delta t}{2\delta}\left(\bar{\mathsf{v}}_{i,j-1/2} - \bar{\mathsf{v}}_{i,j+3/2}\right) - \frac{4\Delta t}{\delta^2 \mathrm{Re}}. \tag{6.18b}$$

(3) *Corrector* step 2:
Solution of the second PPE can be expressed as:

$$\begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{u}'^{\dagger\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger\dagger}_{i,j+1/2} \\ \mathsf{p}'^{k+1}_{i,j} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{\delta}{4\Delta t} & \frac{\delta}{4\Delta t} & a_3 & a_4 & 0 \end{bmatrix}}_{A_{\mathrm{C2p}} \in \mathbb{R}^{5\times 5}} \begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{u}'^{\dagger\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger\dagger}_{i,j+1/2} \\ \mathsf{p}'^{\dagger}_{i,j} \end{bmatrix}, \tag{6.19}$$

and solution of the second corrector momentum equation can be expressed as:

$$\begin{bmatrix} \mathsf{u}'^{k+1}_{i+1/2,j} \\ \mathsf{v}'^{k+1}_{i,j+1/2} \\ \mathsf{p}'^{k+1}_{i,j} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & a_5 & 0 & \frac{\Delta t}{\delta} \\ 0 & 1 & 0 & a_6 & -\frac{\Delta t}{\delta} \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{A_{\mathrm{C2m}} \in \mathbb{R}^{3\times 5}} \begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{u}'^{\dagger\dagger}_{i+1/2,j} \\ \mathsf{v}'^{\dagger\dagger}_{i,j+1/2} \\ \mathsf{p}'^{k+1}_{i,j} \end{bmatrix}. \tag{6.20}$$

The overall discrete-time system which maps states from timestep k to $k+1$ can therefore be written:

$$\underbrace{\begin{bmatrix} \mathsf{u}'^{k+1}_{i+1/2,j} \\ \mathsf{v}'^{k+1}_{i,j+1/2} \\ \mathsf{p}'^{k+1}_{i,j} \end{bmatrix}}_{\mathbf{x}_{i,j}(k+1)} = \underbrace{A_{\mathrm{C2m}} A_{\mathrm{C2p}} A_{\mathrm{C1m}} A_{\mathrm{C1p}} A_{\mathrm{P}}}_{A_{\mathrm{PISO}i,j} \in \mathbb{R}^{3\times 3}} \underbrace{\begin{bmatrix} \mathsf{u}'^{k}_{i+1/2,j} \\ \mathsf{v}'^{k}_{i,j+1/2} \\ \mathsf{p}'^{k}_{i,j} \end{bmatrix}}_{\mathbf{x}_{i,j}(k)}, \tag{6.21}$$

and the poles of the system are equal to the eigenvalues of $A_{\mathrm{PISO}i,j}$. In order to compare the dynamics of this system with those of the computational atom used in the RHSP-based construction method, the continuous-time system (6.1) was discretised in time using the same backward Euler difference scheme. For an arbitrary spatial location in the wake $(\mathsf{x}, \mathsf{y}) = (2, -0.5)$, timestep $\Delta t = 0.001$, and mesh spacing $\delta = 0.005$, the poles of both systems are plotted in Figure 6.13.

The poles of the PISO algorithm state-space representation are $z \in \{0, 0.457, 0.926\}$, whilst the poles of the computational atom used in the RHSP-based construction method are $z \in \{0, 0, 0.926\}$. Whilst the systems have two poles in common, the location of the third differs. The atom used for the RHSP-based construction method has two poles located at the origin, corresponding to two poles at minus infinity in

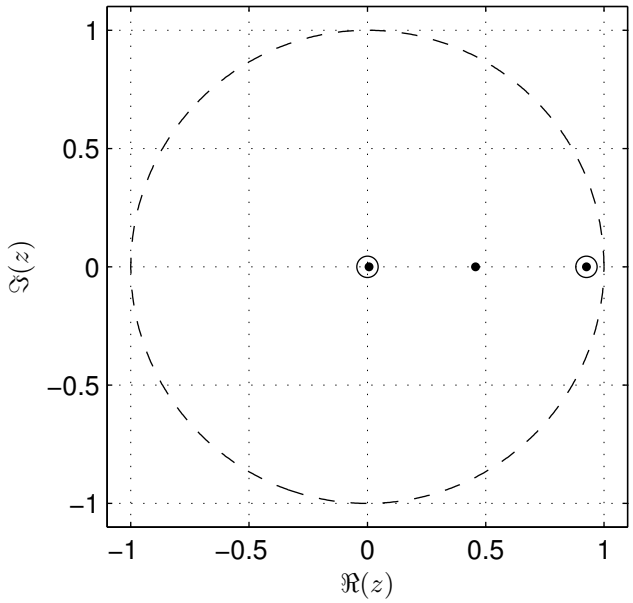


Figure 6.13: Discrete-time poles of PISO algorithm individual node subsystem state-space representation (•), and discrete-time poles of individual node subsystem used in RHSP-based construction method (◯).

the continuous-time case, whilst the PISO algorithm atom has only one located at the origin. This disagreement highlights a fundamental difference in the dynamics of a direct spatial discretisation of the linearised Navier-Stokes equations, and the operator splitting method of the PISO algorithm, however it is not yet clear why this difference arises. Due to this difference at the nodal level, one cannot necessarily expect the backward facing step system models obtained from linear system identification based on this algorithm to agree with those obtained with the RHSP-based construction method.

This important result begs the question 'when *is* the use of the PISO algorithm appropriate?', and is an area which certainly requires further investigation.

6.4 Summary

In this chapter, the RHSP-based construction method was applied to the 2D backward facing step flow with ZNMF actuation on the step rear edge for two different output configurations. In both cases, the RHSP-based construction method yielded frequency responses which converged upon mesh refinement, and 4^{th}-order transfer functions were fitted using least squares regression. These models were compared to models obtained from a CFD-based linear system identification study, from which significant discrepancy was observed. In order to explain the cause of this discrepancy, a discrete-time state-

space representation of the standard PISO algorithm was constructed (for a single computational node), and compared to the computational atom used in the RHSP-based construction method. This revealed differing poles, explaining the difference between the backward facing step models obtained using the RHSP-based construction method and those obtained using system identification based on the PISO algorithm.

This result brings into question the validity of the PISO algorithm and, as such, when its use is appropriate requires further investigation.

Chapter 7

Conclusions and future work

In this closing chapter, a summary of the work presented in Chapters 3–6 is given, before the main contributions of the work are listed, and avenues for future work are suggested.

7.1 Summary

The focus of this work has been in developing a computationally efficient modelling technique for obtaining linear, low-order models of fluid flows around complex geometries, suitable for feedback control design. In the introductory chapter of this work, the term 'complex geometry flows' was defined to mean backward facing step flows, D-shaped bluff body flows, and cylinder flows, and relevant literature on the modelling and control of such flows was discussed in Chapter 2. In Chapter 3, the 2D backwards facing step flow which would serve as a test case later on was introduced in detail. The main features of the flow were discussed, before details of the numerical simulation were presented. This included details of the LES solver, how the computational mesh was generated, scalability of the simulations on parallel HPC clusters, and how ZNMF slot jet actuation was implemented on the step rear edge. The assumption that the flow could be modelled as a linear system was verified by considering the input-output behaviour whilst actuating the flow harmonically at a number of forcing frequencies and amplitudes. Comparing the forced and unforced output spectra showed that for small forcing amplitudes the flow indeed behaved in a linear fashion. This justifed the use of linear systems theory for such a flow.

In Chapter 4, the modelling approach used in this work – referred to here as the *RHSP-based construction method* – was presented. The technique constructs the overall frequency response of a system governed by a linear PDAE in a fashion that avoids the

necessity to construct, store, or invert extremely large state-space matrices, or run expensive simulations. This is achieved by exploiting the inherent structure of spatially discretised PDAEs, and building up the overall system frequency response by chaining together the frequency responses of individual computational node subsystems.

For the two spatial dimension case, a domain decomposition optimisation was presented, and Lemma 1 established the existence of a globally minimal computational cost which could be achieved by choosing the decomposition parameters optimally. It was then proven in Theorem 1 that by employing this optimisation, the computational cost of the modelling approach could be reduced from $\mathcal{O}\left(\varrho^4\right)$ to $\mathcal{O}\left(\varrho^2\right)$ (whilst obtaining the frequency response directly from the full-scale system matrices costs $\mathcal{O}\left(\varrho^6\right)$).

The effectiveness of the modelling approach was demonstrated by application to the 2D wave-diffusion equation example. This yielded identical frequency response to that obtained directly from the full-order state-space system, and confirmed its cost to be $\mathcal{O}\left(\varrho^2\right)$. In addition, both the computational memory requirements and conditioning numbers of matrices involved were orders of magnitude lower than those of the direct approach.

In order to apply the RHSP-based construction method to a fluids problem, a suitable spatial discretisation of the linearised Navier-Stokes equations was required. Chapter 5 addressed the issues involved with choosing such a discretisation, highlighting the ambiguity in the literature. It was noted that that whilst some formulations of the governing equations may be suitable for simulation purposes, they may not be for control design.

The dynamics of a single computational node formed by spatial discretisation of the governing equations in both primitive variables and PPE formulations were considered, revealing fundamental numerical differences at the nodal level. The effects of these differences on the linearised system dynamics at the full system level were then exemplified by considering the corresponding formulations of a 2D channel flow, subjected to a variety of different boundary conditions, and comparing the models to the benchmark Orr-Sommerfeld model. Anaylsis of the systems' poles, zeros, pseudospectra, eigenfunctions, and frequency responses highlighted the differences in dynamics between the different model formulations, and suggested that the only formulation of the governing equations which was dynamically accurate without ambiguity in choice of boundary conditions was the primitive variables formulation discretised on a staggered mesh. The ν-gaps between the different model formulations and the benchmark model were computed, and these confirmed which formulations were significantly different in a closed-loop sense.

Finally, in Chapter 6, the RHSP-based construction method was used to obtain

the frequency response of the 2D backward facing step flow introduced in Chapter 3. Informed by the findings of Chapter 5, the primitive variables formulation of the linearised Navier-Stokes equations was used, discretised on a staggered mesh. The same ZNMF slot jet actuation of Chapter 3 was assumed, and two output configurations were considered: pressure fluctuations on the rear face, and wall-normal velocity fluctuations in the wake.

Upon refinement of the computational mesh, the frequency responses obtained for both outputs converged. Transfer function models were fitted to each of the frequency responses using least squares regression, and the RMS error of fitted models of increasing order suggested that 4^{th}-order models were sufficiently accurate.

These results were compared with those obtained from a computational system identification study, which applied subspace identification techniques to input-output data from a number of different open-loop forced simulation experiments in order to fit transfer functions. Analysis of the Hankel singular values of the models produced suggested that 3^{rd}-order models were sufficient for both output configurations.

There was significant discrepancy between the results obtained with the RHSP-based construction method, and those of the system identification study, suggesting that the dynamics of the numerical solver were fundamentally different to those obtained from a direct discretisation of the linearised Navier-Stokes equations. In order to investigate this difference, a discrete-time state-space representation of the linearised version of the standard PISO algorithm discretised on a staggered mesh was constructed, and compared to the computational node subsystem used in the RHSP-based construction method. This revealed differing system poles, explaining the difference in backward facing step flow models. As a result of this fundamental difference, the validity of the PISO algorithm was brought into question – something of considerable importance which should be investigated further.

7.2 Main contributions

To the best of the author's knowledge, the following elements of this work are original contributions:

i. Developing a generalisation of the modelling approach used by Baramov *et al.* [10, 11] – referred to in this work as the *RHSP-based construction method* – suitable for obtaining low-order models of systems governed linear PDEs and PDAEs on complex spatial domains.

ii. Developing a spatial domain decomposition optimisation for RHSP-based construc-

tion method, and showing that for the two spatial dimension case there exists a globally minimal computational cost which can be achieved by choosing the decomposition parameters optimally.

iii. A proof that for the two spatial dimension case, using the RHSP-based construction method in conjunction with the domain decomposition optimisation reduces the computational cost from $\mathcal{O}\left(\varrho^4\right)$ to $\mathcal{O}\left(\varrho^2\right)$ (whilst obtaining the frequency response directly from the full-scale system matrices costs $\mathcal{O}\left(\varrho^6\right)$).

iv. A rigorous systems theoretic analysis of different formulations and discretisations of the linearised Navier-Stokes equations, drawing conclusions about which formulation and boundary condition combinations are dynamically correct and suited to modelling fluids for the purpose of feedback control design, and which are only appropriate for simulation purposes. This work has been published in International Journal for Numerical Methods in Fluids.

v. Deriving a discrete-time state-space representation of the linearised PISO algorithm (for a single computational node) and showing that the underlying dynamics differ to those of a direct discretisation of the linearised Navier-Stokes equations.

7.3 Future work

Based on the work presented in this thesis, the following is a list of suggested avenues for future work:

i. An extension of the results presented in Chapter 4 to the three spatial dimension case. Ultimately, real world fluid flows (and many other physical phenomenon governed by PDEs and PDAEs) are 3D, and as such the natural progression from this work is to extend the modelling methodology, along with the results regarding optimisation by domain decomposition, to the 3D case.

ii. Application of the RHSP-based construction method to a bluff body that better represents a road vehicle, such as the Ahmed body in ground proximity.

iii. Developing a compiled C/C++ implementation of the RHSP-based construction method for increased computational speed.

iv. Developing a framework for automating the use of the RHSP-based construction method in conjunction with readily available meshing software in order to make it an accessible tool for a new user.

v. Using the RHSP-based construction method to inform actuator and sensor placement. For a given geometry, assuming a number of actuators and sensors in different

locations, the RHSP-based construction method could be used to obtain MIMO models from which controllability and observability properties could be deduced for different actuator and sensor configurations. With this, sensible choices of actuator and sensor placement could be made.

vi. Further investigation into the equivalency, or lack thereof, between a direct spatial discretisation of the Navier-Stokes equations and the PISO algorithm used in many CFD solvers. In Chapter 6 it was shown that in the linearised case, the poles of a state-space representation of the PISO algorithm are different to those of a direct discretisation of the governing equations, and it is unclear where/why this discrepancy arises. This is a significant discrepancy, and as such an explanation should be sought.

Appendix A

Elements of $A_{i,j}$ and $B_{i,j}$ in (6.1)

$$A_{i,j} := \begin{bmatrix} a_{11} & 0 & a_{13} \\ 0 & a_{22} & a_{23} \\ a_{31} & a_{32} & 0 \end{bmatrix}, \tag{A.1}$$

where:

$$a_{11} := -0.5\delta^{-1}\left(\bar{\mathsf{u}}_{i+3/2,j} - \bar{\mathsf{u}}_{i-1/2,j}\right) - 4\delta^{-2}\mathrm{Re}^{-1}, \tag{A.2a}$$
$$a_{13} := \delta^{-1}, \tag{A.2b}$$
$$a_{22} := -0.5\delta^{-1}\left(\bar{\mathsf{v}}_{i,j-1/2} - \bar{\mathsf{v}}_{i,j+3/2}\right) - 4\delta^{-2}\mathrm{Re}^{-1}, \tag{A.2c}$$
$$a_{23} := -\delta^{-1}, \tag{A.2d}$$
$$a_{31} := 1, \tag{A.2e}$$
$$a_{32} := -1. \tag{A.2f}$$

$$B_{1_{i,j}} := \begin{bmatrix} b_{1_{11}} & b_{1_{12}} & 0 \\ 0 & b_{1_{22}} & 0 \\ 0 & b_{1_{32}} & 0 \end{bmatrix}, \tag{A.3}$$

where:

$$b_{1_{11}} := -0.25\delta^{-1}\left(\bar{\mathsf{v}}_{i,j-1/2} + \bar{\mathsf{v}}_{i+1,j+1/2}\right) + \delta^{-2}\mathrm{Re}^{-1}, \tag{A.4a}$$
$$b_{1_{12}} := -0.25\delta^{-1}\left(\bar{\mathsf{u}}_{i+1/2,j-1} - \bar{\mathsf{u}}_{i+1/2,j+1}\right), \tag{A.4b}$$
$$b_{1_{22}} := -0.5\delta^{-1}\bar{\mathsf{v}}_{i,j+1/2} + \delta^{-2}\mathrm{Re}^{-1}, \tag{A.4c}$$
$$b_{1_{32}} := 1. \tag{A.4d}$$

$$B_{2_{i,j}} := \begin{bmatrix} b_{2_{11}} & b_{2_{12}} & b_{2_{13}} \\ 0 & b_{2_{22}} & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{A.5}$$

where:

$$b_{2_{11}} := -0.5\delta^{-1}\bar{\mathsf{u}}_{i+1/2,j} + \delta^{-2}\mathrm{Re}^{-1}, \tag{A.6a}$$
$$b_{2_{12}} := 0.25\delta^{-1}\left(\bar{\mathsf{u}}_{i+1/2,j-1} - \bar{\mathsf{u}}_{i+1/2,j+1}\right), \tag{A.6b}$$
$$b_{2_{13}} := -\delta^{-1}, \tag{A.6c}$$
$$b_{2_{22}} := -0.25\delta^{-1}\left(\bar{\mathsf{u}}_{i-1/2,j} + \bar{\mathsf{u}}_{i+1/2,j+1}\right) + \delta^{-2}\mathrm{Re}^{-1}. \tag{A.6d}$$

$$B_{3_{i,j}} := \begin{bmatrix} b_{3_{11}} & 0 & 0 \\ b_{3_{21}} & b_{3_{22}} & b_{3_{23}} \\ 0 & 0 & 0 \end{bmatrix}, \tag{A.7}$$

where:

$$b_{3_{11}} := 0.25\delta^{-1}\left(\bar{\mathsf{v}}_{i,j-1/2} + \bar{\mathsf{v}}_{i+1,j+1/2}\right) + \delta^{-2}\mathrm{Re}^{-1}, \tag{A.8a}$$
$$b_{3_{21}} := 0.25\delta^{-1}\left(\bar{\mathsf{v}}_{i+1,j+1/2} - \bar{\mathsf{v}}_{i-1,j+1/2}\right), \tag{A.8b}$$
$$b_{3_{22}} := 0.5\delta^{-1}\bar{\mathsf{v}}_{i,j+1/2} + \delta^{-2}\mathrm{Re}^{-1}, \tag{A.8c}$$
$$b_{3_{23}} := \delta^{-1}. \tag{A.8d}$$

$$B_{4_{i,j}} := \begin{bmatrix} b_{4_{11}} & 0 & 0 \\ b_{4_{21}} & b_{4_{22}} & 0 \\ b_{4_{31}} & 0 & 0 \end{bmatrix}, \tag{A.9}$$

where:

$$b_{4_{11}} := 0.5\delta^{-1}\bar{\mathsf{u}}_{i+1/2,j} + \delta^{-2}\mathrm{Re}^{-1}, \tag{A.10a}$$
$$b_{4_{21}} := -0.25\delta^{-1}\left(\bar{\mathsf{v}}_{i+1,j+1/2} - \bar{\mathsf{v}}_{i-1,j+1/2}\right), \tag{A.10b}$$
$$b_{4_{22}} := 0.25\delta^{-1}\left(\bar{\mathsf{u}}_{i-1/2,j} + \bar{\mathsf{u}}_{i+1/2,j+1}\right) + \delta^{-2}\mathrm{Re}^{-1}, \tag{A.10c}$$
$$b_{4_{31}} := -1. \tag{A.10d}$$

Bibliography

[1] Results of the review of the community strategy to reduce CO_2 emissions from passenger cars and light-commercial vehicles. Technical report, European Commission, 2007.

[2] Energy consumption in the UK. Technical report, Department of Energy & Climate Change, UK Government, 2015.

[3] O. M. Aamo and M. Krstić. *Flow Control by Feedback: Stabilization and Mixing.* Springer-Verlag, London, Great Britain, 1st edition, 2003.

[4] P. Agrawal, V. D. Agrawal, M. L. Bushnell, and J. Sienicki. Superlinear speedup in multiprocessing environment. In *First International Workshop on Parallel Processing*, Bangalore, December 1994.

[5] C. K. Aidun and J. R. Clausen. Lattice-Boltzmann method for complex flows. *Annual Review of Fluid Mechanics*, 42:439–472, 2010.

[6] I. Akhtar, M. Naqvi, J. Borggaard, and J. A. Burns. Using dominant modes for optimal feedback control of aerodynamic forces. *Proceedings of the IMechE Part G: Journal of Aerospace Engineering*, 227(12):1859–1869, 2012.

[7] A. Altaf, A. A. Omar, and W. Asrar. Passive drag reduction of square back road vehicles. *Journal of Wind Engineering and Industrial Aerodynamics*, 134:30–43, 2016.

[8] A. C. Antoulas. An overview of approximation methods for large-scale dynamical systems. *Annual Reviews in Control*, 29:181–190, 2005.

[9] K. J. Aström and R. M. Murray. *Feedback Systems.* Princeton University Press, Princeton, New Jersey, USA, 1st edition, 2008.

[10] L. Baramov, O. R. Tutty, and E. Rogers. $\mathscr{H}_\infty$ control for non-periodic planar channel flows. In *Proceedings of the 40th IEEE Conference on Decision and Control*, Orlando, Florida, USA, December 2001.

[11] L. Baramov, O. R. Tutty, and E. Rogers. $\mathscr{H}_\infty$ control of nonperiodic two-dimensional channel flow. *IEEE Transactions on Control Systems Technology*, 12(1):111–122, 2004.

[12] A. Barbagallo, G. Dergham, D. Sipp, P. J. Schmid, and J.-C. Robinet. Closed-loop control of unsteadiness over a rounded backward-facing step. *Journal of Fluid Mechanics*, 703:326–362, 2012.

[13] T. Bewley, P. Luchini, and J. Pralits. Methods for solution of large optimal control problems that bypass open-loop model reduction. *Meccanica*, 51:2997–3014, 2016.

[14] T. R. Bewley. Flow control: new challenges for a new renaissance. *Progress in Aerospace Sciences*, 37:21–58, 2001.

[15] R. Bradley. Technology roadmap for the 21st century truck program. Technical Report 21CT-001, United States Department of Energy, Washington DC, USA, 2000.

[16] K. E. Brenan, S. L. V. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 1996.

[17] K. M. Butler and B. F. Farrell. Three-dimensional optimal perturbations in viscous shear flow. *Physics of Fluids A*, 4(8):1637–1650, 1992.

[18] L. N. Cattafesta and M. Shelpak. Actuators for active flow control. *Annual Review of Fluid Mechanics*, 43:247–272, 2011.

[19] X. Chai and K. Mahesh. Dynamic k-equation model for large eddy simulation of compressible flows. In *40th Fluid Dynamics Conference and Exhibit*, Chicago, Illinois, USA, July 2010.

[20] S. Chaligné, T. Castelain, M. Michard, and D. Juvé. Active control of the flow behind a two-dimensional bluff body in ground proximity. *Comptes Rendus Mecanique*, 341:289–297, 2013.

[21] S. Chen and D. Martínez. On boundary conditions in lattice Boltzmann methods. *Physics of Fluids*, 8:2527–2536, 1996.

[22] H. Choi, W.-P. Jeon, and J. Kim. Control of flow over a bluff body. *Annual Review of Fluid Mechanics*, 40:113–139, 2008.

[23] H. Choi, J. Lee, and H. Park. Aerodynamics of heavy vehicles. *Annual Review of Fluid Mechanics*, 46:441–468, 2014.

[24] T. J. Chung. *Computational Fluid Dynamics*. Cambridge University Press, Cambridge, UK, 2nd edition, 2010.

[25] R. Curtain and K. Morris. Transfer functions of distributed parameter systems: A tutorial. *Automatica*, 45:1101–1116, 2009.

[26] J. D'Adamo, R. Sosa, and G. Artana. Active control of a backward facing step flow with plasma actuators. *Journal of Fluids Engineering*, 136, 2014.

[27] J. Dahan. *Linear feedback control for form-drag reduction on bluff bodies with a blunt trailing edge*. PhD thesis, Imperial College London, January 2013.

[28] J. A. Dahan, A. S. Morgans, and S. Lardeau. Feedback control for form-drag reduction on a bluff body with a blunt trailing edge. *Journal of Fluid Mechanics*, 704:360–387, 2012.

[29] L. Dai, editor. *Singular Control Systems*. Lecture Notes in Control and Information Sciences. Springer-Verlag, 1989.

[30] P. A. Davidson. *Turbulence: An Introduction for Scientists and Engineers*. Oxford University Press, Oxford, UK, 1st edition, 2004.

[31] T. A. Davis. *Direct Methods for Sparse Linear Systems*. Fundamentals of Algorithms. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1st edition, 2006.

[32] R. C. Dorf and R. H. Bishop. *Modern Control Systems*. Prentice Hall, USA, 10th edition, 2004.

[33] J. Doyle, A. Packard, and K. Zhou. Review of LFTs, LMIs and μ. In *Proceedings of the 30th Conference on Decision and Control*, Brighton, England, 1991.

[34] D. M. Driver, H. L. Seegmiller, and J. G. Marvin. Time-dependent behavior of a reattaching shear layer. *AIAA Journal*, 25(7):914–919, 1987.

[35] M. G. el Hak. *Flow Control: Passive, Active, and Reactive Flow Management*. Cambridge University Press, Cambridge, UK, 1st edition, 2000.

[36] A. Faghri, Y. Zhang, and J. Howell. *Advanced Heat and Mass Transfer*. Global Digital Press, Missouri, USA, 1st edition, 2010.

[37] J. H. Ferziger and M. Perić. *Computational Methods for Fluid Dynamics*. Springer-Verlag, Berlin, Germany, 2nd edition, 1997.

[38] T. L. B. Flinois and A. S. Morgans. Feedback control of unstable flows: a direct modelling approach using the Eigensystem Realisation Algorithm. *Journal of Fluid Mechanics*, 793:41–78, 2016.

[39] C. Foiaş and A. E. Frazho. Redheffer products and the lifting of contractions on Hilbert space. *Journal of Operator Theory*, 11:193–196, 1984.

[40] N. Gautier and J.-L. Aider. Upstream open loop control of the recirculation area downstream of a backward-facing step. *Comptes Rendus Mecanique*, 342:382–388, 2014.

[41] N. Gautier, J.-L. Aider, T. Duriez, B. R. Noack, M. Segond, and M. Abel. Closed-loop separation control using machine learning. *Journal of Fluid Mechanics*, 770:442–457, 2015.

[42] G. H. Golub and C. F. V. Loan. *Matrix Computations*. The Johns Hopkins University Press, Maryland, USA, 3rd edition, 1996.

[43] P. M. Gresho. Incompressible fluid dynamics: Some fundamental formulation issues. *Annual Review of Fluid Mechanics*, 23:413–453, 1991.

[44] P. M. Gresho and R. L. Sani. On pressure boundary conditions for the incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 7:1111–1145, 1987.

[45] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The Physics of Fluids*, 8(12):2182–2189, 1965.

[46] P. H. Heins, B. L. Jones, and A. S. Sharma. Passivity-based output-feedback control of turbulent channel flow. *Automatica*, 69:348–355, 2016.

[47] L. Henning and R. King. Drag reduction by closed-loop control of a separated flow over a bluff body with a blunt trailing edge. In *44th IEEE Conference on Decision and Control, and the European Control Conference*, Seville, Spain, 2005.

[48] A. Hervé, D. Sipp, P. J. Schmid, and M. Samuelides. A physics-based approach to flow control using system identification. *Journal of Fluid Mechanics*, 207:26–58, 2012.

[49] J. D. Hoffman and S. Frankel. *Numerical Methods for Engineers and Scientists*. CRC Press, USA, 2nd edition, 2001.

[50] P. Holmes, J. L. Lumley, and G. Berkooz. *Turbulence, Coherent Structures, Dynamical Systems and Symmetry*. Cambridge University Press, Cambridge, UK, 1st edition, 1996.

[51] W.-H. Hucho, editor. *Aerodynamics of Road Vehicles*. Society of Automotive Engineers, USA, 4th edition, 1998.

[52] P. Huerre and P. A. Monkewitz. Local and global instabilities in spatially developing flows. *Annual Review of Fluid Mechanics*, 22:473–537, 1990.

[53] S. J. Illingworth. Model-based control of vortex shedding at low Reynolds numbers. *Theoretical and Computational Fluid Dynamics*, 2016.

[54] R. I. Issa. Solution of the implicitly discretised fluid flow equations by operator-splitting. *Journal of Computational Physics*, 62:40–65, 1985.

[55] B. L. Jones. Gap metric bound construction from frequency response data. In *19th World Congress for The International Federation of Automatic Control*, Cape Town, South Africa, 2014.

[56] B. L. Jones, P. H. Heins, E. C. Kerrigan, J. F. Morrison, and A. S. Sharma. Modelling for robust feedback control of fluid flows. *Journal of Fluid Mechanics*, 769:687–722, 2015.

[57] B. L. Jones, E. C. Kerrigan, and J. F. Morrison. A modeling and filtering framework for the semi-discretised Navier-Stokes equations. In *Proceedings of the European Control Conference*, Budapest, Hungary, August 2009.

[58] J. Kim. Physics and control of wall turbulence for drag reduction. *Philosophical Transactions of the Royal Society A*, 369:1396–1411, 2011.

[59] J. Kim and T. R. Bewley. A linear systems approach to flow control. *Annual Review of Fluid Mechanics*, 39:383–417, 2007.

[60] J. Kim, S. Hahn, J. Kim, D. Lee, J. Choi, W. Jeon, and H. Choi. Active control of turbulent flow over a model vehicle for drag reduction. *Journal of Turbulence*, 5:1–12, 2004.

[61] L. D. Kral, J. F. Donovan, A. B. Cain, and A. W. Cary. Numerical simulation of synthetic jet actuators. In *4th Shear Flow Control Conference*, Snowmass Villave, CO, USA, July 1997.

[62] P. Kunkel and V. Mehrmann. *Differential-Algebraic Equations: Analysis and Numerical Solution*. European Mathematical Society, Helsinki, Finland, 2006.

[63] L. Ljung. *System Identification: Theory for the User*. Prentice Hall, Upper Saddle River, NJ, USA, 2nd edition, 1999.

[64] L. M. Mack. A numerical study of the termporal eigenvalue spectrum of the Blasius boundary layer. *Journal of Fluid Mechanics*, 73:497–520, 1976.

[65] F. Martinelli, M. Quadrio, J. McKernan, and J. F. Whidborne. Linear feedback control of transient energy growth and control performance limitations in subcritical plane Poiseuille flow. *Physics of Fluids*, 23, 2011.

[66] L. Mathelin, M. Abbas-Turki, L. Pastur, and H. Abou-Kandil. Closed-loop fluid flow control using a low dimensional model. *Mathematical and Computer Modelling*, 52:1161–1168, 2010.

[67] B. J. McKeon, A. S. Sharma, and I. Jacobi. Experimental manipulation of wall turbulence: A systems approach. *Physics of Fluids*, 25, 2013.

[68] J. McKernan, G. Papadakis, and J. F. Whidborne. A linear state-space representation of plane Poiseuille flow for control design - a tutorial. *International Journal of Modelling, Identification and Control*, 1(4):272–280, 2006.

[69] M. Milano, P. Koumoutsakos, X. Giannakopoulos, and J. Schmidhuber. Evolving strategies for active flow control. In *Proceedings of the 2000 Congress on Evolutionary Computation*, La Jolla, California, USA, 2000.

[70] M. Milovanovic and O. M. Aamo. Attenuation of vortex shedding by model-based output feedback control. *IEEE Transactions on Control Systems Technology*, 21(3):617–625, 2013.

[71] V. J. Modi, S. S. Hill, and T. Yokomizo. Drag reduction of trucks through boundary-layer control. *Journal of Wind Engineering and Industrial Aerodynamics*, 54/55:583–594, 1995.

[72] W. M. F. Orr. The stability or instability of the steady motions of a perfect liquid and of a viscous liquid. Part I: A perfect liquid. *Proceedings of the Royal Irish Academy*, 27:9–68, 1907.

[73] S. A. Orszag. Accurate solution of the Orr-Sommerfeld stability equation. *Journal of Fluid Mechanics*, 50:689–703, 1971.

[74] D. J. Parkin, M. C. Thompson, and J. Sheridan. Numerical analysis of bluff body wakes under periodic open-loop control. *Journal of Fluid Mechanics*, 739:94–123, 2014.

[75] M. Pastoor, L. Henning, B. R. Noack, R. King, and G. Tadmor. Feedback shear layer control for bluff body drag reduction. *Journal of Fluid Mechanics*, 608:161–196, 2008.

[76] M. Pastoor, R. King, B. R. Noack, A. Dillmann, and G. Tadmor. Model-based coherent-structure control of turbulent shear flows using low-dimensional vortex models. In *33rd AIAA Fluid Dynamics Conference and Exhibit*, Orlando, Florida, USA, June 2003.

[77] C. Pozrikidis. *Introduction to Theoretical and Computational Fluid Dynamics*. Oxford University Press, New York, USA, 1st edition, 1997.

[78] M. H. Protter and H. F. Weinberger. *Maximal Principles in Differential Equations*. Springer-Verlag, Berlin, Germany, 1984.

[79] R. Raju, E. Aram, R. Mittal, and L. Cattafesta. Simple models of zero-net mass-flux jets for flow control simulations. *International Journal of Flow Control*, 1(3):179–197, 2009.

[80] H. P. Rani and T. W. H. Sheu. Nonlinear dynamics in a backward-facing step flow. *Physics of Fluids*, 18, 2006.

[81] H. P. Rani, T. W. H. Sheu, and E. S. F. Tsai. Eddy structures in a transitional backward-facing step flow. *Journal of Fluid Mechanics*, 588:43–58, 2007.

[82] S. Rashidi, M. Hayatdavoodi, and J. A. Esfahani. Vortex shedding suppression and wake control: A review. *Ocean Engineering*, 126:57–80, 2016.

[83] D. Rempfer. Low-dimensional modeling and numerical simulation of transition in simple shear flows. *Annual Review of Fluid Mechanics*, 35:229–265, 2003.

[84] D. Rempfer. On boundary conditions for incompressible Navier-Stokes problems. *Applied Mechanics Reviews*, 59:107–125, 2006.

[85] E. Robertson, V. Choudhury, S. Bhushan, and D. K. Walters. Validation of OpenFOAM numerical methods and turbulence models for incompressible bluff body flows. *Computers and Fluids*, 123:122–145, 2016.

[86] C. W. Rowley. Model reduction for fluids, using balanced proper orthogonal decomposition. *International Journal of Bifurcation and Chaos*, 15(3):997–1013, 2005.

[87] R. L. Sani, J. Shen, O. Pironneau, and P. M. Gresho. Pressure boundary condition for the time-dependent incompressible Navier-Stokes equations. *International Journal for Numerical Methods in Fluids*, 50:673–682, 2006.

[88] P. J. Schmid. Nonmodal stability theory. *Annual Review of Fluid Mechanics*, 39:129–162, 2007.

[89] P. J. Schmid and D. S. Henningson. *Stability and Transition in Shear Flows*. Springer-Verlag, New York, USA, 1st edition, 2001.

[90] A. S. Sharma, J. F. Morrison, B. J. McKeon, D. J. N. Limebeer, W. H. Koberg, and S. J. Sherwin. Relaminarisation of $\mathrm{Re}_\tau = 100$ channel flow with globally stabilising linear feedback control. *Physics of Fluids*, 23, 2011.

[91] D. Shirokoff and R. R. Rosales. An efficient method for the incompressible Navier-Stokes equations on irregular domains with no-slip boundary conditions, high

order up to the boundary. *Journal of Computational Physics*, 230:8619–8646, 2011.

[92] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. Wiley, West Sussex, UK, 2nd edition, 2005.

[93] R. Temam. Remark on the pressure boundary condition for the projection method. *Theoretical and Computational Fluid Dynamics*, 3:181–184, 1991.

[94] L. N. Trefethen. *Spectral Methods in MATLAB*. Society for Industrial and Applied Mathematics, Philadelphia, USA, 2000.

[95] L. N. Trefethen and M. Embree. *Spectra and Pseudospectra: The Behaviour of Nonnormal Matrices and Operators*. Princeton University Press, Princeton, New Jersey, USA, 2005.

[96] L. N. Trefethen, A. E. Trefethen, S. C. Reddy, and T. A. Driscoll. Hydrodynamic stability without eigenvalues. *Science*, 261(5121):578–584, 1993.

[97] G. Vinnicombe. Frequency domain uncertainty and the graph topology. *IEEE Transactions on Automatic Control*, 38(9):1371–1383, 1993.

[98] G. Vinnicombe. *Uncertainty and Feedback*. Imperial College Press, London, UK, 1st edition, 2000.

[99] T. von Kármán. Classics: Collapse of the Tacoma Narrows Bridge. *Resonance*, 10(8):97–102, 2005.

[100] A. W. Vreman. The projection method for the incompressible Navier-Stokes equations: The pressure near a no-slip wall. *Journal of Computational Physics*, 263:353–374, 2014.

[101] J. Weickert. Navier-Stokes equations as a differential-algebraic system, 1996.

[102] J. A. C. Weideman and S. C. Reddy. A MATLAB differentiation matrix suite. *ACM Transactions on Mathematical Software*, 26(4):465–519, 2000.

[103] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computer in Physics*, 12:620–631, 1998.

[104] J. Weller, S. Camarri, and A. Iollo. Feedback control by low-order modelling of the laminar flow past a bluff body. *Journal of Fluid Mechanics*, 634:405–418, 2009.

[105] K. Willcox and J. Peraire. Balanced model reduction via the proper orthogonal decomposition. *AIAA Journal*, 40(11):2323–2330, November 2002.

[106] T. R. Yechout. *Introduction to Aircraft Flight Mechanics: Performance, Static Stability, Dynamic Stability, and Classical Feedback Control.* American Institute of Aeronautics and Astronautics, Inc., Virginia, USA, 1st edition, 2003.