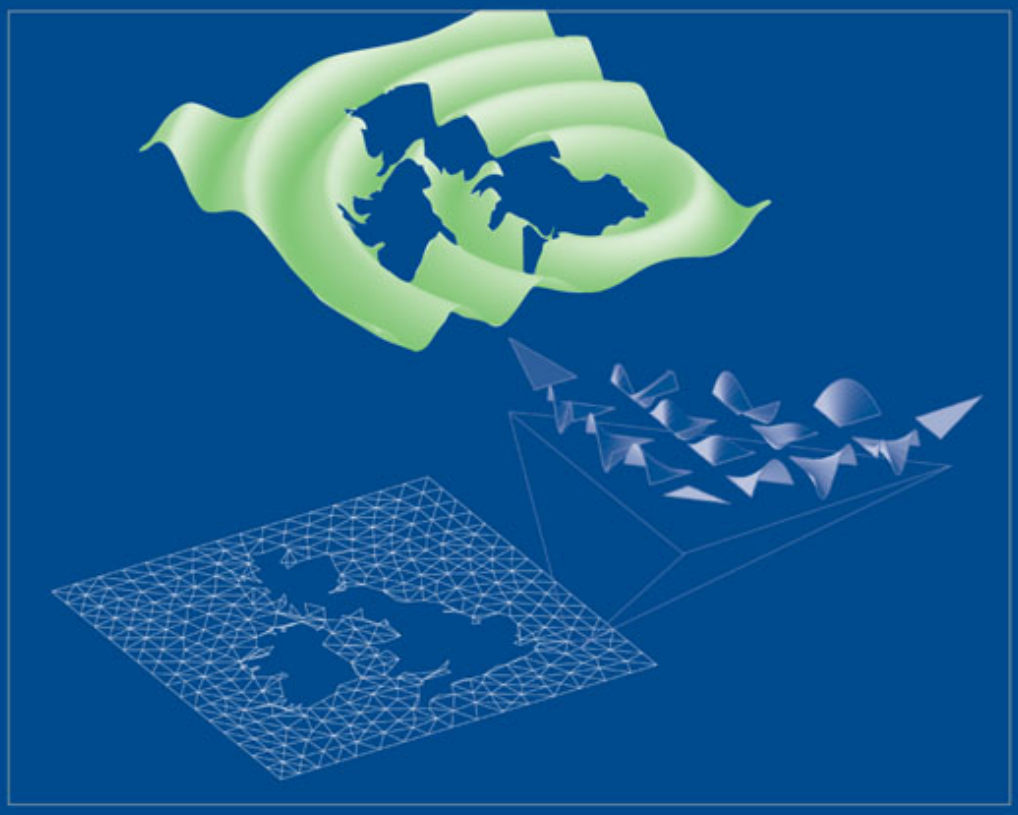
NUMERICAL MATHEMATICS
AND SCIENTIFIC COMPUTATION

# Spectral/hp Element Methods for Computational Fluid Dynamics

SECOND EDITION

GEORGE EM KARNIADAKIS
and SPENCER SHERWIN



OXFORD SCIENCE PUBLICATIONS

NUMERICAL MATHEMATICS AND SCIENTIFIC COMPUTATION

*Series Editors*

G. H. GOLUB A. GREENBAUM
A. M. STUART E. SÜLI

# NUMERICAL MATHEMATICS AND SCIENTIFIC COMPUTATION

*Books in the series*

Monographs marked with an asterisk (*) appeared in the series 'Monographs in Numerical Analysis' which has been folded into, and is continued by, the current series.

*P. Dierckx: *Curve and surface fittings with splines*

*J. H. Wilkinson: *The algebraic eigenvalue problem*

*I. Duff, A. Erisman, and J. Reid: *Direct methods for sparse matrices*

*M. J. Baines: *Moving finite elements*

*J. D. Pryce: *Numerical solution of Sturm–Liouville problems*

K. Burrage: *Parallel and sequential methods for ordinary differential equations*

Y. Censor and S. A. Zenios: *Parallel optimization: theory, algorithms, and applications*

M. Ainsworth, J. Levesley, W. Light, and M. Marletta: *Wavelets, multilevel methods, and elliptic PDEs*

W. Freeden, T. Gervens, and M. Schreiner: *Constructive approximation on the sphere: theory and applications to geomathematics*

Ch. Schwab: *p- and hp-finite element methods: theory and applications to solid and fluid mechanics*

J. W. Jerome: *Modelling and computation for applications in mathematics, science, and engineering*

Alfio Quarteroni and Alberto Valli: *Domain decomposition methods for partial differential equations*

G. E. Karniadakis and S. J. Sherwin: *Spectral/hp element methods for CFD*

I. Babuška and T. Strouboulis: *The finite element method and its reliability*

B. Mohammadi and O. Pironneau: *Applied shape optimization for fluids*

S. Succi: *The lattice Boltzmann equation for fluid dynamics and beyond*

P. Monk: *Finite element methods for Maxwell's equations*

A. Bellen and M. Zennaro: *Numerical methods for delay differential equations*

J. Modersitzki: *Numerical methods for image registration*

M. Feistauer, J. Felcman, and I. Straškraba: *Mathematical and computational methods for compressible flow*

W. Gautschi: *Orthogonal polynomials: computation and approximation*

M. K. Ng: *Iterative methods for Toeplitz systems*

Michael Metcalf, John Reid, and Malcolm Cohen: *Fortran 95/2003 explained*

George Em Karniadakis and Spencer Sherwin: *Spectral/hp element methods for CFD*, Second edition

# Spectral/*hp* Element Methods for CFD

George Em Karniadakis

*Division of Applied Mathematics*
*Brown University*
*Providence, RI 02912, USA*

and

Spencer J. Sherwin

*Department of Aeronautics*
*Imperial College London*
*South Kensington Campus*
*London, SW7 2AZ, UK*

OXFORD
UNIVERSITY PRESS

OXFORD
UNIVERSITY PRESS

Great Clarendon Street, Oxford OX2 6DP

Oxford University Press is a department of the University of Oxford.
It furthers the University's objective of excellence in research, scholarship,
and education by publishing worldwide in

Oxford New York

Auckland Cape Town Dar es Salaam Hong Kong Karachi
Kuala Lumpur Madrid Melbourne Mexico City Nairobi
New Delhi Shanghai Taipei Toronto

With offices in

Argentina Austria Brazil Chile Czech Republic France Greece
Guatemala Hungary Italy Japan South Korea Poland Portugal
Singapore Switzerland Thailand Turkey Ukraine Vietnam

Oxford is a registered trade mark of Oxford University Press
in the UK and in certain other countries

Published in the United States
by Oxford University Press Inc., New York



First published 2005



British Library Cataloguing in Publication Data

Data available

Library of Congress Cataloging in Publication Data

Data available

ISBN 0 19 852869 8 9780198528692

1 3 5 7 9 10 8 6 4 2

Typeset by Julie M. Harris using LaTeX

Printed in Great Britain
on acid-free paper by
Biddles Ltd., King's Lynn, Norfolk

# PREFACE TO SECOND EDITION

There has been significant progress in the development of multi-domain spectral methods both at the fundamental as well as at the application level in the last few years. We have, therefore, undertaken the 'non-trivial' task of updating the book in order to include these new developments. We also wanted to make these methods easier to comprehend and implement by the students, responding directly to the many requests and feedback we received after the publication of the first edition of our book. We are grateful to Oxford University Press, and in particular to the Mathematics and Statistics editor Dr Alison Jones, who gave us this opportunity.

The new developments are primarily in the discontinuous Galerkin methods, in non-tensorial nodal spectral element methods in simplex domains, and on stabilisation and filtering techniques. From the practical point of view, high-order solutions in complex geometries require high-order meshes and high-order post-processing, a subject that is often neglected in the everyday 'production' computing and simulation. Such subjects are now addressed in some detail in this new edition of the book. We have also seen the spectral/*hp* element method applied to less traditional fields, such as seismology, climate modelling, and magneto-hydrodynamics (MHD), and we have included some elements of modelling such applications in this revised version.

Finally, another objective in revising the book has been to provide more details on implementing various aspects of the method. To this end, we have put some emphasis on implementation and technical issues with exercises in the founding Chapters 2 to 5 to aid in implementing basic spectral element solvers, which can be used as building blocks for more complex application codes.

Overall, the book has been increased in new material by almost 50% in order to include all the aforementioned topics.

We would like to thank our many students and colleagues who have provided good critical feedback on the first version of the book. In particular, we would like to thank Drs H. Blackburn, J. Hesthaven, R. M. Kirby, J. Peiró, V. Theofilis, and Z. Yosibash who contributed to the new edition directly by providing plots and other information from their own work. We also want to thank our families who have supported us again during the course of this effort.

*Boston, Massachusetts* G. E. K.
*London, England* S. J. S.
*Spring* 2004

# PREFACE TO FIRST EDITION

Our aim in writing this book is to introduce a wider audience to the use of spectral/*hp* element methods with particular emphasis on their application to unstructured meshes. These methods, as their name suggests, incorporate both multi-domain spectral methods (based on a development of original ideas by A. T. Patera) and also high-order finite element methods (based on original ideas of B. A. Szabó). In this book, we provide a unified description of both methods building on previously published works as well as on new material not previously published.

Although spectral methods have long been popular in direct and large eddy simulation of turbulent flows, their use in areas with complex-geometry computational domains has historically been much more limited. For example, in computational aerodynamics, which typically involve the use of unstructured meshes, the preferred methods have been low-order finite element and finite volume methods. More recently, however, the need to find accurate solutions to the viscous flow equations around complex aerodynamic configurations has led to the development of high-order discretisation procedures on unstructured meshes. High-order discretisation is also recognised as more efficient for solution of time-dependent oscillatory solutions over long time periods, for example, in the new field of computational electromagnetics in aerospace design.

Polynomial spectral methods were first introduced in Gottlieb and Orszag [199] and are covered extensively in Canuto *et al.* [86], Boyd [71], and Fornberg [164]. Szabó and Babuška [445] and Bernardi and Maday [45] deal with *hp* finite element and spectral element methods, respectively, and should be consulted for background reading. This book reviews most of the fundamental concepts but does assume the reader has some familiarity with the basic concepts of finite element discretisation and the Galerkin approximation technique.

The material contained in this book draws on the ideas and influence of many people, but particular thanks are due to A. T. Patera, S. A. Orszag, and D. Gottlieb, who have advised and collaborated on our research work over the past fifteen years. Much of this book is based on the doctoral thesis of the second author (SJS), as supervised by the first author (GEK). This book has also drawn heavily on doctoral theses of other students supervised by GEK, notably those of R. D. Henderson, I. G. Giannakouros, A. Beskok, C. H. Crawford, T. C. Warburton, I. Lomtev, and J. Trujillo, and also doctoral theses supervised by A. T. Patera, notably E. M. Ronquist, C. Mavriplis, and G. Anagnostou. We would like to thank J. Peiro for his help with the unstructured meshing, and are particularly grateful to our colleagues who provided original figures which we have included in this book. Thanks are also due to those who have given their

time and effort to read the proof of this book.

GEK would like to acknowledge the sponsorship of his research program by the Office of Naval Research, the Air Force Office of Scientific Research, the Department of Energy, and the National Science Foundation.

Finally, we would like to thank our wives, Helen and Tracey, for their understanding and patience during the writing of this book. We would especially like to thank Tracey, who was the first person to read the book in its entirety, correcting our grammar and spelling, and also providing a notable contribution to the prose.

*Boston, Massachusetts* G. E. K.
*London, England* S. J. S.
*Fall* 1997

# BOOK OUTLINE

In Chapter 1, we present reduced models of the compressible and incompressible Navier–Stokes equations which are used in the various discretisation concepts discussed in the following chapters. The convergence philosophy of spectral and finite element methods, the combination of which provides a dual path of convergence, is also introduced.

In Chapter 2, we present the fundamental concepts that are further developed in the remainder of the book, in the context of a one-dimensional formulation. In doing so, we illustrate the principles and underlying theory behind the construction of the spectral/*hp* element method. In this second revision we have included more details on implementing boundary conditions and used margin identifiers to highlight formulation details, using a '♣' symbol, and implementation details, using a '♦' symbol. We have also added new sections on nodal p-type expansions and on integration errors and polynomial aliasing. The series of exercises included at the end of the chapter solely focuses on developing a one-dimensional spectral element solver.

In Chapter 3, we consider the extension of the one-dimensional formulation to two and three dimensions by the development of expansion bases in standard regions such as triangles or rectangles in two dimensions, and tetrahedrons, prisms, pyramids, and hexahedrons in three dimensions. The construction of these bases uses a unified approach which permits the development of computationally efficient expansions. In this new revision we now formulate the modal basis as solutions to a generalised Sturm–Liouville problem. We also present optimal nodal points, the so-called Fekete points, as well as the electrostatic points on a simplex, and include related approximation results. The exercises at the end of the chapter target construction of multi-dimensional elemental matrices.

Compared to the first revision of the book, Chapter 4 has been restructured with extra emphasis on implementation aspects. In this chapter we complete the multi-dimensional formulation by explaining how the two- and three-dimensional expansions developed in Chapter 3 can be extended into a tessellation of multiple domains. These extensions are decomposed into three sections: local operations such as integration and differentiation; global operations such as the construction of global matrix systems; and pre- and post-processing issues such as boundary representation, high-order mesh generation, and particle tracking. The chapter introduces a matrix formulation to help illustrate the algebraic systems which need to be constructed when computationally implementing the spectral/*hp* method. Formulation of both Galerkin and collocation projections are considered in this manner. The exercises in this chapter include implementation of a two-dimensional spectral/*hp* element solver for a multi-element Galerkin

projection.

In this revised version Chapter 5 now considers the diffusion equation; an implicit in time discretisation leads to the Helmholtz equation. This chapter discusses both the temporal discretisation and eigenspectra of second-order operators that dictate time-step restrictions. We also expand on appropriate preconditioning techniques for inversion of the stiffness matrix. The final part of this chapter discusses non-smooth solutions due to geometric singularities and this revised section now includes elements from recent advances in three-dimensional domains. The final exercises section focuses on building on the exercises of Chapters 3 and 4 to implement a two-dimensional standard Galerkin *hp* solution to the Helmholtz problem.

In Chapter 6, we focus on the scalar advection equation and develop a Galerkin discretisation using the techniques described in Chapter 4. We then include an extended presentation of the discontinuous Galerkin formulation for advection equations. Similar to Chapter 5, we also review eigenspectra of the advection operators in both two and three dimensions which are relevant for explicit time stepping. A further new addition is the discussion on two forms of a semi-Lagrangian method for advection (strong and auxiliary forms) that could potentially prove very effective in enhancing the speed and accuracy of spectral/*hp* element methods in advection-dominated problems. A further new section on stabilisation techniques is then introduced that discusses filters, spectral vanishing viscosity, and upwind collocation. This new material is important as it relates directly to the robustness of the method, especially in marginally resolved simulations.

In Chapter 7 (previously Chapter 5) we introduce and discuss the topic of non-conforming elements for second-order operators. This chapter has been extended to include a comprehensive presentation of the discontinuous Galerkin method with a comparison of different versions from theoretical, computational, and implementation standpoints.

In Chapter 8, we present different ways of formulating the incompressible Navier–Stokes equations based on primitive variables, that is, velocity and pressure, as well as velocity–vorticity algorithms. We consider both coupled, splitting, and least-squares formulations for primitive variables, and in this revision we now discuss both the Uzawa coupled algorithm and a new substructured solver. The discussion on primitive variables time-splitting has been rewritten to include recent theoretical advances in the pressure-correction and velocity-correction schemes as well as the rotational formulation of the pressure boundary condition. Whilst an important issue for primitive variable formulation is the efficient incorporation of the divergence-free constraint, an analogous problem for the velocity–vorticity formulation is the accurate imposition of the boundary conditions which is also discussed. The final section is devoted to nonlinear terms; it includes a discussion of spatial and temporal discretisation with focus on the semi-Lagrangian method for the incompressible Navier–Stokes equations.

In Chapter 9, we discuss numerical simulations of the incompressible Navier–Stokes equations. First, we present exact Navier–Stokes solutions that can be used as benchmarks to validate new codes and evaluate the accuracy of a particular discretisation. In the current revision this section has been restructured, with more benchmark solutions, to emphasise verification and validation of spectral/*hp* element solvers. A new section on three-dimensional stability (biglobal stability) is also included—spectral/*hp* elements are particularly effective in this field. The chapter continues by discussing some aspects of direct numerical simulation (DNS) and large-eddy simulation (LES). The issue of stabilisation at high Reynolds number is then presented using the concepts of dynamic subgrid modelling, over-integration, and spectral vanishing viscosity. A new parallel paradigm based on multi-level parallelism is introduced that can help realise adaptive refinement more easily; the final section includes a heuristic refinement method for Navier–Stokes equations.

Chapter 10 has been expanded to consider not only compressible Euler and Navier–Stokes equations but general hyperbolic conservation laws. This is an area in which high-order methods have had little success in the past. The principle issue is how to effectively use the high-order expansions of the spectral/*hp* method whilst honouring the inherent monotonicity and conservation properties of the analytic system. We consider different ways of dealing with these fundamental issues for both the Euler and the Navier–Stokes equations. A new section for the shallow water equations is also included and the section on the discontinuous Galerkin method has been rewritten. Finally, the last section discusses modelling of plasma flows, i.e., the so-called magneto-hydrodynamic (MHD) equations.

Finally, the appendices provide details on Jacobi and Askey polynomials as well as numerical integration and differentiation which are essential building blocks of the spectral/*hp* element techniques. A full description of commonly used expansion bases is provided, which now includes nodal points for non-tensorial electrostatic and Fekete point distribution in simplex domains. The final appendix also details Riemann solvers commonly used in the solution of the Euler equations.

# CONTENTS

# NOMENCLATURE

*Sets*

| | |
|---|---|
| $\Re$ | Real numbers |
| $\bigcup$ | Set union |
| $\bigcap$ | Set intersection |
| $\emptyset$ | Empty set |
| $\in$ | Is a member of; belongs to |
| $\notin$ | Is not a member of; does not belong to |
| $\subset$ | Is a subset of |
| $\not\subset$ | Is not a subset of |

*Expansion basis notation*

| | |
|---|---|
| $\phi_{pq}, \phi_{pqr}$ | Expansion basis |
| $\psi^a_p, \psi^b_{pq}, \psi^c_{pqr}$ | Modified principal functions |
| $\widetilde{\psi}^a_p, \widetilde{\psi}^b_{pq}, \widetilde{\psi}^c_{pqr}$ | Orthogonal principal functions |
| $h_p(\xi)$ | One-dimensional Lagrange polynomial of order $p$ |
| $L_i^{N_m}(\boldsymbol{\xi})$ | Two-dimensional Lagrange polynomial through $N_m$ nodes $\boldsymbol{\xi}_i$ |
| $P_i$ | Polynomial order in the $i$th direction |
| $Q_i$ | Quadrature order in the $i$th direction |
| $x_1, x_2, x_3, \boldsymbol{x}$ | Global Cartesian coordinates |
| $\xi_1, \xi_2, \xi_3, \boldsymbol{\xi}$ | Local Cartesian coordinates |
| $\eta_1, \eta_2, \eta_3$ | Local collapsed Cartesian coordinates |
| $\chi_i(\xi)$ | Local Cartesian to global coordinate mapping |

*Various constants*

| | |
|---|---|
| $N_{\text{dof}}$ | Number of global degrees of freedom |
| $N_b$ | Number of global boundary degrees of freedom |
| $N_m$ | Number of elemental degrees of freedom |
| $N_{\text{eof}}$ | Total number of elemental degrees of freedom $N_{\text{eof}} \simeq N_{\text{el}} N_m$ |
| $N_Q$ | Total number of quadrature points $N_Q = Q_1 Q_2 Q_3$ |
| $N_{\text{el}}$ | Number of elements |
| $\lambda$ | Helmholtz equation constant |
| $e$ | Element number $1 \leqslant e \leqslant N_{\text{el}}$ |
| $i, j, k$ | General summation indices |
| $p, q, r$ | General summation indices |

*Elemental arrays*

| | |
|---|---|
| $\boldsymbol{B}$ | Basis matrix |
| $\boldsymbol{W}$ | Diagonal weight/Jacobian matrix |
| $\boldsymbol{D}_\xi$ | Elemental derivative matrix with respect to $\xi$ |
| $\boldsymbol{\Lambda}(u)$ | Diagonal matrix of $u(\xi_1, \xi_2)$ evaluated at quadrature points |
| $\boldsymbol{M}^e$ | Elemental mass matrix |

| | |
|---|---|
| $\boldsymbol{L}^e$ | Elemental Laplacian matrix |
| $\boldsymbol{H}^e$ | Elemental Helmholtz matrix |
| $\boldsymbol{f}^e$ | Force vector of the $e$th element |
| $\boldsymbol{u}^e$ | Vector containing function evaluated at quadrature points |
| $\hat{\boldsymbol{u}}^e$ | Vector of expansion coefficients |

*Global arrays*

| | |
|---|---|
| $\boldsymbol{W^e}$ | Block diagonal extension of matrix $\boldsymbol{W^e}$ |
| $\boldsymbol{f}^e$ | Concatenation of elemental vector $\boldsymbol{f}^e$ |
| $\boldsymbol{\mathcal{A}}^\top$ | Matrix global assembly |
| $\boldsymbol{M}$ | Mass matrix ($= \boldsymbol{\mathcal{A}}^\top \underline{\boldsymbol{M}^e} \boldsymbol{\mathcal{A}}$) |
| $\boldsymbol{L}$ | Laplacian matrix ($= \boldsymbol{\mathcal{A}}^\top \underline{\boldsymbol{L}^e} \boldsymbol{\mathcal{A}}$) |
| $\boldsymbol{H}$ | Helmholtz matrix ($= \boldsymbol{\mathcal{A}}^\top \underline{\boldsymbol{H}^e} \boldsymbol{\mathcal{A}}$) |
| $\hat{\boldsymbol{v}}_g$ | List of all elemental coefficients ($= \underline{v^e}$) |
| $\hat{\boldsymbol{v}}_l$ | Global list of coefficients |

*Solution domains*

| | |
|---|---|
| $\Omega$ | Solution domain |
| $\partial\Omega$ | Boundary of $\Omega$ |
| $\Omega^e$ | Elemental region |
| $\partial\Omega^e$ | Boundary of $\Omega^e$ |
| $\partial\Omega_{\mathcal{D}}$ | Domain boundary with Dirichlet conditions |
| $\partial\Omega_{\mathcal{N}}$ | Domain boundary with Neumann conditions |
| $\boldsymbol{n}$ | Unit outward normal |

*Differential operators*

| | |
|---|---|
| $\nabla^2$ | Laplacian |
| $\nabla \cdot$ | Divergence |
| $\nabla \times$ | Curl |

*Spaces*

| | |
|---|---|
| $\mathcal{X}$ | Space of trial solutions |
| $\mathcal{X}^\delta$ | Finite-dimensional space of trial solutions |
| $\mathcal{V}$ | Space of test functions |
| $\mathcal{V}^\delta$ | Finite-dimensional space of test functions |
| $\mathcal{P}_P(\Omega)$ | Polynomial space of order $P$ over $\Omega$ |

*Operators*

| | |
|---|---|
| $\mathcal{I}$ | Interpolation operator |
| $\mathcal{I}^\delta$ | Discrete interpolation operator |
| $\mathbb{P}$ | Projection operator |
| $\mathbb{P}^\delta$ | Discrete projection operator |
| $\mathbb{L}(u)$ | Linear operator in $u$ |

*Fluid variables*

| | |
|---|---|
| $\mathbf{v}$ | Velocity $[u, v, w]^\top$ |
| $p$ | Pressure |
| $\boldsymbol{\omega}$ | Vorticity |
| $\rho$ | Density |
| $\mu, \nu$ | Dynamic, kinematic viscosities |

# 1

# INTRODUCTION

## 1.1 The basic equations of fluid dynamics

Consider fluid flow in the non-deformable control volume $\Omega$ bounded by the control surface $\partial\Omega$ with $\boldsymbol{n}$ being the unit outward normal. The equations of motion can then be derived in an absolute reference frame by applying the principles of mechanics and thermodynamics [39]. They can be formulated in integral form for mass, momentum, and total energy, respectively, as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\,\mathrm{d}\Omega+\int_{\partial\Omega}\rho\mathbf{v}\cdot\boldsymbol{n}\,\mathrm{d}S=0\,, \tag{1.1.1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}\rho\mathbf{v}\,\mathrm{d}\Omega+\int_{\partial\Omega}[\rho\mathbf{v}(\mathbf{v}\cdot\boldsymbol{n})-\boldsymbol{n}\boldsymbol{\sigma}]\,\mathrm{d}S=\int_{\Omega}\boldsymbol{f}\,\mathrm{d}\Omega\,, \tag{1.1.1b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega}E\,\mathrm{d}\Omega+\int_{\partial\Omega}(E\mathbf{v}-\boldsymbol{\sigma}\mathbf{v}+\mathbf{q})\cdot\boldsymbol{n}\,\mathrm{d}S=\int_{\Omega}\boldsymbol{f}\cdot\mathbf{v}\,\mathrm{d}\Omega\,. \tag{1.1.1c}$$

Here $\mathbf{v}(\boldsymbol{x},t)=(u,v,w)$ is the velocity field, $\rho$ is the density, and $E=\rho(e+1/2\mathbf{v}\cdot\mathbf{v})$ is the total specific energy where $e$ represents the internal specific energy. Also, $\boldsymbol{\sigma}$ is the stress tensor, $\mathbf{q}$ is the heat flux vector, and $\boldsymbol{f}$ represents all external forces acting on this control volume. For Newtonian fluids, the stress tensor, which consists of the normal components ($p$ for pressure) and the viscous stress tensor $\boldsymbol{\tau}$, is a *linear* function of the velocity gradient, that is,

$$\boldsymbol{\sigma}=-p\mathbf{I}+\boldsymbol{\tau}\,, \tag{1.1.2a}$$

$$\boldsymbol{\tau}=\mu[\nabla\mathbf{v}+(\nabla\mathbf{v})^{\top}]+\lambda(\nabla\cdot\mathbf{v})\mathbf{I}\,, \tag{1.1.2b}$$

where $\mathbf{I}$ is the unit tensor, and $\mu$ and $\lambda$ are the first and second coefficients of viscosity, respectively. They are related by the Stokes hypothesis, that is, $2\mu+3\lambda=0$, which expresses local thermodynamic equilibrium. The heat flux vector is related to temperature gradients via the Fourier law of diffusion, that is,

$$\mathbf{q}=-k\nabla T\,, \tag{1.1.3}$$

where $k(T)$ is the thermal conductivity which may be a function of temperature $T$.

In the case of a deformable control volume, the velocity in the flux term should be recognised as in a frame of reference relative to the control surface

and the appropriate time rate-of-change term be used. Considering, for example, the mass conservation equation, we have the forms

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \rho \,\mathrm{d}\Omega + \int_{\partial\Omega} \rho \mathbf{v}_r \cdot \boldsymbol{n}\,\mathrm{d}S = 0$$

or

$$\int_\Omega \frac{\partial \rho}{\partial t}\,\mathrm{d}\Omega + \int_{\partial\Omega} \rho \mathbf{v}_r \cdot \boldsymbol{n}\,\mathrm{d}S + \int_{\partial\Omega} \rho \mathbf{v}_{\mathrm{cs}} \cdot \boldsymbol{n}\,\mathrm{d}S = 0\,,$$

where $\mathbf{v}_{\mathrm{cs}}$ is the velocity of the control surface, $\mathbf{v}_r$ is the velocity of the fluid with respect to the control surface, and the total velocity of the fluid with respect to the chosen frame is $\mathbf{v} = \mathbf{v}_r + \mathbf{v}_{\mathrm{cs}}$. The above forms are equivalent but the first expression may be more useful in applications where the time history of the volume is of interest.

Equations (1.1.1a)–(1.1.1c) can be transformed into an equivalent set of partial differential equations by applying Gauss' theorem (assuming that sufficient conditions of differentiability exist), that is,

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0\,, \tag{1.1.4a}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v} - \sigma) = \boldsymbol{f}\,, \tag{1.1.4b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (E\mathbf{v} - \boldsymbol{\sigma}\mathbf{v} + \mathbf{q}) = \boldsymbol{f} \cdot \mathbf{v}\,. \tag{1.1.4c}$$

The momentum and energy equations can be rewritten in the following form by using the continuity equation (1.1.4a) and the constitutive equations (1.1.2a) and (1.1.2b)

$$\rho \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla p + \nabla \cdot \boldsymbol{\tau} + \boldsymbol{f}\,, \tag{1.1.5a}$$

$$\rho \frac{\mathrm{D}e}{\mathrm{D}t} = -p\nabla \cdot \mathbf{v} - \nabla \cdot \mathbf{q} + \Phi\,, \tag{1.1.5b}$$

where $\Phi = \boldsymbol{\tau} \cdot \nabla \mathbf{v}$ is the dissipation function and $\mathrm{D}/\mathrm{D}t = \partial/\partial t + \mathbf{v} \cdot \nabla$ is the material derivative.

In addition to the governing conservation laws, an equation of state is required. For ideal gases, it has the simple form

$$p = \rho \mathcal{R} T\,, \tag{1.1.6}$$

where $\mathcal{R}$ is the ideal gas constant defined as the difference of the constant specific heats, that is, $\mathcal{R} = C_p - C_v$, where $C_v = (\partial e/\partial T)|_\rho$ and $C_p = \gamma C_v$, with $\gamma$ being the adiabatic index. For ideal gases, the energy equation can be rewritten in terms of the temperature since $e = p/(\rho(\gamma - 1)) = C_v T$, and so eqn (1.1.5b) becomes

$$\rho C_v \frac{\mathrm{D}T}{\mathrm{D}t} = -p\nabla \cdot \mathbf{v} + \nabla \cdot (k \nabla T) + \Phi\,. \tag{1.1.7}$$

The system of equations (1.1.4a), (1.1.5a), (1.1.6), and (1.1.7) is called the *compressible Navier–Stokes equations* and contains six unknown variables $(\rho, \mathbf{v}, p, T)$

with six scalar equations. This is an *incomplete parabolic* system as there are no second-order derivative terms in the continuity equation.

A hyperbolic system arises in the case of inviscid flow, that is, $\mu = 0$ (assuming that we also neglect heat losses by thermal diffusion, that is, $k = 0$). In that case we obtain the *Euler equations*, which in the absence of external forces or heat sources have the form

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0\,, \tag{1.1.8a}$$

$$\frac{\partial (\rho \mathbf{v})}{\partial t} + \nabla \cdot (\rho \mathbf{v}\mathbf{v}) = -\nabla p\,, \tag{1.1.8b}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot [(E + p)\mathbf{v}] = 0\,. \tag{1.1.8c}$$

Appropriate boundary conditions will be discussed in Chapter 10. This system admits discontinuous solutions, and it can also describe the transition from a subsonic flow (where $|\mathbf{v}| < c$) to supersonic flow (where $|\mathbf{v}| > c$), where $c = (\gamma \mathcal{R} T)^{1/2}$ is the speed of sound. Typically, the transition is obtained through a shock wave, which represents a discontinuity in flow variables. In such a region the integral form of the equations should be used by analogy with eqns (1.1.1a)–(1.1.1c).

### 1.1.1 *Incompressible flow*

For an incompressible fluid, where $\mathrm{D}\rho/\mathrm{D}t = 0$, the mass conservation (or continuity) equation simplifies to

$$\nabla \cdot \mathbf{v} = 0\,. \tag{1.1.9a}$$

Typically, when we refer to an incompressible fluid we mean that $\rho =$ constant, but this is not necessary for a divergence-free flow; for example, in thermal convection the density varies with temperature variations. The corresponding momentum equation has the form

$$\rho \frac{\mathrm{D}\mathbf{v}}{\mathrm{D}t} = -\nabla p + \nabla \cdot \left[\mu[\nabla \mathbf{v} + (\nabla \mathbf{v})^{\top}]\right] + \boldsymbol{f}\,, \tag{1.1.9b}$$

where the viscosity $\mu(\boldsymbol{x}, t)$ may vary in space and time due to physics or a subgrid model in formulations of large-eddy simulations (see Chapter 9). The pressure $p(\boldsymbol{x}, t)$ is not a thermodynamic quantity but can be thought of as a constraint that projects the solution $\mathbf{v}(\boldsymbol{x}, t)$ onto a divergence-free space. In other words, the isentropic equation $p = C\rho^{\gamma}$ is no longer valid as it will make the incompressible Navier–Stokes system over-determined.

The acceleration terms can be written in various equivalent ways so that, in their discrete form, they conserve total linear momentum $\int_\Omega \mathbf{v}\,\mathrm{d}\Omega$ and total kinetic energy $\int_\Omega \mathbf{v} \cdot \mathbf{v}\,\mathrm{d}\Omega$ in the absence of viscosity and external forces. In particular, the following forms are often used:

- convective form: $\mathrm{D}\mathbf{v}/\mathrm{D}t = \partial \mathbf{v}/\partial t + (\mathbf{v} \cdot \nabla)\mathbf{v}$;

- conservative (flux) form: $\mathrm{D}\mathbf{v}/\mathrm{D}t = \partial\mathbf{v}/\partial t + \nabla\cdot(\mathbf{v}\mathbf{v})$;
- rotational form: $\mathrm{D}\mathbf{v}/\mathrm{D}t = \partial\mathbf{v}/\partial t - \mathbf{v}\times(\nabla\times\mathbf{v}) + 1/2\nabla(\mathbf{v}\cdot\mathbf{v})$;
- skew-symmetric form: $\mathrm{D}\mathbf{v}/\mathrm{D}t = \partial\mathbf{v}/\partial t + 1/2[(\mathbf{v}\cdot\nabla)\mathbf{v} + \nabla\cdot(\mathbf{v}\mathbf{v})]$.

In semi-discrete systems, that is, systems that are continuous in time but discrete in space, where inexact integration of nonlinear terms or pointwise discretisation (for example, collocation) is employed, only the rotational and skew-symmetric forms conserve both linear momentum and kinetic energy in the inviscid limit. The flux form conserves only linear momentum while the convective form conserves neither. Numerical experiments with turbulence simulations have shown that the skew-symmetric form is the most effective in minimising aliasing errors although it is computationally the most expensive. In three dimensions, it requires the calculation of eighteen derivatives versus six derivatives in the rotational form and nine derivatives in the convective form. A more detailed discussion on the discrete form of the advection terms is included in Chapter 9.

The incompressible Navier–Stokes equations (1.1.9a) and (1.1.9b) are written in terms of the primitive variables $(\mathbf{v}, p)$. An alternative form is to rewrite these equations in terms of the velocity $\mathbf{v}$ and vorticity $\boldsymbol{\omega} = \nabla\times\mathbf{v}$. This is a more general formulation than the standard vorticity streamfunction, which is limited to two dimensions. The following system is equivalent to eqns (1.1.9b) and (1.1.9a) assuming that $\rho$ and $\mu$ are constant:

$$\rho\frac{\mathrm{D}\boldsymbol{\omega}}{\mathrm{D}t} = (\boldsymbol{\omega}\cdot\nabla)\mathbf{v} + \mu\nabla^2\boldsymbol{\omega} \quad \text{in } \Omega, \tag{1.1.10a}$$

$$\nabla^2\mathbf{v} = -\nabla\times\boldsymbol{\omega} \quad \text{in } \Omega, \tag{1.1.10b}$$

$$\nabla\cdot\mathbf{v} = 0 \quad \text{in } \Omega, \tag{1.1.10c}$$

$$\boldsymbol{\omega} = \nabla\times\mathbf{v} \quad \text{in } \Omega, \tag{1.1.10d}$$

where the elliptic equation for the velocity $\mathbf{v}$ is obtained using a vector identity and the divergence-free constraint. We also assume here that the domain $\Omega$ is simply connected. An equivalent system in terms of velocity and vorticity is studied in Chapter 8, where it is reformulated for easier implementation. The problem with the lack of direct boundary conditions for the vorticity also exists in the more-often-used vorticity-streamfunction formulation.

Finally, a note regarding non-dimensionalisation. Consider the free-stream flow $U_0$ past a body of characteristic size $D$ in a medium of dynamic viscosity $\mu$, as shown in Fig. 1.1. There are two characteristic time-scales in the problem, the first one representing the convective time-scale $t_c = D/U_0$, and the second one representing the diffusive time-scale $t_d = D^2/\nu$, where $\nu = \mu/\rho$ is the kinematic viscosity. If we non-dimensionalise all lengths with $D$, the velocity field with $U_0$, and the vorticity field with $U_0/D$, we obtain two different non-dimensional equations corresponding to the choice of the time non-dimensionalisation:

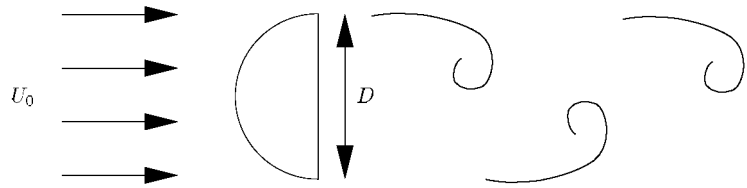



FIG. 1.1. Free-stream flow past a half-cylinder in a viscous fluid.

$$\frac{\partial \boldsymbol{\omega}}{\partial t_c^*} + \nabla \cdot (\mathbf{v}\boldsymbol{\omega}) = (\boldsymbol{\omega} \cdot \nabla)\mathbf{v} + \mathrm{Re}^{-1}\nabla^2\boldsymbol{\omega}\,,$$
$$\frac{\partial \boldsymbol{\omega}}{\partial t_d^*} + \mathrm{Re}\nabla \cdot (\mathbf{v}\boldsymbol{\omega}) = \mathrm{Re}(\boldsymbol{\omega} \cdot \nabla)\mathbf{v} + \nabla^2\boldsymbol{\omega}\,,$$

where $t_c^*$ and $t_d^*$ are the non-dimensionalised time variables with respect to $t_c$ and $t_d$, respectively, and $\mathrm{Re} = U_0 D/\nu$ is the Reynolds number. Both forms are useful in simulations, the first in high Reynolds number simulations, and the second in low Reynolds number (creeping) flows.

When the nonlinear terms can be neglected we obtain the *Stokes equations*, which we can cast in the form

$$-\nu\nabla^2\mathbf{v} + \nabla p = \boldsymbol{f} \quad \text{in } \Omega\,, \tag{1.1.11a}$$
$$\nabla \cdot \mathbf{v} = 0 \quad \text{in } \Omega\,, \tag{1.1.11b}$$

along with appropriate boundary conditions for $\mathbf{v}$. This system is studied in detail in Chapter 8 as it provides the setting for the variational formulation of the Navier–Stokes equations.

The theory on incompressible Navier–Stokes equations is treated in several articles and books, including Constantin and Foias [107], Lions [306], and Temam [454]. The latter provides a theoretical framework for many of the implementation concepts that will be developed in this book in the context of variational formulation of the incompressible Navier–Stokes equations. It is fair to say, however, that the mathematical theory is fairly complete in two spatial dimensions and that several results regarding the existence of solutions, uniqueness, regularity, and continuous dependence on the data have been proved. However, there are several questions still unanswered regarding the Navier–Stokes equation in three dimensions, and little progress has been made in the theory since the work of Leray [293].

New concepts from modern dynamical systems theory introduced in the context of the incompressible Navier–Stokes system have helped in our understanding of the computational complexity of these equations. For example, the concept of determining modes [108] provided for the first time a rigorous theory for the classical Kolmogorov heuristic argument on the excited degrees of freedom in turbulence. Moreover, the analysis by Constantin *et al.* [108] showed that such

an estimate is only a sufficient upper bound and applies to general unsteady flows for which boundedness of vorticity is assumed.

### 1.1.2 *Reduced models*

The mathematical nature of the Navier–Stokes equations varies depending on the flow that we model and the corresponding terms that dominate in the equations. For example, for an inviscid compressible flow, we obtain the Euler equations, which are of hyperbolic nature, whereas the incompressible Euler equations are of hybrid type corresponding to both real and imaginary eigenvalues. The unsteady incompressible Navier–Stokes equations are of mixed parabolic/hyperbolic nature, but the steady incompressible Navier–Stokes are of elliptic/parabolic type. To simplify the discretisation of Navier–Stokes and motivate the formulation adopted in this book we follow a hierarchical approach in reducing the Navier–Stokes equations to simpler equations, so that each introduces one new concept.

Take as an example the incompressible Navier–Stokes equations (1.1.9a) and (1.1.9b), a simpler model is the *unsteady Stokes* system. This retains all the complexity but not the nonlinear terms, that is,

$$\frac{\partial \mathbf{v}}{\partial t} = -\frac{\nabla p}{\rho} + \nu \nabla^2 \mathbf{v} + \boldsymbol{f}\,,$$
$$\nabla \cdot \mathbf{v} = 0\,.$$

The Stokes system (eqns (1.1.11a) and (1.1.11b)) is recovered by dropping the time derivative. Alternatively, we can drop the divergence-free constraint and study the purely parabolic scalar equation for a variable $u$, that is,

$$\frac{\partial u}{\partial t} = \nu \nabla^2 u + f\,. \tag{1.1.12}$$

This equation is very helpful in studying the stability properties of the Navier–Stokes equations and analysing different time-stepping schemes. If we instead drop all terms on the right-hand side of eqn (1.1.9b), as well as the divergence-free constraint, we obtain a nonlinear advection equation. This equation also serves as a good model for studying time-stepping algorithms and issues associated with the stability and long-time integration of the Navier–Stokes equations. These topics are presented in Chapters 6 and 10.

Finally, by dropping the time derivative in the parabolic equation (1.1.12), we obtain the *Poisson* equation $-\nu \nabla^2 u = f$, which is useful in dictating the continuity requirements and corresponding functional spaces in the variational formulation context adopted in this book. A treatment of the one-dimensional problem in Chapter 2 and in multiple dimensions in Chapter 3 is based on the Poisson equation. In addition, the study of solution algorithms appropriate for the global system inversion required in the Navier–Stokes equations is motivated by the Poisson equation and is covered in Chapter 4.

## 1.2 Numerical discretisations

### 1.2.1 *The finite element method*

There are more than one hundred thousand references on the finite element method today, including textbooks, monographs, conference proceedings, and journals. The majority of these references are devoted to structural mechanics, but finite element methods have also proved very successful in fluid dynamics, although the initial developments did not target this field. The reason for this may be the difficulties with the nonlinear terms in the Navier–Stokes equations and the original difficulties with the application of finite element methods to non-symmetric operators.

The idea of building up a solution to a differential equation from a sequence of local approximations is an old one. While Courant [112] used a network of triangles to represent with piecewise linear interpolation an approximate solution to the Dirichlet problem, it was Argyris [14] who introduced the variational method of approximation. Patching the triangles or other subdomains (elements) together is an automatic procedure today known as 'global assembly', or 'direct stiffness assembly', and was introduced in analysing the structural behaviour of various components of an aircraft. The 1960s were the formative years of finite element methods, focusing primarily on linear plane elasticity problems. Most of the earlier finite element methods used a low-order polynomial approximation as expansion basis, with the exception of the work of Oden [341] who used Fourier series in an assembly of rectangular subdomains. This was perhaps the first attempt to develop spectral elements which provide high-order piecewise approximations. A comprehensive review of the significant developments in finite element methodology and its mathematical theory is given in [342]. There are also many textbooks that develop the fundamental ideas of the finite element discretisation, including the two volumes by Zienkiewicz and Taylor [511], the six-volume series by Carey and Oden [88], the standard textbook by Hughes [247], and the more theoretical book of Brenner and Scott [75].

Finite elements were introduced in fluid mechanics in the late 1970s and were used routinely in large-scale codes in flow simulations in the 1980s. A major contribution to these developments came from the theoretical work of Babuška [22] and Brezzi [76] on the so-called *inf–sup* condition that is very useful in studying constrained elliptic problems. The discretisation of the Stokes problem requires the satisfaction of such a condition to produce a stable finite element discretisation. The monograph by Girault and Raviart [191] presents an in-depth analysis of the Stokes problem on these issues.

Finite elements have been used very successfully in inviscid aerodynamic simulations [379] where the geometric complexity involved makes finite difference methods less efficient. The algorithms developed in computational aerodynamics allow for very efficient discretisation techniques and unstructured mesh generation strategies based on fast triangulation and tetrahedralisation algorithms. Since the accomplishment of the accurate solution of the Euler equations on un-

structured meshes, however, interest has now shifted to the simulation of time-dependent Navier–Stokes equations requiring accurate resolution of boundary layers and minimum dispersion errors over a long-time integration interval.

### 1.2.2 *Spectral discretisation*

The formulation of modern spectral methods was first presented in the monograph of Gottlieb and Orszag [199]. Multi-dimensional discretisations were formulated as tensor products of one-dimensional constructs in separable domains, that is, orthogonal simply-connected domains. The textbook of Canuto *et al.* [86] focuses on fluid dynamics algorithms and includes both practical, as well as theoretical, aspects of global spectral methods.

Global spectral methods use a single representation of a function $u(x)$ throughout the domain via a truncated series expansion, for instance,

$$u(x) \approx u_N(x) = \sum_{n=0}^{N} \hat{u}_n \phi_n \,,$$

where $\phi_n(x)$ are the basis functions. This series is then substituted into a differential (or integral) equation and upon the minimisation of the residual function the unknown coefficients $\hat{u}_n$ are computed. The basis functions may be the often-used Chebyshev polynomials $T_n(x)$, the Legendre polynomials $L_n(x)$, or another member of the family of the Jacobi polynomials $P_n^{\alpha,\beta}$ (see Appendix A).

Spectral methods can be broadly classified into two categories: the pseudo-spectral or collocation methods and the modal or Galerkin methods. The first category is associated with a grid, that is, a set of nodes, and that is why it is sometimes referred to as *nodal* methods. The unknown coefficients $\hat{u}$ are then obtained by requiring the residual function to be zero exactly at a set of nodes. The second category is associated with the method of weighted residuals where the residual function is weighted with a set of *test functions* and after integration is set to zero. The test functions are the same as the basis functions, but in the so-called Petro–Galerkin formulation they may be different. Spectral-tau methods are similar to Galerkin methods, but the boundary conditions are satisfied by a supplementary set of equations and not directly via the basis functions. Another difference is that in the collocation approach the coefficients represent the nodal value of the physical variable, unlike the Galerkin or the spectral-tau method.

The convergence of both Galerkin and pseudo-spectral method is exponential, similar to the Fourier spectral method. This property follows directly from the theory of singular Sturm–Liouville boundary value problems—the basis functions are such solutions. Unlike finite element and finite difference methods, the order of the convergence is not fixed and it is related to the maximum regularity of the solution. Exponential or *spectral convergence* for a very smooth solution, in practice, implies that as the number of collocation points or the number of modes is doubled, the error in the numerical solutions decreases by at least two orders of magnitude and not a fixed factor as in low-order methods. This fast convergence is easily lost if the solution has finite regularity or if the domain is irregular.

### 1.2.3 *Why high-order accuracy in CFD?*

High-order numerical methods, that is, spectral and implicit finite difference schemes, have been used almost exclusively in the direct numerical simulation of turbulent flows in the last two decades [265]. They provide fast convergence, small diffusion and dispersion errors, easier implementation of the *inf–sup* condition for the incompressible Navier–Stokes equations, better data volume-over-surface ratio for efficient parallel processing, and better input/output handling due to the smaller volume of data.

For many engineering applications where accuracy of the order of 10% is acceptable, quadratic convergence is usually sufficient for stationary problems. However, this may not be true in time-dependent flow simulations where long-time integration is required. Therefore, we must ask how *long-time integration* relates to the formal order of accuracy of a numerical scheme, and what is the corresponding computational cost? Consider the convection of a waveform at a constant speed. Let us now assume that there are $N^{(k)}$ grid points required per wavelength to reduce the error to a level $\varepsilon$, where $k$ denotes the formal order of the scheme. In addition, let us assume that we integrate for $M$ *time periods.* We can neglect temporal errors $\mathcal{O}(\Delta t)^J$ (where $J$ is the order of the time integration) by assuming a sufficiently small time-step $\Delta t$. We wish to estimate the phase error in this simulation for second- $N^{(2)}$, fourth- $N^{(4)}$, and sixth- $N^{(6)}$ order finite difference schemes.

The following results can be obtained by following the analysis of Kreiss and Oliger [287]. Assuming an 'engineering accuracy' of $\varepsilon = 10\%$, we obtain

$$N^{(2)} \propto 20M^{1/2}, \quad N^{(4)} \propto 7M^{1/4}, \quad N^{(6)} \propto 5M^{1/6}.$$

We therefore see that the required resolution depends on the number of time periods $M$, and the lower the order of the scheme the stronger that dependence. To compare the corresponding computational cost $W^{(k)}$, we have to consider that higher-order schemes have wider stencils and thus a higher operation count. For this particular example, we find that the work required to achieve 10% accuracy is

$$W^{(2)} \propto 20M^{1/2}, \quad W^{(4)} \propto 14M^{1/4}, \quad W^{(6)} \propto 15M^{1/6},$$

where the superscripts on $W$ refer to the order of the scheme. In Fig. 1.2 we compare the efficiency of these three different discretisations for the same phase error by plotting the computational work required to maintain an 'engineering' accuracy of $\varepsilon = 10\%$ versus the number of time periods for the integration. This comparison favours the fourth-order scheme for short times ($M \propto \mathcal{O}(1)$) over both the second-order and the sixth-order schemes. However, for long-time integration ($M \propto \mathcal{O}(100)$), even for this engineering accuracy of 10%, the sixth-order scheme is superior as the corresponding operation count $W^{(6)}$ is about 6 times lower than the operation count of the second-order scheme $W^{(2)}$, and half the work of the fourth-order scheme $W^{(4)}$.

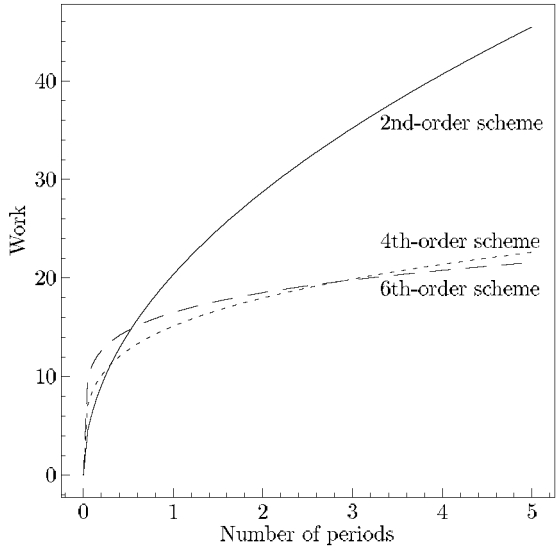



FIG. 1.2. Computational work (number of floating-point operations) required to integrate a linear advection equation for $M$ periods while maintaining a cumulative phase error of $\varepsilon = 10\%$.

In a full Navier–Stokes simulation other considerations may be in place, including the time discretisation, which may change the 'break-even' point regarding the order of the scheme with the highest efficiency. The trend that we have established between resolution requirements and formal accuracy order is still valid for engineering accuracy despite, perhaps, our perception of the opposite! For an accuracy of 1% in the solution of this convection problem, the sixth-order scheme is superior even for short-time integration. For example, for one time period (for example, $M = 1$) the sixth-order scheme costs about 37% of the second-order scheme and 90% of the fourth-order scheme. In the limit of high accuracy and long-time integration, similar analysis suggests that spectral-based algorithms are computationally more efficient.

The reason why high accuracy is required in fluid dynamics simulations, even in stationary flow, can be demonstrated by considering the zero Reynolds number (Stokes) flow in a wedge with a driven lid (see Fig. 1.3). Using a similarity solution Moffatt [334] derived an asymptotic result for the strength and location of an 'infinite' number of eddies generated inside the wedge. The relations are dependent on the wedge angle; for example, for a wedge angle (28.1°) it is predicted that the strength of each eddy should asymptotically (that is, away from the forced top section) be about 406 times weaker than the previous eddy. This means that in order to resolve more than eight eddies, scales twenty orders of magnitude apart have to be captured. Moffatt's measure of the 'intensity' of

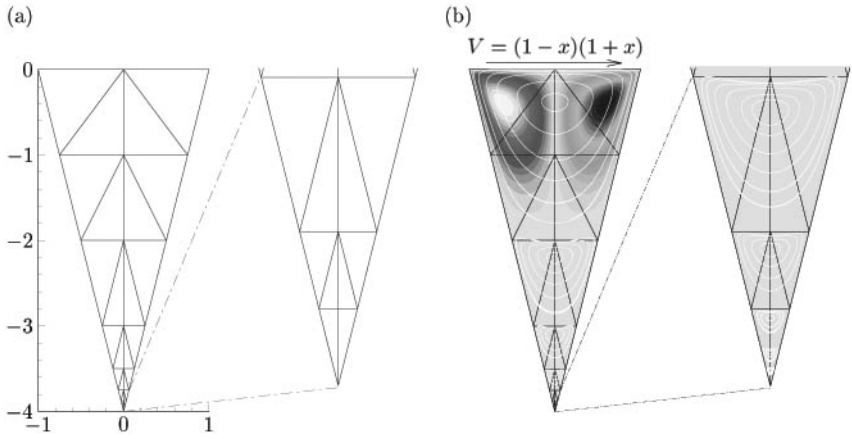



FIG. 1.3. (a) A wedge with an aspect ratio of 2 : 1 was discretised using thirty elements and an expansion order of $P = 17$. Stokes flow was then computed in this domain driven by a prescribed lid velocity. At steady state, nine eddies were observed, as indicated by the streamline plot in (b) (there are three eddies in the last two elements).

successive eddies was the ratio of the local maximum transverse velocity along the centre-line. Therefore, if we take a profile along the centre-line and plot the transverse ($x_1$ direction) velocity as a function of perpendicular distance from the top of wedge, then we obtain the distributions shown in Fig. 1.4 for the four resolutions using high-order expansions.

At the centre of an eddy we expect the transverse velocity to be zero, which

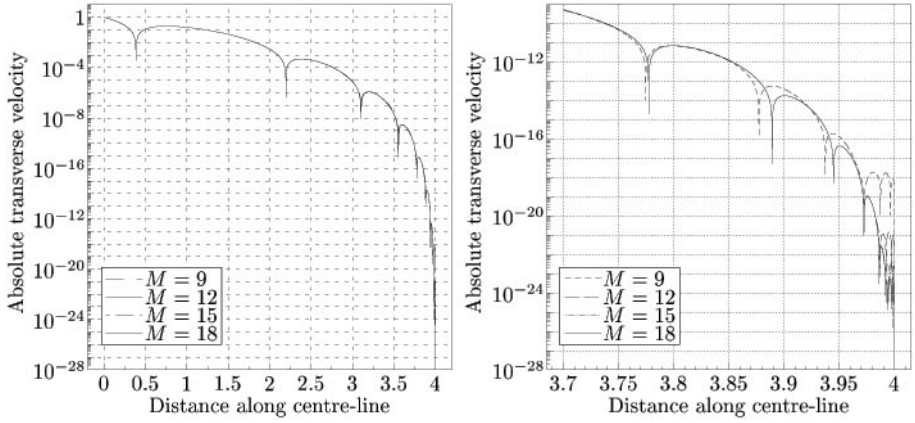



FIG. 1.4. The centre-line transverse velocity as a function of perpendicular height from the top of the wedge shown in Fig. 1.3. (Expansion order $P = M - 1$.)

is indicated by the spikes in Fig. 1.4. After each spike we note that there are local maxima which we use to determine the ratio of maximum velocities in order to evaluate Moffatt's eddy 'intensity'. We note from Fig. 1.4 that at all resolutions the first four eddies have been resolved to the accuracy of the plot. Refining the mesh at the lower corner ($p$-refinement, see next section) we are able to resolve up to nine eddies with only modest total resolution. We will revisit this example in Chapter 8.

### 1.2.4 *Structured versus unstructured discretisation*

The refinement procedure required in the wedge flow becomes more efficient if unstructured discretisation is employed. Features such as selective local refinement and efficient mesh adaptation are critically dependent on the flexibility of a discretisation in decomposing a computational domain into triangles and tetrahedra or other polymorphic elements in three dimensions or into *non-conforming* quadrilateral and hexahedral elements. This class of discretisations is what we characterise as unstructured. While structured discretisations have been the prevailing choice so far for static or quasi-static problems, it is inevitable that with the emphasis shifting towards time-dependent problems unstructured discretisation on non-fixed grids will be used almost exclusively in the future.

The theory required for non-conforming discretisations is presented in Chapter 7. Here we give an example for a two-dimensional unsteady flow past a half-cylinder (see Fig. 1.1) using a non-conforming and a conforming mesh in the near wake. The conforming mesh uses 276 elements, while the non-conforming mesh uses 176 elements. In Fig. 1.5 we plot vorticity contours for the two discretisations. The vorticity is obtained from $\nabla \times \mathbf{v}$, and thus the 'noisy' solution on the conforming mesh can be interpreted as a measure of the under-resolution that arises because of small-scale features unresolved by the mesh. The singular corner presents an additional difficulty in modelling this flow. The non-conforming discretisation of the domain isolates these features within a few elements close to the cylinder surface and resolves the fine scales present in the forming vortex. At the same time, it removes unnecessary elements away from the body where the solution is smooth, and maintains far-field boundaries at the same distance.

The second example is a triangulation of a complicated domain as shown in Fig. 1.6. This domain uses a variety of triangular elements of different aspect ratios and orientations in an almost random triangulation. Within this domain, we have solved an elliptic Helmholtz problem (see Chapter 5) of the form

$$\nabla^2 u(x_1, x_2) - u(x_1, x_2) = f(x_1, x_2)\,.$$

The exact solution considered was

$$u(x_1, x_2) = \sin\left[\frac{\pi}{4}(\sqrt{(x_1 - 15)^2 + (x_2 - 8)^2})\right].$$

Also shown in Fig. 1.6 is the $H^1$ error plotted with respect to the polynomial order of the expansion. Exponential convergence is observed, as indicated by the

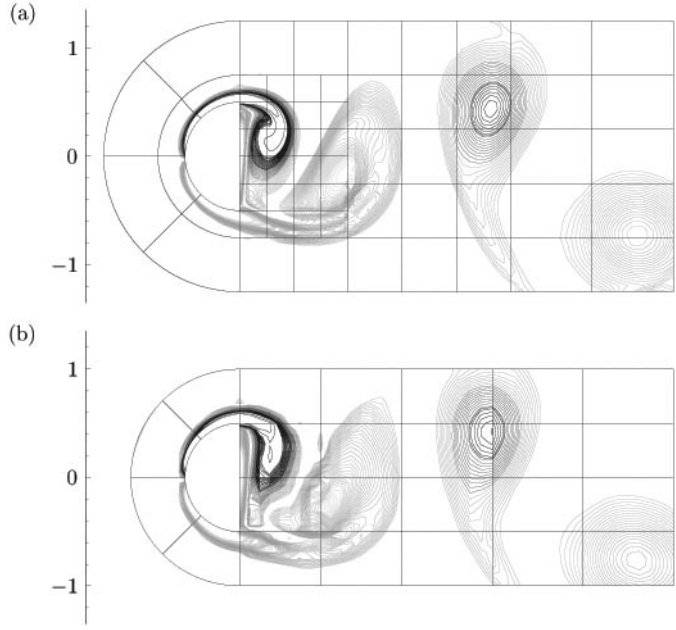



FIG. 1.5. Vortex shedding simulation using (a) non-conforming and (b) conforming discretisation. Shown are instantaneous vorticity contours in the near-wake.

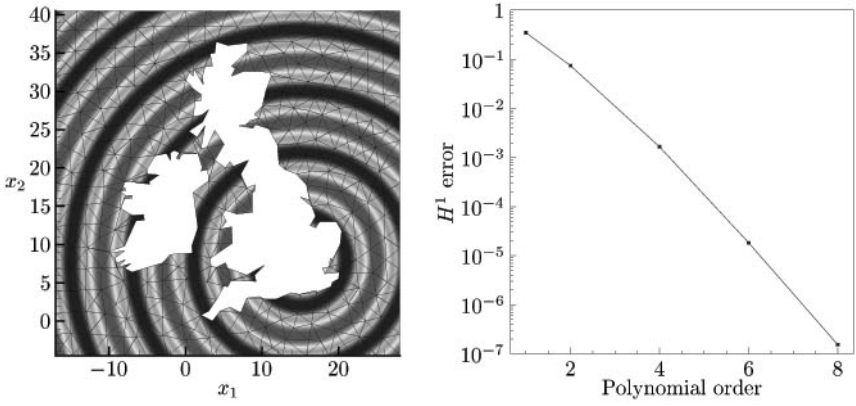



FIG. 1.6. Convergence to the Helmholtz problem with solution $u(x_1, x_2) = \sin(\pi(R(x_1, x_2) - R_0))$. Exponential convergence is obtained independently of the complexity of the discretisation. The exact solution is prescribed at all domain boundaries.

asymptotic linear behaviour of the curves on this linear–log plot. We note that the solution domain does not include the region immediately around $x_1 = 15$, $x_2 = 8$, since within this region it is not possible to bound all the derivatives of $u(x_1, x_2)$.

### 1.2.5 *What is hp convergence?*

The mathematical theory of finite elements in the 1970s has established rigorously the convergence of the $h$-version of the finite element. The error in the numerical solution decays algebraically by refining the mesh, that is, introducing more elements while keeping the (low) order of the interpolating polynomial fixed. An alternative approach is to keep the number of subdomains fixed and increase the order of the interpolating polynomials in order to reduce the error in the numerical solution. This is called $p$-type refinement and is typical of polynomial spectral methods [199]. For infinitely smooth solutions $p$-refinement usually leads to an exponential decay of the numerical error. Recognising the advantages of both types of convergence in mechanics problems, B. A. Szabó proposed and implemented a new method that he coined the *hp* version of the finite element. In this combined approach we simultaneously increase the number of subdomains (elements) and increase the interpolation order within the element either uniformly throughout the domain or selectively depending on the resolution requirements.

To give an example of an *hp* refinement we revisit the problems shown in Fig. 1.6. In the $h$-refinement strategy we refine the mesh as shown in Fig. 1.7(a)–(d), whereas in the $p$-refinement strategy we can fix the mesh and increase the

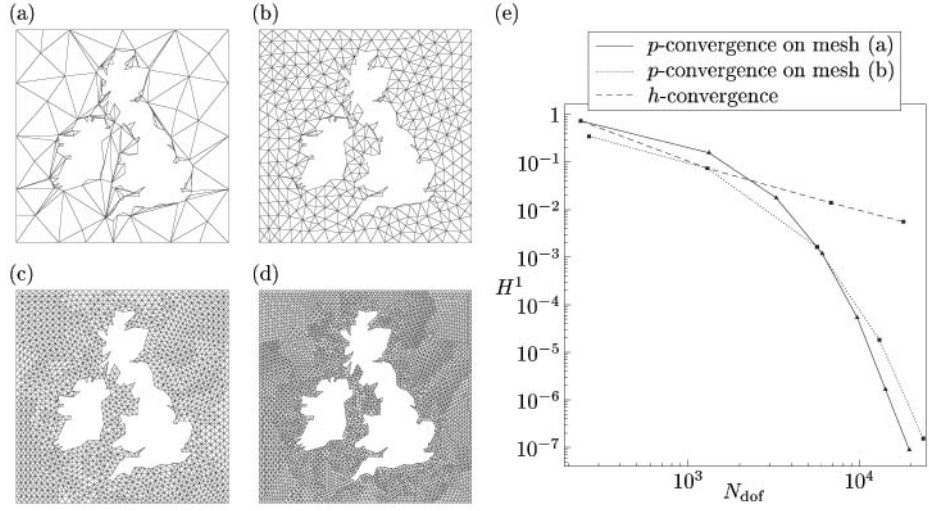



FIG. 1.7. Convergence history using $h$- and $p$-type refinement for the elliptic problem described in Fig. 1.6.

order of the polynomial expansion. The error in the $H^1$ norm as a function of the total degrees of freedom is shown in Fig. 1.7(e) for both the $h$-refinement with a fixed polynomial order of $P = 2$ and a $p$-refinement based on meshes (a) and (b). The $h$-refinement initially resolves the solution faster than the $p$-refinement on mesh (a); however, as the asymptotic exponential convergence is achieved the $p$-refinement takes over the $h$-refinement process. If we just consider the meshes shown in Fig. 1.7(a)–(d), the optimum convergence path as a function of degrees of freedom involves using both mesh (b) and mesh (a), thereby using both $h$- and $p$-refinement. In general, we would like to know the error as a function of computational cost, which is much harder to measure. However, for smooth solutions the concept of $hp$ refinement still provides the optimal convergence strategy.

# 2

# FUNDAMENTAL CONCEPTS IN ONE DIMENSION

In this chapter we illustrate the fundamental concepts behind the design and implementation of the spectral/*hp* element method for one-dimensional linear elliptic problems. The basic mechanics of this formulation will help to illustrate useful techniques for a variety of different types of mathematical problems, such as hyperbolic and parabolic equations, as well as different types of formulations such as the discontinuous Galerkin formulation discussed in Section 6.2.2. It will also provide the basis for understanding the multi-dimensional formulation which is discussed in Chapters 3 and 4.

The chapter starts by discussing the general framework of different formulations in the context of the method of weighted residuals in Section 2.1. This is followed by a more detailed description of the Galerkin method in Section 2.2. The efficiency of the spectral/*hp* element technique can be attributed to the elemental decomposition and implementation at this level. These ideas are introduced in Section 2.3 where we discuss the *h*-type elemental decomposition from a global expansion and then the *p*-type polynomial expansion within each elemental region. In Section 2.4 we then detail the principal elemental operations of numerical integration and differentiation. Finally, in Section 2.5 we outline some of the theory and results relating to error estimation of the technique, and in Section 2.6 we provide some exercises focused towards writing a one-dimensional spectral/*hp* element solver with examples.

♦[1] For the reader primarily interested in the implementation of the technique we have introduced margin identifiers to indicate key formulation ♣ and implementation ♦ details. The implementation details start with the descriptive formulation of the Galerkin problem in Section 2.2.1. This section also discusses the Galerkin implementation of Neumann boundary conditions through the weak form of the problem in Section 2.2.1.2 and the enforcement of Dirichlet boundary conditions through homogenisation of the solution in Section 2.2.1.3. Readers familiar with the finite element technique may already be aware of these techniques, although these sections also provide the authors' perspective on these issues. The next implementation-related topic is found in Section 2.3.1, which discusses the elemental (*h*-type) decomposition of the spectral/*hp* element approach. Section 2.3.1.4 also discusses how to transform from the elemental decomposition to the global problem, and Sections 2.3.3.3 and 2.3.4.2 present the most commonly used *p*-type polynomial expansion bases in one dimension. To

[1] Layout of chapter from an implementation point of view.

complete the implementation of the method requires knowledge of numerical integration and differentiation, which are discussed in Sections 2.4.1 and 2.4.2. Finally, in Section 2.6 we provide a structured series of exercises directed towards the implementation of the one-dimensional solver.

For a more detailed description, Section 2.2.3 discusses the Galerkin formulation in a more mathematical framework and Section 2.2.4 highlights the classical properties associated with the Galerkin formulation. After introducing the elemental decomposition in Section 2.3.1, we provide a more general discussion on the design and construction of $p$-type polynomial expansions in Section 2.3.2.1. To complement the discussion on numerical quadrature in Section 2.4.1 we also expand on this topic in Section 2.4.1.2 by discussing the effect of under-integration of nonlinear products of the polynomial solution, which is important when considering the nonlinear advection terms of the Navier–Stokes equations. Finally, in Section 2.5 we outline the basic formulation and error estimation results associated with one-dimensional spectral/$hp$ element methods.

Although our principal focus will be on developing the Galerkin formulation for the spectral/$hp$ element approach using higher-order polynomial expansions, the governing theory is taken from the traditional finite element technique which has been well documented in many texts, see, for example, [75, 247, 443, 511].

*Historical setting of the finite element method*

Structural engineers were responsible for the original implementation of the finite element method. It took approximately a decade before the method was recognised as a form of the Rayleigh–Ritz problem. The relation between these two techniques comes from considering the variational form of the problem [113]. For example, the quadratic functional

$$\mathcal{F}(u) = \int_0^1 \left[p(x)(u'(x))^2 + q(x)(u(x))^2 - 2f(x)u(x)\right] \mathrm{d}x \tag{2.0.1}$$

has a minimum with respect to a variation in $u(x)$ given by the Euler equation

$$-\frac{\mathrm{d}}{\mathrm{d}x}\left(p(x)\frac{\mathrm{d}u(x)}{\mathrm{d}x}\right) + q(x)u(x) = f(x)\,. \tag{2.0.2}$$

Therefore, instead of solving for the differential equation (2.0.2) to determine $u(x)$, an alternative but equivalent solution is to find the value of $u(x)$ which minimises the functional of eqn (2.0.1).

The Rayleigh–Ritz idea approximates the solution by a finite number of functions $u(x) = \sum_i^N q_i \Phi_i(x)$ to determine the unknown weights $q_i$, which minimise the functional of eqn (2.0.1). In the finite element method the solution is also approximated by a finite number of functions, which are typically local in nature as opposed to the global functions used in the Rayleigh–Ritz approach. However, the starting-point for a finite element method is the differential equation (2.0.2), which is formulated into an integral form (also known as the Galerkin

formulation) so that the problem can be reduced to an algebraic system which can be solved numerically. The connection between the two methods was made when it was realised that the integral form of the finite element method was exactly the same as the functional form used in the Rayleigh–Ritz method for a linear problem. In structural mechanics it is also possible to form a functional directly from a statement of equilibrium without ever having to determine the Euler equation.

This relation between the finite element method and the Rayleigh–Ritz technique was very significant since it made the finite element technique mathematically respectable. It also ultimately proved to be somewhat misleading as it implied that a functional form was needed to formulate the problem. This is, in fact, not the case, as a more general formulation is possible using the method of weighted residuals which leads to the standard Galerkin formulation.

## 2.1 Method of weighted residuals

In approximating an exact solution numerically we are typically replacing an *infinite* expansion with a *finite* representation. Such approximation necessarily means that the differential equation cannot be satisfied everywhere in our region of interest and so we are only able to satisfy a finite number of *conditions*. It is the choice of the *conditions* which are to be satisfied that defines the type of numerical method or projection operator of the scheme. For example, the collocation method refers to a method where the differential equation is satisfied at a few distinct positions rather than at every point in the solution region.

The method of weighted residuals illustrates how the choice of different weight (or test) functions in an integral or weak form of the equation can be used to construct many of the common numerical methods.

To describe the method of weighted residuals we consider a *linear* differential equation in a domain $\Omega$ denoted by

$$\mathbb{L}(u) = 0\,, \tag{2.1.1}$$

subject to appropriate initial and boundary conditions. It is assumed that the solution $u(\boldsymbol{x}, t)$ can be accurately represented by the approximate solution of the form

$$u^{\delta}(\boldsymbol{x}, t) = u_0(\boldsymbol{x}, t) + \sum_{i=1}^{N_{\text{dof}}} \hat{u}_i(t)\Phi_i(\boldsymbol{x})\,, \tag{2.1.2}$$

where $\Phi_i(\boldsymbol{x})$ are analytic functions called the *trial* (or *expansion*) *functions*, $\hat{u}_i(t)$ are the $N_{\text{dof}}$ unknown coefficients, and $u_0(\boldsymbol{x}, t)$ is selected to satisfy the initial and boundary conditions. We note that, by definition, $\Phi_i(\boldsymbol{x})$ satisfies homogeneous boundary conditions (i.e., zero on Dirichlet boundaries) since the known function $u_0(\boldsymbol{x}, t)$ already satisfies the boundary conditions of the problem. Substitution of the approximation (2.1.2) into eqn (2.1.1) produces a nonzero residual, $R$, such that

$$\mathbb{L}(u^\delta) = R(u^\delta)\,. \tag{2.1.3}$$

The approximation has the form given by eqn (2.1.2) but we have no unique way of determining the coefficients $\hat{u}_i(t)$. To do so, we can place a restriction on the residual $R$ which in turn will reduce eqn (2.1.3) to a system of ordinary differential equations in $\hat{u}_i(t)$. If the original eqn (2.1.1) is independent of time then the coefficients $\hat{u}_i$ can be determined directly from the solution of a system of algebraic equations.

To define the type of restriction to be placed on the residual $R$ we must first introduce the Legendre inner product $(f, g)$ over the domain $\Omega$ defined as

$$(f, g) = \int_\Omega f(\boldsymbol{x}) g(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x}\,. \tag{2.1.4}$$

The restriction placed on $R$ is that the inner product of the residual with respect to a *weight* (or *test*) function is equal to zero, that is,

$$(v_j(\boldsymbol{x}), R) = 0\,, \quad j = 1, \ldots, N_{\mathrm{dof}}\,,$$

where the function $v_j(\boldsymbol{x})$ is the test or weight function. The weighted residual is then said to be zero and it is from this expression that the technique takes its name.

Upon convergence as $N_{\mathrm{dof}} \to \infty$, the residual $R(\boldsymbol{x})$ tends to zero since the approximate solution $u^\delta(\boldsymbol{x}, t)$ approaches the exact solution $u(\boldsymbol{x}, t)$. However, the nature of the scheme is determined by the choice of the expansion function $\Phi_i(\boldsymbol{x})$ and the test function $v_j$. A list of the most commonly used test functions and the computational method they produce is shown in Table 2.1 and will be briefly outlined in the following sections.

*Collocation method*

In the collocation method the test function is the Dirac delta function such that $v_j(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_j)$, where $\boldsymbol{x}_j$ denotes a set of given collocation points. At a collocation point the residual is set to zero ($R(\boldsymbol{x}_j) = 0$) and, accordingly, the differential equation is exactly satisfied at this point.

TABLE 2.1. Test functions $v_j(\boldsymbol{x})$ used in the method of weighted residuals and the method produced.

| Test/weight function | Type of method |
|---|---|
| $v_j(\boldsymbol{x}) = \delta(\boldsymbol{x} - \boldsymbol{x}_j)$ | Collocation |
| $v_j(\boldsymbol{x}) = \begin{cases} 1\,, & \text{inside } \Omega^j\,, \\ 0\,, & \text{outside } \Omega^j \end{cases}$ | Finite volume (subdomain) |
| $v_j(\boldsymbol{x}) = \dfrac{\partial R}{\partial \hat{u}_j}$ | Least-squares |
| $v_j(\boldsymbol{x}) = \Phi_j$ | Galerkin |
| $v_j(\boldsymbol{x}) = \Psi_i\ (\neq \Phi_j)$ | Petrov–Galerkin |

*Finite volume/subdomain methods*

The finite volume or subdomain method is described by splitting the solution domain $\Omega$ into $N_{\text{dof}}$ non-overlapping subdomains $\Omega^j$ and using a test function of the form

$$v_j = \begin{cases} 1\,, & \text{inside } \Omega^j\,, \\ 0\,, & \text{outside } \Omega^j\,, \end{cases}$$

where the union of $\Omega^j$ is equal to $\Omega$ (i.e., $\bigcup_{j=1}^{N_{\text{el}}} \Omega_j = \Omega$). This method has been very popular in computational aerodynamics. It can be considered as a technique to recover a conservation statement from a partial differential equation.

*Least-squares method*

The least-squares method originates from the idea of least-squares estimation developed by Gauss. In this method the residual is set to $v_j = \partial R/\partial \hat{u}_j$. This choice determines the coefficients $\hat{u}_i$ which minimises $(R, R)$. This formulation using a spectral/*hp* element discretisation has recently increased in popularity and is discussed further in Section 8.5.

*Galerkin method*

Finally, we consider the Galerkin method (also known as the Bubnov–Galerkin method). In this method the test functions are chosen to be the same as the trial or expansion functions such that $v_j = \Phi_j$. A broader class of the Galerkin method known as the Petrov–Galerkin method, or sometimes the generalised Galerkin method, uses test functions that may be similar, but not identical, to the trial functions ($v_j \neq \Phi_j$). The choice of Petrov–Galerkin test functions is typically based upon a perturbation of the trial functions, where the additional contribution is chosen to improve the numerical stability of the scheme or to impose an upwind condition [222,248]. For further details on the background of the methods of weighted residuals, please see Finlayson [160] and Fletcher [163].

This book is primarily concerned with Galerkin methods and for the majority of cases we shall be considering the standard Bubnov–Galerkin method. In Chapters 6 and 10 we will also discuss the *discontinuous* Galerkin methods in the context of solving hyperbolic conservation laws, and, in Chapter 7, in the context of second-order elliptic equations.

The method of weighted residuals illustrates how to construct different types of numerical techniques and defines the projection operator being employed in each method. It does not define the type of expansion function or approximation space, although the use of the terminology *spectral* or *finite element* does provide further insight. It is generally understood that spectral methods use a set of *global* expansion functions, that is, the expansion functions $\Phi_i(\boldsymbol{x})$ has a nonzero definition throughout the solution domain (i.e., like a sine or cosine function). The finite element or, alternatively, the finite volume technique uses a set of expansion functions $\Phi_i(\boldsymbol{x})$ which are only defined in a local 'finite' region. In the

finite element expansion these regions are typically made up of non-overlapping tessellations of the total solution domain. Theoretically, both the spectral and finite-element-type expansions may be used with any of the numerical methodologies described above. The projection operators can also be mixed. This is commonly the case when considering nonlinear problems in spectral methods where the collocation projection is used to evaluate nonlinear products in the so-called *pseudo-spectral* method [199].

## 2.2 Galerkin formulation

Finite element methods typically use the Galerkin formulation introduced in the previous section. In this section, we describe how to formulate the Galerkin problem. We start in Section 2.2.1 by considering an informal formulation in order to solve the one-dimensional Poisson equation to introduce the basic concepts. The formulation is then illustrated by a worked example using linear finite elements in Section 2.2.2. A mathematical description of the formulation is presented in Section 2.2.3 and some important mathematical properties of the Galerkin formulation are discussed in Section 2.2.4.

### 2.2.1 *Descriptive formulation*

Our example problem is the Poisson equation ♣1

$$\mathbb{L}(u) \equiv \nabla^2 u + f = 0\,. \tag{2.2.1}$$

This equation arises in many areas of physics such as irrotational fluid flow and steady-state heat conduction, as well as in problems involving electrical and gravitational potentials. In one dimension, eqn (2.2.1) becomes

$$\mathbb{L}(u) \equiv \frac{\partial^2 u}{\partial x^2} + f = 0\,. \tag{2.2.2}$$

#### 2.2.1.1 *Strong form and definition of boundary conditions*

For this problem to be well posed and thus have a unique solution we need to specify boundary conditions. If we consider the solution in a domain $\Omega = \{x \mid 0 \leqslant x \leqslant 1\}$, then we might consider the following boundary conditions:

$$u(0) = g_{\mathcal{D}}\,, \quad \frac{\partial u}{\partial x}(1) = g_{\mathcal{N}}\,,$$

where $g_{\mathcal{D}}$ and $g_{\mathcal{N}}$ are given constants.

The boundary condition $u(0) = g_{\mathcal{D}}$ specifies a condition on the solution and is referred to as a *Dirichlet* or *essential* boundary condition. The boundary condition $\partial u(1)/\partial x = g_{\mathcal{N}}$, however, specifies a condition on the derivative of the solution and is referred to as a *Neumann* or *natural* boundary condition.

[1] Galerkin problem statement and implementation of boundary conditions.

As we shall see, in the Galerkin formulation Dirichlet boundary conditions have to be specified explicitly whereas Neumann conditions are dealt with implicitly as part of the formulation. If the boundary conditions stated above are applied to eqn (2.2.2) it becomes a two-point boundary value problem and is said to be in the *strong* or *classical* form.

2.2.1.2 *Weak form and implementation of Neumann boundary conditions*

♣2 To construct a weak approximation to eqn (2.2.2), we multiply this equation by a weight of test function $v(x)$, which by definition is *zero on all Dirichlet boundaries* $\partial\Omega_D$, and integrate over the domain $\Omega$ to obtain the inner product of $\mathbb{L}(u)$ with respect to $v$:

$$(v, \mathbb{L}(u)) = \int_0^1 v \left( \frac{\partial^2 u}{\partial x^2} + f \right) \mathrm{d}x = 0 . \tag{2.2.3}$$

We note that eqn (2.2.3) is equivalent to setting the weighted residual to zero. If $u^\delta$ is an approximation to $u$ (recalling that $\mathbb{L}(u^\delta) = R(u^\delta)$) then eqn (2.2.3) is equivalent to the condition $(v, R) = 0$.

The next important step in the classical Galerkin spectral/*hp* element formulation is to integrate eqn (2.2.3) by parts to obtain

$$\int_0^1 \frac{\partial v}{\partial x} \frac{\partial u}{\partial x} \mathrm{d}x = \int_0^1 v f \, \mathrm{d}x + \left[ v \frac{\partial u}{\partial x} \right]_0^1 . \tag{2.2.4}$$

In higher dimensions we would have used Gauss' divergence theorem to achieve an analogous result. As the test functions are defined to be zero on Dirichlet boundaries we know that $v(0) = 0$. Therefore, we can enforce the Neumann boundary condition $\partial u(1)/\partial x = g_\mathcal{N}$ by substitution into the last term of eqn (2.2.4), which simplifies to

$$\int_0^1 \frac{\partial v}{\partial x} \frac{\partial u}{\partial x} \mathrm{d}x = \int_0^1 v f \, \mathrm{d}x + v(1) g_\mathcal{N} . \tag{2.2.5}$$

In this last step we see how the Neumann boundary conditions are naturally included in the formulation through the action of the integration of parts. Note that for zero Neumann condition the last term vanishes; then, to impose the zero Neumann condition we do nothing! This operation not only reduces the order of the maximum derivative of the discrete problem but, as we shall see in Section 2.2.2, it also makes the resulting discrete matrix equation symmetric. The integral form of the problem given by eqns (2.2.4) and (2.2.5) is referred to as the *weak* form of the problem.

The Galerkin approximation of problem (2.2.2) is the solution to the weak form of the eqn (2.2.5) when the exact solution $u(x)$ is approximated by a finite

[2] Treatment of second-order differential operators and how to impose Neumann boundary conditions.

expansion denoted by $u^\delta(x)$. The function $v(x)$ in eqn (2.2.5) is also replaced by a finite expansion, denoted by $v^\delta(x)$, and so eqn (2.2.5) becomes

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}\,\mathrm{d}x = \int_0^1 v^\delta f\,\mathrm{d}x + v^\delta(1)g_{\mathcal{N}}\,. \tag{2.2.6}$$

We recall that the set of functions used in the finite expansion of the solution $u^\delta$ are referred to as the *trial* functions, whereas the functions contained within $v^\delta$ are referred to as the *test* functions.

2.2.1.3 *Enforcing Dirichlet boundary conditions: lifting a known solution*

By definition, all Dirichlet boundary conditions are known. We can extend (or lift) these known functions on the boundary into the interior of the solution domain by any convenient function, which is contained within the solution space. The word 'lift' originates from the French 'relevement', as introduced by Lions [306]. The action of lifting a known solution is equivalent to decomposing the approximate solution $u^\delta$ into a known lifted function, $u^{\mathcal{D}}$, which satisfies the Dirichlet boundary conditions, and an unknown homogeneous function, $u^{\mathcal{H}}$, which is zero on the Dirichlet boundaries, i.e., ♣3

$$u^\delta = u^{\mathcal{H}} + u^{\mathcal{D}}\,, \tag{2.2.7}$$

where

$$u^{\mathcal{H}}(\partial\Omega_{\mathcal{D}}) = 0\,, \quad u^{\mathcal{D}}(\partial\Omega_{\mathcal{D}}) = g_{\mathcal{D}}\,.$$

By substituting eqn (2.2.7) into our weak formulation of the problem given by eqn (2.2.6), we obtain

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\left[\frac{\partial u^{\mathcal{D}}}{\partial x} + \frac{\partial u^{\mathcal{H}}}{\partial x}\right]\mathrm{d}x = \int_0^1 v^\delta f\,\mathrm{d}x + v^\delta(1)g_{\mathcal{N}}\,,$$

which can be rearranged to obtain

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^{\mathcal{H}}}{\partial x}\,\mathrm{d}x = \int_0^1 v^\delta f\,\mathrm{d}x + v^\delta(1)g_{\mathcal{N}} - \int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^{\mathcal{D}}}{\partial x}\,\mathrm{d}x\,. \tag{2.2.8}$$

Since $u^{\mathcal{D}}$ is a known function which satisfies the boundary conditions, all terms on the right-hand side of eqn (2.2.8) are known and this equation has been lifted in the sense that the unknown function $u$ satisfies homogeneous (i.e., zero) Dirichlet boundary conditions.

As will be illustrated in Section 2.2.2, eqn (2.2.8) can be solved as a finite linear algebraic system as all the terms on the right-hand side are known, and the homogeneous solution $u^{\mathcal{H}}$ and the test function $v^\delta$ contain a finite number of functions. The Galerkin formulation has, therefore, reduced the differential

[3] Lifting a known solution to impose Dirichlet boundary conditions.

problem (2.2.2) to an algebraic matrix problem, which we can solve on a computer.

The process of lifting a known solution is an important part of the Galerkin spectral/$hp$ formulation since we require that the same set of basis functions that are used to represent the test functions, $v^\delta$, are also used in the representation of the solution $u^\delta$. From Section 2.2.1.2 we recall that the expansion functions used for the test functions $v^\delta$ are defined to be zero on all Dirichlet boundaries and after the lifting step we can define the homogeneous solution vector $u^{\mathcal{H}}$ with the same expansion basis since this function also has zero boundary conditions on Dirichlet boundaries. This, therefore, permits us to use the same expansion space for $u^{\mathcal{H}}$ as we use for $v^\delta$.

Although this decomposition may appear unnecessarily complicated, this step plays an important role in the Galerkin formulation. Without lifting a known solution out of our problem we will have more degrees of freedom in the test space than we have in the trial space. In implementation terms, this means that the algebraic system which results from evaluating the weak problem (2.2.8) will not be square.

An alternative approach to enforcing Dirichlet boundary conditions, commonly used in finite element methods, is to assemble a matrix system including all degrees of freedom in our approximation for both the test and trial functions. Dirichlet boundary conditions can then be enforced by zeroing rows which correspond to the known degrees of freedom, placing a unit term on the diagonal, and setting the right-hand side to the known value. The resulting matrix system will not be symmetric even if the original problem was symmetric. The matrix system which arises from the lifting approach is a submatrix of the full problem. The second right-hand-side term in eqn (2.2.8) can also be understood as another submatrix of the full problem multiplied by the known boundary conditions. In the lifted solution approach, the resulting matrix problem remains symmetric if the original problem was symmetric. The drawback of the lifting approach is that it requires a numbering system to reorder the matrix which is not required in the row-zeroing approach. We note, however, that the lifting approach also permits any known function satisfying the Dirichlet boundary conditions to be applied, which can be convenient when treating iterative solutions. Further, using the lifting approach, there is no need to assemble unnecessary components of the matrix problem, which can be quite costly in multiple dimensions. There is, however, a more complicated right-hand side to evaluate in eqn (2.2.8).

Finally, we note that to construct a matrix problem necessarily implies a linear problem. Quite often when treating nonlinear problems the nonlinear terms are treated explicitly in time or are linearised so that only linear terms are handled implicitly in time. As a result the above technique can still be applied.

#### 2.2.1.4 *Mixed or Robin boundary conditions*

Another type of boundary condition is a *mixed* or *Robin* boundary condition. This commonly arises in convective heat transfer problems and is a linear combination of a Dirichlet and Neumann condition of the form ♣4

$$\alpha \frac{\partial u(1)}{\partial x} + \beta u(1) = g_{\mathcal{R}} \quad (\text{with } \alpha \neq 0)\,,$$

where $\alpha$, $\beta$, and $g_{\mathcal{R}}$ are known. To impose this condition we can substitute

$$\frac{\partial u(1)}{\partial x} = \frac{1}{\alpha}(g_{\mathcal{R}} - \beta u(1))$$

into eqn (2.2.6) and thus obtain

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}\,\mathrm{d}x + \frac{\beta}{\alpha} v^\delta(1) u^\delta(1) = \int_0^1 v^\delta f\,\mathrm{d}x + \frac{v^\delta(1) g_{\mathcal{R}}}{\alpha}\,.$$

We have placed the term $\beta v^\delta(1)u^\delta(1)/\alpha$ on the left-hand side as the value of $u^\delta(1)$ has to be implicitly solved as part of the algebraic system. The practical difference between the implementation of the Robin and Neumann conditions is simply the modification of the matrix arising from the term $(\beta/\alpha)v^\delta(1)u^\delta(1)$.

### 2.2.2 *Two-domain linear finite element example*

Having defined the weak form of the problem and explained how to impose boundary conditions in Sections 2.2.1.2 to 2.2.1.4 we continue our overview of the Galerkin formulation with a worked example of a two-subdomain linear finite element solution. To illustrate the mechanics of the formulation we will use a globally-defined expansion basis. It should be noted, however, that in a practical implementation of a problem involving many elemental domains the normal practice is to use an elemental construction description of the basis, as discussed in Section 2.3. ♦2

Once again, we consider the one-dimensional Poisson equation in the interval $0 < x \leqslant 1$:

$$\mathbb{L}(u) \equiv \frac{\partial^2 u}{\partial x^2} + f = 0\,,$$

where $f(x)$ is a known function and the boundary conditions are

$$u(0) = g_{\mathcal{D}} = 1\,, \quad \frac{\partial u}{\partial x}(1) = g_{\mathcal{N}} = 1\,.$$

Following the formulation introduced in Sections 2.2.1.2 and 2.2.1.3, we construct our weak Galerkin approximation in two steps.

[4] Imposing mixed or Robin boundary conditions.

[2] Worked example of the Galerkin solution to an elliptic problem.

1. We start by considering the *weak form* by multiplying the problem by a discrete test space $v^\delta$ and integrating the second-order derivative by parts, which allows us to impose the Neumann boundary condition to arrive at

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}\,\mathrm{d}x = \int_0^1 v^\delta f\,\mathrm{d}x + v^\delta(1)g_{\mathcal{N}}\,,$$

   where $u^\delta$ is our discrete solution.
2. We now *lift* a known solution from the problem by decomposing $u^\delta$ into a known solution satisfying the Dirichlet boundary conditions $u^{\mathcal{D}}$ and a homogeneous solution $u^{\mathcal{H}}$ such that $u^\delta = u^{\mathcal{D}} + u^{\mathcal{H}}$, and so our weak solution becomes

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^{\mathcal{H}}}{\partial x}\,\mathrm{d}x = \int_0^1 v^\delta f\,\mathrm{d}x + v^\delta(1)g_{\mathcal{N}} - \int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^{\mathcal{D}}}{\partial x}\,\mathrm{d}x\,. \tag{2.2.9}$$

In our problem the solution is to be approximated by piecewise linear functions over two subdomains $\Omega^1$ and $\Omega^2$, as shown in Fig. 2.1. This type of approximation is known as an $h$-type approximation, where the $h$ parameter represents the characteristic size of a subdomain (in one dimension, its length). Convergence to the exact solution is achieved by subdividing the solution domain $\Omega$ into smaller and smaller subdomains, so that $h \to 0$. We note, however, that

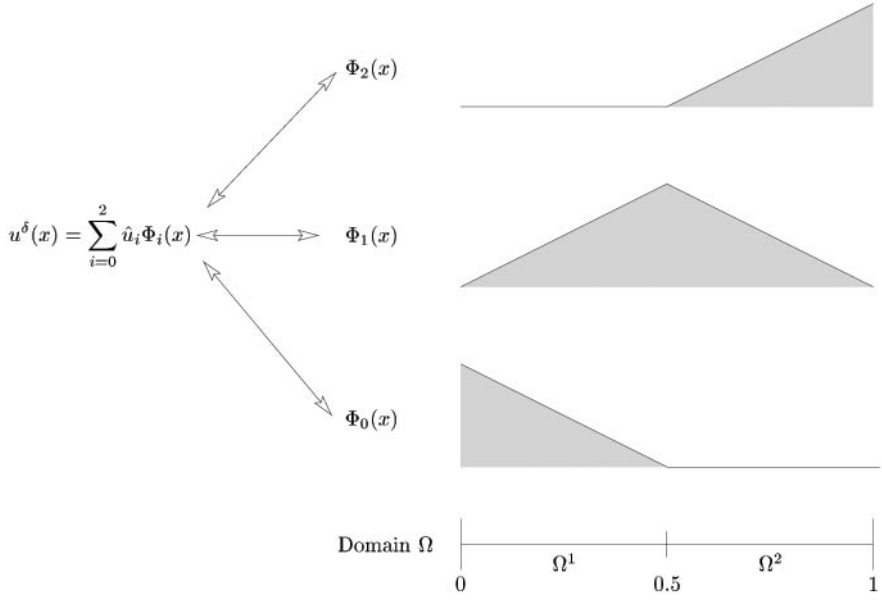



FIG. 2.1. Linear finite element approximation $u^\delta(x) = \sum_{i=0}^{2} \hat{u}_i \Phi_i(x)$, in a domain $\Omega$, using two elemental subdomains, $\Omega^1$ and $\Omega^2$.

the elemental decomposition is still a vital component in the spectral/*hp* element decomposition. The main difference is that we use higher than linear order polynomials in each element. This is known as the *p*-type approximation.

For our linear two-subdomain case the approximate expansion has the form

$$u^\delta = \sum_{i=0}^{2} \hat{u}_i \Phi_i(x)\,,$$

where $\Phi_i(x)$ is defined as

$$\Phi_0(x) = \begin{cases} 1-2x\,, & 0 \leqslant x \leqslant \frac{1}{2}\,, \\ 0\,, & \frac{1}{2} \leqslant x \leqslant 1\,, \end{cases} \qquad \Phi_1(x) = \begin{cases} 2x\,, & 0 \leqslant x \leqslant \frac{1}{2}\,, \\ 2(1-x)\,, & \frac{1}{2} \leqslant x \leqslant 1\,, \end{cases}$$

$$\Phi_2(x) = \begin{cases} 0\,, & 0 \leqslant x \leqslant \frac{1}{2}\,, \\ 2x-1\,, & \frac{1}{2} \leqslant x \leqslant 1\,. \end{cases}$$

The only way to satisfy the Dirichlet boundary condition at $x = 0$ is to set $\hat{u}_0 = g_D$ because $\phi_1(x)$ and $\phi_2(x)$ are zero at $x - 0$. Therefore, one choice for a lifted solution is to use the decomposition $u^\delta = u^{\mathcal{H}} + u^{\mathcal{D}}$, such that

$$u^{\mathcal{H}} = \hat{u}_1 \Phi_1(x) + \hat{u}_2 \Phi_2(x)\,,$$
$$u^{\mathcal{D}} = g_D \Phi_0(x)\,,$$

where $\hat{u}_1$ and $\hat{u}_2$ are still to be determined. Note that we could also have chosen $u^{\mathcal{D}}$ to be a known function of $\Phi_1(x)$ and $\Phi_2(x)$ if, for example, we had the solution to a previous problem. The expansion set used to define $u^{\mathcal{H}}$ contains the same functions used as homogeneous test functions. We can therefore define the test functions as

$$v^\delta(x) = \hat{v}_1 \Phi_1(x) + \hat{v}_2 \Phi_2(x)\,,$$

where $\hat{v}_1$ and $\hat{v}_2$ are also unknown. As we shall see, these will never need to be determined. Finally, we need a representation of the function $f(x)$. This function is known explicitly and therefore it is theoretically possible to evaluate exactly any operations involving $f(x)$ with other known functions such as $\Phi_i(x)$. However, in practice, in order to treat an arbitrary function within an *efficient* computational implementation, the function is usually represented using the same expansion as applied to $u^\delta$, i.e.,

$$f(x) = \sum_{i=0}^{2} \hat{f}_i \Phi_i(x) = \hat{f}_0 \Phi_0(x) + \hat{f}_1 \Phi_1(x) + \hat{f}_2 \Phi_2(x)\,.$$

Clearly, if $f(x)$ is a constant or a linear function then it will be exactly represented by this expression. For more complex functions the coefficients $\hat{f}_0$, $\hat{f}_1$, and $\hat{f}_2$ need to be determined and can be chosen to satisfy an interpolation approximation where $\hat{f}_0 = f(0)$, $\hat{f}_1 = f(0.5)$, and $\hat{f}_2 = f(1)$.

Evaluating the terms in eqn (2.2.9), we find

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^{\mathcal{H}}}{\partial x}\,\mathrm{d}x = \int_0^{1/2} (2\hat{v}_1)(2\hat{u}_1)\,\mathrm{d}x + \int_{1/2}^1 (-2\hat{v}_1 + 2\hat{v}_2)(-2\hat{u}_1 + 2\hat{u}_2)\,\mathrm{d}x$$

$$= \begin{bmatrix} \hat{v}_1 & \hat{v}_2 \end{bmatrix} \begin{bmatrix} 4 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \end{bmatrix}, \tag{2.2.10a}$$

$$\int_0^1 v^\delta f\,\mathrm{d}x = \int_0^{1/2} (\hat{v}_1 2x)\left[\hat{f}_0(1-2x) + \hat{f}_1(2x)\right]\mathrm{d}x$$

$$+ \int_{1/2}^1 [\hat{v}_1 2(1-x) + \hat{v}_2(2x-1)]\left[\hat{f}_1 2(1-x) + \hat{f}_2(2x-1)\right]\mathrm{d}x$$

$$= \begin{bmatrix} \hat{v}_1 & \hat{v}_2 \end{bmatrix} \begin{bmatrix} \frac{1}{12}\hat{f}_0 + \frac{1}{3}\hat{f}_1 + \frac{1}{12}\hat{f}_2 \\ \frac{1}{12}\hat{f}_1 + \frac{1}{6}\hat{f}_2 \end{bmatrix}, \tag{2.2.10b}$$

$$v^\delta(1) g_{\mathcal{N}} = [\hat{v}_1\Phi_1(1) + \hat{v}_2\Phi_2(1)]\, g_{\mathcal{N}} = \begin{bmatrix} \hat{v}_1 & \hat{v}_2 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} g_{\mathcal{N}}, \tag{2.2.10c}$$

$$\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^{\mathcal{D}}}{\partial x}\,\mathrm{d}x = \int_0^{1/2} (2\hat{v}_1)(-2g_{\mathcal{D}})\,\mathrm{d}x = \begin{bmatrix} \hat{v}_1 & \hat{v}_2 \end{bmatrix} \begin{bmatrix} -2g_{\mathcal{D}} \\ 0 \end{bmatrix}. \tag{2.2.10d}$$

Therefore, eqn (2.2.9) can be written as the discrete matrix problem

$$\begin{bmatrix} \hat{v}_1 & \hat{v}_2 \end{bmatrix} \left\{ \begin{bmatrix} 4 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \end{bmatrix} - \begin{bmatrix} \frac{1}{12}\hat{f}_0 + \frac{1}{3}\hat{f}_1 + \frac{1}{12}\hat{f}_2 \\ \frac{1}{12}\hat{f}_1 + \frac{1}{6}\hat{f}_2 \end{bmatrix} - \begin{bmatrix} 0 \\ g_{\mathcal{N}} \end{bmatrix} + \begin{bmatrix} -2g_{\mathcal{D}} \\ 0 \end{bmatrix} \right\} = 0.$$

For arbitrary choices of $\hat{v}_1$ and $\hat{v}_2$ we can solve this equation by evaluating the matrix equation in the curly brackets. Recalling that $g_{\mathcal{D}} = 1$ and $g_{\mathcal{N}} = 1$, the matrix equation becomes

$$\begin{bmatrix} 4 & -2 \\ -2 & 2 \end{bmatrix} \begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \end{bmatrix} = \begin{bmatrix} 2 + \frac{1}{12}\hat{f}_0 + \frac{1}{3}\hat{f}_1 + \frac{1}{12}\hat{f}_2 \\ 1 + \frac{1}{12}\hat{f}_1 + \frac{1}{6}\hat{f}_2 \end{bmatrix},$$

which has a solution

$$\begin{bmatrix} \hat{u}_1 \\ \hat{u}_2 \end{bmatrix} = \begin{bmatrix} \frac{3}{2} + \frac{1}{24}\hat{f}_0 + \frac{5}{24}\hat{f}_1 + \frac{1}{8}\hat{f}_2 \\ 2 + \frac{1}{24}\hat{f}_0 + \frac{1}{4}\hat{f}_1 + \frac{5}{24}\hat{f}_2 \end{bmatrix}.$$

The finite element approximation $u^\delta(x) = g_{\mathcal{D}}\Phi_0(x) + \hat{u}_1\Phi_1(x) + \hat{u}_2\Phi_2(x)$ is therefore

$$u^\delta = \begin{cases} 1 + x + \dfrac{x}{12}\hat{f}_0 + \dfrac{5x}{12}\hat{f}_1 + \dfrac{x}{4}\hat{f}_2, & 0 \leqslant x \leqslant \frac{1}{2}, \\ 1 + x + \dfrac{1}{24}\hat{f}_0 + \dfrac{2+x}{12}\hat{f}_1 + \dfrac{1+4x}{24}\hat{f}_2, & \frac{1}{2} \leqslant x \leqslant 1. \end{cases}$$

Having gone through the worked example we might now question what components are required to construct a general Galerkin approximation based on

multiple elemental decompositions of the solution domain. Implicit in the implementation of this example was the assumption that we can differentiate and integrate the basis functions over the solution domain. In general, it is not practical analytically to integrate and differentiate, and so we adopt numerical rules to be discussed in Sections 2.4.1 and 2.4.2. However, to make this possible we must develop techniques to treat each element separately and thus permit us to automate the implementation. The construction of local elemental bases and the assembly of these into a global definition will be discussed in Section 2.3.

### 2.2.3 *Mathematical formulation*

In this section we shall construct the Galerkin approximation to a linear partial differential equation, similar to that discussed in Section 2.2.1, in a more mathematical framework. We consider the more general one-dimensional Helmholtz equation

$$\mathbb{L}(u) = \frac{\partial^2 u}{\partial x^2} - \lambda u + f = 0\,, \tag{2.2.11}$$

where $\lambda$ is a real positive constant. The equation is presumed to be supplemented with appropriate boundary conditions such as

$$u(0) = g_{\mathcal{D}}\,, \quad \frac{\partial u}{\partial x}(l) = g_{\mathcal{N}}\,.$$

As indicated by the boundary conditions, we wish to determine the solution in the interval $0 < x < l$, which we shall denote by $\Omega$.

Multiplying eqn (2.2.11) by an arbitrary test function $v(x)$, the properties of which are to be defined, and integrating over the domain $\Omega$, we obtain

$$\int_0^l v \frac{\partial^2 u}{\partial x^2}\,\mathrm{d}x - \int_0^l \lambda v u\,\mathrm{d}x + \int_0^l v f\,\mathrm{d}x = 0\,.$$

Providing $u(x)$ and $v(x)$ are sufficiently smooth, we can integrate the first term by parts to arrive at

$$\int_0^l \frac{\partial v}{\partial x}\frac{\partial u}{\partial x}\,\mathrm{d}x + \int_0^l \lambda v u\,\mathrm{d}x = \int_0^l v f\,\mathrm{d}x + \left[v\frac{\partial u}{\partial x}\right]_0^l\,. \tag{2.2.12}$$

If we introduce the notation

$$a(v,u) = \int_0^l \left(\frac{\partial v}{\partial x}\frac{\partial u}{\partial x} + \lambda v u\right)\mathrm{d}x\,,$$
$$f(v) = \int_0^l v f\,\mathrm{d}x + \left[v\frac{\partial u}{\partial x}\right]_0^l\,,$$

then eqn (2.2.12) can be written as

$$a(v,u) = f(v)\,. \tag{2.2.13}$$

In structural mechanics, $a(u,u)$ is referred to as the *strain energy*, and the space of all functions which have a finite strain on $\Omega$ is called the *energy* space, which is denoted by $E(\Omega)$:

$$E(\Omega) = \{u \mid a(u,u) < \infty\}\,.$$

Associated with the energy space is the energy norm $||u||_E$ defined as

$$||u||_E = \sqrt{a(u,u)}\,. \tag{2.2.14}$$

Functions that belong to the energy space are called $H^1$ functions and satisfy the condition that the integral of the square of the function plus the square of its derivative are bounded.

We consider solutions to eqn (2.2.11) where the forcing function $f(x)$ is *well behaved* in the sense that $f(v)$ is finite. Therefore, we only consider candidate or *trial* solutions to eqn (2.2.12) which lie in the energy space and satisfy the Dirichlet boundary condition. This space is called the *trial space* and is denoted by $\mathcal{X}$. For our problem the trial space is defined by

$$\mathcal{X} = \{u \mid u \in H^1,\ u(0) = g_D\}\,.$$

Similarly, we define the space of all test functions, denoted by $\mathcal{V}$, which are homogeneous on all Dirichlet boundaries, that is,

$$\mathcal{V} = \{v \mid v \in H^1,\ v(0) = 0\}\,.$$

The test space $\mathcal{V}$ is sometimes said to be in $H^1_0$, where the subscript 0 refers to the fact that it is in the homogeneous space. We can now define the generalised or weak formulation of eqn (2.2.11) as follows:

find $u \in \mathcal{X}$, such that

$$a(v,u) = f(v)\,, \quad \forall\, v \in \mathcal{V}\,. \tag{2.2.15}$$

The weak problem is still an infinite-dimensional problem because the trial and test spaces, $\mathcal{X}$ and $\mathcal{V}$, contain an infinite number of functions. Therefore, we select subspaces $\mathcal{X}^\delta$ ($\mathcal{X}^\delta \subset \mathcal{X}$) and $\mathcal{V}^\delta$ ($\mathcal{V}^\delta \subset \mathcal{V}$) which contain a finite number of functions. In the spectral/*hp* element method we have two discretisation approaches, as denoted by $h$ (element size) and $p$ (polynomial order). We therefore interpret the use of $\delta$ in $\mathcal{X}^\delta$ and $\mathcal{V}^\delta$ to refer to these discretisation concepts, and so $\delta$ may be thought of as being a function of $h$ (or similarly $N_{\text{el}}$) as well as $p$. The approximate form of the weak solution can then be stated as follows:

find $u^\delta \in \mathcal{X}^\delta$, such that

$$a(v^\delta, u^\delta) = f(v^\delta)\,, \quad \forall\, v^\delta \in \mathcal{V}^\delta\,. \tag{2.2.16}$$

We note that in eqn (2.2.16) we have not imposed any Dirichlet boundary conditions. To impose Dirichlet boundary conditions we lift the solution by decomposing the function $u^\delta \in \mathcal{X}^\delta$ into a known component, $u^{\mathcal{D}}$, which lies in the

trial space ($u^{\mathcal{D}} \in \mathcal{X}^\delta$) and satisfies the Dirichlet boundary condition, and an unknown component, $u^{\mathcal{H}}$, which lies in the test space ($u^{\mathcal{H}} \in \mathcal{V}^\delta$) and is homogeneous or zero on the Dirichlet boundary. In other words,

$$u^\delta = u^{\mathcal{H}} + u^{\mathcal{D}}\,,$$

where

$$u^{\mathcal{H}}(0) = 0\,, \quad u^{\mathcal{D}}(0) = g_{\mathcal{D}}\,.$$

In the standard Galerkin approximation the same set of functions are used for both the test and trial functions. This is now possible since $u^{\mathcal{H}}$ and $v^\delta$ are both in $\mathcal{V}^\delta$. The Galerkin form of the problem can now be stated as follows:

find

$$u^\delta = u^{\mathcal{D}} + u^{\mathcal{H}}\,, \quad \text{where } u^{\mathcal{H}} \in \mathcal{V}^\delta,\ u^\delta \in \mathcal{X}^\delta\,,$$

such that

$$a(v^\delta, u^{\mathcal{H}}) = f^*(v^\delta)\,, \quad \forall\, v^\delta \in \mathcal{V}^\delta\,, \tag{2.2.17}$$

where

$$f^*(v^\delta) = f(v^\delta) - a(v^\delta, u^{\mathcal{D}})\,.$$

For this linear equation another way of constructing the Galerkin solution is from a variational point of view. Equation (2.2.11) is the minimal solution to the functional

$$\mathcal{F}(v) = \int_0^l \left[ \left( \frac{\partial v}{\partial x} \right)^2 + \lambda\,(v)^2 - 2vf \right] \mathrm{d}x\,.$$

Therefore, if we minimise $\mathcal{F}(v)$ over the infinite-dimensional space $\mathcal{V}$ we will find the solution to eqn (2.2.11) which is the Euler equation of this functional. Replacing the variational problem by a finite-dimensional subspace $\mathcal{V}^\delta$ leads to the Ritz–Galerkin method (see Strang and Fix [443]).

### 2.2.4 *Mathematical properties of the Galerkin approximation*

In this section we introduce some significant properties of the Galerkin approximation. We consider the approximation $u^\delta$ to the solution $u$, where $u^\delta \in \mathcal{X}^\delta$ and satisfies

$$a(v^\delta, u^\delta) = f(v^\delta)\,, \quad \forall\, v^\delta \in \mathcal{V}^\delta\,. \tag{2.2.18}$$

We mention that $a(v, u)$ is a *symmetric*, *bilinear form* which means

$$a(v, u) = a(u, v)\,, \tag{2.2.19a}$$

$$a(c_1 v + c_2 w, u) = c_1 a(v, u) + c_2 a(w, u)\,, \tag{2.2.19b}$$

where $c_1$ and $c_2$ are constants and $u$, $v$, and $w$ are functions. Further, the operator $a(v, u)$ is said to be continuous (or bounded) if

$$|a(v, u)| \leqslant C_1 ||v||_1 ||u||_1\,, \tag{2.2.19c}$$

where $C_1 < \infty$ and the subscript denotes the norm in $H^1$. It is elliptic (or coercive) if

$$a(u,u) \geqslant C_2 ||u||_1^2 , \tag{2.2.19d}$$

where $C_2 > 0$.

We note that eqn (2.2.18) is equivalent to eqn (2.2.17) since $a(v^\delta, u^\delta) = a(v^\delta, u^D) + a(v^\delta, u^{\mathcal{H}})$ using the bilinearity of $a(v,u)$ (eqn (2.2.19b)).

If $a(v^\delta, u^\delta)$ is a continuous, elliptic, bilinear form that is not necessarily symmetric and $f(v^\delta)$ is in the dual space of $\mathcal{V}^\delta$, then the Lax–Milgram theorem guarantees both *existence* and *uniqueness* of the solution of the Galerkin problem (2.2.18) (see Brenner and Scott [75]).

2.2.4.1 *Uniqueness*

To show that the solution $u^\delta$ is unique we assume that there are two distinct solutions $u_1$ and $u_2$ ($u_1, u_2 \in \mathcal{X}^\delta$) which satisfy

$$a(v^\delta, u_1) = f(v^\delta) , \quad \forall\, v^\delta \in \mathcal{V}^\delta \tag{2.2.20a}$$

and

$$a(v^\delta, u_2) = f(v^\delta) , \quad \forall\, v^\delta \in \mathcal{V}^\delta . \tag{2.2.20b}$$

Subtracting eqn (2.2.20a) from eqn (2.2.20b), we obtain

$$a(v^\delta, u_1) - a(v^\delta, u_2) = a(v^\delta, u_1 - u_2) = 0 \tag{2.2.20c}$$

using the bilinearity of $a(v,u)$. Now $u_1 - u_2 \in \mathcal{V}^\delta$ and therefore we can set $v^\delta = u_1 - u_2$, so eqn (2.2.20c) becomes

$$a(u_1 - u_2, u_1 - u_2) = 0 .$$

However, this implies that $||u_1 - u_2||_E = 0$. This is only possible if $u_1 = u_2$, which contradicts the assumption that they are distinct. We therefore conclude that there is only one unique solution. Strictly speaking, $||u_1 - u_2||_E = 0$ only implies that $u_1 = u_2$ if $\lambda \neq 0$. When $\lambda = 0$ the solution is only unique up to an arbitrary constant, that is, $u_1 - u_2 = C$. The constant, $C$, is necessarily zero if Dirichlet boundary conditions are specified, although the norm $||u_1 - u_2||_E$ cannot distinguish between functions that differ by an arbitrary constant when $\lambda = 0$.

2.2.4.2 *Orthogonality of the error to the test space in the energy norm*

The error between the exact and approximate solution, $\varepsilon = u - u^\delta$, is orthogonal to all functions in the finite-dimensional test space $\mathcal{V}^\delta$ in the energy norm, that is,

$$a(v^\delta, \varepsilon) = 0 , \quad \forall\, v^\delta \in \mathcal{V}^\delta . \tag{2.2.21a}$$

To prove this property we recall that the exact solution satisfies the weak equation (2.2.15); in other words,

$$a(v,u) = f(v)\,, \quad \forall\, v \in \mathcal{V}\,,$$

and the approximation satisfies eqn (2.2.18). The finite-dimensional test space $\mathcal{V}^\delta$ is a subspace of $\mathcal{V}$, and so the exact solution also satisfies

$$a(v^\delta,u) = f(v^\delta)\,, \quad \forall\, v^\delta \in \mathcal{V}^\delta\,. \tag{2.2.21b}$$

Subtracting eqn (2.2.18) from eqn (2.2.21b) with $\varepsilon = u - u^\delta$ and using the bilinearity of $a(v,u)$ gives eqn (2.2.21a).

2.2.4.3 *Minimal property of error in the energy norm*

We can show that the finite element solution $u^\delta$ is the solution in $\mathcal{X}^\delta$ which minimises the energy norm of the error, that is,

$$||u - u^\delta||_E = \min_{w^\delta \in \mathcal{X}^\delta} ||u - w^\delta||_E\,. \tag{2.2.22a}$$

To demonstrate this result we let $\varepsilon = u - u^\delta$ and observe that for any $w^\delta \in \mathcal{X}^\delta$ we can write

$$||u - w^\delta||_E^2 = ||u - u^\delta + u^\delta - w^\delta||_E^2 = ||\varepsilon + v^\delta||_E^2\,,$$

where $v^\delta = u^\delta - w^\delta \in \mathcal{V}^\delta$. From the definition of the energy norm (2.2.14) and using the bilinearity of $a(v,u)$ (eqn (2.2.19b)), we obtain

$$||u - w^\delta||_E^2 = a(\varepsilon + v^\delta, \varepsilon + v^\delta) = a(\varepsilon,\varepsilon) + 2a(v^\delta,\varepsilon) + a(v^\delta,v^\delta)\,. \tag{2.2.22b}$$

Now, since $v^\delta \in \mathcal{V}^\delta$, we know from eqn (2.2.21a) that $a(v^\delta,\varepsilon) = 0$. Therefore, if there were any choices of $w^\delta$ which gave a smaller error than $u - u^\delta$, in the energy norm, it would have to make the last term of eqn (2.2.22b) negative. However, if $v \neq 0$ then $a(v,v) > 0$, and so the minimising choice of $w^\delta$ is one that sets $v^\delta = 0$, thus implying that $w^\delta = u^\delta$ and proving eqn (2.2.22a).

2.2.4.4 *Equivalence of polynomial bases in the energy norm*

An almost trivial observation from the uniqueness of the Galerkin approximation is that any two linearly-independent expansions which span the same trial space $\mathcal{X}^\delta$ necessarily have the same approximate solution $u^\delta(x)$. So if we consider two solutions $u_1^\delta(x) = \sum_i^P \alpha_i\psi_i(x)$ and $u_2^\delta(x) = \sum_i^P \beta_i h_i(x)$, where both expansion functions are in a polynomial space of order $P$ (i.e., $\psi_i(x), h_i(x) \in \mathcal{P}_P$), and if the solutions $u_1^\delta(x)$ and $u_2^\delta(x)$ are both determined as solutions to the Galerkin approximation (2.2.18) then we know that

$$u_1^\delta(x) = u_2^\delta(x) \quad \Rightarrow \quad \sum_{i=0}^{P} \alpha_i\psi_i(x) = \sum_{i=0}^{P} \beta_i h_i(x)\,.$$

The important implication of this statement is that any error estimates are independent of the type of the polynomial expansion and only depend on the polynomial space. Nevertheless, different choices of polynomial expansion bases can have an important effect on the numerical conditioning of the algebraic systems resulting from the Galerkin approximation, as discussed in Section 2.3.2.1.

### 2.2.5 *Residual equation for the $C^0$ test and trial functions*

As we have seen in the example of Section 2.2.2, the finite element approximation $u^\delta$ to a second-order differential equation can be constructed from a class of functions which are $C^0$ continuous, that is, the approximation (but not the derivative of the approximation) is continuous everywhere in the domain $\Omega$. Therefore, when the solution domain $\Omega$ is subdivided into finite elements, denoted by $\Omega^e$, although the derivative is continuous within each element, at the boundary between elements the derivative may be discontinuous.

We note that for a $C^0$ approximation the substitution of $u^\delta$ into the weak equation (2.2.5) is not equivalent to setting the weighted residual $(v^\delta, R)$ equal to zero (i.e., $(v^\delta, R) \neq 0$, where $\mathbb{L}(u^\delta) = R$ from eqn (2.2.3)). To appreciate why, we can integrate the left-hand side of eqn (2.2.6) by parts and recover a form similar to eqn (2.2.3).

We observe that the integrand on the left-hand side of eqn (2.2.6) involves the derivatives of $u^\delta$ and $v^\delta$, which are only piecewise continuous within each element when using a $C^0$ expansion. Therefore, to evaluate this integral we have to perform a series of integrals over each element $\Omega^e$. If there are $N_{\text{el}}$ elements we find

$$\begin{aligned}\int_0^1 \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}\,\mathrm{d}x &= \sum_{e=1}^{N_{\text{el}}} \int_{\Omega^e} \frac{\partial v^\delta}{\partial x}\frac{\partial u^\delta}{\partial x}\,\mathrm{d}x \\ &= -\sum_{e=1}^{N_{\text{el}}} \int_{\Omega^e} v^\delta \frac{\partial^2 u^\delta}{\partial x^2}\,\mathrm{d}x + \sum_{e=1}^{N_{\text{el}}} \left[ v^\delta \frac{\partial u^\delta}{\partial x} \right]_{\Omega_L^e}^{\Omega_R^e} \\ &= -\int_0^1 v^\delta \frac{\partial^2 u^\delta}{\partial x^2}\,\mathrm{d}x + \sum_{e=1}^{N_{\text{el}}} \left[ v^\delta \frac{\partial u^\delta}{\partial x} \right]_{\Omega_L^e}^{\Omega_R^e}, \end{aligned} \tag{2.2.23}$$

where $\Omega_R^e$ and $\Omega_L^e$ denote the $x$ values of the left and right ends of the domain $\Omega^e$, respectively. If the approximating function was $C^1$ continuous (that is, the function and its first derivative are continuous everywhere) then by definition

$$v^\delta \frac{\partial u^\delta}{\partial x}\bigg|_{\Omega_R^e} = v^\delta \frac{\partial u^\delta}{\partial x}\bigg|_{\Omega_L^{e+1}},$$

and we recover the standard 'integration by parts' result. However, since the finite element is globally only $C^0$ continuous, all the terms at the interior elemental boundaries remain. Now, by substituting eqn (2.2.23) into eqn (2.2.6) and rearranging, we obtain

$$\begin{aligned}&-\int_0^1 v^\delta \left( \frac{\partial^2 u^\delta}{\partial x^2} + f \right) \mathrm{d}x - v^\delta \frac{\partial u^\delta}{\partial x}\bigg|_{\Omega_L^1} \\ &\quad + \sum_{e=1}^{N_{\text{el}}-1} \left[ v^\delta \frac{\partial u^\delta}{\partial x}\bigg|_{\Omega_R^e} - v^\delta \frac{\partial u^\delta}{\partial x}\bigg|_{\Omega_L^{e+1}} \right] + \left[ v^\delta \frac{\partial u^\delta}{\partial x}\bigg|_{\Omega_R^{N_{\text{el}}}} - v^\delta(1) g_{\mathcal{N}} \right] = 0 . \end{aligned} \tag{2.2.24}$$

The first term is the standard weighted residual. The second term is zero since $\Omega_L^1$ is a Dirichlet boundary and so $v^\delta(\Omega_L^1) = 0$. The third term represents the *jump* in the derivative of the approximation at the element boundaries in the interior of the domain, and the last term represents the difference between the exact and approximate Neumann boundary conditions. Upon convergence to the exact solution, the jump in the derivative must therefore become zero and the Neumann condition must also be exactly satisfied.

If we use a $C^1$ expansion which exactly satisfies the Neumann boundary conditions, then eqn (2.2.24) becomes the standard weighted residual, that is, $(v^\delta, R) = 0$.

## 2.3 One-dimensional expansion bases

Having defined the finite element framework in terms of the Galerkin formulation, we can now consider different types of one-dimensional expansion bases and provide some explanation of their construction.

An essential part of constructing different expansion bases will be to introduce a standard elemental region within which we will define the standard expansions. We will then discuss how to assemble the global expansion bases from these local definitions. This type of elemental construction also provides an efficient way to numerically implement the spectral/*hp* element technique once we have addressed how to numerically integrate and differentiate polynomial functions. This will be dealt with in Section 2.4; elemental expansion bases in multiple dimensions will be dealt with in Chapter 3.

At this stage we will only be concerned with polynomial expansions. Traditionally, the finite element method has always used polynomial expansions. This may be attributed to the historical use of Taylor series expansions which allow analytical functions to be expressed in terms of polynomials. Polynomial functions also have the added advantage of discrete integration rules which enable easy computer implementation.

In the *h*-type method, a fixed-order polynomial is used in every element and convergence is achieved by reducing the size of the elements. This is the so-called *h*-type extension, where $h$ represents the characteristic size of an element, and was illustrated for two elements in Section 2.2.2. This type of extension aids in geometric flexibility, especially in high dimensions.

In the *p*-type method, a fixed mesh is used and convergence is achieved by increasing the order of the polynomial in every element. This is the so-called *p*-type extension, where $p$ represents the expansion order in the elements. This type of extension aids rapid convergence for smooth problems. If the whole solution domain is treated as a single element then the *p*-type method becomes a spectral method.

The spectral/*hp* element method combines attributes from both the *h-type* and *p-type* extensions, permitting a combination of both approaches. We also note that, with the exception of a global spectral method, in most *p-type* methods

there is an implied *h-type* decomposition to generate the initial mesh upon which the *p-type* extension is applied.

### 2.3.1 *Elemental decomposition: the h-type extension*

♣5 One of the primary advantages of the finite element and finite volume methods is the ability to resolve complex geometries. This capability is inherently dependent on being able to decompose the solution domain into small subdomains or elements. In this section, we will demonstrate the technique of decomposing the expansion into elemental contributions, that is, the '*h*-type extension' process. We note, however, that elemental decomposition is an integral part of both *h*- and *p*-type finite element methods as the *p*-type extension is based upon an initial mesh or *h*-type discretisation.

As discussed in the next three sections, the principal use of elemental representation is to enable the treatment of operations on a local elemental basis. This not only simplifies the implementation but also allows many operations to be performed more efficiently. For the one-dimensional case, the decomposition may seem unnecessarily involved; however, the same principles are applied to the decomposition in multiple dimensions. Therefore, the one-dimensional case is explained in detail as a building block for understanding decomposition in multiple dimensions. The decomposition is explained in terms of the linear finite element expansion; however, the same techniques can be applied with the higher-order *p*-type expansions discussed in Section 2.3.2.1.

#### 2.3.1.1 *Partitioning of the solution domain*

When using an *h*-type method the solution domain is subdivided or partitioned into *non-overlapping* subdomains or elements within which a polynomial expansion is used.

Considering a solution domain $\Omega$, we can partition it into a mesh containing $N_{\text{el}}$ elements, denoted by $\Omega^e$, such that the union of the non-overlapping elements equals the original domain, that is,

$$\Omega = \bigcup_{e=1}^{N_{\text{el}}} \Omega^e \,, \quad \text{where} \bigcap_{e=1}^{N_{\text{el}}} \Omega^e = \emptyset \,.$$

For the domain $\Omega = \{x \mid 0 < x < l\}$ a specific mesh can be denoted by the points

$$0 = x_0 < x_1 < \cdots < x_{N_{\text{el}}-1} < x_{N_{\text{el}}} = l \,.$$

Therefore, the $e$th element is defined as

$$\Omega^e = \{x \mid x_{e-1} < x < x_e\} \,.$$

[5] Decomposition of a solution domain into elemental regions: *h*-type extensions.

As an example we consider the case shown in Fig. 2.2 where the solution domain, $\Omega = \{x \mid 0 < x < l\}$, is subdivided into $N_{\text{el}} = 3$ non-equal elements. The mesh is denoted by the $N_{\text{el}} + 1$ points $x_0 = 0,\ x_1,\ x_2,\ x_3 = l$, and therefore the first element is defined as

$$\Omega^1 = \{x \mid x_0 < x < x_1\}\,.$$

2.3.1.2 *The standard element and the one-dimensional linear finite element expansion*

In Fig. 2.2 the global expansion modes for the linear finite element expansion over the $N_{\text{el}} = 3$ elemental domains are also shown. As is typical in a linear finite element expansion, each mode has a unit value at the end of one of the elemental domains and decays linearly to zero across the neighbouring elements. Therefore, there are $N_{\text{dof}} = 4$ degrees of freedom in this expansion which are $\Phi_0(x), \Phi_1(x), \Phi_2(x)$, and $\Phi_3(x)$. The global modes are nonzero on, at most, two elemental regions. It would therefore be very uneconomical to consider an expansion in terms of global modes, particularly when using a large number of elements.

We can see that on an elemental level each global mode only consists of two linearly-varying functions which are also shown on the right of Fig. 2.2. Therefore, if we introduce the one-dimensional standard element, $\Omega_{\text{st}}$, such that

$$\Omega_{\text{st}} = \{\xi \mid -1 \leqslant \xi \leqslant 1\}\,,$$

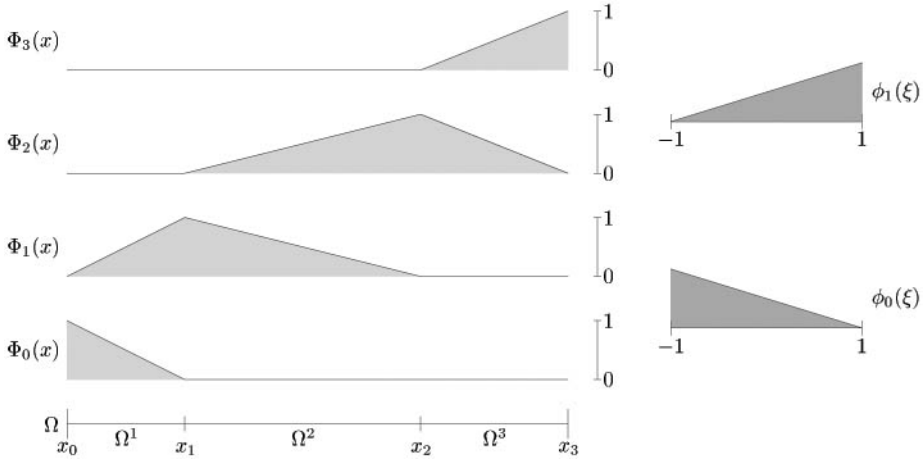



FIG. 2.2. Elemental decomposition of the solution domain $\Omega$ into three elements $\Omega^1$, $\Omega^2$, and $\Omega^3$. Above this domain we show the global expansion modes $\Phi_0(x)$, $\Phi_1(x)$, $\Phi_2(x)$, and $\Phi_3(x)$ for a linear finite element expansion over the domain $\Omega$. On the right are the local expansion bases $\phi_0(\xi)$ and $\phi_1(\xi)$ defined in the standard region $\Omega^{\text{st}}$ which can be used to define the global expansion modes.

then we can define a similar linearly-varying function over $\Omega_{\rm st}$ in terms of the local coordinate $\xi$ as

$$\phi_0(\xi) = \begin{cases} \dfrac{1-\xi}{2}, & \xi \in \Omega_{\rm st}\,, \\ 0\,, & \xi \notin \Omega_{\rm st}\,, \end{cases} \qquad \phi_1(\xi) = \begin{cases} \dfrac{1+\xi}{2}, & \xi \in \Omega_{\rm st}\,, \\ 0\,, & \xi \notin \Omega_{\rm st}\,. \end{cases}$$

The standard element $\Omega_{\rm st}$ can be mapped to any elemental domain $\Omega^e$ via the transformation $\chi^e(\xi)$, which expresses the global coordinate $x$ in terms of the local coordinate $\xi$ as

$$x = \chi^e(\xi) = \frac{1-\xi}{2} x_{e-1} + \frac{1+\xi}{2} x_e\,, \quad \xi \in \Omega_{\rm st}\,. \tag{2.3.1}$$

This mapping has an analytic inverse, $(\chi^e)^{-1}(x)$, of the form

$$\xi = (\chi^e)^{-1}(x) = 2\,\frac{x - x_{e-1}}{x_e - x_{e-1}} - 1\,, \quad x \in \Omega^e\,.$$

The global modes $\Phi_i(x)$ can now be represented in terms of the local elemental expansion modes $\phi_p(\xi)$ by mapping the standard element $\Omega_{\rm st}$ to each elemental domain $\Omega^e$. For example, the first two global expansion modes $\Phi_0(x)$ and $\Phi_1(x)$ in Fig. 2.2 can be written as

$$\Phi_0(x) = \begin{cases} \dfrac{x - x_1}{x_0 - x_1}, & x \in \Omega^1\,, \\ 0\,, & x \notin \Omega^1\,, \end{cases} = \begin{cases} \phi_0(\xi) = \phi_0([\chi^1]^{-1}(x))\,, & x \in \Omega^1\,, \\ 0\,, & x \notin \Omega^1\,, \end{cases}$$

$$\Phi_1(x) = \begin{cases} \dfrac{x - x_0}{x_1 - x_0}, & x \in \Omega^1\,, \\ \dfrac{x - x_2}{x_1 - x_2}, & x \in \Omega^2\,, \\ 0\,, & \text{otherwise}\,, \end{cases} = \begin{cases} \phi_1(\xi) = \phi_1([\chi^1]^{-1}(x))\,, & x \in \Omega^1\,, \\ \phi_0(\xi) = \phi_0([\chi^2]^{-1}(x))\,, & x \in \Omega^2\,, \\ 0\,, & \text{otherwise}\,. \end{cases}$$

If a mapping for $\chi^e(\xi)$ other than the one given in eqn (2.3.1) has been used then the inverse mapping will not necessarily be analytic. This situation can arise in multiple dimensions where elements may be curved.

#### 2.3.1.3 *Parametric mapping*

The transformation $\chi^e(\xi)$ given in eqn (2.3.1) maps the *local* coordinate $\xi$ to the *global* coordinate $x$ ($x \in \Omega_e$) and can be interpreted as expanding the global coordinate, $x$, in terms of a linear finite element expansion. It, therefore, could have been written as

$$x = \chi^e(\xi) = \phi_0(\xi)\; x_{e-1} + \phi_1(\xi)\; x_e\,, \quad \xi \in \Omega_{\rm st}\,.$$

This technique of expressing the global coordinate, $x$, in terms of the local expansion function is known as *parametric mapping*. Typically, we refer to the

mapping as being *iso-parametric* if we use the same-order expansion to map the coordinates as we use to represent the dependent variables. If we use a higher- or lower-order mapping for the coordinates as compared to the dependent variable then the mapping is referred to as *super-* or *sub*-parametric, respectively. As we shall see in Section 4.1.3.2, parametric mappings provide a convenient way to express curved domains.

We note that the mapping in eqn (2.3.1) is linear and therefore so is its inverse. This means that the local expansion mode $\phi_p(\chi_e^{-1}(x))$ is a polynomial in $x$ as well as in $\xi$, and therefore under the mapping (2.3.1) the global expansion modes are also polynomials in $x$. However, when a higher-order polynomial mapping is used, as is necessary for curved elements, the global expansion may not remain a polynomial in $x$ although, by definition, it is always a polynomial in $\xi$.

#### 2.3.1.4 *Global assembly/direct stiffness summation*

To relate the concepts of local and global expansion bases we need to introduce the concept of global assembly or direct stiffness summation, as it is sometimes known. In this section we shall describe the process for a one-dimensional linear basis, but the same idea can be used in higher-order expansions and multiple dimensions. Let us recall that the finite element approximation $u^\delta$ in terms of the global modes is written as

$$u^\delta(x) = \sum_{i=0}^{N_{\text{dof}}-1} \hat{u}_i \Phi_i(x)\,.$$

We have seen in Section 2.3.1.2 that the global modes $\Phi_i(x)$ can be expressed in terms of the local expansion modes $\phi_p(\xi)$, and therefore we can express $u^\delta$ in terms of $\phi_p(\xi)$ as

$$u^\delta(x) = \sum_{i=0}^{N_{\text{dof}}-1} \hat{u}_i \Phi_i(x) = \sum_{e=1}^{N_{\text{el}}} \sum_{p=0}^{P} \hat{u}_p^e \phi_p^e(\xi)\,,$$

where in this case $P$ is the polynomial order of the expansion and $\phi_p^e(\xi) = \phi_p([\chi^e]^{-1}(x))$ (the superscript denotes the element in which the function is nonzero). As there are more of the local expansion coefficients, $\hat{u}_p^e$, than global expansion coefficients, $\hat{u}_i$, some further conditions are required to relate the local and global definitions of the solution $u^\delta(x)$.

For the linear finite element example shown in Fig. 2.3 where $P = 1$ and $N_{\text{el}} = 3$, the constraint is that the global modes are continuous everywhere, which implies

$$\begin{aligned} \hat{u}_1^1 &= \hat{u}_0^2\,, \\ \hat{u}_1^2 &= \hat{u}_0^3\,. \end{aligned} \tag{2.3.2}$$

The relationship between the local and global expansion coefficients is therefore

[3] Global assembly: assembling global bases and operations from local bases and operators.

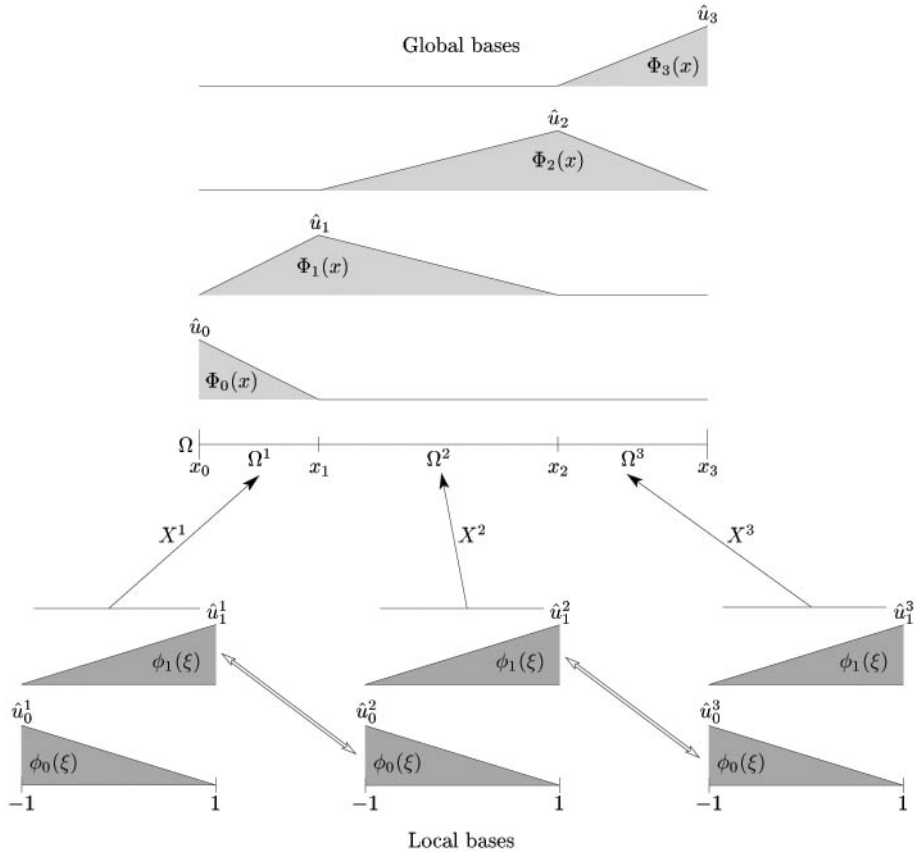



FIG. 2.3. Global and local expansion coefficients and bases in a three-element decomposition of the domain $\Omega$.

$$\begin{aligned}
\hat{u}_0^1 &= \hat{u}_0\,, \\
\hat{u}_1^1 &= \hat{u}_0^2 = \hat{u}_1\,, \\
\hat{u}_1^2 &= \hat{u}_0^3 = \hat{u}_2\,, \\
\hat{u}_1^3 &= \hat{u}_3\,.
\end{aligned}$$

In this example it can be seen that the local representation of the function has six elemental degrees of freedom ($N_{\text{eof}} = N_{\text{el}} \cdot (P+1) = 6$) but only four global degrees of freedom ($N_{\text{dof}} = 4$). The two constraints shown in eqn (2.3.2) ensure that $u^\delta(x)$ is $C^0$ continuous, which is a sufficient condition to ensure that the expansion is in $H^1$ space and thereby can be an admissible function for the trial space $\mathcal{X}^\delta$ for a second-order elliptic problem.

To construct a more general description of the local to global mapping we let $\hat{\boldsymbol{u}}_g$ denote a vector of all global coefficients,

$$\hat{\boldsymbol{u}}_g = [\hat{u}_0, \ldots, \hat{u}_{N_{\text{dof}}-1}]^\top ,$$

and if $\hat{\boldsymbol{u}}^e$ is a vector of the local coefficients (that is, $\hat{\boldsymbol{u}}^e = [\hat{u}_0^e, \hat{u}_1^e]$) in element $e$, then the vector of all local coefficients, denoted by $\hat{\boldsymbol{u}}_l$, can be written as

$$\hat{\boldsymbol{u}}_l = \begin{bmatrix} \hat{\boldsymbol{u}}^1 \\ \hat{\boldsymbol{u}}^2 \\ \vdots \\ \hat{\boldsymbol{u}}^{N_{\text{el}}} \end{bmatrix} .$$

The relationship between the local degrees of freedom and the global degrees can be expressed in terms of an *assembly matrix* $\boldsymbol{\mathcal{A}}$ such that

$$\boxed{\hat{\boldsymbol{u}}_l = \boldsymbol{\mathcal{A}} \hat{\boldsymbol{u}}_g ,} \tag{2.3.3a}$$

where $\boldsymbol{\mathcal{A}}$ is a very sparse matrix whose entries are typically 1 (but may be $-1$ in multiple dimensions or even contain a submatrix for non-conforming elements). For our example in Fig. 2.3 the full form of eqn (2.3.3a) is

$$\hat{\boldsymbol{u}}_l = \begin{bmatrix} \hat{u}_0^1 \\ \hat{u}_1^1 \\ \hat{u}_0^2 \\ \hat{u}_1^2 \\ \hat{u}_0^3 \\ \hat{u}_1^3 \end{bmatrix} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ 1 & & & \\ & & 1 & \\ & & 1 & \\ & & & 1 \end{bmatrix} \begin{bmatrix} \hat{u}_0 \\ \hat{u}_1 \\ \hat{u}_2 \\ \hat{u}_3 \end{bmatrix} .$$

We can also consider the *reverse* operation of constructing global operations from local operations. This is advantageous since we can perform operations locally within the elements and then assemble the global operation. In the Galerkin formulation this assembly process is typically associated with integral operations, which implies that we have to sum the local (elemental) contributions. For example, if we consider the integral of $u^\delta(x)$ with the global modes $\Phi_1(x)$ shown in Fig. 2.3, then we find

$$\int_\Omega \Phi_1(x) u^\delta(x)\, \mathrm{d}x = \int_{-1}^{1} \phi_1^1(\xi) u^\delta(\chi^1) \frac{\mathrm{d}\chi^1}{\mathrm{d}\xi}\, \mathrm{d}\xi + \int_{-1}^{1} \phi_0^2(\xi) u^\delta(\chi^2) \frac{\mathrm{d}\chi^2}{\mathrm{d}\xi}\, \mathrm{d}\xi .$$

From this we see that all integrals may also be computed in a standard region ($[-1, 1]$) which is convenient when performing numerical quadrature, as discussed in Section 2.4.1. The process of reassembling the global expansion from the local expansion on the elemental domains is called *global assembly* or *direct stiffness summation.*

The global assembly operation which constructs the global operations from the local operations is the *transpose* operation $\boldsymbol{\mathcal{A}}^\top$. To illustrate this operation

we can consider the integration of the global base, $\Phi(x)$, with respect to the function $u^\delta(x)$. Following the same convention as the local and global degrees of freedom, we denote the integration with respect to the global basis as $\boldsymbol{I}_g$, i.e.,

$$\boldsymbol{I}_g[i] = \int_\Omega \Phi_i(x) u^\delta(x)\, \mathrm{d}x\,,$$

and the integration with respect to the local bases $\phi(x)$ in element $e$ as $\boldsymbol{I}^e$, so that we can define a vector of local integral contributions, $\boldsymbol{I}_l$, such that

$$\boldsymbol{I}_l = \begin{bmatrix} \boldsymbol{I}^1 \\ \boldsymbol{I}^2 \\ \vdots \\ \boldsymbol{I}^{N_{\mathrm{el}}} \end{bmatrix}, \quad \text{where } \boldsymbol{I}^e = \begin{bmatrix} \displaystyle\int_{-1}^{1} \phi_0(\xi) u^\delta(\chi^e) \frac{\mathrm{d}\chi^e}{\mathrm{d}\xi}\, \mathrm{d}\xi \\ \vdots \\ \displaystyle\int_{-1}^{1} \phi_{P-1}(\xi) u^\delta(\chi^e) \frac{\mathrm{d}\chi^e}{\mathrm{d}\xi}\, \mathrm{d}\xi \end{bmatrix}.$$

The vector $\boldsymbol{I}_g$ can then be related to the local elemental vector $\boldsymbol{I}_l$ using the assembly matrix $\boldsymbol{\mathcal{A}}^\top$ by the operation

$$\boldsymbol{I}_g = \boldsymbol{\mathcal{A}}^\top \boldsymbol{I}_l\,. \tag{2.3.3b}$$

This operation essentially performs a summation of the local modes into the global expansion.

We also note that

$$\hat{\boldsymbol{u}}_g \neq \boldsymbol{\mathcal{A}}^\top \boldsymbol{\mathcal{A}} \hat{\boldsymbol{u}}_g$$

as the operation $\boldsymbol{\mathcal{A}}$ *scatters* the global degrees of freedom to the local elements. However, $\boldsymbol{\mathcal{A}}^\top$ *assembles* the global contribution by summing together various terms of the local degrees of freedom.

Here $\boldsymbol{\mathcal{A}}$ and $\boldsymbol{\mathcal{A}}^\top$ represent key operations required in the construction of a Galerkin spectral/*hp* element method since they permit us to define a series of local operators which can then be assembled using these operators. It can be appreciated that only global modes which are split into elemental contributions will have multiple entries in the columns of the $\boldsymbol{\mathcal{A}}$ matrix. When using a higher-order *p*-type expansion, as discussed in Section 2.3.2.1, the extra interior modes are all global degrees of freedom and will not need to be assembled in this fashion.

♦[4] In practice, we never construct the assembly matrix $\boldsymbol{\mathcal{A}}$ as it is very sparse and therefore numerically very inefficient to use as a matrix operator. An equivalent numerical operation is to use a *mapping array* for each element which contains the global location of every local degree of freedom. If we denote this array by

[4] Construction of a mapping array for global to local scatter and local to global assembly.

'map[$e$][$i$]', where $e$ denotes the element and $i$ is the local mode index, then for the example in Fig. 2.3 the array would be defined as

$$\text{map}[1][i] = \left\{ \begin{matrix} 0 \\ 1 \end{matrix} \right\}, \quad \text{map}[2][i] = \left\{ \begin{matrix} 1 \\ 2 \end{matrix} \right\}, \quad \text{map}[3][i] = \left\{ \begin{matrix} 2 \\ 3 \end{matrix} \right\}.$$

The scatter operation denoted by $\boldsymbol{\mathcal{A}}$ (see eqn (2.3.3a)) can then be evaluated as follows:

$$\left.\begin{array}{l} \text{do } e = 1, N_{\text{el}} \\ \quad \text{do } i = 0, N_m^e - 1 \\ \quad\quad \hat{\boldsymbol{u}}^e[i] = \hat{\boldsymbol{u}}_g[\text{map}[e][i]] \\ \quad \text{continue} \\ \text{continue} \end{array}\right\} \quad \Leftrightarrow \quad \hat{\boldsymbol{u}}_l = \boldsymbol{\mathcal{A}} \hat{\boldsymbol{u}}_g \,,$$

where $N_m^e = P^e + 1$. Alternatively, the global assembly operation may be written as follows:

$$\left.\begin{array}{l} \text{do } e = 1, N_{\text{el}} \\ \quad \text{do } i = 0, N_m^e - 1 \\ \quad\quad \hat{\boldsymbol{u}}_g[\text{map}[e][i]] = \hat{\boldsymbol{u}}_g[\text{map}[e][i]] + \hat{\boldsymbol{u}}^e[i] \\ \quad \text{continue} \\ \text{continue} \end{array}\right\} \quad \Leftrightarrow \quad \hat{\boldsymbol{u}}_g = \boldsymbol{\mathcal{A}}^\top \hat{\boldsymbol{u}}_l \,.$$

### 2.3.2 *Polynomial expansions: the p-type extension*

In multiple dimensions, complex domains make it difficult to identify global expansions analytically. The introduction of complex geometries can also generate different *scales* in the solution, which may have a very localised structure. Such considerations require the use of elemental decomposition, as discussed in Section 2.3.1. Therefore, if we decompose the solution domain into elemental regions which broadly capture either the geometry or the local scale of the problem then the application of the $p$-type extension can prove to be a numerically efficient approach to achieving a very accurate solution. In all that follows we will interpret the $p$-type extension as increasing the order of the polynomial expansion within an elemental region.

Before discussing the different types of $p$-type extension, we first define the $hp$ element space in one dimension. Recalling the definition of the standard element, $\Omega_{\text{st}}$, and the coordinate mapping $\chi^e(\xi)$ from $\Omega_{\text{st}}$ to an elemental region $\Omega^e$, we start by denoting the space of all polynomials of degree $P$ defined on the standard element $\Omega_{\text{st}}$ by $\mathcal{P}_P(\Omega_{\text{st}})$. The discrete $hp$ expansion space $\mathcal{X}^\delta$ is the set of all functions $u^\delta(x)$ which exist in $H^1$ and that are polynomials in $\xi$ within every element (e.g., $u^\delta(\chi^e(\xi)) \in \mathcal{P}_P(\Omega_{\text{st}})$), which is formally written as

$$\mathcal{X}^\delta = \{u^\delta \mid u^\delta \in H^1,\ u^\delta(\chi^e(\xi)) \in \mathcal{P}_{P^e}(\Omega_{\text{st}}),\ e = 1, \ldots, N_{\text{el}}\}\,. \tag{2.3.4}$$

This definition allows both the mapping $\chi^e(\xi)$ and the polynomial order $P^e$ to vary within each element $e$ thereby permitting both $h$-type refinement, which alters $\chi^e(\xi)$ and $N_{\text{el}}$, and $p$-type refinement, which alters $P^e$.

In principle, all of the construction discussed in Section 2.3.1 applies equally well to an *hp* elemental decomposition. As we shall see in Section 2.3.2.2, the most standard polynomial decompositions have what is known as a boundary and interior decomposition, which permits us to directly use the construction adopted for linear elements in Section 2.3.1 for higher-order polynomial expansions. Examples of polynomial expansions with this type of decomposition will be discussed in Sections 2.3.3.3 and 2.3.4. However, before introducing these expansions in Section 2.3.2.1 we will first try to explain why certain forms of polynomial expansions are more favourable than others.

#### 2.3.2.1 *Construction of a polynomial expansion*

In an *hp* elemental discretisation we can apply a polynomial expansion of any order within each elemental region. It is therefore appropriate to start our discussion of $p$-type methods by considering what makes an acceptable $p$-type expansion in a single domain.

The steps involved in designing an elemental $p$-type expansion, which we will also later adopt in constructing the unstructured basis in Section 3.2, are as follows.

- Determine a favourable expansion within a standard region.
- Modify the expansion so that it can easily be numerically implemented.

In the first step, a favourable expansion is typically an orthogonal or near-orthogonal set of functions within the standard regions. In the second step, the computational considerations of implementing this basis are taken into account and the basis is modified, if necessary, to facilitate this process. Typically, the basis is decomposed into contributions on the boundary and interior of the standard region since this simplifies the elemental decomposition process.

*Modal and nodal expansions*

Before discussing the benefits of different types of polynomial expansions, we first need to introduce the concepts of *modal* and *nodal* expansions. To illustrate the difference between a modal and a nodal polynomial expansion we introduce three expansion sets denoted by $\Phi_p^A(x), \Phi_p^B(x)$, and $\Phi_p^C(x)$ $(0 \leqslant p \leqslant P)$, in the region $\Omega_{\text{st}} = \{x \mid -1 \leqslant x \leqslant 1\}$. All of these expansions represent a complete set of polynomials up to order $P$ and are mathematically defined as

$$\begin{aligned}
\Phi_p^A(x) &= x^p\,, && p = 0, \ldots, P\,,\\
\Phi_p^B(x) &= \frac{\prod_{q=0,q\neq p}^{P}(x - x_q)}{\prod_{q=0,q\neq p}^{P}(x_p - x_q)}\,, && p = 0, \ldots, P\,,\\
\Phi_p^C(x) &= L_p(x)\,, && p = 0, \ldots, P\,.
\end{aligned}$$

The shape of these expansions can be seen in Fig. 2.4(a–c). The first expansion set simply increases the order of $x$ in a monomial fashion and we shall refer to it as the *moment* expansion (each order contributing an extra moment to the expansion). This basis is referred to as a *modal* or a *hierarchical* expansion because the expansion set of order $P-1$ is contained within the expansion set of order $P$. There is a notion of hierarchy in the sense that higher-order expansion sets are built from the lower-order expansion sets. If we denote the trial space containing all the polynomials in $\Phi_p^A(x)$ up to order $P$ by $\mathcal{X}_P^\delta$ then a hierarchical expansion is one where $\mathcal{X}_{P-1}^\delta \subset \mathcal{X}_P^\delta$; so if

$$\mathcal{X}_2^\delta = \{1, x, x^2\}$$

then

$$\mathcal{X}_3^\delta = \{1, x, x^2, x^3\} = \mathcal{X}_2^\delta \bigcup \{x^3\}\,.$$

The second polynomial $\Phi_p^B(x)$ is a Lagrange polynomial which is based on a series of $P+1$ nodal points $x_q$ which are chosen beforehand and could be, for example, equispaced in the interval (for further details see Section 2.3.4). The

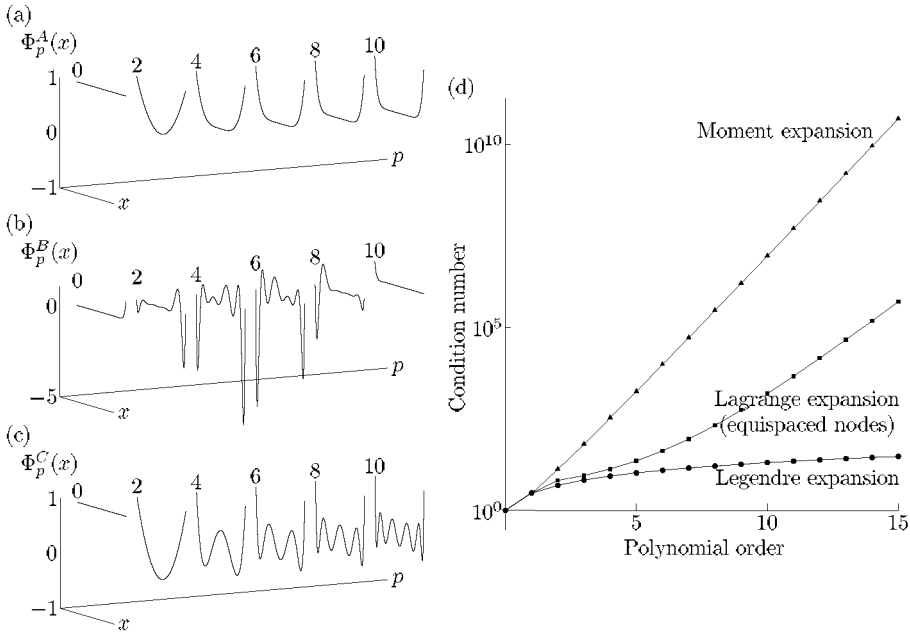



FIG. 2.4. Expansion modes ($p$ even) in space for three expansion bases: (a) $\Phi_p^A(x)$(moment), (b) $\Phi_p^B(x)$ (Lagrange), and (c) $\Phi_p^C(x)$ (Legendre) of order $P = 10$ in the region $-1 \leqslant x \leqslant 1$. (d) Linear–log plot of the condition number of the mass matrix versus polynomial order for the bases $\Phi_p^A(x), \Phi_p^B(x)$, and $\Phi_p^C(x)$.

Lagrange polynomial is a non-hierarchical basis (that is, $\mathcal{X}_P^\delta \not\subset \mathcal{X}_{P+1}^\delta$) because it consists of $P+1$ polynomials of order $P$. This can be contrasted with the hierarchical expansion $\Phi_p^A(x)$ which consists of polynomials of increasing order. The Lagrange basis has the notable property that $\Phi_p^B(x_q) = \delta_{pq}$, where $\delta_{pq}$ represents the Kronecker delta. This property implies that

$$u^\delta(x_q) = \sum_{p=0}^{P} \hat{u}_p \Phi_p^B(x_q) = \sum_{p=o}^{P} \hat{u}_p \delta_{pq} = \hat{u}_q \,,$$

where we see that the expansion coefficient $\hat{u}_p$ can be defined in terms of the approximate solution at the point $x_q$.

The coefficients, therefore, have a physical interpretation in that they represent the approximate solution at the points $x_q$. The points $x_q$ are referred to as *nodes* and the Lagrange expansion basis is referred to as a *nodal* expansion. Linear finite elements are an example of a nodal expansion where the nodal points are at the ends of the domain.

We draw a distinction between a *nodal* expansion and the *collocation* method (or collocation projection). In the collocation method, the equation being solved is *exactly* satisfied at the collocation points (see Section 2.1), whereas in a *nodal* expansion the expansion coefficients represent the *approximate* solution at a given set of nodes. However, a nodal expansion can be used in different types of methods such as the Galerkin or collocation method. It must be remembered that an approximate solution using a nodal expansion does not necessarily satisfy the equation exactly at the nodal points.

The final expansion, $\Phi_p^C(x)$, is also a hierarchical or modal expansion. However, in this case the expansion is the Legendre polynomial $L_p(x)$ (see Appendix A). By definition, this polynomial is orthogonal in the Legendre inner product

$$(L_p(x), L_q(x)) = \int_{-1}^{1} L_p(x) L_q(x)\, \mathrm{d}x = \frac{2}{2p+1}\delta_{pq} \,.$$

As we shall see, orthogonality has important numerical implications for the Galerkin method.

As a final point, we should comment on a potential confusion over the use of the word *modes*. In general, we shall refer to all expansion sets, whether they are modal or nodal, as consisting of expansion modes (usually called *shape functions* in structural mechanics).

*Choice of an expansion set*

The choice of an expansion set is influenced by its numerical efficiency, conditioning, and the linear independence of the basis, as well as its approximation properties. To illustrate some of these factors we consider the three expansions $\Phi_p^A(x), \Phi_p^B(x)$, and $\Phi_p^C(x)$ in a Galerkin projection.

The Galerkin or $L^2$ projection of a smooth function $f(x)$ in the domain $\Omega_{\mathrm{st}}$ onto the polynomial expansion $u^\delta(x)$ is the solution to the following problem:

find $u^\delta \in \mathcal{X}^\delta$ such that

$$(v^\delta, u^\delta) = (v^\delta, f)\,, \quad \forall\, v^\delta \in \mathcal{V}\,, \tag{2.3.5}$$

where $(u, v)$ is the Legendre inner product, see eqn (2.1.4).

In the absence of explicit boundary conditions, which need not be prescribed to obtain a solution for this problem, the trial and test space are both in the space of square integrable functions (that is, $\mathcal{X}^\delta = \mathcal{V}^\delta \subset L^2$). Letting $u^\delta(x) = v^\delta(x) = \sum_{p=0}^{P} \hat{u}_p \Phi_p(x)$, problem (2.3.5) is then equivalent to solving the matrix equation

$$\hat{\boldsymbol{v}}^\top [\boldsymbol{M}\hat{\boldsymbol{u}} = \boldsymbol{f}] \quad \Rightarrow \quad \boldsymbol{M}\hat{\boldsymbol{u}} = \boldsymbol{f}\,,$$

where

$$\boldsymbol{M}_{pq} = (\Phi_p, \Phi_q)\,, \quad \hat{\boldsymbol{u}} = [\hat{u}_0, \cdots, \hat{u}_P]^\top\,, \quad \boldsymbol{f}_p = (\Phi_p, f)\,.$$

The matrix $\boldsymbol{M}$ is known as the *mass matrix*. It is a square non-singular matrix of order $P+1$ and can be inverted to determine the solution

$$\hat{\boldsymbol{u}} = \boldsymbol{M}^{-1}\boldsymbol{f}\,.$$

The question of numerical efficiency for this problem is twofold. The first issue is the computational cost of constructing the matrix system, which may involve numerical integration (see Section 2.4.1). The second issue is the computational cost of inverting the matrix system to obtain the solution. It can be appreciated that the construction and inversion of the matrix $\boldsymbol{M}$ can be made far more efficient if there is some known structure to the matrix.

The moment expansion $\Phi_p^A(x)$ produces a mass matrix which has components $(0 \leqslant p, q \leqslant P)$ of the form

$$\begin{aligned}\boldsymbol{M}[p][q] = (\Phi_p^A, \Phi_q^A) = \int_{-1}^{1} x^p x^q \,\mathrm{d}x &= \left[\frac{x^{p+q+1}}{p+q+1}\right]_{-1}^{1} \\ &= \begin{cases} \dfrac{2}{p+q+1}\,, & p+q \text{ even}\,, \\ 0\,, & p+q \text{ odd}\,. \end{cases}\end{aligned}$$

Therefore, when constructing $\boldsymbol{M}$ using this basis we need only calculate half of the components. However, the inverse will still be full and the cost of inverting the matrix is typically the dominant operation.

The second expansion $\Phi_p^B(x)$ is the Lagrange polynomial and so it is associated with a set of nodal points $x_q$. For the purpose of this example we shall define the nodes as being equispaced in the domain $\Omega_{\mathrm{st}}$, and so in the interval $x_q = 2q/P - 1$ $(0 \leqslant q \leqslant P)$ which is a common finite element nodal expansion. As we shall see in Section 2.3.4.2, a much better choice of points is at the Gaussian quadrature zeros. There is no explicit form for the mass matrix when using numerical integration and the matrix is full. Therefore, the construction of the

mass matrix using $\Phi_p^B(x)$ is twice as expensive as $\Phi_p^A(x)$, although the matrix inversion is no more expensive.

The third expansion $\Phi_p^C(x)$ is the Legendre polynomial. By definition, this expansion produces a mass matrix which is diagonal because the components of the mass matrix are

$$\boldsymbol{M}[p][q] = (\Phi_p^C, \Phi_q^C) = \int_{-1}^{1} L_p(x) L_q(x)\, \mathrm{d}x = \frac{2}{2p+1}\delta_{pq}\,.$$

This matrix is very easy to construct and invert, and therefore might be considered to be numerically the most efficient of the three expansions. We note, however, that the basis cannot easily be extended to an elemental decomposition which is globally $C^0$ continuous since the continuity constraints destroy the orthogonality of the global matrix structure.

A further consideration is the conditioning of the matrix $\boldsymbol{M}$ which is related to the linear independence of the expansion. The condition number $\kappa_2$ is very important in the numerical inversion of matrix systems; a full discussion can be found in Isaacson and Keller [251]. The condition number $\kappa_2$ is defined as

$$\kappa_2 = ||\boldsymbol{M}||_2 \cdot ||\boldsymbol{M}^{-1}||_2\,,$$

where $||\boldsymbol{M}||_2$ denotes the matrix $L^2$ norm of $\boldsymbol{M}$.

When numerically inverting a matrix system there is an error associated with the inexact representation of the matrix due to round-off error. If a matrix system is ill-conditioned the round-off error in the matrix system can lead to large errors in the solution. Further, when using iterative techniques to invert the system the number of iterations required to perform the inversion typically depends on the conditioning of the matrix.

The condition number in the $L^2$ norm for the three types of expansion bases $\Phi_p^A(x), \Phi_p^B(x)$, and $\Phi_p^C(x)$ is shown in Fig. 2.4(d) as a function of polynomial order. We see that the condition number of the mass matrix for the moment expansion grows as $\kappa_2 \propto 10^P$. Initially, the conditioning of the equispaced Lagrange basis is relatively good; however, after about $P \approx 5$ the condition number also starts to grow as $\kappa_2 \propto 10^P$. In contrast, the Legendre basis is very well conditioned for all values of $P$. This is because the $L^2$ matrix norm for a real symmetric matrix is the ratio of the maximum to minimum eigenvalues, and so the condition number for the Legendre mass matrix is exactly $\kappa_2 = 2P + 1$.

The poor conditioning of the moment and Lagrange expansion reflects the fact that the basis is becoming numerically linearly dependent. This is particularly evident for $\Phi_p^A(x)$, as shown in Fig. 2.4(a), where we have plotted the even moment expansion modes as a function of $x$, for different polynomial orders $p$. We observe that the mode for $p = 8$ is practically indistinguishable from the mode when $p = 10$. Although each mode of $\Phi_p^B(x)$ is clearly distinguishable from the other in Fig. 2.4(b), the poor conditioning of this basis can be attributed to the high level of oscillations towards the end of the region, which can be seen in

modes $p = 4$ and $p = 6$. As we shall see in Section 2.3.4.2, these oscillations are controlled by a better choice of nodal points, which makes it possible to obtain independently-shaped modes with well-behaved bounds, as shown by the modes of $\Phi_p^C(x)$ in Fig. 2.4(c).

Another set of orthogonal polynomials which have been extensively used in spectral methods are the Chebyshev polynomials (see Gottlieb and Orszag [199]). These polynomials have constant amplitude oscillations throughout the region, which leads to their optimal convergence property in the maximum norm.

#### 2.3.2.2 *Boundary interior decomposition of polynomial bases*

From the discussion in Section 2.3.2.1, we might deduce that the 'best' choice for an expansion set is orthogonal polynomials such as the Legendre polynomials. This is true in so far as the hierarchy and orthogonality tend to lead to well-conditioned matrices. However, we also want to combine the expansion with the $h$-type elemental decomposition. The difficulty arises when we try to ensure a degree of continuity in the global expansion at elemental boundaries. For a second-order partial differential equation we have seen that it is sufficient to guarantee that the approximate solution $u^\delta$ is in $H^1$. Typically, in the finite element methods this is satisfied by imposing a $C^0$ continuity between elemental regions, that is, the global expansion modes are continuous everywhere in the solution domain although the derivatives may not be.

If we used the moment or the Legendre expansion basis in each elemental domain ($\phi_p(x) = \Phi_p^A(x)$ or $\phi_p(x) = \Phi_p^C(x)$) then the requirement that the approximation be in $H^1$ might be satisfied by prescribing an interface-matching condition of the form

$$\sum_{p=0}^{P} \hat{u}_p^e \phi_p^e(1) = \sum_{p=0}^{P} \hat{u}_p^{e+1} \phi_p^{e+1}(-1)\,,$$

where the superscripts $e$ and $e+1$ denote two adjacent domains.

Such a condition couples all of the degrees of freedom in one element with the modes in the adjacent element. Not only is this more difficult to implement than the standard finite element methods but it also destroys the orthogonality of the global matrix structure.

If the local expansions were constructed so that only a few expansion modes have magnitude at an elemental boundary then the matching condition can be imposed far more easily. For example, if we define an elemental expansion $\phi_p(\xi)$ in the region $-1 \leqslant \xi \leqslant 1$, where

$$\phi_p(-1) = \begin{cases} 1\,, & p = 0\,, \\ 0\,, & p \neq 0\,, \end{cases} \qquad \phi_p(1) = \begin{cases} 1\,, & p = P\,, \\ 0\,, & p \neq P\,, \end{cases}$$

then the $C^0$ continuity of the expansion is simply enforced by ensuring

$$\hat{u}_P^e \phi_P^e(1) = \hat{u}_0^{e+1} \phi_0^{e+1}(-1)\,.$$