RICE UNIVERSITY

Utilizing Parallel Optimization in Computational Fluid Dynamics

by

Michael Kokkolaras

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Approved, Thesis Committee:

Andrew J. Meade, Jr., Chairman
Associate Professor
Mechanical Engineering and Materials Science

John E. Dennis, Jr.,
Noah Harding Professor
Computational and Applied Mathematics

Pol D. Spanos
L. B. Ryon Chair
Mechanical Engineering and Materials Science

Houston, Texas

November, 1997

Abstract

Utilizing Parallel Optimization in Computational Fluid Dynamics

by

Michael Kokkolaras

General problems of interest in computational fluid dynamics are investigated by means of optimization. Specifically, in the first part of the dissertation, a method of optimal incremental function approximation is developed for the adaptive solution of differential equations. Various concepts and ideas utilized by numerical techniques employed in computational mechanics and artificial neural networks (e.g. function approximation and error minimization, variational principles and weighted residuals, and adaptive grid optimization) are combined to formulate the proposed method. The basis functions and associated coefficients of a series expansion, representing the solution, are optimally selected by a parallel direct search technique at each step of the algorithm according to appropriate criteria; the solution is built sequentially. In this manner, the proposed method is adaptive in nature, although a grid is neither built nor adapted in the traditional sense using a-posteriori error estimates. Variational principles are utilized for the definition of the objective function to be extremized in the associated optimization problems, ensuring that the problem is well-posed. Complicated data structures and expensive remeshing algorithms and systems solvers are avoided.

Computational efficiency is increased by using low-order basis functions and concurrent computing. Numerical results and convergence rates are reported for a range of steady-state problems, including linear and nonlinear differential equations associated with general boundary conditions, and illustrate the potential of the proposed method. Fluid dynamics applications are emphasized. Conclusions are drawn by discussing the method's limitations, advantages, and possible extensions.

The second part of the dissertation is concerned with the optimization of the viscous-inviscid-interaction (VII) mechanism in an airfoil flow analysis code. The VII mechanism is based on the concept of a transpiration velocity boundary condition, whose convergence to steady state is accelerated. The number of variables in the associated optimization problem is reduced by means of function approximation concepts to ensure high number of parallel processors to number of necessary function evaluations ratio. Numerical results are presented for the NACA-0012 and the supercritical RAE-2822 airfoils subject to transonic flow conditions using a parallel direct search technique. They exhibit a satisfactory level of accuracy. Speed-up depends on the number of available computational units and increases for more challenging flow conditions and airfoil geometries. The enhanced code constitutes a useful tool for airfoil flow analysis and design and an acceptable alternative to computationally expensive high fidelity codes.

Acknowledgments

I am immensely grateful to my supervising advisor, Dr. Andrew J. Meade, Jr., for his great guidance and encouragement. Working with him is a pleasure, especially because of his kind personality. I am deeply indebted to Dr. John Emory Dennis, Jr., for his keen interest and support, and for serving in my dissertation committee. I am profoundly thankful to Dr. Pol D. Spanos, for his precious advice and suggestions, and for his inspiring geniality. I appreciate the helpful comments of Dr. S. Scott Collis.

I am obliged to Dr. Boris A. Zeldin for numerous discussions, and to David Serafini for his continuously prompt assistance with the optimal utilization of the PDS software.

This work was made possible through the financial support under the NASA grant # CRA2-35504. The access to the computing facilities of the Center for Research on Parallel Computation (CRPC) at Rice University, the technical support of system administrator Michael Pearlman, and the short but effective use of the IBM SP supercomputer of the Texas Center for Computational and Information Sciences (TCCIS) at the University of Houston are highly appreciated.

I would also like to thank my colleagues and the staff of the department for providing a pleasant and stimulating working environment.

Finally, I deliberately do not acknowledge *here* the beloved, special persons who have contributed, as to every aspect of my life, to this dissertation, in the hope that my actions have been, are, and will be expressing my love and gratitude towards them.

Contents

Illustrations

Tables

Chapter 1

Introduction

Fluid mechanics is a branch of physics with tremendous importance to engineering. Large, complicated, and typically nonlinear problems need to be solved for the analysis and design of numerous engineering systems. The lack of closed form solutions and the costly expenditure required for the realization of wind tunnel experiments have motivated the development of computational fluid dynamics (CFD). The solutions obtained by CFD are, of course, approximate. Nevertheless, they constitute a significant and useful source of information on the existence and behavior of solutions to advanced fluid mechanics problems. In this regard, CFD researchers have focused on improving the approximation level and designing algorithms capable of solving high fidelity mathematical models efficiently.

Computational fluid dynamics had reached a high level of sophistication before parallel processing became available. However, there were still problems that took considerable amounts of human effort and computer time to solve. As the physical limits in sequential scientific computing seemed to be reached, parallel processing emerged as the new technological platform for the continued enhancement of computational methods. In order to exploit the newly available computational resources, the development of parallel algorithms began immediately. It seems that most research work has concentrated on the direct application of parallel methodologies to given problems. However, some CFD problems are either not amenable to parallelization, or may require a considerable amount of effort in redesigning algorithms so that they can be solved concurrently. At the same time, parallel techniques are quite effectively developed for, and applied to, other disciplines, e.g. optimization.

In this regard, the research work presented in this dissertation is focused on the development of numerical algorithms and methods which exploit parallel processing advances made in nonlinear optimization instead of directly applying concurrent computing methodologies to computational fluid dynamics. Specifically, algorithms are designed for the optimal incremental approximation to differential equation solutions and the enhancement of an airfoil analysis code by means of optimizing its coupling mechanism.

It should be emphasized at this point that this is a general introduction to the research work presented in this dissertation. Specific introductions and preliminary discussions are provided separately in the two parts of the dissertation.

1.1 Objectives

The development of the numerical method presented in Part I of the dissertation was motivated by a number of ideas borrowed from various computational disciplines. The objective of this research project was to combine techniques from various disciplines and design an efficient algorithm able to yield accurate results for a range of applications. The specific aims include: a) the utilization of artificial neural networks and error minimization concepts in order to design an algorithm for optimal incremental approximation, b) the establishment of relations between the concepts and properties of the proposed method to various numerical techniques, in particular to adaptive grid optimization, traditional Galerkin methods, and conventional iterative techniques, c) the utilization of variational principles in converting mathematical models of engineering problems into well-posed objective functions, d) the employment of the parallel direct search (PDS) optimization technique for the efficient solution of the associated optimization problem(s) at each stage of the proposed algorithm, and e) the establishment of the adaptive nature of the proposed algorithm and the promotion of its advantages by comparing it to traditional adaptive and multigrid techniques. In addition, generalization aspects of the proposed method are discussed and future work directions are proposed.

The work presented in Part II of the dissertation has the following goals: a) to enhance a given viscous-inviscid-interaction (VII) code for airfoil flow analysis in terms of accuracy and efficiency, b) to model the VII coupling mechanism in the form of a well-posed optimization problem, c) to solve the resulting optimization problem efficiently and successfully by means of the PDS technique, demonstrating in this manner the usefulness of PDS in solving engineering problems of interest, d) to develop heuristics to further enhance speed-up, e) to test the enhanced code under challenging conditions, and f) to establish the code as a useful alternative to higher fidelity parallel codes for airfoil analysis and design.

1.2 Outline

The text is organized as follows.

In the first part of the dissertation, a method for optimal incremental function approximation is proposed for the solution of differential equations. The concepts and ideas, borrowed from various scientific computing areas, are reviewed in Chapter 2. The proposed method is presented and discussed in Chapter 3. For illustration purposes, function approximation problems are investigated in Section 3.4. Linear self- and nonself-adjoint differential equations are considered, and numerical examples are presented, in Chapter 4. In particular, the Poisson's equation in one and two dimensions and the linear but challenging one-dimensional convection-diffusion equation are solved in Sections 4.1.1 and 4.2.1, respectively.

The issue of computational cost is addressed in Section 4.1.2. General boundary conditions are discussed in Section 4.1.3. Nonlinear applications are considered in Chapter 5. In particular, the problem of two-dimensional, compressible, irrotational, inviscid flow around a cylinder is investigated. The variational integral, the computational domain, and the boundary conditions are presented in Sections 5.1.2, 5.1.3, and 5.1.4, respectively. Numerical results are presented for three test cases, simulating transonic flow conditions, in Section 5.1.5. Chapter 6 deals with aspects of generalizing the proposed method. Conclusions are drawn in Section 10.1.

In the second part of the dissertation, a method for optimizing the functionality of the coupling mechanism in a VII code is developed. Some general background on airfoil flow analysis and design is provided in Section 7.1, while the specific VII code is described in Section 7.2. The main idea and the adopted strategy for the optimization are presented in Section 8.1. A technique for reducing the number of optimization variables by means of function approximation concepts is developed in Section 8.2. Convergence criteria are discussed in Section 8.3. Numerical results are reported for different airfoil geometries and transonic flow conditions in Chapter 9. Concluding remarks are addressed in Section 10.2.

Part I

Optimal Incremental Approximation of Differential Equation Solutions.

Chapter 2

Theoretical background and review

A numerical method for optimal sequential function approximation is proposed, implemented, and evaluated in this part of the dissertation. The objective of the development of this numerical technique is the employment of the latter in the solution of differential equations.

The proposed method is based on a variety of techniques borrowed from distinct areas of computational methods. In particular, motivated by similarities observed in adaptive grid optimization techniques, an algorithm originated and employed in artificial neural networks for iterative function approximation has been reformulated with the help of variational principles and the closely related method of weighted residuals.

The objective of this chapter is to review the previously mentioned ideas and illustrate their exact role in, and influence on, the proposed method.

2.1 Function approximation theory

It can be argued that function approximation is the foundation of all numerical methods. Corresponding problems may involve linear or nonlinear, differential or integral operators and vary from data-fitting problems to problems governed by systems of partial differential equations. In engineering applications, the problem of function approximation is addressed by the assembly of interpolation functions that adequately reproduces the dependent variable(s) of interest. For the approximation to be the “best”, according to some appropriate criterion (Appendix A.1), these functions have to be linearly independent members of a complete subset of functions, the so-called basis. For example, the popular finite element method is based on the use of simple, local, low-order polynomials which yield accurate and computationally efficient global function approximation.

In general, a one-dimensional continuous function $f(x)$, defined on a closed interval I, can be approximated by a series expansion of the form

$$f(x) \approx f_n^a(x) = \sum_{i=0}^{n} c_i g_i(x) \ , \tag{2.1}$$

where $g_i(x)$ and c_i are the i-th basis function and expansion coefficient, respectively. The quality of the approximation should be improved, i.e. the error should decrease, as n goes to infinity. For example, in polynomial approximation the approximate function $f_n^a(x)$ is represented by a polynomial $p_n(x) = c_n x^n + c_{n-1}x^{n-1} + \ldots + c_0$ of degree n. In this case, the functions $g_i(x)$ are the monomials x^i (i.e. $x^0 = 1, x^1 = x, x^2, x^3 \ldots$). In fact, the choice of the functions $g_i(x)$ determines the approximation method. For example, when a function $f(x) \in \mathbf{C}^{n+1}[a,b]$ is approximated by a polynomial of degree $\leq n$ on $n+1$ nodes x_i in $[a,b]$, the error of the polynomial interpolation is given by

$$e(x) = f(x) - p_n(x) = \frac{1}{(n+1)!} f^{(n+1)}(\xi_x) \prod_{i=0}^{n} (x - x_i) \tag{2.2}$$

for some $\xi_x \in (a,b)$. For $a = -1$ and $b = 1$, this error can be minimized if the nodes x_i are chosen to be the roots of the Chebyshev polynomial $T_{n+1}(x)$. In fact, Chebyshev polynomials, and other polynomial families, arise as solutions to versions of the Sturm-Liouville eigenvalue problems [27]. Most of these polynomial families possess beneficial properties, e.g. the orthogonality property over certain domains with respect to certain weighting functions. They can be used then in the expansion of Equation (2.1), as the functions $g_i(x)$, to form the generalized polynomial that approximates the function $f(x)$. If the set of functions $\{g_1, g_2, \ldots, g_n\}$ is orthogonal, then Equation (2.1) becomes a Fourier series expansion and the associated Fourier coefficients c_i, $i = 1, \ldots, n$, can be readily and efficiently calculated. Orthogonality can be achieved for any set of basis (linearly independent) functions by menas of some numerical technique, e.g. the popular Gram-Schmidt algorithm.

In the case of multi-dimensional function approximation, Equation (2.1) becomes

$$f(\bar{x}) \approx f_n^a(\bar{x}) = \sum_{i=0}^{n} c_i(\bar{x}_1) g_i(\bar{x}_2) \ , \tag{2.3}$$

where $\bar{x} \in \mathbf{R^d}$, $\bar{x}_1 \in \mathbf{R^n}$, $\bar{x}_2 \in \mathbf{R^m}$, d is the dimension of the function $f(\bar{x})$, and n, m, are integers with $n + m = d$ and $0 \leq n, m \leq d$.

Polynomial interpolation is only one, though popular, approximation method. The associated theory is fairly developed for one-dimensional problems, but the transition to two- and three-dimensional problems is not without problems [72]. An alternative method of approximation is based on splines interpolation and has been successfully employed in the finite element method. Families of splines that form bases suitable for use in series expansions, the so called B-splines, are utilized. They depend on parameters which increase in number with continuity requirements. B-splines are typically defined locally, i.e. their support is finite and covers only a part of the domain. The B_1- and B_3-spline families are introduced in Appendix A.2. The former are preferred for use in the applications presented in this dissertation due to their simplicity, which eventually translates into computational efficiency.

Function approximation by series expansion is an essential part of the proposed method. The basis functions and the expansion coefficients are determined such that they minimize the associated approximation error.

2.2 The method of weighted residuals

One of the most popular methods for the solution of differential equations is the method of weighted residuals. The basic idea is to require the inner product of the residual of a differential equation, with respect to some weighting function, to vanish. Specifically, if an approximate solution function (reproduced by a series expansion with specified functions $g_i(x)$ and unknown expansion coefficients c_i in Equation (2.1)) is introduced into the differential equation, it will yield a nonzero equation residual R. The inner product of the residual with respect to some weighting function w is then defined as the integral of their product over the domain of interest (Appendix A.1), and is set to zero. In this manner, a system of n equations of the form $\langle R, w_i \rangle$, $i = 1, \ldots, n$, is formed and solved for the unknown coefficients. If the set G of functions $\{g_1, g_2, \ldots, g_n\}$ is orthogonal, then the matrix $\mathbf{A}$ of the associated system of algebraic equations $\mathbf{A}\bar{c} = \bar{b}$ is diagonal and the coefficients are calculated with high computational efficiency (see Section 2.1, Fourier series expansion). In general though, a function $g_i \in G$ is orthogonal with respect to most, but not all, functions $g_j \in G$, $j = 1, \ldots, n-1$, $j \neq i$. Accordingly, the matrix $\mathbf{A}$ will be sparse and banded (in the best case tridiagonal). In spectral methods, the matrix $\mathbf{A}$ will be fully, or nearly-fully, occupied. The choice of the weighting functions w_i determines the method of weighted residuals. Among the most popular are the collocation method, the subdomain method, the least-squares method, the method of moments, and the Galerkin methods (Appendix A.3). The method of weighted residuals and its applicability to different classes of problems is extensively discussed in reference [25]. It is important to emphasize that no matter how significant the choice of the weighting (or test) functions is, the importance of the choice of the basis (or trial) functions used in the series expansion should not be underestimated.

As will be demonstrated by means of specific applications, the proposed numerical method for the solution of differential equations via optimal sequential approximation may involve concepts of the method of weighted residuals.

2.3 Variational principle methods

Variational methods in mathematics, physics, and engineering have been extensively studied and developed. A selection of the more general and popular texts include references [63, 50, 76, 67], and [25].

The variational principle method is based on the calculus of variations. Calculus of variations is concerned with variations of functionals, where a functional is defined as some form of correspondence between a function and the set of the real numbers. Such functionals (also called variational principles) may be readily defined for many problems of practical importance in engineering. The approximate solution (given in terms of specified basis functions and their corresponding unknown coefficients) of a given problem is introduced into the associated functional, and the unknown coefficients are determined by making the functional stationary. In this regard, the first variation of some functional $F(u)$, defined by

$$\delta F = \epsilon \lim_{\epsilon \to 0} \frac{F(u + \epsilon w) - F(u)}{\epsilon} \ , \tag{2.4}$$

where u and w are some functions and $\epsilon \in \mathbf{R}$, is required to vanish. The functional $F(u)$ is typically given by a variational integral, and is made stationary by requiring its first variation to vanish; the associated integrand is then set equal to zero, yielding the so-called Euler-Lagrange equation [64]. As mentioned above, for many physical problems of the general form

$$H[u] - f = 0 \ , \tag{2.5}$$

an associated variational principle $F(u)$, whose Euler-Lagrange equation is Equation (2.5), can be defined, and the Raleigh-Ritz method can be employed for the computation of the unknown solution expansion coefficients. In Equation (2.5), $H[\cdot]$ is some differential operator, u is the solution function, and f is some forcing function. It is important to note that the vanishing of the first variation of the functional is only a sufficient condition for the existence of an extremum. To determine whether a functional is extremized, i.e. minimized or maximized, the examination of its second variation $\delta^2 F$ is necessary. In particular, if $\delta^2 F$ can be said to be positive, then the functional possess a minimum; if $\delta^2 F$ can be said to be negative, then the functional possess a maximum, and if the sign of the second variation is indefinite, then the functional merely possess a saddle point.

From a physical point of view, variational principles manifest the tendency of nature to favor the presence of extrema in physical systems. The mathematician Euler expressed his certainty that "*...there is no doubt that all the effects of the world can be derived by the method of maxima and minima from their final causes as well as from their efficient ones*" [25](page 335). However, the derivation of "natural" variational principles for all classes of problems is far from being fully understood and successful. This fact makes the use of "contrived" variational principles necessary. In particular, variational principles are defined based on the properties of the previously introduced differential operator $H[\cdot]$.

If H is linear, i.e. $H = L$, then L is either self-adjoint or not self-adjoint (Appendix A.4). If L is self-adjoint, then a variational principle is readily defined in terms of L, u, and f. If L is not self-adjoint and the operator L can be transformed to a self-adjoint operator $L^{'}$ without changing the associated Euler-Lagrange equation, then a variational principle is defined in terms of $L^{'}$, u, and f. If the nonself-adjoint operator L cannot be transformed to a self-adjoint operator, then a variational principle is defined in terms of L, u, f, L^*, v, and g, where $L^*[v] - g = 0$ is the adjoint problem of $L[u] - f = 0$.

If H is nonlinear, i.e. $H = N$, then the term "self-adjoint" cannot be used, because the adjoint of a nonlinear operator cannot be identical to the operator, and is substituted by the term "self-sufficient" [6]. If N is self-sufficient, variational principles can be defined without extreme effort. However, if N is not self-sufficient, the use of Frechret derivatives (Appendix A.4) cannot be avoided and the tediously derived variational principles are quite complicated and not amenable to computational implementations.

It is interesting to note that there always exists a Galerkin method (a member of the method of weighted residuals) that is equivalent to the variational method; in fact it was Galerkin himself who noted this fact [25](page 223). Due to this significant property, Galerkin methods have been extensively used in finite element methods.

Members of the method of weighted residuals, in particular the Galerkin method, have been primarily developed and used for the solution of differential equations due to their simplicity and computational efficiency. Less frequently, variational methods have been utilized by the Raleigh-Ritz method. For both of these cases, the Euler-Lagrange equation is solved numerically in order to determine the function which satisfies it and the associated initial and/or boundary conditions.

However, it is argued by Finlayson [25](page 310) and Greenspan [29] that the numerical approximation of the function that extremizes a given variational principle is to be preferred to the numerical approximation of the function that satisfies the associated Euler-Lagrange equation.

The proposed method for optimal sequential function approximation for the solution of differential equations is based on the use of variational principles. In particular, the variational principle is adopted as the objective function(al) in the optimization problem(s) to be solved numerically in order to obtain the extremal function that satisfies the associated Euler-Lagrange equation and the boundary conditions. A major advantage of the use of variational principles is that coupled and/or uncoupled equations with general boundary conditions can be concisely and succinctly represented by a single functional.

2.4 Adaptive grid optimization

Numerical methods for the solution of fluid mechanics problems rely on the discretization of the continuous spatial (and/or temporal) domain into a grid of nodal points. The requisite computational cost and the efficiency of the solution method are strongly affected by the type of discretization. For example, slow convergence and numerical instability, resulting from poor condition numbers of the associated discretization matrices, have been reported in the literature for uniform grids. Therefore, optimization in computational mechanics has focused primarily on the grid selection for a given problem.

Jensen introduced a nodal density function based on local discretization error estimates as a criterion for optimal node distribution [42], while Oliveira showed that optimal node distribution provides minimal total potential energy of the system [69]. In fact, it was proven that such distributed nodes lie along isoenergetic contours.

Felippa solved the direct and computationally expensive grid optimization problem for a given finite element configuration, and then relocated the nodes according to a total energy criterion [22, 23]. This process was repeated iteratively until convergence to an optimal grid was achieved. His approach was limited to problems governed by self-adjoint, positive definite, differential operators. Moreover, unavailable exact derivatives and/or inaccurate derivative approximations dramatically decreased the efficiency of the method. Results were reported for extremely coarse two-dimensional grids (8 to 36 nodes), due to the complexity (constraints) and high dimensionality of the associated optimization problem.

Diaz, Kikuchi, and Taylor classified the optimal nodes distribution as a relocation technique, or r-method, and included it among the well known h-, p-, and h-p adaptive methods [17]. In h-methods, the grid spacing h is reduced; additional interpolation functions are introduced to improve the quality of the solution. In p-methods, the order of the interpolation functions is increased, while h-p methods combine the introduction of additional functions with the increase of the order of the existing and/or introduced functions. In the r-method, the location of the nodes of the grid, i.e. the location of the functions, is adjusted for higher accuracy. The grid optimization algorithm of Diaz et al. was based on the minimization of an estimated upper bound on the potential energy of the finite element solution. The associated objective function involved a global interpolation estimate (given by simple summation of local element interpolation error estimates) which serves as an upper bound on the discretization error.

Recently developed adaptive methods of computational mechanics substitute the principle of the system potential energy minimum with the weaker criterion of the homogeneous distribution of the local approximation error over the elements [14, 18, 16].

The common characteristic of adaptive grid techniques is that a grid is generated and a solution is found. Based on the solution, the grid is adapted and regenerated in order to repeat the described process. Multigrid methods share this a-posteriori error estimation philosophy. It is well-known that most of the computational time required by codes that utilize adaptive or multigrid techniques is spent on the necessary grid generation [68, 38, 21]. According to Oden [68], the major challenges to adaptive methods in computational fluid dynamics include 1) the use of unstructured meshes and therefore elaborate and complicated data structures, 2) the necessity of explicit or iterative solution techniques due to the poor performance of direct solvers on dynamically evolving unstructured meshes, 3) stability issues of the associated numerical schemes stemming from the continuous changes in the data structures and polynomial orders, and 4) the computational overhead of the error estimation and the adaptation process.

The method presented in this dissertation features the advantages of adaptive methods without sharing their disadvantages. Specifically, while the method is adaptive in an iterative manner and is based on the optimization of some appropriate criterion, the dimensionality of the associated optimization problem is kept low, derivatives are not required, and most importantly, a grid is not built in the traditional sense. Unstructured mesh data structures are not required for the realization of the method due to the use of simple tensor products of low-order basis functions. Moreover, computationally expensive matrix-assembling algorithms and solvers of systems of equations are avoided. Variational principles and weighted residuals concepts are utilized to define well-posed optimization problems; numerical stability issues are limited to the nonlinear optimization process. Most significantly, the grid, defined by the location of the optimally selected basis functions, evolves with the solution and is not based on a-posteriori error estimates; the computational overhead of the a-posteriori error estimation and respective regridding is eliminated. Lastly, the parallel processing of the optimization problem, defined by the proposed algorithm, dramatically increases the efficiency of the proposed method.

2.5 Artificial neural networks

Any approximation algorithm that uses a combination of basis functions can be mapped into a graph-directed representation called a network. In this regard, function approximation in the framework of neural network computations has been primarily based on the results of Cybenko [10] and Hornik, Stinchcombe, and White [40], who showed that any continuous function on a compact subset of $\mathbf{R}^d$ can be approximated arbitrarily well by a linear combination of simple one-dimensional functions $\{\phi_i(\cdot) : \mathbf{R}^{d+m} \to \mathbf{R}\}$

$$f \approx f_n(\bar{x}) = c_0 + \sum_{i=1}^{n} c_i \phi_i(y(\bar{x}, \bar{p}_i)) \ , \tag{2.6}$$

where $\bar{x} \in \mathbf{R}^d$, $\bar{p}_i \in \mathbf{R}^m$, and y, $c_i \in \mathbf{R}$, represent the independent variables, function parameters, function argument, and linear combination coefficients, respectively. The appropriate linear and nonlinear network parameters in Equation (2.6) are traditionally selected by solving a non-linear optimization problem with the objective function given by the mean square error over some domain Ω

$$\epsilon^2 \equiv \int_\Omega (f - f_n)^2 d\bar{x} \ . \tag{2.7}$$

In the neural network literature, the numerical minimization of the error in Equation (2.7) by the steepest descent method is known as the backpropagation algorithm [77]. More sophisticated optimization methods, including the conjugate gradient and the Levenberg-Marquardt methods, have been also used for neural network training. However, it has been found that even these advanced optimization methods are prone to poor convergence [78]. Clearly, the training algorithms must address a multi-dimensional optimization problem with non-linear dependence on the network parameters $\bar{p}_i$.

Barron showed that the error in Equation (2.7) cannot be smaller than $O(n^{-2/d})$ if the functions ϕ_i are fixed, a phenomenon which he referred to as "the curse of dimensionality" [4]. Therefore, he proposed, based on the theoretical work of Jones [45, 46], the following iterative method for sequential approximation

$$f_n(\bar{x}) = \alpha_n f_{n-1}(\bar{x}) + c_n \phi(y(\bar{x}, \bar{p}_n)) \ , \tag{2.8}$$

where $\bar{p}_n$, c_n, and α_n are optimally selected at each iteration of the algorithm. As a result, the high-dimensional optimization problem associated with neural network training is reduced to a series of simpler low-dimensional problems. A general principle of statistics was utilized to show that the upper bound of the error ϵ is of the order $C/\sqrt{n}$, where C is a positive constant.

The functions ϕ_i, introduced above, have traditionally been sigmoidals in neural network computations, i.e. functions with the properties

$$\lim_{z \to -\infty} \phi_i(z) = 0 \quad \text{and} \quad \lim_{z \to \infty} \phi_i(z) = 1 \ .$$

Alternatively, Orr introduced radial basis functions in his forward selection method of sequential network training [70]. Radial basis functions are of the form $\phi(\bar{x}, \bar{c}, r)$, where $\bar{x}, \bar{c} \in \mathbf{R}^d$ and $r \in \mathbf{R}$ are the independent variables, the center point, and the radius, respectively. Their value decreases (if they are bounded) or increases (if they are not bounded) monotonically with increasing distance from the central point. Orr's method is in essence a method of incremental function approximation. At each iteration of the algorithm, an additional basis function, which produces the largest reduction of error in the previous iteration, is chosen from a given set of functions and added to the approximation. However, the forward selection training method can be inefficient in that it may require significant computational resources when the set of trial functions is large.

A similar principle is utilized in Platt's resource allocating networks (RAN) [71]. Whenever an unusual pattern is presented to the network in an on- or off-line network training procedure, a new computational unit (according to the neural network terminology), is allocated. The computational units respond to local regions of the input space.

The concept of incremental (or sequential) approximation is one of the major features of the proposed method. Following the main idea of Barron's algorithm [4], optimal basis functions with local support are added to the approximating series expansion. A broad class of functions can be employed. B-splines are popular and resemble radial basis functions, but Gaussian functions (see Appendix A.2) and families of locally defined orthogonal polynomials can also be used.

Chapter 3

The proposed method

The proposed method of optimal incremental function approximation for the solution of differential equations is presented in this chapter. The development of this method was motivated by the recent advances in parallel optimization techniques and by numerical concepts reviewed in the previous chapter. The main features of the proposed method are a) the iterative nature of the proposed algorithm for the approximation and b) the solution of a nonlinear optimization problem at each stage of the algorithm. The objective of the development of the method was the employment of the latter in the solution of differential equations.

3.1 The proposed algorithm

The developed algorithm can be summarized as follows [62]. Given a problem $Q[u(\bar{x})] = 0$, governed by one (or more) algebraic or differential equation(s), the dependent variable(s) is (are) approximated by a basis expansion of the form given in Equation (2.1). An incremental set of basis functions is built sequentially to improve the expansion-based approximation. At each stage of the algorithm, the parameters of the new basis function and the corresponding expansion coefficient are optimally determined by solving a low-dimensional, nonlinear optimization problem. Assuming that the algorithm has reached step i, the approximating function is given by

$$u(\bar{x}) \approx u_i^a(\bar{x}) = \sum_{j=1}^{i} c_j \Phi_j(\bar{x}, \bar{p}_j) = c_i \Phi_i(\bar{x}, \bar{p}_i) + u_{i-1}^a(\bar{x}) \ . \qquad (3.1)$$

The coefficient c_i and the parameters $\bar{p}_i$ of the associated basis function $\Phi_i(\bar{x}, \bar{p}_i)$ are optimally computed according to an appropriate criterion, i.e. by solving a nonlinear optimization problem, while the coefficients c_j and the basis functions $\Phi_j(\bar{x}, \bar{p}_j)$, $j = 1, \ldots, i-1$, are held fixed.

If the given problem is linear, the nonlinear optimization problem can be decomposed into two lower-dimensional problems; one for the computation of the parameters $\bar{p}_i$ and one for the calculation of the coefficient c_i. In this manner, computational efficiency is enhanced.

A broad class of functions can be used, e.g. low-order polynomials, B-splines, and radial basis functions. It is emphasized that there are no restrictions on the distribution of the interpolation functions over the domain of interest; overlapping is possible.

Orthogonality or orthonormality properties can be attained, if desired or required, by means of the Gram-Schmidt algorithm, either during the basis-building process or at its completion. The latter requires the recalculation of the coefficients and reduces the computational efficiency of the proposed method since a system of equations has to be solved.

The algorithm can be initialized with either an empty set or an arbitrary number of predetermined functions and coefficients; the second option enables the use of coarse grid solutions obtained, for example, from a previous finite element analysis of the given problem. The sequential algorithm is halted when the relative difference between the previous and current approximations falls below a user-defined tolerance level.

The details of the method will be presented and computational issues will be addressed with the help of examples of increasing complexity. Results for specific numerical examples (with emphasis on fluid dynamics applications) will be presented and discussed, and the overall performance will be evaluated based on parameter studies.

3.2 The parallel direct search optimization technique

The computational efficiency of the method depends on the cost of the nonlinear optimization problem solution. The dimensionality of the nonlinear problem is kept low due to the iterative nature of the algorithm. Moreover, the parallel direct search (PDS) optimization technique is used to solve the optimization problem efficiently.

PDS is a software package that consists of a collection of Fortran routines for solving both unconstrained and constrained nonlinear optimization problems using direct search methods. Direct search methods have the advantage of requiring information about the objective function only. Derivatives do not need to be calculated, and the condition for convergence is that the objective function has to be continuously differentiable. The user has to provide objective function and constraint(s) evaluation routines and an input file. This fact makes PDS quite easy to use. In this manner, the combined user effort and computer time may often be smaller than that required for gradient-based methods since the latter are theoretically more efficient but practically more complicated to implement. In addition, ingredients essential to gradient-based methods, i.e. derivatives, may be inaccurate, noisy, not available, or quite expensive to compute. PDS can achieve relatively high performance rates because of the parallel execution of the necessary objective function evaluations. Exceptional scalability is achieved by fully exploiting additional processors, yielding almost linear speed-up. The PDS optimization technique and its theoretical background are presented in references [15, 80, 52], and [81]. Reference [9] reports the application of PDS to an optimal design problem.

The working scheme of PDS is as follows. Given an initial guess for the optimization variables vector of length n, PDS evaluates the objective function at d vertices of a simplex defined on the hyperplane of dimension n. The "best" vertex is determined and compared to the previous iterate. If the difference lies within a user-defined tolerance tol, the search is terminated; otherwise, a new iteration is performed with the current iterate as the initial vertex. The search size is controlled by the user by the input of the parameters d and tol. For unconstrained problems the restriction $d \geq 2n$ applies, while for constrained problems $d \geq n^2+n$. However, d should be typically chosen large in order to ensure a fine-grained search.

PDS works best when the number of the optimization variables is kept small, i.e. when the ratio of the number of necessary function evaluations to the number of available processors is low. This requirement is satisfied by the proposed method for incremental function approximation. Computational implementation issues will be addressed in the following sections and in Appendix A.5.

3.3 The iterative character of the proposed algorithm

This section intends to demonstrate how the iterative nature of the proposed algorithm relates the method of optimal incremental approximation to the Gauss-Seidel method for the solution of a linear system of equations.

There are two types of numerical methods for the solution of a linear system of algebraic equations of the form $\mathbf{A}\bar{z} = \bar{b}$, where $\mathbf{A} \in \mathbf{R}^{n\times n}$ and $\bar{b} \in \mathbf{R}^n$ are known and $\bar{z} \in \mathbf{R}^n$ is the unknown solution vector; direct methods yield the exact solution, as far as computer arithmetic accuracy allows, while iterative methods approximate the exact solution up to a user specified level of accuracy. Iterative methods are computationally more efficient than direct methods for large systems (large n), especially when the matrix $\mathbf{A}$ is sparse, but their performance, i.e. convergence rate, strongly depends on the structure and properties of the matrix $\mathbf{A}$.

In general, iterative methods are more stable, as they will usually dampen errors with increasing number of iterations [48]. Various methods and algorithms have been developed to enhance and/or exploit the properties of the matrix $\mathbf{A}$ [3]. Iterative methods are classified as either stationary or nonstationary. In stationary iterative methods (Jacobi, Gauss-Seidel, SOR, etc.), the matrix $\mathbf{A}$ remains invariant, while in nonstationary methods (e.g. conjugate gradients (CG), GMRES, etc.) $\mathbf{A}$ varies with each iteration. The objective of the latter is to accelerate the convergence to a sufficiently accurate solution.

The Gauss-Seidel method belongs to the class of stationary iterative methods. Diagonal dominance is required for convergence, and a low spectral radius is desired for a satisfactory rate of convergence (Appendix A.6). In the Gauss-Seidel method, the matrix $\mathbf{A}$ is decomposed into its lower triangular part, which contains the diagonal of the original matrix, and its strictly upper triangular part, i.e. an upper triangular matrix with a zero diagonal,

$$\mathbf{A}\bar{z} = \bar{b} \Leftrightarrow (\mathbf{L} + \mathbf{D})\bar{z} + \mathbf{U}\bar{z} = \bar{b}$$

and the iterative algorithm is

$$(\mathbf{L} + \mathbf{D})\bar{z}^{k+1} = \bar{b} - \mathbf{U}\bar{z}^{k} \ . \tag{3.2}$$

If $\mathbf{A}$ is diagonal dominant, the Gauss-Seidel method converges for any starting vector $\bar{z}^0$. Typically, $\bar{z}^0 = \bar{0}$.

Given a system of algebraic equations, e.g. a differential equation discretized by means of the Bubnov-Galerkin method, the use of the Gauss-Seidel method for its solution can be interpreted as following. The first coefficient z_1 of the solution vector $\bar{z}$ is readily calculated by solving the first equation of the system. Notice that the first equation depends only on the first interpolation function Φ_1 since the Bubnov-Galerkin scheme implies that the inner product $\langle R, \Phi_i \rangle$ (where R is the differential equation residual and the weighting function Φ_i is identical with the trial function Φ_i) vanishes $\forall i$, assuming that $\bar{z}^0 = \bar{0}$. The second equation is then solved for z_2 with known z_1, employing the functions Φ_1 and Φ_2. This process is repeated for all equations; it is clear that at step i all components z_j with $j < i$ are known and the employed functions are the functions Φ_j with $j \leq i$. After all equations have been solved, a new outer iteration begins to improve the accuracy of the solution vector.

The proposed algorithm of Section 3.1 resembles the Gauss-Seidel method in that it proceeds in the same manner, with the exception that at each step of the algorithm the i-th interpolation function is optimized instead of being fixed and preselected. It can be argued that the analogy of the matrix $(\mathbf{L} + \mathbf{D})$ is not predefined and constant, but built in an adaptive way. It therefore assimilates a "nonstationary" Gauss-Seidel method. Consequently, the convergence rate should be improved, compared to that of the stationary Gauss-Seidel method. Moreover, the proposed algorithm is not just an improved technique for the solution of linear systems of algebraic equations. It is an adaptive, optimal approximation algorithm that imitates grid optimization schemes and can be also employed in nonlinear problems, as will be demonstrated in Chapter 5.

3.4 Function approximation

In this section, the implementation of the proposed method for two simple nonlinear function approximation problems in one and two dimensions will be demonstrated.

3.4.1 Implementation

At the i-th stage of the proposed sequential approximation algorithm, some function $u(\bar{x})$ is approximated by Equation (3.1). Let $e_i = u(\bar{x}) - u_i^a(\bar{x})$. The squared error between the function and its approximation can be defined as

$$\epsilon \equiv \|e_i\|^2 = \langle e_i, e_i \rangle = \int_\Omega e_i^2 d\bar{x} \tag{3.3}$$

and can be rewritten as

$$\begin{aligned}
&\int_\Omega (u(\bar{x}) - u_i^a(\bar{x}))^2 d\bar{x} = \int_\Omega (u(\bar{x}) - u_{i-1}^a(\bar{x}) - c_i\Phi_i(\bar{x}))^2 d\bar{x} \quad = \\
&\int_\Omega (e_{i-1} - c_i\Phi_i(\bar{x}, \bar{p}_i))^2 d\bar{x} = \int_\Omega e_{i-1}^2 d\bar{x} + c_i^2 \int_\Omega (\Phi_i(\bar{x}, \bar{p}_i))^2 d\bar{x} \quad - \\
&2c_i \int_\Omega e_{i-1}\Phi_i(\bar{x}, \bar{p}_i) d\bar{x} \quad = \\
&\|e_{i-1}\|^2 + c_i^2 \|\Phi_i(\bar{x}, \bar{p}_i)\|^2 - 2c_i\langle e_{i-1}, \Phi_i(\bar{x}, \bar{p}_i)\rangle \ .
\end{aligned} \tag{3.4}$$

The linear coefficient c_i can be determined by differentiating ϵ with respect to c_i and requiring the derivative to vanish

$$\frac{\partial \epsilon}{\partial c_i} = 0 = 2c_i\|\Phi_i(\bar{x}, \bar{p}_i)\|^2 - 2\langle e_{i-1}, \Phi_i(\bar{x}, \bar{p}_i)\rangle \ , \tag{3.5}$$

which yields

$$c_i = \frac{\langle e_{i-1}, \Phi_i(\bar{x}, \bar{p}_i)\rangle}{\|\Phi_i(\bar{x}, \bar{p}_i)\|^2} \ . \tag{3.6}$$

Note that the procedure for evaluating the coefficient c_i is equivalent to the Bubnov-Galerkin scheme since it provides for the orthogonality of the error e_i to the weighting function $\Phi_i(\bar{x}, \bar{p}_i)$. Indeed, Equation (3.5) shows that

$$\int_\Omega e_{i-1}\Phi_i(\bar{x}, \bar{p}_i)d\bar{x} - c_i \int_\Omega (\Phi_i(\bar{x}, \bar{p}_i))^2 d\bar{x} = \langle e_i, \Phi_i(\bar{x}, \bar{p}_i)\rangle = 0 \ . \tag{3.7}$$

Substituting Equation (3.6) into Equation (3.4) yields

$$\epsilon = \|e_{i-1}\|^2 - \frac{\langle e_{i-1}, \Phi_i(\bar{x}, \bar{p}_i)\rangle^2}{\|\Phi_i(\bar{x}, \bar{p}_i)\|^2} \ . \tag{3.8}$$

To minimize the error, the parameters of the i-th basis function must be selected to maximize the last term of equation (3.8). This nonlinear optimization problem can be solved efficiently by PDS. Clearly, the proposed method resembles some of the conventional adaptive numerical techniques from computational mechanics in that the new basis function is selected to provide the largest projection on the error of the preceding iteration. Specifically, the new basis function is positioned at the location of the largest previous error. Moreover, the basis function can be also viewed as a weighting function; the proposed method resembles the method of weighted residuals.

3.4.2 Numerical examples

The proposed method has been applied to two nonlinear function approximation examples. In particular the one-dimensional function $u(\bar{x}) = x^2$ $(\bar{x} = x)$ and the two-dimensional function $u(\bar{x}) = x^2 + y^2$ $(\bar{x} = (x, y)^{\mathrm{T}})$ have been approximated. Piecewise linear "hat" (B_1-splines) basis functions were used. B_1-splines are defined as

$$\Phi(x) = \begin{cases} \frac{x-(x_M-\Delta x_l)}{\Delta x_l} & \text{if} \quad x_M - \Delta x_l \leq x \leq x_M \\ \frac{(x_M+\Delta x_r)-x}{\Delta x_r} & \text{if} \quad x_M \leq x \leq x_M + \Delta x_r \\ 0 & \text{otherwise,} \end{cases} \tag{3.9}$$

where the parameters x_M, Δx_l, and Δx_r denote the location of the center of the function, its width to the left, and its width to the right, respectively. The simple product of two one-dimensional functions has been used in the two-dimensional case, i.e. $\Phi(x, y) = \Phi(x)\Phi(y)$. The optimization variables are given by

$$\bar{p}_i = (x_M, \Delta x_l, \Delta x_r)_i^{\mathrm{T}} \tag{3.10}$$

for the one-dimensional case and by

$$\bar{p}_i = (\bar{p}_x, \bar{p}_y)_i^{\mathrm{T}} = (x_M, \Delta x_l, \Delta x_r, y_M, \Delta y_l, \Delta y_r)_i^{\mathrm{T}} \tag{3.11}$$

for the two-dimensional case. The initial values for the optimization variables, required by PDS, are chosen so that each new basis function is initially centered in the domain of interest and encompasses it. As an alternative, the initial guess can be chosen such that the center of the basis function is located at the position of the maximum error which has been calculated in the previous iteration in order to improve the efficiency of the PDS technique. However, this does not affect the accuracy of the results since PDS performs a fine-grained search throughout the domain.

The approximations obtained by the proposed method are satisfactorily accurate. The RMS error, defined as

$$\mathrm{RMS}_M^N = \sqrt{\frac{\sum_{i=1}^{M}(u(\bar{x}_i) - u_N^a(\bar{x}_i))^2}{M}}\ , \tag{3.12}$$

where M is some sufficient number of uniformly distributed trial points, is plotted versus the number of optimized basis functions N in Figure 3.1 for both the one-dimensional and the two-dimensional problems using $M = 11$ and $M = 101$ uniformly distributed evaluation points, respectively. Linear and quadratic conver-

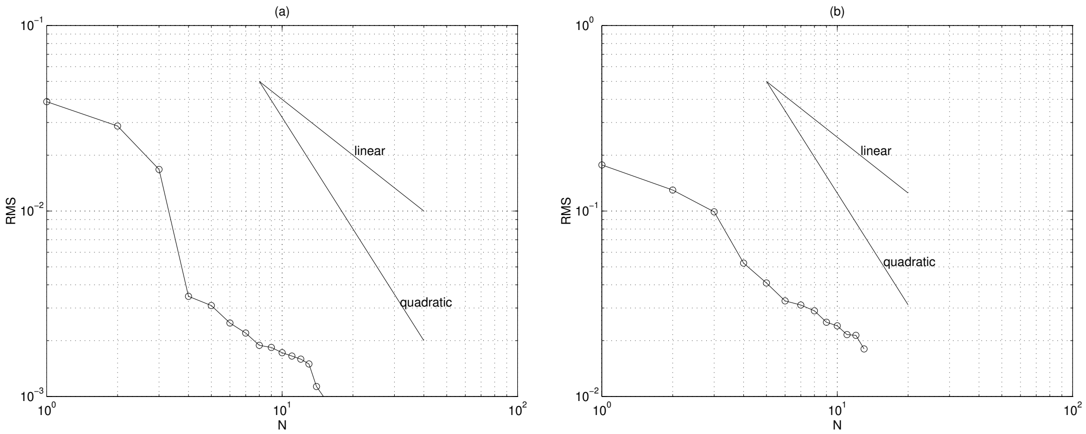

Figure 3.1 Convergence rates for the function approximation problems: (a) one-dimensional case and (b) two-dimensional case.

gence lines are included to illustrate the satisfactory convergence rate. Convergence rates with respect to two different numbers of used basis functions N_1 and N_2 are defined as linear if

$$\mathrm{RMS}_M^{N_2} = \frac{\mathrm{RMS}_M^{N_1}}{N_2/N_1} \, , \tag{3.13}$$

and quadratic if

$$\mathrm{RMS}_M^{N_2} = \frac{\mathrm{RMS}_M^{N_1}}{(N_2/N_1)^2} \, , \tag{3.14}$$

in order to be compatible with the convergence terminology used in the finite element method, where the grid spacing h is used as the convergence metric (for example h^2-convergence etc.). If Equations 3.13 and 3.14 become inequalities with larger right sides, the convergence rates are defined as superlinear and superquadratic, respectively. When finite element methods are applied on uniform grids, the grid spacing h and the number of used basis functions N are directly related through the formulas $h = 1/(N-1)$, for one-dimensional problems, and $h_x h_y = (\frac{1}{N_x-1})(\frac{1}{N_y-1})$, for two-dimensional problems.

Chapter 4

Linear differential equations

4.1 Linear self-adjoint problems

Consider the general boundary value problem with homogeneous Dirichlet boundary conditions

$$L[u(\bar{x})] = f(\bar{x}) \ \text{ in } \Omega, \quad u(\bar{x}) = 0 \ \text{ on } \partial\Omega \ , \tag{4.1}$$

where $L[\cdot]$ is a linear, self-adjoint differential operator. Homogeneous Dirichlet boundary conditions are assumed for the sake of simplicity. The extension to problems with general boundary conditions is possible but requires additional assumptions on the trial functions to be used and/or the modification and adjustment of the associated variational principle, as will be demonstrated in Sections 4.1.3 and 4.2.

For linear, self-adjoint problems associated with homogeneous Dirichlet boundary conditions, the variational principle associated with Equation (4.1) is given by the functional

$$E \equiv \frac{1}{2} l(u(\bar{x}), u(\bar{x})) - \langle f(\bar{x}), u(\bar{x}) \rangle \ , \tag{4.2}$$

where $l(\cdot, \cdot)$ denotes the bilinear symmetric differential form associated with the operator L, i.e. $l(\cdot, \cdot) = \langle L[\cdot], \cdot \rangle = \langle \cdot, L[\cdot] \rangle$, and the expression $\langle \cdot, \cdot \rangle$ denotes an inner product (Appendix A.1). Typically, this functional is called the energy functional, and the associated norm $\sqrt{l(\cdot,\cdot)}$ is called the energy norm [49].

If the functional E is numerically minimized by a function $u_i^a(\bar{x})$, then $u_i^a(\bar{x})$ automatically satisfies the associated Euler-Lagrange Equation (4.1); moreover, if L is positive definite, the solution is unique [63](pages 74-75). However, a function $u_i^a(\bar{x})$ that satisfies Equation (4.1) does not necessarily minimize the functional E. It is due to this fact that variational principles are employed by the proposed method. In this regard, the approximate function given by Equation (3.1), is introduced into the functional E to yield

$$E \equiv \frac{1}{2} l(u_i^a(\bar{x}), u_i^a(\bar{x})) - \langle u_i^a(\bar{x}), f(\bar{x}) \rangle \ . \tag{4.3}$$

This functional may be used as the objective function to be minimized in an optimization problem in order to determine the optimal parameters $\bar{p}_i$ and the associated coefficient c_i. However, in this case, as mentioned in Section 3.1, the properties of the differential operator can be exploited to simplify the optimization problem. Since the differential operator is self-adjoint, the boundary conditions are homogeneous and of the Dirichlet type, and $\langle u_i^a(\bar{x}), f(\bar{x}))\rangle = l(u_i^a(\bar{x}), u(\bar{x}))$, Equation (4.3) can be rewritten as

$$E = \frac{1}{2} l(u(\bar{x}) - u_i^a(\bar{x}), u(\bar{x}) - u_i^a(\bar{x})) - \frac{1}{2} l(u(\bar{x}), u(\bar{x})) \ . \tag{4.4}$$

At the i-th step of the proposed sequential method, the approximate solution is given by Equation (3.1). Again, let $e_i = u(\bar{x}) - u_i^a(\bar{x})$. Then, $e_i = e_{i-1} - c_i \Phi_i(\bar{x}, \bar{p}_i)$, and Equation (4.4) becomes

$$E = \frac{1}{2}l(e_{i-1} - c_i\Phi_i(\bar{x},\bar{p}_i), e_{i-1} - c_i\Phi_i(\bar{x},\bar{p}_i)) - \frac{1}{2}l(u(\bar{x}),u(\bar{x})) \quad = \\ \frac{1}{2}l(e_{i-1},e_{i-1}) + \frac{1}{2}c_i^2 l(\Phi_i(\bar{x},\bar{p}_i),\Phi_i(\bar{x},\bar{p}_i)) - c_i l(e_{i-1},\Phi_i(\bar{x},\bar{p}_i)) \quad - \\ \frac{1}{2}l(u(\bar{x}),u(\bar{x})) \ . \tag{4.5}$$

As in the function approximation case, the coefficient c_i can be determined by equating the partial derivative of E with respect to c_i to zero

$$\frac{\partial E}{\partial c_i} = 0 = c_i l(\Phi_i(\bar{x},\bar{p}_i),\Phi_i(\bar{x},\bar{p}_i)) - l(e_{i-1},\Phi_i(\bar{x},\bar{p}_i)) \ . \tag{4.6}$$

The coefficient c_i can then be calculated from

$$c_i = \frac{l(e_{i-1},\Phi_i(\bar{x},\bar{p}_i))}{l(\Phi_i(\bar{x},\bar{p}_i),\Phi_i(\bar{x},\bar{p}_i))} \tag{4.7}$$

and Equation (4.5) can be rewritten as

$$E = \frac{1}{2}l(e_{i-1},e_{i-1}) - \frac{1}{2}l(u(\bar{x}),u(\bar{x})) - \frac{1}{2}\frac{(l(e_{i-1},\Phi_i(\bar{x},\bar{p}_i)))^2}{l(\Phi_i(\bar{x},\bar{p}_i),\Phi_i(\bar{x},\bar{p}_i))} \ . \tag{4.8}$$

The equivalence of the method to the Bubnov-Galerkin method is, as in Section 3.4.1, readily deduced by examining Equation (4.6):

$$0 = c_i l(\Phi_i(\bar{x},\bar{p}_i),\Phi_i(\bar{x},\bar{p}_i)) - l(e_{i-1},\Phi_i(\bar{x},\bar{p}_i)) = \\ \langle L[c_i\Phi_i(\bar{x},\bar{p}_i)],\Phi_i(\bar{x},\bar{p}_i)\rangle - \langle L[e_{i-1}],\Phi_i(\bar{x},\bar{p}_i)\rangle = \langle R_i,\Phi_i(\bar{x},\bar{p}_i)\rangle \ . \tag{4.9}$$

To improve the accuracy of the approximation, the last term of Equation (4.8) must be maximized by appropriately selecting the parameters $\bar{p}_i$ of the basis function $\Phi_i(\bar{x}, \bar{p}_i)$ since the first two terms of Equation (4.8) are constant with respect to $\bar{p}_i$. Once again, PDS has been used to solve this nonlinear problem efficiently.

4.1.1 Numerical example: The Poisson's equation

There is a number of significant applications in science and engineering that are governed by linear and self-adjoint differential operators. One of the most representative linear, self-adjoint boundary value problems is the Poisson equation

$$-\nabla^2 u(\bar{x}) = f(\bar{x}) \; . \tag{4.10}$$

If $f(\bar{x}) = 0$, the Laplace equation is recovered. The Poisson equation can be used to model a number of physical problems including a broad class of engineering applications like the drum membrane problem, the magnetic field equation, the steady heat conduction, and steady laminar flow through a duct. Moreover, $f(\bar{x})$ can be a linear or a mildly nonlinear function of u, as, for example, in the optimal control problem studied in reference [47] and the problems considered in reference [30].

The Poisson's equation associated with homogeneous Dirichlet boundary conditions has been successfully solved in one and two dimensions by means of the proposed method. The two-dimensional problem models, as mentioned earlier, steady laminar flow through a rectangular duct. The domains of interest were defined as $\Omega = [0, 1]$ and $\Omega = [-1, 1] \times [-1, 1]$, while the forcing functions were given by $f(x) = x^2$ and $f(x, y) = 2\pi$, for the one-dimensional and two-dimensional cases, respectively. Once again, locally defined piecewise linear (B_1-splines) basis functions were used, and the optimization variables vectors were defined as in Section 3.4.2. Integration by parts was necessary due to the presence of the second order differential operator. Numerical integration was based on the popular Gaussian quadrature method, and the basis functions were constrained to satisfy the homogeneous Dirichlet boundary conditions, i.e. they were required to vanish at the boundaries (more details on computational issues are given in Section A.5.1).

The solution obtained for the one-dimensional problem is compared to the exact in Figure 4.1(a). Figure 4.1(b) compares the RMS error convergence rate of the proposed method to that obtained by the Galerkin technique using linear Lagrangian shape functions on a uniform grid. The RMS errors were calculated based on 101 uniformly distributed trial points.

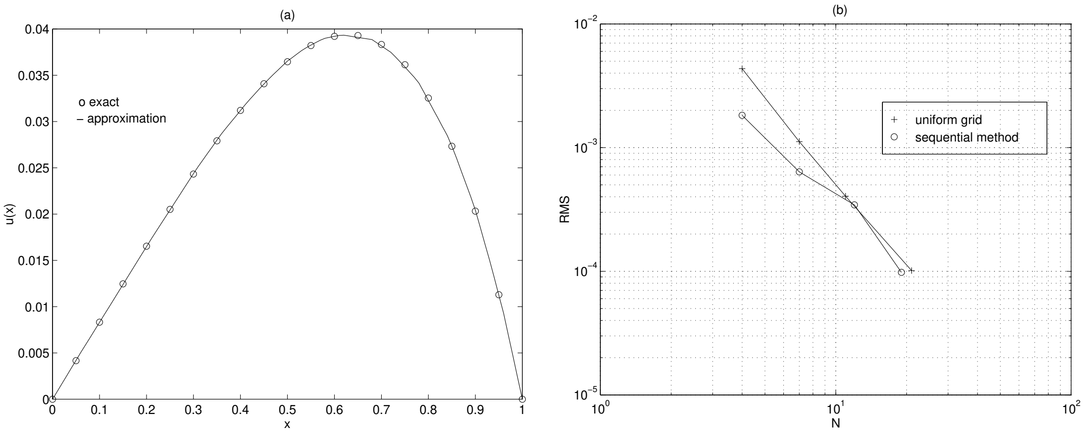

Figure 4.1 One-dimensional Poisson's equation: (a) solution comparison using 19 optimal basis functions and (b) convergence rate comparison.

Figure 4.2 illustrates the difference in the distribution of the basis functions through the domain. Figure 4.2(a) shows the standard uniform distribution of the basis functions when $h = 0.1$ (or $N = 11$), while Figure 4.2(b) shows the optimal distribution of the basis functions as obtained using the optimal incremental approximation method. For the sake of illustration, only the first 11 of the 19 basis functions are plotted.

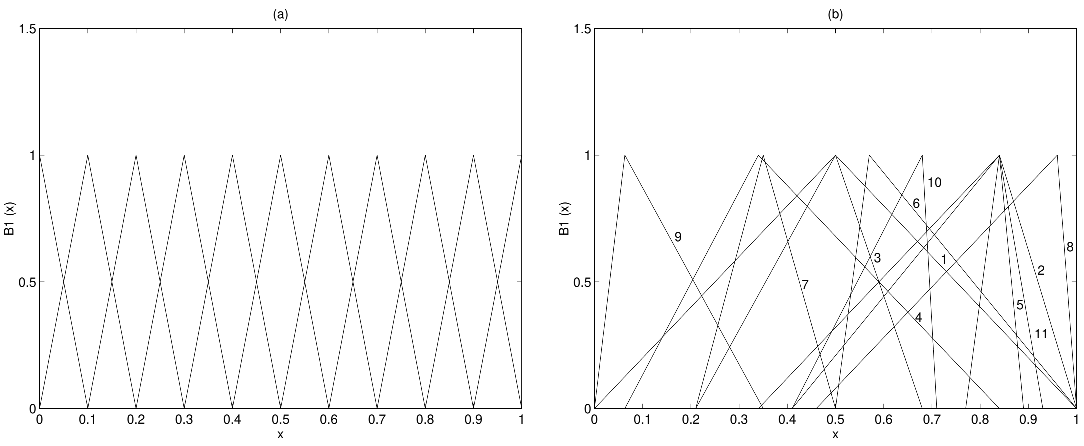

Figure 4.2 Basis functions distributions for the one-dimensional Poisson problem: (a) standard uniform and (b) optimal.

The evolution of the incrementally built solution for the two-dimensional problem is depicted on the left of Figure 4.3, while the evolution of the associated squared error distribution is illustrated on the right. The respective evolutions obtained by the Galerkin finite element method on a uniform grid are presented in Figure 4.4. It can be readily concluded by direct comparison that the proposed method is more efficient.

The RMS error convergence rate, compared to that obtained by the Galerkin technique using uniformly distributed bilinear Lagrangian shape functions, is presented in Figure 4.5. The RMS errors were calculated based on 961 uniformly distributed trial points. The "kinks" in the convergence curves of Figure 4.1(b) and Figure 4.5 come from a combination of two sources. Firstly, PDS optimizes by direct search. As such it can only evaluate a finite combination of parameters which results in a finite interval of uncertainty. Secondly, the algorithm is not

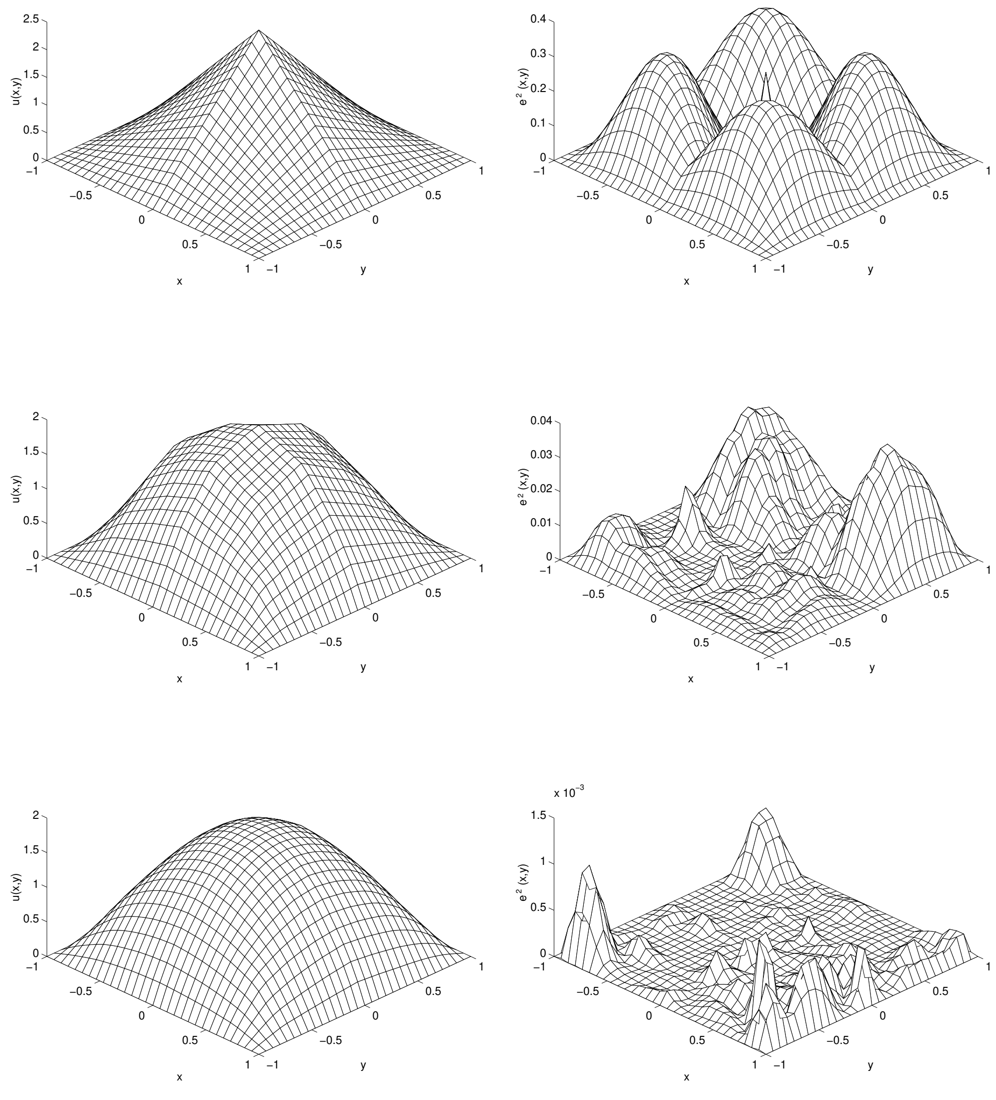

Figure 4.3 Two-dimensional Poisson's equation: (left) proposed method solutions and (right) associated squared error distributions for (top to bottom) $N = 1$, 10, and 100 optimal basis functions.

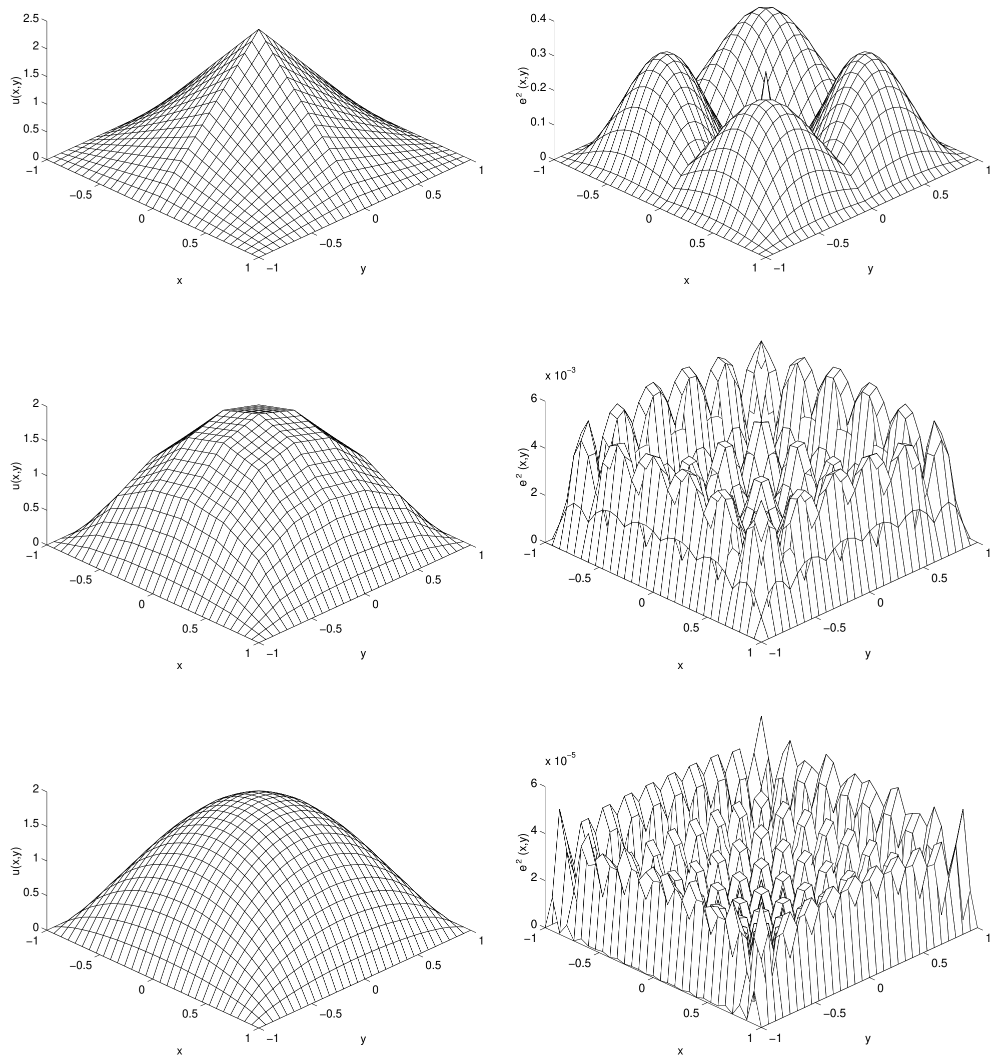

Figure 4.4 Two-dimensional Poisson's equation: (left) uniform grid finite element solutions and (right) associated squared error distributions for (top to bottom) $N = 9$, 49, and 441 bilinear Lagrangian shape functions.

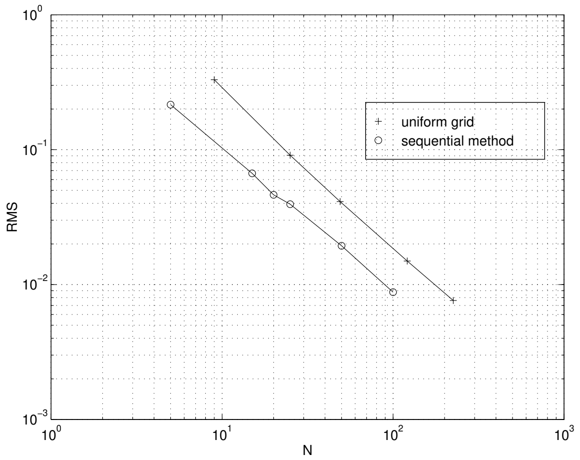

Figure 4.5 Two-dimensional Poisson's equation: convergence rate comparison.

designed to explicitly minimize the RMS error; the algorithm effectively minimizes Equation (4.8). However, this equation acts as an upper bound on the RMS, so though successive RMS values can increase they cannot cause the value of E to exceed its previous value. This can result in a sawtooth or step-like oscillation in the convergence curves.

4.1.2 Computational cost

Two PDS input parameters control the number of required PDS iterations and function and constraint evaluations; the number of pattern points d for the search directions and the user-selected tolerance tol, which constitutes the convergence criterion of the PDS algorithm. PDS requires $d \geq 2m$ pattern points for unconstrained problems and $d \geq m^2 + m$ pattern points for constrained problems, where m is the number of optimization variables. Numerical experimentations have shown that d should be chosen to be larger than $2m$ or $m^2 + m$, respectively, in order to ensure a detailed search and, as far as this is possible, a global extremum.

The number of pattern points for the search directions should be relatively small if the objective function is convex, and large if the objective function is suspected to be nearly concave. Typically, if a small d is chosen, a small tolerance should also be chosen since the search interval is relatively coarse. Similarly, if a large d is selected for the same number of design variables, a larger tolerance should be chosen to ensure computational efficiency.

Table 4.1 summarizes the performance characteristics of PDS for the two-dimensional numerical example for both a moderate and a large number of pattern points d. It is clear that a moderate number of pattern (or scheme) points ($d = 100$)

Table 4.1 Amount of computational work required by PDS for the two-dimensional Poisson problem.

Algorithm step	# of PDS iterations	Total # of function eval.	Total # of constraint eval.
	$d = 100$	$tol = 10^{-6}$	
1	19	459	1812
30	39	2303	3818
60	39	1535	3588
90	39	3431	3812
	$d = 1000$	$tol = 10^{-4}$	
1	7	1066	6097
30	13	3156	11632
60	15	12718	13996
90	8	909	6574

will require more PDS iterations, due to the smaller tolerance (10^{-6}), but fewer function and constraint evaluations, while a large number of points ($d = 1000$) will require fewer PDS iterations, due to the larger tolerance (10^{-4}), but more function and constraint evaluations. Table 4.1 shows that the numbers of PDS iterations and function and constraint evaluations fluctuate but remain "bounded" and are independent from the number of the algorithm steps, for both small and large values of d. Moreover, these evaluations can always be computed faster with increasing number of available parallel processors. The only cost linked to the increase in algorithm steps stems from the increasing number of numerical integrations as more functions are added to the series expansion; this cost is mitigated by performing the necessary numerical integrations in parallel.

4.1.3 General boundary conditions

In this section, the linear, self-adjoint problem $L[u(\bar{x})] = f(\bar{x})$, associated with general boundary conditions $B[u(\bar{x})] = g(\bar{x})$, will be considered. The boundary operator B may represent a mixture of Dirichlet and Neumann conditions.

Assuming that an arbitrarily chosen function $s(x)$ satisfies the non-homogeneous boundary conditions but not necessarily the differential equation, then the function $q(x) = u(x) - s(x)$ will satisfy the homogeneous boundary conditions, provided that $u(x)$ satisfies the non-homogeneous boundary conditions; since $B[\cdot]$ is the general linear boundary operator, then $B[q] = B[u] - B[s] = 0$. If $q^{'} = u_{ex} - s$, where u_{ex} is the exact solution of the problem, then, since L is linear

$$L[q^{'}] = L[u_{ex}] - L[s] = f - L[s] = f^{'} \ . \tag{4.11}$$

In this manner, the non-homogeneous boundary value problem has been transformed to a homogeneous boundary value problem. The variational principle is then

$$E(q^{'}) \equiv \frac{1}{2}\langle q, L[q]\rangle - \langle q, f^{'}\rangle \tag{4.12}$$

and can be reformulated in terms of the unknown function u

$$E(u) = \int_\Omega (uL[u] - 2uf + uL[s] - sL[u])d\bar{x} + \int_\Omega (2sf - sL[s])d\bar{x} \ , \qquad (4.13)$$

where the second integral can be omitted due to its independence with respect to u. Since the differential operator is self-adjoint, integration by parts would force the last two terms of the first integral to cancel each other, but the presence of non-homogeneous boundary conditions will result in the appearance of non-zero boundary terms. The same is true for the integration by parts of the first term of the first integral, i.e. the boundary terms will only be zero for homogeneous boundary conditions. It is clear that the trial functions have to satisfy the Dirichlet boundary conditions only; the Neumann boundary conditions are automatically satisfied by the variational principle.

The issues addressed above will be illustrated by means of the one- and two-dimensional problems presented in Section 4.1.1 which are to be solved now for general boundary conditions. In particular, the one-dimensional problem has been solved for the boundary conditions

$$u(0) = 0 \quad \text{and} \quad \left[\frac{du}{dx}\right]_{x=1} = 1$$

and the two-dimensional problem has been solved for the boundary conditions

$$u = 0 \quad \text{on} \quad x = -1 \quad \text{and} \quad y = 1 \ ,$$
$$\frac{\partial u}{\partial x} = 0 \quad \text{on} \quad x = 1 \ , \quad \text{and} \quad \frac{\partial u}{\partial y} = 0 \quad \text{on} \quad y = -1 \ .$$

For the sake of simplicity, the derivation of the final variational principle, to be utilized as the objective function for the nonlinear optimization problem to be solved by PDS, will be presented for the one-dimensional case only. The forcing function is assumed here to be constant; specifically, $f = 1$. The variational principle then reads as

$$\begin{aligned}
E(u) = \int_0^1 u\frac{d^2u}{dx^2}dx - 2\int_0^1 udx + \int_0^1 u\frac{d^2s}{dx^2}dx - \int_0^1 s\frac{d^2u}{dx^2}dx = \\
-\int_0^1 \frac{du}{dx}\frac{du}{dx}dx + \left[u\frac{du}{dx}\right]_{x=0}^{x=1} - 2\int_0^1 udx - \int_0^1 \frac{du}{dx}\frac{ds}{dx}dx + \\
\left[u\frac{ds}{dx}\right]_{x=0}^{x=1} + \int_0^1 \frac{du}{dx}\frac{ds}{dx}dx - \left[s\frac{du}{dx}\right]_{x=0}^{x=1} .
\end{aligned} \tag{4.14}$$

After the introduction of the boundary conditions one is left with

$$E(u) = -\int_0^1 \frac{du}{dx}\frac{du}{dx}dx + u(1) - 2\int_0^1 udx + \left[u\frac{ds}{dx}\right]_{x=1} - s(1) , \tag{4.15}$$

where the last term can be neglected since it is invariant with respect to the function $u(x)$.

PDS has been employed to solve the nonlinear problem of minimizing Equation (4.15). The objective function for the two-dimensional problem was derived in exact analogy. The optimization variables vectors were given by $\bar{p}_i = (x_M, \Delta x_l, \Delta x_r, c)_i^{\mathrm{T}}$ and $\bar{p}_i = (x_M, \Delta x_l, \Delta x_r, y_M, \Delta y_l, \Delta y_r, c)_i^{\mathrm{T}}$ for the one-dimensional and the two-dimensional cases, respectively, i.e. the nonlinear optimization problem was not decomposed as in Section 4.1.1.

Figure 4.6 presents (a) the exact and approximate solutions, (b) the first derivative, and (c) the RMS error convergence rate for the one-dimensional Poisson's equation when using 18 optimal basis functions in the series expansion. The RMS error was calculated based on 101 uniformly distributed trial points. The accuracy level of the approximate solution is quite satisfactory. The accuracy of the approximation of the first derivative is acceptable considering the fact that piecewise linear basis functions are used in the series expansion of the solution; consequently, piecewise constant basis functions are employed to represent the first derivative of the latter. The convergence rate is at least linear and at most superquadratic.

Figure 4.7 illustrates the distribution of the first 10 optimally selected basis functions for the one-dimensional problem. Note that although the actual domain of the problem is the interval [0,1], the computational domain has been extended to the interval [0,1.5] in order to facilitate the employment of basis functions that vanish at the boundaries in the presence of a Neumann boundary condition.

For the two-dimensional problem, and for illustration purposes, two views of the solution of the two-dimensional Poisson's equation, using 83 optimal basis functions, are shown in Figure 4.8. Figure 4.8(a) includes a contour plot of isolines (lines of equal value).

4.2 Linear nonself-adjoint problems

In this section, a linear, second-order, nonself-adjoint ordinary differential equation with non-homogeneous Dirichlet boundary conditions will be solved by means of the optimal incremental approximation method. The transformation of the nonself-adjoint operator into a self-adjoint operator will be demonstrated. In the case when this transformation is not possible, the shortcoming of the method will be discussed and potential alternatives to overcome this shortcoming will be proposed.

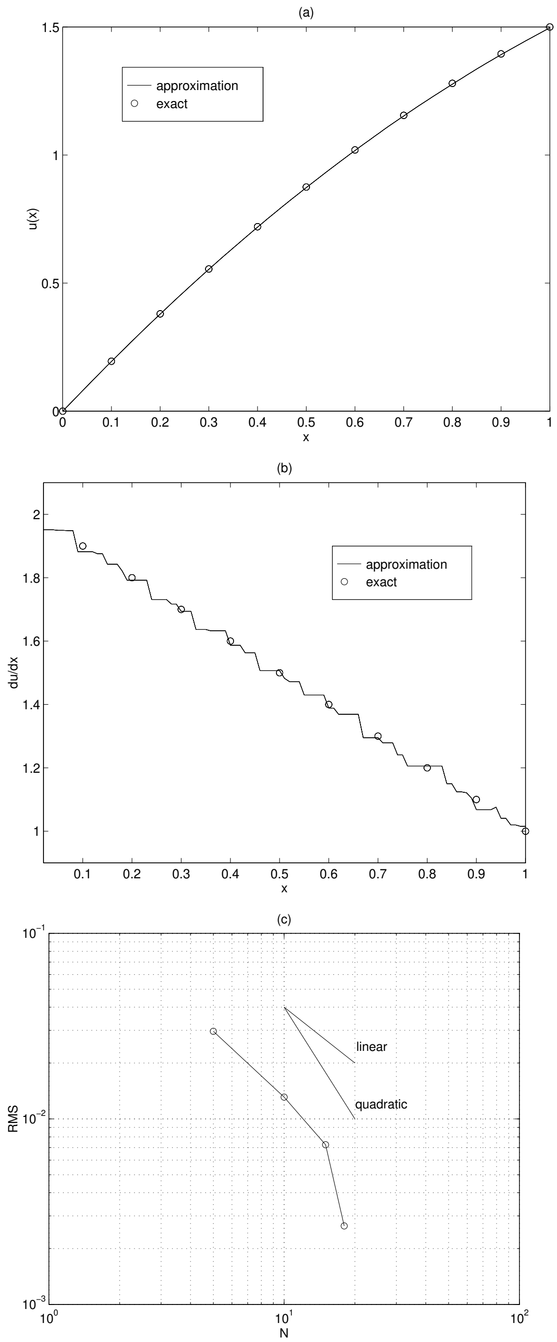

Figure 4.6 One-dimensional Poisson's equation with Neumann boundary condition: (a) exact and approximate solution, (b) exact and approximate first derivative, and (c) convergence rate using 18 optimal basis functions.

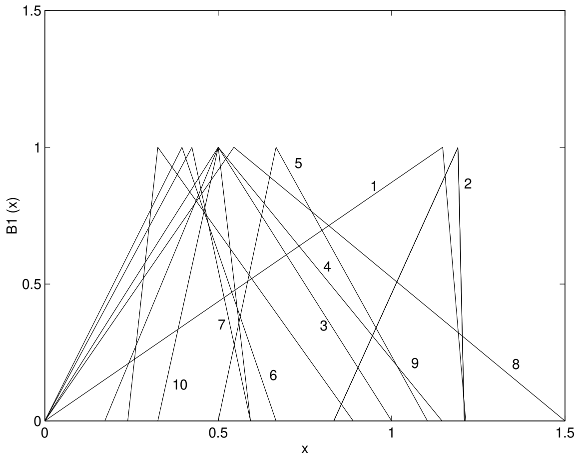

Figure 4.7 Distribution of the first 10 optimal basis functions over the extended computational domain for the one-dimensional Poisson's problem (associated with a Neumann boundary condition).

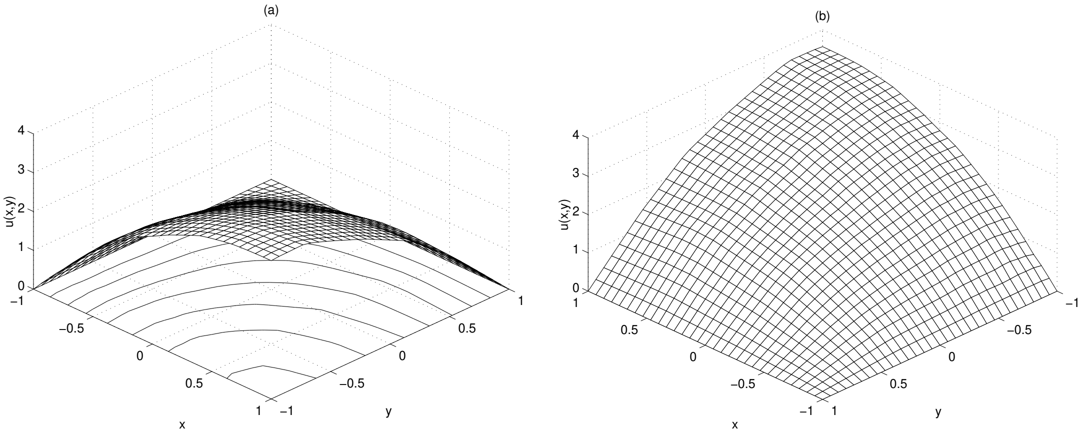

Figure 4.8 Two-dimensional Poisson's equation with Neumann boundary conditions: (a) contour plot and (b) mesh plot of the solution using 83 optimal basis functions.

Consider the general, linear, second-order, nonself-adjoint differential equation

$$L[u] = \alpha \frac{d^2u}{dx^2} + \beta \frac{du}{dx} + \gamma u = f \quad , \quad x \in [a,b] \; , \tag{4.16}$$

associated with general boundary conditions. The coefficients α, β, and γ of the differential equation may also be functions of the independent variable x. The operator L may be adjusted for self-adjointness by premultiplication with some function of the independent variable x, say $P(x)$ [85], [25](page 309). In order for the new operator L' to be self-adjoint, it is necessary and sufficient (Appendix A.4) that the following expression for two arbitrary functions z and w

$$\begin{aligned} &\int_a^b wL'[z]dx - \int_a^b zL'[w]dx = \int_a^b wPL[z]dx - \int_a^b zPL[w]dx = \\ &\int_a^b wP\left(\alpha\frac{d^2z}{dx^2} + \beta\frac{dz}{dx} + \gamma z\right)dx - \int_a^b zP\left(\alpha\frac{d^2w}{dx^2} + \beta\frac{dw}{dx} + \gamma w\right)dx \end{aligned} \tag{4.17}$$

is a function of boundary terms that arise from integration by parts only. Integration by parts yields

$$\begin{aligned} &\int_a^b \left(wP\alpha\frac{d^2z}{dx^2} + wP\beta\frac{dz}{dx} + wP\gamma z\right)dx \;\; - \\ &\int_a^b \left(zP\alpha\frac{d^2w}{dx^2} + zP\beta\frac{dw}{dx} + zP\gamma w\right)dx = \\ &\int_a^b \left(-\frac{d(wP\alpha)}{dx}\frac{dz}{dx} + wP\beta\frac{dz}{dx} + wP\gamma z\right)dx + (\text{B.T.})_1 \;\; - \\ &\int_a^b \left(-\frac{d(zP\alpha)}{dx}\frac{dw}{dx} + zP\beta\frac{dw}{dx} + zP\gamma w\right)dx - (\text{B.T.})_2 = \\ &-\int_a^b P\alpha\frac{dw}{dx}\frac{dz}{dx}dx - \int_a^b w\frac{d(P\alpha)}{dx}\frac{dz}{dx}dx + \int_a^b wP\beta\frac{dz}{dx}dx \;\; + \\ &\int_a^b wP\gamma z dx + (\text{B.T.})_1 + \int_a^b P\alpha\frac{dw}{dx}\frac{dz}{dx}dx \;\; + \end{aligned}$$

$$\int_a^b z\frac{d(P\alpha)}{dx}\frac{dw}{dx}dx - \int_a^b zP\beta\frac{dw}{dx} - \int_a^b zP\gamma w dx - (\mathrm{B.T.})_2 \quad =$$
$$\int_a^b w\frac{dz}{dx}\left(P\beta - \frac{d(P\alpha)}{dx}\right)dx + (\mathrm{B.T.})_1 \quad -$$
$$\int_a^b z\frac{dw}{dx}\left(P\beta - \frac{d(P\alpha)}{dx}\right)dx - (\mathrm{B.T.})_2 \ , \tag{4.18}$$

where $(\mathrm{B.T.})_1$ and $(\mathrm{B.T.})_2$ denote the boundary terms stemming from the two integrations by parts. By examining Equation (4.18), it is obvious that for nonzero α and β, the necessary and sufficient condition for L' to be self-adjoint is

$$P\beta - \frac{d(P\alpha)}{dx} = 0 \ , \tag{4.19}$$

which yields after integration

$$P(x) = e^{\xi} \ , \tag{4.20}$$

where $\xi = \int \alpha^{-1}(\beta - \frac{d\alpha}{dx})dx$. Consequently, the operator L' is self-adjoint, and a variational principle for Equation (4.16) is readily defined as [12]

$$E \equiv \frac{1}{2}\langle u, L'[u]\rangle - \langle u, f\rangle \ , \tag{4.21}$$

provided that the boundary conditions are homogeneous. If the boundary conditions are non-homogeneous, then the functional is defined, according to the developments of Section 4.1.3, as

$$E \equiv \frac{1}{2}\langle q, L'[q]\rangle - \langle q, f'\rangle \tag{4.22}$$

and can be rewritten in terms of u

$$E = \int_a^b (uL'[u] - 2uf + uL'[s] - sL'[u])dx + \int_a^b (2sf - sL'[s])dx \ . \qquad (4.23)$$

Since the second integral is invariant with respect to the solution of the problem u, it can be dropped from the functional.

It is interesting to note that after integrating by parts all terms of the first integral involving the differential operator, and if the non-homogeneous boundary conditions are of the Dirichlet type, the functional (4.23) will be apparently identical to the functional (4.21), derived for the homogeneous boundary value problem, after this is also integrated by parts. The difference is that in this case the function $u(x)$ has to satisfy the non-homogeneous Dirichlet boundary condition essentially; this is why Dirichlet boundary conditions are also called essential boundary conditions. In the case of Neumann boundary conditions, the functional (4.23) will include terms that represent the Neumann boundary conditions. In this regard, the function $u(x)$ does not have to satisfy the Neumann boundary conditions since they are naturally included in the functional. This is why Neumann boundary conditions are also called natural boundary conditions.

It should be noted that the first variation of the functional (4.23) yields problem (4.16) as the associated Euler-Lagrange equation since the transformation of the operator can be interpreted as a simple variable transformation [25](page 309). The second variation of E indicates that the functional should be minimized.

Remarks

1. If the coefficients of the differential equation are constant, then even derivatives are self-adjoint while odd derivatives are nonself-adjoint.

2. If $\beta = 0$, then the original operator L is self-adjoint and no transformation or adjustment is necessary.

3. If $\alpha = 0$, then a transformation or an adjustment is impossible. The introduction of the adjoint problem $L^*[v] = g$ into the variational principle is required to yield [6]

$$E \equiv \int_a^b (v(L[u] - f) - gu)dx = \int_a^b (u(L^*[v] - g) - fv)dx \ . \tag{4.24}$$

   Unfortunately, it is possible that the sign of the second variation of (4.24) is indefinite. In this case the functional cannot be extremized. An alternative would be to minimize the square of the first variation of the functional, in other words to apply the least squares optimization method to the associated Euler-Lagrange equation. However, the simplicity and efficiency of the proposed method would decrease dramatically. Alternatives for the generalization of the method due to the lack of variational principles are considered in Chapter 6.

4.2.1 Numerical example: The convection-diffusion equation

The linear ordinary differential equation

$$-\alpha \frac{d^2u}{dx^2} + \beta \frac{du}{dx} = 0 \quad \text{with} \quad \alpha, \beta > 0 \quad \text{and} \quad 0 \leq x \leq 1 \tag{4.25}$$

is a mathematical model of adequate fidelity for the steady-state balance between convection and diffusion. Guymon derived in references [33] and [34] variational principles for the unsteady problem in one and two dimensions using a similar approach and utilized them for the solution by means of the finite element method. The coefficients α and β represent the viscosity and convection coefficients, respectively, and will assumed to be constant for the sake of simplicity. Equation (4.25) is nondimensionalized and, in combination with the non-homogeneous Dirichlet boundary conditions $u(0) = 0$ and $u(1) = 1$, possess the exact solution

$$u(x) = \frac{e^{\frac{\beta}{\alpha}x} - 1}{e^{\frac{\beta}{\alpha}} - 1} \; . \tag{4.26}$$

The solution function, representing velocity, is characterized by the presence of a boundary layer adjacent to the boundary at $x = 1$ [26]. From a physical point of view, dissipation is, for many flow problems, of significance only in the region of the boundary layer. Therefore, numerical solutions are often oscillatory when the dependent variable exhibits large gradients across the boundary layer. In finite difference methods, for example, the so-called cell Reynolds number, defined in analogy to the freestream Reynolds number (Appendix A.7) by $Re_{cell} = \frac{\beta \Delta x}{\alpha}$, is quite significant for the accuracy and oscillations of the numerical solution [26]. Equation (4.25) poses, in this regard, a challenging problem to be solved by the proposed method.

Given the fact that the differential operator $L[\cdot] = -\alpha \frac{d^2 \cdot}{dx^2} + \beta \frac{d \cdot}{dx}$ is nonselfadjoint, and following the approach presented in the previous section, it can be found that the operator L can be transformed into a self-adjoint operator L', if it is premultiplied by the function $P(x) = e^{-\frac{\beta}{\alpha}x}$. The associated variational principle is then defined, according to Equation (4.23) and for $f = 0$, as

$$E = \int_0^1 (uL'[u] + uL'[s] - sL'[u])dx \; , \tag{4.27}$$

where $s(x) = x$ is chosen to satisfy the boundary conditions. Integrating by parts the first term of the integral yields

$$\int_0^1 ue^{-\frac{\beta}{\alpha}x}\left(-\alpha\frac{d^2u}{dx^2}+\beta\frac{du}{dx}\right)dx = \frac{\beta}{\alpha}\int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{du}{dx}dx - \int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{d^2u}{dx^2}dx =$$
$$\frac{\beta}{\alpha}\int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{du}{dx}dx + \int_0^1 \frac{d(ue^{-\frac{\beta}{\alpha}x})}{dx}\frac{du}{dx}dx - \left[ue^{-\frac{\beta}{\alpha}x}\frac{du}{dx}\right]_0^1 =$$
$$\frac{\beta}{\alpha}\int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{du}{dx}dx + \int_0^1 e^{-\frac{\beta}{\alpha}x}\left(\frac{du}{dx}\right)^2 dx - \frac{\beta}{\alpha}\int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{du}{dx}dx - \left[ue^{-\frac{\beta}{\alpha}x}\frac{du}{dx}\right]_0^1 =$$
$$\int_0^1 e^{-\frac{\beta}{\alpha}x}\left(\frac{du}{dx}\right)^2 dx - \left[e^{-\frac{\beta}{\alpha}}\frac{du}{dx}\right]_{x=1} , \tag{4.28}$$

while integrating by parts the second and third terms of the integral in Equation (4.27) yields

$$\int_0^1 ue^{-\frac{\beta}{\alpha}x}\left(-\alpha\frac{d^2s}{dx^2}+\beta\frac{ds}{dx}\right)dx - \int_0^1 se^{-\frac{\beta}{\alpha}x}\left(-\alpha\frac{d^2u}{dx^2}+\beta\frac{du}{dx}\right)dx =$$
$$\frac{\beta}{\alpha}\int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{ds}{dx}dx - \int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{d^2s}{dx^2}dx - \frac{\beta}{\alpha}\int_0^1 se^{-\frac{\beta}{\alpha}x}\frac{du}{dx}dx + \int_0^1 se^{-\frac{\beta}{\alpha}x}\frac{d^2u}{dx^2}dx =$$
$$\frac{\beta}{\alpha}\int_0^1 ue^{-\frac{\beta}{\alpha}x}\frac{ds}{dx}dx + \int_0^1 \frac{d(ue^{-\frac{\beta}{\alpha}x})}{dx}\frac{ds}{dx}dx - \left[ue^{-\frac{\beta}{\alpha}x}\frac{ds}{dx}\right]_0^1 -$$
$$\frac{\beta}{\alpha}\int_0^1 se^{-\frac{\beta}{\alpha}x}\frac{du}{dx}dx - \int_0^1 \frac{d(se^{-\frac{\beta}{\alpha}x})}{dx}dx + \left[se^{-\frac{\beta}{\alpha}x}\frac{du}{dx}\right]_0^1 =$$
$$\int_0^1 e^{-\frac{\beta}{\alpha}x}\frac{du}{dx}\frac{ds}{dx}dx - \left[e^{-\frac{\beta}{\alpha}}\frac{ds}{dx}\right]_{x=1} - \int_0^1 e^{-\frac{\beta}{\alpha}x}\frac{ds}{dx}\frac{du}{dx}dx + \left[e^{-\frac{\beta}{\alpha}}\frac{du}{dx}\right]_{x=1} =$$
$$e^{-\frac{\beta}{\alpha}}\left[\frac{du}{dx} - \frac{ds}{dx}\right]_{x=1} . \tag{4.29}$$

The final functional is obtained by combining Equations (4.28) and (4.29)

$$E = \int_0^1 e^{-\frac{\beta}{\alpha}x}\left(\frac{du}{dx}\right)^2 dx - \left[e^{-\frac{\beta}{\alpha}}\frac{ds}{dx}\right]_{x=1} . \tag{4.30}$$

The second term of the functional can be omitted since it does not depend on the solution $u(x)$. This functional can be directly minimized in order to approximate the function $u(x)$ that satisfies the differential equation (4.16). The expansion of the approximate solution is given by

$$u_i^a(x) = x + \sum_{j=1}^{i} c_j \Phi_j(x, \bar{p}_j) = u_{i-1}^a(x) + c_i \Phi_i(x, \bar{p}_i) \ , \tag{4.31}$$

where the first term is included in order to satisfy the boundary conditions. By introducing Equation (4.31) into Equation (4.30), the augmented optimization variables vector $\bar{p}_i^{'} = (\bar{p}_i, c_i)^{\mathrm{T}}$ can be computed using PDS in one step. However, like for the Poisson problem, this nonlinear optimization problem can be decomposed into two lower-order problems. By differentiating E with respect to c_i, requiring the derivative to vanish, and abbreviating $\Phi(x, \bar{p})$ to Φ for simplicity, the following expression is obtained

$$2 \int_0^1 e^{-\frac{\beta}{\alpha}x} \frac{du_i^a}{dx} \frac{d\Phi_i}{dx} dx = 0 \ . \tag{4.32}$$

The derivative of (4.31) with respect to the independent variable x is given by

$$\frac{du_i^a(x)}{dx} = 1 + \sum_{j=1}^{i} c_j \frac{d\Phi_j}{dx} = \frac{du_{i-1}^a(x)}{dx} + c_i \frac{d\Phi_i}{dx} \ . \tag{4.33}$$

Substituting Equation (4.33) into Equation (4.32) and solving for the coefficient c_i yields

$$c_i = \left[\int_0^1 e^{-\frac{\beta}{\alpha}x} \frac{d\Phi_i}{dx} \frac{d\Phi_i}{dx} dx \right]^{-1} \times$$

$$\left[- \int_0^1 e^{-\frac{\beta}{\alpha}x} \frac{d\Phi_i}{dx} dx - \sum_{j=1}^{i-1} c_j \int_0^1 e^{-\frac{\beta}{\alpha}x} \frac{d\Phi_j}{dx} \frac{d\Phi_i}{dx} dx \right] . \tag{4.34}$$

Combining Equations (4.34), (4.33), and (4.30), a new objective function is obtained

$$E(u_i^a) = \int_0^1 e^{-\frac{\beta}{\alpha}x} \left(1 + \sum_{j=1}^{i-1} c_j \frac{d\Phi_j}{dx} + \left[\int_0^1 e^{-\frac{\beta}{\alpha}x} \frac{d\Phi_i}{dx} \frac{d\Phi_i}{dx} dx \right]^{-1} \times \right.$$
$$\left. \left[- \int_0^1 e^{-\frac{\beta}{\alpha}x} \frac{d\Phi_i}{dx} dx - \sum_{j=1}^{i-1} c_j \int_0^1 e^{-\frac{\beta}{\alpha}x} \frac{d\Phi_j}{dx} \frac{d\Phi_i}{dx} dx \right] \frac{d\Phi_i}{dx} \right)^2 dx \, . \tag{4.35}$$

The parameters $\bar{p}_i$ of the i-th basis function are determined by minimizing the functional in Equation (4.35), while the associated coefficient c_i is readily calculated from Equation (4.34).

The approximate solutions are compared to the exact in Figure 4.9(a) for $\frac{\beta}{\alpha} = 5$ and Figure 4.9(b) for $\frac{\beta}{\alpha} = 20$. The relatively low number of necessary basis functions and the excellent agreement between exact and approximate solutions, even in the presence of a boundary layer characterized by a large gradient, confirm the potential of the method.

Two plots from Fletcher [26] are reproduced in Figure 4.10 in order to illustrate the large oscillations in the solutions obtained by traditional numerical methods (in this case by the finite difference method). The problem and the boundary conditions associated with Figure 4.10 are the same as those discussed in this section with the parameter ratio is $\frac{\beta}{\alpha} = 20$. The symbol T has been chosen by Fletcher to represent the variable for which the symbol u was used in this text, while the symbol u has been used by Fletcher for the parameter introduced here as β.

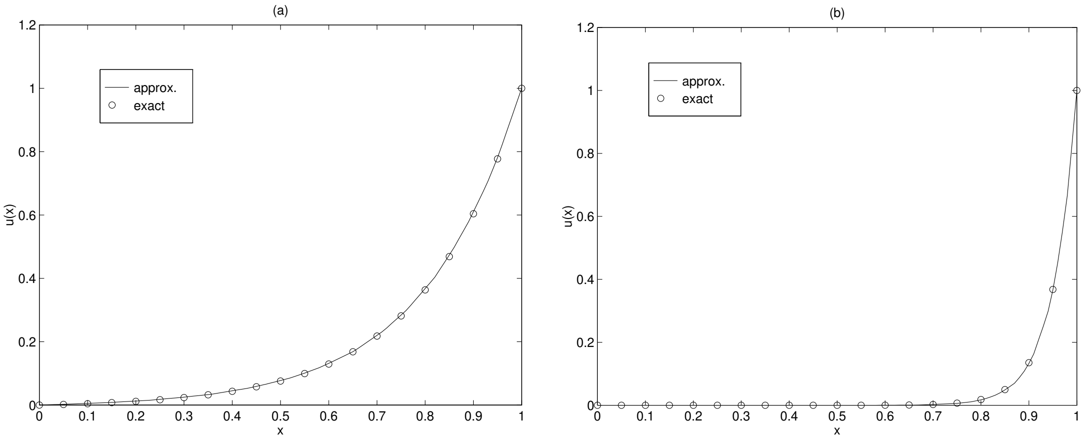

Figure 4.9 One-dimensional convection-diffusion: exact and approximate solutions for (a) $\frac{\beta}{\alpha} = 5$ and (b) $\frac{\beta}{\alpha} = 20$ using 25 and 14 optimal basis functions, respectively.

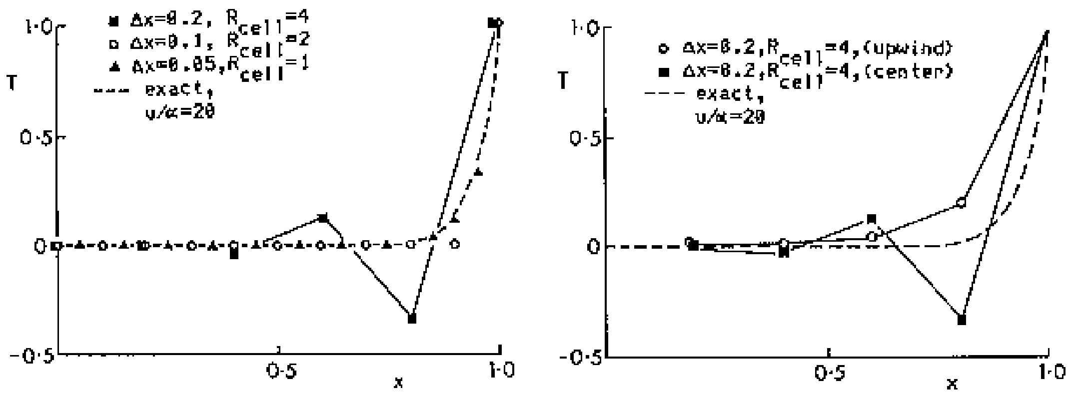

Figure 4.10 One-dimensional convection-diffusion: (left) exact and approximate solutions for different grid spacing and cell Reynolds numbers and (right) exact and approximate solutions in the presence of an upwind scheme; $\frac{\beta}{\alpha} = 20$, plots reproduced from reference [26].

Figure 4.11 includes the solutions obtained for $\frac{\beta}{\alpha} = 20$ by the finite element method when using the Bubnov-Galerkin technique and linear Lagrangian shape functions on a uniform grid for 5, 9, 11, and 21 interpolation functions, respectively. The same sort of oscillations can be observed when using a low number of grid points; the approximate solution is in good agreement with the exact only when using a high number of interpolation functions, i.e. only for fine grids. In contrast, Figure 4.12 presents the approximate solutions obtained by the proposed method when using 1, 2, 3, and 4 basis functions only. There are no oscillations and the solution possess an acceptable level of accuracy considering the number of used basis functions.

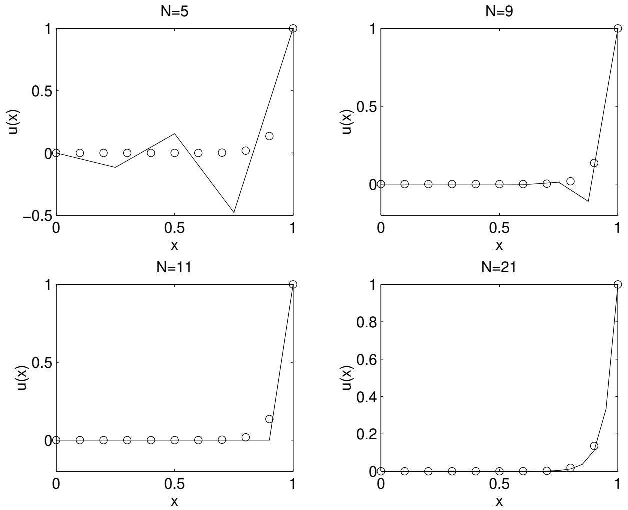

Figure 4.11 One-dimensional convection-diffusion: the evolution of the solution with increasing number of functions for $\frac{\beta}{\alpha} = 20$; Bubnov-Galerkin.

Figure 4.13(a) depicts the RMS error convergence rates obtained for $\frac{\beta}{\alpha} = 5$ by executing PDS using (curve 1) a tolerance *tol* of 10^{-4} and 1000 pattern points d, introduced in Section 4.1.2, and (curve 2) a tolerance of 10^{-6} and 100 points. The

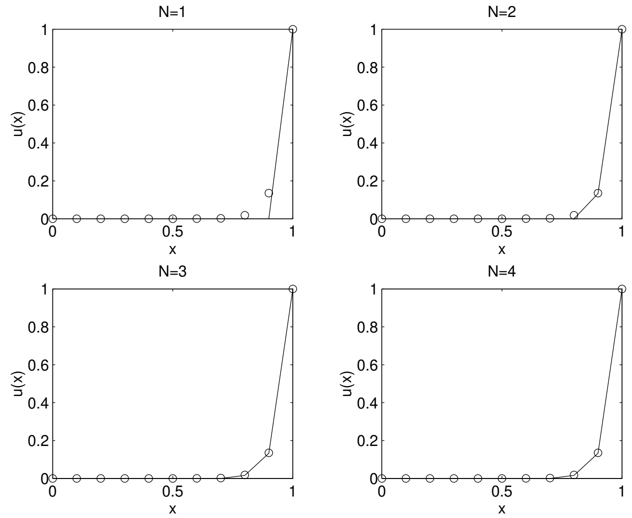

Figure 4.12 One-dimensional convection-diffusion: the evolution of the solution with increasing number of functions for $\frac{\beta}{\alpha} = 20$; proposed method.

RMS error was calculated based on 101 uniformly distributed trial points. A small quantitative difference in the convergence rates can be observed. In fact, when compared to the convergence rate obtained by the finite element method using linear Lagrangian shape functions on a uniform grid, it is clear that the size of the search slightly affects the efficiency of the method. The RMS error convergence rate for $\frac{\beta}{\alpha} = 20$, $tol = 10^{-4}$, and $d = 1000$ is plotted in Figure 4.13(b) and compared to that obtained by the finite element method. It can be concluded that when the solution of the problem has a relatively large gradient in some region of the domain, the proposed method will fully exploit its adaptive nature and perform much better than traditional methods.

It can be argued that the adjustment of the differential operator L by the introduction of the premultiplying function $P(x)$ is equivalent to the application of the Petrov-Galerkin method to the original problem, when the latter is solved by

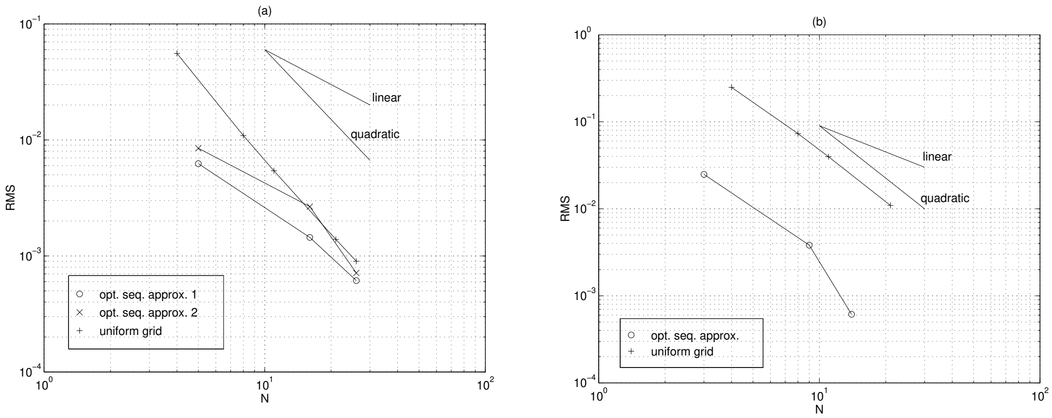

Figure 4.13 One-dimensional convection-diffusion: convergence rates for (a) $\frac{\beta}{\alpha} = 5$ (curve 1: $tol = 10^{-4}$, $d = 1000$, curve 2: $tol = 10^{-6}$, $d = 100$) and (b) $\frac{\beta}{\alpha} = 20$ ($tol = 10^{-4}$, $d = 1000$).

the traditional finite element method. Since the new, equivalent problem is

$$L^{'}[u(x)] = P(x)f(x) = P(x)L[u(x)] \tag{4.36}$$

or, more specifically,

$$e^{-\frac{\beta}{\alpha}x}\left(-\alpha\frac{d^2u}{dx^2} + \beta\frac{du}{dx}\right) = 0 \ , \tag{4.37}$$

the equation residual, after introducing the approximate solution u^a, is

$$e^{-\frac{\beta}{\alpha}x}\left(-\alpha\frac{d^2u^a}{dx^2} + \beta\frac{du^a}{dx}\right) = R^{'} \ . \tag{4.38}$$

Applying the Bubnov-Galerkin method of weighted residuals to the residual $R^{'}$ is equivalent to applying the Petrov-Galerkin method of weighted residuals to the original residual $R = -\alpha\frac{d^2u^a}{dx^2} + \beta\frac{du^a}{dx}$ since

$$\langle R', \Phi_j\rangle = \langle R, e^{-\frac{\beta}{\alpha}x}\Phi_j\rangle \ . \tag{4.39}$$

Moreover, the system of equations obtained by applying the Raleigh-Ritz method to the problem, i.e. the n algebraic equations obtained by requiring the derivatives of the variational principle of Equation (4.30) with respect to the coefficients c_j, $j = 1, \ldots, n$, to vanish, is equivalent to the system of equations obtained by the Petrov-Galerkin. It was found that the equivalent Petrov-Galerkin and Raleigh-Ritz methods eliminated the oscillatory behavior of the solution but did not enhance the RMS error convergence rate when utilized using piecewise linear Lagrangian shape functions on a uniform grid; the convergence rate is slightly worse, as seen in Figure 4.14. In addition, it can be seen that the convergence rate

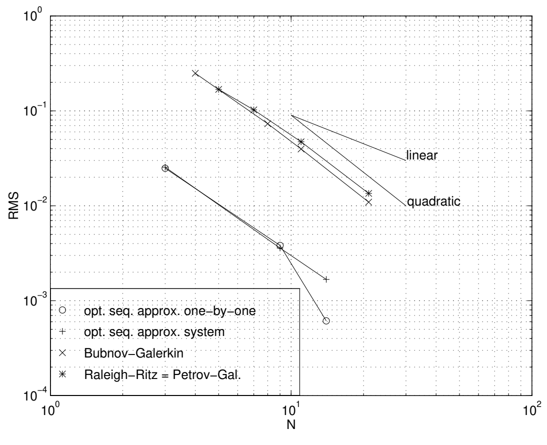

Figure 4.14 One-dimensional convection-diffusion: convergence rates comparison considering the Petrov-Galerkin and Raleigh-Ritz methods; $\frac{\beta}{\alpha} = 20$.

of the proposed method is better when the coefficients are computed one by one at each stage of the algorithm, as discussed in the previous sections, and not when all the coefficients are recomputed by solving a system of equations at each step, as will be discussed in Section 6.2.2.

Figure 4.15 illustrates the adaptive nature of the proposed method. Specifically, the first 10 optimally selected basis functions are plotted for the test case with $\frac{\beta}{\alpha} = 20$. It is clear that the basis functions are concentrated in the area of the steep gradient. An additional characteristic of the method is illustrated in Figure 4.16. Since the method is based on the adaptive improvement of the series expansion, the absolute value of the coefficients should decrease with increasing number of expansion terms. The oscillations observed in Figure 4.16 can be explained by the fact that the coefficients do not necessarily have to decrease strictly monotonically, especially when their sign oscillates in the early stage of the proposed algorithm.

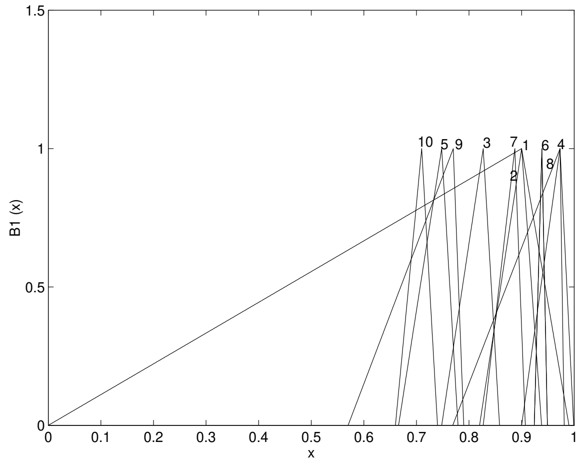

Figure 4.15 One-dimensional convection-diffusion: distribution of the first 10 optimally selected basis functions for $\frac{\beta}{\alpha} = 20$.

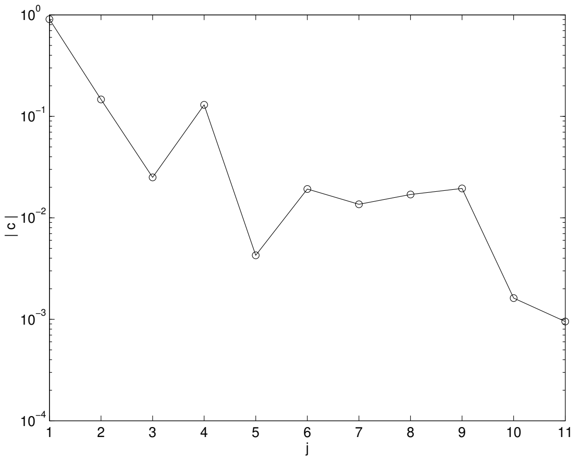

Figure 4.16 One-dimensional convection-diffusion: absolute value of the coefficients c_j for $\frac{\beta}{\alpha} = 20$.

4.2.2 Computational cost

Table 4.2 summarizes the amount of computational work required by PDS at selected iterations of the proposed algorithm. As observed for the linear, self-adjoint problem, the number of required iterations and function and constraint evaluations does not fluctuate significantly. As expected, a low number of pattern points associated with a tighter tolerance requires more iterations but fewer function and constraint evaluations than a large number of pattern points associated with a relaxed tolerance. It is interesting to note that in both cases the algorithm requires less work in its initial stage than in all the subsequent steps.

Table 4.2 Amount of computational work required by PDS for the linear convection-diffusion problem ($\frac{\beta}{\alpha} = 5$).

Algorithm step	# of PDS iterations	Total # of function eval.	Total # of constraint eval.
	$d = 100$	$tol = 10^{-6}$	
1	24	836	2272
8	18	1493	1536
16	15	1307	1365
24	16	1168	1335
	$d = 1000$	$tol = 10^{-4}$	
1	7	916	5344
8	11	5721	6683
16	12	6970	8719
24	10	7401	8475

Chapter 5

Nonlinear differential equations

The derivation of natural variational principles is certainly not straightforward and usually not possible for problems governed by nonlinear differential operators. The use of Frechret derivatives is typically required for the development of contrived variational principals; the associated work and effort make this procedure unattractive [24].

There exist, however, many applications of interest for which the derivation of variational principles is tractable. Oden, for example, presents in reference [66] a variational principle for the nonlinear partial differential equation

$$2u\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)+\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2=f(x,y) \quad \text{in} \quad \Omega \tag{5.1}$$

subject to the Dirichlet boundary conditions $u(s)=g(s)$ on $\partial\Omega$, namely the functional

$$E(u)=\frac{1}{2}\int_\Omega\left(u\left(\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial x}\right)^2\right)-fu\right)d\bar{x}-\oint_{\partial\Omega}ugds\ , \tag{5.2}$$

where $\bar{x}=(x,y)^{\mathrm{T}}$. Greenspan studies in references [29, 30], among others, the so-called Plateau problem of finding the surface of least area bounded by a given closed curve, i.e. minimizing

$$E(u)=\int_\Omega\sqrt{1+\left(\frac{\partial u}{\partial x}\right)^2+\left(\frac{\partial u}{\partial y}\right)^2}\,d\bar{x}\ . \tag{5.3}$$

The associated Euler-Lagrange equation turns out to be

$$\left(1+\left(\frac{\partial u}{\partial y}\right)^2\right)\frac{\partial^2 u}{\partial x^2}-2\frac{\partial u}{\partial x}\frac{\partial u}{\partial y}\frac{\partial^2 u}{\partial x \partial y}+\left(1+\left(\frac{\partial u}{\partial x}\right)^2\right)\frac{\partial^2 u}{\partial y^2}=0\ . \tag{5.4}$$

5.1 Two-dimensional, compressible, inviscid, irrotational, isentropic flow

Many kinds of flow of practical interest in fluid dynamics, including compressible flow *, can be assumed to be irrotational. When a flow is irrotational, the vorticity of the fluid, which is proportional to its angular velocity, is equal to zero. This is the case for supersonic flows around sharp edges or cones, for two-dimensional or axisymmetric nozzle flows, and it can be assumed to be the case for flows around slender sharp-nosed bodies. However, supersonic flow around blunt bodies and viscous flow inside a boundary layer are certainly rotational [2].

Whenever a flow can be assumed to be irrotational, the velocity vector can be defined as the gradient of a scalar called the velocity potential, i.e. $\bar{u} = \nabla\varphi$. In this manner, the governing equations (continuity and momentum equations) can be combined into a single nonlinear partial differential equation with φ as the single unknown. Once φ is found, all other flow properties can be calculated.

Variational principles for the problem of compressible, inviscid, unsteady flow have been derived for the one- and two-dimensional cases from Prozan [74, 73]. Manwell discusses in reference [57] a variational principle for steady, compressible, inviscid flow with shocks, but does not present numerical results. His approach is similar to that presented earlier by Greenspan and Jain in reference [31], where

*A flow is assumed to be compressible if the freestream Mach number M_∞ exceeds the numerical value of 0.3.

a variational principle, derived for the problem of steady, irrotational, inviscid, compressible flow, was extremized using finite differences. In fact, the results of an extensive literature search indicated that Greenspan has been one of few researchers to work directly with the variational principle and attempt to determine the extremal function by direct optimization [29, 31, 30]. In this regard, the proposed method for optimal sequential approximation will be applied to the problem studied by Greenspan and Jain in reference [31].

5.1.1 The velocity potential equation

The equations governing steady, two-dimensional, homentropic, irrotational, inviscid, compressible flow past a fixed body are given by the continuity and momentum equations, respectively

$$\frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0 \tag{5.5}$$

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial x} = 0 \tag{5.6}$$

$$u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} + \frac{1}{\rho}\frac{\partial p}{\partial y} = 0\ . \tag{5.7}$$

In these equations, u and v are the x- and y- Cartesian velocity components, respectively, ρ is density, and p is pressure. The velocity vector is defined as the gradient of the velocity potential φ

$$\bar{u} = (u, v)^{\mathrm{T}} = \nabla\varphi = \left(\frac{\partial\varphi}{\partial x}, \frac{\partial\varphi}{\partial y}\right)^{\mathrm{T}} \tag{5.8}$$

and has magnitude $V^2 = \parallel \bar{u} \parallel_2^2 = u^2 + v^2$. The set of equations is completed by the equation of state for an ideal gas

$$p = a\rho^{\gamma} \ , \tag{5.9}$$

where a is a constant and γ is the ratio of specific heats. Equations (5.5) - (5.7) can be combined to yield the potential velocity equation [2, 31]

$$\left(c^2 - \left(\frac{\partial\varphi}{\partial x}\right)^2\right)\frac{\partial^2\varphi}{\partial x^2} - 2\frac{\partial\varphi}{\partial x}\frac{\partial\varphi}{\partial y}\frac{\partial^2\varphi}{\partial x\partial y} + \left(c^2 - \left(\frac{\partial\varphi}{\partial y}\right)^2\right)\frac{\partial^2\varphi}{\partial y^2} \ , \tag{5.10}$$

where $c^2 = dp/d\rho$ is the square of the local speed of sound. According to the equation of state, for an ideal gas

$$c^2 = a\gamma\rho^{\gamma-1} \ . \tag{5.11}$$

The associated boundary conditions are

$$\frac{\partial\varphi}{\partial n} = 0 \quad \text{on the body surface, and} \tag{5.12}$$

$$\nabla\varphi = (V_\infty, 0)^{\mathrm{T}} = constant \quad \text{far from the body,} \tag{5.13}$$

where n is the direction normal to the body surface and V_∞ is the freestream velocity, which is recovered in a distance far from the body.

5.1.2 The variational integral

Bateman proposed in reference [5] two variational principles whose variations yield the governing equations for steady, irrotational, isentropic, compressible, inviscid flow. The first is implicitly expressed in terms of the velocity potential and has to be maximized if the stationary flow is subsonic throughout the domain of interest. The second is defined with help of the stream function ψ, where

$$\bar{u} = (u, v)^{\mathrm{T}} = \left(\frac{\partial \psi}{\partial y}, \frac{\partial \psi}{\partial x}\right)^{\mathrm{T}} , \tag{5.14}$$

and has to be minimized [56]. In the work presented in this document, the first variational principle has been considered. Since pressure is a function of density, and density is in turn a function of the velocity magnitude, the variational integral is given by

$$E = \int_{\Omega} p d\bar{x} = \int_{\Omega} f(u^2 + v^2) d\bar{x} = \int_{\Omega} f((\partial \varphi / \partial x)^2 + (\partial \varphi / \partial y)^2) d\bar{x} \tag{5.15}$$

and is associated with the following Euler-Lagrange equation

$$\frac{\partial^2 f}{\partial u^2}\frac{\partial^2 \varphi}{\partial x^2} + 2\frac{\partial^2 f}{\partial u \partial v}\frac{\partial^2 \varphi}{\partial x \partial y} + \frac{\partial^2 f}{\partial v^2}\frac{\partial^2 \varphi}{\partial y^2} = 0 \ . \tag{5.16}$$

By comparing Equations (5.10) and (5.16) it can be concluded that

$$\frac{\partial^2 f}{\partial u^2} = c^2 - \left(\frac{\partial \varphi}{\partial x}\right)^2 , \tag{5.17}$$

$$\frac{\partial^2 f}{\partial u \partial v} = -\frac{\partial \varphi}{\partial x}\frac{\partial \varphi}{\partial y} , \quad \text{and} \tag{5.18}$$

$$\frac{\partial^2 f}{\partial v^2} = c^2 - \left(\frac{\partial \varphi}{\partial y}\right)^2 . \tag{5.19}$$

The functional of Equation (5.15) is divergent for infinite domains. This drawback has been remedied by Lush and Cherry by means of introducing appropriate terms into Equation (5.15) that do not affect the associated Euler-Lagrange equation

[56]. The modified variational integral is convergent and has to be maximized. It is defined as

$$
\begin{aligned}
E(\varphi) = & \int_\Omega \left(p - p_\infty + V_\infty \rho_\infty \frac{\partial (V_\infty \chi)}{\partial x} \right) d\bar{x} + \\
& \oint_C \left(\rho \varphi \frac{\partial \varphi}{\partial n} - V_\infty^2 \rho_\infty \chi \frac{\partial \chi}{\partial n} \right) ds \ ,
\end{aligned}
\tag{5.20}
$$

where ∞ denotes freestream conditions, χ is some unknown auxiliary function, and C represents the closed body curve. The potential φ is given by

$$
\varphi = V_\infty x + V_\infty \chi \ . \tag{5.21}
$$

According to the boundary condition, χ has to vanish far from the body. The body used by Greenspan and Jain in reference [31] was a unit cylinder, and in this regard, polar coordinates were utilized to derive the final functional

$$
\frac{E(\chi)}{p_\infty} = \int_{\theta=0}^{2\pi} \int_{r=1}^{\infty} (F - 1 + \gamma M_\infty^2 H) r dr d\theta \ , \tag{5.22}
$$

where

$$
\begin{aligned}
F = & \left(1 - \left(\frac{\gamma - 1}{2} M_\infty^2 \right) \times \right. \\
& \left. \left(2\cos(\theta) \frac{\partial \chi}{\partial r} - 2 \frac{\sin(\theta)}{r} \frac{\partial \chi}{\partial \theta} + \left(\frac{\partial \chi}{\partial r} \right)^2 + \frac{1}{r^2} \left(\frac{\partial \chi}{\partial \theta} \right)^2 \right) \right)^{\frac{\gamma}{\gamma - 1}}
\end{aligned}
\tag{5.23}
$$

and

$$
H = \left(1 - \frac{1}{r^2} \right) \frac{\partial \chi}{\partial r} \cos(\theta) - \left(1 + \frac{1}{r^2} \right) \frac{\partial \chi}{\partial \theta} \frac{\sin(\theta)}{r} \ . \tag{5.24}
$$

M_∞ is the freestream Mach number given by

$$M_\infty = \frac{V_\infty}{c_\infty} \; , \tag{5.25}$$

where c_∞ is the freestream speed of sound. Finally, the boundary conditions are

$$\frac{\partial \chi}{\partial r} = -\cos(\theta) \quad \text{at} \quad r = 1 \tag{5.26}$$

and

$$\chi = 0 \quad \text{far from the body} \; . \tag{5.27}$$

5.1.3 The computational domain

The variational principle of Equation (5.22) is defined over an infinite domain. However, it is assumed that the radius spans from $r = 1$ (body surface) to $r = 20$ (far field). This assumption is more than justified considering the fact that typical numerical grids in airfoil flow analysis, extend six to seven times the length of the bodies. In addition, the flow is completely symmetrical about the lines $\theta = 0$, $\theta = \pi$, $\theta = \pi/2$, and $\theta = 3\pi/2$. In this regard, the computational domain can be restricted to the range $\pi/2 \leq \theta \leq \pi$. With respect to the above assumptions, the variational integral to be numerically maximized is given by

$$\int_{\theta_1=\pi/2}^{\theta_2=\pi} \int_{r_1=1}^{r_2=20} (F - 1 + \gamma M_\infty^2 H) r dr d\theta \; . \tag{5.28}$$

Figure 5.1 depicts the physical (x-y-plane) and computational (r-θ-plane) domains. The points A,B,C, and D with x-y coordinates (-20,0), (-1,0), (0,1), and (0,20),

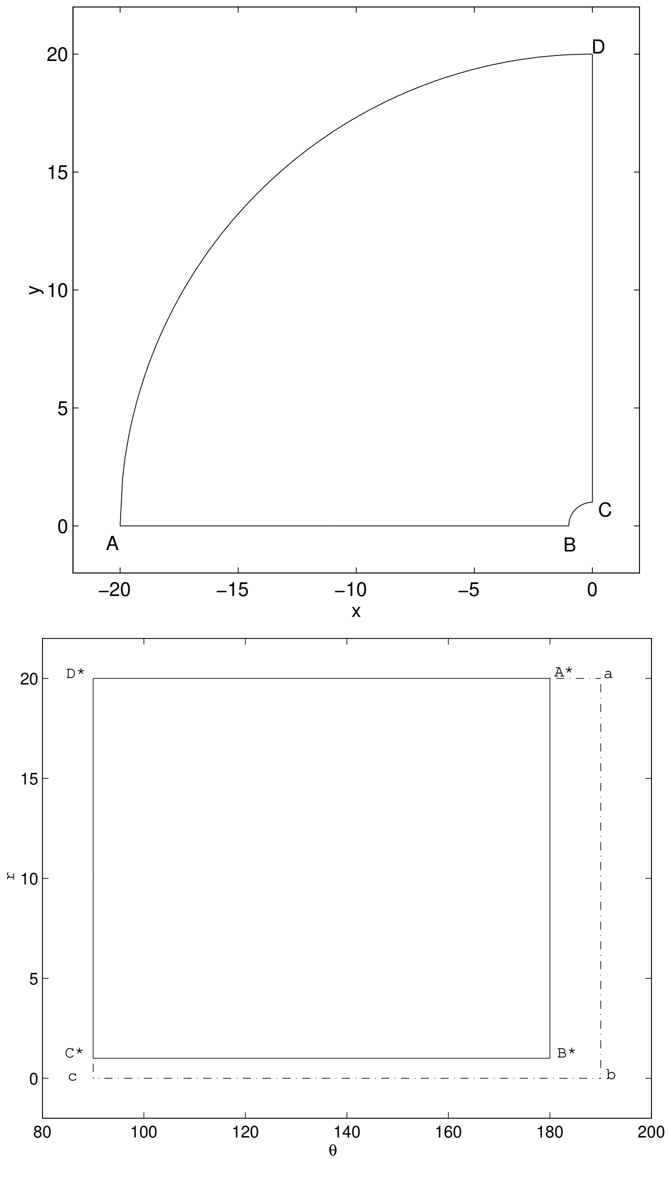

Figure 5.1 Compressible inviscid flow: (top) the physical (x-y-plane) and (bottom) computational (r-θ-plane) domains.

respectively, are mapped on the points A*, B*, C*, and D*, with r-θ coordinates (20,π), (1,π), (1,$\pi/2$), and (20,$\pi/2$), respectively.

The presence of homogeneous Dirichlet boundary conditions on the lines $r = 20$ and $\theta = \pi/2$ facilitates the use of basis functions that vanish at the domain boundaries. In this regard, and due to the presence of Neumann boundary conditions on the lines $r = 1$ and $\theta = \pi$, the computational domain has been extended, as shown in Figure 5.1. The extended computational domain is described by the points a, b, c, and D^* in the r-θ plane.

5.1.4 Boundary conditions

The boundary conditions have to be adjusted for the modified computational domain. Specifically, the appropriate boundary conditions are

$$\chi_r = -\cos(\theta) \quad \text{at} \quad r = 1 \ , \tag{5.29}$$

$$\chi = 0 \quad \text{at} \quad r = 20 \quad \text{and} \quad \theta = \frac{\pi}{2} \ , \quad \text{and} \tag{5.30}$$

$$\chi_\theta = 0 \quad \text{at} \quad \theta = \pi \ , \tag{5.31}$$

where the subscripts denote partial differentiation.

It must now be determined whether the Neumann boundary conditions are naturally satisfied by the variational principle or if additional terms and/or modifications of the integral are necessary. Assuming that the Dirichlet boundary conditions are satisfied by the trial functions, the variation of $E(\chi)$ with respect to a variation in χ, defined as

$$\Xi = \chi + \epsilon\eta \tag{5.32}$$

with

$$\Xi_r = \chi_r + \epsilon\eta_r \quad \text{and} \quad \Xi_\theta = \chi_\theta + \epsilon\eta_\theta \ , \tag{5.33}$$

is given by

$$\left[\frac{\partial E}{\partial \epsilon}\right]_{\epsilon=0} = \int\int \left[\frac{\partial(Fr)}{\partial \epsilon}\right]_{\epsilon=0} drd\theta + \int\int \left[\frac{\partial(\gamma M^2 Hr)}{\partial \epsilon}\right]_{\epsilon=0} drd\theta \tag{5.34}$$

and must vanish, for E to be extremized.

In this regard, and by noting that

$$F = F(r, \theta, \chi_r, \chi_\theta) \quad , \quad F \neq F(\chi)$$
$$H = H(r, \theta, \chi_r, \chi_\theta) \quad , \quad H \neq H(\chi)$$
$$r \neq r(\epsilon) \quad \text{, and} \quad \theta \neq \theta(\epsilon) \ ,$$

it can be found that

$$\left[\frac{\partial(Fr)}{\partial \epsilon}\right]_{\epsilon=0} = \left[\frac{\partial(Fr)}{\partial \Xi_r}\frac{\partial \Xi_r}{\partial \epsilon}\right]_{\epsilon=0} + \left[\frac{\partial(Fr)}{\partial \Xi_\theta}\frac{\partial \Xi_\theta}{\partial \epsilon}\right]_{\epsilon=0} \tag{5.35}$$

and

$$\left[\frac{\partial(\gamma M^2 Hr)}{\partial \epsilon}\right]_{\epsilon=0} = \left[\frac{\partial(\gamma M^2 Hr)}{\partial \Xi_r}\frac{\partial \Xi_r}{\partial \epsilon}\right]_{\epsilon=0} + \left[\frac{\partial(\gamma M^2 Hr)}{\partial \Xi_\theta}\frac{\partial \Xi_\theta}{\partial \epsilon}\right]_{\epsilon=0} \tag{5.36}$$

with

$$\left[\frac{\partial \Xi_r}{\partial \epsilon}\right]_{\epsilon=0} = \eta_r \ , \quad \left[\frac{\partial \Xi_\theta}{\partial \epsilon}\right]_{\epsilon=0} = \eta_\theta \ , \tag{5.37}$$

and

$$\left[\frac{\partial(Fr)}{\partial \Xi_r}\right]_{\epsilon=0} = -G^{\frac{1}{\gamma-1}}\gamma r M^2(\cos\theta + \chi_r) \equiv P \tag{5.38}$$

$$\left[\frac{\partial(Fr)}{\partial \Xi_\theta}\right]_{\epsilon=0} = -G^{\frac{1}{\gamma-1}}\gamma M^2\left(\frac{1}{r}\chi_\theta - \sin\theta\right) \equiv Q \tag{5.39}$$

$$\left[\frac{\partial(\gamma M^2 Hr)}{\partial \Xi_r}\right]_{\epsilon=0} = \gamma M^2\left(r - \frac{1}{r}\right)\cos\theta \equiv R \tag{5.40}$$

$$\left[\frac{\partial(\gamma M^2 Hr)}{\partial \Xi_\theta}\right]_{\epsilon=0} = -\gamma M^2 \left(1+\frac{1}{r^2}\right)\sin\theta \equiv S \ , \tag{5.41}$$

where

$$G = 1 - \frac{1}{2}(\gamma-1)M^2 \left(2\cos\theta\chi_r - 2\frac{1}{r}\sin\theta\chi_\theta + \chi_r^2 + \frac{1}{r^2}\chi_\theta^2\right) \ . \tag{5.42}$$

Substituting Equations (5.35)-(5.41) into Equation (5.34) yields

$$E = \int\int(P\eta_r + Q\eta_\theta)drd\theta + \int\int(R\eta_r + S\eta_\theta)drd\theta \ . \tag{5.43}$$

Integrating by parts the first terms of each integral with respect to r and the second terms of each integral with respect to θ yields

$$\begin{aligned} E = &-\int\int\frac{\partial P}{\partial r}\eta drd\theta - \int\int\frac{\partial Q}{\partial \theta}\eta drd\theta + \int[P\eta]_{r_1}^{r_2}d\theta + \int[Q\eta]_{\theta_1}^{\theta_2}dr \ - \\ &\int\int\frac{\partial R}{\partial r}\eta drd\theta - \int\int\frac{\partial S}{\partial \theta}\eta drd\theta + \int[R\eta]_{r_1}^{r_2}d\theta + \int[S\eta]_{\theta_1}^{\theta_2}dr \ . \end{aligned} \tag{5.44}$$

But

$$\frac{\partial R}{\partial r} = -\frac{\partial S}{\partial \theta} \tag{5.45}$$

and η is zero on $r = r_2$ and $\theta = \theta_1$ (due to the homogeneous Dirichlet boundary conditions on these boundaries) and so the functional variation reads as

$$\begin{aligned} E = &-\int\int\left(\frac{\partial P}{\partial r} + \frac{\partial Q}{\partial \theta}\right)\eta drd\theta - \int[P\eta]_{r_1}d\theta \ + \\ &\int[Q\eta]_{\theta_2}dr - \int[R\eta]_{r_1}d\theta + \int[S\eta]_{\theta_2}dr \ . \end{aligned} \tag{5.46}$$

In addition, $[R]_{r_1} = 0$ and $[S]_{\theta_2} = 0$; the final functional variation is given therefore by

$$E = -\int\int \left(\frac{\partial P}{\partial r} + \frac{\partial Q}{\partial \theta}\right) \eta dr d\theta - \int [P\eta]_{r_1} d\theta + \int [Q\eta]_{\theta_2} dr \ . \tag{5.47}$$

In order for E to vanish for arbitrary η, all the integrands must vanish, i.e.

$$\frac{\partial P}{\partial r} + \frac{\partial Q}{\partial \theta} = 0 \ , \tag{5.48}$$

$$[P]_{r_1} = 0 \ , \quad \text{and} \tag{5.49}$$

$$[Q]_{\theta_2} = 0 \ . \tag{5.50}$$

Equation (5.48) is the associated Euler-Lagrange equation of the functional, and Equations (5.49) and (5.50) are the Neumann boundary conditions on $r = r_1$ and $\theta = \theta_2$, respectively. Specifically, from Equation (5.49)

$$\begin{aligned} &[-G^{\frac{1}{\gamma-1}} \gamma r M^2 (\cos\theta + \chi_r)]_{r_1} = 0 \Longrightarrow \\ &[G^{\frac{1}{\gamma-1}}]_{r_1} = 0 \ \text{ or } \ [\cos\theta + \chi_r]_{r_1} = 0 \end{aligned} \tag{5.51}$$

and from Equation (5.50)

$$\begin{aligned} &\left[-G^{\frac{1}{\gamma-1}} \gamma M^2 \left(\frac{1}{r}\chi_\theta - \sin\theta\right)\right]_{\theta_2} = 0 \Longrightarrow \\ &[G^{\frac{1}{\gamma-1}}]_{\theta_2} = 0 \ \text{ or } \ \left[\frac{1}{r}\chi_\theta\right]_{\theta_2} = 0 \ . \end{aligned} \tag{5.52}$$

Since $G^{\frac{1}{\gamma-1}}$ represents the density of the fluid, and can therefore not be equal to zero, the Neumann boundary conditions are given by

$$\chi_r = -\cos\theta \quad \text{on} \quad r = r_1 \tag{5.53}$$

and

$$\chi_\theta = 0 \quad \text{on} \quad \theta = \theta_2 \ , \tag{5.54}$$

which are indeed the (both homogeneous and non-homogeneous) Neumann boundary conditions of the problem and are therefore naturally satisfied by the functional (5.22).

5.1.5 Numerical procedure and results

The proposed algorithm for optimal incremental approximation has been utilized to numerically approximate the extremal function $\chi(r,\theta)$ that maximizes the variational principle defined in Section 5.1.2 and adjusted to the computational domain discussed in Sections 5.1.3 and 5.1.4. The function $\chi(r,\theta)$ is approximated by the series expansion [61]

$$\chi_k^a(r,\theta) = \sum_{i=1}^{k} c_i \Phi_i(r, \bar{p}_i^r)\Phi_i(\theta, \bar{p}_i^\theta) \ , \tag{5.55}$$

where the basis functions Φ_i are, as previously introduced, B_1-splines with parameters $\bar{p}_i^r = (r_M, \Delta r_l, \Delta r_r)_i^{\mathrm{T}}$ and $\bar{p}_i^\theta = (\theta_M, \Delta\theta_l, \Delta\theta_r)_i^{\mathrm{T}}$. The c_i are the corresponding expansion coefficients.

The function $\chi(r,\theta)$ appears only implicitly, in the form of its partial derivatives with respect to r and θ, in the functional. They are accordingly approximated by

$$\chi_r \approx \frac{\partial \chi_k^a}{\partial r} = \sum_{i=1}^{k} c_i \frac{\partial \Phi_i(r, \bar{p}_i^r)}{\partial r} \Phi_i(\theta, \bar{p}_i^\theta) \tag{5.56}$$

$$\chi_\theta \approx \frac{\partial \chi_k^a}{\partial \theta} = \sum_{i=1}^{k} c_i \Phi_i(r, \bar{p}_i^r) \frac{\partial \Phi_i(\theta, \bar{p}_i^\theta)}{\partial \theta} \; . \tag{5.57}$$

These approximations are introduced into the variational integral to be maximized at each step of the algorithm, and PDS is employed to determine the optimal parameters and expansion coefficient.

Due to the high complexity and nonlinearity of the functional, the optimization problem cannot be decomposed into lower-dimensional problems; all seven optimization variables, represented by the vector $\bar{p}_i = (r_M, \Delta r_l, \Delta r_r, \theta_M, \Delta\theta_l, \Delta\theta_r, c)_i^{\mathrm{T}}$, are computed in one step by solving a seven-dimensional highly nonlinear optimization problem using PDS. To enhance computational efficiency in the absence of simplification and decomposition possibilities, the scaling option of PDS was activated (Appendix A.5.2). The components of the initial guess vector for each step of the algorithm were the following: r_M was positioned close to the cylinder surface, while Δr_l and Δr_r were chosen so that $\Phi(r)$ covers the computational domain in the r-direction; θ_M was positioned in the middle of the domain (with respect to the θ-direction), while $\Delta\theta_l$ and $\Delta\theta_r$ were also chosen so that $\Phi(\theta)$ covers the computational domain in the θ-direction. Finally, the initial guess for the coefficient c was set equal to zero so that the functional value achieved in the previous step of the algorithm will be the initial value in the next step. In this manner, it is guaranteed that the functional value will be strictly monotonically improved with each iteration of the proposed algorithm.

Numerical results are presented for transonic flow of air about a unit cylinder and compared to results presented by Greenspan and Jain in reference [31]. The following flow properties were estimated after solving for the function χ. The nondimensionalized speed of the fluid

$$\left(\frac{V}{V_\infty}\right)^2 = 1 + 2\cos(\theta)\chi_r - 2\frac{\sin(\theta)}{r}\chi_\theta + (\chi_r)^2 + \frac{1}{r^2}(\chi_\theta)^2 \ , \tag{5.58}$$

the nondimensionalized local speed of sound

$$\left(\frac{c}{V_\infty}\right)^2 = \frac{1}{M_\infty^2} + \frac{1}{2}(\gamma - 1)\left(1 - \left(\frac{V}{V_\infty}\right)^2\right) \ , \tag{5.59}$$

the nondimensionalized pressure

$$\frac{p}{p_\infty} = \left(1 + \frac{1}{2}(\gamma - 1)M_\infty^2\left(1 - \left(\frac{V}{V_\infty}\right)^2\right)\right)^{\frac{\gamma}{\gamma-1}} \ , \tag{5.60}$$

the pressure coefficient

$$C_p = \frac{\frac{p}{p_\infty} - 1}{\frac{1}{2}\gamma M_\infty^2} \ , \tag{5.61}$$

and the nondimensionalized density

$$\frac{\rho}{\rho_\infty} = \left(\frac{p}{p_\infty}\right)^{\frac{1}{\gamma}} \ , \tag{5.62}$$

where $\gamma = 1.405$. Transonic flow conditions are simulated by freestream Mach numbers in the range between 0.4 and 0.5. Greenspan and Jain reported results for many test cases, but their Newton-like algorithm failed to converge for freestream

Mach numbers that were greater then 0.43. In their numerical test cases, they used $\Delta\theta = 9°$ and $\Delta r = 0.1$, 0.15, and 0.2, i.e. they used a finite difference grid of $11 \times 191 = 2101$, $11 \times 127 = 1397$, and $11 \times 96 = 1056$ nodes, respectively. As will be shown on the illustrations provided below, the results reported by Greenspan and Jain are quite sensitive with respect to the value of Δr. The numerical test cases presented here include the freestream Mach numbers 0.4, 0.43, and 0.46. The results obtained by the proposed method are compared to those presented in reference [31]. The latter include results obtained previously by means of other numerical and analytical approximation methods and reported by Lush and Cherry [56] and Imai [41] for $M_\infty = 0.4$.

Test case I: $M_\infty = 0.4$

The first numerical test case simulates near sonic conditions. However, the flow remains subsonic throughout the domain of interest.

As illustrated in Figures 5.2 and 5.3, the extremal function $\chi(r, \theta)$ was satisfactorily approximated after 31 optimal basis functions were determined and used in the series expansion. The values of the function χ at different radii are compared to results presented in reference [31], which include numerical values obtained by 1) Imai and 2) Greenspan and Jain using a) $\Delta r = 0.1$ and b) $\Delta r = 0.2$. The agreement is good for $r = 1$. For higher radii the results obtained by the proposed algorithm deviate, although not significantly, from those reported by Greenspan and Jain. It should be emphasized, however, that these values have been reproduced from scanned plots due to the absence of tabulated data in reference [31]. Consequently, error has been introduced into the presented values. Moreover, the numerical results presented by Greenspan and Jain are over three decades old. Therefore, they merely constitute a comparison basis and not an accurate refer-

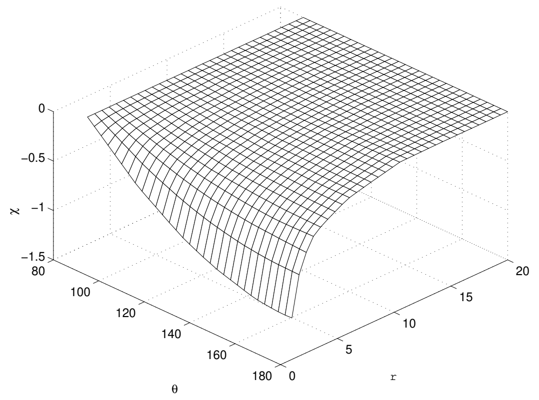

Figure 5.2 Compressible inviscid flow: the approximate velocity potential auxiliary function $\chi(r, \theta)$ for $M_\infty = 0.4$ using 31 optimal basis functions.

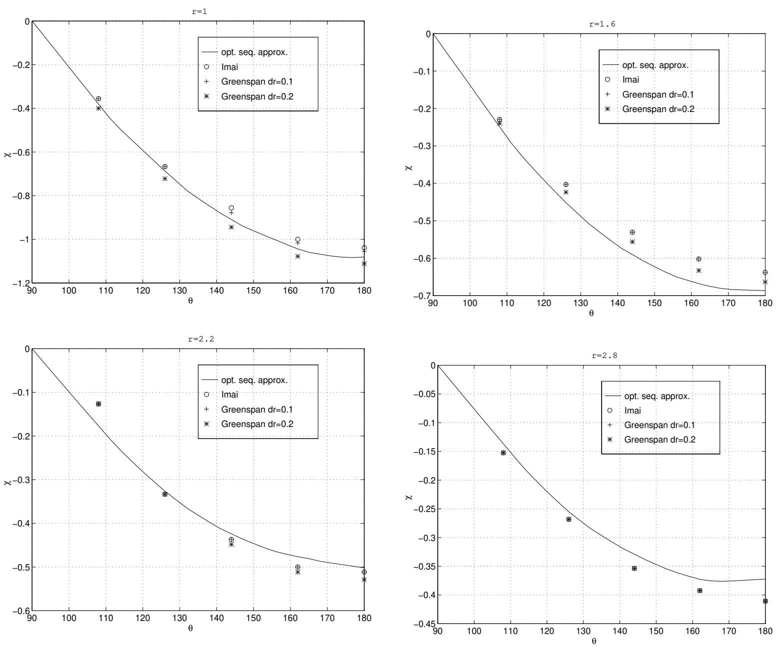

Figure 5.3 Compressible inviscid flow: the approximate velocity potential auxiliary function $\chi(r, \theta)$ at different radii (on and near the cylinder surface) for $M_\infty = 0.4$ using 31 optimal basis functions.

ence. In addition, these methods failed to yield results for test cases with freestream Mach numbers greater than 0.43, while the proposed method succeeded; the associated results will be presented later.

The nondimensionalized fluid velocity on the surface of the cylinder ($r = 1$) is plotted in Figure 5.4 and compared to the results of Greenspan and Jain, Lush and Cherry, and Imai. Figure 5.4 also depicts the nondimensionalized local speed of sound on the cylinder surface. It can be seen that the fluid velocity does not exceed the sonic speed in any part of the domain; the flow is completely subsonic.

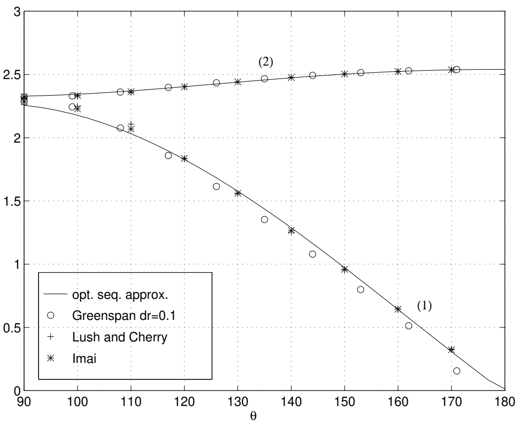

Figure 5.4 Compressible inviscid flow: nondimensionalized (curve 1) fluid velocity and (curve 2) sonic speed on the cylinder surface for $M_\infty = 0.4$.

The pressure coefficient distribution C_p and the nondimensionalized density on the cylinder surface are drawn in Figure 5.5.

Figure 5.6 presents the local Mach number distribution over the entire flow field and some representative Mach-isolines (lines of equal Mach number) for the region close to the cylinder surface.

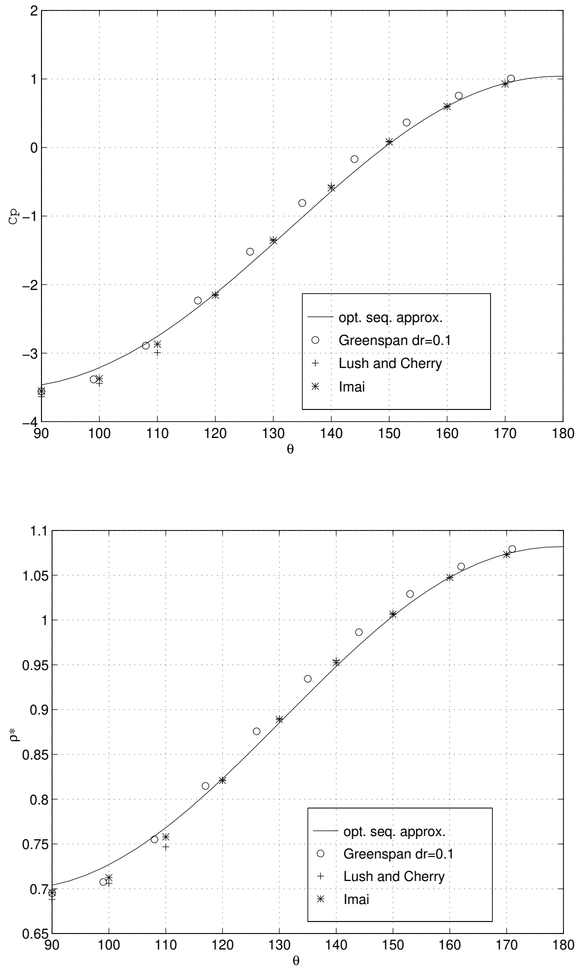

Figure 5.5 Compressible inviscid flow: (top) pressure coefficient distribution and (bottom) nondimensionalized density on the cylinder surface for $M_\infty = 0.4$.

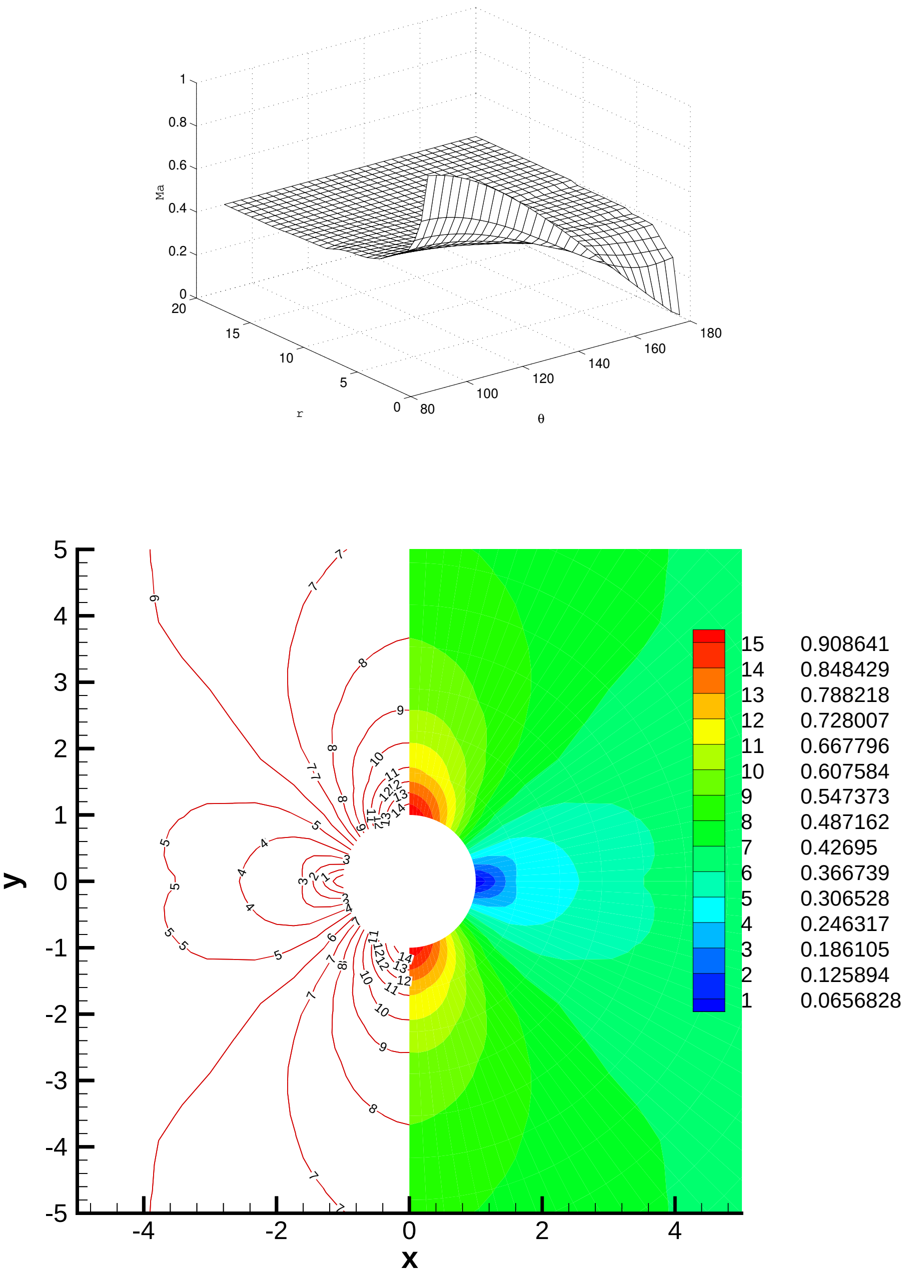

Figure 5.6 Compressible inviscid flow, $M_\infty = 0.4$: (top) local Mach number distribution over the entire flow field and (bottom) Mach-isolines in a region close to the cylinder surface.

Test case II: $M_\infty = 0.43$

Transonic conditions are successfully simulated when the freestream Mach number takes the value of 0.43. Specifically, a supersonic bubble appears in the vicinity of $\theta = \pi/2$ on and near the cylinder surface.

The numerically approximated function $\chi(r,\theta)$ is depicted in Figures 5.7 and 5.8 using 30 optimal basis functions. It can be concluded by comparison with

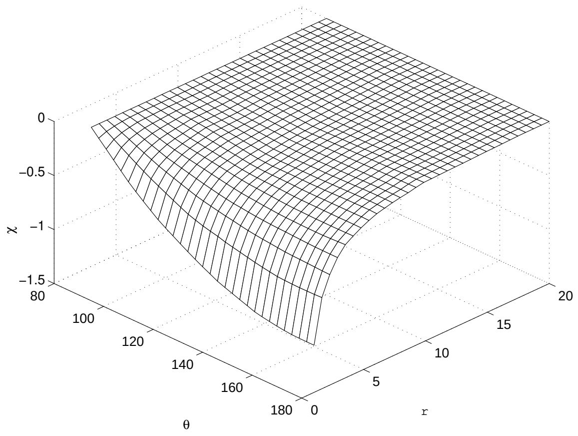

Figure 5.7 Compressible inviscid flow: the approximate velocity potential auxiliary function $\chi(r,\theta)$ for $M_\infty = 0.43$ using 30 optimal basis functions.

Figures 5.2 and 5.3 that the function reaches higher absolute values for the present test case. Once again, the values of the function χ at different radii are compared to results presented in reference [31], which include numerical values obtained by 1) Imai and 2) Greenspan and Jain using a) $\Delta r = 0.1$ and b) $\Delta r = 0.15$. The agreement seems to be better for this test case.

The nondimensionalized fluid velocity and local speed of sound on the surface of the cylinder, presented in Figure 5.9, illustrate the appearance of the supersonic

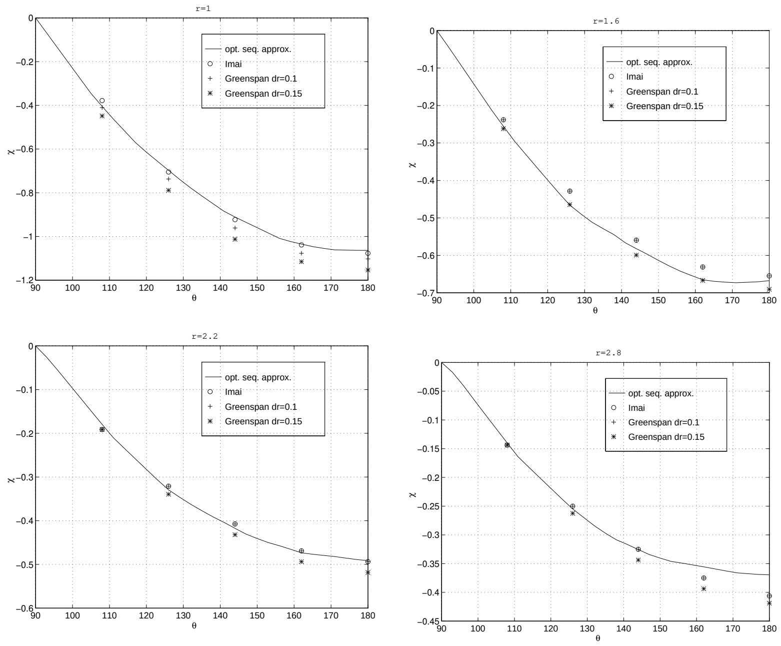

Figure 5.8 Compressible inviscid flow: the approximate velocity potential auxiliary function $\chi(r, \theta)$ at different radii (on and near the cylinder surface) for $M_\infty = 0.43$ using 30 optimal basis functions.

bubble. In the vicinity of $\theta = 90°$, the fluid velocity exceeds the local speed of sound and the flow becomes supersonic. The sensitivity of the results with respect to the Δr value is obvious. In addition, it can be concluded that the fluid velocity is overestimated by the numerical method utilized by Greenspan and Jain, especially in the supersonic flow range. Accordingly, the pressure coefficients and the nondimensionalized density, plotted in Figure 5.10, are overestimated in absolute value.

Figure 5.11 presents the local Mach number distribution over the entire flow field and representative Mach-isolines for the region close to the cylinder surface. The presence of regions where the flow is supersonic is clear.

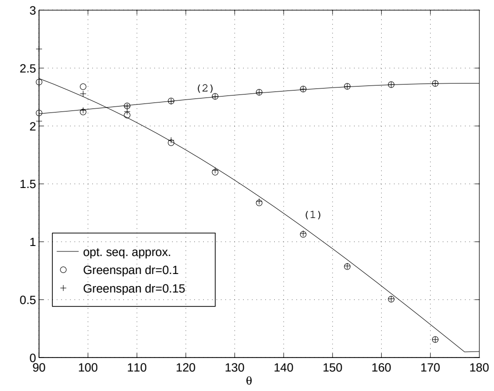

Figure 5.9 Compressible inviscid flow: nondimensionalized (curve 1) fluid velocity and (curve 2) sonic speed on the cylinder surface for $M_\infty = 0.43$.

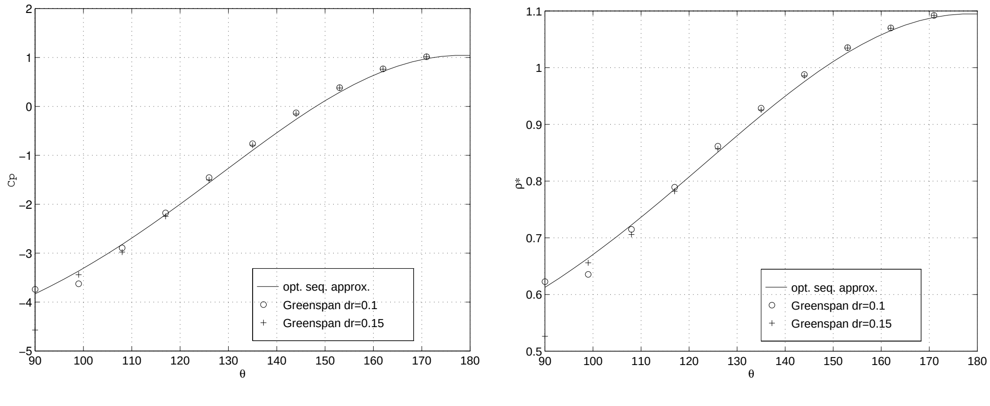

Figure 5.10 Compressible inviscid flow: (left) pressure coefficient distribution and (right) nondimensionalized density on the cylinder surface for $M_\infty = 0.43$.

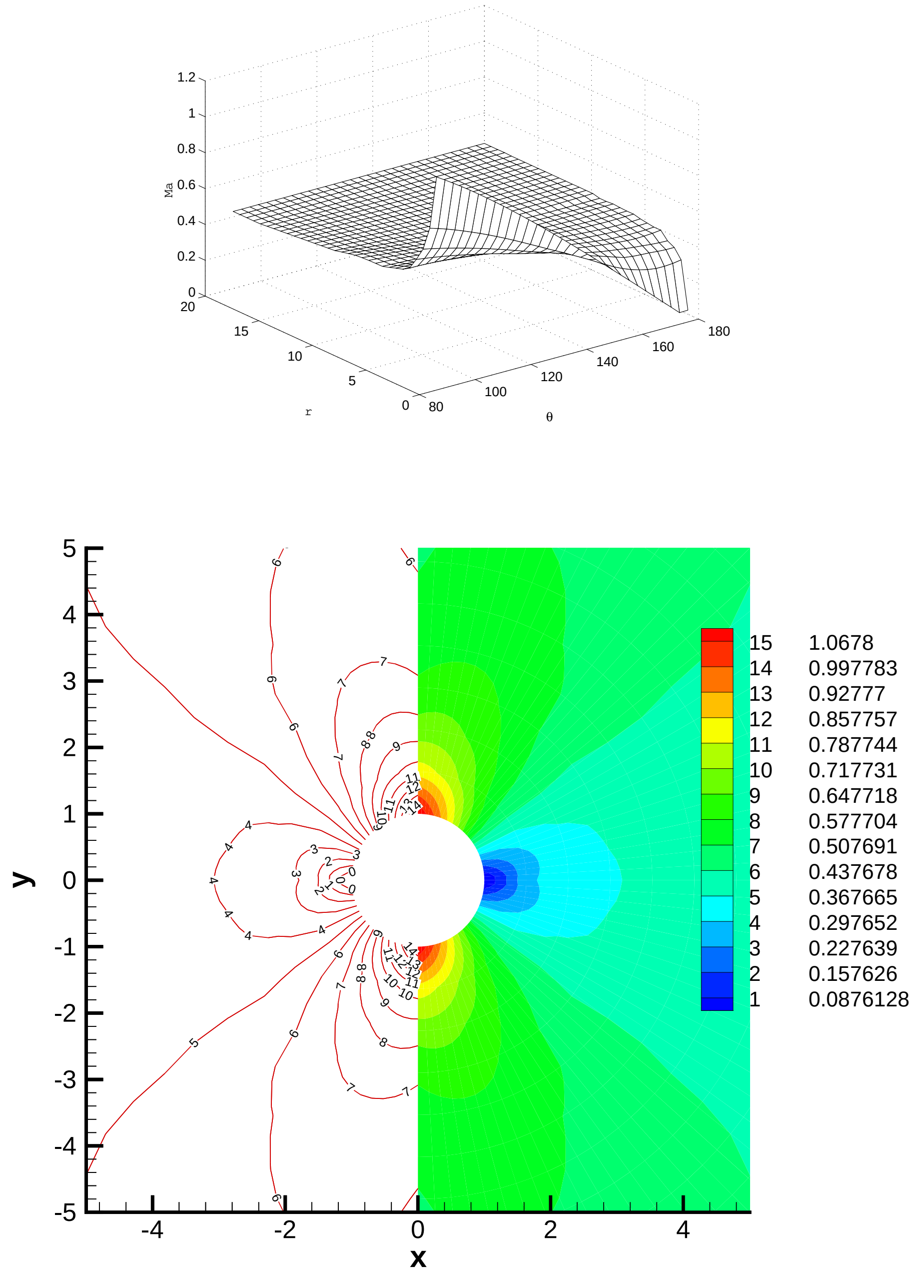

Figure 5.11 Compressible inviscid flow, $M_\infty = 0.43$: (top) local Mach number distribution over the entire flow field and (bottom) Mach-isolines in a region close to the cylinder surface.

Test case III: $M_\infty = 0.46$

As mentioned, the numerical method utilized by Greenspan and Jain failed to converge for freestream Mach numbers greater than 0.43. The proposed algorithm for optimal incremental approximation did not fail in numerical test cases with larger Mach numbers.

In fact, results are presented, without comparison and for future reference, as a complement to the results presented in reference [31], for $M_\infty = 0.46$ using 41 optimal basis functions to approximate the velocity potential auxiliary function $\chi(r, \theta)$, which is depicted in Figures 5.12 and 5.13.

The nondimensionalized fluid velocity and sonic speed plots of Figure 5.14 confirm that the flow conditions are transonic.

Figure 5.15 presents the pressure coefficient distribution and the nondimensionalized density on the cylinder surface.

The local Mach number distribution over the entire flow field and representative Mach-isolines near the cylinder surface are illustrated for future reference in Figure 5.16.

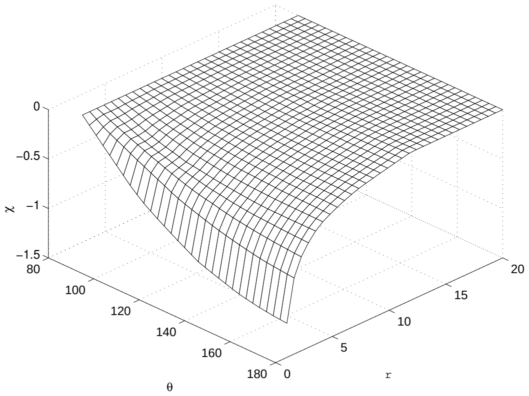

Figure 5.12 Compressible inviscid flow: the approximate velocity potential auxiliary function $\chi(r, \theta)$ for $M_\infty = 0.46$ using 41 optimal basis functions.

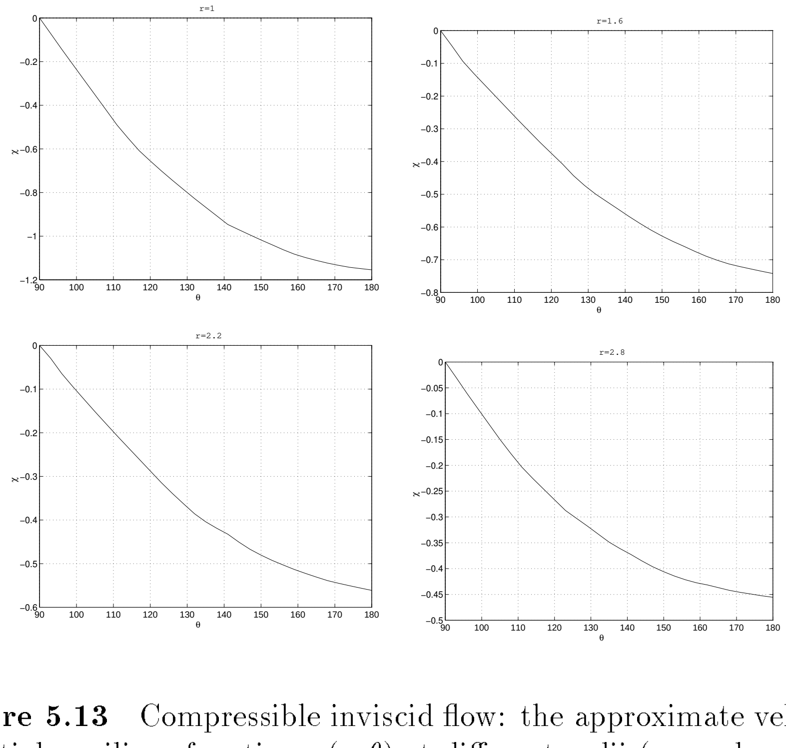

Figure 5.13 Compressible inviscid flow: the approximate velocity potential auxiliary function $\chi(r, \theta)$ at different radii (on and near the cylinder surface) for $M_\infty = 0.46$ using 41 optimal basis functions.

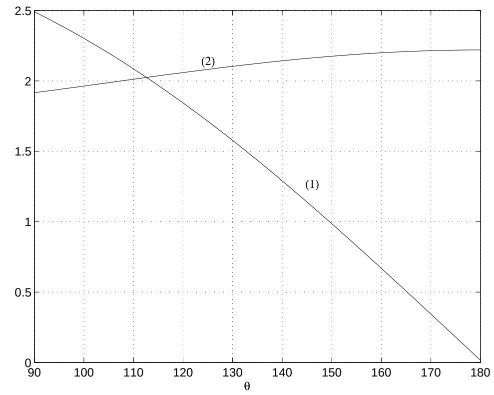

Figure 5.14 Compressible inviscid flow: nondimensionalized (curve 1) fluid velocity and (curve 2) sonic speed on the cylinder surface for $M_\infty = 0.46$.

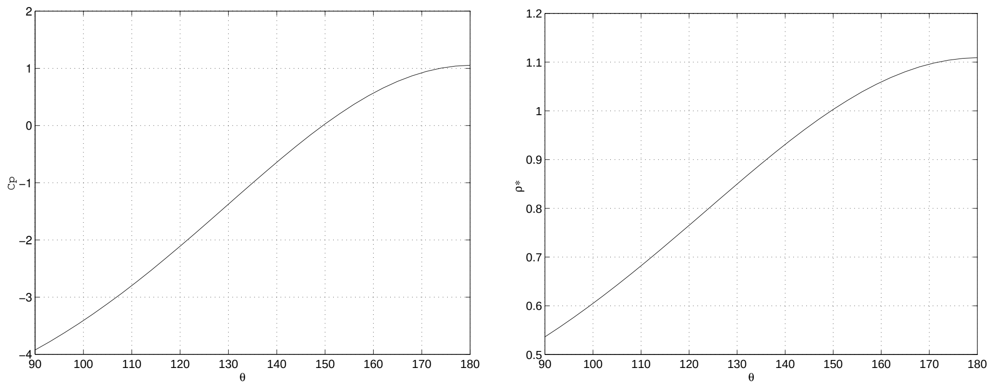

Figure 5.15 Compressible inviscid flow: (left) pressure coefficient distribution and (right) nondimensionalized density on the cylinder surface for $M_\infty = 0.46$.

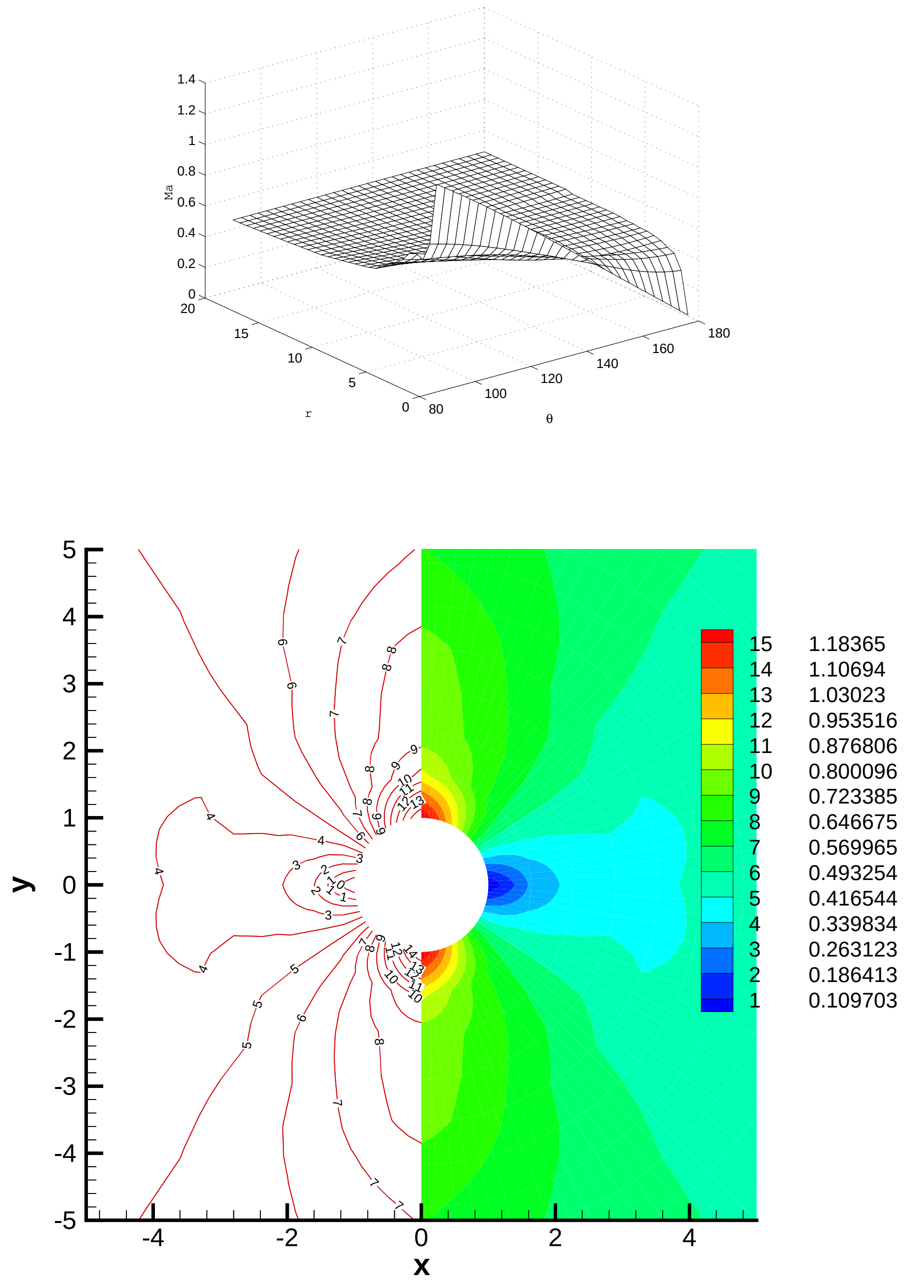

Figure 5.16 Compressible inviscid flow, $M_\infty = 0.46$: (top) local Mach number distribution over the entire flow field and (bottom) Mach-isolines in a region close to the cylinder surface.

Computational work

The amount of computational work that PDS required at each step of the proposed algorithm is presented in Table 5.1.

Computational work is recorded in terms of the numbers of required PDS iterations and function and constraint evaluations. The strictly monotonically decreasing functional value is included. As can be concluded by inspection of Table 5.1, the amount of computational work does not exhibit high fluctuations as the algorithm advances. Even when the maximum numerical value of the functional is asymptotically approached, as the number of optimal basis function increases, the amount of work necessary for the improvement of the variational integral value is of the same order as for the previous iterations.

Table 5.1 Amount of computational work required by PDS for the two-dimensional compressible inviscid irrotational flow problem using $d = 1120$ and $tol = 10^{-4}$.

Algorithm step	# of PDS iterations	Total # of function eval.	Total # of constraint eval.	Functional value
		$M = 0.4$		
1	21	6337	17202	0.053511
10	16	11464	12155	0.093902
20	42	33811	41611	0.095980
30	17	13711	15000	0.096385
		$M = 0.43$		
1	17	8014	12655	0.069409
10	15	7941	13806	0.106047
20	22	20452	21232	0.111336
30	24	19443	24323	0.113558
		$M = 0.46$		
1	17	7543	12654	0.080411
10	31	23691	27128	0.128511
20	44	28435	39596	0.143601
30	24	23453	23643	0.149425
40	37	36678	38842	0.150619

Chapter 6

Generalization aspects of the proposed method

The proposed method has been based on the utilization of variational principles. Variational integrals, whenever available in a natural or contrived manner, succinctly and concisely represent differential equations and the associated boundary conditions. In addition, the employment of the variational forms, as the objective function in the formulated optimization problems, guarantee that the latter are well-posed.

However, the derivation of variational principles for all types of problems is far from being straightforward. Moreover, variational integrals may not always be prone to extremization, as indicated in the last remark of Section 4.2. In this regard, an alternative is necessary.

6.1 Utilizing the method of weighted residuals

The method of weighted residuals could act as the necessary alternative. For example, the minimization of the square of the residual of a differential equation corresponds to the well-known least-squares weighted residual method; if the objective functional $\langle R, R\rangle$ is differentiated with respect to the coefficients c_i, $i = 1, \ldots, n$, of a series expansion, a system of n algebraic equations is formed for the solution of the unknown vector $\bar{c}$. The functional $\langle R, R\rangle$ could be used as the objective function to be minimized in the proposed algorithm of sequential approximation. In this case, both the coefficients c_i and the parameters of the basis functions Φ_i of the series expansion constitute the optimization variables; the proposed method is similar to the least-squares weighted residuals method. However, the disadvantages include: 1) the necessity of developing techniques for the accommodation of the boundary conditions, 2) the possible solution of a multi-objective optimization problem in the event that a given problem is governed by more than one differential equations, and 3) solution uniqueness problems, especially in the case of nonlinear differential equations.

An alternative to the least-squares method that is within the class of the method of weighted residuals, can be deduced by noting that the numerator of the last term of Equation (4.8), which is to be maximized, can be rewritten as

$$\begin{aligned}
&(l(e_{i-1}, \Phi_i(\bar{x})))^2 = (l(u(\bar{x}) - u^a_{i-1}(\bar{x}), \Phi_i(\bar{x})))^2 = \\
&(l(u(\bar{x}), \Phi_i(\bar{x})) - l(u^a_{i-1}(\bar{x}), \Phi_i(\bar{x})))^2 = \\
&(\langle L[u(\bar{x})], \Phi_i(\bar{x})\rangle - \langle L[u^a_{i-1}(\bar{x})[, \Phi_i(\bar{x})\rangle)^2 = \\
&(\langle f(\bar{x}), \Phi_i(\bar{x})\rangle - \langle L[u^a_{i-1}(\bar{x})], \Phi_i(\bar{x})\rangle)^2 = \langle R_{i-1}, \Phi_i(\bar{x})\rangle^2 \ ,
\end{aligned} \tag{6.1}$$

where the equation residual R_{i-1} is equal to $-(L[u^a_{i-1}(\bar{x})] - f(\bar{x}))$. From a geometric point of view, Equation (6.1) indicates that by maximizing $\langle R_{i-1}, \Phi_i\rangle^2$, the i-th basis function will be selected such that the basis function Φ_i is as parallel as possible to the equation residual. For the sake of generalization, the objective function $\langle R_{i-1}, \Psi_i\rangle^2$ will be used, where Ψ_i is a function of Φ_i.

The selection of the associated coefficient c_i according to another geometric consideration follows; the residual R_i is forced to be as orthogonal as possible to Ψ_i. In this regard, the objective function to be minimized is $\langle R_i, \Psi_i\rangle^2$, with c_i being the optimization variable. The equivalence of this technique to the Petrov-Galerkin method of weighted residuals is apparent. In fact, for linear problems the coefficient can be readily calculated by the Galerkin relation $\langle R_i, \Psi_i\rangle = 0$ rather than minimizing $\langle R_i, \Psi_i\rangle^2$. However, the implementation of the boundary conditions may not be straightforward, especially if they are of the Neumann type, and this alternative approach may face solution uniqueness problems.

6.2 Numerical examples

6.2.1 Boundary value problem

As demonstrated in Section 4.1, the properties of the linear self-adjoint operator L of the boundary value problem $L[u] = f$ can be exploited, in combination with the homogeneous Dirichlet boundary conditions, to simplify the numerical problem of extremizing the associated variational principle $E(u)$. Specifically, instead of directly minimizing $E(u)$ (Equation (4.3)), the optimal basis functions and the associated coefficients of the series expansion are determined by a two-step procedure; firstly, the optimal basis function parameters are computed by maximizing the last term of Equation (4.8) is maximized and, secondly, the expansion coefficient is calculated by the Galerkin equation (4.9).

For this problem, directly extremizing the functional is equivalent to the above described two-step procedure; the numerical results, presented in Section 4.1.1, are identical.

6.2.2 Initial value problem

Consider the first order differential equation (initial value problem)

$$\frac{du}{dx} + \alpha u = f \tag{6.2}$$

in the interval $a \leq x \leq b$, completed by the initial condition $u(a) = u_0$. For the sake of simplicity, let $a = 0$, $b = 1$, $\alpha = 1$, $f = 0$, and $u_0 = 1$. The exact solution is given then by $u(x) = e^{-\alpha x}$.

In attempting to place Equation (6.2) in a variational form, it is noted that the differential operator $L[\cdot] = \frac{d[\cdot]}{dx} + \alpha[\cdot]$ is linear and non-self adjoint. As mentioned in the remarks of Section 4.2, no modification or adjustment of the operator L can yield a self-adjoint operator L'. The introduction of the adjoint problem $L^*[v] = g$ is necessary in order to define the associated variational principle

$$E(u,v) = \int_a^b (v(L[u] - f) - gu)dx = \int_a^b (u(L^*[v] - g) - fv)dx \ . \tag{6.3}$$

There exist three major disadvantages with this approach: Firstly, the adjoint forcing function g has to be determined, secondly, an additional function v is introduced and has to be solved for, and, thirdly, the sign of the second variation of the variational principle may be indefinite, in which case the functional cannot be extremized.

In this regard, the alternative of maximizing $\langle R_{i-1}, \Psi_i\rangle^2$ to compute the parameters of the i-th basis function has been adopted. Since the problem is linear, the associated coefficient is readily calculated from $\langle R_i, \Psi_i\rangle = 0$.

The solution $u(x)$ is approximated by the series expansion

$$u_i^a(x) = 1 + \sum_{j=1}^{i} c_j \Phi_j(x, \bar{p}_j) \ , \tag{6.4}$$

where $\bar{p}_i$ denotes the vector of the parameters of the i-th interpolation function. Once again, B_1-splines were used, so that $\bar{p} = (x_M, \Delta x_l, \Delta x_r)_i^{\mathrm{T}}$. With Equation (6.4), the initial value is satisfied if the B_1 splines are forced to vanish at the boundary $x = a = 0$. However, the basis functions should be able to take nonzero values at the boundary $x = b = 1$. In this regard the computational domain is expanded to the interval $[0, 1 + \Delta x]$ with the basis functions vanishing at $x = b' = 1 + \Delta x$. A typical value for Δx is 0.5. The parallel direct search (PDS) technique was employed for the solution of both low-order nonlinear optimization problems.

Preliminary numerical experiments failed to converge for the second part of the optimization problem, i.e. the determination of the coefficients c_i, for $\Psi_i = \Phi_i$. This nonconvergence phenomenon could be explained by the similarity of the proposed algorithm to the Gauss-Seidel algorithm, which was addressed in Section 3.3. Specifically, if the differential Equation (6.2) is discretized with $\Psi_i = \Phi_i$, the matrix $\mathbf{A}$ of the resulting system of algebraic equations $\mathbf{A}\bar{c} = \bar{b}$ is not diagonally dominant; the Gauss-Seidel method will not converge for any initial vector $\bar{c}$. This drawback is remedied by modifying the weighting function; specifically, the weighting function $\Psi_i = \Phi_i + \frac{d\Phi_i}{dx}$ was utilized. In this regard, the Petrov-Galerkin method is adopted for the computation of the coefficient.

Figure 6.1 presents the numerical results compared to the exact solution when using 20 optimally selected basis functions. The agreement is satisfactory.

Figure 6.2 illustrates the RMS error convergence rate. The RMS error was calculated based on 101 uniformly distributed trial points. As can be seen in Figure 6.2, the convergence rate of the proposed algorithm is not as good as that obtained by the finite element method using linear Lagrangian shape functions on a uniform grid. It is well-known in computational mechanics that a numerical method is optimal when both the grid and the node coefficients are determined optimally. In this case, the basis functions are optimally selected, as can be deduced by direct examination of the convergence rate that has been obtained when the coefficients

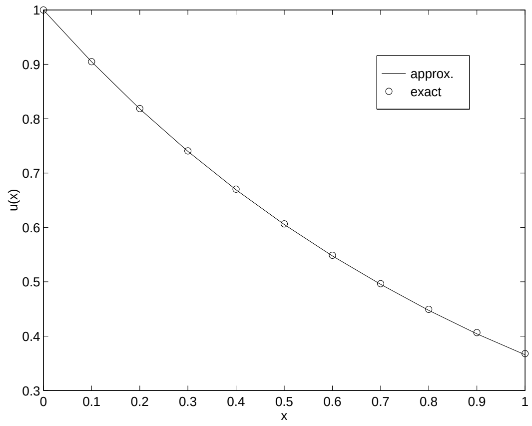

Figure 6.1 Initial value problem: exact and approximate solutions when using 20 optimal basis functions.

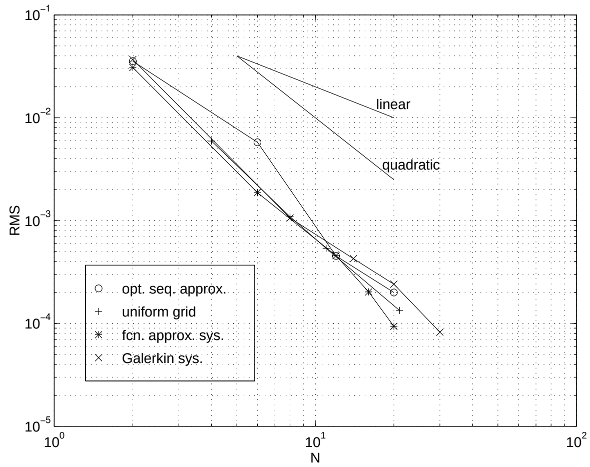

Figure 6.2 Initial value problem: convergence rate comparison.

are computed according to function approximation concepts [†]. Specifically, at each step of the algorithm i, a system of equations with size $i \times i$ is solved for the simultaneous computation of the i-th coefficient and recomputation of the previous coefficients c_j, where $j = 1, \ldots, i-1$. In this manner, the error $e_i = u_e - u_i^a$ is forced to be orthogonal to each basis function. It is obvious that for this problem the computation of each new coefficient without the adjustment of the previous coefficients is insufficient for a satisfactory convergence rate. In this regard, the Petrov-Galerkin method should be utilized for the recomputation of the coefficient vector by solving a system of equations at each step of the algorithm after the optimal selection of the new basis function. In this manner, the residual R_i is forced to be orthogonal to each basis function and not just to the latest one. The associated convergence rate, presented in Figure 6.2, confirms the preceding discussion.

However, it should also be emphasized that the proposed method exploits its potential best in problems whose solutions exhibit large gradients, like the convection-diffusion problem presented in Section 4.2.1 and the compressible flow problem of Section 5.1. Indeed, the proposed method performed best for these problems, and a high level of accuracy was achieved for a relatively low number of optimal basis functions due to the adaptive nature of the algorithm. When the problem solution lacks regions with steep gradients, the proposed method will not necessarily perform better than traditional methods.

Tables 6.1 summarizes the amount of computational work that was required by PDS for the solution of the optimization problem for the selection of the basis func-

[†]The coefficients are determined using function approximation concepts by minimizing $\int_a^b (u_e - u_i^a)^2 dx$, where u_e is the exact solution.

tion parameters. It can be concluded that the requisite computational resources do not fluctuate significantly among the steps of the proposed algorithm.

Table 6.1 Amount of computational work required by PDS for the initial value problem; optimal basis function parameters computation ($d = 100$, $tol = 10^{-4}$).

Algorithm step	# of PDS iterations	Total # of function eval.	Total # of constraint eval.
1	12	581	1031
5	14	685	748
10	10	684	717
15	13	715	746
20	10	576	605

Part II

Optimization of the Coupling Mechanism in a Viscous-Inviscid-Interaction Airfoil Analysis Code

Chapter 7

Airfoil flow analysis and design

7.1 General background

The numerical simulation of a transonic flowfield for airfoil flow analysis and design is considered a challenging task because it can involve the simultaneous modelling of subsonic and supersonic flow, shock waves, turbulence, turbulent transition, and flow separation. Two approaches exist in transonic airfoil flow analysis: one can solve either some form of the Navier-Stokes equations over the entire flow field or, alternatively, the boundary layer equations in the shear layer region of the flow and either the potential or the Euler equations in the region of the flow approximated as inviscid. Both methods have their advantages and disadvantages.

In the first approach, the direct numerical simulation (DNS) of the Navier-Stokes equations, which does not require a turbulence model, would be the ideal choice since it is arguably the most accurate existing method. Unfortunately, DNS solvers are in general impractical because of difficulties at high Reynolds numbers and large computational time and memory requirements [75]. Codes that solve high-fidelity solvable forms of the Navier-Stokes equations (which include artificial turbulence models) have been developed and are continually improving in accuracy and speed at high Reynolds numbers. Navier-Stokes solvers are especially advantageous when the flowfield experiences flow separation about the airfoil, which is likely to occur at off-design conditions.

The second approach requires an interaction scheme between a viscous and inviscid solver so that smooth transition from one region of the flow to the other is guaranteed. These viscous-inviscid-interaction (VII) codes have the main advantage of providing accurate results for certain aerodynamic problems in less computational time than the high-fidelity Navier-Stokes methods [39, 55]. The primary disadvantage of VII codes is that they cannot model conditions with extensive flow separation. However, since on-design conditions usually include the reduction or elimination of flow separation, VII codes are especially useful in analyses near the final airfoil design.

While the continual progress and rapid growth of supercomputer speed, as well as the development of parallel processing technology, motivate research on full Navier-Stokes methods [53, 20], there is no indication that these same technological advances could not be implemented in an existing VII code. It will be shown that rather than directly applying parallel processing methods to the boundary layer and Euler codes, the parallel direct search (PDS) can be used to enhance the functionality of the interaction scheme. The speed-up of this enhanced code combination can be controlled by the number of processors and design variables without adversely affecting the accuracy of the results.

7.2 The VII code

The VII code, whose interaction mechanism was improved, consists of an inviscid flow code for the solution of the two-dimensional unsteady Euler equations and an efficient and robust viscous flow code for the solution of the two-dimensional, steady, compressible, turbulent boundary-layer equations. The former utilizes a floating shock fitting technique which, combined with an implicit upwind numerical scheme, allows accurate calculations on nonadaptive grids [36, 37], and the latter is based on the semi-discrete Galerkin method [13, 58, 59].

The classical interaction approach is to 1) obtain a solution to the inviscid flow approximation, 2) extract the velocity and pressure fields from the inviscid solution and use them as external conditions in the viscous layer approximation, 3) extract displacement thickness from the viscous solution, and 4) use the displacement thickness to modify the original body geometry and obtain another estimate of the inviscid flow. These iterative cycles, depicted in Figure 7.1, continue until the inviscid and viscous solutions are converged and compatible.

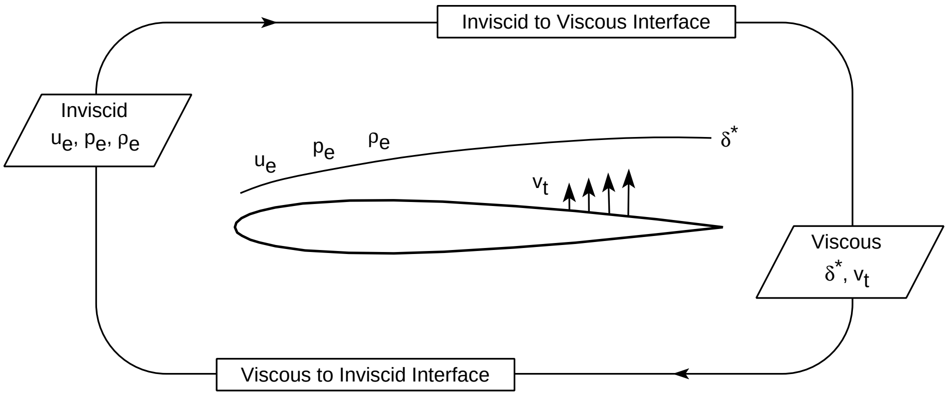

Figure 7.1 The VII global iteration scheme.

The above combination of codes and coupling mechanism is by no means unique. Different viscous and inviscid flow codes can be combined with a number of different interaction mechanisms. Two of these distinct combination examples can be found in references [65] and [82].

An alternative to adding the displacement thickness distribution to the original body thickness is to impose a transpiration velocity boundary condition at the body surface in the inviscid flow solver. The transpiration velocity is an inviscid normal-velocity boundary condition which is imposed at the body surface to simulate the displacement of the inviscid flow by the viscous flow momentum defect

[54]. An expression for the transpiration velocity can be obtained by integrating the difference between the inviscid and viscous continuity equations across the boundary layer while applying the Prandtl boundary layer and uniform inviscid-flow assumptions [11]

$$v_t = \frac{1}{\rho_e} \frac{\partial(\rho_e u_e \delta^*)}{\partial x} \;, \tag{7.1}$$

where ρ_e, u_e, δ^*, and x, are the density at the edge of the boundary layer, the velocity at the edge of the boundary layer, the displacement thickness (the distance from the body to the edge of the boundary layer), and the independent spatial variable, respectively. Using the compressible Dorodnitsyn formulation, the relation can be rewritten as

$$v_t = u'_e \frac{d\delta'^*}{dx'} + \left(\frac{du'_e}{dx'} + \frac{u'_e}{x'} \frac{d\rho'_e}{dx'} \right) \delta'^* \;, \tag{7.2}$$

where the superscript $'$ denotes nondimensionalization [13, 59]. From a physical perspective, the transpiration velocity distribution should result in an inviscid streamline that is coincident with the height of the effective body obtained by the displacement thickness approach. The advantages of the transpiration velocity approach are that the inviscid grid does not need to be regenerated after every viscous iteration and that the interaction always allows a smooth transition to separation [51]. The coupling mechanism employed in the given VII code is based on the transpiration velocity concept.

A number of physical quantities are considered during the process of airfoil flow analysis and design. Design decisions need to be taken according to some standardized form of these quantities, the so-called aerodynamic coefficients (Appendix

A.7). In addition, numerical methods can only be validated by comparing the obtained results to experimental data and/or results obtained from previously validated numerical methods. In the given VII code, the aerodynamic lift and drag coefficients are calculated by numerically integrating the pressure coefficients obtained from the Euler solver and the friction coefficients obtained from the boundary layer solver. Naturally, both aerodynamic coefficients considered here are affected by the performance of the inviscid and viscous flow codes which in turn depend on their interaction. It is obvious that the functionality of the coupling mechanism is quite significant to the stability, efficiency, and accuracy of the VII code [55, 11].

Remarks

1. The Euler code [36, 37] was initially developed for the solution of the general two-dimensional, unsteady Euler equations. Its efficiency was improved by adjusting it to acount for airfoil analysis only, and by integrating a steady-state detection routine in order to avoid unnecessary iterations.

2. The parallelization of the Euler and boundary layer solvers has been considered. It was found that a potential parallelization of the the Euler code would not be worth the effort due to the considerable amount of requisite work considering the relatively low expected speed-up. The boundary layer code, however, should and could be parallelized due to its "embarrassingly parallelizable", as referred to in the concurrent computing terminology, parabolic character. In an initial effort, the space marching computations over the two (upper and lower) sides of the airfoil were executed synchronously by means of the parallel virtual machine (PVM) routines library [28]. However,

PDS views the VII code as a “black-box” that needs to be executed many times in order to obtain the necessary objective function values; each available processor is assigned with one function evaluation, i.e. one sequential execution of the Euler and boundary layer solvers. In this regard, the VII code cannot be processed concurrently. Therefore, advanced parallelization attempts were aborted.

Chapter 8

The optimization of the interaction mechanism

As mentioned in the previous chapter, the coupling mechanism between the viscous and the inviscid regions of the flow is based on the transpiration velocity concept. Since the transpiration velocity is numerically evaluated for a finite number of points on the airfoil and its wake due to the discretization procedure, the term transpiration velocity vector is adopted. The transpiration velocity vector imposes a boundary condition for the Euler solver, and is assumed to be initially equal to zero. The first transpiration velocity vector with nonzero entries is obtained from the first execution of the boundary layer code and, in the original version of the VII code, undergoes a strong ($\omega \leq 0.1$) underrelaxation according to the formula

$$v_{t,rel}^{n} = \omega v_{t}^{n} + (1-\omega) v_{t,rel}^{n-1} \ , \tag{8.1}$$

where v_t, $v_{t,rel}$, and ω represent the transpiration velocity vector, the relaxed transpiration velocity vector, and the relaxation coefficient, respectively. The superscripts indicate the global VII iteration number [‡]. This relaxation is necessary since the direct use of the calculated transpiration velocity was found to cause numerical instabilities which prevented the VII code from converging. The relaxation concept is a tool used often by CFD researchers [84].

Since the transpiration velocity concept in nonadaptive grid methods is analogous to the effective body concept in adaptive grid methods, convergence of the

[‡]For the sake of simplicity, the serial execution of the Euler and the boundary layer codes will be referred to as a "black-box" run or a "global VII iteration" in the rest of the text.

transpiration velocity to its steady-state value corresponds to convergence of the effective body to its final geometric shape, and therefore yields smooth transition between the two different regions of the flow. In this regard, the objective is to minimize the difference between the transpiration velocity of two consecutive global VII iterations. Given the fact that the coupling mechanism strongly affects the number of global VII iterations necessary for convergence, an accelerated convergence of the transpiration velocity to its steady-state value that will provide smooth transition from the boundary layer to the inviscid flow region is desired. Clearly, an optimization problem needs to be defined and solved.

8.1 The formulation of the optimization problem

The strategy of utilizing PDS to solve the coupling mechanism optimization problem is the following [60]. Firstly, the objective function is defined as

$$f = \sum_{i=1}^{N} (v_i^{output} - v_i^{input})^2 \ , \tag{8.2}$$

where $\bar{v}^{input}$ and $\bar{v}^{output}$ are the transpiration velocity vectors before and after a global VII iteration, respectively. By examining Equation (7.2), it becomes obvious that the transpiration velocity depends both implicitly and explicitly on a number of dependent and independent variables; the idea of using PDS as a derivative-free optimization method is attractive.

As for the original VII code, zero transpiration velocity is assumed in the first execution of the Euler and the boundary layer solvers. The first nonzero transpiration velocity vector is relaxed with a relaxation coefficient of $\omega = 0.1$, and constitutes the initial guess for the optimization problem to be solved. PDS creates a number of perturbed transpiration velocity vectors, which in turn form different Euler code normal velocity boundary conditions. A global VII iteration is executed for all these input transpiration velocity vectors. The objective function is then evaluated, i.e. the difference between every single input transpiration velocity vector and its corresponding output transpiration velocity vector is calculated, and, finally, the input transpiration velocity vector which provides the smallest difference is chosen as the initial guess for the next PDS iteration. Figure 8.1 illustrates the PDS iteration scheme. On the left side of each PDS iteration box, there is an input transpiration velocity vector, in the box there are m function evaluations for m perturbed transpiration velocity vectors, and on the right side of each box there is the chosen output transpiration velocity vector, which becomes the initial guess for the next PDS iteration.

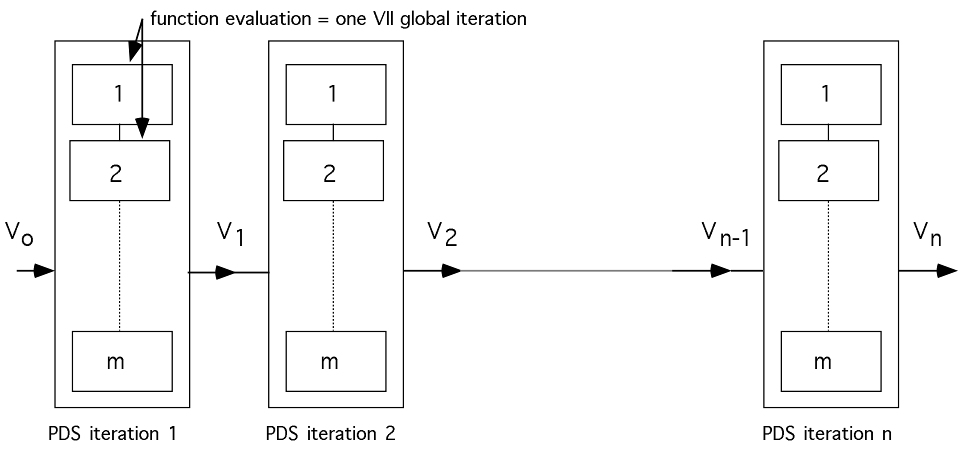

Figure 8.1 The PDS iteration scheme.

Initial numerical experimentations included 161 airfoil, wake, and farfield grid points. However, for stability reasons, the transpiration velocity values for the 14 grid points assigned to the farfield were set equal to zero. Thus, an initial number of 147 design variables was formed by the full (in the sense of non-zero entries) transpiration velocity vector. PDS was executed in parallel by means of the message passing interface (MPI) environment [32]. Although PDS performs best for small numbers of optimization variables, preliminary numerical tests clearly showed the potential of the method. However, when the optimization problem is unconstrained, PDS requires at least $2n$ function evaluations per iteration, i.e. $m \geq 2n$, where n is the number of optimization variables; even parallel computers with a large number of available processors would not be capable to achieve satisfactory speed-up.

8.2 Reducing the number of variables

The necessary reduction of the number of the optimization variables has been achieved by means of function approximation concepts [60].

A one-dimensional finite element code, based on the least-squares method, has been developed in order to interpolate the available data, namely the transpiration velocity values corresponding to the finite number of grid points on the airfoil and its wake, with a small number of nodes. An additional routine has been developed for the reconstruction of the transpiration velocity vector to its original size. In this manner, the interpolated transpiration velocity vector was optimized, while the reconstructed transpiration velocity vector was used by the VII code. In other words, the original transpiration velocity vector $\bar{v}$ of length N has been approximated by

$$\bar{v_t} \approx v_t^a(x) = \sum_{j=1}^{M} c_j \Phi_j(x) \ , \tag{8.3}$$

where M is the number of nodes, $\Phi_j(x)$ are the interpolation functions, and c_j the corresponding coefficients. Quadratic Lagrangian shape functions were used since least-squares finite element methods are known to perform better when higher order functions are employed [8]. Consequently, the number of the optimization variables was reduced from N to M. The choice of the exact number of nodes used to approximate the transpiration velocity will be discussed by means of specific numerical examples in Chapter 9.

8.3 Convergence criteria

An important issue, which needs to be addressed, is the selection of the convergence criterion for the enhanced VII code. In the original code, the difference of the aerodynamic coefficients before and after a global VII iteration formed the execution terminating criterion. However, the convergence criterion of the PDS algorithm is based on the relative change in the values of the optimization variables in every PDS iteration and not of the aerodynamic coefficients. In addition, numerical instabilities in the Euler code were encountered during numerous numerical tests which primarily affected the values of the lift coefficient and the pressure contribution to the drag coefficient [§]. Fortunately, the viscous contribution to the drag coefficient (obtained from the boundary layer code) was found to be consistently stable. As a result, two convergence criteria can be used, namely the evolution of the viscous contribution to the drag coefficient and/or the change of the objective function used by PDS.

[§]This phenomenon was also observed during the execution of the original version of the code, where it caused the values of the aerodynamic coefficients to oscillate and decelerated convergence.

Chapter 9

Numerical results

Numerical experiments were performed by simulating two-dimensional, transonic, turbulent, attached flow around the numerically and experimentally well-studied NACA-0012 airfoil and the supercritical RAE-2822 airfoil [60].

In each of the cases, a 161×33 C-grid was used by the Euler code, as illustrated in Fig 9.1. For both the NACA-0012 and the RAE-2822 airfoils, there are 80 points per airfoil side stretching in the x-direction of the grid; 66 stretch over the airfoil ($0 < x \leq 1$), 7 over the wake ($1 < x \leq 2.2$), and 7 over the farfield of the grid ($2.2 < x \leq 6$). There is one common grid point for both sides at $x = 0$. The previously mentioned zero entries of the transpiration velocity vector for each side of the airfoil are the ones lying between $x = 2.2$ and $x = 6$, and are excluded from the function range to be approximated. The number of the nodes used to approximate the original function can be chosen by the user. Numerical tests have been performed for 5 and 9 nodes per airfoil side, respectively. The 5 nodes version approximates and reconstructs the transpiration velocity values in the range $0 \leq x \leq 1$ and the 9 nodes version those in the range $0 \leq x \leq 2.2$. Therefore, only the 9 (18) nodes version takes the wake region into consideration. Using 10 (5 for each side) and 18 (9 for each side) optimization variables results in at least 20 and 36 function evaluations per PDS iteration, respectively. In this regard, if the enhanced code is executed on parallel computers with a number of available processors at least as high as the number of the required function evaluations, it will converge in less computational time than the original serial code.

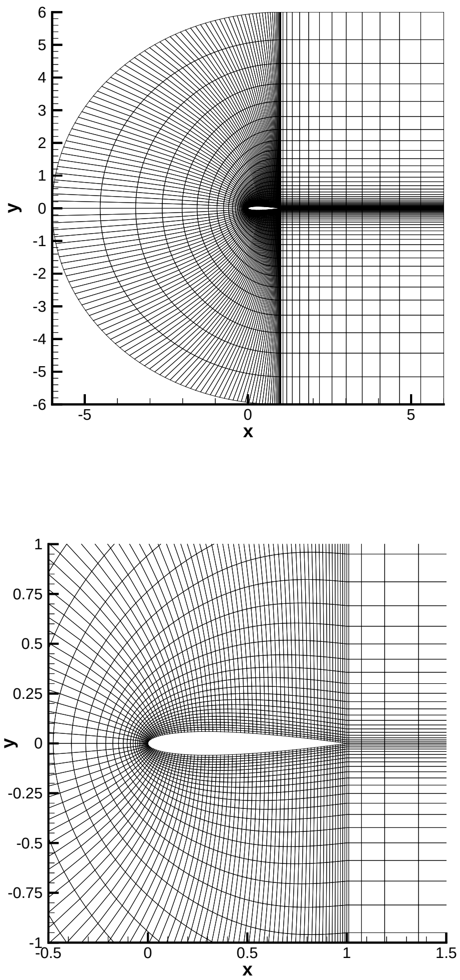

Figure 9.1 The computational grids for the (left) NACA-0012 and (right) RAE-2822 airfoils; (top) entire mesh and (bottom) zoom-in.

Three test cases were selected for comparison purposes from references [35] and [19], and are presented in Table 9.1, where α_{num} stands for the numerical angle of attack, α_{exp} for the corresponding experimental angle of attack, M_∞ for the freestream Mach number, Re for the chord based Reynolds number, $\frac{x}{c}$ for the location of the numerically tripped transition to turbulence (activation of numerical turbulence modelling), and ω for the relaxation parameter in the original version of the code. The turbulence model utilized by the boundary layer code is described in reference [44] and validated in references [39] and [13].

Table 9.1 The VII numerical test cases.

Case	airfoil	α_{num}	α_{exp}	M_∞	Re	$\frac{x}{c}$	ω
1	NACA-0012	1.37	1.86	0.700	9.0×10^6	0.00	0.1
2	RAE-2822	1.90	2.40	0.676	5.7×10^6	0.11	0.05
3	RAE-2822	2.10	2.55	0.725	6.5×10^6	0.03	0.05

9.1 NACA-0012 airfoil

At first, 9 nodes per airfoil side have been used to approximate and reconstruct the transpiration velocity vector in the range containing both the airfoil and its wake. The evolution of the aerodynamic coefficients and objective function values is presented in Table 9.2. It can be seen that convergence, based on the change of the objective function, is achieved after 4 PDS iterations, satisfying a tolerance of 0.0001 in absolute difference. In this Table, C_{D_p} is the pressure contribution to the drag coefficient and C_{D_v} is the friction contribution to the drag coefficient. The drag coefficient C_D is given in counts, where 1 count equals 0.0001. The aerodynamic coefficients strongly oscillate but it can be assumed, by observing the values obtained after 10 PDS iterations, that they eventually converge. However, due to the repeatedly observed numerical instabilities in the Euler code and their influence on the aerodynamic coefficients, one can rely on the objective function change as the convergence criterion for this case.

Table 9.2 Aerodynamic coefficients and objective function for test case # 1 using 18 nodes to approximate the transpiration velocity.

# of PDS iterations	C_L	C_D	C_{D_p}	C_{D_v}	Obj. function value
1	0.247	59	-5	64	0.0055
2	0.263	61	-3	64	0.0033
3	0.272	62	-1	63	0.0024
4	0.251	88	25	63	0.0023
10	0.246	84	21	63	0.0019
Experiment	0.241	77			
Original code	0.239	64			

For the sake of comparison, and in order to draw conclusions about the effect of the wake transpiration velocity on the coupling mechanism, 5 nodes per airfoil side were used to approximate and reconstruct the transpiration velocity vector in a range containing only the airfoil. Convergence, based on the change of the objective function, was achieved after 5 PDS iterations. The aerodynamic coefficients and objective function values after 5 and 10 PDS iterations are tabulated in Table 9.3. An overestimation of the lift coefficient C_L is observed. However, interestingly enough, the drag coefficient comes closer to the experimental value, and the objective function values are an order of magnitude smaller than those from the 18 nodes approximation.

In Figure 9.2, the distributions of the pressure coefficients (over both, upper and lower, airfoil surfaces) obtained by optimizing the coupling mechanism using 18

Table 9.3 Aerodynamic coefficients and objective function for test case # 1 using 10 nodes to approximate the transpiration velocity.

# of PDS iterations	C_L	C_D	Obj. function value
5	0.250	72	0.00099
10	0.259	70	0.00083
Experiment	0.241	77	
Original code	0.239	64	

and 10 nodes, respectively, are compared to those obtained from the original code and to experimental data. The match is excellent, and the tiny scatter near the trailing edge of the airfoil when using 10 nodes to approximate the transpiration velocity has been observed in numerous illustrations presented by various CFD researchers in reference [39]. One possible explanation for this phenomenon is the displacement action of the boundary layer [79].

Figure 9.3 illustrates the Mach-isolines around the airfoil obtained from the original and the enhanced codes when using 18 nodes to approximate the transpiration velocity. The match is satisfactory both qualitatively and quantitatively.

As shown, convergence is achieved for the NACA-0012 airfoil test case after 4 and 5 PDS iterations when using 10 and 18 nodes to approximate the transpiration velocity vector, respectively. Having in mind that one PDS iteration requires at least (ideally) 20 and 36 function evaluations, respectively, whether or not the enhanced code will be faster than the original one depends on the number of available processors. Considering that the original code with the relaxation scheme required 10 global iterations for convergence, it is concluded that a parallel computer with a number of processors available at least as high as the number of the required

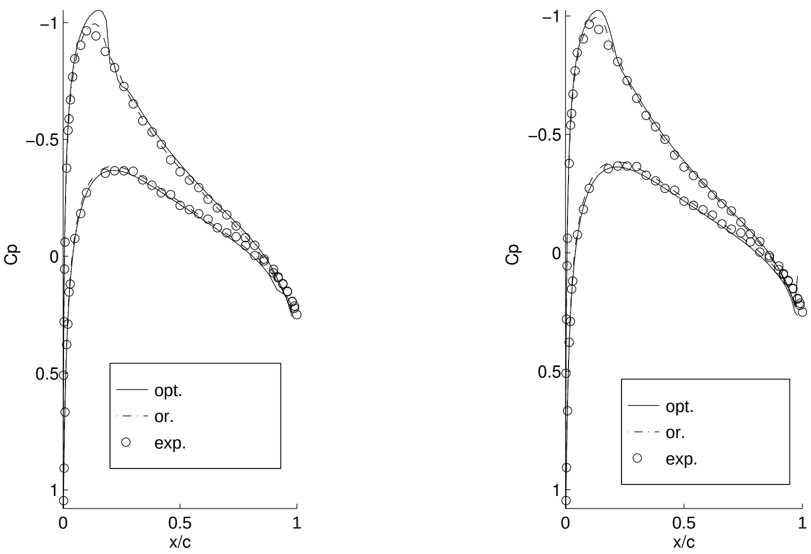

Figure 9.2 Pressure coefficients distribution over both sides of the NACA-0012 airfoil using (left) 18 and (right) 10 nodes to approximate the transpiration velocity (Opt. ≡ optimized code, or. ≡ original code, exp. ≡ experimental data).

function evaluations will execute the enhanced code in less time. Table 9.4 summarizes the above conclusions for all the test cases and confirms the success of the enhanced code. The formula used to calculate the maximum (ideal) speed-up reads as

$$S = \frac{i_{OR}}{\mathrm{int}(\frac{2n}{p})i_{PDS}} \ , \tag{9.1}$$

where S is the speed-up, i_{OR} is the number of global VII iterations required for convergence by the original code, i_{PDS} is the number of PDS iterations required for convergence by the enhanced code, n is the number of nodes in the transpiration velocity approximation, and p is the number of processors available. The function

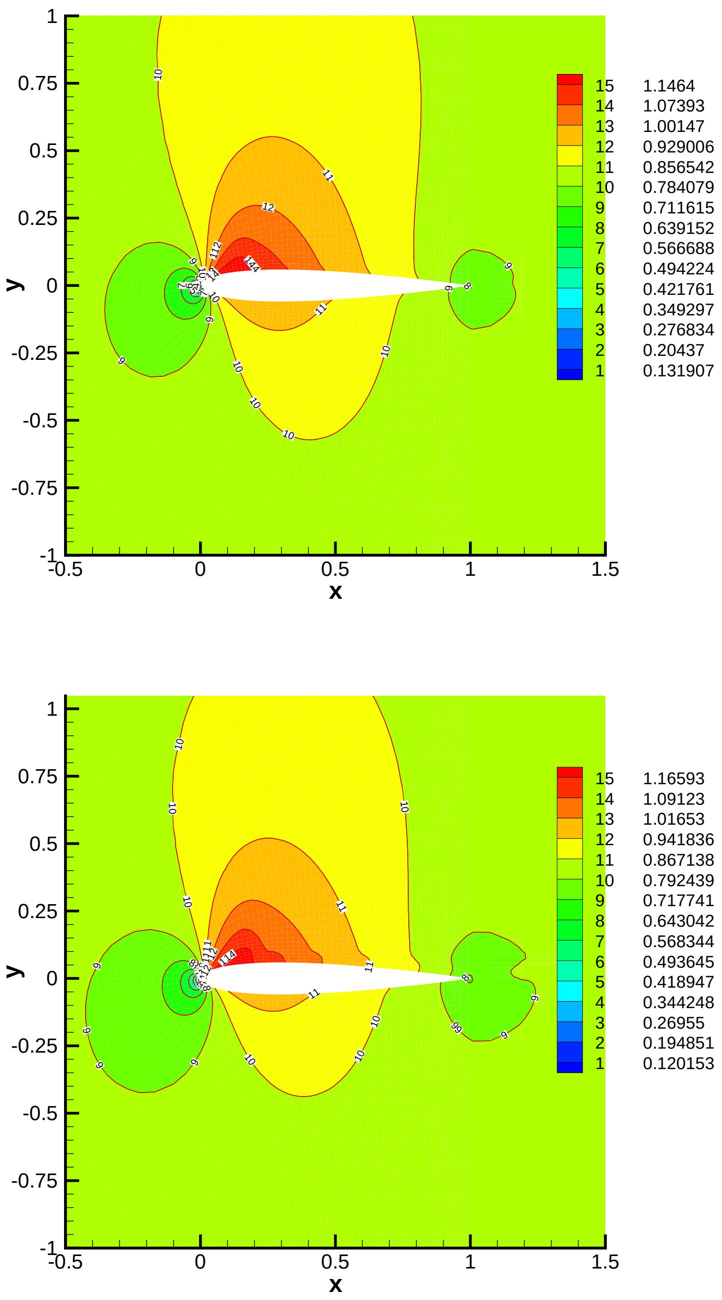

Figure 9.3 The Mach-isolines around the NACA-0012 airfoil obtained from (top) the original code and (bottom) the enhanced code when using 18 nodes to approximate the transpiration velocity.

int forces up-rounding if the result of the division $2n/p$ has a remainder. In this manner, the theoretically maximum speed-up of infinity (∞) as $2n/p$ approaches 0 is replaced by the in practice maximum attainable speed-up i_{OR}/i_{PDS} since $2n/p$ can not be smaller than 1. It is obvious that the ratio $\frac{2n}{p}$ is primarily important for the speed-up, while the ratio $\frac{i_{OR}}{i_{PDS}}$ is secondary.

Table 9.4 Speed-up comparisons.

Case	Original iterations	Approx. nodes	Enhanced iterations	Speed-up with p processors				
				4	8	16	32	64
1	10	18	4	0.27	0.5	0.83	1.25	2.5
1	10	10	5	0.4	0.67	1.0	2.0	2.0
2	41	18	5	0.91	1.64	2.73	4.1	8.2
2	41	10	6	1.36	2.27	3.41	6.83	6.83
3	42	18	5	0.93	1.68	2.8	4.2	8.4
3	42	10	6	1.4	2.33	3.5	7.0	7.0

9.2 RAE-2822 airfoil

The RAE-2822 airfoil is a supercritical airfoil with a moderate amount of aft camber which poses a challenge in achieving VII convergence. In addition, the test cases (summarized in Table 9.1) simulate critical transonic flow conditions. Therefore, these test cases constitute an excellent validation criterion for the enhanced code.

For test case 2 of Table 9.1, convergence of the aerodynamic coefficients (with a tolerance of 0.001 in absolute difference) has been achieved after 5 PDS iterations using 18 nodes and after 6 PDS iterations using 10 nodes to approximate the transpiration velocity, while the original code required 41 global iterations. As can be seen in Table 9.5, the aerodynamic coefficients are in satisfactory agreement with experimental values using 18 nodes, while they diverge from the experimental values when using 10 nodes. The conclusion one has to draw from that observation, is that the transpiration velocity values in the wake are not to be neglected in such critical transonic flow conditions when a supercritical airfoil is used. In addition, it should be noted that the pressure contribution to the drag coefficient, especially when using 10 nodes, is the one that causes the large errors in the aerodynamic coefficients.

Table 9.5 Aerodynamic coefficients and objective function for test case # 2.

# of nodes of approximation	# of PDS iterations	C_L	C_D	Obj. function value
18	5	0.577	54	0.00012
10	6	0.634	16	0.00009
Experiment		0.566	85	
Original code		0.554	72	

It is interesting to observe, as Figure 9.4 indicates, that while the aerodynamic coefficients do not match exactly the experimental values, the pressure coefficient distributions are in satisfactory agreement. The reason for the aerodynamic coefficients mismatch might lie in the method utilized for their calculation; as mentioned before, aerodynamic coefficients are calculated based on the pressure and friction coefficients distributions without taking into account the wake effects, while it is customary in experimental aerodynamics to consider the wake effects in great detail.

The original code required 42 global iterations to achieve convergence for test case 3 of Table 9.1. However, the enhanced code required only 5 PDS iterations

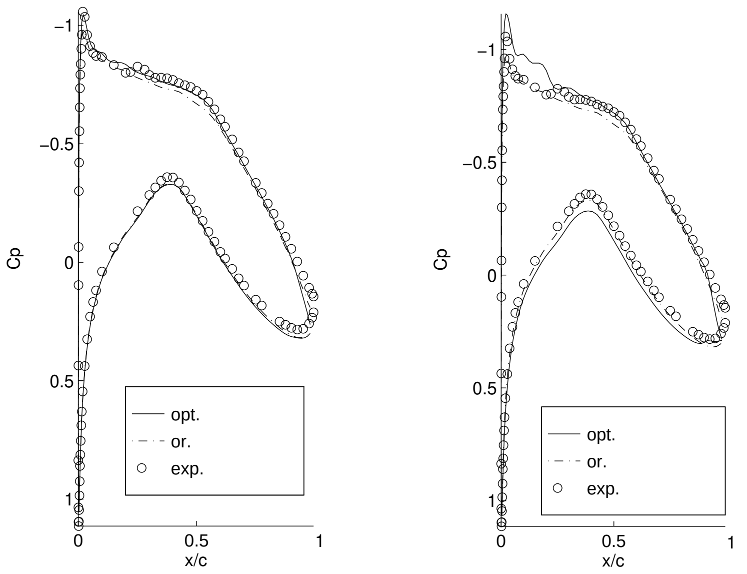

Figure 9.4 RAE-2822 airfoil, test case # 2: pressure coefficients distribution over both sides using (left) 18 and (right) 10 nodes to approximate the transpiration velocity.

when using 18 nodes and 6 PDS iterations when using 10 nodes to approximate the transpiration velocity. Table 9.6 summarizes the aerodynamic coefficient and objective function values evolution.

In Figure 9.5, the pressure and friction coefficients distributions obtained from the enhanced code are compared to those obtained from the original code and to experimental values. The match is satisfactory with the exception of the shock wave location prediction. However, Figure 9.6 presents two figures reproduced from reference [39] which illustrate the difficulties that CFD researchers face in predicting the location of the shock waves and handling the induced boundary layer separations. The curves represent the distributions of pressure coefficients and friction coefficients obtained from various VII and Navier-Stokes codes for flow

Table 9.6 Aerodynamic coefficients and objective function for test case # 3.

# of nodes of approximation	# of PDS iterations	C_L	C_D	C_{D_p}	C_{D_v}	Obj. function value
18	2	0.724	76	12	64	0.00032
18	3	0.723	80	17	63	0.00026
18	4	0.722	75	12	63	0.00015
18	5	0.720	76	13	63	0.00014
10	1	0.726	77	14	63	0.00024
10	2	0.731	76	13	63	0.00017
10	3	0.743	74	11	63	0.00013
10	4	0.738	73	10	63	0.00011
10	5	0.740	74	11	63	0.000109
10	6	0.742	75	12	63	0.000107
Experiment		0.658	107			
Original code		0.657	80			

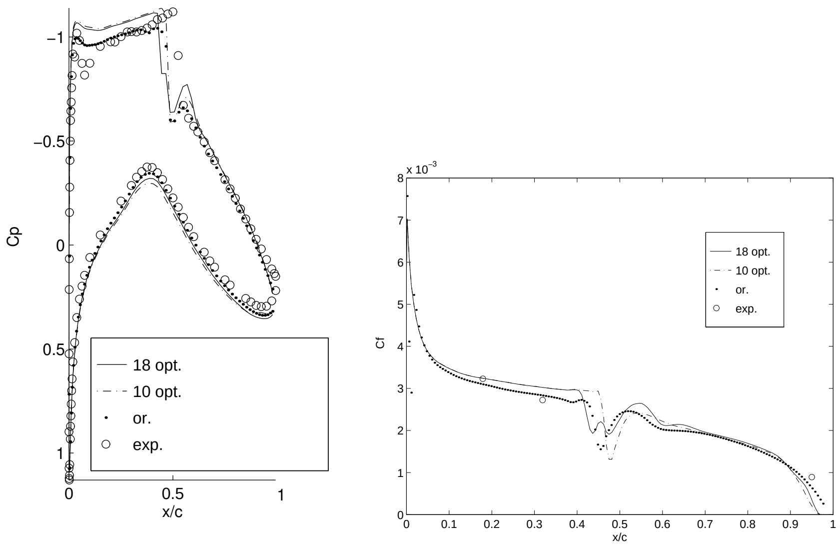

Figure 9.5 RAE-2822 airfoil, test case # 3: (left) pressure and (right) friction coefficients distribution over both sides and the upper side, respectively.

conditions that differ slightly in the angle of attack (α_{num} = 2.92 instead of α_{num} = 2.55) from those of test case 3. Considering the results presented in reference

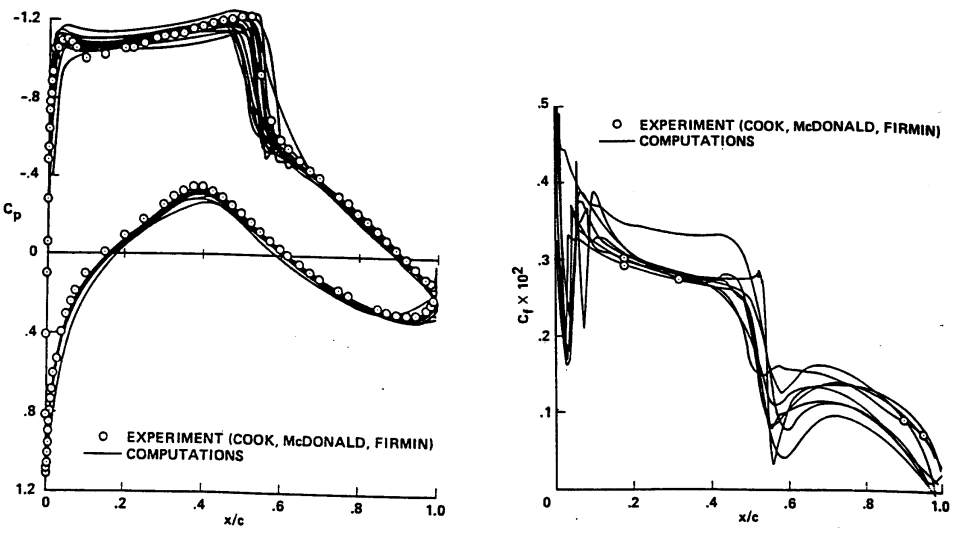

Figure 9.6 Results reported for the RAE-2822 airfoil, test case # 3 (with slightly different α_{num}), from various CFD researchers in reference [39]: (left) pressure and (right) friction coefficients distribution over both sides and the upper side, respectively.

[39], it can be concluded that VII codes tend to underpredict, while Navier-Stokes codes tend to overpredict, the location of the shock wave.

Chapter 10

Conclusions

The objective of the work presented in this dissertation is the utilization of parallel optimization in computational fluid dynamics as an alternative to the direct application of concurrent computing methodologies to fluid mechanics problems. In particular, numerical methods and algorithms are developed for the solution of differential equations and the enhancement of an airfoil flow analysis code by utilizing the parallel direct search (PDS) optimization technique.

10.1 Optimal incremental approximation

In Part I of the dissertation, a computational method for optimal incremental function approximation is proposed for the solution of time-invariant differential equations. The development of the method is motivated by a number of concepts that are utilized by numerical techniques employed in artificial neural networks and computational mechanics.

In particular, iterative approximation concepts originating in artificial neural networks are combined with error minimization ideas from function approximation theory to formulate an algorithm; this algorithm involves the incremental determination of the basis functions and the associated coefficients used in the series expansion for the representation of the solution. The proposed algorithm requires the solution of a nonlinear optimization problem at each step. Variational principles, associated with the given differential equations and the boundary conditions, are utilized to define the objective function(al). The major advantage in the use of variational principles is that the latter succinctly and concisely represent governing equations and associated boundary conditions in a single variational integral which is then extremized. In this manner, the well-posed formulation of the optimization problem is guaranteed. Computational efficiency is achieved by utilizing PDS for the solution of the optimization problem and using B_1-splines as basis functions.

The proposed method is adaptive in nature although a grid is neither built nor adapted in the traditional sense. It can be classified as an h-r adaptive technique. The computational overhead of the a-posteriori error estimation and adaptive process, shared by traditional adaptive and multigrid techniques, is avoided; the "grid", defined by the location of the optimally selected basis functions, evolves with the solution. Moreover, additional disadvantages of traditional adaptive unstructured grid techniques are not present in the proposed method; complicated data structures are not required since systems of equations are neither assembled nor solved, and numerical stability issues are limited to the nonlinear optimization problem.

If the optimization problem can be decomposed (e.g. in linear problems), the affinity of the method to that of weighted residuals (in particular to the Galerkin method), becomes apparent. The proposed algorithm is iterative in nature and may be viewed as a nonstationary version of the Gauss-Seidel method for the solution of systems of algebraic equations. It should be mentioned that although the basis functions are locally defined, they have a tendency to encompass a fair amount of the computational domain and overlap, adding a strong spectral character to the method. However, for problems whose solutions have large gradients, the basis functions are clearly concentrated in the regions of steep gradients; the proposed method performs best by fully exploiting its adaptive nature.

It is shown that the proposed method can be successfully applied to a range of steady-state problems, including one- and two-dimensional, linear self- and nonself-adjoint, and nonlinear differential equations associated with Dirichlet and Neumann types of boundary conditions. Emphasis is given to fluid dynamics applications. Accurate numerical results and satisfactory convergence rates are reported. Specifically, the accuracy level of the presented numerical results is quite satisfactory considering the relatively low number and order of the optimal basis functions used in the solution representation. Moreover, the proposed method does not face the difficulties that traditional methods do in some applications, for example the oscillations observed in the finite difference and finite element solutions for the linear convection-diffusion problem presented in Section 4.2.1 and the failure of the Newton-like algorithm to converge for critical Mach numbers in the two-dimensional flow problem of Section 5.1. The obtained convergence rates are at least superlinear and at most superquadratic and better than those obtained by using uniform grids for problems whose solutions exhibit large gradients.

Computational cost is reduced and efficiency is enhanced by solving low-dimensional optimization problems using the parallel direct search technique. Moreover, the optimization problem may be further decomposed in linear applications. Readily parameterized B_1-splines, and their bilinear tensor products in two-dimensions, are used to ensure simplicity and computational efficiency. However, a broad class of functions (e.g. low-order polynomials, higher-order B-splines, and radial basis functions) can be utilized. Parameter studies demonstrate that the computational work required by PDS for the solution of the nonlinear optimization problems, does not increase as the algorithm advances. The numbers of required PDS iterations and function and constraint evaluations do not fluctuate dramatically. It should be noted that the PDS technique is quite efficient and successful, considering the fact that it was designed for function and not functional optimization.

The proposed method is based on the utilization of natural and contrived variational principles. Such variational formulations exist for many problems of practical interest in engineering and have been studied extensively. However, there still exist problems for which variational principles have not yet been defined. It is demonstrated how alternative formulations of optimization problems can be adopted by the proposed method for the determination of the optimal basis functions and the associated coefficients.

Future research work could be focused on the extension of the method to three-dimensional and/or unsteady problems including irregular boundaries.

10.2 Viscous-inviscid-interaction code enhancement

In Part II of the dissertation, a viscous-inviscid-interaction (VII) code, consisting of an Euler-solver and a boundary-layer-solver, is enhanced by optimizing the functionality of its coupling mechanism.

In particular, the VII mechanism is improved by accelerating the convergence of the transpiration velocity, on which the concept of the mechanism is based, to steady state. For this purpose, an associated nonlinear optimization problem is formulated and solved by means of PDS.

It is shown, by means of transonic airfoil flow simulations, that the enhanced code achieves convergence in fewer computational time than the original code. However, a strong dependency on the number of available processors exists; the ratio of the number of the necessary function evaluations to the number of available processors has to be as small as possible for maximal speed-up. In this regard, function approximation concepts are utilized to reduce the number of variables in the optimization problem.

The quality of the results obtained from the enhanced code is maintained at the level of quality of the results obtained from the original code. It is observed that for the NACA-0012 airfoil the speed-up is moderate compared to that of the two RAE-2822 airfoil test cases; the latter are challenging in terms of both transonic flow conditions and supercritical airfoil geometry. Moreover, speed-up can be improved by simultaneously increasing the number of nodes used in the function approximation and the number of available computational units; in this manner, greater accuracy is achieved in fewer iterations.

The physical effects of the wake are not taken into account during the aerodynamic coefficients calculation, resulting in some mismatches with the experimental values. However, considering the fidelity of the VII model and the simplicity of the enhanced optimization algorithm, the quality of the numerical results can be characterized as satisfactory.

The work presented in this dissertation is focused on the analysis of airfoil flow. However, the proposed algorithm can be readily implemented for design purposes. Specifically, the mathematical model, i.e. the objective function of the optimization problem, can be modified for the solution of so-called inverse design problems. For example, the objective function can be defined such to minimize the difference between the actual and target value of some physical magnitude of interest, e.g. pressure.

It can be concluded that the enhanced VII code constitutes a useful airfoil flow analysis and design tool and an acceptable alternative to more costly and less efficient high-fidelity codes.

Bibliography

[1] J.D. Anderson, Jr. *Fundamentals of Aerodynamics.* McGraw Hill, 1990. Page 21.

[2] J.D. Anderson, Jr. *Modern Compressible Flow.* McGraw Hill, 1990. Pages 242-250.

[3] R. Barrett, M. Berry, T.F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods.* SIAM Publications, 1994. Pages 5-7.

[4] A.R. Barron. "Universal approximation bounds for superpositions of a sigmoidal function". *IEEE Transactions on Information Theory*, 39(3):930–945, 1993.

[5] H. Bateman. "Irrotational motion of a compressible inviscid fluid". *Proc. Nat. Acad. Sci*, 16:816–825, 1930.

[6] M. Becker. *The Principles and Applications of Variational Methods.* MIT Press, 1964. Research Monograph No. 27, pages 7-10.

[7] E.W. Cheney. *Approximation Theory.* McGraw Hill, 1966. Pages 3-6.

[8] T.J. Chung. *Finite Element Analysis in Fluid Dynamics.* McGraw Hill, 1977. Page 43.

[9] S. Cox and M. Overton. "Designing for optimal energy absorption - III Numerical minimization of the spectral abscissa". *Structural Optimization*, 13(1):17–

22, February 1997. Journal of the International Society for Structural and Multidisciplinary Optimization (ISSMO), Springer Verlag.

[10] G. Cybenko. "Approximations by superpositions of a sigmoidal function. *Math. Contr. Signals, Syst.*, 2:303–314, 1989.

[11] W.R. Dalsem, J.L. Steger, and K.V. Rao. "Some experiences with the viscous-inviscid interaction approach". Technical Report TM 100015, NASA, 1987. Pages 1-44.

[12] A.J. Davies. *The Finite Element Method, a First Approach.* Clarendon Press, 1980. Pages 48-58.

[13] B.A. Day. "Semi-discrete Galerkin solution of the compressible boundary-layer equations with viscous-inviscid interaction". Master's thesis, Rice University, Houston, Texas, December 1992.

[14] L. Demkowicz and J.T. Oden. "On a mesh optimization method based on a minimization of interpolation error". *Int. J. Engng. Sci.*, 24(1):55–68, 1986.

[15] J.E. Dennis, Jr. and V. Torczon. "Direct search methods on parallel machines". *SIAM Journal of Optimization*, 1(4):448–474, November 1991.

[16] R.A. DeVore. "Degree of nonlinear approximation". In C. K. Chui, L. L. Schumaker, and J. D. Ward, editors, *Approximation Theory VI*, pages 175–201. Academic Press, 1989.

[17] A.R. Diaz, N. Kikuchi, and J.E. Taylor. "A method of grid optimization for finite element methods". *Computer Methods in Applied Mechanics and Engineering*, 41:29–45, 1983.

[18] A.R. Diaz, N. Kikuchi, and J.E. Taylor. "Optimal design formulations for finite element grid adaptation". In V. Komokov, editor, *Lecture Notes in Mathematics: Sensitivity of Functionals with Applications in Engineering Science*. Springer Verlag, Berlin, 1984.

[19] M. Drela. *Two-Dimensional Transonic Aerodynamic Design and Analysis Using the Euler Equations*. PhD thesis, Massachusetts Institute of Technology, December 1985.

[20] D. Drikakis and E. Schreck. "Development of parallel implicit Navier-Stokes solvers on MMD multi-processor systems". In *31st AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1993. AIAA 93-0062.

[21] P.R. Eiseman, Y.K. Choo, and R.E. Smith. "Algebraic grid generation with control points". In T.J. Chung, editor, *Finite Elements in Fluids, Volume 8*. Hemisphere Publ. Corp., 1992.

[22] C.A. Felippa. "Optimization of finite element grids by direct energy search". *Appl. Math. Modelling*, 1:93–96, 1976.

[23] C.A. Felippa. "Numerical experiments in finite element grid optimization by direct energy search". *Appl. Math. Modelling*, 1:239–244, 1977.

[24] B.A. Finlayson. "Existence of variational principles for the Navier-Stokes equation". *Physics of Fluids*, 15(6):963–967, June 1972.

[25] B.A. Finlayson. *The Method of Weighted Residuals and Variational Principles*. Academic Press, 1972.

[26] C.A.J. Fletcher. *Computational Techniques for Fluid Dynamics*, volume 1. Springer Verlag, 1988. Pages 293-299.

[27] D. Funaro. *Polynomial Approximation of Differential Equations.* Springer Verlag, 1992. Pages 1-2.

[28] A. Geist. *PVM 3 User's Guide and Reference Manual.* Oak Ridge National Laboratory, May 1994. TM 12187.

[29] D. Greenspan. "On approximating extremals of functionals - I The method and examples for boundary value problems". *Bull. Int. Comp. Centre*, 4(2):99–120, April-June 1965. University of Rome.

[30] D. Greenspan. "On approximating extremals of functionals - II Theory and generalizations related to boundary value problems for nonlinear differential equations". *Int. J. Engng Sci.*, 5:571–588, 1967.

[31] D. Greenspan and P.C. Jain. "Application of a method for approximating extremals of functionals to compressible flow". *Mathematical Analysis and Applications*, 18(1):85–111, April 1967. Also available as technical report of the Mathematical Research Center of the University of Wisconsin, TR 597, January 1966.

[32] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI.* MIT Press, 1994.

[33] G.L. Guymon. "A finite element solution of the one-dimensional diffusion-convection equation". *Water Resources Research*, 6(1):204–210, February 1970.

[34] G.L. Guymon, V.H. Scott, and L.R. Herrmann. "A general numerical solution of the two-dimensional diffusion-convection equation by the finite element method". *Water Resources Research*, 6(6):1611–1617, December 1970.

[35] C.D. Harris. “Two-dimensional aerodynamic characteristics of the NACA-0012 airfoil in the Langley 8-foot transonic pressure tunnel”. Technical Report TM 81927, NASA, April 1981.

[36] P.M. Hartwich. “Fresh look at floating shock fitting”. In *28th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1990. AIAA 90-0108.

[37] P.M. Hartwich. “Split coefficient matrix (SCM) method with floating shock fitting for transonic airfoils”. In *12th International Conference on Numerical Methods in Fluid Dynamics*, Oxford, United Kingdom, July 1990.

[38] O. Hassan, K. Morgan, and J. Peraire. “Finite-element solution scheme for compressible viscous flow”. In T.J. Chung, editor, *Finite Elements in Fluids, Volume 8*. Hemisphere Publ. Corp., 1992.

[39] T.L. Holst. “Viscous transonic airfoil workshop compendium of results”. *Journal of Aircraft*, 25:1073–1087, December 1988.

[40] K. Hornik, M. Stinchcombe, and H. White. “Multi-layer feedforward networks are universal approximators”. *Neural Networks*, 2:359–366, 1989.

[41] I. Iami. “On the flow of a compressible fluid past a circular cylinder”. In *Proc. Phys. - Math. Soc. Japan*, pages 180–193, 1941. Ser. 3, volume 23.

[42] P.S. Jensen. “Advances in computer methods for partial differential methods”. In R. Vivhnevestjy, editor, *AICA Symposium*, pages 80–85, Lehigh University, 1975.

[43] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, 1990. Pages 33-38.

[44] D.A. Johnson and L.S. King. "A new turbulence closure model for boundary layer flows with strong adverse pressure gradients and separation". In *22nd AIAA Aerospace Sciences Meeting and Exhibit*, Reno, Nevada, January 1984. AIAA 84-0175.

[45] L.K. Jones. "Constructive approximations for neural networks by sigmoidal functions". *Proceedings of the IEEE*, 78(10):1586–1589, 1990.

[46] L.K. Jones. "A simple lemma on greedy approximation in Hilbert space and convergence rates for projection pursuit regression and neural network training". *The Annals of Statistics*, 20(1):608–613, 1992.

[47] R. Kalaba. "Computational considerations for some deterministic and adaptive control processes". In G. Leitmann, editor, *Optimization Techniques with Applications to Aerospace Systems*. Academic Press, 1962.

[48] D. Kincaid and W. Cheney. *Numerical Analysis*. Brooks and Cole, 1990. Pages 181-182.

[49] M. Krizek and P. Neittaanmaeki. *Finite Element Approximation of Variational Problems and Applications*. Longman Scientific & Technical, 1990. Page 14.

[50] C. Lanczos. *The Variational Principles in Mechanics*. University of Toronto Press, 1952.

[51] J.C. LeBalleur. "Numerical flow calculation and viscous-inviscid interaction techniques". *Computation Methods in Viscous Flows*, 23:419–450, 1984.

[52] R.M. Lewis and V. Torczon. "Pattern search algorithms for bound constrained minimization". Technical Report 96-20, ICASE, NASA Langley Research

Center, Hampton, Virginia, 1996. Submitted to *SIAM Journal of Optimization.*

[53] H. Li and C.J. Chang. "A fast scheme to analyze 3d disk airflow on a parallel computer". In H. D. Simon, editor, *Parallel Computational Fluid Dynamics, Implementations and Results.* MIT Press, 1992.

[54] M.J. Lighthill. "On displacement thickness". *Journal of Fluid Mechanics*, 4:383–392, 1958.

[55] R.C. Lock and B.R. Williams. "Viscous-inviscid interactions in external aerodynamics". *Progress in Aerospace Sciences*, 24:51–171, 1987.

[56] P.E. Lush and T.M. Cherry. "The variational method in hydrodynamics". *J. Mech. Appl. Math.*, 9:6–21, 1956.

[57] A.R. Manwell. "A variational principle for steady homenergic compressible flow with finite shocks". *Wave motion*, 2(1):83–95, 1980.

[58] A.J. Meade, Jr. *Semi-Discrete Galerkin Modelling of Compressible Viscous Flow Past a Circular Cone at Incidence.* PhD thesis, University of California, Berkeley, 1989.

[59] A.J. Meade, Jr. "The semi-discrete Galerkin finite element modelling of compressible viscous flow past an airfoil". Technical Report NAG-1-1196, Rice University, 1992.

[60] A.J. Meade, Jr. and M. Kokkolaras. "Enhancement of a viscous-inviscid-interaction airfoil analysis code using the parallel direct search algorithm". *Mathematical and Computer Modelling*, 25(6):85–96, March 1997.

[61] A.J. Meade, Jr., M. Kokkolaras, and B.A. Zeldin. "Optimal incremental approximation for the adaptive solution of differential equations". *International Journal for Numerical Methods in Engineering*, 1997. Submitted.

[62] A.J. Meade, Jr., M. Kokkolaras, and B.A. Zeldin. "Sequential function approximation for the solution of differential equations". *Communications in Numerical Methods in Engineering*, 1997. To appear.

[63] S.G. Mikhlin. *Variational Methods in Mathematical Physics.* MacMillan - Pergamon, 1964.

[64] P.M. Morse and H. Feshbach. *Methods of Theoretical Physics*, volume 1. McGraw-Hill, 1953. Pages 276-280.

[65] U.R. Mueller and H. Henke. "Computation of subsonic viscous and transonic viscous-inviscid unsteady flow". *Computers and Fluids*, 22:649–661, 1993.

[66] J.T. Oden. "A general theory of finite elements - II Applications". *Int. J. Num. Meth. Engng.*, 1:247–259, 1969.

[67] J.T. Oden. *Variational Methods in Theoretical Mechanics.* Springer Verlag, 1976.

[68] J.T. Oden. "Adaptive methods in computational fluid dynamics". In T.J. Chung, editor, *Finite Elements in Fluids, Volume 8.* Hemisphere Publ. Corp., 1992.

[69] E.R. Oliveira. "Optimization of finite element solutions". In *Proc. Third Conf. Matrix Methods in Struct. Mech.*, Ohio, 1971. Air Force Flight Dynamics Laboratory, Wright-Patterson AFB. AFFDL TR 71-160.

[70] M.J. Orr. "Regularization in the selection of radial basis function centres". *Neural Computation*, 7(3):606–623, 1995.

[71] J. Platt. "A resource-allocating network for function interpolation". *Neural Computation*, 3:213–225, 1991.

[72] P.M. Prenter. *Splines and Variational Methods.* John Wiley and Sons, 1989. Pages 116-133.

[73] R.J. Prozan. "A variational principle for compressible fluid mechanics - Discussion of the multi-dimensional theory". Technical Report 3614, NASA, 1982.

[74] R.J. Prozan. "A variational principle for compressible fluid mechanics - Discussion of the one-dimensional theory". Technical Report 3526, NASA, April 1982.

[75] M.M. Rai and P. Moin. "Direct numerical simulation of transition and turbulence in a spatially evolving boundary layer". In *Proceedings of the 10th AIAA Computational Fluid Dynamics Conference*, pages 890–914, 1991. AIAA 91-1607.

[76] K. Rektorys. *Variational Methods in Mathematics, Science, and Engineering.* Dordrecht, 1980. 2nd edition.

[77] D. Rumelhart, G.E. Hinton, and R.J. Williams. "Learning internal representations by error propagation". In D. Rumelhart, J. L. McClelland, and the PDP Research Group, editors, *Parallel Distributed Processing: Exploration in the Microstructure of Cognition*, pages 318–362. MIT Press, Cambridge, Massachusetts, 1986.

[78] S. Saarinen, R. Bramley, and G. Cybenko. "Ill-conditioning in neural network training problems". *SIAM J. Sci. Comput.*, 14(3):693–714, 1993.

[79] H. Schlichting. *Boundary-Layer Theory.* McGraw Hill, 1979. Page 22.

[80] V. Torczon. "PDS: Direct search methods for unconstrained optimization on either sequential or parallel machines". Technical Report TR 92206, CRPC, March 1992.

[81] V. Torczon. "On the convergence of pattern search methods". *SIAM Journal of Optimization*, February 1997.

[82] I.H. Tuncer, J.A. Ekaterinaris, and M.F. Platzer. "A zonal viscous-inviscid interaction method for 2-D compressible flows". In *Computational Aero- and Hydro-Acoustics*, pages 19–28. ASME, 1993. FED 147.

[83] M.M. Vainberg. *Variational Methods for the Study of Nonlinear Operators.* Holden-Day Inc., 1964. Pages 54-58.

[84] V.N. Vatsa, M.J. Werle, and J.M. Verdon. "Viscid-inviscid-interaction at laminar and turbulent symmetric trailing edges". In *20th AIAA Aerospace Sciences Meeting*, Orlando, Florida, January 1982. AIAA 82-0165.

[85] O.C. Zienkiewicz and R.L. Taylor. *The Finite Element Method.* McGraw-Hill, 1989. Pages 240-242.

Appendix A

A.1 Norms, inner products, and orthogonality

In order to determine the "best approximation" of a (solution) function, it is necessary to define a metric which measures the distance between the exact and the approximate [7]. Any linear space of functions (or vectors) $\mathbf{Q}$, i.e. a set S of elements obeying the addition and multiplication operations according to the usual rules of arithmetic, is called normed if a real-valued function exists and has the following properties

(a)	$\| f \| > 0, \quad \text{if } f \neq 0$	(positivity)
(b)	$\| \eta f \| = \mid \eta \mid \| f \|$	(homogeneity)
(c)	$\| f + g \| \leq \| f \| + \| g \|$	(triangle inequality),

where f, $g \in \mathbf{Q}$ and $\eta \in \mathbf{R}$. The metric for two functions f and g of this space is then given by $\| f - g \|$. Since a large number of different norms can be defined, it is obvious that the concepts of best approximation and convergence depend on the norm choice. For example, the definition of the norm

$$\| f(x) \| = \max_{a \leq x \leq b} \mid f(x) \mid \tag{A.1}$$

for a function $f(x)$ defined on an interval $[a, b]$ enables the discussion of uniform convergence.

However, it is often useful to associate the norm with an operation defined as the inner product. The inner product between two functions f and g is denoted by $\langle f, g\rangle$ and has the following properties

(a) $\langle f, f\rangle > 0, \quad \text{if } f \neq 0$ (positivity)

(b) $\langle f, g\rangle = \langle g, f\rangle$ (symmetry)

(c) $\langle f, \eta g + \zeta h\rangle = \eta\langle f, g\rangle + \zeta\langle f, h\rangle$ (linearity),

where η, $\zeta \in \mathbf{R}$. Then, the norm of the inner-product space is automatically defined as

$$|| f || = +\sqrt{\langle f, f\rangle} \tag{A.2}$$

and the following properties hold

(a) $| \langle f, g\rangle | \leq || f || \, || g ||$ (Cauchy-Schwarz inequality)

(b) $|| f + g || \leq || f || + || g ||$ (triangle inequality)

(c) $|| f + g ||^2 + || f - g ||^2 = 2(|| f ||^2 + || g ||^2)$ (parallelogram law)

(d) $\langle f, g\rangle = 0 \Rightarrow || f + g ||^2 = || f ||^2 + || g ||^2$ (Pythagorean law).

If $f(\bar{x})$ is continuous on some domain $\bar{\Omega}$, where $\bar{\Omega} = \Omega \bigcup \partial\Omega$ with $\partial\Omega$ being the boundary of Ω, and square-integrable, i.e $\int_\Omega f(\bar{x})^2 d\bar{x} < \infty$, the Hilbert space $L_2(\Omega)$ is defined, and the associated inner product and L_2-norm are given by

$$\langle f, f\rangle = \int_\Omega (f(\bar{x}))^2 d\bar{x} = || f ||^2_{L_2\Omega} \quad . \tag{A.3}$$

This is a quite popular inner product definition, and is used throughout this dissertation.

Two functions f and g are said to be orthogonal if $\langle f, g\rangle = 0$. In terms of definition (A.3), two functions $f(\bar{x})$ and $g(\bar{x})$ are said to be orthogonal with respect to some function $w(\bar{x})$ on some domain $\bar{\Omega}$ if

$$\langle f, g\rangle_w = \int_\Omega f(\bar{x})g(\bar{x})w(\bar{x})d\bar{x} = 0 \ . \tag{A.4}$$

If an orthogonal set of functions has the additional property of $\| f_i \| = 1$, $\forall i$, it is said to be orthonormal. Therefore, the functions of the set share the cardinality property

$$\langle f_i, f_j\rangle = \delta_{ij} = \begin{cases} 1 \ , & \text{if} \ \ i = j \\ 0 \ , & \text{if} \ \ i \neq j \ , \end{cases} \tag{A.5}$$

where δ_{ij} is the Kronecker delta.

Finally note that a complete normed linear space is called a Banach space, while an inner product space that is complete in the associated norm is called a Hilbert space. A space S is said to be complete if every Cauchy sequence $\{x_n : n = 1, 2, \cdots\}$ with $x_i \in S$ converges to some $x \in S$. A sequence is Cauchy if $\lim_{n,m\to\infty} \| x_n - x_m \| = 0$. Hilbert spaces are widely used in computational mechanics, especially in finite element methods [43]. A Hilbert space $H^k(\Omega)$ consists of functions that are, together with their k-th derivatives, square-integrable on Ω. The associated inner product is then defined as

$$\langle f, f\rangle_{H^k(\Omega)} = \int_\Omega ((f(\bar{x}))^2 + (f'(\bar{x}))^2 + \cdots + (f^k(\bar{x}))^2)d\bar{x} = \| f \|^2_{H^k(\Omega)} \ , \tag{A.6}$$

where $f^k(\bar{x})$ is the k-th derivative of $f(\bar{x})$. A Hilbert space $H^k_0(\Omega)$ consists of functions that are, together with their k-th derivatives, square-integrable on Ω

and vanish on its boundary $\partial\Omega$. Hilbert spaces of the latter form are very useful for problems associated with homogeneous Dirichlet boundary conditions.

A.2 B-splines and Gaussian functions

The widely used B_1-splines were introduced in Section 3.4.2, equation (3.9). Figure A.1(a) illustrates a B_1-spline with center $x_M = 1.0$, width to the left $\Delta x_l = 0.5$, and width to the right $\Delta x_r = 0.5$.

Another popular B-splines family consists of B_3-splines, also called cubic splines. Cubic splines are defined in terms of five parameters, compared to the three parameters that are needed for the definition of the B_1-splines,

$$B_3(x) = \begin{cases} \frac{(x-x_M-\Delta x_{l1}-\Delta x_{l2})^3}{(\Delta x_{l2})^3}\,, \quad \text{if} \;\; x_M - \Delta x_{l1} - \Delta x_{l2} \leq x \leq x_M - \Delta x_{l1} \\ \frac{(\Delta x_{l1})^3+3(\Delta x_{l1})^2(x-x_M-\Delta x_{l1})+3\Delta x_{l1}(x-x_M-\Delta x_{l1})^2-3(x-x_M-\Delta x_{l1})^3}{(\Delta x_{l1})^3}\,, \\ \text{if} \;\; x_M - \Delta x_{l1} \leq x \leq x_M \\ \frac{(\Delta x_{r1})^3+3(\Delta x_{r1})^2(x_M+\Delta x_{r1}-x)+3\Delta x_{r1}(x_M+\Delta x_{r1}-x)^2-3(x_M+\Delta x_{r1}-x)^3}{(\Delta x_{r1})^3}, \\ \text{if} \;\; x_M \leq x \leq x_M + \Delta x_{r1} \\ \frac{(x_M+\Delta x_{r1}+\Delta x_{r2}-x)^3}{(\Delta x_{r2})^3}\,, \quad \text{if} \;\; x_M + \Delta x_{r1} \leq x \leq x_M + \Delta x_{r1}\Delta x_{r2} \\ 0\,, \quad \text{otherwise.} \end{cases} \tag{A.7}$$

A B_3-spline with center $x_M = 1.0$, first width to the left $\Delta x_{l1} = 0.25$, second width to the left $\Delta x_{l2} = 0.25$, first width to the right $\Delta x_{r1} = 0.25$, and second width to the right $\Delta x_{r2} = 0.25$ is plotted in Figure A.1(b). The maximum value of a B_3-spline is 4.0, while the maximum value of the B_1-spline was 1.0. The latter occurs for the B_3-spline at the locations of the two new parameters, namely at $x_M - \Delta x_{l1}$ and $x_M + \Delta x_{r1}$ The advantages offered by the use of B_3-splines include the improved adjustment for nonlinear approximations and the availability of up to third order derivatives.

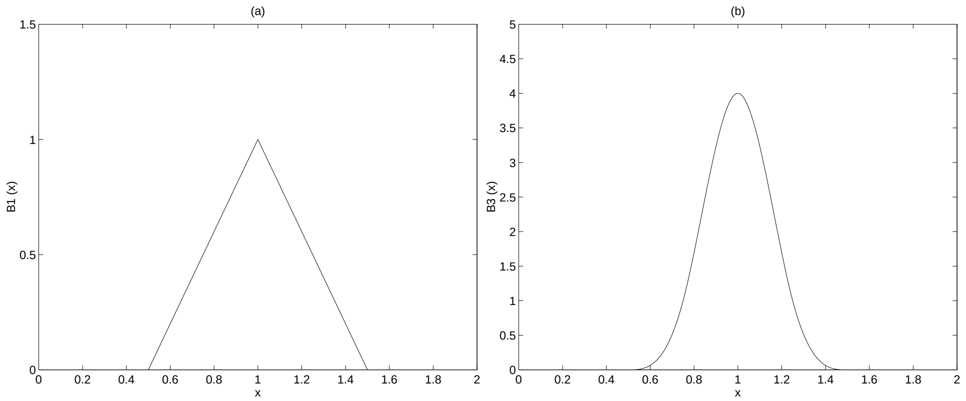

Figure A.1 B-splines: (a) a B_1-spline and (b) a B_3-spline.

Finally, the most popular of the radial basis functions is the well known Gaussian function, defined in terms of two parameters, its center c and radius r [70]

$$G(x) = e^{-\frac{1}{r^2}(x-c)^2} \ . \tag{A.8}$$

The Gaussian function is symmetric and can be employed for nonlinear approximations by exploiting its nonlinear nature and its simplicity due to the low number of parameters that are to be optimized. However, its symmetry decreases its flexibility. A Gaussian function with center $c = 1.5$ and radius $r = 0.5$ is depicted in Figure A.2.

For all numerical examples considered in this dissertation, B_1-splines were chosen to be employed as the basis functions. This choice was made for the reasons that follow. Firstly, B_1-splines are quite easy to use and implement computationally. In addition, they depend on three and six parameters for the one- and two-dimensional cases, respectively. In this manner, the size of the optimization problem is kept as low as possible. Secondly, the derivatives of B_3-splines can

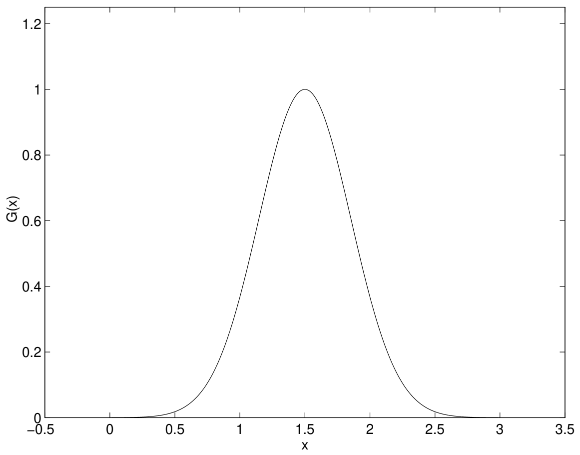

Figure A.2 A Gaussian function with center $c = 1.5$ and radius $r = 0.5$.

achieve relatively large values, or "spikes", which are the source of numerical instabilities. Moreover, they depend on five and ten parameters for the one- and two-dimensional cases, respectively. Accordingly, not only is the size of the associated optimization problems increased, but the computational representation of the optimization variables becomes more difficult, as will be shown in Section A.5. Finally, Gaussian functions depend on only two and three parameters to be optimized for the one- and two-dimensional cases, respectively. However, the domain of influence (or support) of the functions cannot be as readily controlled as for the B_1-splines. Consequently, the necessary constraints for enforcing homogeneous Dirichlet boundary conditions may not be as straightforwardly to implement as for the B_1-splines, as will be illustrated in Section A.5. In this regard, Gaussian functions are more useful in problems associated with Neumann boundary conditions, where the basis functions do not have to vanish at certain boundaries.

A.3 Members of the method of weighted residuals

The members of the method of weighted residuals can be classified according to the choice of the weighting function as follows.

1. The *subdomain method*, where the domain of interest Ω is divided into n smaller subdomains ω_i, $i = 1, \ldots, n$, and the weighting functions $w_i(\bar{x})$ are given by

$$w_i(\bar{x}) = \begin{cases} 1 \ , & \bar{x} \in \omega_i \\ 0 \ , & \bar{x} \notin \omega_i \ . \end{cases} \tag{A.9}$$

2. The *collocation method*, were the weighting function is given by the Dirac delta function, $w_i = \delta(\bar{x} - \bar{x}_i)$, and possess the following property

$$\int_\Omega \delta(\bar{x} - \bar{x}_i) d\bar{x} = \begin{cases} 1 \ , & \bar{x} = \bar{x}_i \\ 0 \ , & \bar{x} \neq \bar{x}_i \ . \end{cases} \tag{A.10}$$

   In this manner,

$$\langle R(\bar{x}) w_i(\bar{x}) \rangle = \int_\Omega R(\bar{x}) w_i(\bar{x}) d\bar{x} = R \mid_{\bar{x}_i} \ . \tag{A.11}$$

3. The *least-squares method*, where the weighting function w_i is given by

$$w_i = \frac{\partial R}{\partial c_i} \ , \tag{A.12}$$

   where c_i is the i-th coefficient of the series expansion.

4. The *method of moments*, where the weighting functions are given by the monomials

$$w_i = x^i, \quad i = 0, \ldots, n-1 \; . \tag{A.13}$$

5. The *Bubnov-Galerkin method*, where the weighting functions are given by the basis (or trial) functions used in the series expansion of the approximate solution, i.e. $w_i = \Phi_i$. A modification of the form $w_i = g_i \Phi_i$ forms the *Petrov-Galerkin method.*

A.4 Classification of differential operators

Differential operators are classified into linear and nonlinear operators. The adjoint of an operator H is found by integrating $\int_\Omega vH[u]d\bar{x}$ by parts to obtain $\int_\Omega uH^*[v]d\bar{x} + (\text{B.T.})$, where the term (B.T.) denotes the boundary terms stemming from the performed integration(s) by parts. Linear operators are classified into self- and nonself-adjoint operators. A linear operator is said to be self-adjoint if $L[\cdot] = L^*[\cdot]$. The adjoint of a nonlinear operator N^* cannot be the same as the operator N. The term adjoint is therefore inappropriate for nonlinear operators; it is substituted by the term self-sufficient. If the operator is nonlinear, the Frechret differential must be examined. The Frechret differential $N'_w[z]$ of a nonlinear operator N is defined as [25](page 301)

$$N'_w[z] \equiv \lim_{\epsilon \to 0} \frac{N[w + \epsilon z] - N[w]}{\epsilon} = \frac{\partial}{\partial \epsilon}(N[w + \epsilon z]) \mid_{\epsilon=0} \; , \tag{A.14}$$

where N'_w is the Frechret derivative with respect to the function w of the operator

N operating on z. If the Frechret derivative is symmetric, i.e.

$$\int_\Omega v N_w'[z]\bar{x} = \int_\Omega z N_w'[v]\bar{x} \ , \tag{A.15}$$

then the operator N is said to be self-sufficient. Equation (A.15) is the condition necessary for the existence of a functional that gives the operator N as its gradient [83]. It is the analogy of the condition that is required for a vector field to be able to be derived from a potential, extented to function spaces [25](page 300). Note that the case of linear operators is in fact a special case of nonlinear operators since the Frechret derivative is given by the operator itself

$$L_w'[z] \equiv \lim_{\epsilon \to 0} \frac{L[w + \epsilon z] - L[w]}{\epsilon} = \lim_{\epsilon \to 0} \frac{L[w] + \epsilon L[z] - L[w]}{\epsilon} = L[z] \ . \tag{A.16}$$

Variational principles are readily available for problems governed by self-adjoint differential operators.

A differential operator is said to be positive definite if

$$\int_\Omega u L[u] d\bar{x} \ > 0 \ \ , \ u \neq 0 \ . \tag{A.17}$$

Elliptic operators are self-adjoint and positive definite in contrast to hyperbolic or parabolic operators. Steady-state problems typically yield elliptic operators, while time-dependent problems are typically associated with hyperbolic or parabolic operators.

A.5 PDS computational issues

A.5.1 The formulation of the optimization variables

The computational representation of the parameters to be optimized is neither unique nor trivial. For the B_1-splines, for example, one option to define the PDS optimization variables in terms of the parameters to be optimized is given by Equation (3.10). For this representation

$$x_M = \mathtt{x(1)}\ , \quad \Delta x_l = \mathtt{x(2)}\ , \quad \text{and} \quad \Delta x_r = \mathtt{x(3)}\ , \tag{A.18}$$

where x(1), x(2), and x(3) are the PDS variables. An alternative formulation could be the following

$$x_M = \mathtt{x(1)}\ , \quad x_L = \mathtt{x(2)}\ , \quad \text{and} \quad x_R = \mathtt{x(3)}\ , \tag{A.19}$$

where $x_L = x_M - \Delta x_l$ and $x_R = x_M + \Delta x_r$.

The first formulation requires the additional satisfaction of constraints since the functions to be chosen have to satisfy homogeneous boundary conditions, i.e. vanish at the boundaries. Specifically, the constraints

$$\mathtt{x(1)} - \mathtt{x(2)} \geq x_{\Omega,L} \quad \text{and} \quad \mathtt{x(1)} + \mathtt{x(3)} \leq x_{\Omega,R} \tag{A.20}$$

need to be included in the PDS optimization problem formulation, where $x_{\Omega,L}$ and $x_{\Omega,R}$ denote the left and right boundaries of a one-dimensional domain, respectively.

This is not necessary for the second representation, where the homogeneous boundary conditions can be readily satisfied by appropriate bounds, which PDS requires in all cases for the optimization variables. However, the first representation was proven, by means of computational experimentations, to be more efficient than the second.

It is obvious that with increasing number of parameters, as is the case for B_3-splines, the number of choices increases exponentially, and a numerical study should be performed to determine the optimal representation.

Lastly, note that an adequately large lower bound on Δx is necessary to avoid relatively large values in the evaluation of the derivatives of the basis functions. A typical value for the lower bound is 1 % of the width of the computational domain.

A.5.2 Scaling

An issue of significance in direct search optimization methods is the correct scaling among the variables. The objective of scaling is to avoid uneven searches in the directions of the optimization variables that constitute the dimensionality of the search hyperplane.

PDS offers the option of enforcing diverse scaling factors for the optimization variables in a very simple and user-friendly manner. The scaling factor of each variable is given by the range of values that this variable can take. For example, if the variable `x(1)` is likely to take any value between -3 and 6, its scaling factor is 9. Scaling is unnecessary, obviously, if all optimization variables have approximately the same scaling factor, in which case PDS assigns a scaling factor of magnitude 1 to each variable. A scaling factor of magnitude 1 is to be selected in the absence of information.

A.6 Diagonal dominance and spectral radius

A matrix $\mathbf{A} \in \mathbf{R}^{n \times n}$ is said to be diagonal dominant if $\forall i$

$$| a_{ii} | > \sum_{j=1, j \neq i}^{n} | a_{ij} |, \quad j = 1, \ldots, n \; . \tag{A.21}$$

The spectral radius of the matrix $\mathbf{A}$ is given by

$$\rho(\mathbf{A}) = \max | \lambda_i |, \quad i = 1, \ldots, n \; , \tag{A.22}$$

where λ_i are the eigenvalues of the matrix $\mathbf{A}$, i.e. the roots of the characteristic polynomial $\det(\mathbf{A} - \lambda \mathbf{I}) = 0$, with $\mathbf{I}$ being the $n \times n$ identity matrix.

A.7 Aerodynamic coefficients and similarity parameters

The lift and drag coefficients of an airfoil for a given flow configuration (angle of attack, Mach number, and Reynolds number) are given by

$$C_L = \frac{L'}{q_\infty c} \tag{A.23}$$

and

$$C_D = \frac{D'}{q_\infty c} \; , \tag{A.24}$$

respectively. In these equations, L' and D' are the lift and drag forces per unit width, respectively, c is the airfoil chord, and q_∞ is the dynamic pressure given by $q_\infty = \rho_\infty V_\infty^2/2$, with ρ_∞ and V_∞ being the freestream density and velocity, respectively.

Typically, L' and D' are calculated by integrating the pressure (C_p) and skin friction (C_f) coefficients, respectively. The pressure and skin friction coefficients are given by the following relations

$$C_p = \frac{p - p_\infty}{q_\infty} \tag{A.25}$$

$$C_f = \frac{\tau}{q_\infty} \ , \tag{A.26}$$

where p and τ are the pressure and wall shear stress, respectively, and p_∞ is the freestream pressure [1].

The Reynolds and Mach numbers are defined as

$$Re = \frac{uL}{\nu} \tag{A.27}$$

and

$$Ma = \frac{u}{a} \ , \tag{A.28}$$

respectively, where u is the velocity of the fluid in motion, a is the local sound velocity, L is some characteristic length (for example, the diameter of the pipe in internal flows), and $\nu = \mu/\rho$ is the kinematic viscosity of the fluid in motion, where μ is the dynamic viscosity and ρ is the density of the fluid in motion.

A.8 Numerical integration

All numerical integrations were performed by means of ESSL library routines. ESSL is a numerical library provided by the IBM SP2 parallel computer, on which all codes presented in this dissertation were executed. In particular, the double precision routines `DGLNQ` and `DGLNQ2`, which utilize the Gauss-Legendre quadrature rule for one- and two-dimensional numerical integration, respectively, were used.

As mentioned in the main text of the dissertation, the optimally chosen basis functions are allowed to overlap; a structured grid is not built to provide comfortably determined integration bounds. In addition, B_1-splines are not uniformly defined, but can take three different values according to their parameters and position in space. In this regard, it is necessary to develop algorithms for the efficient and accurate numerical integration of the variational integrals.

In particular, two algorithms were developed. The first algorithm can be employed when the structure of the functional allows the decomposition of the variational integral into simpler integrals, where the summations from the series expansions can be taken outside the integrals. For example, when the following algebraic manipulation can be performed

$$
\begin{aligned}
&\int\int u_k^a(x,y)\frac{\partial(u_k^a(x,y))}{\partial x}dxdy = \\
&\int\int \sum_{i=1}^{k} c_i\Phi_i(x)\Phi_i(y)\sum_{j=1}^{k} c_j\frac{d\Phi_j(x)}{dx}\Phi_j(y)dxdy = \\
&\sum_{i=1}^{k}\sum_{j=1}^{k} c_ic_j\int\int \Phi_i(x)\Phi_i(y)\frac{d\Phi_j(x)}{dx}\Phi_j(y)dxdy = \\
&\sum_{i=1}^{k}\sum_{j=1}^{k} c_ic_j\int \Phi_i(x)\frac{d\Phi_j(x)}{dx}dx\int \Phi_i(y)\Phi_j(y)dy \ ,
\end{aligned}
\tag{A.29}
$$

the integral of the product of two (or more) basis functions and/or its derivatives in one dimension has to be computed. The algorithm used to determine the number m of the necessary subintegrations and the associated bounds a_j and b_j, $j = 1, \ldots, m$, for n basis functions (B_1-splines), assuming that the bounds of the domain are x_1 and x_2, is the following

Algorithm 1

Find the maximum x_L among the $x_{L,i}$, $i = 1, \ldots, n$;

Find the minimum x_R among the $x_{R,i}$, $i = 1, \ldots, n$;

If $x_{L,max} \leq x_1$, set $x_{L,max} = x_1$;

If $x_{R,min} \geq x_2$, set $x_{R,min} = x_2$;

Set $m = 0$;

If $x_{L,max} \geq x_{R,min}$, stop;

For $i = 1, \ldots, n$

  If $x_{L,max} < x_{M,i} < x_{R,min}$

    $m \leftarrow m + 1$;

    $a_m = x_{L,max}$; $b_m = x_{M,i}$;

    Set $x_{L,max} = x_{M,i}$;

  End if;

End for;

$m \leftarrow m + 1$;

$a_m = x_{L,max}$; $b_m = x_{R,min}$.

In the case that the variational integral cannot be simplified and/or decomposed as in Equation (A.29), the approximate functions have to be numerically evaluated on an adequate number of Gauss points so that the integral can be calculated accurately. Both the unstructured and relatively complicated overlapping of the basis functions and the change of their definition over certain spatial domains prevent the numerical integration of being efficiently and accurately performed, even for a quite high number of Gauss points. A domain decomposition, according to the basis function parameters and their location, is necessary to yield dependable numerical results. An algorithm was designed for this purpose and utilized for the two-dimensional integration in the compressible, inviscid, irrotational flow application. Assuming that the optimal sequential approximation has reached step n and that the variational integral I^n has to be evaluated within a rectangular domain of interest with boundaries x_1, x_2, y_1, and y_2, the integral is decomposed into $(nsubx-1)\times(nsuby-1)$ subintegrals $I^n_{i,j}$, $i=1,\dots,nsubx-1$, $j=1,\dots,nsuby-1$ and evaluated with help of the following algorithm

Algorithm 2

Merge the n-dimensional arrays x_L, x_M, and x_R

  into a 3×n-dimensional array called $xcopy$;

Merge the n-dimensional arrays y_L, y_M, and y_R

  into a 3×n-dimensional array called $ycopy$;

Sort the 3×n-dimensional arrays $xcopy$ and $ycopy$;

Set $nxsub = 0$;

If $xcopy(1) \le x_1$

  $nxsub \leftarrow nxsub + 1$;

  $xsub(nxsub) = x_1$;

End if;

Algorithm 2 (cont.)

For $i = 1, \ldots, n$

  If $(x_1 < xcopy(i) < x_2)$ and $(xcopy(i) \neq xsub(nxsub))$

    $nxsub \leftarrow nxsub + 1;$

    $xsub(nxsub) = x_1;$

  End if;

End for;

If $xcopy(n) \geq x_2$

  $nxsub \leftarrow nxsub + 1;$

  $xsub(nxsub) = x_2;$

End if;

Set $nysub = 0;$

If $ycopy(1) \leq y_1$

  $nysub \leftarrow nysub + 1;$

  $ysub(nysub) = y_1;$

End if;

For $i = 1, \ldots, n$

  If $(y_1 < ycopy(i) < y_2)$ and $(ycopy(i) \neq ysub(nysub))$

    $nysub \leftarrow nysub + 1;$

    $ysub(nysub) = y_1;$

  End if;

End for;

Algorithm 2 (cont.)

If $ycopy(n) \geq y_2$

$nysub \leftarrow nysub + 1;$

$ysub(nysub) = y_2;$

End if;

Set $I^n = 0;$

For $i = 1, \ldots, nsubx - 1$

For $j = 1, \ldots, nsuby - 1$

Compute $I^n_{i,j};$

$I^n \leftarrow I^n + I^n_{i,j};$

End for;

End for.