Finite Element Programming of the Navier-Stokes Equations

C. TAYLOR
Department of Civil Engineering, University College of Swansea, U.K.

T. G. HUGHES
W. S. Atkins and Ptrs., Swansea, U.K.

PINERIDGE PRESS LIMITED
Swansea, U.K.

First Published in 1981 by
Pineridge Press Ltd.,
91, West Cross Lane, West Cross, Swansea, U.K.

ISBN 0–906674–16–6

Taylor C.
Finite element programming in fluids
1. Fluid dynamics – Mathematics
2. Finite element method
I. Title II. Hughes, T. G.
532'.051015353 QA911
ISBN 0–906674–16–6

Printed in Great Britain by
Robert MacLehose and Co. Ltd.
Printers to the University of Glasgow

Finite Element Programming of the Navier-Stokes Equations

Preface

This introductory text is directed at undergraduate and postgraduate students, as well as practising engineers, who wish to utilise the finite element method to solve irrotational and viscous flow problems. The basic concepts are presented in a manner which does not require any previous knowledge of the technique being outlined. The simplistic approach is accompanied by a modular type software development in which each section or subroutine can be examined in isolation. By following such an approach and adopting an engineering style of presentation, the mathematical concepts and formal proofs have been kept to a minimum, although some have been included, as required, to maintain continuity of the evolution of the technique.

The adopted modular approach is particularly useful for those who wish to devise specialised software directed at solving a narrow range of problems. Since each subroutine can be considered individually quite sophisticated modifications can be incorporated with a minimum of reference to the remainder of the program. This feature can, obviously, be used to advantage for contraction and expansion of the overall program.

Steady state incompressible two dimensional irrotational and viscous flow can be analysed using the programs included in the text. Indeed, an introduction to the utilisation of the technique to solve turbulent flow problems is presented in Chapter 9. Since the solution procedure is iterative the software can be readily expanded to include time dependency and sophisticated models of turbulence. Descriptive examples are included, where appropriate, the majority of which can be solved by hand calculations. The sequential processes associated with each subroutine are described in detail so that modifications can be conducted with a minimum of effort. In order to facilitate any development a complete set of software is available, on request, via the publishers.

The material contained in the current text is based on the experience gained when teaching and conducting research into a wide range of engineering problems. The authors wish to thank their colleagues who, during discussions, contributed to the clarification of many of the ideas presented.

1981

C. TAYLOR
T. G. HUGHES

Contents

Chapter 1

An Introduction to the Numerical Approach

1.1 Introduction

A set of mathematical equations governing the flow of a viscous fluid was developed early in the 19th century and presented as the well known Navier-Stokes equations[1]. These equations, in their most general form, together with the equation of conversation of mass are purported to interconnect the pertinent dependent variables which describe the flow of a viscous fluid. Since that time the basic form of the equations has remained unchanged even when turbulence phenomena are introduced[2]. Indeed, most present day developments in research where turbulence is significant are based on the Navier-Stokes equations [3]. The work of subsequent researches has, therefore, been directed at evolving equation solution techniques and subsequently extending such techniques to include additional non-linearities. These include such non-linearities as turbulence, heat transfer, shock waves etc.

The development of methods of solution has followed, as in all branches of science, well trodden paths. Early analytical approaches were later, circa 1900, supplemented and sometimes superseded by numerical methods. The significant contribution made by Thom[4] is widely considered to be the first definitive work on numerical methods in the present field of interest. This seved as a bench mark for a considerable number of subsequent numerical analysts. During the late 1940's, electronic computers emerged as a powerful tool which could be employed to advantage by engineers and scientists. This era was accompanied by a renewed vigorous interest in the development of numerical techniques to solve hitherto intractable problems. As always, the availability of a new research tool resulted in rapid advances in the technological field. In fluid mechanics these have been so significant that a re-appraisal of the basic concepts associated with the evaluation of the hydrodynamical equations has also been possible. At the present time, since a suitable solution algorithm has been devised, an in-depth scrutiny of the basic parameters embodied in an equation becomes trivial and is simply a matter of repetition. This, of course, is of prime importance if research is to proceed on a rational basis.

Prior to the late 1960's the most widely used computer based numerical methods were the *Method of Characteristics*[5,6] and the *Finite Difference Approach*[3]. Both are usually utilised to create an approximate model of the governing equations which is stored, in matrix form, on a computer. Each

achieved notable success and, in particular circumstances, can be considered to be the **best** approach. Non-linearities can, in most circumstances, be accommodated with little difficulty and have been used to solve a wide range of engineering problems. The refinement of such techniques has, over the years, resulted in highly sophisticated modelling methods, particularly when finite difference methods are employed[7,8]. Such techniques fall outside the scope of the present text since a wide range of excellent literature is already available to the interested reader.

The immediate success of the application of the Finite Element Method (*F.E.M.*) in solid mechanics[9] provided, in the late 1960's, the initial impetus for the utilisation of the method in Fluid Mechanics. It was thought that the significant advantages gained in structural mechanics would again be open to exploitation. This is not always the case but the advantages which were apparent merited the development of the method. Indeed, it has been demonstrated that for the analysis of particular problems there are distinct advantages leading to a significant progress in specific research areas. It is becoming increasingly clear that the *F.E.M.* has its rightful place in the development of numerical techniques for solving complex flow problems.

The range of problems encountered in fluid mechanics is so great that the authors have adopted a deliberate restrictive policy for the present text. Only one technique is developed for solving the steady state incompressible form of the two dimensional Navier-Stokes equations. This allows an **in depth** development of the necessary computer programming and solution techniques. The primitive variables of velocity and pressure are used as the dependent variables. This then forms a basis for further development by the reader. Indeed, the final chapter illustrates the possible enhancement of the program to solve turbulent flow problems.

1.2 Basic concepts

The basic concepts associated with the application of the *F.E.M.* to solve problems in fluid mechanics are summarised in the idealised block diagram, Fig. 1.1. Once the governing equation, or set of equations, has been defined then the basic procedure is conceptually straightforward. The new concepts are embodied in blocks (3)–(5) where the procedure is highly dependent on the way in which both the geometry of the flow domain and spatial variation of the variables is defined. Readers versed in the finite difference techniques will be familiar with the manner in which this is accomplished. For completeness, particular techniques for matrix formation and solution will also be presented. A detailed account of the required procedures for the *F.E.M.* will be outlined in subsequent chapters. However, a brief resumé of the pertinent steps will be included in order to introduce some basic definitions and outline the procedure peculiar to the present method of approach.

Initially, the governing equations are usually subjected to some minimisation procedure. Following the expressed current restrictive policy only one

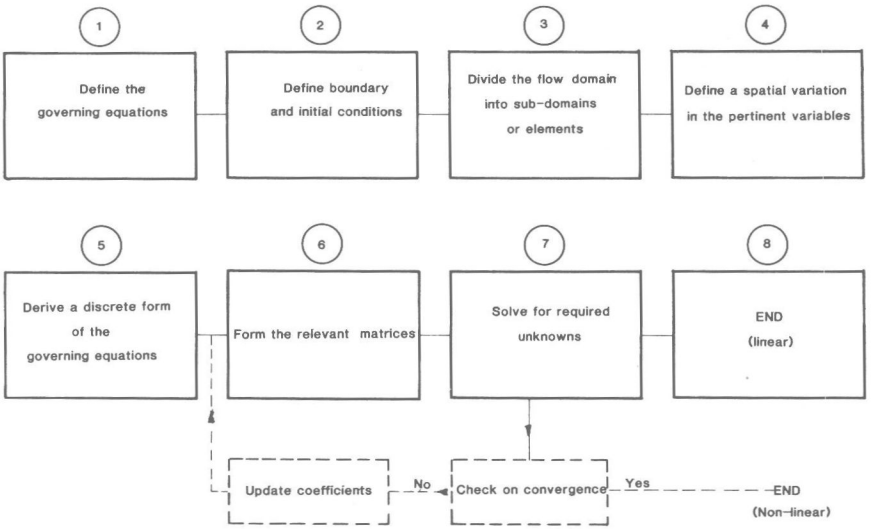

Fig. 1.1 Block diagram showing essential steps in the application of the F.E.M.

such procedure is used — the *Galerkin Weighted Residuals Approach*[10]. Other techniques, e.g. functional, collocation, least squares, have been adequately presented in other standard texts[11,12,13]. In the Galerkin approach the basic equation is *weighted* with appropriate discrete weighting functions and the resulting equation integrated over the region of interest and equated to zero. For instance, if the flow is governed by the Laplace equation,

$$\nabla^2\varphi = 0 \tag{1.1}$$

where φ is a velocity potential, the application of the Galerkin approach results in,

$$\int_{\Omega} W_k(\nabla^2\varphi)\, d\Omega = 0 \qquad k = 1,2,3\ldots m \tag{1.2}$$

where W_k denotes a weighting function associated with a discrete point, k, within the domain, Ω, and m is the number of such discrete points. The weighting function is usually a function of space only.

If (1.2) is to be solved then a spatial variation of φ must also be defined. The first step is to subdivide the flow domain into a number of sub-domains called *elements*, Fig. 1.2(a). Each element is associated with a number of discrete points or *nodes* located within or on the element boundary. The spatial variation in φ within the element is then defined in terms of the values at the node points. For instance, if the element has eight node points with associated

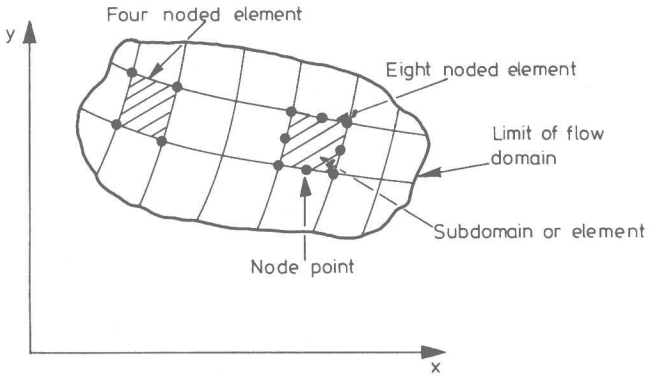

Fig. 1.2(a) Domain sub division into elements

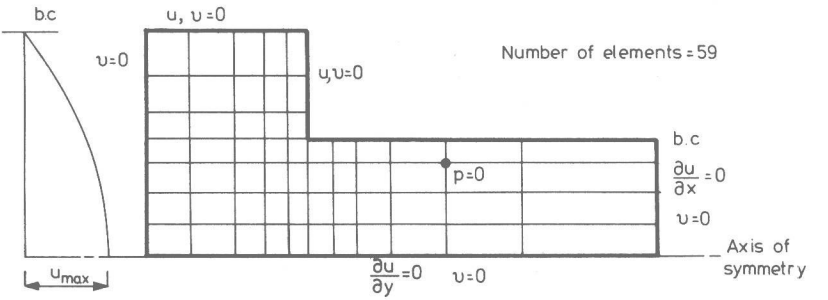

Fig. 1.2(b) Typical distribution and boundary conditions for laminar flow through a symmetric expansion

values $\varphi_1,\varphi_2,\varphi_3,\ldots\varphi_8$ then the value of φ within the element is written,

$$\varphi = N_i\varphi_i \qquad i=1,2,3\ldots8 \tag{1.3}$$

in which N is a function of coordinates of the node points. These polynomials are usually referred to as *shape functions.* A full derivation of the shape functions employed is presented in Chapter 3.

Having completed a spatial subdivision of the domain of interest and allocated suitable shape functions the next stage is to integrate equation (1.2), over each element and summate the contributions from each element. This is usually accomplished by some form of numerical integration leading to an element contribution in matrix form,

$$[\mathbf{h}]\cdot\begin{Bmatrix}\varphi_1\\ \cdot\\ \cdot\\ \cdot\\ \varphi_8\end{Bmatrix} \tag{1.4}$$

All such element contributions are summated leading to a final matrix equation

$$[\mathbf{H}]\cdot\begin{Bmatrix}\varphi_1\\ \cdot\\ \cdot\\ \cdot\\ \varphi_n\end{Bmatrix}=0 \tag{1.5}$$

where n is the total number of nodal points in the domain. The resulting matrix equation is then solved, after the introduction of appropriate boundary conditions, for the nodal values of the variable. The processes outlined above, leading to an explicit solution, must be modified if more complicated equations are to be solved. Consider a simple form of the steady state heat conduction equation in two dimensions,

$$\frac{\partial}{\partial x}\left(K_x\frac{\partial\varphi}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_y\frac{\partial\varphi}{\partial y}\right)=0 \tag{1.6}$$

where the material properties are dependent on both space and local temperature,

$$\begin{aligned}K_x&=f_{\mathrm{I}}(x,y,\varphi)\\ K_y&=f_{\mathrm{II}}(x,y,\varphi)\end{aligned} \tag{1.7}$$

The processes for setting up the final matrix form of (1.6) follow those already outlined for the Laplace equation. However, the solution technique now becomes iterative since the values of conductivity, K, are not initially known. Equation (1.6) could then be written,

$$\frac{\partial}{\partial x}\left(K_x^{k-1}\frac{\partial\varphi^k}{\partial x}\right)+\frac{\partial}{\partial y}\left(K_y^{k-1}\frac{\partial\varphi^k}{\partial y}\right)=0 \tag{1.8}$$

where k denotes the current iteration value. Another loop must therefore be introduced into Fig. 1.1 where the values of the conductivity are evaluated, if the solution has not converged to within specified tolerances on, say, $(\varphi^k-\varphi^{k-1})$, then the matrices are reformed and an updated solution is sought for the temperature distribution φ. The equation used in this example is, obviously, a simplified form of the equations depicting laminar flow. Nevertheless, the basic concepts and necessary procedures for solving a single non-linear equation are the same.

1.3 General program structure

The general structure of the program is shown on Fig. 1.3. The general routines pertaining to the setting up of the necessary matrices are, as far as possible, given a mnemonic identifier. All relevant routines are called from a

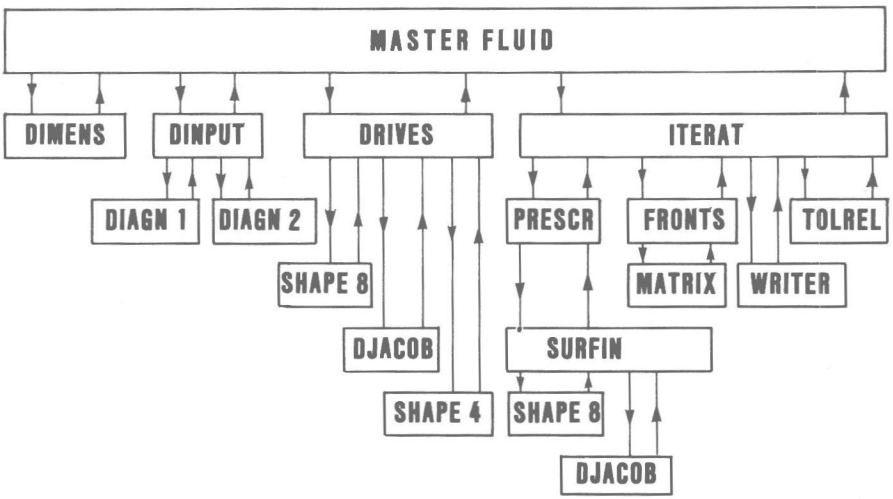

Fig. 1.3 General program structure

master segment, *FLUID*, which calls, in sequence, all the required remaining subroutines. A brief description will be included to familiarise the reader with the overall procedure and subroutine names.

1.3.1 Subroutines

i) DIMENS

A number of vectors and arrays are utilised during the 'house-keeping' processes involved in the formulation of the matrix to be solved. The dimensions required for each of these vectors or arrays is set in subroutine DIMENS. This permits a form of dynamic dimensioning to be used. The program can, therefore, be increased or decreased in size to suite a particular problem.

ii) DINPUT

The data required is read in subroutine DINPUT. This includes nodal point coordinate location, element numbers and associated nodal points, physical properties of the fluid, boundary and initial conditions. As the data is input two further subroutines DIAGN1 and DIAGN2 are called to check that the data which has been input complies with certain simple checks.

(a) **Subroutine DIAGN1** This subroutine checks the overall control parameter which governs the number of nodes, number of elements, boundary conditions and initial conditions for the particular problem under investigation.

(b) **Subroutine DIAGN2** Various checks are incorporated to ensure that the geometric data obeys some simple criteria.

If any check fails the data which had not been read to that point will be read and printed.

iii) Subroutine DRIVES

A subroutine in which the constants necessary for numerical integration of the relevant equations are defined. Subroutines SHAPE8, SHAPE4 and DJACOB are called from DRIVES.

(a) **Subroutine SHAPE8** The subroutine SHAPE8 calculates the shape functions, N, required for defining the variation of the independent variable over an element which contains eight nodes on the element boundary, Fig. 1.2(a).

(b) **Subroutine SHAPE4** Subroutine SHAPE4 calculates the shape functions for a four noded element, Fig. 1.2(a). Two shape function subroutines are used in the program evolved in subsequent chapters. This is necessary since the pressure and velocity in the Navier-Stokes equations are associated with four and eight nodes of an element, respectively. The reason for this is discussed in Chapter 4 and reference[14,15,16].

(c) **Subroutine DJACOB** In DJACOB the first order differentials of the shape function, with respect to the chosen coordinate system, are calculated. These are required when such terms as, $\frac{\partial \varphi}{\partial x}, \frac{\partial \varphi}{\partial y}$ are necessary. For instance, having defined

$$\varphi = [N_1, N_2, N_3 \ldots N_8] \cdot \begin{Bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \cdot \\ \cdot \\ \cdot \\ \varphi_8 \end{Bmatrix} \tag{1.9}$$

then first order differentials can be written in the form,

$$\frac{\partial \varphi}{\partial x} = \left[\frac{\partial N_1}{\partial x}, \frac{\partial N_2}{\partial x}, \frac{\partial N_3}{\partial x} \ldots \frac{\partial N_8}{\partial x} \right] \cdot \begin{Bmatrix} \varphi_1 \\ \varphi_2 \\ \varphi_3 \\ \vdots \\ \varphi_8 \end{Bmatrix} \tag{1.10}$$

since N is a function of spatial coordinates x,y.

iv) Subroutine ITERAT

This is the main subroutine which calls the necessary subroutines for the evaluation and solution of the necessary matrix equations, transformation of