Think Before You Compute

Every fluid dynamicist will at some point need to use computation. Thinking about the physics, constraints and the requirements early on will be rewarded with benefits in time, effort, accuracy and expense. How these benefits can be realised is illustrated in this guide for would-be researchers and beginning graduate students to some of the standard methods and common pitfalls of computational fluid mechanics. Based on a lecture course that the author has developed over 20 years, the text is split into three parts. The quick introduction enables students to solve numerically a basic nonlinear problem by a simple method in just three hours. The follow-up part expands on all the key essentials, including discretisation (finite differences, finite elements and spectral methods), time-stepping and linear algebra. The final part is a selection of optional advanced topics, including hyperbolic equations, the representation of surfaces, the boundary integral method, the multigrid method, domain decomposition, the fast multipole method, particle methods and wavelets.

E. J. Hinch has been a teacher and researcher in fluid mechanics and applied mathematics at the University of Cambridge for over 45 years. He is the author of Perturbation Methods (Cambridge University Press, 1991) and has been awarded the Fluid Dynamics prizes of the European Mechanics Society and the American Physical Society Division of Fluid Dynamics.
Think Before You Compute
A Prelude to Computational Fluid Dynamics

E. J. HINCH
University of Cambridge
## Contents

**Preface**  
page xi

### PART I A FIRST PROBLEM  

1 The driven cavity  

1.1 The problem  

1.2 Know your physics  

1.3 Know your PDEs  

1.4 Special physics of the corner  

1.5 Nondimensionalisation  

1.6 Steady vs transient calculations  

1.7 Pressure!  

2 Streamfunction-vorticity formulation  

2.1 Formulation  

2.2 Finite differences (simple)  

2.3 Poisson problem  

2.4 Test the code  

2.5 Code quality  

2.6 Simple graphs  

2.7 Vorticity evolution  

2.8 Time-step instability  

2.9 Accuracy consistency  

2.10 Results  

3 Primitive variable formulation  

3.1 Formulation  

3.2 Pressure equation  

3.3 Algorithm 1 with pressure equation  

3.4 Incompressibility as a constraint: split time-step  

© in this web service Cambridge University Press  

www.cambridge.org
3.5 Algorithm 2 by projection – spurious pressure modes 40
3.6 Algorithm 3 with a staggered grid 42
3.7 Results from algorithm 3 46

PART II GENERALITIES 51

4 Finite differences 53
4.1 Higher orders 53
  4.1.1 Central differencing 53
  4.1.2 One-sided differencing 54
  4.1.3 Nonequispaced points 55
4.2 Compact fourth-order Poisson solver 56
  4.2.1 One-dimensional version 56
  4.2.2 Two dimensions 56
4.3 Upwinding 58
4.4 Other grids 59
4.5 Conservative schemes 60

5 Finite elements 64
5.1 The two ideas 64
5.2 Representations in one dimension 65
  5.2.1 Constant elements 65
  5.2.2 Linear elements 65
  5.2.3 Quadratic elements 65
  5.2.4 Cubic elements 66
  5.2.5 Basis functions 67
5.3 Representations in two dimensions 67
  5.3.1 Constant elements 68
  5.3.2 Linear elements 68
  5.3.3 Quadratic elements 69
  5.3.4 Cubic elements 70
  5.3.5 Basis functions 71
  5.3.6 Rectangles 71
5.4 Variational statement of the Poisson problem 72
5.5 Details in one dimension 73
5.6 Details in two dimensions 74
5.7 Galerkin formulation 78
5.8 Diffusion equation 80
  5.8.1 Weak formulation 80
  5.8.2 In one dimension 80
  5.8.3 In two dimensions 81
## Contents

5.9 Navier–Stokes equation 81
  5.9.1 Weak formulation 81
  5.9.2 Time integration 82
  5.9.3 Pressure problem – locking 82
  5.9.4 Pressure problem – spurious modes 84

6 Spectral methods 86
  6.1 The two ideas 86
  6.2 Global vs local 87
  6.3 Choice of spectral basis functions 89
  6.4 Chebyshev polynomials 90
  6.5 Rates of convergence 90
  6.6 Gibbs phenomenon 91
  6.7 Discrete Fourier Transform 93
  6.8 Aliasing 94
  6.9 Fast Fourier Transform (FFT) 96
  6.10 Differential matrix 97
  6.11 Navier–Stokes 98
  6.12 Bridging the gap 99

7 Time integration 100
  7.1 Stability 100
  7.2 Forward Euler 102
  7.3 Backward Euler 103
  7.4 Midpoint Euler 104
  7.5 Crank–Nicolson 104
  7.6 Leapfrog 105
  7.7 Runge–Kutta 105
  7.8 Multistep methods 106
  7.9 Symplectic integrators 107
  7.10 Navier–Stokes 108

8 Linear algebra 110
  8.1 LAPACK 111
  8.2 Gaussian elimination 112
    8.2.1 Pivoting 113
    8.2.2 LU decomposition 114
    8.2.3 Errors 115
  8.3 QR decomposition 115
    8.3.1 QR by Gram–Schmidt 116
    8.3.2 QR by Givens rotations 117
    8.3.3 QR by Householder reflections 118
## Contents

8.4 Sparse matrices 119  
8.5 Conjugate gradients 119  
8.6 Eigenproblems 121  
8.7 Power iteration 122  
8.8 Jacobi 122  
8.9 Main method 122

### PART III SPECIAL TOPICS 125

9 Software packages and FreeFem++ 127  
9.1 Poisson problem 128  
9.2 Driven cavity 131

10 Hyperbolic equations 136  
10.1 Simplest, but unstable 137  
10.2 Lax–Friedricks, too stable 139  
10.3 Upwinding 142  
10.4 Crank–Nicolson 142  
10.5 Lax–Wendroff 144  
10.6 Angled Derivative 145  
10.7 Propagation of discontinuities 148  
10.8 Flux limiters 149  
10.9 Nonlinear advection 151  
10.10 Godunov method 153

11 Representation of surfaces 156  
11.1 Curves in two dimensions 157  
11.1.1 Splines 157  
11.2 Surfaces in three dimensions 158  
11.2.1 Redistributing points 158  
11.2.2 Curvature 159  
11.3 Volume of Fluid (VoF) method 162  
11.4 Diffuse interface method 163  
11.5 Level sets 165  
11.5.1 Fast Marching Method 165

12 Boundary integral method 167  
12.1 Integral equation for Laplace equation 167  
12.1.1 Greens functions 168  
12.1.2 Eigensolutions 169  
12.1.3 Singular integrals 169  
12.2 Discretisation 169
### Contents

12.2.1 Evaluation of the matrix elements 170
12.2.2 Avoiding the eigensolution 171
12.2.3 Tests 171
12.2.4 Costs 172
12.3 Free-surface potential flows 173
12.4 Stokes flows 173
13 Fast Poisson solvers 175
  13.1 Multigrid method 175
  13.1.1 A V-cycle 176
  13.1.2 Accuracy and costs 177
  13.2 Fast Fourier Transforms 178
  13.3 Domain decomposition 180
  13.3.1 Costs 182
14 Fast Multipole Method 183
  14.1 Trees, roots and leaves 184
  14.2 Barnes–Hut algorithm 184
  14.3 Fast Multipole algorithm 186
    14.3.1 Upward pass 186
    14.3.2 Downward pass 187
    14.3.3 Errors 187
    14.3.4 Costs 188
15 Nonlinear considerations 190
  15.1 Finding steady states 190
    15.1.1 Finding the Jacobian 191
    15.1.2 Example of the limit cycle of the Van der Pol oscillator 192
  15.2 Parameter continuation 193
  15.3 Searching for singularities of physical problems 193
    15.3.1 Use of computer algebra 195
16 Particle methods 197
  16.1 Molecular dynamics 197
  16.2 Lattice Gas 199
  16.3 Lattice Boltzmann 201
  16.4 Dissipative particle dynamics 204
  16.5 Stokesian dynamics 205
    16.5.1 Hydrodynamic interactions 205
    16.5.2 Brownian motion 207
  16.6 Force Coupling Method 209
16.7 Granular media simulations 211
16.8 Smooth Particle Hydrodynamics 212

## 17 Wavelets
17.1 Continuous Wavelet Transform 218
17.2 Discrete Wavelet Transform 220
17.3 Fast Wavelet Transform 221
17.4 Daubechies wavelets 224

Index 226
Preface

This book, based on a graduate course in Cambridge, is aimed at students starting research into fluid mechanics who are thinking about computing a flow, as one amongst other tools of investigation. It is an educational book for beginners, using the simplest methods appropriate, rather than an advanced text for those already familiar with the methods. It is certainly not a research monograph about the very latest techniques. It is for those using a little computing for research in fluid mechanics. It is not for those researching into computational methods, either proving their mathematical properties or creating new methods.

The book is designed for students who have taken an undergraduate course on fluid mechanics and an undergraduate course on computing simple numerical methods, designed to lead those students to some understanding of computing flows. The course on fluid mechanics should have discussed the incompressible Navier–Stokes equation, the Reynolds number, boundary layers, vorticity and streamfunctions. The course on numerical methods should have included simple finite differencing of differential equations and iterative solutions. This book will then develop numerical methods appropriate to fluid mechanics. On the other hand, the book will not develop fluid mechanics. This means that no examples are included of numerical calculations in acoustics, aeronautics, compressible flows, combustion and reactions, biology, atmospheres, oceans, geology, non-Newtonian fluids and many industries. It is important to point out the many models of turbulence are also not included.

The book is divided into three parts. Part I is short and composed of three chapters. It tackles a very simple problem in fluid mechanics by very simple numerical methods. By making everything simple, students should be able to obtain results for a nonlinear flow after just one week of lectures. Some MATLAB code is available on my website,\(^1\) so that students do not even have

\(^1\) www.damtp.cam.ac.uk/user/hinch/teaching/CMIFM_Handouts/****.m, where **** is PoissonTest, StrfnVort and PrimVarb.
Preface

to spend time coding the programs themselves. But more than quick results, the first part delivers a far-from-hidden message of the need to think about what one is doing. There are issues of understanding the formulation of the question, of designing and monitoring the accuracy, of noting where time is consumed, of handling instabilities, of producing evidence that the answer is correct. There is also the special numerical issue in fluid mechanics of how to find the pressure. The simple problem tackled in Part I is the driven cavity, a square domain with a prescribed tangential velocity on the top surface. The simple numerical methods used are finite differences, central in space and forward in time, and successive over-relaxation of Gauss–Seidel iteration to solve a Poisson problem.

Part II gives a more detailed treatment of the general issues, such as turning a continuous partial differential equation into a finite discrete problem, i.e. discretisation by finite differences, finite elements and spectral methods, and general issues of time-stepping and solving large sparse systems of linear equations. Topics covered under discretisation include conservative formulations, a compact fourth-order Poisson solver, problems with pressure in finite elements, local vs global representations and the need for a pseudospectral approach. While time-stepping is only a discretisation in time, it deserves a more careful examination. There are issues of controlling the accuracy, not being too stable and sometimes avoiding excessive expensive evaluations of derivatives. While fluid mechanics is strictly nonlinear, large linear problems occur in the Poisson problem to find pressure or in considering the linear stability of a flow; hence the chapter on linear algebra. Students are strongly recommended not to code up finite elements or solvers for linear algebra but rather to use safe professionally written packages. The two chapters on these topics are included to explain what the packages are doing, so that the correct package can be used wisely.

Part III is an incomplete collection of specialised topics. The first chapter of Part III gives a quick introduction to one particular finite element package, FreeFem++. I selected this package having tried several, because I have found that in less than an hour students can learn to use it to compute a flow. Solving hyperbolic equations numerically is unwise, and Chapter 10 illustrates the problems that arise with seemingly good schemes; only the one-dimensional case is presented. Some fluid mechanics problems involve moving boundaries. Chapter 11 discusses various representations of surfaces. This is followed by a chapter on the boundary integral method, which for potential flow and Stokes flow only uses data on the surface and so is highly suited for computing moving boundary problems. A Poisson problem typically consumes much of the time in computing a flow, so fast methods have been
developed to solve it. For simple geometry, the multigrid method is probably the fastest, while in complex geometries domain-decomposition is particularly good with parallel computing. When the forcing of the Poisson problem is by many point sources, a fast multipole method can be useful, but only when there is a very large number of sources. While fluid mechanics is essentially about a continuum medium, particle descriptions naturally occur, whether one studies molecules moving in a gas, colloidal particles in a suspension, dry grains in a flowing granular medium or parcels of fluid in a Lagrangian description. Chapter 16 describes all these. The final chapter gives a quick introduction to wavelets, which have been found useful for analysing flows and identifying isolated regions of great activity.

A cautionary remark. One of the difficulties in computing flows is that every branch of fluid mechanics has its special physics, and that special physics should be reflected in some special numerics. Note the implication that there is no universal method or package applicable to every fluids problem. In fact I would go further to say that even for a particular problem there is no best method, one should always be able to dream up something better.

And finally I must acknowledge the enormous assistance of my colleagues Stephen Cowley, Paul Dellar and Paul Metcalfe in developing the graduate lecture course in Cambridge over a period of years.