# FLUID DYNAMICS

## *Theory, Computation, and Numerical Simulation*

Accompanied by the software library *FDLIB*

*by*

**C. Pozrikidis**
*University of California, San Diego*
*La Jolla, California 92093-0411*
*U.S.A.*

*Email: cpozrikidis@ucsd.edu*
*Internet URL: http://stokes.ucsd.edu/c_pozrikidis*

**KLUWER ACADEMIC PUBLISHERS**
**Boston / Dordrecht / London**

**Distributors for North, Central and South America:**
Kluwer Academic Publishers
101 Philip Drive
Assinippi Park
Norwell, Massachusetts 02061 USA
Telephone (781) 871-6600
Fax (781) 681-9045
E-Mail <kluwer@wkap.com>

**Distributors for all other countries:**
Kluwer Academic Publishers Group
Distribution Centre
Post Office Box 322
3300 AH Dordrecht, THE NETHERLANDS
Telephone 31 78 6392 392
Fax 31 78 6546 474
E-Mail <services@wkap.nl>

Electronic Services <http://www.wkap.nl>

**Library of Congress Cataloging-in-Publication Data**

Pozrikidis, C.
Fluid dynamics: theory, computation, and numerical simulation / by C. Pozrikidis
p.cm.
"Accompanied by the software library FDLIB. "
Includes bibliographical references and index.
ISBN 0-7923-7351-0 (acid-free paper)
1. Fluid dynamics. I. Title

QA911 .P63 2001
532'.05—dc21

2001029458



*Printed on acid-free paper.*

Printed in the United States of America.

# Preface

Ready access to computers at an institutional and personal level has defined a new era in teaching and learning. The opportunity to extend the subject matter of traditional science and engineering disciplines into the realm of scientific computing has become not only desirable, but also necessary. Thanks to portability and low overhead and operating costs, experimentation by numerical simulation has become a viable substitute, and occasionally the only alternative, to physical experimentation.

The new environment has motivated the writing of texts and monographs with a modern perspective that incorporates numerical and computer programming aspects as an integral part of the curriculum: methods, concepts, and ideas should be presented in a unified fashion that motivates and underlines the urgency of the new elements, but does not compromise the rigor of the classical approach and does not oversimplify.

Interfacing fundamental concepts and practical methods of scientific computing can be done on different levels. In one approach, theory and implementation are kept complementary and presented in a sequential fashion. In a second approach, the coupling involves deriving computational methods and simulation algorithms, and translating equations into computer code instructions immediately following problem formulations. The author of this book is a proponent of the second approach and advocates its adoption as a means of enhancing learning: interjecting methods of scientific computing into the traditional discourse offers a powerful venue for developing analytical skills and obtaining physical insight.

The goal of this book is to offer an introductory course in fluid mechanics, covering traditional topics in a way that unifies theory, computation, computer programming, and numerical simulation. The approach is truly introductory, in the sense that a minimum of prerequisites are required. The intended audience includes not only advanced undergraduate and entry-level graduate students, but also a broad class of scientists and engineers with a general interest in scientific computing.

The discourse is distinguished by two features. First, solution procedures and algorithms are developed immediately after problem formulations. Second, numerical methods are introduced on a need-to-know basis and in increasing order of difficulty: function interpolation, function differentiation, function integration, solution of algebraic equations, finite-difference methods, etc.

A supplement to this book is the FORTRAN software library *FDLIB* whose programs explicitly illustrate how computational algorithms translate into computer code instructions. The codes of *FDLIB* range from introductory to advanced, and the problems considered span a broad range of applications; from laminar channel flows, to vortex flows, to flows in aerodynamics. The input is either entered from the keyboard or read from data files. The output is recorded in output files in numerical form so that it can be read and displayed using independent graphics, visualization, and animation applications on any computer platform. Computer problems at the end of each section ask the student to run the programs for various flow conditions, and thus study the effect of the various parameters characterizing a flow. Instructions for downloading the source code and a description of the library contents are given on page 651.

In concert with the intended usage of this book as a stand-alone text and as a tutorial on numerical fluid dynamics and scientific computing, references are not provided in the text. Instead, a selected compilation of introductory, advanced, and specialized references on fluid dynamics, calculus, numerical methods, and computational fluid dynamics are listed in the bibliography on page 666. The reader who wishes to focus on a particular topic is directed to these resources for further details.

I would like to extend special thanks to Vasilis Bontozoglou for his friendship and encouragement, and to Yuan Chih-Chung, Rhodalynn Degracia, Audrey Hill, and Kurt Keller for helping me with the preparation of the manuscript.

C. Pozrikidis

San Diego

January, 2001

Email: cpozrikidis@ucsd.edu
Book internet site: http://stokes.ucsd.edu/c_pozrikidis/FD_TCNS

# Contents

# *FDLIB* Software Library

The software library *FDLIB* contains a collection of FORTRAN 77 programs and subroutines that solve a broad range of problems in fluid dynamics using a variety of numerical methods. At the time of this printing, *FDLIB* consist of thirteen main directories, each containing a multitude of nested subdirectories. The contents of the subdirectories are listed on pages 655-667.

*Downloading*

The source codes of *FDLIB* and accompanying *User Guide*, available in the pdf format, can be downloaded from the internet site:
**http://stokes.ucsd.edu/c_pozrikidis/FDLIB**

*Installation and compilation on UNIX or LINUX*

- The library has been archived using the tar UNIX facility into the file *FDLIB.tar*. To unravel the directories on a UNIX or LINUX system, please execute the UNIX command:
  `tar xvf FDLIB.tar`

- The downloaded package does not contain object files or executables. An application can be built using the makefile provided in each subdirectory. A makefile is a UNIX script that instructs the operating system how to compile the main program and subroutines, and then link the object files into an executable using an f77 compiler.

- To compile the programs using a FORTRAN 90 compiler, simply make appropriate compiler call substitutions in the makefiles.

- To compile the application named *pindos*, go to the subdirectory where it resides, and type:
  `make pindos`

- To remove the object files and output files of the application named *pindos*, go to the subdirectory where it resides, and type:
  `make purge`

- To remove the object files, output files, and executable of the application named *pindos*, go to the subdirectory where it resides, and type:
  `make clean`

*Installation and compilation on Windows and Macintosh*

- To unravel the directories on a Windows or Macintosh platform, double-click on the archived tar file and follow the on-screen instructions of the invoked application.

- To compile the link the programs, follow the instructions of your FORTRAN 77 or FORTRAN 90 compiler.

*CFDLAB*

A subset of *FDLIB* has been combined with the X11 graphics library *vogle* into an integrated application that visualizes the results of the simulations. The source code of *CFDLAB* can be downloaded from the internet site:
**http://stokes.ucsd.edu/c_pozrikidis/CFDLAB**

# *FDLIB* Directories

| | Subject | Directory | Units |
|---|---|---|---|
| 1 | Numerical methods | *01_num_meth* | 89 |
| 2 | Grids | *02_grids* | 22 |
| 3 | Hydrostatics | *03_hydrostat* | 5 |
| 4 | Various | *04_various* | 27 |
| 5 | Lubrication | *05_lub* | 4 |
| 6 | Stokes flow | *06_stokes* | 33 |
| 7 | Potential flow | *07_ptf* | 25 |
| 8 | Hydrodynamic stability | *08_stab* | 15 |
| 9 | Vortex motion | *09_vortex* | 16 |
| 10 | Boundary layers | *10_bl* | 3 |
| 11 | Finite difference methods | *11_fdm* | 2 |
| 12 | Boundary element methods | *12_bem* | 6 |
| 13 | Turbulence | *13_turbo* | 1 |

## Directory Contents

The thirteen main directories consist of subdirectories that include main programs, assisting subroutines, and utility subroutines. Linked with drivers, the utility subroutines become stand-alone modules; all drivers are provided. A list of the subdirectories with a brief statement of their contents follows. The indicated number of units is the sum of the number of main programs and utility subroutines; assisting subroutines and drivers of utility subroutines are not counted. An extensive description of the problem statement, mathematical formulation, and numerical methods can be found in the *FDLIB User Guide* available from the *FDLIB* internet site.

## *01_num_meth*

### *General purpose numerical methods in scientific computing.* [1]

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| 01_num_comp | General aspects of numerical computation. | 14 |
| 02_lin_calc | Linear algebra and linear calculus. | 11 |
| 03_lin_eq | Systems of linear algebraic equations. | 10 |
| 04_nl_eq | Nonlinear algebraic equations. | 8 |
| 05_eigen | Eigenvalues and eigenvectors of matrices. | 9 |
| 06_interp_diff | Function interpolation and differentiation. | 8 |
| 07_integration | Function integration. | 10 |
| 08_approximation | Function approximation. | 8 |
| 09_ode_ivp | Ordinary differential equations; initial-value problems. | 1 |
| 10_ode_bvp | Ordinary differential equations; boundary value problems. | 2 |
| 11_pde | Partial differential equations. | 5 |
| 12_spec_fnc | Computation of special functions. | 10 |

[1] This directory accompanies the book: C. Pozrikidis 1998 *Numerical Computation in Science and Engineering*, Oxford University Press.

*02_grids*

***Adaptive discretization, parametrization, representation, and meshing of planar lines, three-dimensional lines, and three-dimensional surfaces.***

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| grid_2d | Discretization of a planar line into a graded mesh of straight or circular elements. | 3 |
| prd_2d | Adaptive parametrization of planar lines. | 5 |
| prd_3d | Adaptive parametrization of three-dimensional lines. | 5 |
| prd_ax | Adaptive parametrization of planar lines representing the trace of axisymmetric surfaces in a meridional plane. | 3 |
| rec_2d | Interpolation through a rectangular grid. | 1 |
| rec_2d_strml | Streamline pattern by interpolation through a rectangular grid. | 1 |
| sm_3d_cl_df | Smoothing of a function on a closed surface by surface diffusion. | 1 |
| sm_3d_cl_tr | Smoothing of a function on a closed surface by Legendre spectrum truncation. | 1 |
| trgl_octa | Triangulation of a closed surface. | 1 |
| trgl_octa_hs | Triangulation of an open surface. | 1 |
| trgl_sqr | Triangulation of a square patch. | 1 |

*03_hydrostat*

***Shapes of interfaces in hydrostatics.***

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| drop_2d | Shape of a two-dimensional pendant or sessile drop on a plane. | 1 |
| drop_ax | Shape of an axisymmetric pendant or sessile drop on a plane. | 1 |
| men_2d | Shape of a two-dimensional meniscus between two parallel plates. | 1 |
| men_2d_plate | Shape of a two-dimensional meniscus attached to an inclined plate. | 1 |
| men_ax | Shape of an axisymmetric meniscus in a circular tube. | 1 |

## *04_various*

### *Structure and kinematics of various flows.*

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| flow_1d | Steady unidirectional flow in a tube with arbitrary cross section. | 1 |
| flow_1d_osc | Oscillatory unidirectional flow in a tube with arbitrary cross section. | 1 |
| flow_1d_shear | Unidirectional shear flow over an array of cylinders. | 1 |
| spf | Similarity solutions for stagnation-point flows. | 1 |
| strml | Streamline patterns of a broad range of flows. | 1 |
| strmll | Light version of strml. | 1 |
| uni_flow | Steady unidirectional flows with rectilinear or circular streamlines. | 15 |
| uni_flow_u | Unsteady unidirectional flows with rectilinear or circular streamlines. | 8 |

*05_lub*

## *Nearly unidirectional lubrication flows at low Reynolds numbers.*

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| bear_2d | Dynamical simulation of the motion of a slider bearing pressing against a wall. | 1 |
| chan_2l_exp | Dynamical simulation of the evolution of two superposed viscous layers in a horizontal or inclined channel, computed by an explicit finite-diffrence method. | 1 |
| chan_2l_imp | Dynamical simulation of the evolution of two superposed viscous layers in a horizontal or inclined channel, computed by an implicit finite-difference method. | 1 |
| films | Evolution of an arbitrary number of superposed films on a horizontal or plane wall. | 1 |

---

## *06_stokes*

### *Viscous flows at vanishing Reynolds numbers.*

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| caps_2d | Dynamical simulation of the motion of a two-dimensional drop or elastic capsule, for a variety of flow configurations. | 1 |
| caps_3d | Dynamical simulation of the motion of a three-dimensional elastic capsule. | 1 |
| caps_ax | Dynamical simulation of the motion of an axisymmetric drop or elastic capsule, for a variety of flow configurations. | 1 |
| cop_ax | Shear flow over an axisymmetric cavity, orifice, or protrusion. | 1 |
| drop_3d | Dynamical simulation of the motion of a three-dimensional drop with constant or varying surface tension. | 1 |
| drop_3dw | Dynamical simulation of the deformation of a three-dimensional drop adhering to a plane wall. | 1 |
| em_2d | Dynamical simulation of the motion of a suspension of two-dimensional drops or elastic capsules, for a variety of flow configurations. | 1 |
| films | Dynamical simulation of the motion of superimposed layers in a channel, or two films flowing down a plane wall. | 1 |
| flow_2d | Two-dimensional flow in a domain with arbitrary geometry. | 1 |
| layers | Dynamical simulation of the motion of an arbitrary number of layers in a channel, or films flowing down a plane wall. | 1 |
| prtcl_2d | Flow past a fixed bed of two-dimensional particles with arbitrary shapes, for a variety of flow configurations, computed by a boundary-element method. | 1 |

*06_stokes (Continued)*

***Viscous flows at vanishing Reynolds numbers.***

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| prtcl_2d_se | Flow past a fixed bed of two-dimensional particles with arbitrary shapes for a variety of flow configurations, computed by a spectral-element method. | 1 |
| prtcl_3d | Flow past, or due to the motion of, a three-dimensional particle, for a variety of configurations, computed by a boundary-element method. | 1 |
| prtcl_ax | Flow past, or due to the motion of, a collection of axisymmetric particles, computed by a boundary-element method. | 1 |
| prtcl_sw | Swirling flow produced by the rotation of an axisymmetric particle, computed by a boundary-element method. | 1 |
| sgf_2d | Green's functions of two-dimensional Stokes flow. | 6 |
| sgf_3d | Green's functions of three-dimensional Stokes flow. | 5 |
| sgf_3dax | Green's functions of Stokes flow in an axisymmetric domain. | 1 |
| sgf_ax | Green's functions of axisymmetric Stokes flow. | 4 |
| susp_2d | Dynamical simulation of the motion of a suspension of two-dimensional rigid particles with arbitrary shapes, for a variety of flow configurations, computed by a boundary-element method. | 1 |
| susp_2d_se | Dynamical simulation of the motion of a suspension of two-dimensional rigid particles with arbitrary shapes, for a variety of flow configurations, computed by a spectral-element method. | 1 |
| thread_ax | Dynamical simulation of the evolution of a fluid thread or annular layer. | 1 |

---

*07_ptf*

## *Potential flows.*

| *Subdirectory* | *Topic* | *Units* |
|---|---|---|
| airf_2d | Shapes of airfoils. | 1 |
| airf_2d_cdp | Flow past an airfoil computed by the constant-dipole-panel method. | 1 |
| airf_2d_csdp | Flow past an airfoil computed by the constant-source-dipole-panel method. | 1 |
| airf_2d_lvp | Flow past an airfoil computed by the linear-vortex-panel method. | |
| body_2d | Flow past, or due to the motion of, a two-dimensional body, computed by a boundary element method. | 1 |
| body_ax | Flow past, or due to the motion of, an axisymmetric body, computed by a boundary element method. | |
| bubble_3d | Dynamical simulation of the deformation, collapse, or oscillations of a three-dimensional bubble. | 1 |
| cvt_2d | Flow in a rectangular cavity, computed by a finite difference method. | 1 |
| drop_3d | Dynamical simulation of the surface-tension induced oscillations of a three-dimensional inviscid drop suspended in vacuum. | 1 |
| flow_2d | Two-dimensional flow in an arbitrary domain, computed by a boundary element method. | 1 |
| lgf_2d | Green and Neumann functions of Laplace's equation in two dimensions. | 8 |
| lgf_3d | Green and Neumann functions of Laplace's equation in three dimensions. | 5 |
| lgf_ax | Green and Neumann functions of Laplace's equation in axisymmetric domains. | 3 |
| tank_2d | Dynamical simulation of liquid sloshing in a rectangular tank, computed by a boundary integral method. | 1 |