Appendix C

Companion Software

C.1 Overview

The software described in this appendix solves for: (a) free shear flow far-field behavior; (b) pipe and channel flow; and, (c) detailed sublayer and defect-layer behavior. In all cases, accurate algorithms are used that guarantee grid-independent solutions on any computer from an IBM PC to a Cray Y/MP. These programs serve two purposes. First, they solve basic **building-block** flow problems and can thus be helpful in developing or modifying a turbulence model. Second, these programs provide a definitive separation of turbulence-model error and numerical error.

As computers have increased in power, there has been a tendency away from analytical methods such as similarity solutions and singular perturbation methods. The mathematics of these procedures can be tedious, and it can be a lot easier to use a parabolic marching program for the types of flows described above. Eventually, marching far enough in space, self-similarity is achieved with such a program. However, complete avoidance of analytical methods can lead to an ignorance of important flow details such as singularities and important asymptotic behavior that can be masked by numerical error. More alarmingly, improper treatment of such flow detail can be the source of numerical error.

While the view presented here may appear to be a bit overcautious, it is justified by the difficulties so often encountered in solving turbulence transport equations. Exact solutions are virtually nonexistent. Experimental measurements cannot objectively be used to test for numerical accuracy. Consequently, it is difficult to assess the accuracy of a new turbulent-flow program. The programs described in this appendix generate very accurate

solutions for a variety of simple turbulent flows and turbulence models, and can be used to assess numerical accuracy of more complicated programs.

The accompanying diskette includes source code for all of the programs, as well as several auxiliary routines referenced by the programs. Plotting programs are included that can be used on IBM PC and compatible computers with support for a variety of video displays and hardcopy devices. Only the executable versions of the plotting programs are included as they are based on proprietary plotting software. Section C.6 lists the files included on the distribution diskette. Appendix E presents details of the hardware supported by the plotting programs.

C.1.1 Program Structure

All of the programs use time-marching methods to solve the nonlinear two-point boundary-value problems attending use of the similarity-solution method for simple turbulent flows. The solution algorithm used is based on implicit Crank-Nicolson differencing. To render straightforward and easy to modify programs, each equation of a given turbulence model is solved independently using a standard tridiagonal matrix inversion algorithm.

In the interest of portability, the programs have been written so that they run on IBM PC and compatible microcomputers, SUN Workstations, VAX (VMS based) computers, the Silicon Graphics IRIS and Cray super computers. The programs all use an **include** file named **cpuid** that defines a single parameter called *icpu*. This parameter is passed to a subroutine named **NAMSYS** that returns system-dependent and compiler-specific parameters. The file **cpuid** contains the following statements.

```
c-----------------------------------------------------------------------
c    S E L E C T   A P P R O P R I A T E   C O M P I L E R / C P U
c-----------------------------------------------------------------------
c     icpu =  0...SVS Fortran (680x0/80x86)
c          =  1...Lahey Fortran/Microsoft Fortran (80x86)
c          =  2...VAX/VMS
c          =  3...SUN Fortran (68020/SPARC)
c          =  4...Cray/Unicos
c          =  5...Silicon Graphics Iris
c-----------------------------------------------------------------------
      icpu=1
```

Set *icpu* to the value appropriate for your system. If your computer and/or Fortran compiler is not listed, you will have to modify subroutine **NAMSYS**. See Subsection C.5.2 for a detailed explanation of what is required as well as a listing of subroutine **NAMSYS**.

The same basic structure has been used for all of the programs. A standardized set of subroutine, input and output file, variable and common

block names has been used throughout. For example, if the program name is **PROGNAME**, the input data file is **progname.dat** and the main output disk file is **progname.prt**.

The main program coordinates all computations and program logic by calling a collection of subroutines. In all programs, the computational sequence is as follows:

1. Call **NAMIN** to coordinate reading the input data file. Note that **NAMIN** calls **NAMSYS** to set all system-dependent and compiler-specific parameters.

2. Call **GRID** to set up the finite-difference grid.

3. Call **START** to set closure coefficients and initial conditions.

4. Enter the main computation loop and repeat the following steps until convergence is achieved.

   - Call **GETETA** to compute $\eta = \int \mu_T \, d\xi$ for programs that use the Rubel-Melnik (1984) transformation.
   - Call **CALCS** to compute eddy viscosity, vertical velocity, etc.
   - Call **TMESTP** to compute the timestep.
   - Call **EDDY** to advance the solution in time.

5. When the solution has either converged or the maximum allowable number of timesteps has been reached, call **EDIT** to either write program output to a disk file or directly to the printer. For the free shear layer programs, call **GROW** to compute spreading rate.

6. Write a disk file that can be used for making a plot of program output.

C.1.2 Program Input

A standardized method, reminiscent of the non-Ansi-Standard NAMELIST scheme, has been used to provide input to the programs. As noted above, if the program name is **PROGNAME**, input is provided in a disk file named **progname.dat** (sample input for each program is provided on the distribution diskette). The format for integer quantities is (1x,a12,i4) while the format for floating-point quantities is (1x,a12,e13.6). The (1x,a12) permits entering the variable name and an equal sign. Typical input thus appears as follows.

```
                iunit1    =    2
                iunit2    =    7
                model     =    0
                etin      =  1.000000e-07
                wtin      =  4.000000e-01
                |         |  | |          |
Column number:  2        12 15 17        26
```

C.1.3 Program Output

The output from program **PROGNAME** consists of a disk file that can be used to plot computed results and, depending upon user preference, printed output that is directed to either the system printer, or to a disk file named **progname.prt**. The name of the plotting-data disk file depends upon the turbulence model used as follows.

komega.dat	k-ω model [Wilcox (1988a)]
multi.dat	Multiscale model [Wilcox (1988b)]
kepsln.dat	k-ϵ model [Launder and Sharma (1974)]
mixlen.dat	Mixing-length model with $\ell_{mix} = \alpha\delta(x)$
csmith.dat	Cebeci-Smith model [Smith and Cebeci (1967)]
blomax.dat	Baldwin-Lomax model [Baldwin and Lomax (1978)]
newmod.dat	User-Defined model

All of the programs make provision for a user-defined model so that the supplied plotting utilities can be used for customized versions of the various programs. Only a subset of the models listed above is supported by any one program. Also, the contents of the plotting-data file are a bit different for each program. See the appropriate section to determine which models are supported by the program of interest and to determine the format of the plotting-data disk file.

The first input parameter for all of the programs is an integer variable named *iunit1*. Setting *iunit1* = 6 will cause the printed output to go to the system line printer for IBM PC implementations. Any other integer will send the printed output to a disk file whose name is **progname.prt**.

Printed output consists of three segments. First, all input data are printed. Next, the maximum error and other flow properties such as spreading rate for free shear flows are printed; this information is also shown on the video display as the run proceeds. Finally, profiles of computed mean-flow and turbulence properties are printed in a self-explanatory manner. The precise format of the printed information differs slightly from one program to the next.

C.2 Free Shear Flows

There are three free shear flow programs on the distribution diskette, viz., Programs **WAKE**, **MIXER** and **JET**. These programs solve the self-similar form of the turbulent-flow equations that are asymptotically approached far downstream. Section 4.5 of the main text presents the equations of motion in physical variables and in similarity form. An additional transformation devised by Rubel and Melnik (1984) has been used that greatly improves numerical accuracy. Specifically, we introduce a new independent variable, ξ, defined in terms of the similarity variable, η, by

$$d\xi = \frac{d\eta}{N(\eta)} \quad \text{or} \quad \frac{d}{d\xi} = N(\eta)\frac{d}{d\eta} \tag{C.1}$$

where $N(\eta)$ is the dimensionless eddy viscosity appearing in the similarity solution. In terms of this variable, the equations assume the following form:

Mean Momentum:

$$\mathcal{V}\frac{d\mathcal{U}}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\eta^j \frac{d\mathcal{U}}{d\xi}\right] = S_u N\mathcal{U} \tag{C.2}$$

Turbulence Kinetic Energy:

$$\mathcal{V}\frac{dK}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\sigma^*\eta^j \frac{dK}{d\xi}\right] = S_k NK + \left(\frac{d\mathcal{U}}{d\xi}\right)^2 - \beta^* K^2 \tag{C.3}$$

***k*-ω Model:**

$$\mathcal{V}\frac{dW}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\sigma\eta^j \frac{dW}{d\xi}\right] = S_w NW + \alpha\frac{W}{K}\left(\frac{d\mathcal{U}}{d\xi}\right)^2 - \beta KW \tag{C.4}$$

$$N = \frac{K}{W} \tag{C.5}$$

***k*-ϵ Model:**

$$\mathcal{V}\frac{dE}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\frac{\eta^j}{\sigma_\epsilon}\frac{dE}{d\xi}\right] = S_e NE + C_{\epsilon 1}\frac{E}{K}\left(\frac{d\mathcal{U}}{d\xi}\right)^2 - C_\mu C_{\epsilon 2} KE \tag{C.6}$$

$$N = C_\mu \frac{K^2}{E} \tag{C.7}$$

Mixing-Length Model:

$$N = \alpha \sqrt{\left| \frac{d\mathcal{U}}{d\xi} \right|} \tag{C.8}$$

where $\mathcal{U}$, $\mathcal{V}$, K, W and E are the transformed velocity components, turbulence energy, specific dissipation rate and dissipation rate, respectively. See Section 4.5 of the main text for additional details on notation and other features of the similarity solution.

This transformation greatly improves numerical accuracy primarily because it removes numerical difficulties associated with the presence of sharp turbulent-nonturbulent interfaces. The edge of the shear layer that occurs at a finite value of η is moved to infinity in terms of the transformed independent variable ξ. Inspection of converged solutions shows a well behaved asymptotic approach to freestream conditions, a feature rarely observed when the equations are solved without the transformation. Consequently, a much tighter convergence criterion can be satisfied. Additionally, there is weaker coupling amongst the turbulence-model equations which also improves the convergence rate.

The only drawback to this transformation is the need to determine an appropriate maximum value of ξ. Using too large or too small a value can slow convergence and even cause the solution to blow up. All of the programs automatically compute the value of ξ_{max} that is suitable for the turbulence models implemented. If you add a new turbulence model, it may be necessary to empirically determine a suitable value for ξ_{max}.

Boundary conditions for these equations must be satisfied at $\xi = 0$ and as $\xi \to \infty$, so that we must solve a two-point boundary-value problem. This is conveniently done by adding unsteady terms to the left-hand sides of Equations (C.2), (C.3), (C.4) and (C.6), making an initial guess, and letting the solution evolve in time. The solution to the desired two-point boundary-value problem is obtained when temporal variations vanish. Thus, for example, we replace the mean-momentum equation by the following.

$$\frac{\partial \mathcal{U}}{\partial t} + \mathcal{V}\frac{\partial \mathcal{U}}{\partial \xi} - \frac{1}{\eta^j}\frac{\partial}{\partial \xi}\left[\eta^j \frac{\partial \mathcal{U}}{\partial \xi}\right] = S_u N \mathcal{U} \tag{C.9}$$

The resulting time-dependent system of equations is solved using implicit Crank-Nicolson differencing that is second-order accurate in both t and ξ. Using 101 mesh points, all of the free shear flow programs require computing times of less than 10 seconds on a 33 MHz 80486-based microcomputer and two minutes on a 10 MHz 8086-based laptop computer with an 8087 math chip.

C.2.1 Program WAKE: Far Wake

Program **WAKE** computes two-dimensional flow in the far wake of an object in an incompressible stream.

Input-parameter description:

Program **WAKE** reads the following input parameters in the order listed below from disk file **wake.dat**. Integer quantities must be formatted according to (1x,a12,i4) while floating-point quantities must be formatted as (1x,a12,e13.6). See Subsection C.1.2 for a sample input-data file.

iunit1	Output-file unit number
	$= 6$ Printed output sent to printer
	$\neq 6$ Printed output saved in disk file **wake.prt**
iunit2	Plotting-data disk file unit number
model	Turbulence-model identification flag
	0 Wilcox k-ω model
	2 Standard k-ϵ model
	9 Mixing-length model with $\ell_{mix} = \alpha\delta(x)$
	99 User-defined model
etin	Freestream value of transformed turbulence kinetic energy
wtin	Freestream value of transformed specific dissipation rate
xih	ω-equation closure coefficient $\hat{\xi}$
jmax	Number of grid points
maxn	Maximum number of timesteps
nedit	Profile-print modulus; profiles are printed every *nedit* steps
nfreq	Short-print modulus; maximum error, shear-layer growth rate, etc. are printed every *nfreq* steps

Output description:

VIDEO OUTPUT includes the timestep number, maximum error and spreading rate every *nfreq* timesteps. PRINTED OUTPUT is sent to the line printer throughout the run if *iunit1* = 6. DISK-FILE OUTPUT is saved in the appropriate plotting-data disk file (see Subsection C.1.3 for file names). The file is **unformatted** and is created using the following statements:

```
jaxi=10
write(iunit2) jmax,jaxi
write(iunit2) (eta(j),uoum(j),j=1,jmax)
```

where eta(j) is η_j and uoum(j) is $U(\eta_j)$. The parameter *jaxi* is used by plotting program **PLOTF**. Additionally, for any value of *iunit1* other than

6, printed output is saved in disk file **wake.prt** at the conclusion of the run.

Comments:

- k-ω model far-wake predictions are very sensitive to the value of *wtin*; optimum agreement with measurements occurs for $wtin = 0.4$.
- Program **WAKE** runs most efficiently for $jmax \geq 101$. A smaller number of grid points tends to slow convergence.

C.2.2 Program MIXER: Mixing Layer

Program **MIXER** computes two-dimensional flow in the mixing layer between two streams of differing velocity, including effects of compressibility.

Input-parameter description:

Program **MIXER** reads the following input parameters in the order listed below from disk file **mixer.dat**. Integer quantities must be formatted according to (1x,a12,i4) while floating-point quantities must be formatted as (1x,a12,e13.6). See Subsection C.1.2 for a sample input-data file.

iunit1 Output-file unit number
- $= 6$ Printed output sent to printer
- $\neq 6$ Printed output saved in disk file **mixer.prt**

iunit2 Plotting-data disk file unit number

model Turbulence-model identification flag
- 0 Wilcox k-ω model
- 2 Standard k-ϵ model
- 9 Mixing-length model with $\ell_{mix} = \alpha\delta(x)$
- 99 User-defined model

etin Freestream value of transformed turbulence kinetic energy

gam Specific heat ratio, γ

prt Turbulent Prandtl number, Pr_T

rho2 Density ratio, ρ_2/ρ_1 where subscript 2 corresponds to the slow stream; required only for incompressible flow

u2ou1 Velocity ratio, U_2/U_1 where subscript 2 corresponds to the slow stream; required only for incompressible flow

wtin Freestream value of transformed specific dissipation rate

xih ω-equation closure coefficient $\hat{\xi}$

xis Dilatation-dissipation closure coefficient ξ^*

xma1 Mach number of the fast stream, M_1

xma2 Mach number of the slow stream, M_2

xmt0 Dilatation-dissipation closure coefficient M_{to}

imach Dilatation-dissipation model flag
- 0 Sarkar's model
- 1 Wilcox's model
- 2 Zeman's model

jmax Number of grid points

jzero Index of grid point at the dividing streamline ($\eta = 0$)

maxn Maximum number of timesteps

nedit Profile-print modulus; profiles are printed every *nedit* steps

nfreq Short-print modulus; maximum error, shear-layer growth rate, etc. are printed every *nfreq* steps

nthick Mixing-layer thickness definition flag for short print
- -1 Bogdanoff's vorticity thickness
- 0 Birch's energy thickness
- 1 Roshko's pitot thickness
- 2 Sullin's momentum thickness

Output description:

VIDEO OUTPUT includes the timestep number, maximum error and spreading rate every *nfreq* timesteps. PRINTED OUTPUT is sent to the line printer throughout the run if *iunit1* = 6. DISK-FILE OUTPUT is saved in the appropriate plotting-data disk file (see Subsection C.1.3 for file names). The file is **unformatted** and is created using the following statements:

```
jaxi=99
write(iunit2) jmax,jaxi
write(iunit2) (yox(j),u(j),j=1,jmax)
```

where yox(j) is η_j and u(j) is $U(\eta_j)$. The parameter *jaxi* is used by plotting program **PLOTF**. Additionally, for any value of *iunit1* other than 6, printed output is saved in disk file **mixer.prt** at the conclusion of the run.

Comments:

- k-ω model mixing-layer predictions are very sensitive to the value of *wtin*; optimum agreement with measurements occurs for *wtin* = 0.5.
- Program **MIXER** runs most efficiently for *jmax* $\geq$ 101. A smaller number of grid points tends to slow convergence.
- Program **MIXER** runs most efficiently for Mach numbers up to 5. It will run for larger values, although smaller timesteps are needed. The timestep can be reduced by changing the value of *cflmax* in the main program. The parameter *cflmax* is the maximum value of the Courant-Friedrichs-Lewy number; the timestep is computed as the product of a number less than or equal to 1, *cflmax* and the maximum timestep required for stability of an explicit scheme. Its default value is 1 for the k-ω and k-ϵ models and 2 for the mixing-length model.

C.2.3 Program JET: Plane, Round and Radial Jet

Program **JET** computes far-field flow for plane, round and radial jets issuing into a quiescent incompressible fluid.

Input-parameter description:

Program **JET** reads the following input parameters in the order listed below from disk file **jet.dat**. Integer quantities must be formatted according to (1x,a12,i4) while floating-point quantities must be formatted as (1x,a12,e13.6). See Subsection C.1.2 for a sample input-data file.

iunit1 Output-file unit number
- = 6 Printed output sent to printer
- ≠ 6 Printed output saved in disk file **jet.prt**

iunit2 Plotting-data disk file unit number

model Turbulence-model identification flag
- 0 Wilcox k-ω model
- 2 Standard k-ϵ model
- 9 Mixing-length model with $\ell_{mix} = \alpha\delta(x)$
- 99 User-defined model

etin Freestream value of transformed turbulence kinetic energy

wtin Freestream value of transformed specific dissipation rate

xih ω-equation closure coefficient $\hat{\xi}$

ipope Round-jet modification flag
- 0 Omit Pope's round-jet correction
- 1 Use Pope's round-jet correction

jaxi Geometry flag
- -1 Radial jet
- 0 Plane jet
- 1 Round jet

jmax Number of grid points

maxn Maximum number of timesteps

nedit Profile-print modulus; profiles are printed every *nedit* steps

nfreq Short-print modulus; maximum error, shear-layer growth rate, etc. are printed every *nfreq* steps

Output description:

VIDEO OUTPUT includes the timestep number, maximum error and spreading rate every *nfreq* timesteps. PRINTED OUTPUT is sent to the line printer throughout the run if *iunit1* = 6. DISK-FILE OUTPUT is saved in the appropriate plotting-data disk file (see Subsection C.1.3 for file names). The file is **unformatted** and is created using the following statements:

```
write(iunit2) jmax,jaxi
write(iunit2) (eta(j),uoum(j),j=1,jmax)
```

where eta(j) is η_j and uoum(j) is $\mathcal{U}(\eta_j)$. The parameter *jaxi* is used by plotting program **PLOTF**. Additionally, for any value of *iunit1* other than 6, printed output is saved in disk file **jet.prt** at the conclusion of the run.

Comments:

- k-ω model jet predictions are very sensitive to the value of *wtin*; optimum agreement with measurements occurs for the following values of *wtin*:

Radial jet,	$wtin = 3$
Plane jet,	$wtin = 5$
Round jet,	$wtin = 50$

- Program **JET** runs most efficiently for $jmax \geq 101$. A smaller number of grid points tends to slow convergence.

C.2.4 Program PLOTF: Plotting Utility

Program **PLOTF** creates video and hardcopy plots of free shear flow velocity profiles computed with programs **WAKE**, **MIXER** and **JET** on IBM PC's and compatibles. The program automatically detects the turbulence model used and the type of shear flow for which computations have been done.

Input-parameter description:

Program **PLOTF** reads the following nine input parameters in the order listed below from disk file **plotf.dat**. Integer quantities must be formatted according to (7x,i6) while floating-point quantities must be formatted as (7x,f6.2). This is similar to the format used for Programs **WAKE**, **MIXER** and **JET**.

mon	Monitor type (see Appendix E)
ifore	Foreground color (see Appendix E)
iback	Background color (see Appendix E)
nprin	Printer type (see Appendix E)
mode	Graphics-mode flag for printers; number of pens for plotters (see Appendix E)
ksize	Plot scaling factor. Using 100 yields a 5″ by 5″ hardcopy plot. Smaller values yield a hardcopy plot reduced by *ksize* per cent. Thus, *ksize* = 50 yields a 2.5″ by 2.5″ plot.
symsiz	Size of experimental data symbols, in inches
height	Physical height, in inches, of the video display
width	Physical width, in inches, of the video display

Next, Program **PLOTF** reads a single, free formatted, line to indicate where hardcopy print is directed and where required font files are located. This line comes immediately after the specified value for *width* and defines the following six additional parameters.

devid	Device name of type character*4; valid devices are LPT1, LPT2, LPT3, COM1, COM2, COM3, COM4
nbaud	Baud rate for a serial port; valid baud rates are 110, 150, 300, 600, 1200, 2400, 4800, 9600
parity	Parity of type character*3 or character*4 for a serial port; valid parity settings are 'even', 'odd' and 'none'
nstop	Number of stop bits for a serial port; either 1 or 2
lword	Word length for a serial port; either 7 or 8
path	Path to font files of type character*n where n $\leq$ 40

Output description:

A 5″ by 5″ video plot (see Figure C.1) is created centered on the screen. When the plot is complete, the following message appears:

Hardcopy output (y/n)?

Enter a *y* or a *Y* to create a hardcopy plot. Pressing any other key terminates the run without creating a hardcopy plot.

Comments:

- The following is a sample input data file for a machine with a standard VGA monitor and an HP DeskJet connected to serial port COM1:.

```
mon    =    18              (Standard VGA monitor)
ifore  =    15              (Bright-white foreground)
iback  =     1              (Blue background)
nprin  =     2              (HP DeskJet)
mode   =     3              (300 dots per inch resolution)
ksize  =   100              (Full size plot)
symsiz=   .080              (.08" experimental data symbols)
height= 7.500               (7.5" high video display)
width = 9.250               (9.25" wide video display)
'com1' , 9600 , 'none' , 1 , 8, 'd:\fonts\'
```

  The last line indicates the printer is connected to serial port COM1: and the port is set at 9600 baud, no parity, 1 stop bit and 8 data bits. Also, the plotting font file **sppfnt.002** is located in directory **fonts** on drive **d:**.

  IMPORTANT: The terminating character in the path name must be a backslash, i.e., \.

- If disk file **plotf.dat** is not available, Program **PLOTF** uses the following set of default values :

  mon = 18, *ifore* = 15, *iback* = 1, *nprin* = 24, *mode* = 39, *ksize* = 100, *symsiz* = .08, *height* = 7.5, *width* = 9.25, *devid* = 'LPT1', *path* = '\'

  Note that *nbaud*, *parity*, *nstop* and *lword* are not used for parallel ports. Note also that the default path is the **root directory** of the current drive.

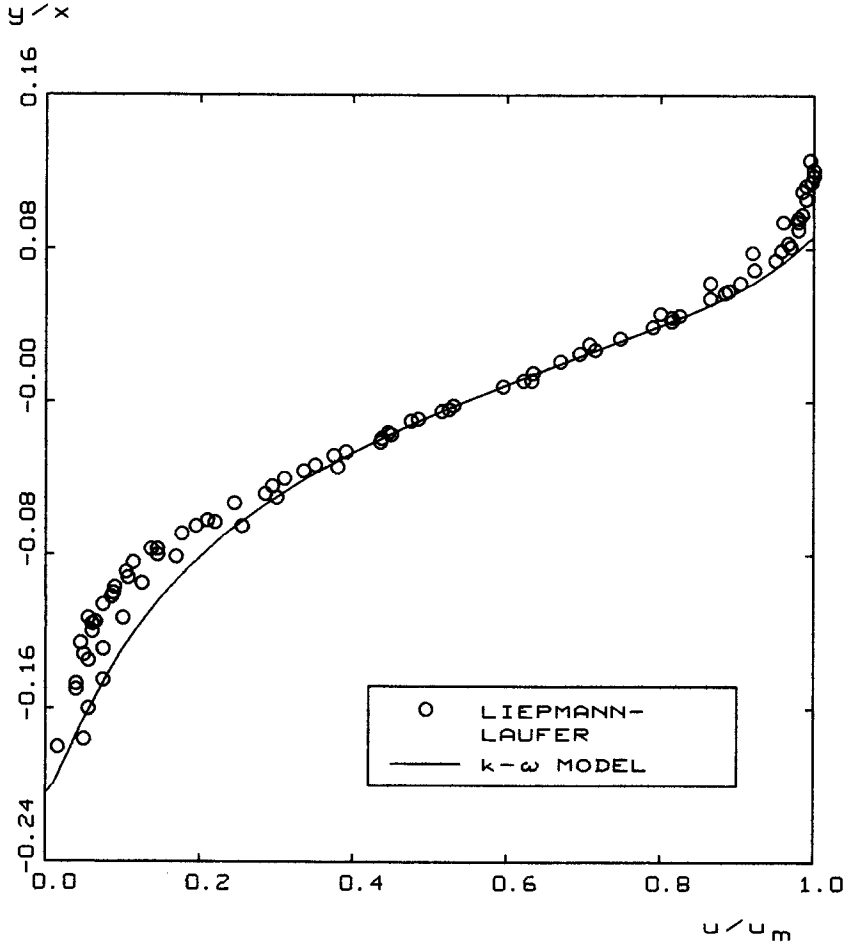

Figure C.1: Sample plot created by Program PLOTF.

C.3 Channel and Pipe Flow

Program **PIPE** can be used to compute incompressible channel flow or pipe flow with several turbulence models. Subsections 3.5.1 and 4.8.1 of the main text describe the channel- and pipe-flow equations. No additional transformations are introduced in Program **PIPE**.

As with the free shear flow programs, we add unsteady terms to the turbulence-model equations to facilitate solution of the two-point boundary-value problem. However, the momentum equation is solved at each timestep by trapezoidal-rule integration. For example, in the case of a two-equation model, we advance the turbulence parameters in time. Then, after updating the eddy viscosity, we determine the velocity by integration of the following equation.

$$\frac{dU^+}{dy^+} = \frac{1 - y^+/R^+}{1 + \mu_T^+} \tag{C.10}$$

All notation in Equation (C.10) is identical to that used in Subsections 3.5.1 and 4.8.1 of the main text.

The only other subtle feature of the program is the way the specific dissipation rate, ω, in the Wilcox (1988a) k-ω model and the Wilcox (1988b) multiscale model is computed close to the solid boundary. To eliminate numerical error associated with computing the singular behavior of ω for perfectly-smooth and slightly-rough surfaces, the exact asymptotic behavior of ω is prescribed close to the surface (see Subsection 7.2.1). That is, we use the fact that, for $y^+ < 2.5$, $\omega^+ = \nu\omega/u_\tau^2$ is given by:

$$\omega^+ \to \frac{N_\omega}{(y^+)^2} \qquad \text{as} \quad y \to 0 \quad \text{(smooth wall)} \tag{C.11}$$

$$\omega^+ \to \frac{\omega_w^+}{\left[1 + \sqrt{\frac{\omega_w^+}{N_\omega}}\, y^+\right]^2} \qquad \text{as} \quad y \to 0 \quad \text{(rough wall)} \tag{C.12}$$

where

$$N_\omega = \begin{cases} 6/\beta, & \text{without viscous corrections} \\ 2/\beta^*, & \text{with viscous corrections} \end{cases} \tag{C.13}$$

The exact analytical behavior of ω is imposed for a prescribed number of mesh points, *jskip*, next to the surface. Using 201 mesh points, Program **PIPE** requires computing times of less than 10 seconds on a 33 MHz 80486-based microcomputer and two minutes on a 10 MHz 8086-based laptop computer with an 8087 math chip.

C.3.1 Program PIPE: Channel and Pipe Flow

Program **PIPE** computes incompressible, fully-developed flow in either a two-dimensional channel or a pipe of circular cross section.

Input-parameter description:

Program **PIPE** reads the following input parameters from disk file **pipe.dat** in the order listed below. Integer quantities must be formatted according to (1x,a12,i4) while floating-point quantities must be formatted as (1x,a12,e13.6). See Subsection C.1.2 for a sample input-data file.

iunit1 Output-file unit number
- $= 6$ Printed output sent to printer
- $\neq 6$ Printed output saved in disk file **pipe.prt**

iunit2 Plotting-data disk file unit number

model Turbulence-model identification flag
- 0 Wilcox k-ω model
- 1 Wilcox multiscale model
- 10 Cebeci-Smith model
- 11 Baldwin-Lomax model
- 99 User-defined model

omegw Surface value of dimensionless specific dissipation rate, ω_w^+

retau $Re_\tau = R^+ = u_\tau R/\nu$, dimensionless channel half height or pipe radius

xih ω-equation closure coefficient $\hat{\xi}$

yone Value of y^+ at the first grid point above the wall

iruff Surface-roughness flag
- 0 Rough surface
- 1 Smooth surface

jaxi Geometry flag
- 0 Channel flow
- 1 Pipe flow

jmax Number of grid points

jskip Grid-point number below which the exact asymptotic solution for specific dissipation rate is used

maxn Maximum number of timesteps

nedit Profile-print modulus; profiles are printed every *nedit* steps

nfreq Short-print modulus; maximum error, Reynolds number, skin friction, etc. are printed every *nfreq* steps

nvisc Viscous-modification flag
- $= 0$ No k-ω/multiscale model viscous modifications
- $\neq 0$ Use k-ω/multiscale model viscous modifications

Output description:

VIDEO OUTPUT includes timestep number, maximum error, Reynolds number and skin friction every *nfreq* timesteps. PRINTED OUTPUT is sent to the line printer throughout the run if *iunit1* = 6. DISK-FILE OUTPUT is saved in the appropriate plotting-data disk file (see Subsection C.1.3 for file names). The file is **unformatted** and is created using the following statements.

```
      write(iunit2) jmax,jaxi,retau,reh,cf
      if(model.eq.10.or.model.eq.11) then
        write(iunit2) (yoh(j),uoum(j),yplus(j),uplus(j),
     *                 upvp(j),j=1,jmax)
      else
        do 60 j=2,jmax
          diss(j)=-betas*bbeta(j)*et(j)*wt(j)
          prod(j)=upvp(j)*dudy(j)
60      continue
        if(nvisc.ne.0) diss(1)=2.*diss(2)-diss(3)
        prod(1)=0.
        write(iunit2) (yoh(j),uoum(j),yplus(j),uplus(j),
     *          upvp(j),et(j),diss(j),prod(j),j=1,jmax)
      endif
```

The quantity *reh* is Reynolds number based on the average velocity and channel height/pipe diameter, while *cf* is skin friction based on average velocity. Also, yoh(j) is y/R, uoum(j) is $U(y)/U(0)$, yplus(j) is y^+, uplus(j) is U^+, upvp(j) is τ_{xy}/τ_w, et(j) is k/u_τ^2, diss(j) is $\beta^*\nu\omega k/u_\tau^4$, and prod(j) is $\nu\tau_{xy}(dU/dy)/u_\tau^4$. Additionally, for any value of *iunit1* other than 6, printed output is saved in disk file **pipe.prt** at the conclusion of the run.

C.3.2 Program PLOTP: Plotting Utility

Program **PLOTP** creates video and hardcopy plots of channel- or pipe-flow properties computed with program **PIPE** on IBM PC's and compatibles. The program automatically detects the turbulence model used and the type flow for which computations have been done.

Input-parameter description:

Program **PLOTP** reads the following seven input parameters in the order listed below from disk file **plotp.dat**. Integer quantities must be formatted according to (7x,i6) while floating-point quantities must be formatted as (7x,f6.2). This is similar to the format used for Program **PIPE**.

mon	Monitor type (see Appendix E)
ifore	Foreground color (see Appendix E)
iback	Background color (see Appendix E)
nprin	Printer type (see Appendix E)
mode	Graphics-mode flag for printers; number of pens for plotters (see Appendix E)
ksize	Plot scaling factor. Using 100 yields a full-size hardcopy plot. Smaller values yield a hardcopy plot reduced by *ksize* per cent. Thus, *ksize* = 50 yields a half-size plot.
symsiz	Size of experimental data symbols, in inches

Next, Program **PLOTP** reads a single, free formatted, line to indicate where hardcopy print is directed and where required font files are located. This line comes immediately after the specified value for *symsiz* and defines the following six additional parameters.

devid	Device name of type character*4; valid devices are LPT1, LPT2, LPT3, COM1, COM2, COM3, COM4
nbaud	Baud rate for a serial port; valid baud rates are 110, 150, 300, 600, 1200, 2400, 4800, 9600
parity	Parity of type character*3 or character*4 for a serial port; valid parity settings are 'even', 'odd' and 'none'
nstop	Number of stop bits for a serial port; either 1 or 2
lword	Word length for a serial port; either 7 or 8
path	Path to font files of type character*n where n $\leq$ 40

In addition to disk file **plotp.dat**, an optional disk file including skin-friction data can be included. This information includes results from a series of runs and must be prepared by the user. The name of the optional disk file depends upon the turbulence model used, viz.:

cfkw.dat	Wilcox k-ω model
cfms.dat	Wilcox multiscale model
cfcs.dat	Cebeci-Smith model
cfbl.dat	Baldwin-Lomax model
cfus.dat	User-supplied model

The first line of the disk file must contain the number of input data pairs with format (i4). This line is followed by Reynolds number/skin friction data pairs with format (2e13.6). For example, a series of 10 runs with the Baldwin-Lomax model for channel flow yields the following results.

```
  10
1.021000e 03 1.919000e-02
2.625000e 03 1.161000e-02
5.510000e 03 8.538000e-03
1.020000e 04 6.922000e-03
1.413000e 04 6.252000e-03
2.718000e 04 5.308000e-03
4.071000e 04 4.828000e-03
6.429000e 04 4.355000e-03
8.835000e 04 4.100000e-03
1.003000e 05 3.990000e-03
```

For channel flow, confine Reynolds number to the range $10^3 \leq Re_H \leq 10^5$. For pipe flow, Reynolds number should be in the range $10^3 \leq Re_D \leq 10^6$.

Output description:

A video plot with six graphs (see Figure C.2) is created on the screen. When the plot is complete, the following message appears:

Hardcopy output (y/n)?

Enter a y or a Y to create a hardcopy plot. Pressing any other key terminates the run without creating a hardcopy plot.

Comments:

- The following is a sample input data file for a machine with a standard VGA monitor and an HP DeskJet connected to serial port COM1:.

```
mon   =    18          (Standard VGA monitor)
ifore =    15          (Bright-white foreground)
iback =     1          (Blue background)
nprin =     2          (HP DeskJet)
```

```
mode  =     3            (300 dots per inch resolution)
ksize =   100            (Full size plot)
symsiz=  .080            (.08" experimental data symbols)
'com1' , 9600 , 'none' , 1 , 8, 'd:\fonts\'
```

The last line indicates the printer is connected to serial port COM1: and the port is set at 9600 baud, no parity, 1 stop bit and 8 data bits. Also, the plotting font file **sppfnt.002** is located in directory **fonts** on drive **d:**.

IMPORTANT: The terminating character in the path name must be a backslash, i.e., \.

- If disk file **plotp.dat** is not available, Program **PLOTP** uses the following set of default values :

  $mon = 18$, $ifore = 15$, $iback = 1$, $nprin = 24$, $mode = 39$, $ksize = 100$, $symsiz = .08$, $devid =$ 'LPT1', $path =$ '\'

  Note that *nbaud*, *parity*, *nstop* and *lword* are not used for parallel ports. Note also that the default path is the **root directory** of the current drive.

- The following data are used for comparison with computed results:

Channel Flow — $Re_\tau < 287$,	Mansour, et al. (1988) [$Re_\tau = 180$]
Channel Flow — $Re_\tau \geq 287$,	Mansour, et al. (1988) [$Re_\tau = 395$]
Pipe Flow — All Re_τ,	Laufer (1952) [$Re_\tau = 1058$]

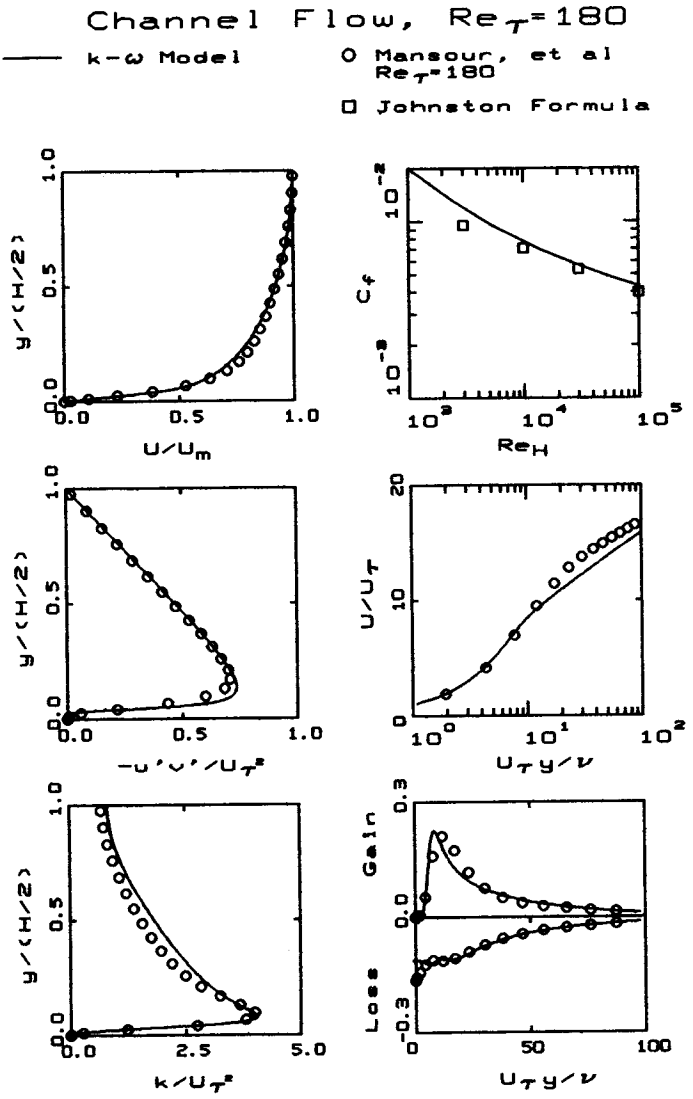

Figure C.2: Sample plot created by Program PLOTP.

C.4 Boundary-Layer Perturbation Analysis

Programs **SUBLAY** and **DEFECT** can be used to compute turbulence-model predicted flow properties in the incompressible viscous sublayer and defect layer, respectively. Section 4.6 of the main text describes the sublayer and defect-layer equations. No additional transformations are introduced in Program **SUBLAY**. Program **DEFECT** uses the Rubel-Melnik (1984) transformation.

As with the free shear flow and pipe-flow programs, we add unsteady terms to the turbulence-model equations to facilitate solution of the two-point boundary-value problems appropriate for the sublayer and the defect layer. In Program **SUBLAY**, the momentum equation is solved at each timestep by trapezoidal-rule integration. For example, in the case of a two-equation model, we advance the turbulence parameters in time. Then, after updating the eddy viscosity, we determine the velocity by integration of the following equation.

$$\frac{dU^+}{dy^+} = \frac{1}{1+\mu_T^+} \tag{C.14}$$

All notation in Equation (C.14) is identical to that used in Section 4.6 of the main text.

The only other subtle feature of Program **SUBLAY** is the way the specific dissipation rate, ω, in the Wilcox (1988a) k-ω model and the Wilcox (1988b) multiscale model is computed close to the solid boundary. To eliminate numerical error associated with computing the singular behavior of ω for perfectly-smooth and slightly-rough surfaces, the exact asymptotic behavior of ω is prescribed close to the surface (see Subsection 7.2.1). That is, we use the fact that, for $y^+ < 2.5$, $\omega^+ = \nu\omega/u_\tau^2$ is given by:

$$\omega^+ \rightarrow \frac{N_\omega}{(y^+)^2} \quad \text{as} \quad y \rightarrow 0 \quad \text{(smooth wall)} \tag{C.15}$$

$$\omega^+ \rightarrow \frac{\omega_w^+}{\left[1+\sqrt{\frac{\omega_w^+}{N_\omega}}\, y^+\right]^2} \quad \text{as} \quad y \rightarrow 0 \quad \text{(rough wall)} \tag{C.16}$$

where

$$N_\omega = \begin{cases} 6/\beta, & \text{without viscous corrections} \\ 2/\beta^*, & \text{with viscous corrections} \end{cases} \tag{C.17}$$

The exact analytical behavior of ω is imposed for a prescribed number of mesh points, *jskip*, next to the surface.

In terms of the transformation devised by Rubel and Melnik, the defect-layer equations are as follows. Note that to avoid numerical difficulties, Program **DEFECT** uses a small nonzero value for $K_0(\xi)$ when $\xi \to \infty$. The nonvanishing boundary conditions quoted for $W_0(\xi)$ and $E_0(\xi)$ as $\xi \to \infty$ are the only choices consistent with the similarity solution.

All Models:

$$\eta = \int_{-\infty}^{\xi} N_0(\xi')\, d\xi' \tag{C.18}$$

$$\frac{d^2U_1}{d\xi^2} + (1+\beta_T)\eta\frac{dU_1}{d\xi} + \beta_T N_0 U_1 = 0 \tag{C.19}$$

$$\sigma^*\frac{d^2K_0}{d\xi^2} + (1+\beta_T)\eta\frac{dK_0}{d\xi} + \sqrt{\beta^*}\left[\left(\frac{dU_1}{d\xi}\right)^2 - K_0^2\right] = 0 \tag{C.20}$$

***k*-ω Model:**

$$\sigma\frac{d^2W_0}{d\xi^2} + (1+\beta_T)\eta\frac{dW_0}{d\xi} + (1+2\beta_T)N_0W_0 + \frac{\sqrt{\beta^*}}{K_0}\left[\alpha\left(\frac{dU_1}{d\xi}\right)^2 - \frac{\beta}{\beta^*}K_0^2\right]W_0 = 0 \tag{C.21}$$

$$W_0(\xi) \to \frac{(1+2\beta_T)\sqrt{\beta^*}}{\beta} \quad \text{as} \quad \xi \to \infty \tag{C.22}$$

$$N_0 = K_0/W_0 \tag{C.23}$$

***k*-ϵ Model:**

$$\sigma_\epsilon^{-1}\frac{d^2E_0}{d\xi^2} + (1+\beta_T)\eta\frac{dE_0}{d\xi} + (1+2\beta_T)N_0E_0 + \frac{\sqrt{C_\mu}}{K_0}\left[C_{\epsilon 1}\left(\frac{dU_1}{d\xi}\right)^2 - C_{\epsilon 2}K_0^2\right]E_0 = 0 \tag{C.24}$$

$$E_0(\xi)/K_0(\xi) \to \frac{(1+2\beta_T)}{C_{\epsilon 2}\sqrt{C_\mu}} \quad \text{as} \quad \xi \to \infty \tag{C.25}$$

$$N_0 = K_0^2/E_0 \tag{C.26}$$

C.4.1 Program SUBLAY: Viscous Sublayer

Program **SUBLAY** computes incompressible viscous sublayer flow, including surface roughness and surface mass transfer.

Input-parameter description:

Program **SUBLAY** reads the following input parameters from disk file **sublay.dat** in the order listed below. Integer quantities must be formatted according to (1x,a12,i4) while floating-point quantities must be formatted as (1x,a12,e13.6). See Subsection C.1.2 for a sample input-data file.

iunit1 Output-file unit number
  $= 6$ Printed output sent to printer
  $\neq 6$ Printed output saved in disk file **sublay.prt**
iunit2 Plotting-data disk file unit number
model Turbulence-model identification flag
  0 Wilcox k-ω model
  1 Wilcox multiscale model
  99 User-defined model
omegw Surface value of dimensionless specific dissipation rate, ω_w^+
vwplus Dimensionless vertical velocity at the surface, v_w/u_τ.
xih ω-equation closure coefficient $\hat{\xi}$
ymax Maximum value of y^+
yone Value of y^+ at the first grid point above the wall
iruff Surface-roughness flag
  0 Rough surface
  1 Smooth surface
jmax Number of grid points
jskip Grid-point number below which the exact asymptotic solution for specific dissipation rate is used
maxn Maximum number of timesteps
nedit Profile-print modulus; profiles are printed every *nedit* steps
nfreq Short-print modulus; maximum error and $B = u^+ - \frac{1}{\kappa}\ell n y^+$ at $y^+ = ymax$ are printed every *nfreq* steps
nvisc Viscous-modification flag
  $= 0$ No k-ω/multiscale model viscous modifications
  $\neq 0$ Use k-ω/multiscale model viscous modifications

Output description:

VIDEO OUTPUT includes the timestep number, maximum error, and the constant in the law of the wall, $B = u^+ - \frac{1}{\kappa}\ell n y^+$, at $y^+ = ymax$ every *nfreq* timesteps. PRINTED OUTPUT is sent to the line printer throughout the run if *iunit1* $= 6$. DISK-FILE OUTPUT is saved in the appropriate

plotting-data disk file (see Subsection C.1.3 for file names). The file is **unformatted** and is created using the following statements.

```
      do 50 j=2,jmax
        wtm(j)=-betas*bbeta(j)*et(j)*wt(j)
        if(model.ne.1) tau(j)=eps(j)*dudy(j)
        etm(j)=tau(j)*dudy(j)
50    continue
      if(nvisc.ne.0) wtm(1)=2.*wtm(2)-wtm(3)
      etm(1)=0.
      write(iunit2) jmax
      write(iunit2) (yplus(j),uplus(j),wtm(j),
                     etm(j),j=1,jmax)
```

The quantity yplus(j) is y^+, uplus(j) is U^+, wtm(j) is $\beta^* \nu \omega k / u_\tau^4$, and etm(j) is $\nu \tau_{xy} (dU/dy)/u_\tau^4$. Additionally, for any value of *iunit1* other than 6, printed output is saved in disk file **sublay.prt** at the conclusion of the run.

C.4.2 Program DEFECT: Defect Layer

Program **DEFECT** computes properties of the incompressible defect-layer including effects of pressure gradient.

Input-parameter description:

Program **DEFECT** reads the following input parameters from disk file **defect.dat** in the order listed below. Integer quantities must be formatted according to (1x,a12,i4) while floating-point quantities must be formatted as (1x,a12,e13.6). See Subsection C.1.2 for a sample input-data file.

iunit1	Output-file unit number
	$= 6$ Printed output sent to printer
	$\neq 6$ Printed output saved in disk file **defect.prt**
iunit2	Plotting-data disk file unit number
model	Turbulence-model identification flag
	0 Wilcox k-ω model
	2 Standard k-ϵ model
	99 User-defined model
betat	Pressure-gradient parameter, β_T
jmax	Number of grid points
maxn	Maximum number of timesteps
nedit	Profile-print modulus; profiles are printed every *nedit* steps
nfreq	Short-print modulus; maximum error, wake strength, etc. are printed every *nfreq* steps

Output description:

VIDEO OUTPUT includes the timestep number, maximum error, wake strength, etc. every *nfreq* timesteps. PRINTED OUTPUT is sent to the line printer throughout the run if *iunit1* = 6. DISK-FILE OUTPUT is saved in the appropriate plotting-data disk file (see Subsection C.1.3 for file names). The file is **unformatted** and is created using the following statements.

```
write(iunit2) jmax,betat
write(iunit2) (eta(j),u(j),j=1,jmax)
```

where betat is β_T, eta(j) is η_j and u(j) is $[U_e - U(\eta_j)]/u_\tau$. Additionally, for any value of *iunit1* other than 6, printed output is saved in disk file **defect.prt** at the conclusion of the run.

Comments:

- The program has been optimized for $-0.5 \leq \beta_T \leq 15$.

C.4.3 Program PLOTS: Sublayer Plotting Utility

Program **PLOTS** creates video and hardcopy plots of viscous sublayer properties computed with program **SUBLAY** on IBM PC's and compatibles. The program automatically detects the turbulence model used.

Input-parameter description:

Program **PLOTS** reads the following seven input parameters from disk file **plots.dat** in the order listed below. Integer quantities must be formatted according to (7x,i6) while floating-point quantities must be formatted as (7x,f6.2). This is similar to the format used for Program **SUBLAY**.

mon Monitor type (see Appendix E)
ifore Foreground color (see Appendix E)
iback Background color (see Appendix E)
nprin Printer type (see Appendix E)
mode Graphics-mode flag for printers; number of pens for plotters (see Appendix E)
ksize Plot scaling factor. Using 100 yields a full-size hardcopy plot. Smaller values yield a hardcopy plot reduced by *ksize* per cent. Thus, *ksize* = 50 yields a half-size plot.
symsiz Size of experimental data symbols, in inches

Next, Program **PLOTS** reads a single, free formatted, line to indicate where hardcopy print is directed and where required font files are located. This line comes immediately after the specified value for *symsiz* and defines the following six additional parameters.

devid Device name of type character*4; valid devices are LPT1, LPT2, LPT3, COM1, COM2, COM3, COM4
nbaud Baud rate for a serial port; valid baud rates are 110, 150, 300, 600, 1200, 2400, 4800, 9600
parity Parity of type character*3 or character*4 for a serial port; valid parity settings are 'even', 'odd' and 'none'
nstop Number of stop bits for a serial port; either 1 or 2
lword Word length for a serial port; either 7 or 8
path Path to font files of type character*n where n $\leq$ 40

Output description:

A video plot with two graphs (see Figure C.3) is created on the screen. When the plot is complete, the following message appears:

Hardcopy output (y/n)?

Enter a *y* or a *Y* to create a hardcopy plot. Pressing any other key terminates the run without creating a hardcopy plot.

Comments:

- The following is a sample input data file for a machine with a standard VGA monitor and an HP DeskJet connected to serial port COM1:.

```
mon    =    18            (Standard VGA monitor)
ifore  =    15            (Bright-white foreground)
iback  =     1            (Blue background)
nprin  =     2            (HP DeskJet)
mode   =     3            (300 dots per inch resolution)
ksize  =   100            (Full size plot)
symsiz=   .080            (.08" experimental data symbols)
'com1' , 9600 , 'none' , 1 , 8, 'd:\fonts\'
```

  The last line indicates the printer is connected to serial port COM1: and the port is set at 9600 baud, no parity, 1 stop bit and 8 data bits. Also, the plotting font file **sppfnt.002** is located in directory **fonts** on drive **d:**.

  IMPORTANT: The terminating character in the path name must be a backslash, i.e., \.

- If disk file **plots.dat** is not available, Program **PLOTS** uses the following set of default values :

  mon = 18, *ifore* = 15, *iback* = 1, *nprin* = 24, *mode* = 39, *ksize* = 100, *symsiz* = .08, *devid* = 'LPT1', *path* = '\'

  Note that *nbaud*, *parity*, *nstop* and *lword* are not used for parallel ports. Note also that the default path is the **root directory** of the current drive.

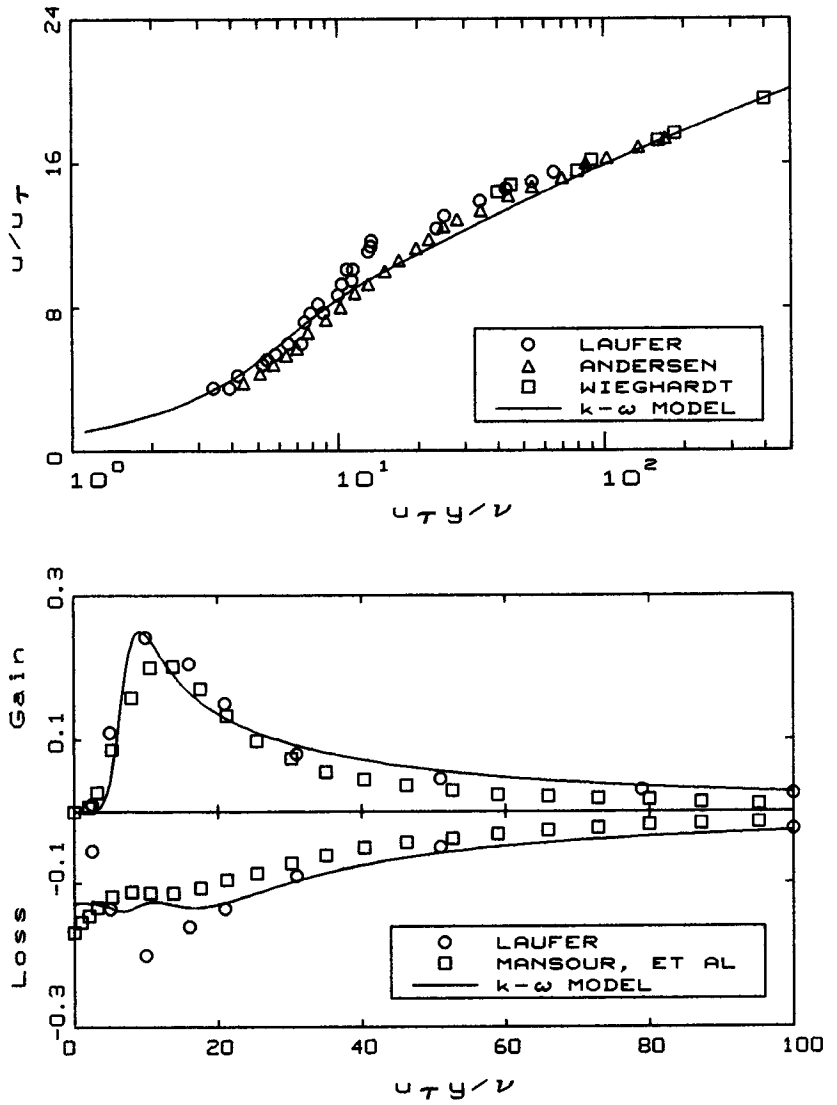

Figure C.3: Sample plot created by Program PLOTS.

C.4.4 Program PLOTD: Defect-Layer Plotting Utility

Program **PLOTD** creates video and hardcopy plots of the defect-layer velocity profile computed with program **DEFECT** on IBM PC's and compatibles. The program automatically detects the turbulence model used.

Input-parameter description:

Program **PLOTD** reads the following nine input parameters from disk file **plotd.dat** in the order listed below. Integer quantities must be formatted according to (7x,i6) while floating-point quantities must be formatted as (7x,f6.2). This is similar to the format used for Program **DEFECT**.

- *mon* Monitor type (see Appendix E)
- *ifore* Foreground color (see Appendix E)
- *iback* Background color (see Appendix E)
- *nprin* Printer type (see Appendix E)
- *mode* Graphics-mode flag for printers; number of pens for plotters (see Appendix E)
- *ksize* Plot scaling factor. Using 100 yields a 5″ by 5″ hardcopy plot. Smaller values yield a hardcopy plot reduced by *ksize* per cent. Thus, *ksize* = 50 yields a 2.5″ by 2.5″ plot.
- *symsiz* Size of experimental data symbols, in inches
- *height* Physical height, in inches, of the video display
- *width* Physical width, in inches, of the video display

Next, Program **PLOTD** reads a single, free formatted, line to indicate where hardcopy print is directed and where required font files are located. This line comes immediately after the specified value for *width* and defines the following six additional parameters.

- *devid* Device name of type character*4; valid devices are LPT1, LPT2, LPT3, COM1, COM2, COM3, COM4
- *nbaud* Baud rate for a serial port; valid baud rates are 110, 150, 300, 600, 1200, 2400, 4800, 9600
- *parity* Parity of type character*3 or character*4 for a serial port; valid parity settings are 'even', 'odd' and 'none'
- *nstop* Number of stop bits for a serial port; either 1 or 2
- *lword* Word length for a serial port; either 7 or 8
- *path* Path to font files of type character*n where n $\leq$ 40

Output description:

A 5″ by 5″ video plot (see Figure C.4) is created centered on the screen. When the plot is complete, the following message appears:

Hardcopy output (y/n)?

Enter a *y* or a *Y* to create a hardcopy plot. Pressing any other key terminates the run without creating a hardcopy plot.

Comments:

- The following is a sample input data file for a machine with a standard VGA monitor and an HP DeskJet connected to serial port COM1:.

```
mon    =    18              (Standard VGA monitor)
ifore  =    15              (Bright-white foreground)
iback  =     1              (Blue background)
nprin  =     2              (HP DeskJet)
mode   =     3              (300 dots per inch resolution)
ksize  =   100              (Full size plot)
symsiz=   .080              (.08" experimental data symbols)
height= 7.500               (7.5" high video display)
width  = 9.250              (9.25" wide video display)
'com1' , 9600 , 'none' , 1 , 8, 'd:\fonts\'
```

  The last line indicates the printer is connected to serial port COM1: and the port is set at 9600 baud, no parity, 1 stop bit and 8 data bits. Also, the plotting font file **sppfnt.002** is located in directory **fonts** on drive **d:**.

  IMPORTANT: The terminating character in the path name must be a backslash, i.e., \.

- If disk file **plotd.dat** is not available, Program **PLOTD** uses the following set of default values :

  mon = 18, *ifore* = 15, *iback* = 1, *nprin* = 24, *mode* = 39, *ksize* = 100, *symsiz* = .08, *height* = 7.5, *width* = 9.25, *devid* = 'LPT1', *path* = '\'

  Note that *nbaud*, *parity*, *nstop* and *lword* are not used for parallel ports. Note also that the default path is the **root directory** of the current drive.

- The following experimental data from Coles and Hirst (1969) are used:

$\beta_T < 0$,	Flow 2800: Herring-Norbury (β_T = -0.5)
$0 \leq \beta_T \leq 2$,	Flow 1400: Wieghardt (β_T = 0.0)
$2 < \beta_T < 7$,	Flow 2600: Bradshaw (β_T = 5.2)
$\beta_T \geq 7$,	Flow 2300: Clauser (β_T = 8.7)

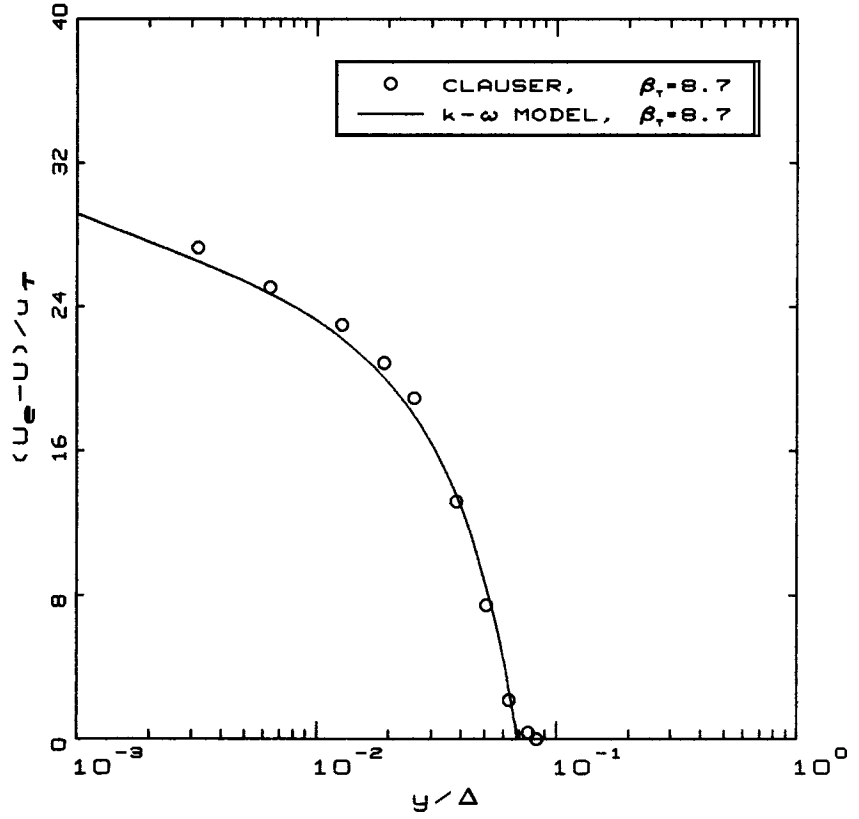

Figure C.4: Sample plot created by Program PLOTD.

C.5 Miscellaneous Routines

This section includes several utility routines called by the various programs described in the preceding sections. They implement several standard mathematical procedures such as the Runge-Kutta predictor-corrector method for integrating a system of ordinary differential equations, the Newton iteration method for solving transcendental equations and Thomas' algorithm for solving a tridiagonal matrix system. These routines are of general usefulness in computational fluid mechanics as well as for the programs that are the main topic of this appendix. We assume the user is familiar with these algorithms and thus include only instructions on use of the subroutines. Users unfamiliar with the techniques should refer to texts such as Abramowitz and Stegun (1965), Hildebrand (1976), Chapra and Canale (1985) and Press, Flannery, Teukolsky and Vetterling (1987).

There is one routine, **NAMSYS**, that is called by all of the programs. The purpose of this routine is to make the programs portable to a variety of computers. The routine sets several system-dependent variables used in opening files, assigning video and hardcopy units, etc. The standardization of Fortran-77, including VAX extensions, by most compiler writers makes it possible to confine virtually all system-dependent parameters to this single subroutine.

For the convenience of users with IBM PC and compatible microcomputers, a library named **dcw.lib** containing the routines described in this section is included on the distribution diskette. The library is compatible with Microsoft Fortran, Version 5.00. The routines have been compiled with the **/FPi** option that generates code that will use a math coprocessor if it is available and provide software emulation if no coprocessor is available. To link the programs described in the preceding chapters (say, for example, **progname**) use the following command.

link progname,,nul,dcw /e;

C.5.1 Function ERF: Error Function

Function **ERF** computes the error function $erf(x)$ defined by

$$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^{\infty} e^{-t^2}\, dt$$

Usage: <real variable> = *erf(x)*

Input-parameter description:

x	Error function argument, x

Output-parameter description:

erf	Computed value, $erf(x)$

Comments:

- A polynomial approximation is used. When $x < 0$, the function uses the fact that

$$erf(-x) = -erf(x)$$

C.5.2 Subroutine NAMSYS: Fortran Portability

Subroutine **NAMSYS** returns several system-dependent and compiler-specific parameters to aid in portability of Fortran programs.

Usage: call namsys(icpu,iin,iv,msdos,newfil,pform)

Input-parameter description:

icpu CPU identification flag
- 0 SVS Fortran (680x0, 80x86)
- 1 Lahey/Microsoft Fortran (8088, 80x86)
- 2 VAX/VMS
- 3 SUN Fortran (SUN Workstation, Definicon SPARC)
- 4 Cray Fortran/Unicos (Cray X/MP, Y/MP)
- 5 Silicon Graphics Iris

Output-parameter description:

iin Input data file logical unit number; set to unit 15 for all CPU's

iv Standard console unit number; set to 5 for all CPU's

msdos Open-printer flag
- 0 Printer opened as *'prn'*
- 1 Don't open *'prn'*

newfil Character*7 string used in opening new files
- *'new'* if compiler writes over an existing file
- *'unknown'* for Ansi-77 standard operation

pform Character*9 string used as format type for printer output that is redirected to a disk file
- *'printer'* for SVS Fortran
- *'print'* for SUN Fortran
- *'formatted'* for all others

Comments:

- This routine is currently configured for the CPU's and Fortran compilers listed in the Input-parameter description. Other CPU's and compilers can be included by adding the appropriate statements to the routine. The following page includes an abbreviated listing of **NAMSYS**.

```
      subroutine namsys(icpu,iin,iv,msdos,newfil,pform)
      character newfil*7,pform*9
      iin=15
      iv=5
c---------------------------------
c  SVS Fortran (680x0 and 80x86)
      if(icpu.eq.0) then
        msdos=0
        pform='printer'
        newfil='new'
c---------------------------------
c  Lahey/Microsoft Fortran (80x86)
      elseif(icpu.eq.1) then
        msdos=0
        pform='formatted'
        newfil='unknown'
c---------------------------------
c              VAX/VMS
      elseif(icpu.eq.2) then
        msdos=1
        pform='formatted'
        newfil='new'
c---------------------------------
c   SUN Fortran...SUN Workstation
      elseif(icpu.eq.3) then
        msdos=0
        pform='print'
        newfil='unknown'
c---------------------------------
c     Cray Fortran...Unicos
      elseif(icpu.eq.4) then
        msdos=1
        pform='formatted'
        newfil='unknown'
c---------------------------------
c     Silicon Graphics Iris
      elseif(icpu.eq.5) then
        msdos=0
        pform='formatted'
        newfil='unknown'
c---------------------------------
c      Error...say so and quit
    else
      write(*,*) 'icpu = ',icpu,' is not supported!!!'
      stop
    endif
    return
    end
```

C.5.3 Subroutine RKGS: Runge-Kutta Integration

Subroutine **RKGS** solves a system of first-order ordinary differential equations defined in an external subroutine with given initial values. The system of equations is of the form

$$\frac{dy_i}{dx} = f_i(x, y_j) \quad \text{for} \quad x_0 \leq x \leq x_1$$

Usage: call rkgs(prmt,y,yp,ndim,ihlf,fct,outp,aux)

Input-parameter description:

fct The name of an external subroutine used to compute the right hand side vector, $\mathbf{f} = yp$. The argument list to this subroutine must be *(x,y,yp)*, and the subroutine must leave the values of x and y unchanged.

ndim Number of equations in the system

prmt Input and output array with dimension ≥ 5, that specifies interval and accuracy parameters and that serves for communication between output subroutine *outp* (furnished by the user) and subroutine **RKGS**. With the exception of *prmt(5)*, the components are not destroyed by subroutine **RKGS** and they are:

- *prmt(1)* Lower bound of the interval, x_0
- *prmt(2)* Upper bound of the interval, x_1
- *prmt(3)* Δx_0, initial increment of x
- *prmt(4)* Upper error bound. If the absolute error is greater than *prmt(4)*, the increment is halved. If the increment is less than Δx_0 and the absolute error is less than *prmt(4)*/50, the increment is doubled. If desired, the user can change *prmt(4)* in output subroutine *outp*.
- *prmt(5)* Termination parameter. Subroutine **RKGS** initially sets *prmt(5)* = 0. In order to terminate subroutine **RKGS** at any output point, change *prmt(5)* to a nonzero value in subroutine *outp*.

y Input vector of initial values, y_0.

yp Input vector of error weights; the sum of its components must be equal to 1.

Output-parameter description:

aux An *(8 x ndim)* auxiliary storage array

ihlf Number of bisections of the initial increment. If *ihlf* exceeds 10. subroutine **RKGS** returns *ihlf* = 11. Additionally, if $\Delta x_0 = 0$. **RKGS** returns with *ihlf* = 12 while if Δx_0 and $(x_1 - x_0)$ differ in sign, **RKGS** returns *ihlf* = 13.

outp The name of an external subroutine used for program output. The form of its argument list must be *(x,y,yp,ihlf,ndim,prmt)*. None of these parameters (except, if necessary, *prmt(4)*, *prmt(5)*, ...) should be changed by subroutine *outp*. If *prmt(5)* is changed to a nonzero value, subroutine **RKGS** is terminated.

prmt Input and output array with dimension ≥ 5, that specifies interval and accuracy parameters and that serves for communication between output subroutine *outp* (furnished by the user) and subroutine **RKGS**.

prmt(6) Although not required by subroutine **RKGS**, additional parameters can be included in array *prmt* provided its dimension is declared to be > 5. Such parameters may be useful for passing values to the routine calling **RKGS** which are obtained by special manipulations of output data in subroutine *outp*.

y Output vector of computed values, **y**, at intermediate points.

yp Output vector of derivatives, corresponding to function values *y* at point *x*.

Comments:

- Computation is done using the fourth-order accurate Runge-Kutta method with Gill's modification. Accuracy is tested by comparing the results of the procedure with single and double increments of the independent variable, Δx. Subroutine **RKGS** automatically adjusts the increment during the computation by halving or doubling Δx. The procedure terminates and returns to the calling routine, if any of the following conditions occur.

  1. More than 10 bisections of the initial increment are necessary to achieve satisfactory accuracy (error flag *ihlf* = 11);
  2. Either the initial increment $\Delta x_0 = 0$ or has the wrong sign (error flags *ihlf* = 12 or *ihlf* = 13);
  3. The end of the integration interval, $x = x_1$, reached;
  4. Subroutine *outp* has changed *prmt(5)* to a nonzero value.

- The calling routine must declare the two user-supplied subroutines *outp(x,y,yp,ihlf,ndim,prmt)* and *fct(x,y,yp)* as external by including the statements **EXTERNAL FCT** and **EXTERNAL OUTP**.

C.5.4 Subroutine RTNI: Newton's Iterations

Subroutine **RTNI** solves a general equation of the form $f(x) = 0$ using Newton's iteration method. The function $f(x)$ is specified by the user in a SUBROUTINE subprogram.

Usage: call rtni(x,f,fp,fct,xst,eps,iend,ier)

Input-parameter description:

eps	Upper bound on the error in x
fct	Name of the external subroutine used. It computes $f = f(x)$ and $fp = df/dx$ for a given value of x. Its argument list must be *(x,f,fp)*.
iend	Maximum number of iterations allowed
xst	Initial guess of the root x_{st}

Output-parameter description:

f	Computed value of $f(x)$ at root x
fp	Computed value of df/dx at root x
ier	Error flag 0 No error 1 No convergence after *iend* iterations 2 $df/dx = 0$ encountered
x	Computed root of $f(x) = 0$

Comments:

- Solution of the equation $f(x) = 0$ is obtained using Newton's iteration method, which starts at the initial guess x_{st} of the root x. Convergence is quadratic if the value of df/dx at root x is not equal to zero. One iteration step requires one evaluation of $f(x)$ and one evaluation of df/dx.

- The subroutine returns with the error flag $ier = 2$ if, at any iteration step, df/dx vanishes.

- The calling routine must declare the subroutine *fct(x,f,fp)* as external by including an **EXTERNAL FCT** statement.

C.5.5 Subroutine TRI: Tridiagonal Matrix Inversion

Subroutine **TRI** solves the tridiagonal matrix equation

$$a_i x_{i-1} + b_i x_i + c_i x_{i+1} = d_i \quad for \quad I_l + 1 \leq i \leq I_u - 1$$

subject to either Dirichlet or Neumann boundary conditions.

Usage: call tri(a,b,c,d,il,iu,ibcl,dl,ibcu,du)

Input-parameter description:

a	Array of matrix elements left of the diagonal, a_i; destroyed in the computation.
b	Array of matrix diagonal elements, b_i; destroyed in the computation.
c	Array of matrix elements right of the diagonal, c_i.
d	Input right-hand-side vector, d_i. This vector is replaced by the solution vector, x_i.
dl	d_{I_l}, lower boundary-condition value
du	d_{I_u}, upper boundary-condition value
ibcl	Lower boundary-condition flag 0 Dirichlet, $x_{I_l} = d_{I_l}$ 1 Neumann, $x_{I_l+1} - x_{I_l} = d_{I_l}$
ibcu	Upper boundary-condition flag 0 Dirichlet, $x_{I_u} = d_{I_u}$ 1 Neumann, $x_{I_u} - x_{I_u-1} = d_{I_u}$
il	I_l, lower bound on i
iu	I_u, upper bound on i; also length of a_i, b_i, c_i and d_i.

Output-parameter description:

d	Solution vector, x_i. This vector replaces input vector d_i.

Comments:

- The solution is obtained using Thomas' algorithm. The input arrays a_i, b_i, c_i and d_i need only be specified for indices in the range

$$I_l + 1 \leq i \leq I_u - 1$$

C.6 Diskette Contents

Flowfield Program Source:

cpuid	Include file specifying CPU type
defect.for	Source code for Program **DEFECT**
jet.for	Source code for Program **JET**
mixer.for	Source code for Program **MIXER**
pipe.for	Source code for Program **PIPE**
sublay.for	Source code for Program **SUBLAY**
wake.for	Source code for Program **WAKE**

Miscellaneous Routine Source:

erf.for	Source code for Function **ERF**
namsys.for	Source code for Subroutine **NAMSYS**
rkgs.for	Source code for Subroutine **RKGS**
rtni.for	Source code for Subroutine **RTNI**
tri.for	Source code for Subroutine **TRI**

Input Data for Flowfield Programs:

defect.dat	Input data for Program **DEFECT**
jet.dat	Input data for Program **JET**
mixer.dat	Input data for Program **MIXER**
pipe.dat	Input data for Program **PIPE**
sublay.dat	Input data for Program **SUBLAY**
wake.dat	Input data for Program **WAKE**

IBM PC Executable Plotting Programs:

plotd.exe	Plotting program for Program **DEFECT**
plotf.exe	Plotting program for **JET**, **MIXER** and **WAKE**
plotp.exe	Plotting program for Program **PIPE**
plots.exe	Plotting program for Program **SUBLAY**

Input Data for Plotting Programs:

plotd.dat	Input data for Program **PLOTD**
plotf.dat	Input data for Program **PLOTF**
plotp.dat	Input data for Program **PLOTP**
plots.dat	Input data for Program **PLOTS**

Miscellaneous Files:

dcw.lib	Miscellaneous routines library
sppfnt.002	Scientific font file for plotting programs