Companion Software: Turbulence Modeling for CFD

Third Edition

by

David C. Wilcox

DCW Industries

Companion Software: Turbulence Modeling for CFD

Last Revision: November 7, 2006

DCW Industries, Inc.
5354 Palm Drive, La Cañada, California 91011
818/790-3844 (FAX) 818/952-1272
World Wide Web: http://www.dcwindustries.com

This book was prepared with LaTeX as implemented by Personal TeX, Inc. of Mill Valley, California.

Contents

Preface

The companion CD for *Turbulence Modeling for CFD* contains a collection of computer programs that can be used to develop and validate turbulence models. The CD includes the following.

- FORTRAN source code
- Executable programs built with the Lahey Fortran-90 compiler
- Menu-driven Visual C++ input-data preparation programs that should function on all versions of the Microsoft Windows operating system
- Visual C++ plotting programs to display program output in graphical form
- This document, which contains detailed technical and user information

The programs supplied on the CD fall into five categories...

1. Free Shear Flows: Programs **JET**, **MIXER** and **WAKE** solve for free-shear-flow farfield behavior (Chapter 1).

2. Channel and Pipe Flow: Program **PIPE** solves for two-dimensional channel flow and axisymmetric pipe flow under fully-developed conditions (Chapter 1).

3. Boundary-Layer Perturbation Analysis: Programs **DEFECT** and **SUBLAY** generate solutions for the classical defect layer and the viscous sublayer (Chapter 1).

4. Boundary Layers: Program **EDDYBL** is a two-dimensional/axisymmetric boundary-layer program applicable to compressible boundary layers under laminar, transitional and turbulent flow conditions (Chapter 2).

5. Separated Flows: Program **EDDY2C** is a two-dimensional/axisymmetric Reynolds-Averaged Navier Stokes (RANS) program applicable to compressible separated flows under laminar and turbulent flow conditions (Chapter 3).

The menu-driven input-data preparation programs include default input values that can be modified as needed. Additionally, for Programs **EDDYBL** and **EDDY2C**, the directory named **Input Data** includes input-data files for many of the test cases discussed throughout *Turbulence Modeling for CFD*. This document indicates the flow each file corresponds to.

NOTE: If you discover any bugs in the software or errors in this documentation, please report what you have found by sending email through DCW Industries' Internet site at **http://www.dcwindustries.com**. As they become available, revisions and/or corrections to the software will be downloadable from the site, so you might want to check for updates from time to time.

Chapter 1

General Utility Programs

1.1 Overview

The software described in this chapter solves for: (a) free-shear-flow farfield behavior; (b) channel and pipe flow; and, (c) detailed defect-layer and sublayer behavior. In all cases, accurate algorithms are used that guarantee grid-independent solutions on any computer. 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 chapter 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.

To use Program **PROGNAME** as supplied, you can do everything you want by running Program **PROGNAME_DATA**. It is written in Microsoft Visual C++ and should function properly on a personal computer with any version of the Windows operating system.

1.2 Program JET: Plane, Round and Radial Jets

Program **JET** computes farfield flow for plane, round and radial jets issuing into a quiescent incompressible fluid. Figure 1.1 shows the opening screen for Program **JET_DATA**, whose operation is straightforward. You can use your mouse or TAB key to select a given input quantity and then enter whatever changes you want. The box at the bottom of the menu provides help when you position your mouse on a given input parameter. Just be sure to press the ENTER key, the TAB key, or click on another input quantity to confirm an input-data change.

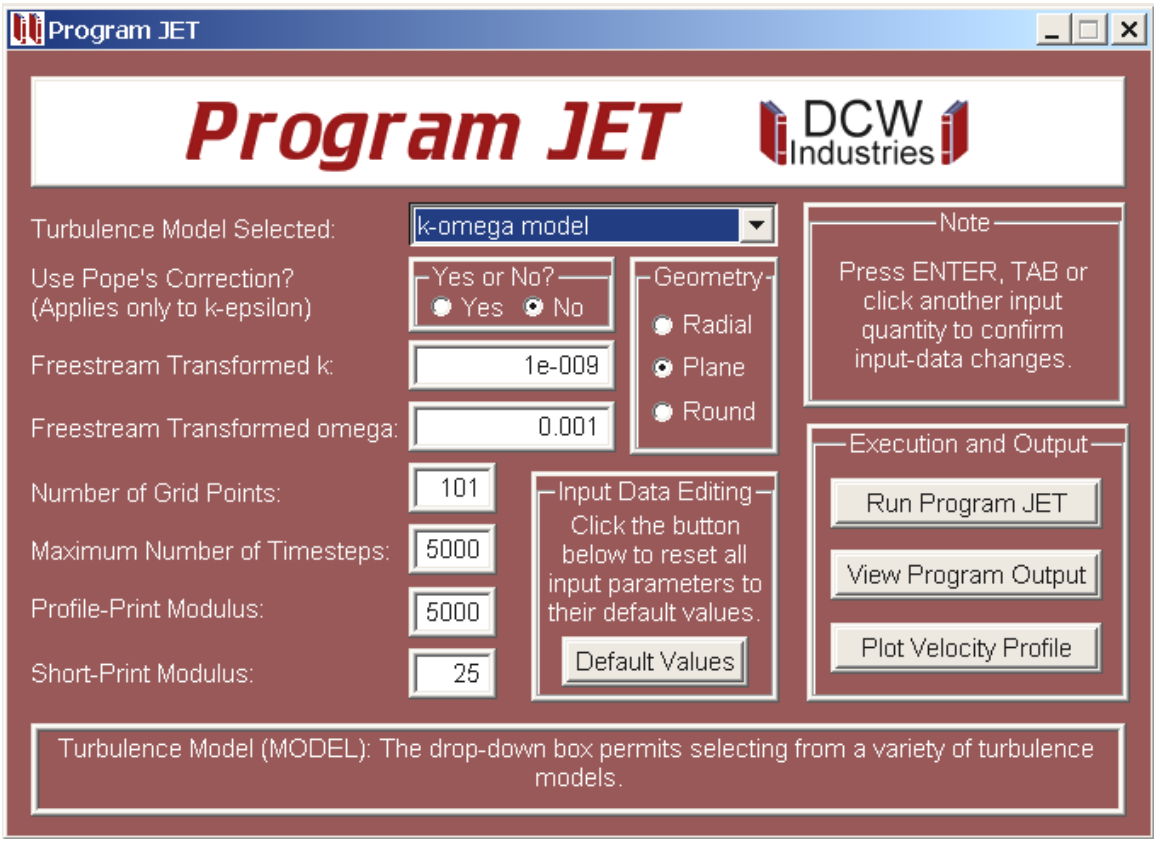

Figure 1.1: *Opening menu of Program* **JET_DATA**.

Default Values Button: Click on this button to reset the default values. The first time you run **JET_DATA**, the menu appears exactly as in the figure. All values shown are the built-in default values. When you close the program, it will save input-data file **jet.dat** with any changes you have made. When you run the program the next time, it will read **jet.dat** and the values of all input parameters will be those that you set in your last run.

Run Program JET Button: Click on this button to run Program **JET**. Control will pass to **JET** and the screen will appear as shown in Figure 1.2. Video output includes timestep number, maximum error and spreading rate. Press the ENTER key or close the window to return to the main menu.

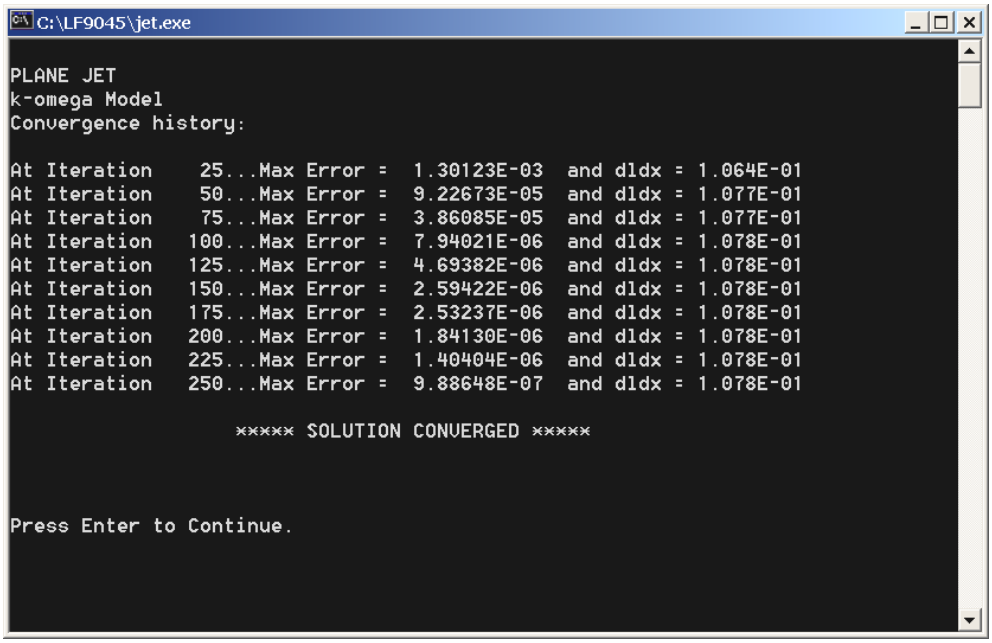

Figure 1.2: *Output from Program* **JET**.

View Program Output Button: After returning to the main menu, click on this button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **jet.prt**.

Plot Velocity Profile Button: Click on this button to create a plot (see Figure 1.3). Program **JET** saves plotting data in a disk file named **plotj.dat**. The file is created using the following Fortran statements:

```
      write(iunit2,230) jmax,jaxi
      write(iunit2,240) (eta(j),uoum(j),j=1,jmax)
 230  format(1x,2i4)
 240  format(1p2e14.6)
```

where *jmax* is the number of grid points, *eta(j)* is $\eta = y/x$ and *uoum(j)* is $\mathcal{U}(\eta)$. The parameter *jaxi* is -1 for a radial jet, 0 for a plane jet and 1 for a round jet.

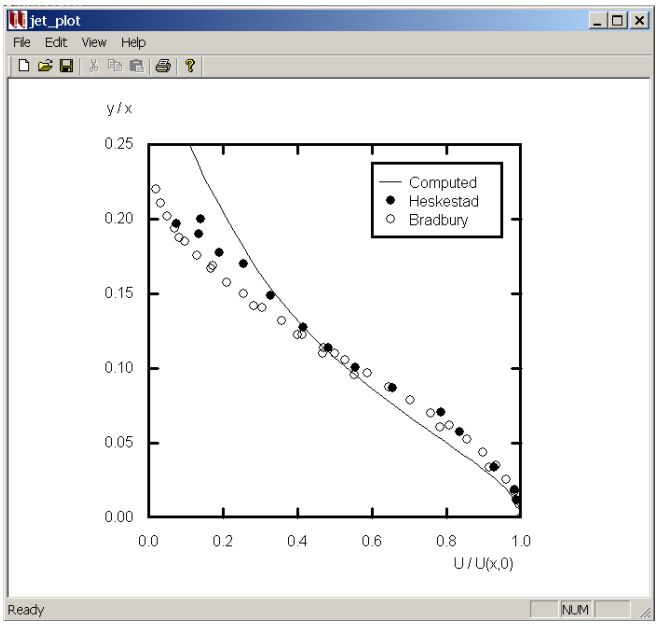

Figure 1.3: *Output from Program* **JET_PLOT**.

1.3 Program MIXER: Mixing Layer

Program **MIXER** computes two-dimensional flow in the mixing layer between two streams of differing velocity, including effects of compressibility. Figure 1.4 shows the opening screen for Program **MIXER_DATA**, whose operation is straightforward. You can use your mouse or TAB key to select a given input quantity and then enter whatever changes you want. The box at the bottom of the menu provides help when you position your mouse on a given input parameter. Just be sure to press the ENTER key, the TAB key, or click on another input quantity to confirm an input-data change.

Default Values Button: Click on this button to reset the default values. The first time you run **MIXER_DATA**, the menu appears exactly as in the figure. All values shown are the built-in default values. When you close the program, it will save input-data file **mixer.dat** with any changes you have made. When you run the program the next time, it will read **mixer.dat** and the values of all input parameters will be those that you set in your last run.

Run Program MIXER Button: Click on this button to run Program **MIXER**. Control will pass to **MIXER** and the screen will appear as shown in Figure 1.5. Video output includes timestep number, maximum error and spreading rate. Press the ENTER key or close the window to return to the main menu.

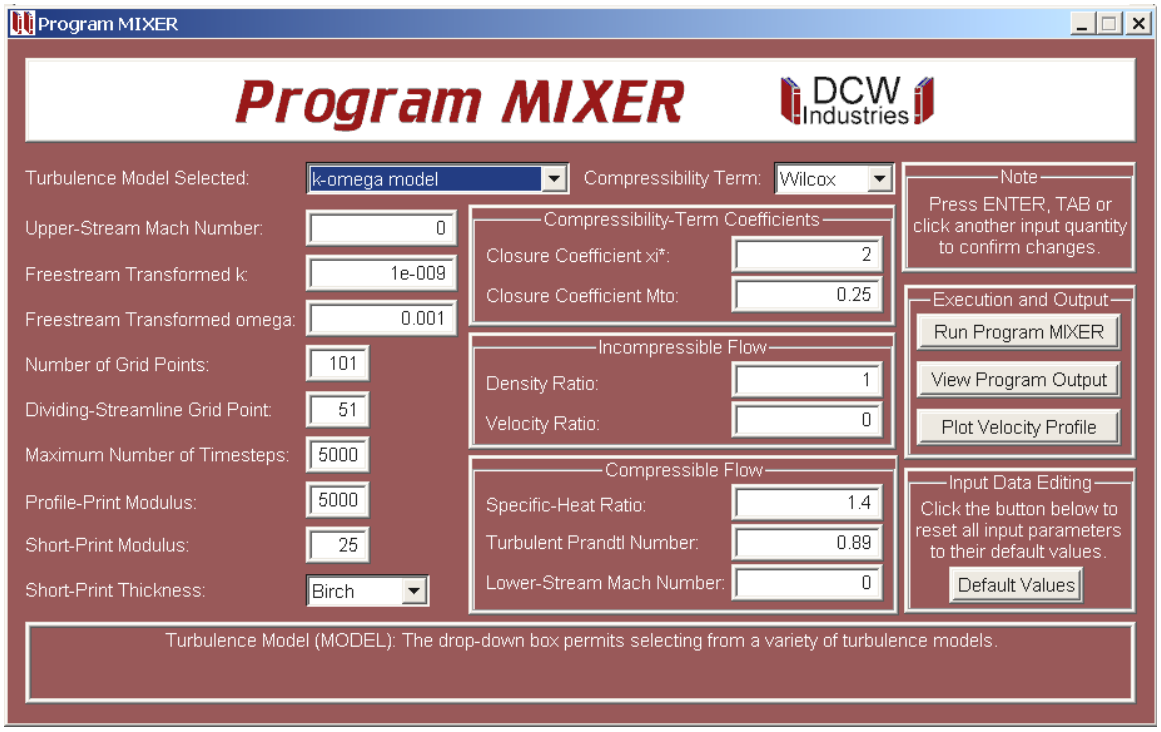

Figure 1.4: *Opening menu of Program* **MIXER_DATA**.

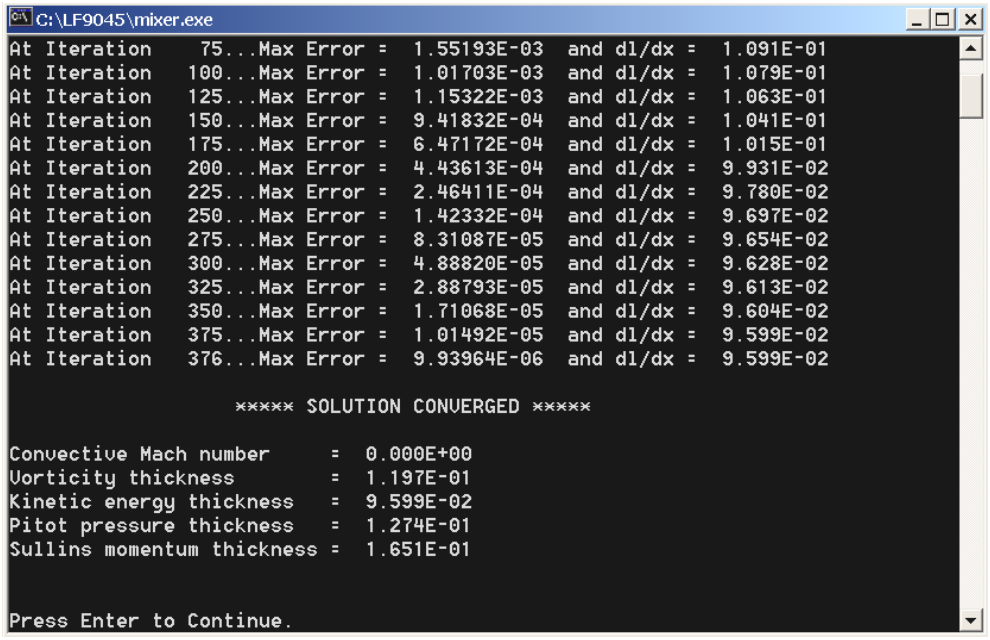

Figure 1.5: *Output from Program* **MIXER**.

View Program Output Button: After returning to the main menu, click on this button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **mixer.prt**.

Plot Velocity Profile Button: Click on this button to create a plot (see Figure 1.6). Program **MIXER** saves plotting data in a disk file named **plotm.dat**. The file is created using the following statements:

```
      write(iunit2,210) jmax
      write(iunit2,220) (yox(j),u(j),j=1,jmax)
 210  format(1x,i4)
 220  format(1p2e14.6)
```

where *jmax* is the number of grid points, *yox(j)* is $\eta = y/x$ and *u(j)* is $\mathcal{U}(\eta)$.

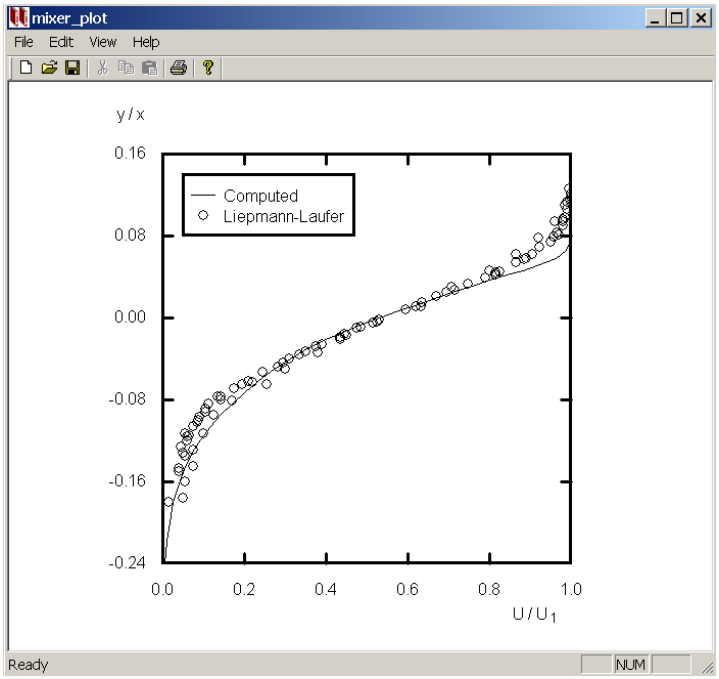

Figure 1.6: *Output from Program* **MIXER_PLOT**.

1.4 Program WAKE: Far Wake

Program **WAKE** computes two-dimensional flow in the far wake of an object in an incompressible stream. Figure 1.7 shows the opening screen for Program **WAKE_DATA**, whose operation is straightforward. You can use your mouse or TAB key to select a given input quantity and then enter whatever changes you want. The box at the bottom of the menu provides help when you position your mouse on a given input parameter. Just be sure to press the ENTER key, the TAB key, or click on another input quantity to confirm an input-data change.

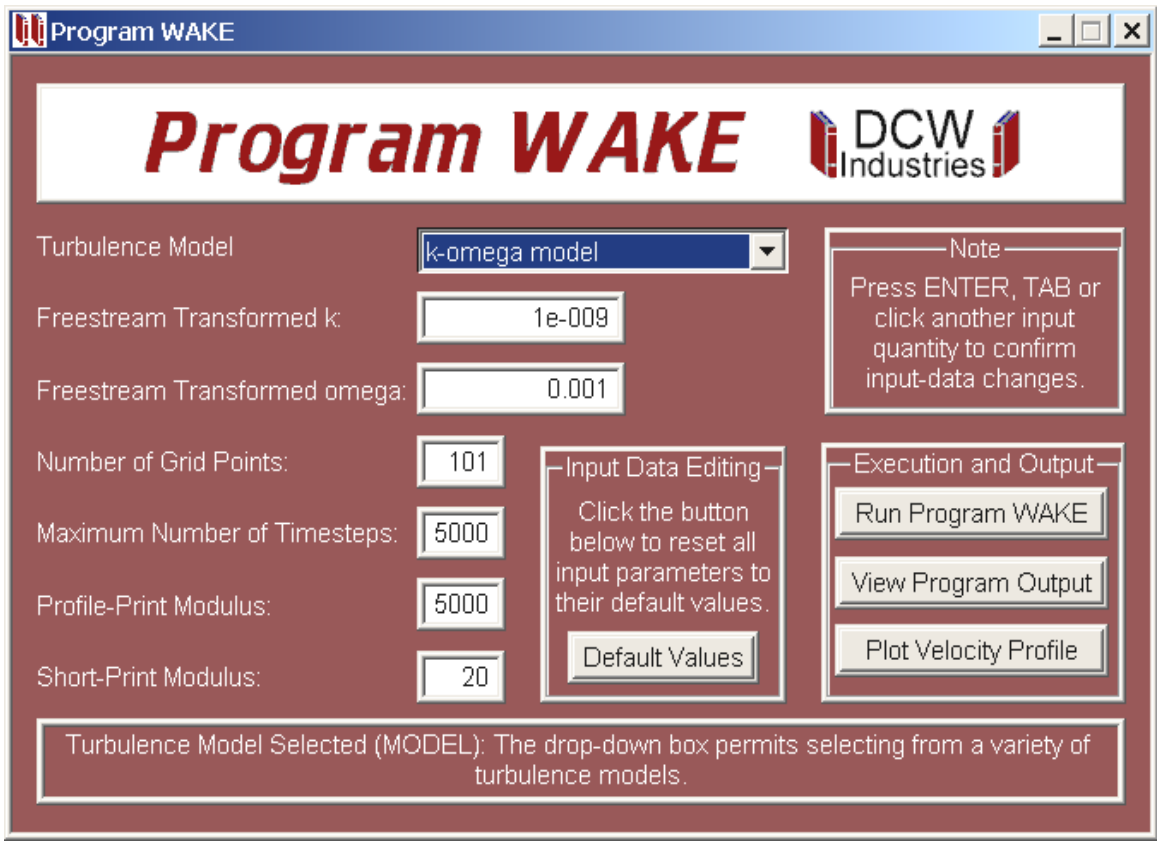

Figure 1.7: *Opening menu of Program* **WAKE_DATA**.

Default Values Button: Click on this button to reset the default values. The first time you run **WAKE_DATA**, the menu appears exactly as in the figure. All values shown are the built-in default values. When you close the program, it will save input-data file **wake.dat** with any changes you have made. When you run the program the next time, it will read **wake.dat** and the values of all input parameters will be those that you set in your last run.

Run Program WAKE Button: Click on this button to run Program **WAKE**. Control will pass to **WAKE** and the screen will appear as shown in Figure 1.8. Video output includes timestep number, maximum error and spreading rate. Press the ENTER key or close the window to return to the main menu.

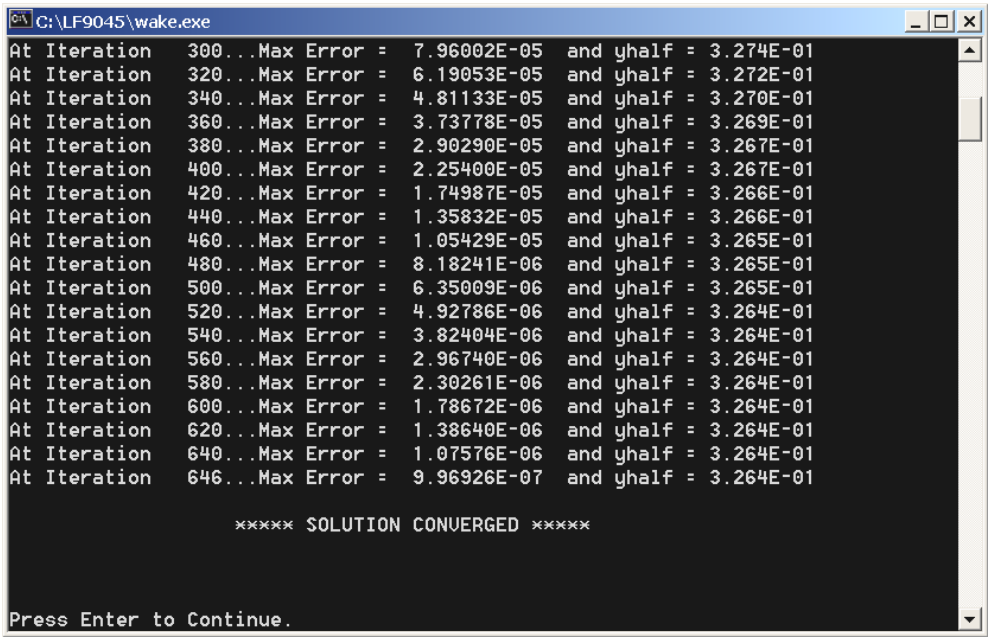

Figure 1.8: *Output from Program* **WAKE**.

View Program Output Button: After returning to the main menu, click on this button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **wake.prt**.

Plot Velocity Profile Button: Click on this button to create a plot (see Figure 1.9). Program **WAKE** saves plotting data in a disk file named **plotw.dat**. The file is created using the following statements:

```
      write(iunit2,200) jmax
      write(iunit2,210) (eta(j),uoum(j),j=1,jmax)
 200  format(1x,i4)
 210  format(1p2e14.6)
```

where *jmax* is the number of grid points, *eta(j)* is η and *uoum(j)* is $\mathcal{U}(\eta)$.

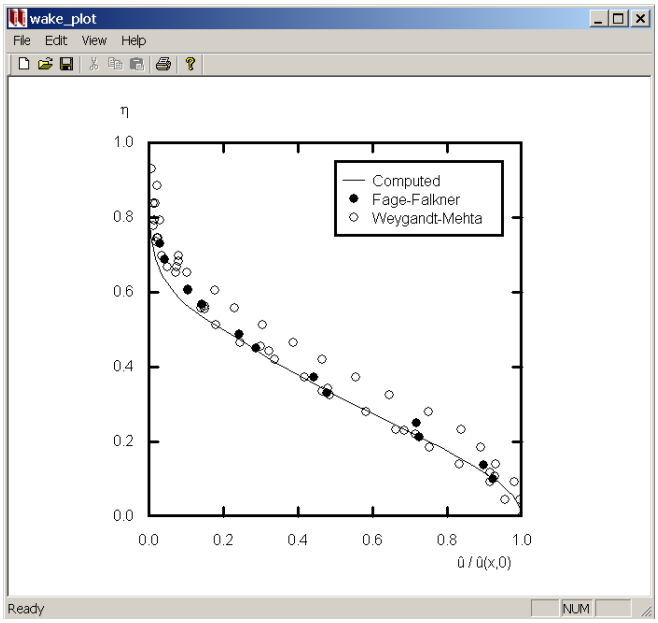

Figure 1.9: *Output from Program* **WAKE_PLOT**.

1.5 Program PIPE: Channel and Pipe Flow

Program **PIPE** can be used to compute incompressible channel flow or pipe flow with several turbulence models. Figure 1.10 shows the opening screen for Program **PIPE_DATA**, whose operation is straightforward. You can use your mouse or TAB key to select a given input quantity and then enter whatever changes you want. The box at the bottom of the menu provides help when you position your mouse on a given input parameter. Just be sure to press the ENTER key, the TAB key, or click on another input quantity to confirm an input-data change.

Default Values Button: Click on this button to reset the default values. The first time you run **PIPE_DATA**, the menu appears exactly as in the figure. All values shown are the built-in default values. When you close the program, it will save input-data file **pipe.dat** with any changes you have made. When you run the program the next time, it will read **pipe.dat** and the values of all input parameters will be those that you set in your last run.

Run Program PIPE Button: Click on this button to run Program **PIPE**. Control will pass to **PIPE** and the screen will appear as shown in Figure 1.11. Video output includes timestep number, maximum error, Reynolds number and skin friction. Press the ENTER key or close the window to return to the main menu.

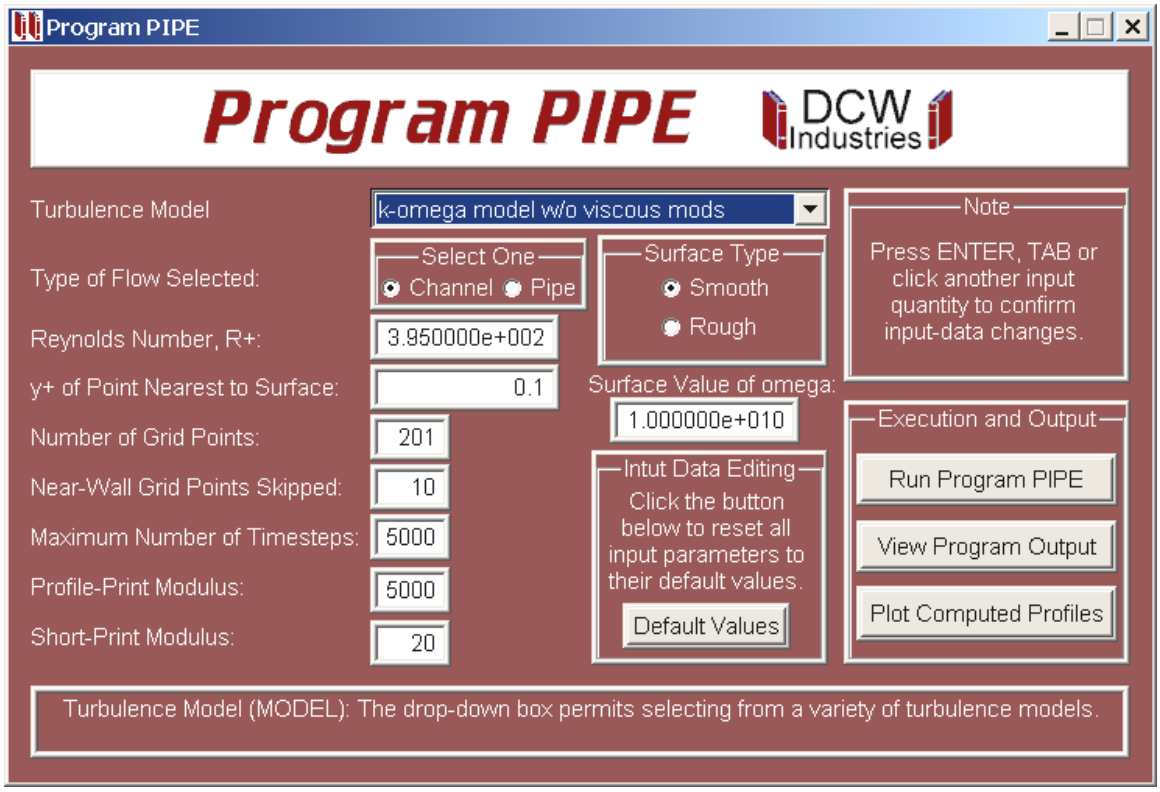

Figure 1.10: *Opening menu of Program* **PIPE_DATA**.

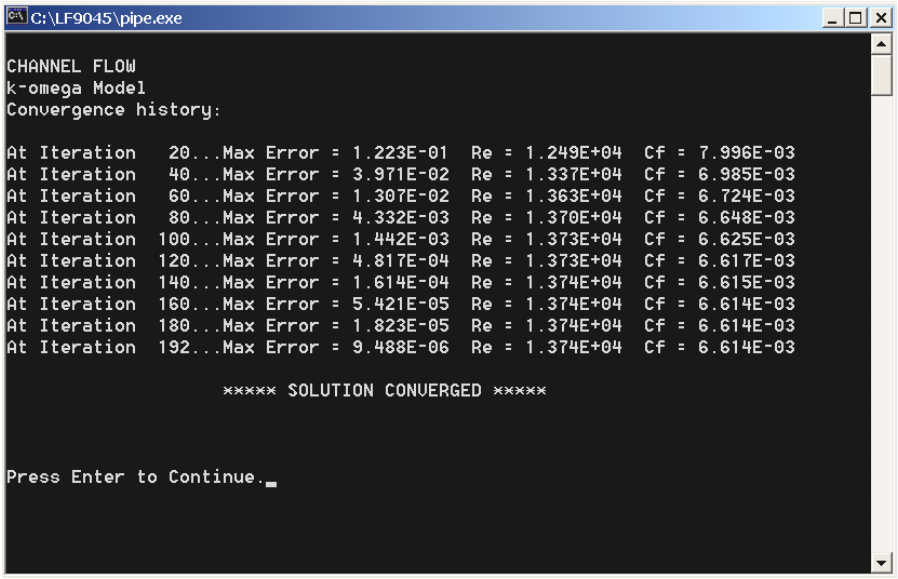

Figure 1.11: *Output from Program* **PIPE**.

View Program Output Button: After returning to the main menu, click on this button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **pipe.prt**.

Plot Computed Profiles Button: Click on this button to create a plot (see Figure 1.12). Program **PIPE** saves plotting data in a disk file named **plotp.dat**. Contents of the file differ depending upon which turbulence model is used. For the Cebeci-Smith and Baldwin-Lomax algebraic models, the Johnson-King half-equation model and the Baldwin-Barth and Spalart-Allmaras one-equation models, only velocity and Reynolds shear stress profiles are plotted. By contrast, for the k-ω and Stress-ω models, the plot also includes the turbulence kinetic energy, k, production of k and dissipation of k. The file is created using the following Fortran statements:

```
      write(iunit2,240) jmax,jaxi,retau
      if(model.ge.3.and.model.le.7) then
        ncols=5
        write(iunit2,250) ncols
        write(iunit2,260) (yoh(j),uoum(j),yplus(j),uplus(j),tau(j),
     *                     j=1,jmax)
      else
        ncols=8
        write(iunit2,250) ncols
        do 60 j=2,jmax
          diss(j)=-betas*bbeta(j)*et(j)*wt(j)
          prod(j)=tau(j)*dudy(j)
 60     continue
        if(nvisc.ne.0) diss(1)=2.*diss(2)-diss(3)
        prod(1)=0.
        write(iunit2,270) (yoh(j),uoum(j),yplus(j),uplus(j),tau(j),
     *                     et(j),diss(j),prod(j),j=1,jmax)
      endif
 240  format(1x,2i4,1pe14.6)
 250  format(1x,i2)
 260  format(1p5e14.6)
 270  format(1p8e14.6)
```

where *jmax* is the number of grid points, *jaxi* is 0 for channel flow and 1 for pipe flow. The quantity *retau* is Reynolds number based on friction velocity and channel half height/pipe radius, i.e., $Re_\tau = u_\tau R/\nu$. Also, ***yoh(j)*** is y/R, ***uoum(j)*** is $U(y)/U(0)$, ***yplus(j)*** is y^+, ***uplus(j)*** is U^+, ***tau(j)*** is τ_{xy}/u_τ^2, ***et(j)*** is k/u_τ^2, ***diss(j)*** is $\beta^*\nu\omega k/u_\tau^4$, and ***prod(j)*** is $\nu\tau_{xy}(dU/dy)/u_\tau^4$.

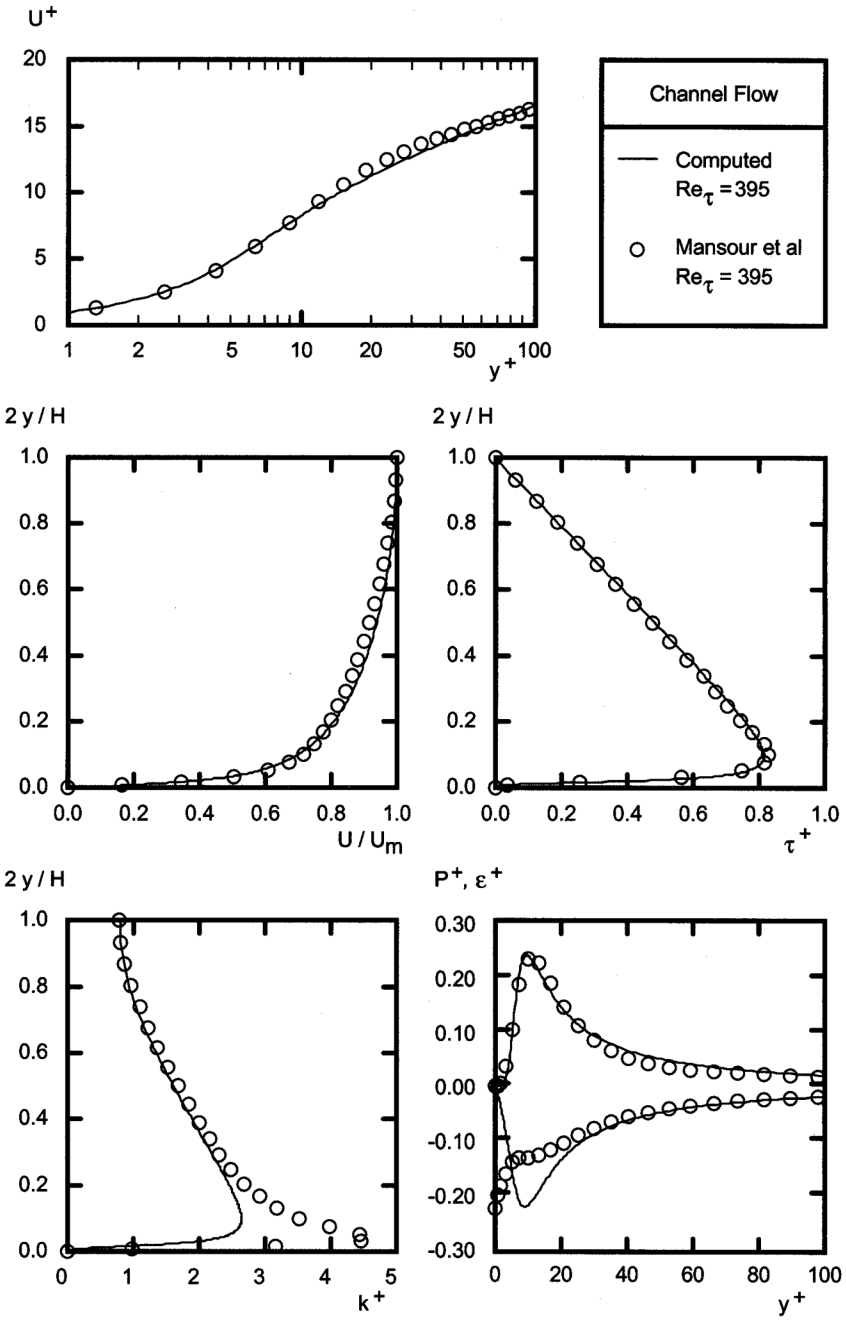

Figure 1.12: *Printed output from Program* **PIPE_PLOT**.

1.6 Program DEFECT: Defect Layer

Program **DEFECT** computes properties of the incompressible defect-layer including effects of pressure gradient. Figure 1.13 shows the opening screen for Program **DEFECT_DATA**, whose operation is straightforward. You can use your mouse or TAB key to select a given input quantity and then enter whatever changes you want. The box at the bottom of the menu provides help when you position your mouse on a given input parameter. Just be sure to press the ENTER key, the TAB key, or click on another input quantity to confirm an input-data change.

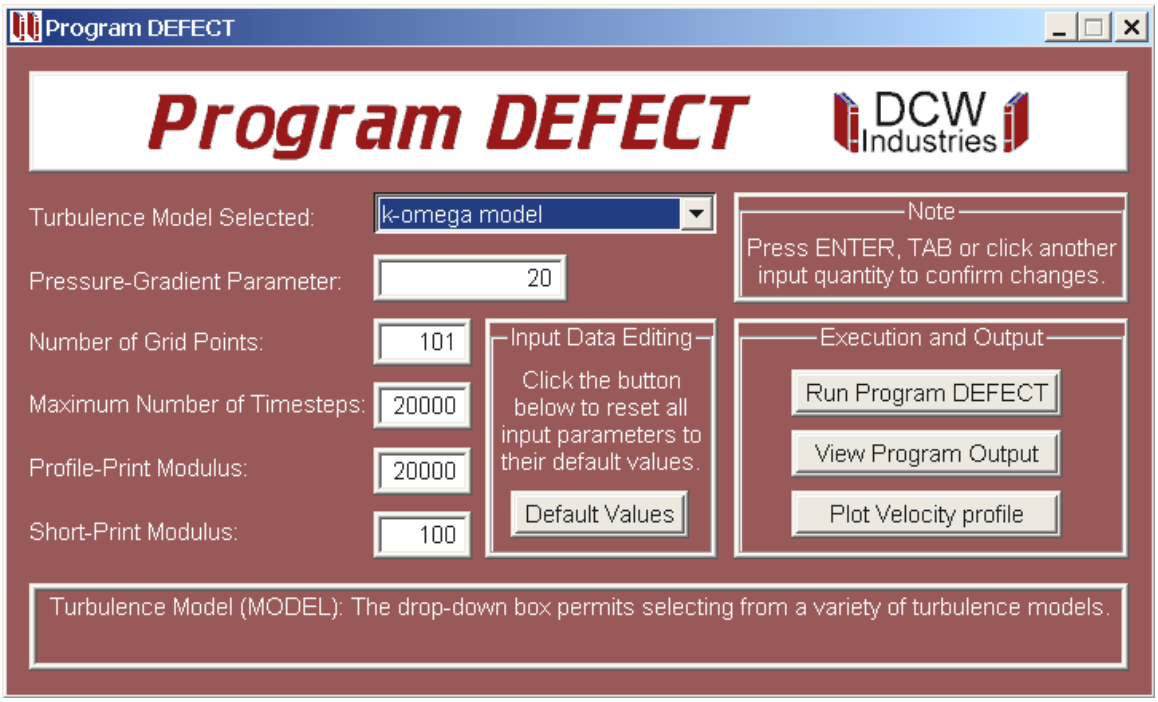

Figure 1.13: *Opening menu of Program* **DEFECT_DATA**.

Default Values Button: Click on this button to reset the default values. The first time you run **DEFECT_DATA**, the menu appears exactly as in the figure. All values shown are the built-in default values. When you close the program, it will save input-data file **defect.dat** with any changes you have made. When you run the program the next time, it will read **defect.dat** and the values of all input parameters will be those that you set in your last run.

Run Program DEFECT Button: Click on this button to run Program **DEFECT**. Control will pass to **DEFECT** and the screen will appear as shown in Figure 1.14. Video output

includes timestep number, maximum error, solution coefficient A [see Equation (4.180) of *Turbulence Modeling for CFD*] and wake-strength parameter, Π. Press the ENTER key or close the window to return to the main menu.

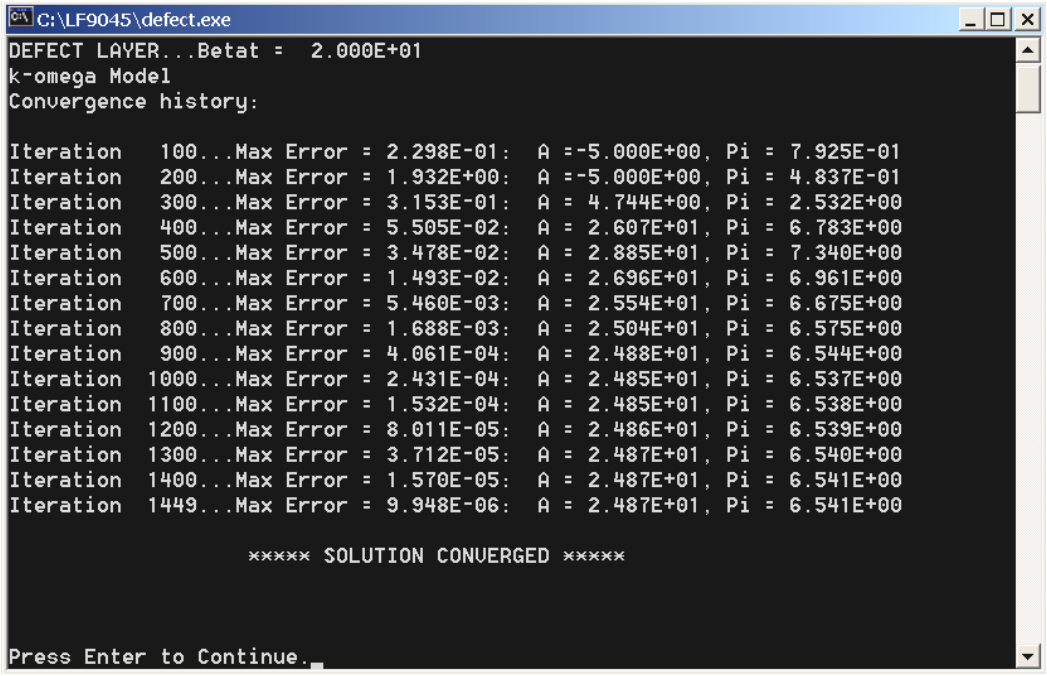

Figure 1.14: *Output from Program* **DEFECT**.

View Program Output Button: After returning to the main menu, click on this button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **defect.prt**.

Plot Velocity Profile Button: Click on this button to create a plot (see Figure 1.15). Program **DEFECT** saves plotting data in a disk file named **plotd.dat**. The file is created using the following statements:

```
      write(iunit2,200) jmax,betat
      write(iunit2,210) (eta(j),u(j),j=1,jmax)
 200  format(1x,i4,1pe14.6)
 210  format(1p2e14.6)
```

where *jmax* is the number of grid points, *betat* is β_T, *eta(j)* is $\eta = y/\Delta$ and *u(j)* is $[U_e - U(\eta)]/u_\tau$.

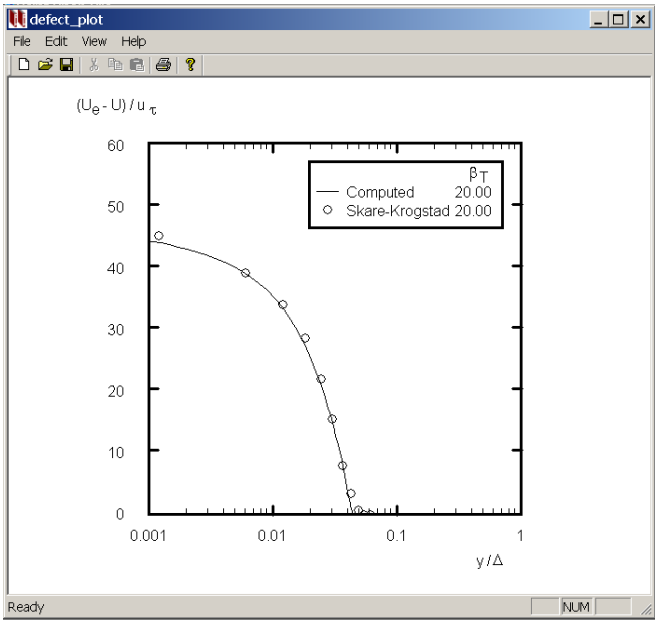

Figure 1.15: *Output from Program* **DEFECT_PLOT**.

1.7 Program SUBLAY: Viscous Sublayer

Program **SUBLAY** computes incompressible viscous sublayer flow, including surface roughness and surface mass transfer. Figure 1.16 shows the opening screen for Program **SUBLAY_DATA**, whose operation is straightforward. You can use your mouse or TAB key to select a given input quantity and then enter whatever changes you want. The box at the bottom of the menu provides help when you position your mouse on a given input parameter. Just be sure to press the ENTER key, the TAB key, or click on another input quantity to confirm an input-data change.

Default Values Button: Click on this button to reset the default values. The first time you run **SUBLAY_DATA**, the menu appears exactly as in the figure. All values shown are the built-in default values. When you close the program, it will save input-data file **sublay.dat** with any changes you have made. When you run the program the next time, it will read **sublay.dat** and the values of all input parameters will be those that you set in your last run.

Run Program SUBLAY Button: Click on this button to run Program **SUBLAY**. Control will pass to **SUBLAY** and the screen will appear as shown in Figure 1.17. Video output includes timestep number, maximum error and law-of-the-wall constant, C. Press the ENTER key or close the window to return to the main menu.

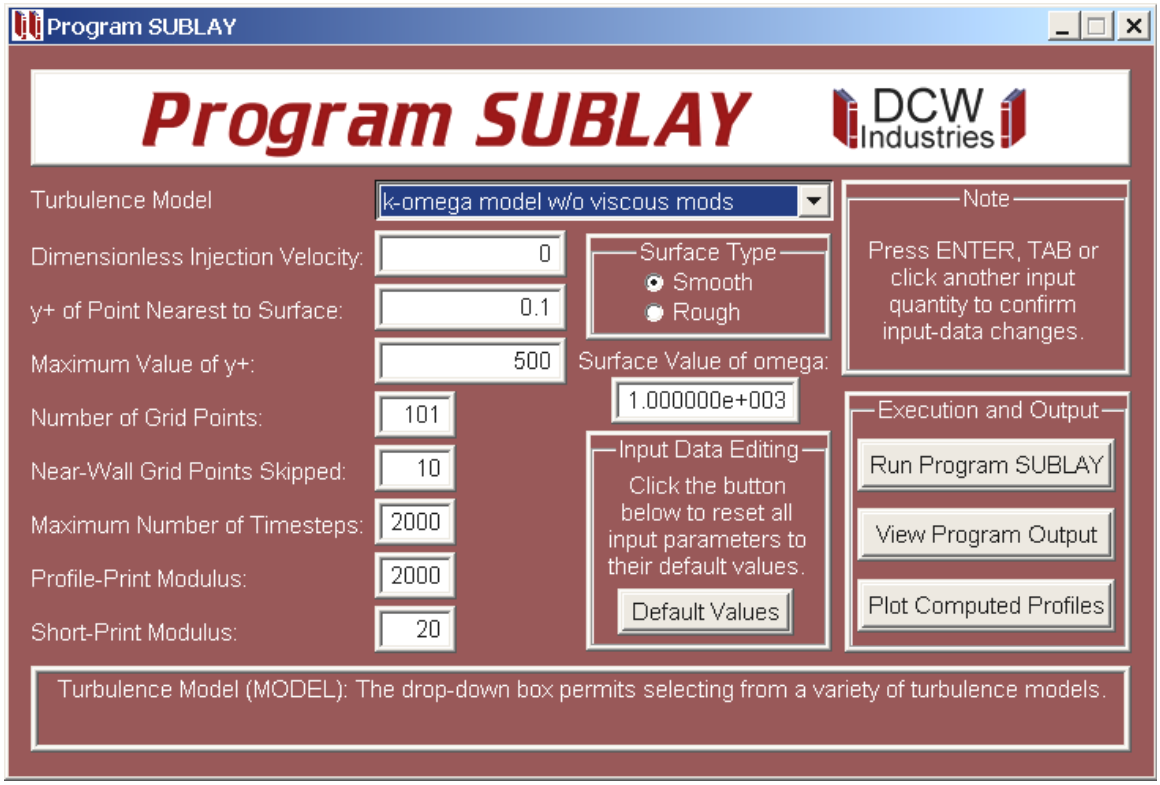

Figure 1.16: *Opening menu of Program* **SUBLAY_DATA**.

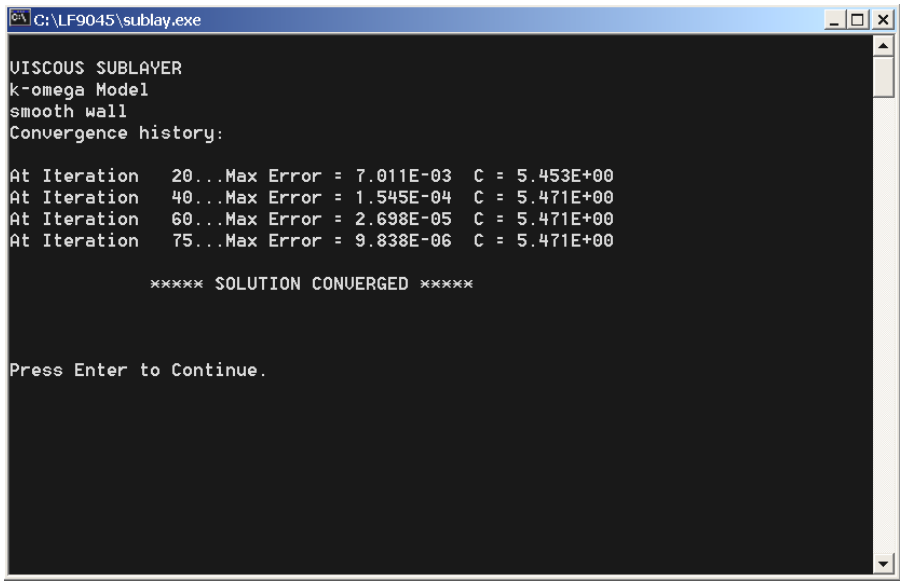

Figure 1.17: *Output from Program* **SUBLAY**.

View Program Output Button: After returning to the main menu, click on this button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **sublay.prt**.

Plot Computed Profiles Button: Click on this button to create a plot (see Figure 1.18). Program **SUBLAY** saves plotting data in a disk file named **plots.dat**. Contents of the file differ depending upon which turbulence model is used. For the Cebeci-Smith and Baldwin-Lomax algebraic models, the Johnson-King half-equation model and the Baldwin-Barth and Spalart-Allmaras one-equation models, only the velocity profile is plotted. By contrast, for the k-ω and Stress-ω models, the plot also includes the turbulence kinetic energy, k, production of k and dissipation of k. The file is created using the following Fortran statements:

```
      write(iunit2,200) jmax,vwplus
      write(iunit2,210) (yplus(j),uplus(j),wtm(j),etm(j),j=1,jmax)
 200  format(1x,i4,1pe14.6)
 210  format(1p4e14.6)
```

where *jmax* is the number of grid points, *vwplus* is v_w/u_τ, *yplus(j)* is y^+, *uplus(j)* is U^+, *wtm(j)* is dissipation, $\beta^*\nu\omega k/u_\tau^4$, and *etm(j)* is production, $\nu\tau_{xy}(dU/dy)/u_\tau^4$.

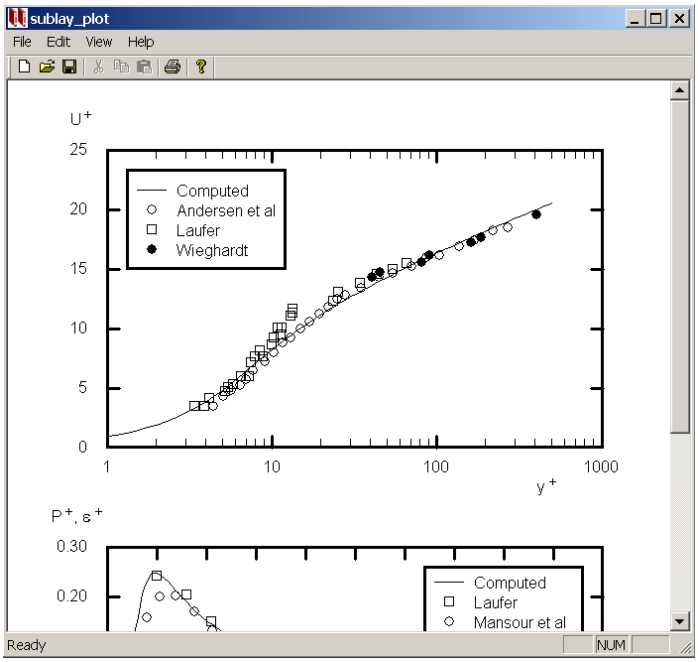

Figure 1.18: *Output from Program* **SUBLAY_PLOT**.

1.8 Technical Details

The companion CD includes Fortran source code for all of the programs described in this Chapter, which can be compiled and run on virtually any computer with a Fortran compiler. The CD also includes executable programs, each of which is augmented with input-data preparation and plotting utilities designed for use on a personal computer with any version of the Microsoft Windows operating system.

1.8.1 Files Included on the CD

There are four files included for each of the six programs described in this chapter. Every Fortran program "includes" the file **cpuid**. Table 1.1 lists the 25 files.

Table 1.1: *Program files on the CD.*

File Name	Directory	Function
defect.exe	Executables	Executable Program **DEFECT**
defect.for	Source Code	Fortran source code for Program **DEFECT**
defect_data.exe	Executables	Input-data preparation Program **DEFECT_DATA**
defect_plot.exe	Executables	Plotting Program **DEFECT_PLOT**
jet.exe	Executables	Executable Program **JET**
jet.for	Source Code	Fortran source code for Program **JET**
jet_data.exe	Executables	Input-data preparation Program **JET_DATA**
jet_plot.exe	Executables	Plotting Program **JET_PLOT**
mixer.exe	Executables	Executable Program **MIXER**
mixer.for	Source Code	Fortran source code for Program **MIXER**
mixer_data.exe	Executables	Input-data preparation Program **MIXER_DATA**
mixer_plot.exe	Executables	Plotting Program **MIXER_PLOT**
pipe.exe	Executables	Executable Program **PIPE**
pipe.for	Source Code	Fortran source code for Program **PIPE**
pipe_data.exe	Executables	Input-data preparation Program **PIPE_DATA**
pipe_plot.exe	Executables	Plotting Program **PIPE_PLOT**
sublay.exe	Executables	Executable Program **SUBLAY**
sublay.for	Source Code	Fortran source code for Program **SUBLAY**
sublay_data.exe	Executables	Input-data preparation Program **SUBLAY_DATA**
sublay_plot.exe	Executables	Plotting Program **SUBLAY_PLOT**
wake.exe	Executables	Executable Program **WAKE**
wake.for	Source Code	Fortran source code for Program **WAKE**
wake_data.exe	Executables	Input-data preparation Program **WAKE_DATA**
wake_plot.exe	Executables	Plotting Program **WAKE_PLOT**
cpuid	Source Code	File included by all six Fortran programs

1.8.2 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 (1947) 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 virtually any computer with a Fortran compiler. 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, Pentium)
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 Appendix A 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 input-data preparation program name is **PROGNAME_DATA** and the plotting program name is **PROGNAME_PLOT**.

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 write program output to a disk file. 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.

1.8.3 Program Input

All of the programs use a standardized method to provide input. **PROGNAME_DATA** creates a disk file named **progname.dat**. The format for integer quantities is (1x,a12,i4) [(1x,a11,i5) for **DEFECT**] 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.

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

1.8.4 Program Output

Program **PROGNAME** can be executed from **PROGNAME_DATA**. Output consists of a disk file that is used to plot computed results and a disk file named **progname.prt**. The plotting program, **PROGNAME_PLOT**, can be run from **PROGNAME_DATA**. The name of the plotting-data disk file is **plotP.dat**, where "**P**" is the first letter in **PROGNAME**.

The programs implement several turbulence models, and all of the programs make provision for a user-defined model so that the supplied input-data preparation and plotting utilities can be used for customized versions of the various programs. Usually, only

a subset of the models listed in Table 1.2 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.

Table 1.2: *Turbulence models implemented.*

Model	Type	*model*	*nvisc*	*kerng*
k-ω (high Re) - Wilcox (2006)	2-Equation	0	0	Not used
k-ω (low Re) - Wilcox (2006)	2-Equation	0	1	Not used
Stress-ω (high Re) - Wilcox (2006)	Stress-Transport	1	0	Not used
Stress-ω (high Re) - Wilcox (2006)	Stress-Transport	1	1	Not used
k-ϵ - Launder-Sharma (1974)	2-Equation	2	Not used	0
RNG k-ϵ - Yakhot et al. (1986)	2-Equation	2	Not used	1
Baldwin-Barth (1990)	1-Equation	3	Not used	Not used
Spalart-Allmaras (1992)	1-Equation	4	Not used	Not used
Cebeci-Smith (1974)	Algebraic	5	Not used	Not used
Baldwin-Lomax (1978)	Algebraic	6	Not used	Not used
Johnson-King (1985)	$\frac{1}{2}$-Equation	7	Not used	Not used
Mixing Length - Prandtl (1925)	Algebraic	9	Not used	Not used
User defined	—	99	Not used	Not used

Printed output, which can be viewed from **PROGNAME_DATA**, 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.

1.8.5 Programs JET, MIXER and WAKE

Programs **JET**, **MIXER** and **WAKE** solve the self-similar form of the turbulent-flow equations for jets, mixing layers and wakes that are asymptotically approached far downstream. Section 4.5.1 of *Turbulence Modeling for CFD* presents the equations of motion in physical variables and in similarity form. An additional transformation devised by Rubel and Melnik (1984) has been used in all three of the free shear flow programs that greatly improves numerical accuracy. Specifically, we introduce a new independent variable, ξ, defined in terms of the similarity variable, η, by

$$d\xi = \frac{d\eta}{N(\eta)} \qquad \text{or} \qquad \frac{d}{d\xi} = N(\eta)\frac{d}{d\eta} \tag{1.1}$$

where $N(\eta)$ is the dimensionless eddy viscosity appearing in the similarity solution. In terms of this variable, the equations for the k-ω and k-ϵ models, for example, are:

Mean Momentum:

$$\mathcal{V}\frac{d\mathcal{U}}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\eta^j f_N \frac{d\mathcal{U}}{d\xi}\right] = S_u N\mathcal{U} \tag{1.2}$$

Turbulence Kinetic Energy:

$$\mathcal{V}\frac{dK}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\sigma^*\eta^j\frac{dK}{d\xi}\right] = S_k NK + f_N\left(\frac{d\mathcal{U}}{d\xi}\right)^2 - \beta^* K^2 \tag{1.3}$$

***k*-ω Model:**

$$\mathcal{V}\frac{dW}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\sigma\eta^j\frac{dW}{d\xi}\right] = S_w NW + \alpha f_N\frac{W}{K}\left(\frac{d\mathcal{U}}{d\xi}\right)^2 + \frac{\sigma_d}{K}\frac{dK}{d\xi}\frac{dW}{d\xi} - \beta KW \tag{1.4}$$

$$N = \frac{K}{W}, \qquad f_N = \min\left\{1, C_{lim}^{-1}\frac{\sqrt{\beta^*}K}{|d\mathcal{U}/d\xi|}\right\} \tag{1.5}$$

***k*-ϵ Model:**

$$\mathcal{V}\frac{dE}{d\xi} - \frac{1}{\eta^j}\frac{d}{d\xi}\left[\frac{\eta^j}{\sigma_\epsilon}\frac{dE}{d\xi}\right] = S_e NE + C_{\epsilon 1}\frac{E}{K}\left(\frac{d\mathcal{U}}{d\xi}\right)^2 - C_\mu C_{\epsilon 2} KE \tag{1.6}$$

$$N = C_\mu\frac{K^2}{E}, \qquad f_N = 1 \tag{1.7}$$

The quantities $\mathcal{U}$, $\mathcal{V}$, K, W, E and f_N are the transformed velocity components, turbulence kinetic energy, specific dissipation rate, dissipation rate, and stress-limiter function respectively. Note that the stress limiter is implemented only for the k-ω model. See Section 4.5.1 of *Turbulence Modeling for CFD* for additional details on notation and other features of the similarity solution.

Also, because its implementation is a bit unusual with the Rubel-Melnik transformation, it is worthwhile to quote the form that the mixing-length model assumes.

Mixing-Length Model:

$$N = \alpha\sqrt{\left|\frac{d\mathcal{U}}{d\xi}\right|}, \qquad f_N = 1 \tag{1.8}$$

This transformation greatly improves numerical accuracy primarily because it removes numerical difficulties that are 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 $\xi \rightarrow \infty$, 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 (1.2), (1.3), (1.4) and (1.6). We then make an initial guess and let 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.

$$\frac{\partial \mathcal{U}}{\partial t} + \mathcal{V}\frac{\partial \mathcal{U}}{\partial \xi} - \frac{1}{\eta^j}\frac{\partial}{\partial \xi}\left[\eta^j f_N \frac{\partial \mathcal{U}}{\partial \xi}\right] = S_u N \mathcal{U} \tag{1.9}$$

The resulting time-dependent system of equations is solved using implicit Crank-Nicolson (1947) differencing that is second-order accurate in both t and ξ. Using 101 mesh points, all of the free-shear-flow programs typically require computing times of just a few seconds on a 3-GHz Pentium-D microcomputer.

1.8.6 Program PIPE

Subsection 3.5.1 of *Turbulence Modeling for CFD* describes 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 various 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.

$$\frac{dU^+}{dy^+} = \frac{1 - y^+/R^+}{1 + \mu_T^+} \tag{1.10}$$

All notation in Equation (1.10) is identical to that used in Subsection 3.5.1 of *Turbulence Modeling for CFD*.

The only other subtle feature of the program is the way the specific dissipation rate, ω, in the k-ω model and the Stress-ω 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 of *Turbulence Modeling for CFD*). That is, we use the fact that, for $y^+ < 2.5$, $\omega^+ = \nu\omega/u_\tau^2$ is given by:

$$\omega^+ \rightarrow \frac{6}{\beta_o (y^+)^2} \qquad \text{as} \quad y \rightarrow 0 \qquad \text{(smooth wall)} \tag{1.11}$$

$$\omega^+ \rightarrow \frac{\omega_w^+}{\left[1 + \sqrt{\frac{\beta_o \omega_w^+}{6}}\; y^+\right]^2} \qquad \text{as} \quad y \rightarrow 0 \qquad \text{(rough wall)} \tag{1.12}$$

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** typically requires computing times of less than 5 seconds on a 3-GHz Pentium-D microcomputer.

1.8.7 Programs DEFECT and SUBLAY

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

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 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 $\xi \rightarrow \infty$. The nonvanishing boundary conditions quoted for $W_0(\xi)$ and $E_0(\xi)$ as $\xi \rightarrow \infty$ are the only choices consistent with the similarity solution.

All Models:

$$\eta = \int_{-\infty}^{\xi} N_0(\xi')\, d\xi' \tag{1.13}$$

$$\frac{d}{d\xi}\left(f_N \frac{dU_1}{d\xi}\right) + (1 + 2\beta_T)\eta \frac{dU_1}{d\xi} + 2\beta_T N_0 U_1 = 0 \tag{1.14}$$

$$\sigma^* \frac{d^2 K_0}{d\xi^2} + 2\beta_T K_0 + (1 + 2\beta_T)\eta \frac{dK_0}{d\xi} + \sqrt{\beta^*} \left[f_N \left(\frac{dU_1}{d\xi} \right)^2 - N_0 E_0 \right] = 0 \tag{1.15}$$

k-ω Model:

$$\sigma \frac{d^2 W_0}{d\xi^2} + (1 + 2\beta_T)\eta \frac{dW_0}{d\xi} + (1 + 4\beta_T) N_0 W_0 + \frac{\sigma_d}{K_0} \frac{dK_0}{d\xi} \frac{dW_0}{d\xi} + \frac{\sqrt{\beta^*}}{K_0} \left[\alpha f_N \left(\frac{dU_1}{d\xi} \right)^2 - \frac{\beta_o}{\beta^*} K_0^2 \right] W_0 = 0 \tag{1.16}$$

$$W_0(\xi) \rightarrow \frac{(1 + 4\beta_T)\sqrt{\beta^*}}{\beta_o} \qquad \text{as} \qquad \xi \rightarrow \infty \tag{1.17}$$

$$N_0 = K_0 / W_0, \qquad N_0 E_0 = K_0^2, \qquad f_N = \min \left\{ 1, C_{lim}^{-1} \frac{K_0}{|d\mathcal{U}_\infty / d\xi|} \right\} \tag{1.18}$$

k-ϵ Model:

$$\sigma_\epsilon^{-1} \frac{d^2 E_0}{d\xi^2} + (1 + 2\beta_T)\eta \frac{dE_0}{d\xi} + (1 + 6\beta_T) N_0 E_0 + \frac{\sqrt{C_\mu}}{K_0} \left[C_{\epsilon 1} \left(\frac{dU_1}{d\xi} \right)^2 - C_{\epsilon 2} K_0^2 \right] E_0 = 0 \tag{1.19}$$

$$E_0(\xi) / K_0(\xi) \rightarrow \frac{(1 + 2\beta_T)}{C_{\epsilon 2}\sqrt{C_\mu}} \qquad \text{as} \qquad \xi \rightarrow \infty \tag{1.20}$$

$$N_0 = K_0^2 / E_0, \qquad f_N = 1 \tag{1.21}$$

In Program **SUBLAY**, just as with Program **PIPE**, 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.

$$\frac{dU^+}{dy^+} = \frac{1}{1 + \mu_T^+} \tag{1.22}$$

All notation in Equation (1.22) is identical to that used in Section 4.6 of *Turbulence Modeling for CFD*. As with Program **PIPE**, Program **SUBLAY** uses Equations (1.11) and (1.12) for a prescribed number of mesh points, *jskip*, next to the surface.

1.8.8 Fortran Compiler Options

All of the Fortran programs have been compiled with the Lahey Fortran-90 compiler. The programs achieve optimum performance in double-precision accuracy. Also, for compatibility with the Visual C++ interface, they are compiled with the Lahey "Windows Console" option. The specific command used in compiling and linking the programs is as follows.

```
lf90 progname -dbl -bind -winconsole
```

Chapter 2

Program EDDYBL

2.1 Overview

The information in this chapter is the user's guide for Program **EDDYBL**, which is a two-dimensional and axisymmetric, compressible boundary-layer program for laminar, transitional and turbulent boundary layers that is included on the companion CD for *Turbulence Modeling for CFD*. It includes an overview of the program's operation along with technical details on the program's structure and operation. The software on the CD includes a program, **EDDYBL_DATA**, that accomplishes the following.

- Guides input-data preparation
- Executes Program **EDDYBL**
- Permits viewing and/or printing program output
- Creates a video and/or hardcopy plot that compares computed results with experimental data

Because Program **EDDYBL_DATA** is written in Microsoft Visual C++, it should run on personal computers with all versions of the Microsoft Windows operating system.

Program **EDDYBL** embodies a wide variety of turbulence models ranging from mixing-length oriented algebraic models to a complete stress-transport model. This program has evolved over the past three decades and can thus be termed a mature software package. Many U. S. Government Agencies have contributed to development of the program that is based on a computer code originally developed by Price and Harris (1972).

Additionally, important improvements have been made to this software package as a result of feedback from users, most notably from the outstanding fluid mechanics students at USC and UCLA. The author owes special thanks to Dr. G. Brereton of the University

of Michigan whose personal efforts resulted in the addition of the option to use either USCS or SI units.

2.2 Getting Started Quickly

In order to acquaint yourself with Program **EDDYBL** and its input-data preparation utility, **EDDYBL_DATA**, and to confirm that the software is fully operational on your computer, perform the following steps.

1. Copy the following files from the CD to your working directory:

File Name	Directory	Function
eddybl.exe	Executables	Executable Program **EDDYBL**
eddybl_data.exe	Executables	Input-data preparation Program **EDDYBL_DATA**
eddybl_plot.exe	Executables	Plotting Program **EDDYBL_PLOT**
eddybl_start.exe	Executables	Initial-profile Program **EDDYBL_START**
Flow5300.zip	Input Data/Eddybl	Input data for Flow 5300

2. Extract the input-data file **eddybl.dat** from the "Zip" file **Flow5300.zip**. These data correspond to Stratford's (1959) incipient-separation flow. You can use this data file without modification to quickly determine that everything is operating properly, and to see how easy it is to use Program **EDDYBL**. Because the input-data preparation utility, **EDDYBL_DATA**, is menu driven, you will find that very little explanation of the program's operation is needed. After successfully completing this benchmark run, you can learn some of the more subtle features of Program **EDDYBL_DATA** by reading Section 2.3.

3. The first step is to run **EDDYBL_DATA**. Figure 2.1 shows the opening screen.

4. Click on the button with the label "Write Input-Data Files." This runs Program **EDDYBL_START**, which generates initial profiles that will be used by Program **EDDYBL**. The screen will appear as shown in Figure 2.2. After you press *ENTER*, control returns to the main menu. At this point you have prepared all input-data files for the benchmark run. Press the ENTER key or close the window to return to the main menu.

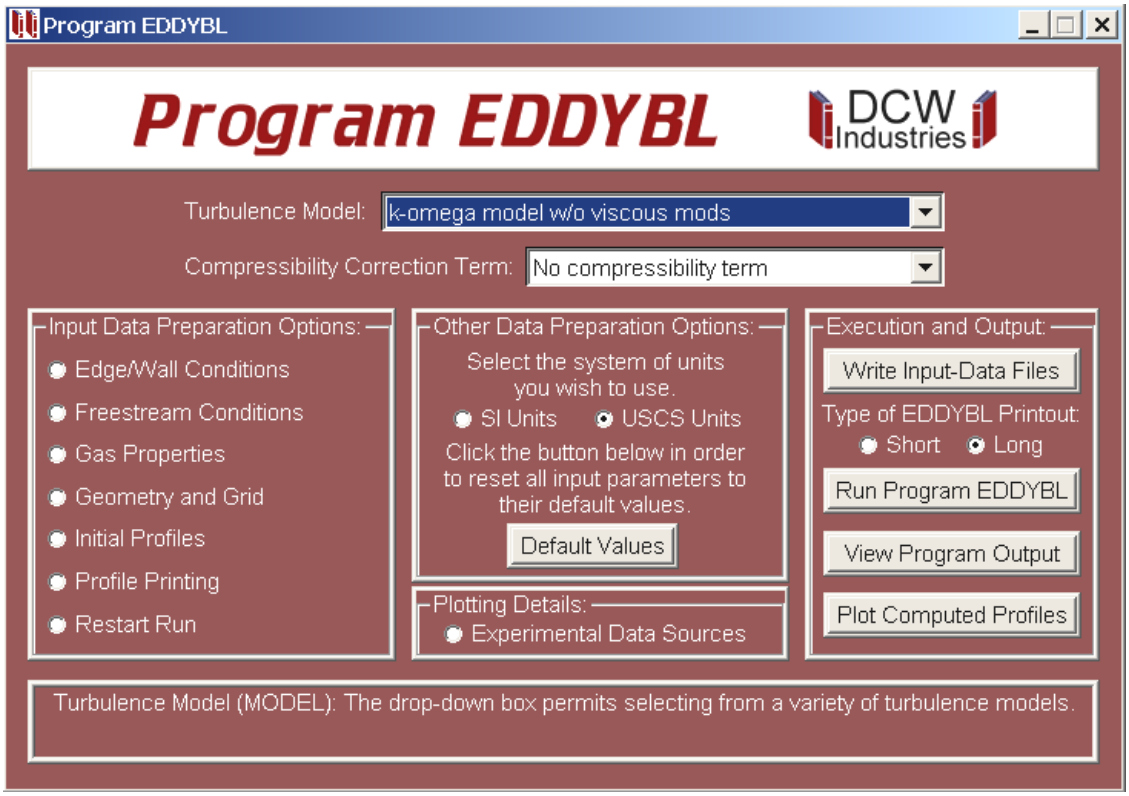

Figure 2.1: *Opening menu of Program* **EDDYBL_DATA**.

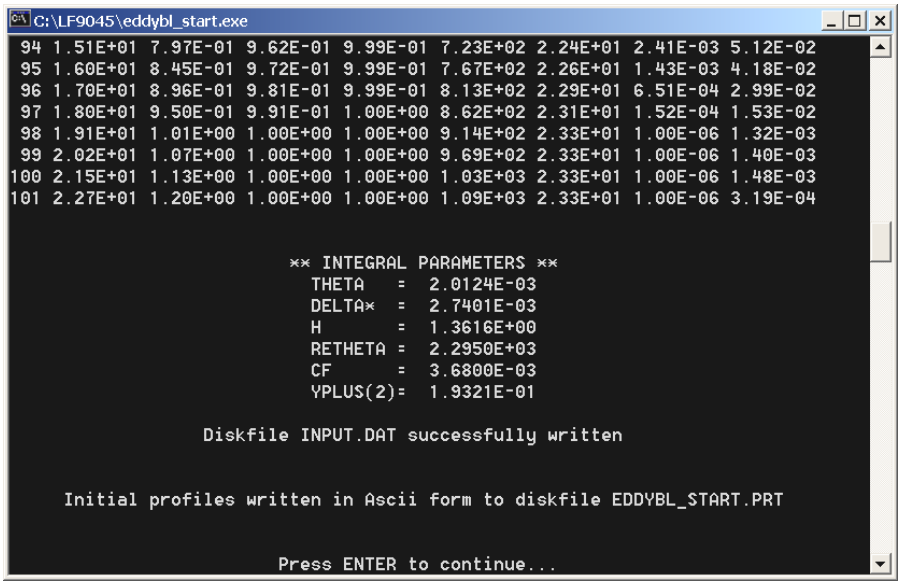

Figure 2.2: *Output from Program* **EDDYBL_START**.

5. All that remains now is to run Program **EDDYBL**. Click on the button with the label "Run Program EDDYBL." This runs Program **EDDYBL**, and the screen will appear as shown in Figure 2.3. Video output includes streamwise step number, m, streamwise distance, s, Reynolds number based on streamwise distance, Re_s, Reynolds number based on momentum thickness, Re_θ, skin friction, c_f, and shape factor, H. Press the Enter key or close the window to return to the main menu.

6. After returning to the main menu, click on the "View Program Output" button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **eddybl.prt**.

7. Click on the "Plot Computed Profiles" button to create a plot (see Figure 2.4). You can generate a hardcopy of the figure by clicking on the printer icon at the top of the screen.

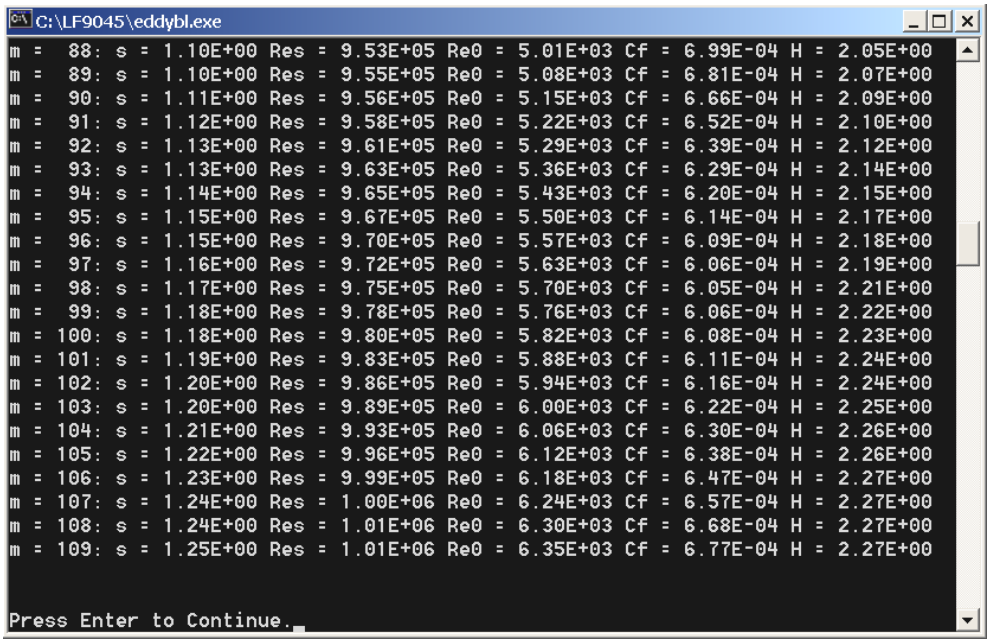

Figure 2.3: *Output from Program* **EDDYBL**.

And that's all there is to running Program **EDDYBL** for any of the input-data files contained in Subdirectory **Input Data/Eddybl** on the companion CD. To run a general case, you will need to use the various menus attached to the main menu. The next section discusses what is contained in each of the eight menus. If you have input-data files created for earlier versions of Program **EDDYBL**, you can use the utility described in Subsection 2.3.9 to convert them to the format recognized by this version of **EDDYBL**.

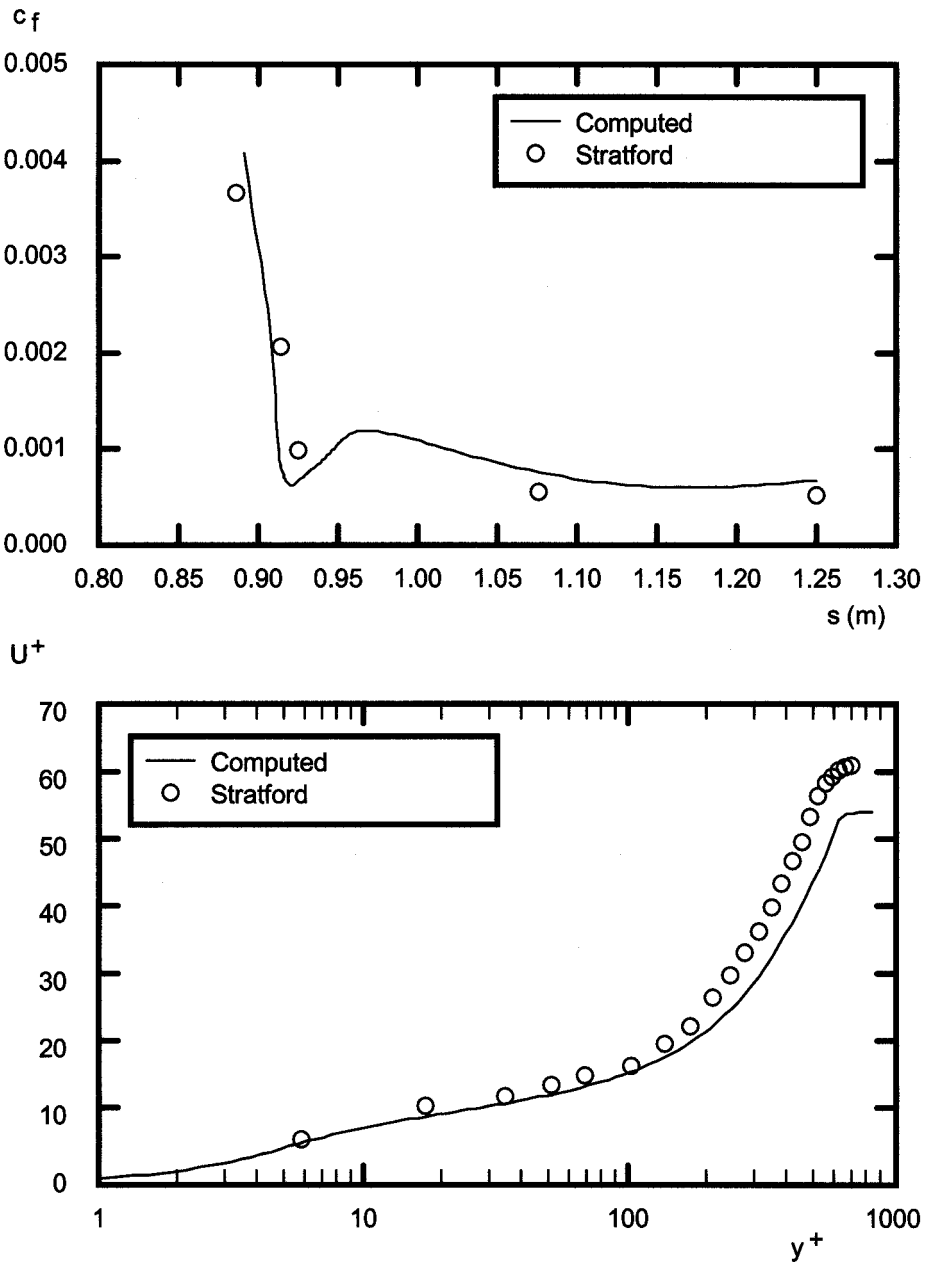

Figure 2.4: *Printed output from Program* **EDDYBL_PLOT**.

2.3 Navigating the Menus

This section explores, in detail, all salient features of the input-data preparation utility, **EDDYBL_DATA**. You will be guided through the various menus and, in the process, you will set up a constant-pressure boundary-layer computation for a Mach 1 freestream with a slightly-cooled surface. For the case you will do, freestream conditions are:

Total pressure, p_{t_∞}	=	23112 $\mathrm{N/m^2}$
Total temperature, T_{t_∞}	=	260 K
Mach number, M_∞	=	1

The surface temperature is 95% of the adiabatic-wall temperature.

Your goal is to initiate the computation at a plate-length Reynolds number, Re_s, of one million and determine the point where the momentum-thickness Reynolds number, Re_θ, is 8000. You might want to do this, for example, in order to provide upstream profiles for a Navier-Stokes computation. You know from a correlation of experimental data that when $Re_s = 1.0 \cdot 10^6$, the boundary layer has the following integral properties:

Skin friction, c_f	=	0.0038
Shape factor, H	=	1.80
Boundary-layer thickness, δ	=	$11.9\,\theta$
Reynolds number, Re_θ	=	1500

Finally, the surface is perfectly smooth, there is no surface mass transfer, and you will use the Baldwin-Lomax (1978) algebraic model.

2.3.1 Opening Menu

To perform this exercise, delete any existing **eddybl.dat** data file that might be in your directory. Although this is not generally necessary, for the purposes of this section it will be easier if you begin with no pre-existing input-data file. Now, run Program **EDDYBL_DATA**. The opening screen will be exactly as in Figure 2.1. For the run you are setting up, you must make the following three changes.

1. Select the Baldwin-Lomax algebraic model
2. Select SI units
3. Select a short printout

You can change turbulence models by clicking on the arrow of the "Turbulence Model" drop-down box at the top of the menu. Select the Baldwin-Lomax model. Select SI units and "short print" by clicking on the appropriate radio buttons. When you have made these changes, the opening menu will appear as in Figure 2.5.

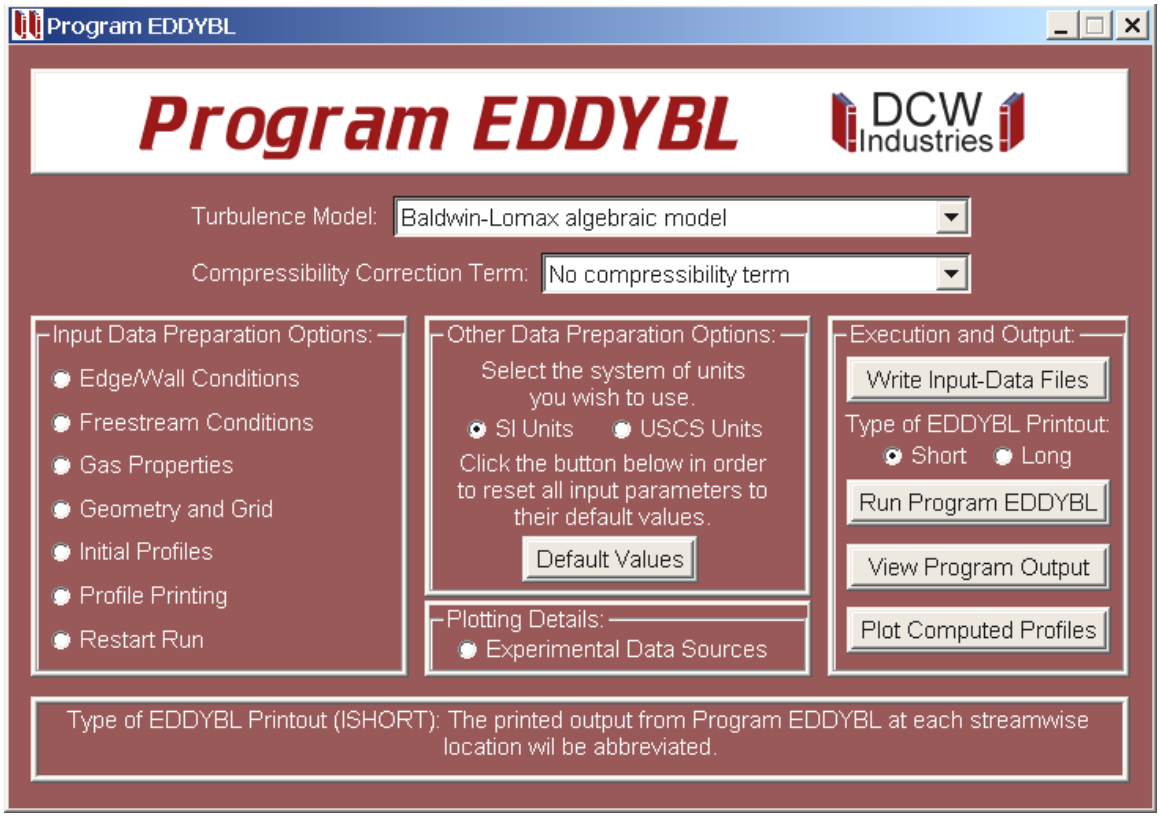

Figure 2.5: *Opening menu of Program* **EDDYBL_DATA** *after modification.*

2.3.2 Freestream Conditions

Next, click on the "Freestream Conditions" radio button, which will bring you to the menu that accepts input for freestream conditions. Click on the number in the "Total Pressure" edit box and change it to $p_{t_\infty} = 23112$ N/m^2 using standard Windows editing techniques. Then, change the total temperature and Mach number to $T_{t_\infty} = 260$ K and $M_\infty = 1$, respectively. Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm the last of your input-data changes. Figure 2.6 shows how the menu will look when you have finished making your changes.

Note that when you click on an input quantity, the box at the bottom of the menu describes the quantity you have clicked. Also, if you watch the right side of the screen, you will observe that each time you change freestream total pressure, total temperature or Mach number, the freestream static conditions change. In the next subsection, you will make use of the freestream static pressure, $p_\infty = 1.220965 \cdot 10^4$ N/m^2, and the freestream unit Reynolds number, $Re_\infty = 4.073846 \cdot 10^6$ m^{-1}. Return to the main menu by closing this window.

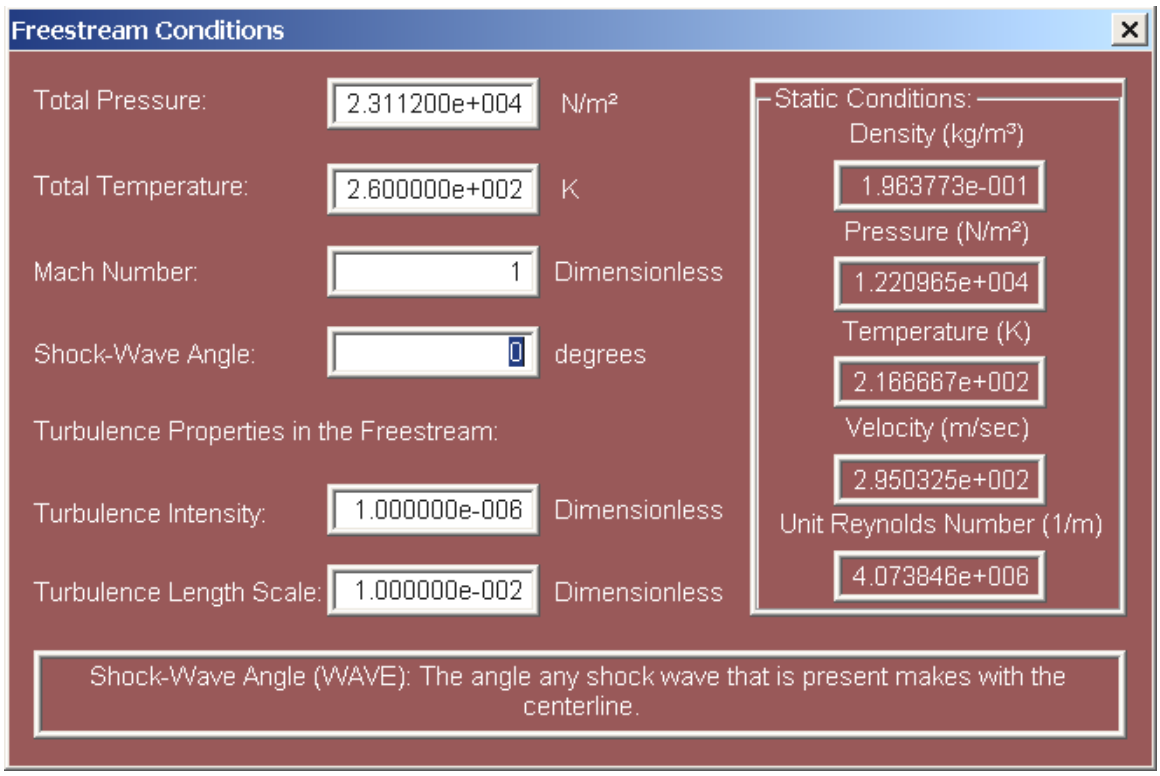

Figure 2.6: *Freestream-Conditions menu of Program* **EDDYBL_DATA** *after modification.*

2.3.3 Geometry and Grid

Click on the "Geometry and Grid" radio button. Here, you will set geometric-progression ratio, k_g, initial stepsize, Δs, initial arclength, s_i, and maximum arclength, s_{stop}.

The first grid point from the surface should be such that $y^+ < 1$. Setting $k_g = 1.09$ is needed to satisfy this constraint. The initial stepsize can be as large as triple the boundary-layer thickness. The initial value of Re_θ is 1500 and the unit Reynolds number is $Re_\infty = 4.07 \cdot 10^6$ m^{-1}, wherefore $\theta = 3.68 \cdot 10^{-4}$ m. Thus, $\delta = 11.9\,\theta = 0.044$ m, which tells you that you can specify an initial stepsize of $\Delta s = 0.013$ m. Since the plate-length Reynolds number at the initial station is $Re_s = 10^6$, the initial arclength is $s_i = 0.245$ m. Finally, correlations indicate that Re_θ will reach the desired value of 8000 when the plate-length Reynolds number is somewhere between three and five million. This corresponds to a maximum distance along the surface of $s = 1.23$ m.

Change the values of k_g, Δs, s_i and s_{stop} to 1.09, 0.013, 0.245 and 1.23, respectively. Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. Figure 2.7 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

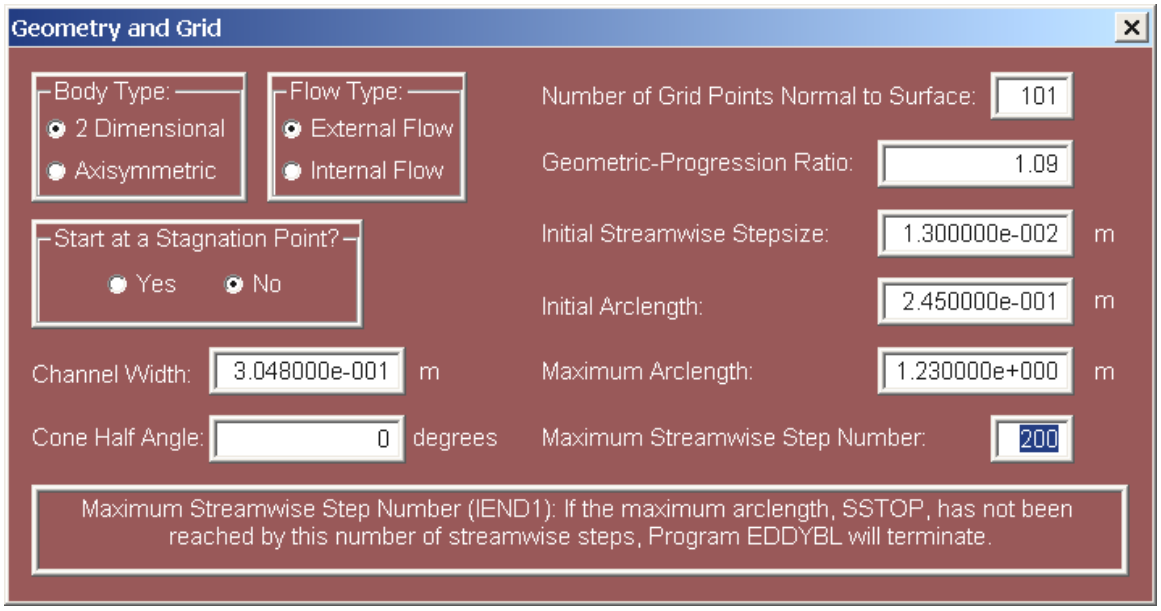

Figure 2.7: *Geometry and Grid menu of Program* **EDDYBL_DATA** *after modification.*

2.3.4 Edge/Wall Conditions

Click on the "Edge/Wall Conditions" radio button. For any boundary-layer computation, at a minimum, you must specify freestream pressure and either surface temperature or surface heat-transfer rate. That is the purpose of this menu.

Click on the "Temperature" radio button. Just below this button, you will note that the adiabatic-wall temperature for this flow is $T_{aw} = 255.23$ K. Thus, the wall temperature, T_w, for this boundary layer is

$$T_w = 0.95 \cdot 255.23 \text{ K} = 242.4685 \text{ K} \tag{2.1}$$

In the bottom section of the menu labeled "Edge/Wall Property Arrays," change the values of boundary-layer edge pressure and surface temperature to $p = 1.220965 \cdot 10^4$ N/m^2 and $T_w = 242.4685$ K, respectively, on "Line No." 1 and 2. As in all **EDDYBL_DATA** menus, you can use standard Windows editing techniques. There is no need to change the default values of the arclength because, in the previous subsection, you estimated that Re_θ will reach the desired value of 8000 when the arclength is $s \approx 1.23$ m. So, the default values of $s_{min} = 0$ m and $s_{max} = 6.096$ m are sufficient for this computation, which extends from $s_i = 0.245$ m to $s_{stop} = 1.23$ m.

Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. Figure 2.8 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

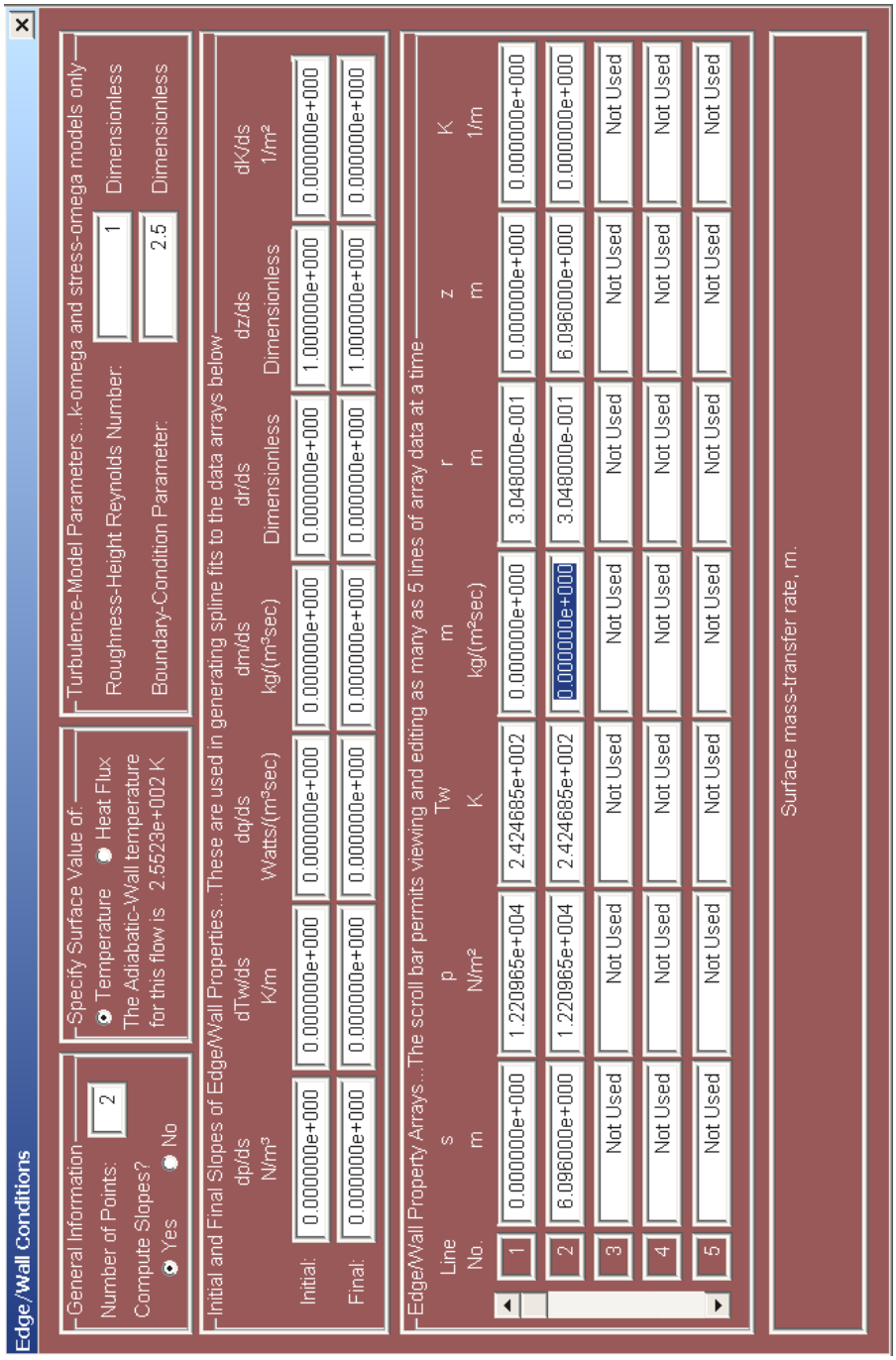

Figure 2.8: *Edge/Wall-Conditions menu of Program* **EDDYBL_DATA** *after modification.*

2.3.5 Initial Profiles

Click on the "Initial Profiles" radio button. For any boundary-layer computation, you must specify flow properties at the initial streamwise station. Program **EDDYBL** provides the option to start either from laminar conditions or from an initially turbulent boundary layer. In this exercise, you have sufficient information to start from turbulent-flow conditions.

Change the values of skin friction, Reynolds number based on momentum thickness, shape factor and initial boundary-layer thickness to $c_f = 0.0038$, $Re_\theta = 1500$, $H = 1.8$ and $\delta = 0.044$ m, respectively. Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. Figure 2.9 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

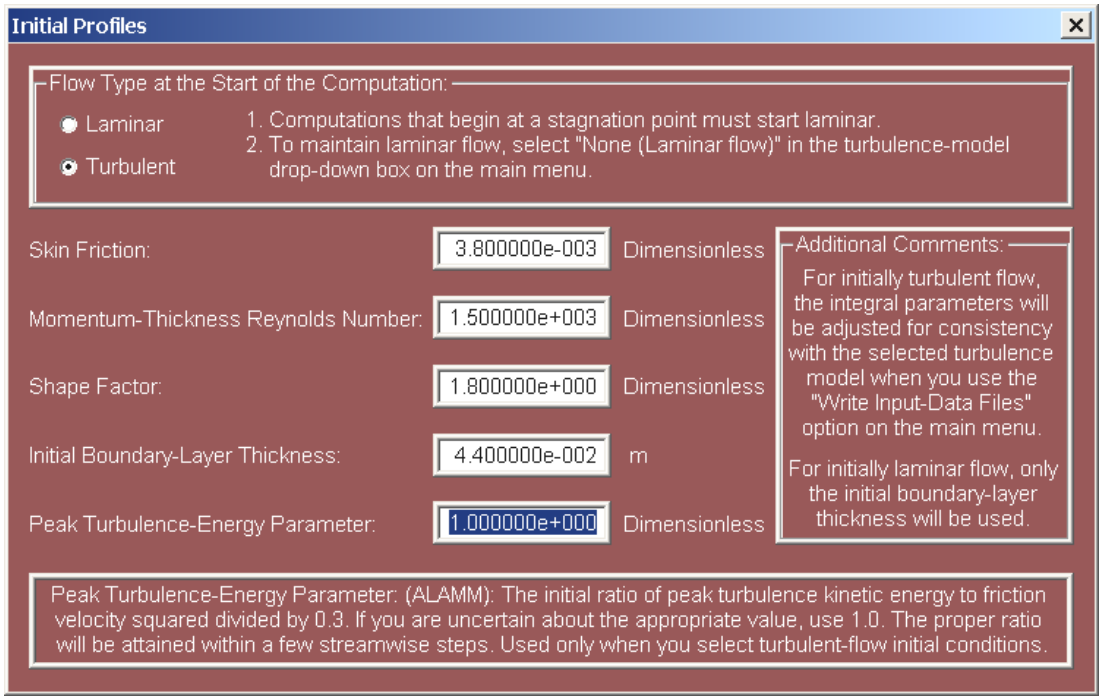

Figure 2.9: *Initial-Profiles menu of Program* **EDDYBL_DATA** *after modification.*

2.3.6 Experimental Data Sources

The final step you will take to set up this computation is to use the "Experimental Data Sources" menu. Click on its radio button, which lies in the bottom central part of the main menu. This menu permits including skin-friction and velocity-profile data that will

be used in creating plots after the computation is done. This, of course, allows you to compare computed results with measurements. For the case at hand, we have no skin-friction data. However, we have a velocity profile that corresponds to $Re_\theta = 8000$. Table 2.1 includes the data you will enter.

Table 2.1: *Experimental velocity-profile data.*

y^+	U^+	y^+	U^+	y^+	U^+	y^+	U^+
4.5	4.44	50.2	14.46	493	21.02	5267	26.14
10.6	8.61	103.6	16.60	1059	23.29	10390	26.18
20.8	11.47	208.7	18.58	2081	25.16	20600	26.18

On the right-hand side of the screen, enter the word "Measured" in the edit box labeled "Data Source." Next, enter the number 12 in the "Number of Points" edit box. Finally, enter the data from Table 2.1. Note that the slider bar in the center of the menu permits scrolling up and down to make the 12 lines you need to fill in accessible. Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. When you have finished entering all of the data points, your screen will appear as in Figure 2.10. Return to the main menu by closing this window.

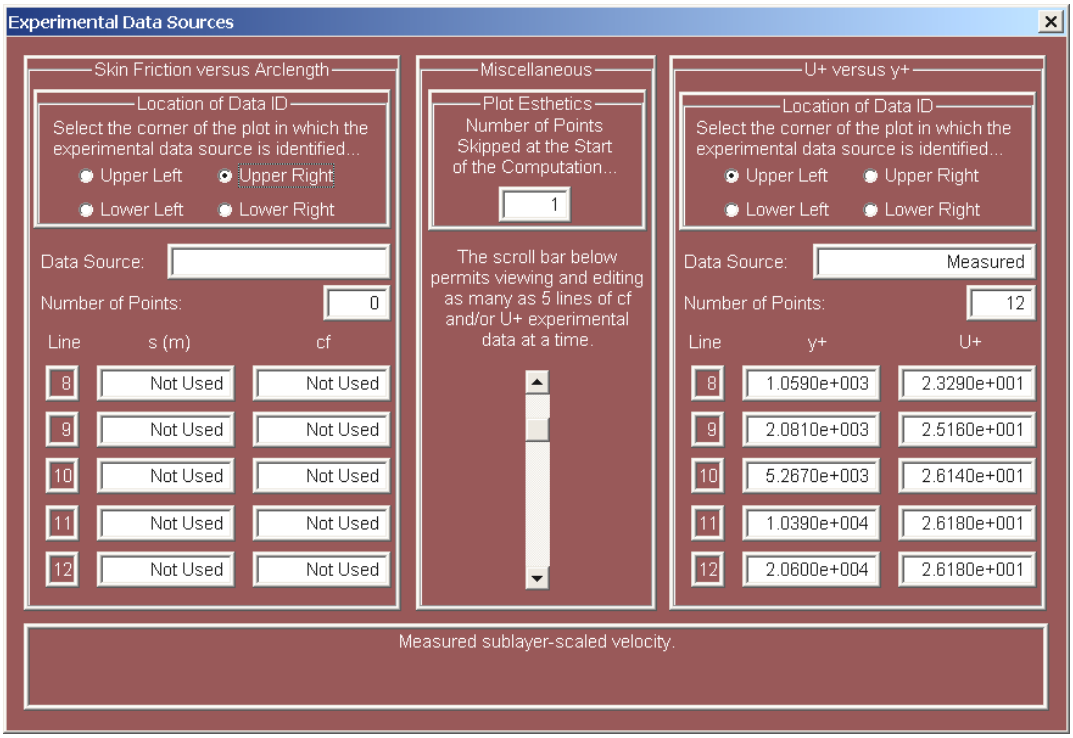

Figure 2.10: *Experimental-Data menu of Program* **EDDYBL_DATA** *after modification.*

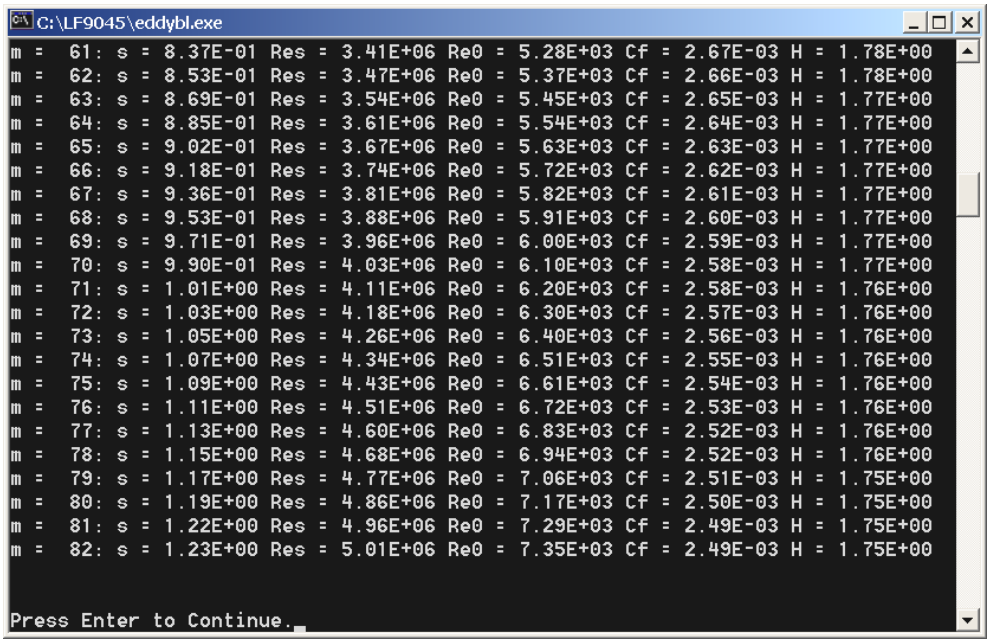

Figure 2.11: *Output from Program* **EDDYBL**.

At this point, you have made all of the input-data changes needed to perform the computation. As with the benchmark case of Section 2.2, you must first click the "Write Input Data Files" button. This saves all input-data changes you have made and then executes Program **EDDYBL_START** in order to generate the initial conditions that Program **EDDYBL** needs. Finally, click the "Run Program EDDYBL" button. If you have made all of the input-data entries correctly, the screen should appear as in Figure 2.11.

2.3.7 Restart Run

Examination of program output reveals that your run didn't go far enough to determine the point where momentum-thickness Reynolds number reaches 8000. After 82 steps, the program stops at s = 1.23 m, and Re_θ is only 7350. Linear extrapolation of your computed Re_θ indicates that you needed to integrate to about s = 1.38 m.

You could go back to the main menu, increase s_{stop} to 1.38 and simply rerun **EDDYBL**. That, in fact, would be the simplest approach for this application. However, there is an alternative that Program **EDDYBL_START** provides through its **Restart Run** option. This permits continuing the computation from where it stopped to a distance farther along the surface.

The restart option is especially helpful for applications where initial turbulent-flow conditions are unknown, which means you must start from laminar conditions and integrate

through transition. While this is very simple for the k-ω and Stress-ω models, it is very difficult with most of the other models implemented in **EDDYBL**, especially the k-ϵ models. That is, you can also change turbulence models when you initiate a restart run. Thus, you can run a k-ω model computation through transition to establish turbulent-flow initial conditions for any of the other models.

Close the output window to return to the main menu and click on the "Restart Run" radio button. First, change the "Maximum Arclength" to 1.38 m. Then click on the "Rename OUTPUT.DAT" button. Program **EDDYBL_DATA** then copies **input.dat** (which is the file created by **EDDYBL_START**) to **input.bak** and renames **output.dat** as **input.dat**. The final output of your original run becomes input for the restart run. Figure 2.12 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

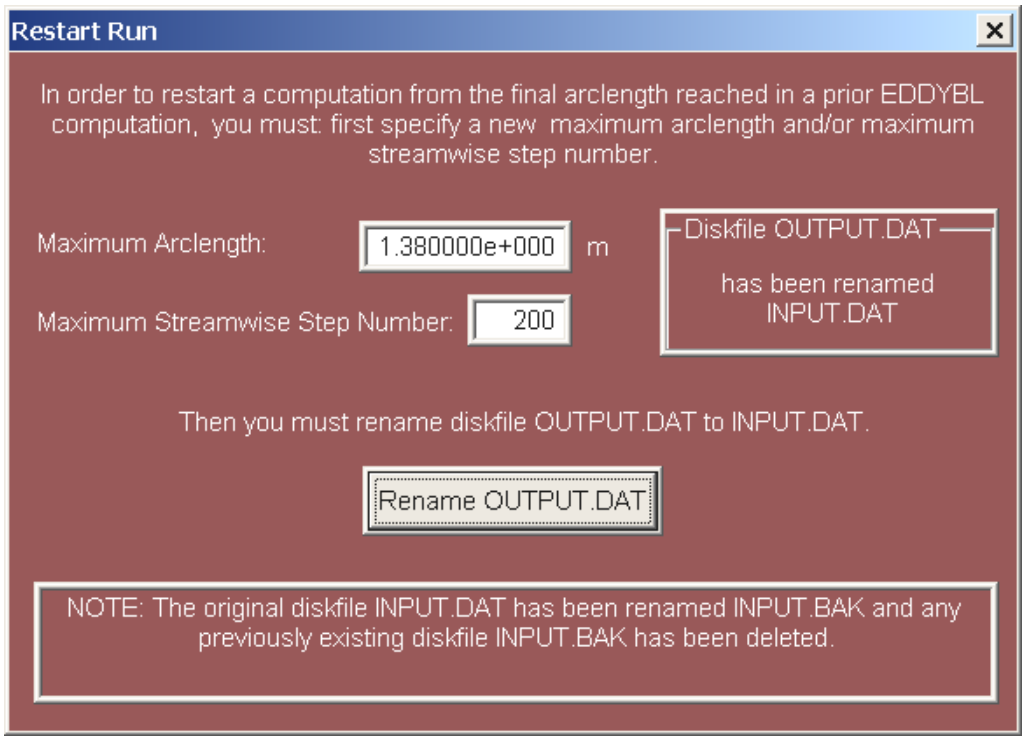

Figure 2.12: *Restart-Run menu of Program* **EDDYBL_DATA** *after modification.*

At this point, input-data files **eddybl.dat** and **input.dat** have been modified as needed to continue your run from where you left off. The file **eddybl.dat** requires no modification as s_{stop} remains smaller than the top end of the interval for which you have defined edge and surface properties. Had you made s_{stop} larger than 6.096 m, you would not be able to use the restart option. It would be necessary to make appropriate changes from the

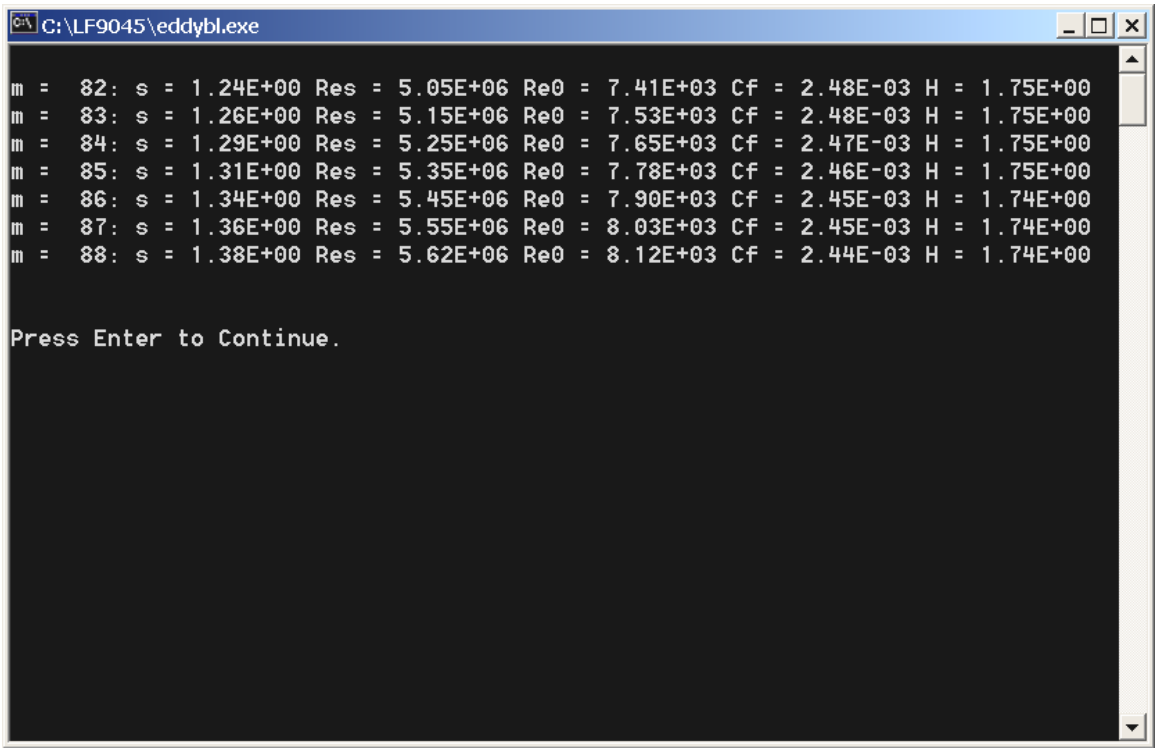

Figure 2.13: *Output from Program* **EDDYBL**.

"Edge/Wall Conditions" menu to make sure edge and surface conditions are defined at least up to the new value of s_{stop} and to then repeat the steps that led to this point.

Now, run Program **EDDYBL** again by clicking the "Run Program EDDYBL" button. Do not click the "Write Input-Data Files" button — doing so would overwrite the file **input.dat**. If you have made no errors the screen will be as in Figure 2.13. Linear interpolation shows that $Re_\theta = 8000$ at an arclength of approximately 1.355 m.

Of course, because computing times with Program **EDDYBL** are just a few seconds on a 3-GHz Pentium-D microcomputer, you could achieve the same result by going to the "Geometry and Grid" menu, changing s_{stop} and rerunning **EDDYBL** until the desired Re_θ is realized at the end of the run. The main purpose of this part of the exercise was to acquaint you the restart feature of **EDDYBL**.

To conclude the exercise, you can now compare your computed velocity profile with the experimental data you entered. To do this, press the ENTER key or close the window to return to the main menu. Now, click on the "Plot Computed Profiles" button to display a plot of your computed results. Scroll down to display the complete velocity profile. Your video display should appear as in Figure 2.14. You can generate hardcopy plot by clicking on the printer icon on the task bar above the plot.

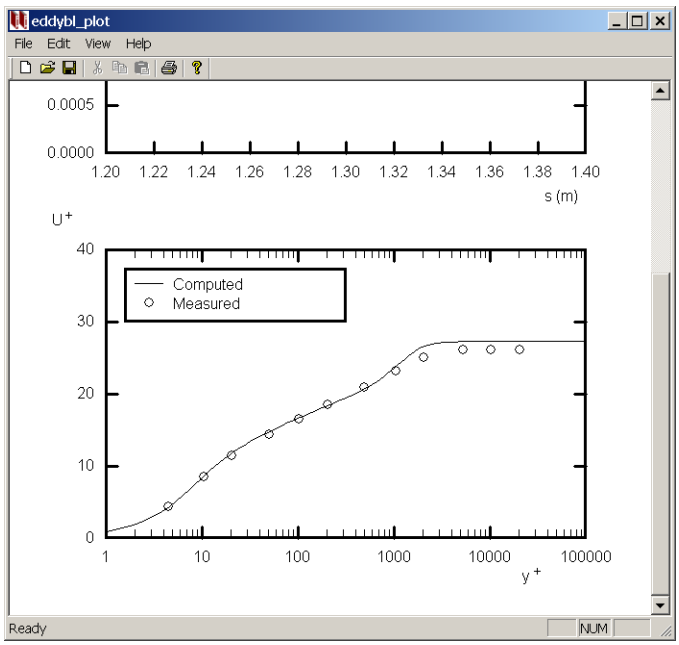

Figure 2.14: *Output from Program* **EDDYBL_PLOT**.

2.3.8 Gas Properties and Profile Printing

Program **EDDYBL_DATA** has two additional menus that you did not use in this exercise. They are the "Gas Properties" menu (Figure 2.15) and the "Profile Printing" menu (Figure 2.16). Note that the menus shown correspond to the **EDDYBL_DATA** default selection of USCS units, not the run you just completed. As with all menus, the values and quoted units are automatically converted when units are changed on the opening menu of Program **EDDYBL_DATA**.

- The "Gas Properties" menu includes several thermodynamic properties and the turbulent Prandtl number, Pr_T. There is one relatively subtle feature regarding the viscosity law. If you wish to use a power-law viscosity relationship rather than the Sutherland law, then you are saying

$$\mu = \mu_r T^{\omega} = \mu_r \frac{T^{1+\omega}}{S+T} \qquad \text{where} \qquad S = 0 \tag{2.2}$$

  Thus, the input parameters are *SU* = 0, *VISCON* = μ_r and *VISPOW* = $1 + \omega$.

- The "Profile Printing" menu selects as many as 10 streamwise stations at which computed velocity, turbulence kinetic energy, etc. profiles are included in the disk file **eddybl.prt**. Profiles are always printed at the final station regardless of what's selected in this menu.

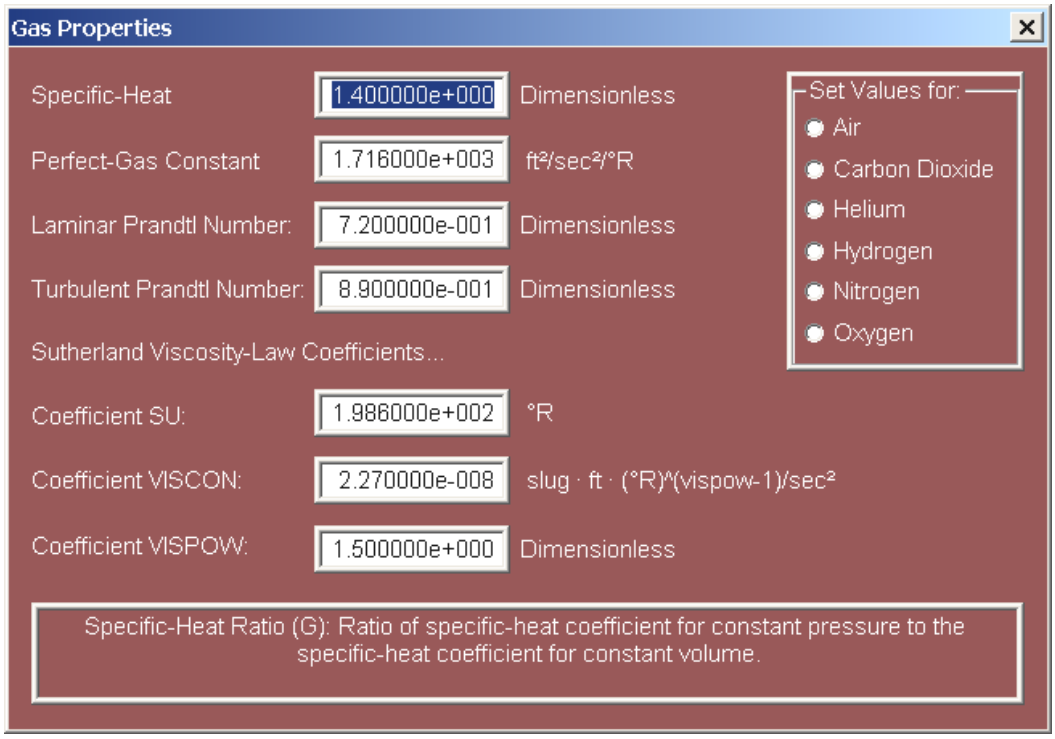

Figure 2.15: *Gas-Properties menu of Program* **EDDYBL_DATA**.

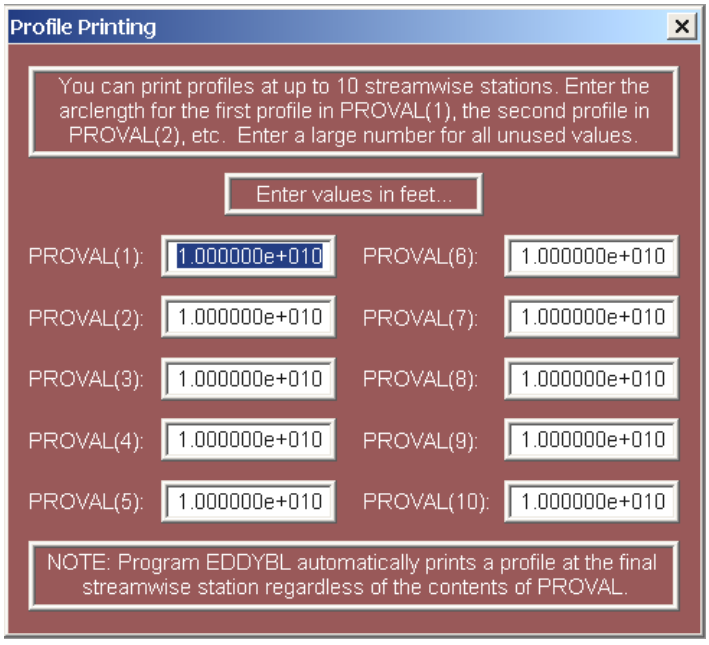

Figure 2.16: *Profile-Printing menu of Program* **EDDYBL_DATA**.

2.3.9 Input-Data Conversion Utility

If you have used earlier versions of **EDDYBL** and would like to use the input-data files you created with this version of **EDDYBL**, you will have to convert them to the new format. Earlier versions of the program used as many as six input-data files. By contrast, All input data is now included in a single file named **eddybl.dat**. The companion CD includes a standalone utility named **EDDYBL_CONVERT** that accomplishes the conversion.

To convert your files, you must first copy the companion CD file **eddybl_convert.exe** and all of the files you created into the same directory. Your data files must include **eddybl.dat**, **presur.dat** and **heater.dat**. Also, if you created any of the three optional files **blocrv.dat**, **exper.dat** and **ploteb.dat**, include them also. When you run Program **EDDYBL_CONVERT**, the following menu will appear (for the case shown, optional file **blocrv.dat** is absent). When you click on the "Convert Files" button, your old **eddybl.dat** is renamed as **eddybl.old** and a new file replaces it. The converted **eddybl.dat** is the only input-data file that you need for this version of Program **EDDYBL**.

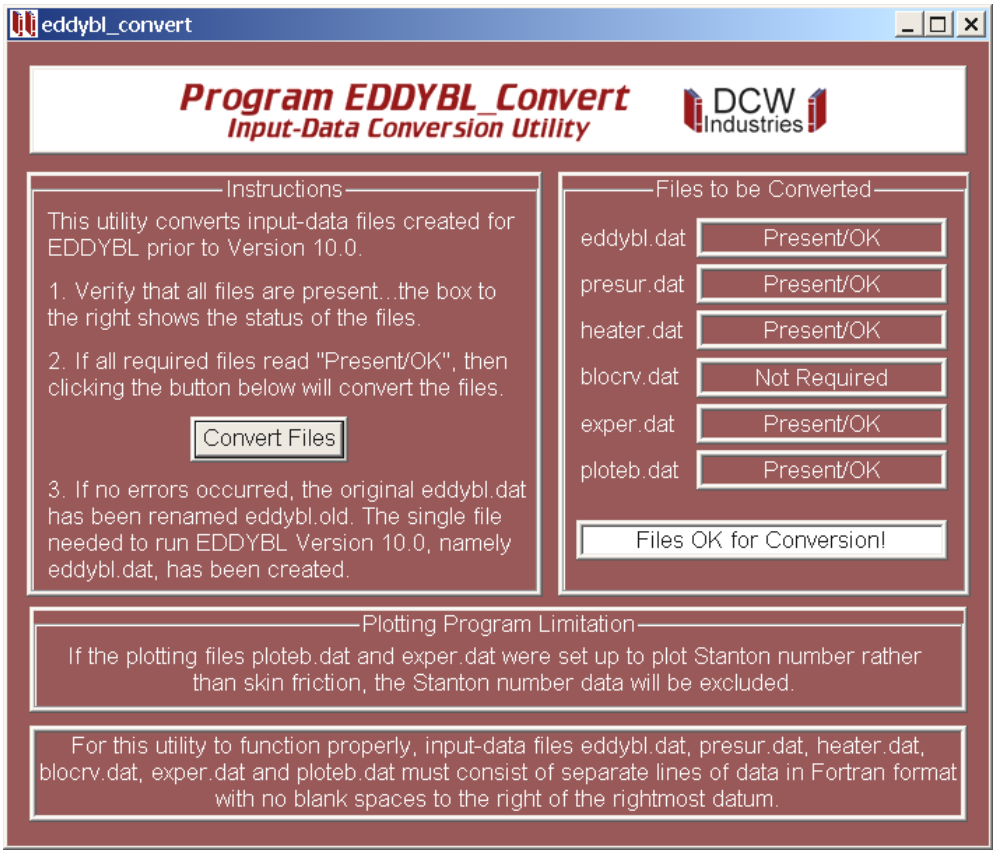

Figure 2.17: *Opening menu of Program* **EDDYBL_CONVERT**.

2.4 Program Details

The companion CD includes Fortran source code for Program **EDDYBL** and its initial-profile utility, Program **EDDYBL_START**. Both can be compiled and run on virtually any computer with a Fortran compiler. The CD also includes executable input-data preparation and plotting programs designed for use on a personal computer with any version of the Microsoft Windows operating system.

2.4.1 Files Included on the CD

There are nine files included on the companion CD for the programs described in this chapter. Programs **EDDYBL_DATA**, **EDDYBL_PLOT** and **EDDYBL_CONVERT** have been written in Microsoft Visual C++, while Programs **EDDYBL** and **EDDYBL_START** are Fortran programs. The companion CD includes source code for the latter two programs. The files on the CD are as follows.

Table 2.2: *Program* **EDDYBL** *files on the CD.*

File Name	Directory	Function
eddybl.exe	Executables	Executable Program **EDDYBL**
eddybl_convert.exe	Executables	File-conversion Program **EDDYBL_CONVERT**
eddybl_data.exe	Executables	Input-data preparation Program **EDDYBL_DATA**
eddybl_plot.exe	Executables	Plotting Program **EDDYBL_PLOT**
eddybl_start.exe	Executables	Initial-profile Program **EDDYBL_START**
comeb	Source Code	File included by Programs **EDDYBL** and **EDDYBL_START**
cpuid	Source Code	File included by Programs **EDDYBL** and **EDDYBL_START**
eddybl.for	Source Code	Fortran source code for Program **EDDYBL**
eddybl_start.for	Source Code	Fortran source code for Program **EDDYBL_START**

The executable files **eddybl.exe** and **eddybl_start.exe** have been compiled using the Lahey Fortran-90 compiler. The programs achieve optimum performance in double precision accuracy. Also, for compatibility with the Visual C++ interface, they are compiled with the Lahey "Windows Console" option. The specific commands used in compiling and linking the programs are as follows.

```
lf90 eddybl -dbl -bind -winconsole

lf90 eddybl_start -dbl -bind -winconsole
```

Additionally, Subdirectory **Input Data/Eddybl** of the companion CD includes input-data files for many of the boundary-layer computations described in *Turbulence Modeling*

for CFD. All of the files are provided in "Zip" format. Each file contains a single input-data file named **eddybl.dat**. Table 2.3 lists all of the files.

Table 2.3: **EDDYBL** *input-data files on the CD.*

File Name	Type of Boundary Layer
Andersen_Blowing.zip	Surface mass injection [Andersen et al. (1972)]
Fernando_Smits.zip	Surface heating, Mach 2.65 [Fernholz and Finley (1981)]
Flow0141.zip	Strong adverse ∇p, Samuel-Joubert [Kline et al. (1981)]
Flow1100.zip	Mild adverse ∇p [Coles and Hirst (1969)]
Flow1200.zip	Strong adverse ∇p [Coles and Hirst (1969)]
Flow1300.zip	Favorable ∇p [Coles and Hirst (1969)]
Flow1400.zip	Zero ∇p [Coles and Hirst (1969)]
Flow2100.zip	Mild adverse ∇p [Coles and Hirst (1969)]
Flow2400.zip	Moderate adverse ∇p [Coles and Hirst (1969)]
Flow2500.zip	Mild adverse ∇p [Coles and Hirst (1969)]
Flow2600.zip	Moderate adverse ∇p [Coles and Hirst (1969)]
Flow2700.zip	Favorable ∇p [Coles and Hirst (1969)]
Flow3300.zip	Moderate adverse ∇p [Coles and Hirst (1969)]
Flow4400.zip	Strong adverse ∇p [Coles and Hirst (1969)]
Flow4500.zip	Moderate adverse ∇p [Coles and Hirst (1969)]
Flow4800.zip	Mild adverse ∇p [Coles and Hirst (1969)]
Flow5300.zip	Strong adverse ∇p [Stratford (1959)]
Flow6300.zip	Favorable ∇p [Coles and Hirst (1969)]
FPBL-Mach0.zip	Zero ∇p, Mach 0.1
FPBL-Mach1.zip	Zero ∇p, Mach 1
FPBL-Mach2.zip	Zero ∇p, Mach 2
FPBL-Mach3.zip	Zero ∇p, Mach 3
FPBL-Mach4.zip	Zero ∇p, Mach 4
FPBL-Mach5.zip	Zero ∇p, Mach 5
Mach10_Watson.zip	Zero ∇p, Mach 10.3 [Fernholz and Finley (1981)]
Mach2_Shutts.zip	Zero ∇p, Mach 2.4 [Fernholz and Finley (1981)]
Mach4_Coles.zip	Zero ∇p, Mach 4.5 [Fernholz and Finley (1981)]
Mach4_Peake.zip	Adverse ∇p, Mach 3.9 [Fernholz and Finley (1981)]
Mach4_Zwarts.zip	Adverse ∇p, Mach 4.0 [Kline et al. (1981)]
So_Mellor_A.zip	Zero ∇p, Curved wall [So and Mellor (1972)]
So_Mellor_B.zip	Adverse ∇p, Curved wall [So and Mellor (1972)]
Transition.zip	Zero ∇p, transitional [Schubauer and Klebanoff (1955)]

2.4.2 Fortran Portability

In the interest of portability, Programs **EDDYBL** and **EDDYBL_START** have been written so that they run on virtually any computer with a Fortran compiler. The programs both use an **include** file named **comeb** that includes all of the common blocks defining program arrays and other key program variables.

Both programs also 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, Pentium)
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 Appendix A for a detailed explanation of what is required as well as a listing of subroutine **NAMSYS**.

2.4.3 Program Input

EDDYBL_DATA creates a disk file named **eddybl.dat**. The format for the quantities included varies. In the first section of the file, 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. Input lines for the Flow 5300 benchmark case (see Section 2.2), for example, appear as follows.

```
2       12 15 17     26  <-- Column number
|        |  | |       |
IUTYPE   =    1
MODEL    =    0
NVISC    =    0
ALPHA    =  5.200000e-01
BETA     =  7.080000e-02
BETAS    =  9.000000e-02
         .
         .
         .
DELTA    =  1.856232e-02
H        =  1.361600e+00
R2D      =  3.048000e-01
RETHET   =  2.295000e+03
SI       =  8.862060e-01
```

The next part of the file includes edge and surface conditions. Program input quantity *number* (Edge/Wall Conditions menu – Subsection 2.3.4) specifies the number of lines of data. The quantities on each line are arclength, s, edge pressure, P_e, wall temperature, T_w, surface heat flux, q_w, surface mass-transfer rate, $\dot{m}$, body radius, r_o, axial distance along the body, z, and body curvature, κ. The format for the *number* lines of data is (1p8e14.6). These data are followed by 7 additional lines for the initial and final slopes of P_e, T_w, q_w, $\dot{m}$, r_o, z, and κ. The format for these lines is (1p2e14.6) as follows.

```
          14            28            42            56            70            84            98           112  <-- Column number
           |             |             |             |             |             |             |             |
 4.361078e-01  9.630181e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  4.361078e-01  0.000000e+00
 8.862060e-01  9.631210e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  8.862060e-01  0.000000e+00
 9.114434e-01  9.632556e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  9.114434e-01  0.000000e+00
 9.136685e-01  9.632896e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  9.136685e-01  0.000000e+00
 9.320784e-01  9.634672e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  9.320784e-01  0.000000e+00
 9.637471e-01  9.635821e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  9.637471e-01  0.000000e+00
 1.356238e+00  9.639996e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  1.356238e+00  0.000000e+00
 1.963125e+00  9.642601e+04  2.800000e+02  0.000000e+00  0.000000e+00  3.048000e-01  1.963125e+00  0.000000e+00
 6.377809e+01  2.182184e+01
 0.000000e+00  0.000000e+00
 0.000000e+00  0.000000e+00
 0.000000e+00  0.000000e+00
 0.000000e+00  0.000000e+00
 1.000000e+00  1.000000e+00
 0.000000e+00  0.000000e+00
```

The final part of **eddybl.dat** includes experimental data information in mixed format as shown below.

```
1         12 15 17  22  <-- Column number
|          |  | |    |
******
 ISTART    =    1
 NCFE      =    5
 8.8605e-01 3.6800e-03
 9.1410e-01 2.0700e-03
 9.2598e-01 9.9000e-04
 1.0762e+00 5.5000e-04
 1.2506e+00 5.3000e-04
             Stratford
 IPOSCF    =    2
 NUPLUS    =   23
 5.8000e+00 6.6400e+00
 1.7000e+01 1.1500e+01
 3.4100e+01 1.3000e+01
 5.1100e+01 1.4590e+01
 6.8100e+01 1.6030e+01
 1.0220e+02 1.7350e+01
 1.3630e+02 2.0640e+01
 1.7040e+02 2.3150e+01
 2.0440e+02 2.7370e+01
 2.3850e+02 3.0770e+01
 2.7260e+02 3.4060e+01
 3.0670e+02 3.7050e+01
 3.4070e+02 4.0370e+01
 3.7480e+02 4.3940e+01
 4.0890e+02 4.7240e+01
 4.4300e+02 5.0110e+01
 4.7700e+02 5.3780e+01
 5.1110e+02 5.6900e+01
 5.4520e+02 5.8740e+01
 5.7930e+02 5.9670e+01
 6.1330e+02 6.0590e+01
 6.4740e+02 6.1130e+01
 6.8150e+02 6.1250e+01
             Stratford
 IPOSUP    =    1
```

2.4.4 Program Output

Program **EDDYBL_START** can be executed from **EDDYBL_DATA** or as a standalone program. Output from **EDDYBL_START** consists of two binary disk files, viz.,

- **input.dat** contains initial profiles for all flow properties.
- **table.dat** contains cubic spline fits for all edge- and wall-condition arrays.

Program **EDDYBL** reads both of these disk files at the beginning of each computation. **EDDYBL_START** also writes initial-profile information to disk file **eddybl_start.prt**. The file's contents are displayed on the video display and can also be viewed from any text editor.

Program **EDDYBL** can also be executed from **EDDYBL_DATA** or as a standalone program. Output from **EDDYBL** consists of the following five disk files.

- **eddybl.prt** is a comprehensive print file detailing results of the computation. The next subsection describes the contents of the file.
- **output.dat** is a binary file with sufficient information to restart a run if desired.
- **profil.dat** is a plotting file containing velocity-profile data at the final station. The first line is the number of points in the profile, *iedge*, with format (i6). This is followed by *iedge* lines of sublayer-scaled distance and velocity pairs, y^+ and U^+, with format (1p2e12.4).
- **profil2c.dat** is a file that can be used to initialize Program **EDDY2C** (Chapter 3). The first line is the number of points in the profiles, *iedge*, with format (i6). This is followed by *iedge* lines of dimensional quantities including normal distance, y, streamwise velocity, U, temperature, T, density, ρ, turbulence kinetic energy, k, specific dissipation rate, ω, and the Reynolds stress components, τ_{zz}, τ_{xx}, τ_{xy}, and τ_{yy} with format (1p10e12.4).
- **wall.dat** is a plotting file containing integral-parameter data. Each line includes streamwise distance, s, and skin friction based on edge conditions, c_{f_e}, with format (1p2e12.4). The last line of the file has $s = c_{f_e} = -999$ to indicate that there are no more data.

Plotting program **EDDYBL_PLOT** uses the three disk files **eddybl.dat**, **profil.dat** and **wall.dat**, and can be run from **EDDYBL_DATA** or as a standalone program.

2.4.5 EDDYBL Output Parameters

Printed output from Program **EDDYBL** consists of dimensionless boundary-layer profiles, and integral parameters, some of which are dimensional. The integral-parameter portion of **eddybl.prt** includes the following quantities.

Symbol	Meaning	USCS Units	SI Units	Symbol	Meaning	USCS Units	SI Units
F	Force	pounds (lb)	Newtons (N)	Q	Heat flux	Btu/second (Btu/sec)	Watts (W)
L	Length	feet (ft)	meters (m)	T	Time	seconds (sec)	seconds (sec)
M	Mass	slugs (sl)	kilograms (kg)	Θ	Temperature	^{o}Rankine	Kelvins

Name	Symbol/Equation	Definition	Dimensions
beta	$\beta = (2\xi/u_e)du_e/d\xi$	Pressure-gradient parameter	None
Cfe	$c_{fe} = 2\tau_w/\rho_e u_e^2$	Skin friction based on $\bar{\rho}_e$	None
Cfw	$c_{fw} = 2\tau_w/\rho_w u_e^2$	Skin friction based on $\bar{\rho}_w$	None
delta	δ	Boundary-layer thickness	L
delta$*$	δ^*	Displacement thickness	L
dPe/ds	$d(\overline{p}/\rho_\infty U_\infty^2)/d\bar{s}$	Dimensionless pressure gradient	None
dTe/ds	$d(T_e/T_{ref})/d\bar{s}$	Dimensionless temperature gradient	None
dUe/ds	$d(u_e/U_\infty)/d\bar{s}$	Dimensionless velocity gradient	None
H	$H = \delta^*/\theta$	Shape factor	None
hdot	$\dot{h} = q_w/(T_w - T_{aw})$	Heat-transfer coefficient	$\mathrm{QL^{-2}\Theta^{-1}}$
Iedge	N	Total number of mesh points in the B.L.	None
Itro		Number of iterations	None
kmax	$\sqrt{\beta^*}(\bar{\rho}k)_{max}/\tau_w$	Maximum turbulence kinetic energy	None
M	m	Streamwise step number	None
Me	M_e	Edge Mach number	None
Mue	μ_e	Edge molecular viscosity	$\mathrm{ML^{-1}T^{-1}}$
Ne	N_e	Mesh point number at the B.L. edge	None
Negtiv		Number of points where $k, \omega, \epsilon < 0$	None
Nerror		Number of points not converged	None
Nskip		Number of points below U^+ = USTOP	None
Nste	$\dot{h}/\rho_e u_e c_p$	Stanton number based on $\bar{\rho}_e$	None
Nstw	$\dot{h}/\rho_w u_e c_p$	Stanton number based on $\bar{\rho}_w$	None
Nue	$Pr\, s\dot{h}/\mu_e c_p$	Nusselt number based on μ_e	None
Nuw	$Pr\, s\dot{h}/\mu_w c_p$	Nusselt number based on μ_w	None
Pe	$\overline{p}$	Edge pressure	$\mathrm{FL^{-2}}$
qw	q_w	Surface heat flux	$\mathrm{QL^{-2}}$
radius	r_o	Body radius	L
Recov	r	Recovery factor	None
Redel$*$	$Re_{\delta^*} = \rho_e u_e \delta^*/\mu_e$	Reynolds number based on δ^*	None
Res	$Re_s = \rho_e u_e s/\mu_e$	Reynolds number based on s	None
Rethet	$Re_\theta = \rho_e u_e \theta/\mu_e$	Reynolds number based on θ	None
Rhoe	ρ_e	Edge density	$\mathrm{ML^{-3}}$
rho$*$vw	$\rho_w v_w$	Surface mass flux	$\mathrm{ML^{-2}T^{-1}}$
s	s	Arclength	L
tauw	τ_w	Surface shear stress	$\mathrm{FL^{-2}}$
Te	T_e	Edge temperature	Θ
theta	θ	Momentum thickness	L
Ue	u_e	Edge velocity	$\mathrm{LT^{-1}}$
utau	u_τ	Friction velocity	$\mathrm{LT^{-1}}$
xi	$\xi = \int_0^s \rho_e u_e \mu_e r_o^{2j} ds$	Transformed streamwise coordinate	L
yplus	y_2^+	Value of y^+ nearest the surface	None
z	z	Axial distance	L

The dimensionless-profiles portion of **eddybl.prt** includes the following.

Name	Symbol/Equation	Definition
i	i	Mesh point number
y/delta	y/δ	Dimensionless normal distance
u/Ue	$\tilde{u}/u_e$	Dimensionless velocity
yplus	$y^+ = u_\tau y/\nu_w$	Compressible sublayer-scaled distance
uplus	$U^+ = u^*/u_\tau$	Compressible sublayer-scaled velocity
k/Ue**2	k/u_e^2	Dimensionless turbulence kinetic energy
omega	$\nu_e\omega/u_e^2$	Dimensionless specific dissipation rate
eps/mu	μ_T/μ	Dimensionless eddy viscosity
L/delta	$\sqrt{k/\beta^*}/(\omega\delta)$	Dimensionless turbulence length scale
uv/tauw	$-\overline{\rho u'v'}/\tau_w$	Dimensionless Reynolds shear stress
T/Te	$\tilde{T}/T_e$	Temperature ratio

2.4.6 Turbulence Models Implemented in EDDYBL

There are 15 turbulence models implemented in Program **EDDYBL**, which uses the two input parameters *model* and *nvisc* to make the selection. Table 2.4 lists the models implemented.

Table 2.4: *Turbulence models implemented in EDDYBL.*

Model	Type	*model*	*nvisc*
None (laminar flow)	—	-1	Not used
k-ω (high Re) - Wilcox (2006)	2-Equation	0	0
k-ω (low Re) - Wilcox (2006)	2-Equation	0	1
Stress-ω (high Re) - Wilcox (2006)	Stress-Transport	1	0
Stress-ω (high Re) - Wilcox (2006)	Stress-Transport	1	1
k-ϵ - Jones-Launder (1972)	2-Equation	2	0
k-ϵ - Launder-Sharma (1974)	2-Equation	2	1
k-ϵ - Lam-Bremhorst (1981)	2-Equation	2	2
k-ϵ - Chien (1982)	2-Equation	2	3
k-ϵ - Yang-Shih (1993)	2-Equation	2	4
k-ϵ - Fan-Lakshminarayana-Barnett (1993)	2-Equation	2	5
Baldwin-Barth (1990)	1-Equation	3	Not used
Spalart-Allmaras (1992)	1-Equation	4	Not used
Cebeci-Smith (1974)	Algebraic	5	Not used
Baldwin-Lomax (1978)	Algebraic	6	Not used
Johnson-King (1985)	$\frac{1}{2}$-Equation	7	Not used

2.4.7 Selecting Laminar, Transitional or Turbulent Flow

Program **EDDYBL** can run in three different modes corresponding to (1) pure laminar flow, (2) transition from laminar to turbulent flow, and (3) pure turbulent flow. The two test cases exercise **EDDYBL** in its pure turbulent mode in which integral parameters are specified.

To run in transitional mode, simply click on the "Laminar" radio button of the "Initial Profiles" menu (see Figure 2.9). As a result, exact laminar velocity and temperature profiles will be generated in conjunction with approximate laminar profiles for the various turbulence-model parameters. The transition point is determined automatically by the model equations and depends strongly upon the freestream values of k and ω that are specified in the "Freestream Conditions" menu (see Figure 2.6). To obtain physically realistic transition Reynolds numbers you must use the k-ω and Stress-ω models with viscous modifications. Although the k-ϵ models are capable of predicting transition, extremely small streamwise steps are needed with **EDDYBL**, and stable computation is very difficult to achieve. Even if you are not interested in transition, this mode is nevertheless useful as it provides an alternate method for generating turbulent starting profiles, e.g., by starting laminar and running up to a desired value of Re_θ.

The k-ω and Stress-ω models are very robust and can be integrated through transition with and without low-Reynolds-number corrections. By contrast, many of the other models require smaller streamwise steps than the k-ω and Stress-ω models, and, in general, cannot be integrated through transition unless extremely small steps are taken. If integral properties are unknown and a solution with a model other than k-ω or Stress-ω is desired, the optimum procedure is to start laminar with the k-ω model and integrate through transition. Then, select the desired model and use the **Restart** option to continue the run (see Subsection 2.3.7).

Finally, to run **EDDYBL** as a pure laminar boundary-layer program, the turbulence model can be suppressed by selecting "None (laminar flow)" as the "Turbulence Model" on the opening menu of Program **EDDYBL_START**. When this is done, turbulence-model computations are bypassed and no transition to turbulence occurs.

2.5 Equation and Transformation Details

Program **EDDYBL** applies to attached, compressible, two-dimensional and axisymmetric boundary layers. The program includes effects of surface roughness, surface mass transfer, surface curvature, and low-Reynolds-number corrections for the Wilcox (2006) k-ω and Stress-ω models. The program also includes the algebraic, one-equation and two-equation models listed in Table 2.4.

Program **EDDYBL** uses conventional Levy-Lees variables [see Hayes and Probstein (1959)] and much of the program notation follows that of Harris and Blanchard (1982). The numerical procedure is the Blottner (1974) variable-grid method augmented with an algorithm devised by Wilcox (1981) to permit large streamwise steps. As implemented in **EDDYBL**, the method has been demonstrated to be second-order accurate is both streamwise and normal directions. Section 7.3 of *Turbulence Modeling for CFD* provides an in-depth discussion of the algorithm.

This section first presents the governing equations for mean-flow properties and all turbulence-model equations implemented in the program. This is followed by the transformed, nondimensional form of the equations for the k-ω, Stress-ω and k-ϵ models.

2.5.1 Mean-Flow Equations

The equations governing conservation of mass, momentum and mean energy for all models are the same. For compressible two-dimensional ($j = 0$) and axisymmetric ($j = 1$) boundary layers, the program uses body-oriented coordinates (s, n), where s is arclength and n is distance normal to the surface. The equations are as follows.

$$\frac{\partial}{\partial s}\left(\bar{\rho}\tilde{u}\right) + \frac{1}{r^j}\frac{\partial}{\partial n}\left(r^j\bar{\rho}\tilde{v}\right) = 0 \tag{2.3}$$

$$\bar{\rho}\tilde{u}\frac{\partial\tilde{u}}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial\tilde{u}}{\partial n} = -\frac{dP}{ds} + \frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu\frac{\partial\tilde{u}}{\partial n} + \bar{\rho}\tau\right)\right] \tag{2.4}$$

$$\bar{\rho}\tilde{u}\frac{\partial\tilde{h}}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial\tilde{h}}{\partial n} = \tilde{u}\frac{dP}{ds} + \mu\left(\frac{\partial\tilde{u}}{\partial n}\right)^2 + \bar{\rho}\epsilon + \frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\frac{\mu}{Pr_L} + \frac{\mu_T}{Pr_T}\right)\frac{\partial\tilde{h}}{\partial n}\right] \tag{2.5}$$

The perfect-gas law is used as the equation of state and the fluid is assumed calorically perfect so that

$$P = \bar{\rho}R\tilde{T} \qquad \text{and} \qquad \tilde{h} = c_p\tilde{T} \tag{2.6}$$

In Equations (2.3) through (2.6): $\tilde{u}$ and $\tilde{v}$ are streamwise and normal mass-averaged velocity components; $\bar{\rho}$, P and $\tilde{h}$ are fluid density, pressure and enthalpy; μ and μ_T are molecular and eddy viscosity; τ is specific Reynolds shear stress; ϵ is turbulence dissipation rate; Pr_L and Pr_T are laminar and turbulent Prandtl numbers; $\tilde{T}$ is mass-averaged temperature; R is the perfect-gas constant; and c_p is specific heat at constant pressure.

2.5.2 k-ω and Stress-ω Model Equations

For both the k-ω and Stress-ω models the dissipation, ϵ, is given by

$$\epsilon = \beta^* \omega k \tag{2.7}$$

where k is turbulence kinetic energy and ω is specific dissipation rate. The equations for k and ω applicable to compressible boundary layers are as follows.

$$\bar{\rho}\tilde{u}\frac{\partial k}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial k}{\partial n} = \bar{\rho}\tau\frac{\partial \tilde{u}}{\partial n} - \beta^*\bar{\rho}\omega k + \frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu + \sigma^*\alpha^*\frac{\bar{\rho}k}{\omega}\right)\frac{\partial k}{\partial n}\right] \tag{2.8}$$

$$\bar{\rho}\tilde{u}\frac{\partial \omega}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial \omega}{\partial n} = \alpha\frac{\omega}{k}\bar{\rho}\tau\frac{\partial \tilde{u}}{\partial n} - \beta\bar{\rho}\omega^2 + \frac{\sigma_d}{\omega}\frac{\partial k}{\partial n}\frac{\partial \omega}{\partial n} + \frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu + \sigma\alpha^*\frac{\bar{\rho}k}{\omega}\right)\frac{\partial \omega}{\partial n}\right] \tag{2.9}$$

For the k-ω model, the Reynolds shear stress is given by

$$\bar{\rho}\tau = \alpha^*\frac{\bar{\rho}k}{\tilde{\omega}}\frac{\partial \tilde{u}}{\partial n}, \qquad \tilde{\omega} = \max\left\{\omega,\ C_{lim}\frac{\partial \tilde{u}/\partial n}{\sqrt{\beta^*_\infty/\alpha^*}}\right\}, \qquad C_{lim} = \frac{7}{8} \tag{2.10}$$

For the Stress-ω model, the Reynolds stresses are computed from the following equations:

$$\begin{aligned}\bar{\rho}\tilde{u}\frac{\partial \tau}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial \tau}{\partial n} = &\left[(1-\hat{\alpha})\sigma_y - \hat{\beta}\sigma_x + \frac{2}{3}(1-\hat{\alpha}-\hat{\beta}+\frac{3}{4}\hat{\gamma})k\right]\bar{\rho}\frac{\partial \tilde{u}}{\partial n} \\ &-C_1\beta^*\bar{\rho}\omega\tau + \frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu + \sigma^*\mu_T\right)\frac{\partial \tau}{\partial n}\right]\end{aligned} \tag{2.11}$$

$$\begin{aligned}\bar{\rho}\tilde{u}\frac{\partial \sigma_x}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial \sigma_x}{\partial n} = &\frac{2}{3}\left[2(1-\hat{\alpha}) + \hat{\beta}\right]\bar{\rho}\tau\frac{\partial \tilde{u}}{\partial n} - C_1\beta^*\bar{\rho}\omega\sigma_x \\ &+\frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu + \sigma^*\mu_T\right)\frac{\partial \sigma_x}{\partial n}\right]\end{aligned} \tag{2.12}$$

$$\begin{aligned}\bar{\rho}\tilde{u}\frac{\partial \sigma_y}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial \sigma_y}{\partial n} = &-\frac{2}{3}\left[(1-\hat{\alpha}) + 2\hat{\beta}\right]\bar{\rho}\tau\frac{\partial \tilde{u}}{\partial n} - C_1\beta^*\bar{\rho}\omega\sigma_y \\ &+\frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu + \sigma^*\mu_T\right)\frac{\partial \sigma_y}{\partial n}\right]\end{aligned} \tag{2.13}$$

where the quantities σ_x and σ_y are stress-deviator components and μ_T is eddy viscosity, given by

$$\mu_T = \alpha^*\frac{\bar{\rho}k}{\omega}, \qquad \sigma_x = \frac{\overline{\rho u'^2}}{\bar{\rho}} - \frac{2}{3}k, \qquad \sigma_y = \frac{\overline{\rho v'^2}}{\bar{\rho}} - \frac{2}{3}k \tag{2.14}$$

The various closure coefficients, viz., α, β, β^*, σ, σ^*, σ_d, C_1, C_2, $\hat{\alpha}$, $\hat{\beta}$ and $\hat{\gamma}$ are given by the following. First, we define the fully turbulent (subscript ∞), incompressible (subscript i) values by

$$\alpha_\infty = \frac{13}{25}, \quad \beta_o = 0.0708, \quad \beta^*_\infty = \frac{9}{100}, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{3}{5} \tag{2.15}$$

$$\hat{\alpha}_\infty = \frac{8 + C_2}{11}, \quad \hat{\beta}_\infty = \frac{8C_2 - 2}{11}, \quad \hat{\gamma}_\infty = \frac{60C_2 - 4}{55}, \quad C_1 = \frac{9}{5}, \quad C_2 = \frac{10}{19} \tag{2.16}$$

$$\sigma_d = \begin{cases} 0, & \dfrac{\partial k}{\partial n}\dfrac{\partial \omega}{\partial n} \le 0 \\ \sigma_{do}, & \dfrac{\partial k}{\partial n}\dfrac{\partial \omega}{\partial n} > 0 \end{cases}, \quad \sigma_{do} = \frac{1}{8} \tag{2.17}$$

On the one hand, if low-Reynolds-number corrections are excluded from the k-ω and Stress-ω models, we simply use:

$$\alpha = \alpha_\infty, \quad \beta^*_i = \beta^*_\infty, \quad \hat{\alpha} = \hat{\alpha}_\infty, \quad \hat{\beta} = \hat{\beta}_\infty, \quad \hat{\gamma} = \hat{\gamma}_\infty \tag{2.18}$$

On the other hand, if low-Reynolds-number corrections are included in the k-ω model, we use the following:

$$\left.\begin{aligned} \alpha^* &= \frac{\alpha^*_o + Re_T/R_k}{1 + Re_T/R_k} \\ \alpha &= \alpha_\infty \cdot \frac{\alpha_o + Re_T/R_\omega}{1 + Re_T/R_\omega} \cdot (\alpha^*)^{-1} \\ \beta^*_i &= \beta^*_\infty \cdot \frac{100\beta_o/27 + (Re_T/R_\beta)^4}{1 + (Re_T/R_\beta)^4} \end{aligned}\right\} \tag{2.19}$$

where

$$\alpha^*_o = \frac{\beta_o}{3}, \quad \alpha_o = \frac{1}{9}, \quad R_\beta = 8, \quad R_k = 6, \quad R_\omega = 2.61 \tag{2.20}$$

The quantity Re_T is turbulence Reynolds number defined by

$$Re_T = \frac{k}{\omega\nu} \tag{2.21}$$

If low-Reynolds-number corrections are included in the Stress-ω model, we use the following:

$$\left.\begin{aligned}
\alpha^* &= \frac{\alpha_o^* + Re_T/R_k}{1 + Re_T/R_k} \\
\alpha &= \alpha_\infty \cdot \frac{\alpha_o + Re_T/R_\omega}{1 + Re_T/R_\omega} \cdot \frac{3 + Re_T/R_\omega}{3\alpha_o^* + Re_T/R_\omega} \\
\beta^* &= \beta_\infty^* \cdot \frac{100\beta_o/27 + (Re_T/R_\beta)^4}{1 + (Re_T/R_\beta)^4} \\
\hat{\alpha} &= \frac{1 + \hat{\alpha}_\infty (Re_T/R_\beta)^4}{1 + (Re_T/R_\beta)^4} \\
\hat{\beta} &= \hat{\beta}_\infty \cdot \frac{(Re_T/R_\beta)^4}{1 + (Re_T/R_\beta)^4} \\
\hat{\gamma} &= \hat{\gamma}_\infty \cdot \frac{\hat{\gamma}_o + (Re_T/R_\beta)^4}{1 + (Re_T/R_\beta)^4} \\
C_1 &= \frac{9}{5} \cdot \frac{5/3 + (Re_T/R_\beta)^4}{1 + (Re_T/R_\beta)^4}
\end{aligned}\right\} \tag{2.22}$$

where

$$\alpha_o^* = \frac{\beta_o}{3}, \quad \alpha_o = \frac{1}{9}, \quad R_\beta = 8, \quad R_k = 6, \quad R_\omega = \frac{22}{9} \tag{2.23}$$

Finally, the compressible values of β and β^* are

$$\beta = \beta_o \left[1 - \frac{\beta_i^*}{\beta_o} \xi^* F(M_t)\right], \quad \beta^* = \beta_i^* \left[1 + \xi^* F(M_t)\right], \quad \xi^* = 2 \tag{2.24}$$

The compressibility function $F(M_t)$ is given by

$$F(M_t) = \begin{cases} 0, & M_t \le M_{to} \\ M_t^2 - M_{to}^2, & M_t > M_{to} \end{cases} \tag{2.25}$$

where $M_t^2 \equiv 2k/a^2$, a is the speed of sound, and M_{to} is given by

$$M_{to} = 1/4 \tag{2.26}$$

2.5.3 k-ϵ Model Equations

For the k-ϵ model the equations for k and ϵ are:

$$\bar{\rho}\tilde{u}\frac{\partial k}{\partial s} + \bar{\rho}\tilde{v}\frac{\partial k}{\partial n} = \bar{\rho}\tau\frac{\partial \tilde{u}}{\partial n} - \bar{\rho}\epsilon + \frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu + \mu_T/\sigma_k\right)\frac{\partial k}{\partial n}\right] \tag{2.27}$$

$$\bar{\rho}\tilde{u}\frac{\partial\tilde{\epsilon}}{\partial s}+\bar{\rho}\tilde{v}\frac{\partial\tilde{\epsilon}}{\partial n}=f_1C_{\epsilon1}\frac{\tilde{\epsilon}}{k}\bar{\rho}\tau\frac{\partial\tilde{u}}{\partial n}-f_2C_{\epsilon2}\bar{\rho}\frac{\tilde{\epsilon}^2}{k}+\bar{\rho}E+\frac{1}{r^j}\frac{\partial}{\partial n}\left[r^j\left(\mu+\mu_T/\sigma_\epsilon\right)\frac{\partial\tilde{\epsilon}}{\partial n}\right] \quad (2.28)$$

where

$$\epsilon=\tilde{\epsilon}+\epsilon_o \quad (2.29)$$

and the eddy viscosity is

$$\mu_T=C_\mu f_\mu\bar{\rho}k^2/\tilde{\epsilon} \quad (2.30)$$

Program **EDDYBL** includes six low-Reynolds-number versions of the k-ϵ model. The models differ in the form of the damping functions f_μ, f_1, f_2, ϵ_o, E, in the values of the closure coefficients, and in the surface boundary condition imposed on $\tilde{\epsilon}$. The damping functions depend upon one or more of the following three dimensionless parameters.

$$Re_T=\frac{k^2}{\tilde{\epsilon}\nu},\qquad R_y=\frac{k^{1/2}n}{\nu},\qquad y^+=\frac{u_\tau n}{\nu} \quad (2.31)$$

The damping functions, closure coefficients and surface boundary condition on $\tilde{\epsilon}$ for the six models are as follows.

Jones-Launder Model

$$\left.\begin{aligned}
&f_\mu=e^{-2.5/(1+Re_T/50)}\\
&f_1=1\\
&f_2=1-0.3e^{-Re_T^2}\\
&\epsilon_o=2\nu\left(\partial\sqrt{k}/\partial n\right)^2\\
&E=2\nu\nu_T\left(\partial^2\tilde{u}/\partial n^2\right)^2\\
&C_{\epsilon1}=1.55,\quad C_{\epsilon2}=2.00,\quad C_\mu=0.09,\quad \sigma_k=1.0,\quad \sigma_\epsilon=1.3\\
&\tilde{\epsilon}=0\quad\text{at}\quad n=0
\end{aligned}\right\} \quad (2.32)$$

Launder-Sharma Model

$$\left.\begin{aligned}
&f_\mu=e^{-3.4/(1+Re_T/50)^2}\\
&f_1=1\\
&f_2=1-0.3e^{-Re_T^2}\\
&\epsilon_o=2\nu\left(\partial\sqrt{k}/\partial n\right)^2\\
&E=2\nu\nu_T\left(\partial^2\tilde{u}/\partial n^2\right)^2\\
&C_{\epsilon1}=1.44,\quad C_{\epsilon2}=1.92,\quad C_\mu=0.09,\quad \sigma_k=1.0,\quad \sigma_\epsilon=1.3\\
&\tilde{\epsilon}=0\quad\text{at}\quad n=0
\end{aligned}\right\} \quad (2.33)$$

Lam-Bremhorst Model

$$\left.\begin{array}{l} f_\mu = \left(1 - e^{-0.0165 R_y}\right)^2 \ (1 + 20.5/Re_T) \\ f_1 = 1 + (0.05/f_\mu)^3 \\ f_2 = 1 - e^{-Re_T^2} \\ \epsilon_o = 0 \\ E = 0 \\ C_{\epsilon 1} = 1.44, \quad C_{\epsilon 2} = 1.92, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3 \\ \tilde{\epsilon} = \nu \dfrac{\partial^2 k}{\partial n^2} \qquad \text{at} \qquad n = 0 \end{array}\right\} \tag{2.34}$$

Chien Model

$$\left.\begin{array}{l} f_\mu = 1 - e^{-0.0115 y^+} \\ f_1 = 1 \\ f_2 = 1 - 0.22 e^{-(Re_T/6)^2} \\ \epsilon_o = 2\nu \dfrac{k}{n^2} \\ E = -2\nu \dfrac{\tilde{\epsilon}}{n^2} e^{-y^+/2} \\ C_{\epsilon 1} = 1.35, \quad C_{\epsilon 2} = 1.80, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3 \\ \tilde{\epsilon} = 0 \qquad \text{at} \qquad n = 0 \end{array}\right\} \tag{2.35}$$

Yang-Shih Model

$$\left.\begin{array}{l} f_\mu = \dfrac{\left[1 - exp\left(-1.5 \cdot 10^{-4} R_y - 5 \cdot 10^{-7} R_y^3 - 10^{-10} R_y^5\right)\right]^{1/2}}{\left(1 + 1/\sqrt{Re_T}\right)} \\ f_1 = \sqrt{Re_T} \Big/ \left(1 + \sqrt{Re_T}\right) \\ f_2 = \sqrt{Re_T} \Big/ \left(1 + \sqrt{Re_T}\right) \\ \epsilon_o = 0 \\ E = \nu \nu_T \left(\dfrac{\partial^2 \tilde{u}}{\partial n^2}\right)^2 \\ C_{\epsilon 1} = 1.44, \quad C_{\epsilon 2} = 1.92, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3 \\ \tilde{\epsilon} = 2\nu \left(\dfrac{\partial \sqrt{k}}{\partial n}\right)^2 \qquad \text{at} \qquad n = 0 \end{array}\right\} \tag{2.36}$$

Fan-Lakshminarayana-Barnett Model

$$\left.\begin{aligned}
&f_\mu = 0.4\frac{f_w}{\sqrt{Re_T}} + \left(1 - 0.4\frac{f_w}{\sqrt{Re_T}}\right)\left(1 - e^{-R_y/42.63}\right)^3 \\
&f_1 = 1 \\
&f_2 = \left[1 - 0.22e^{-(Re_T/6)^2}\right] f_w^2 \\
&f_w = 1 - exp\left[-\frac{\sqrt{R_y}}{2.30} + \left(\frac{\sqrt{R_y}}{2.30} - \frac{R_y}{8.89}\right)\left(1 - e^{-R_y/20}\right)^3\right] \\
&\epsilon_o = 0 \\
&E = 0 \\
&C_{\epsilon 1} = 1.39, \quad C_{\epsilon 2} = 1.80, \quad C_\mu = 0.09, \quad \sigma_k = 1.0, \quad \sigma_\epsilon = 1.3 \\
&\frac{\partial \tilde{\epsilon}}{\partial n} = 0 \qquad \text{at} \qquad n = 0
\end{aligned}\right\} \tag{2.37}$$

2.5.4 Transformed Equations

The boundary-layer equations are singular at the leading edge of a body. As noted above, the program uses conventional Levy-Lees variables (ξ, η) to remove this singularity. Body oriented physical coordinates (s, n) are related to transformed coordinates (ξ, η) according to

$$d\xi = \bar{\rho}_e \tilde{u}_e \mu_e r_o^{2j}\, ds \qquad \text{and} \qquad d\eta = \frac{\bar{\rho}\tilde{u}_e (r_o + n)^j\, dn}{\sqrt{2\xi}} \tag{2.38}$$

where r_o is body radius and subscript e denotes boundary-layer edge. Equivalently, we can write

$$\xi(s) = \int_0^s \bar{\rho}_e \tilde{u}_e \mu_e r_o^{2j}\, ds \qquad \text{and} \qquad \eta(s,n) = \frac{\bar{\rho}_e \tilde{u}_e r_o^j}{\sqrt{2\xi}} \int_0^n \left(\frac{\bar{\rho}}{\bar{\rho}_e}\right) t^j\, dn \tag{2.39}$$

The quantity t is transverse curvature defined by

$$t = \frac{r}{r_o} \tag{2.40}$$

The relations between derivatives in the physical (s, n) and transformed (ξ, η) coordinate system are as follows:

$$\left(\frac{\partial}{\partial s}\right)_n = \bar{\rho}_e \tilde{u}_e \mu_e r_o^{2j} \left(\frac{\partial}{\partial \xi}\right)_\eta + \left(\frac{\partial \eta}{\partial s}\right)_n \left(\frac{\partial}{\partial \eta}\right)_\xi \tag{2.41}$$

$$\left(\frac{\partial}{\partial n}\right)_s = \frac{\bar{\rho}_e \tilde{u}_e r_o^j t^j}{\sqrt{2\xi}} \left(\frac{\bar{\rho}}{\bar{\rho}_e}\right) \left(\frac{\partial}{\partial \eta}\right)_\xi \tag{2.42}$$

The dependent variables are also transformed according to:

$$\left.\begin{aligned} F(\xi,\eta) &= \frac{\tilde{u}}{\tilde{u}_e}, \qquad \Theta(\xi,\eta) = \frac{\tilde{T}-\tilde{T}_e}{\tilde{T}_e} \\ V(\xi,\eta) &= \frac{2\xi}{\bar{\rho}_e\tilde{u}_e\mu_e r_o^{2j}}\left[F\left(\frac{\partial\eta}{\partial s}\right) + \frac{\bar{\rho}\tilde{v}r_o^j t^j}{\sqrt{2\xi}}\right] \\ K(\xi,\eta) &= \frac{k}{\tilde{u}_e^2} \qquad \hat{W}(\xi,\eta) = \frac{2\xi\omega}{\tilde{u}_e^2}, \qquad \hat{\mathcal{E}}(\xi,\eta) = \frac{2\xi\tilde{\epsilon}}{\tilde{u}_e^4} \end{aligned}\right\} \tag{2.43}$$

The transformed equations for the k-ω and k-ϵ models can then be expressed as follows.

$$2\bar{\xi}\frac{\partial F}{\partial\bar{\xi}} + \frac{\partial V}{\partial\eta} + F = 0 \tag{2.44}$$

$$2\bar{\xi}F\frac{\partial F}{\partial\bar{\xi}} + V\frac{\partial F}{\partial\eta} - \frac{\partial}{\partial\eta}\left[t^{2j}L\left(1+\bar{\mu}_T\right)\frac{\partial F}{\partial\eta}\right] + \bar{\beta}\left(F^2-\Theta-1\right) = 0 \tag{2.45}$$

$$\begin{aligned} 2\bar{\xi}F\frac{\partial\Theta}{\partial\bar{\xi}} + V\frac{\partial\Theta}{\partial\eta} - \frac{\partial}{\partial\eta}\left[t^{2j}L\left(\frac{1}{Pr_L}+\frac{\bar{\mu}_T}{Pr_T}\right)\frac{\partial\Theta}{\partial\eta}\right] \\ -\bar{\alpha}t^{2j}L\left(\frac{\partial F}{\partial\eta}\right)^2 - \frac{\bar{\alpha}}{\hat{\rho}_e\hat{\mu}_e\hat{r}_o^{2j}}\left(\mathcal{E}+\mathcal{E}_o\right) = 0 \end{aligned} \tag{2.46}$$

***k*-ω Model:**

$$\begin{aligned} 2\bar{\xi}F\frac{\partial K}{\partial\bar{\xi}} + V\frac{\partial K}{\partial\eta} - \frac{\partial}{\partial\eta}\left[t^{2j}L\left(1+\sigma^*\frac{\bar{\mu}_T}{W/\tilde{W}}\right)\frac{\partial K}{\partial\eta}\right] \\ +2\bar{\beta}FK - t^{2j}L\bar{\mu}_T\left(\frac{\partial F}{\partial\eta}\right)^2 + \frac{\beta^*}{\hat{\rho}_e\hat{\mu}_e\hat{r}_o^{2j}}WK = 0 \end{aligned} \tag{2.47}$$

$$\begin{aligned} 2\bar{\xi}F\frac{\partial W}{\partial\bar{\xi}} + V\frac{\partial W}{\partial\eta} - \frac{\partial}{\partial\eta}\left[t^{2j}L\left(1+\sigma\frac{\bar{\mu}_T}{W/\tilde{W}}\right)\frac{\partial W}{\partial\eta}\right] \\ +2(\bar{\beta}-1)FW - \alpha\frac{W}{K}t^{2j}L\bar{\mu}_T\left(\frac{\partial F}{\partial\eta}\right)^2 \\ +\sigma_d\frac{2\bar{\xi}\hat{\rho}_e}{\hat{\mu}_e\hat{\epsilon}^2}\frac{t^{2j}}{(1+\Theta)^2W}\frac{\partial K}{\partial\eta}\frac{\partial W}{\partial\eta} + \frac{\beta}{\hat{\rho}_e\hat{\mu}_e\hat{r}_o^{2j}}W^2 = 0 \end{aligned} \tag{2.48}$$

$$\bar{\mu}_T = \frac{2\bar{\xi}\hat{\rho}_e}{\hat{\mu}_e\hat{\epsilon}^2}\frac{K}{L(1+\Theta)^2\tilde{W}}, \qquad \mathcal{E} = \beta^* KW, \qquad \mathcal{E}_o = 0 \tag{2.49}$$

$$\tilde{W} = \max\left\{W,\ \frac{C_{lim}}{\sqrt{\beta_i^*/\alpha^*}}\frac{\sqrt{2\bar{\xi}}\hat{\rho}_e\hat{r}_o^j}{\hat{\epsilon}}\frac{t^j\partial F/\partial\eta}{(1+\Theta)}\right\} \tag{2.50}$$

***k*-ϵ Model:**

$$\begin{aligned} &2\bar{\xi}F\frac{\partial K}{\partial\bar{\xi}} + V\frac{\partial K}{\partial\eta} - \frac{\partial}{\partial\eta}\left[t^{2j}L\left(1+\frac{\bar{\mu}_T}{\sigma_k}\right)\frac{\partial K}{\partial\eta}\right] \\ &+2\bar{\beta}FK - t^{2j}L\bar{\mu}_T\left(\frac{\partial F}{\partial\eta}\right)^2 + \frac{1}{\hat{\rho}_e\hat{\mu}_e\hat{r}_o^{2j}}\left(\mathcal{E}+\mathcal{E}_o\right) = 0 \end{aligned} \tag{2.51}$$

$$\begin{aligned} &2\bar{\xi}F\frac{\partial\mathcal{E}}{\partial\bar{\xi}} + V\frac{\partial\mathcal{E}}{\partial\eta} - \frac{\partial}{\partial\eta}\left[t^{2j}L\left(1+\frac{\bar{\mu}_T}{\sigma_\epsilon}\right)\frac{\partial\mathcal{E}}{\partial\eta}\right] \\ &+2(2\bar{\beta}-1)F\mathcal{E} - C_{\epsilon 1}f_1\frac{\mathcal{E}}{K}t^{2j}L\bar{\mu}_T\left(\frac{\partial F}{\partial\eta}\right)^2 \\ &+\frac{C_{\epsilon 2}f_2}{\hat{\rho}_e\hat{\mu}_e\hat{r}_o^{2j}}\frac{\mathcal{E}^2}{K} - \Sigma = 0 \end{aligned} \tag{2.52}$$

$$\bar{\mu}_T = C_\mu f_\mu\frac{2\bar{\xi}\hat{\rho}_e}{\hat{\mu}_e\hat{\epsilon}^2}\frac{K^2}{L(1+\Theta)^2\mathcal{E}}, \qquad \mathcal{E}_o = \frac{A}{U_\infty^3}\frac{2\bar{\xi}}{\hat{u}_e^4}\epsilon_o, \qquad \Sigma = \frac{A^2}{U_\infty^4}\frac{(2\bar{\xi})^2}{\hat{\rho}_e\hat{u}_e^6\hat{\mu}_e\hat{r}_o^{2j}}E \tag{2.53}$$

The quantities $\bar{\alpha}$, $\bar{\beta}$ and L are defined by

$$\bar{\alpha} \equiv \frac{\tilde{u}_e^2}{c_p\tilde{T}_e}, \qquad \bar{\beta} \equiv \frac{2\xi}{\tilde{u}_e}\frac{d\tilde{u}_e}{d\xi}, \qquad L \equiv \frac{\bar{\rho}}{\bar{\rho}_e}\frac{\mu}{\mu_e} \tag{2.54}$$

and the following dimensionless quantities have been introduced:

$$\left.\begin{array}{lll} \bar{\xi} = \dfrac{\xi}{\rho_\infty U_\infty\mu_r A^{2j+1}}, & \hat{r}_o = \dfrac{r_o}{A}, & \hat{u}_e = \dfrac{\tilde{u}_e}{U_\infty} \\ \hat{\rho}_e = \dfrac{\bar{\rho}_e}{\rho_\infty}, & \hat{T}_e = \dfrac{\tilde{T}_e}{T_r}, & \hat{\mu}_e = \dfrac{\mu_e}{\mu_r} \\ \hat{\epsilon} = \sqrt{\dfrac{\mu_r}{\rho_\infty U_\infty A}}, & W = \dfrac{\hat{W}}{\rho_\infty\mu_r A^{2j}}, & \mathcal{E} = \dfrac{\hat{\mathcal{E}}}{\rho_\infty\mu_r A^{2j}} \end{array}\right\} \tag{2.55}$$

Finally, note that subscript ∞ denotes freestream flow condition, A is a reference length, T_r is the reference temperature defined as

$$T_r = U_\infty^2 / c_p \tag{2.56}$$

and μ_r is the value of μ for $T = T_r$.

Chapter 3

Program EDDY2C

3.1 Overview

The information in this chapter is the user's guide for Program **EDDY2C**, which is a two-dimensional/axisymmetric, compressible program for solving the mass-averaged Navier-Stokes equations that is included on the companion CD for *Turbulence Modeling for CFD*. It includes an overview of the program's operation along with technical details on the program's structure and operation. The software on the CD includes a program, **EDDY2C_DATA**, that accomplishes the following.

- Guides input-data preparation
- Executes Program **EDDY2C**
- Permits viewing and/or printing program output
- Creates a video and/or hardcopy plot that compares computed results with experimental data

Because Program **EDDY2C_DATA** is written in Microsoft Visual C++, it should run on personal computers with all versions of the Microsoft Windows operating system.

Program **EDDY2C** embodies the Baldwin-Lomax (1978) algebraic model and the Wilcox (2006) k-ω and Stress-ω modes with and without viscous modifications. This program has evolved over the past two decades and can thus be termed a mature software package. Many U. S. Government Agencies have contributed to development of the program originally created by Prof. R. W. MacCormack of Stanford University.

Additionally, important improvements have been made to this software package as a result of feedback from users. The author owes special thanks to Dr. G. Brereton of the University of Michigan whose personal efforts resulted in the addition of the option to use either USCS or SI units.

3.2 Getting Started Quickly

In order to acquaint yourself with Program **EDDY2C** and its input-data preparation utility, **EDDY2C_DATA**, and to confirm that the software is fully operational on your computer, perform the following steps.

1. Copy the following files from the CD to your working directory:

File Name	Directory	Function
eddy2c.exe	Executables	Executable Program **EDDY2C**
eddy2c_data.exe	Executables	Input-data preparation Program **EDDY2C_DATA**
eddy2c_plot.exe	Executables	Plotting Program **EDDY2C_PLOT**
bump.exe	Executables	Grid-generation Program **BUMP**
general.exe	Executables	Grid-generation Program **GENERAL**
step.exe	Executables	Grid-generation Program **STEP**
wedge.exe	Executables	Grid-generation Program **WEDGE**
eddybl.exe	Executables	Executable Program **EDDYBL**
eddybl_data.exe	Executables	Input-data preparation Program **EDDYBL_DATA**
eddybl_plot.exe	Executables	Plotting Program **EDDYBL_PLOT**
eddybl_start.exe	Executables	Initial-profile Program **EDDYBL_START**
Settles-20.zip	Input Data/Eddy2c	Input data for supersonic compression-corner flow

2. Extract the two input-data files **eddybl.dat** and **eddy2c.dat** from the "Zip" file **Settles-20.zip**. These data correspond to the Mach 2.79 flow into a 20^o compression corner experimentally investigated by Settles, Vas and Bogdonoff (1976). You can use these data files without modification to quickly determine that everything is operating properly, and to see how easy it is to use Program **EDDY2C**. Because the input-data preparation utility, **EDDY2C_DATA**, is menu driven, you will find that very little explanation of the program's operation is needed. After successfully completing this benchmark run, you can learn some of the more subtle features of Program **EDDY2C_DATA** by reading Section 3.3.

3. You will be using Program **EDDYBL** to generate mean-flow and turbulence profiles to specify conditions at the inlet plane. Even if you have not yet read Chapter 2, you can still perform this benchmark run. The first step is to run **EDDYBL_DATA**. Figure 3.1 shows the opening screen.

4. Click on the button with the label "Write Input-Data Files." This runs Program **EDDYBL_START**, which generates initial profiles that will be used by Program **EDDYBL**. The screen will appear as shown in Figure 3.2. After you press *ENTER*, control returns to the main menu. Press the ENTER key or close the window to return to the main menu.

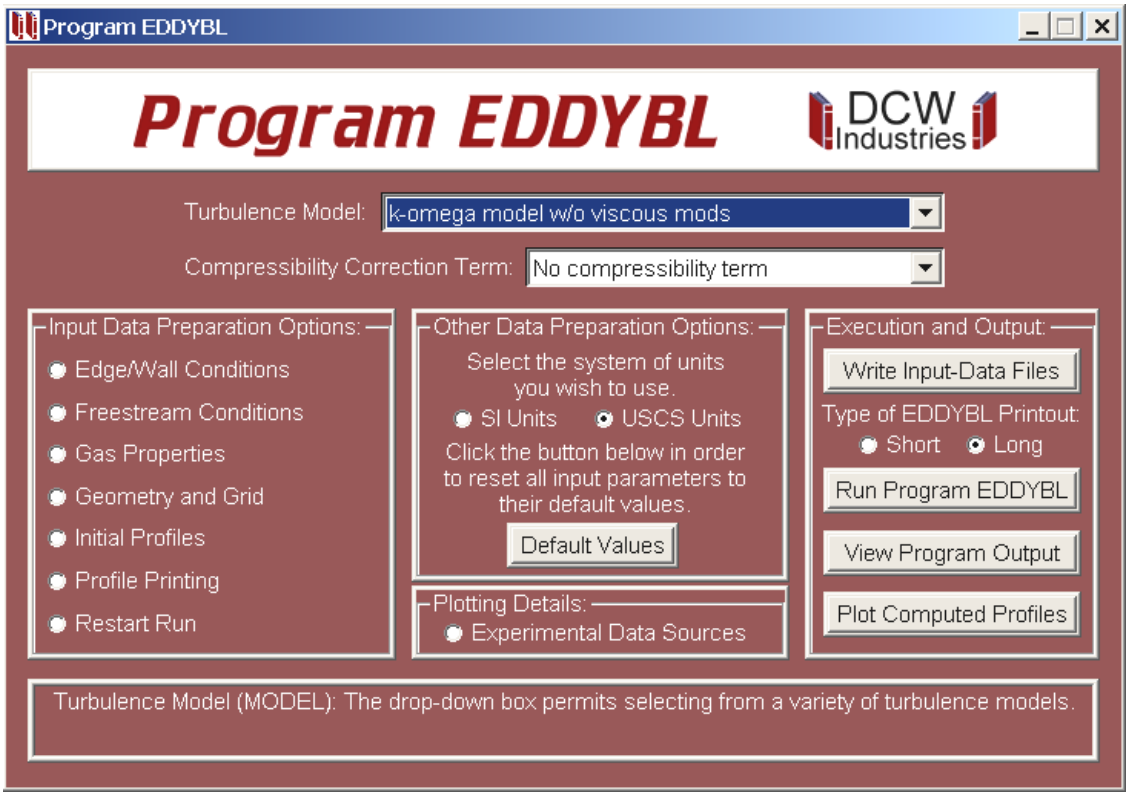

Figure 3.1: *Opening menu of Program* **EDDYBL_DATA**.

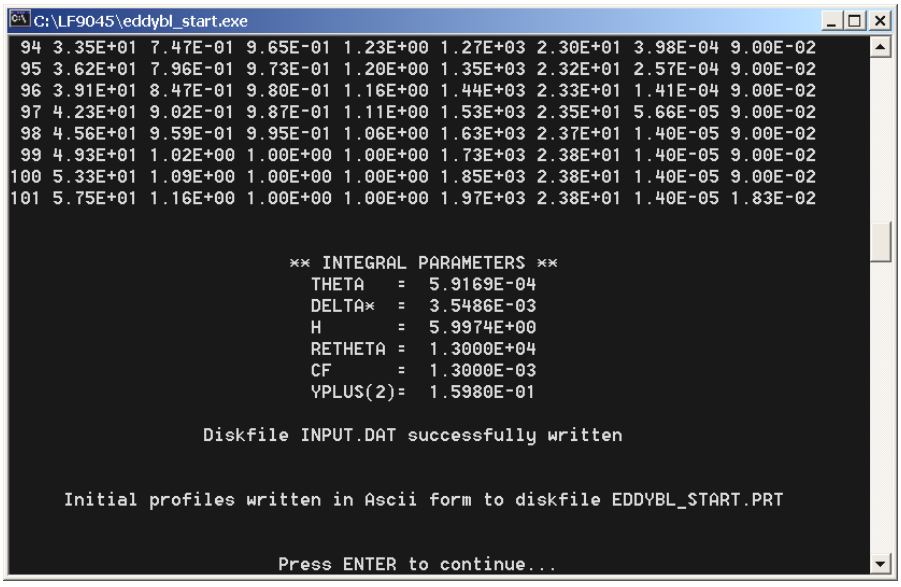

Figure 3.2: *Output from Program* **EDDYBL_START**.

5. Click on the button with the label "Run Program EDDYBL." This runs Program **EDDYBL**, and the screen will appear as shown in Figure 3.3. Press the Enter key or close the window to return to the main menu and then close the main menu to close **EDDYBL_DATA**. What you have done is generate initial profiles corresponding to an initial momentum-thickness Reynolds number $Re_\theta = 9.38 \cdot 10^4$. This is the value appropriate just upstream of the compression corner for this flow.

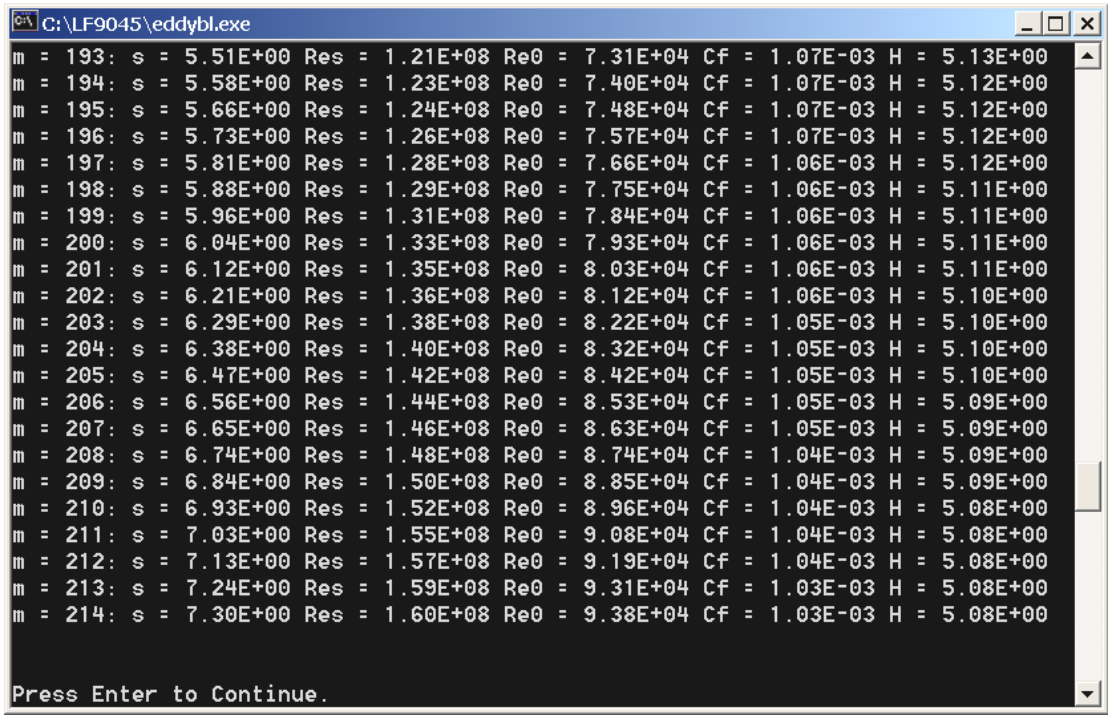

Figure 3.3: *Output from Program* **EDDYBL**.

6. Now, shift your focus to input-data preparation utility **EDDY2C_DATA**, which will guide the rest of the computation. Run **EDDY2C_DATA**. Figure 3.4 shows the opening screen.

7. Click on the button with the label "Generate Grid and Initial Conditions." This passes control to Program **WEDGE**, which generates a finite-difference grid and initial conditions for the flowfield. The screen will appear as shown in Figure 3.5. Press the ENTER key or close the window. You will be asked if you "would like to view the grid and initial conditions in printed form." Click on "No" to return to the main menu. At this point, you have prepared all of the input data needed to run Program **EDDY2C**.

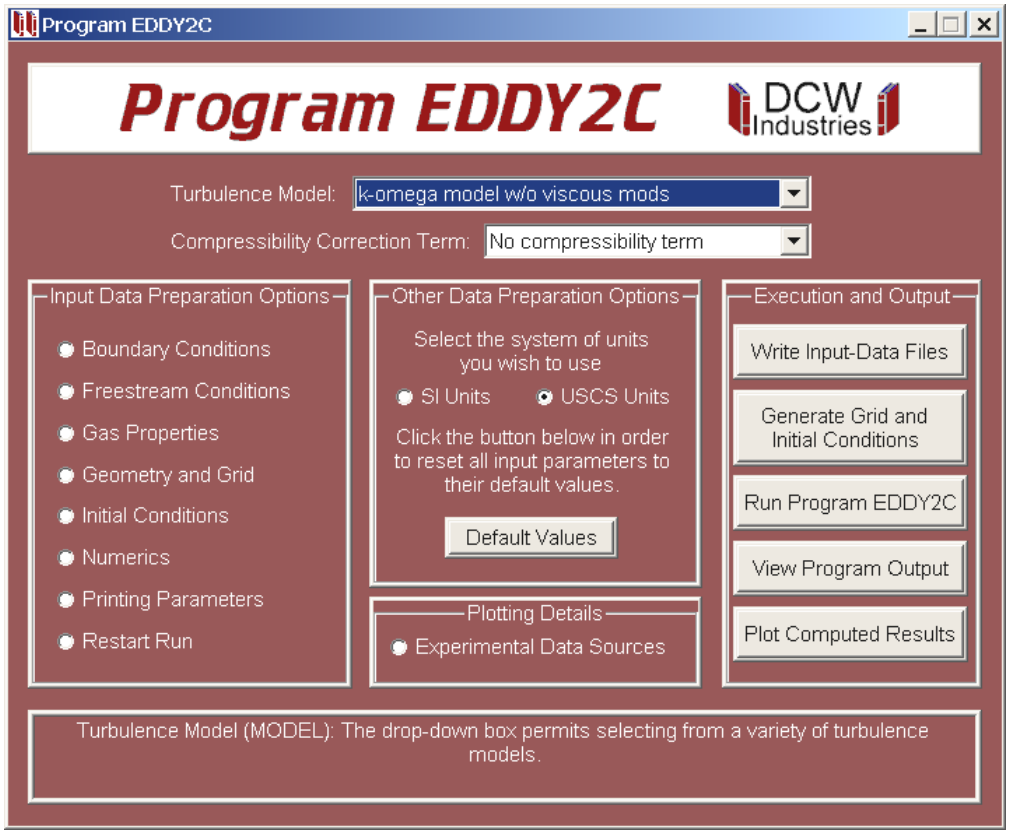

Figure 3.4: *Opening menu of Program* **EDDY2C_DATA**.

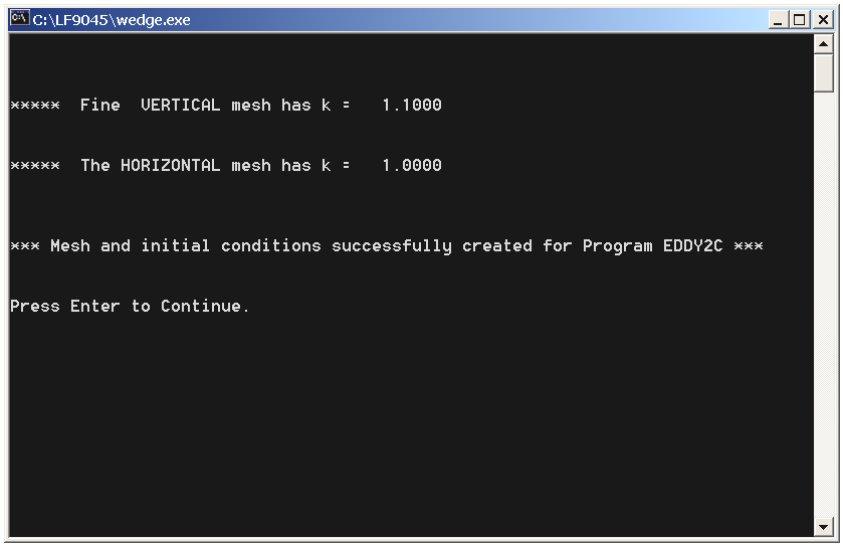

Figure 3.5: *Output from Program* **WEDGE**.

8. Click on the button with the label "Run Program EDDY2C." Control passes to Program **EDDY2C** and the run takes approximately 45 minutes on a 3-GHz Pentium-D microcomputer. As the run proceeds the screen will display the timestep number and maximum residual. Every 20 timesteps, the screen displays horizontal distance, surface pressure, skin friction, surface heat transfer rate, and y^+ at the first grid point above the surface. It also reports separation and reattachment points as well as the length of the separated region. When the run finishes, the screen will appear as shown in Figure 3.6. Press the Enter key or close the window to return to the main menu.

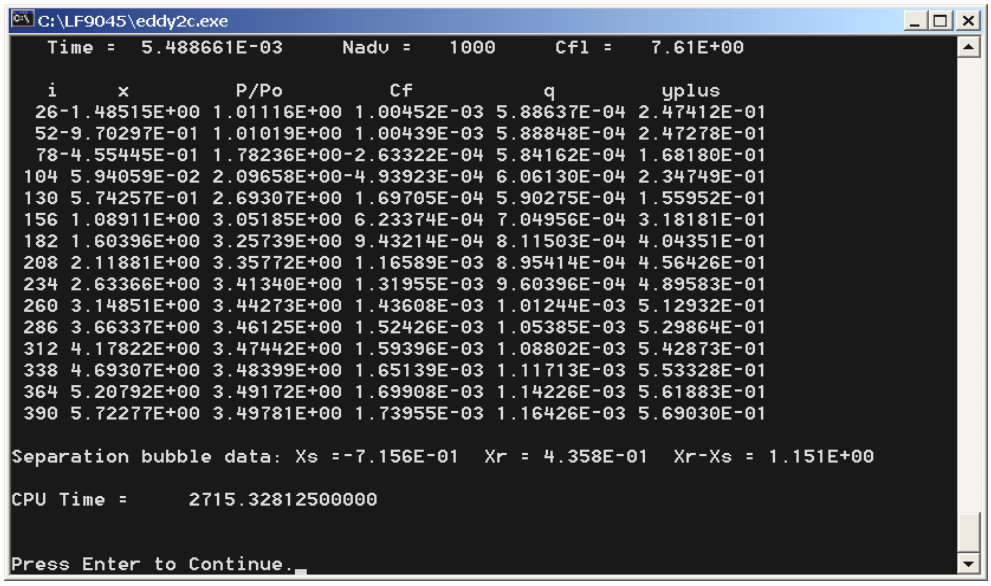

Figure 3.6: *Output from Program* **EDDY2C**.

9. After returning to the main menu, click on the "View Program Output" button to transfer control to Windows text editor **notepad.exe**. This permits you to view, print and even edit output file **eddy2c.prt**.

10. Click on the "Plot Computed Profiles" button to create a plot (see Figure 3.7). You can generate a hardcopy of the figure by clicking on the printer icon at the top of the screen.

And that's all there is to running Program **EDDY2C** for any of the input-data files contained in Subdirectory **Input Data/Eddy2c** on the companion CD. To run a general case, you will need to use the various menus attached to the main menu. The next section discusses what is contained in each of the nine menus.

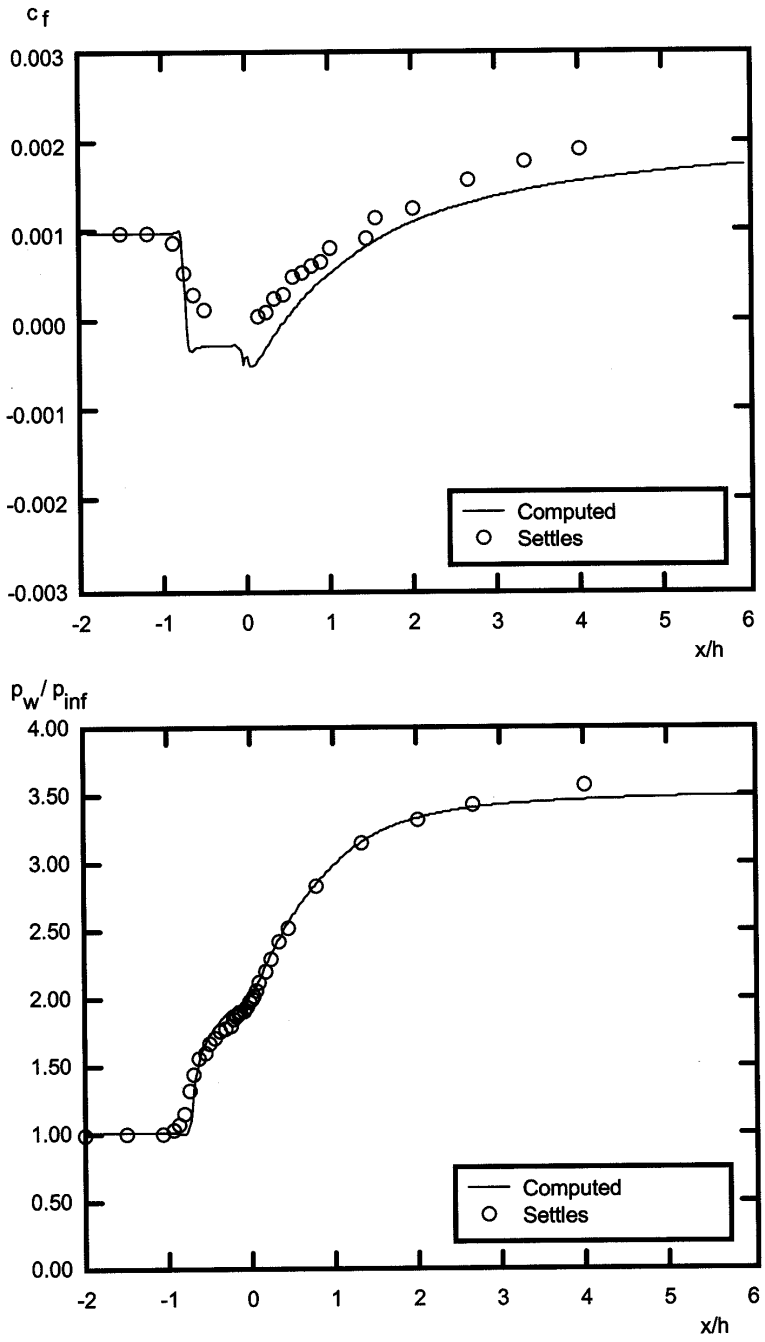

Figure 3.7: *Printed output from Program* **EDDY2C_PLOT**.

3.3 Navigating the Menus

This section explores, in detail, all salient features of the input-data preparation utility, **EDDY2C_DATA**. You will be guided through the various menus and, in the process, you will set up a shock-wave/boundary-layer interaction computation for a Mach 2.9 freestream with an adiabatic surface. The conditions correspond to the experiment conducted by Reda and Murphy (1972) and by Murthy and Rose (1978). For the case you will do, freestream conditions are:

Total pressure, p_{t_∞}	=	$6.8901 \cdot 10^5$ N/m^2
Total temperature, T_{t_∞}	=	291 K
Mach number, M_∞	=	2.9
Incident shock wave angle, Θ	=	31^o
Incident boundary-layer thickness, δ_o	=	1.63 cm
Incident boundary-layer Re_θ	=	$4.70 \cdot 10^4$

This flow has been computed with a 401 x 201 point finite-difference grid using the k-ω model and the length of the separation bubble was found to be $\Delta x/\delta_o = 4.645$ (see Subsection 5.8.7 of *Turbulence Modeling for CFD*). Your goal is to repeat the computation with a 201 x 101 grid and use Richardson extrapolation to estimate the length of the separation bubble for the continuum solution to the differential equations, and thereby place a bound on the error in the 401 x 201 point computation.

3.3.1 Opening Menu

To perform this exercise, delete any existing **eddy2c.dat** data file that might be in your directory. Although this is not generally necessary, for the purposes of this section it will be easier if you begin with no pre-existing input-data file. Also, copy the file **Reda-Murphy.zip** from the companion CD (it's in subdirectory **Input Data/Eddy2c**). Of course, aside from **Settles-20.zip**, you must also have the same files in your working directory that are listed in Step 1 of Section 3.2.

As in the benchmark run, you will again use Program **EDDYBL** to generate flow properties that will be used at the inlet plane. To do so, extract the file **eddybl.dat** from the Reda-Murphy "Zip" file. While you could also extract the file **eddy2c.dat** and simply modify the grid, it is more instructive to begin with no pre-existing **eddy2c.dat** file.

Now, run Program **EDDYBL_DATA**. As in the benchmark case (see Figures 3.1 and 3.2), click first on the "Write Input-Data Files" button to generate initial conditions for **EDDYBL**. Close the window and click on the "Run Program EDDYBL" button. Be sure to verify that the computed momentum-thickness Reynolds number is $Re_\theta = 4.70 \cdot 10^4$. Close Program **EDDYBL_DATA** as you are now finished using it for this application. You will use Program **EDDY2C_DATA** from this point on.

Run Program **EDDY2C_DATA**. The only thing you need to change on the opening menu is the type of units that will be used. Click on the "SI Units" radio button. When you have made this change, the opening menu will appear as in Figure 3.8.

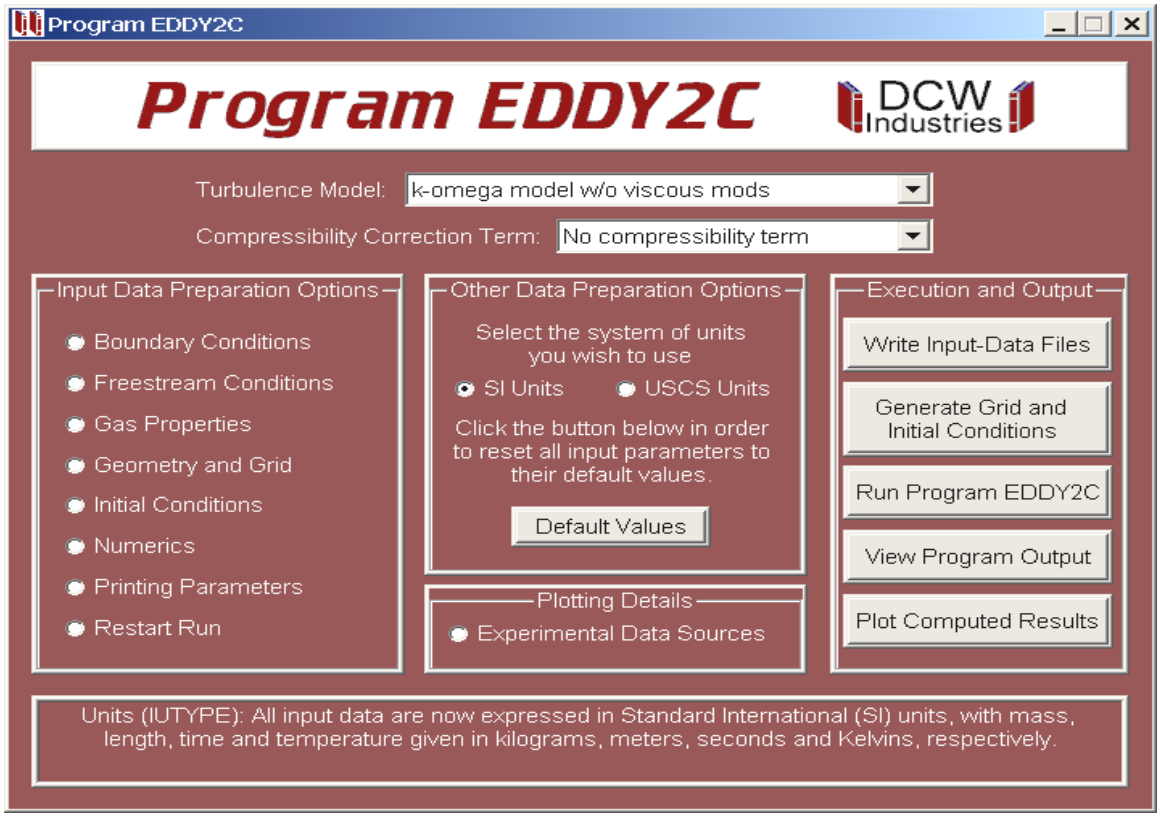

Figure 3.8: *Opening menu of Program* **EDDY2C_DATA** *after modification.*

3.3.2 Freestream Conditions

Next, click on the "Freestream Conditions" radio button, which will bring you to the menu that accepts input for freestream conditions. Click on the number in the "Total Pressure" edit box and change it to $p_{t_\infty} = 6.8901 \cdot 10^5$ N/m^2 using standard Windows editing techniques. Then, change the total temperature and Mach number to $T_{t_\infty} = 291$ K and $M_\infty = 2.9$, respectively. Finally, change the freestream turbulence intensity to 10^{-8}, which is the value used in the 401 x 201 point computation. Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm the last of your input-data changes. Figure 3.9 shows how the menu will look when you have finished making your changes.

Note that when you click on an input quantity, the box at the bottom of the menu describes the quantity you have clicked. Also, if you watch the right side of the screen, you will observe that each time you change freestream total pressure, total temperature or Mach number, the freestream static conditions change. Return to the main menu by closing this window.

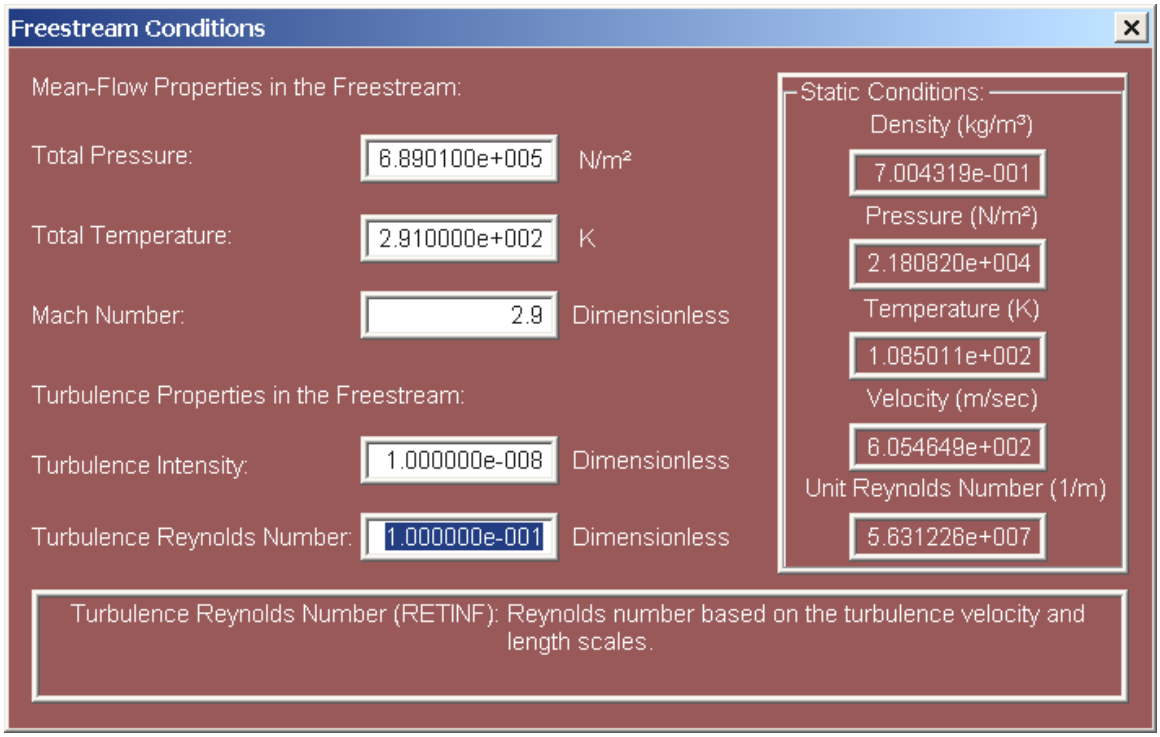

Figure 3.9: *Freestream-conditions menu of Program* **EDDY2C_DATA** *after modification.*

3.3.3 Geometry and Grid

Click on the "Geometry and Grid" radio button. Using this menu, you will set up a rectangular finite-difference grid with the following features. There will be 201 points in the horizontal direction and 101 points normal to the surface. The upstream boundary will be located at $x/\delta_o = -10$ and the downstream boundary at $x/\delta_o = 10$. The upper boundary of the grid will be at $y/\delta_o = 10$. There will be a fine mesh near the surface of thickness δ_o with 41 grid points and the first grid point above the surface will be located at $y = 2 \cdot 10^{-6}$ m. Note that the origin of the coordinate system lies at $x = y = 0$, which is the point that the shock would strike the plate in an inviscid flow. The dimensions of this grid match those of the 401 x 201 point calculation with grid-point spacing doubled.

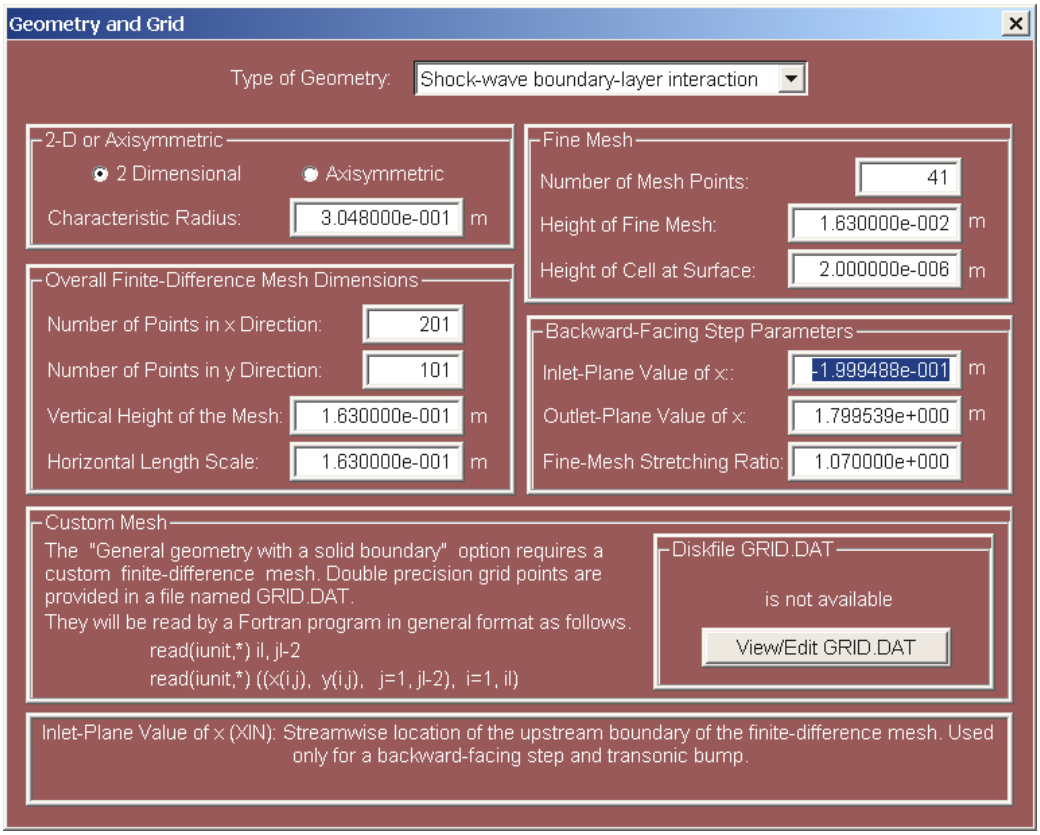

Figure 3.10: *Geometry and Grid menu of Program* **EDDY2C_DATA** *after modification.*

Click on the arrow at the right of the "Type of Geometry" drop-down box. Select the shock-wave boundary-layer interaction option.

In the "Overall Finite-Difference Mesh Dimensions" part of the menu, change the "Number of Points in x Direction" to 201, "Number of Points in y Direction" to 101, "Vertical Height of the Mesh" to 0.163 m and "Horizontal Length Scale" to 0.163 m. Note that the help box tells you that the latter quantity is the distance from the upstream boundary of the mesh to the shock-impingement point. Its definition depends upon the particular geometry selected.

In the "Fine Mesh" part of the menu, change the "Number of Mesh Points" to 41, "Height of Fine Mesh" to 0.0163 m and "Height of Cell at Surface" to $2 \cdot 10^{-6}$ m. Note that the fine-mesh option is available for most (but not all) of the geometries included in **EDDY2C_DATA**.

Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. Figure 3.10 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

3.3.4 Boundary Conditions

Click on the "Boundary Conditions" radio button. Now click on the "Adiabatic" radio button. Just below this button, you will find "Surface-Roughness Height." Change the value to 10^{-6} m, which matches the value used in the 401 x 201 point computation.

In the upper right-hand corner of the menu, change the "Trailing-Edge Point" to 201. Just below, in the part of the menu labeled "Shock-Wave or Compression-Corner Location," change the "Shock-Impingement Point' to 101 (thus putting it at the center of the mesh) and change the "Shock or Wedge Angle" to 31^{o}. Note that when you change the shock-impingement point, the value in the "Point where Corner Begins" edit box automatically changes to 0.

Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. Figure 3.11 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

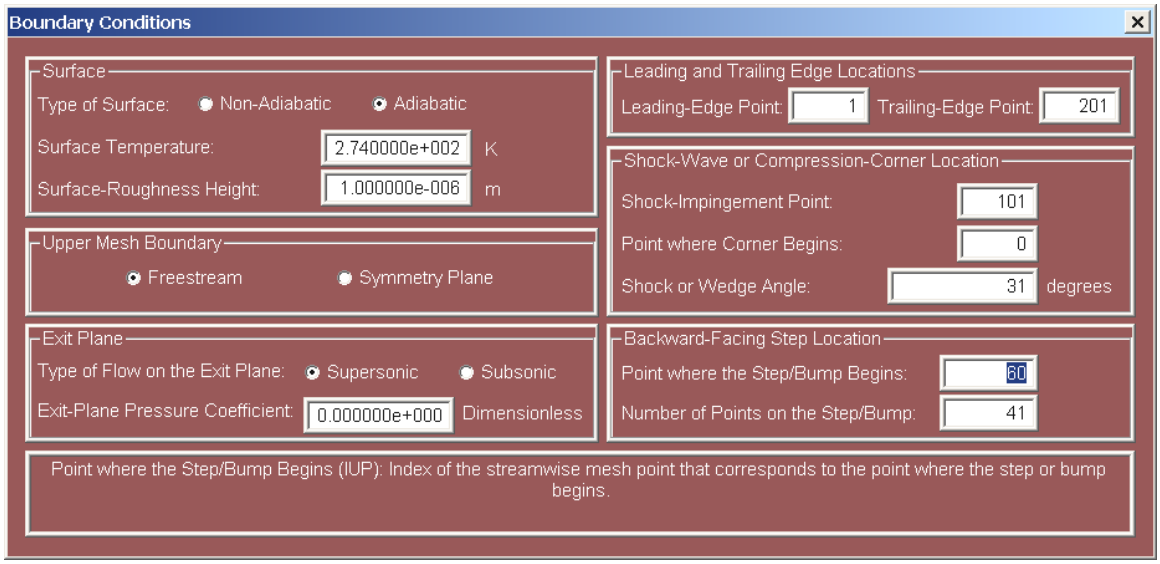

Figure 3.11: *Boundary-Conditions menu of Program* **EDDY2C_DATA** *after modification.*

3.3.5 Numerics

Click on the "Numerics" radio button. The only thing you need to change on this menu is the "Initial Value" of the CFL number. To insure numerical stability for this flow, reduce its value to 0.01. Note that when you change the its value, the "SUMMARY" below changes. It indicates that the CFL number will increase from 0.01 to 10 over the first 40 timesteps and remain constant from that point on.

Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your input-data change. Figure 3.12 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

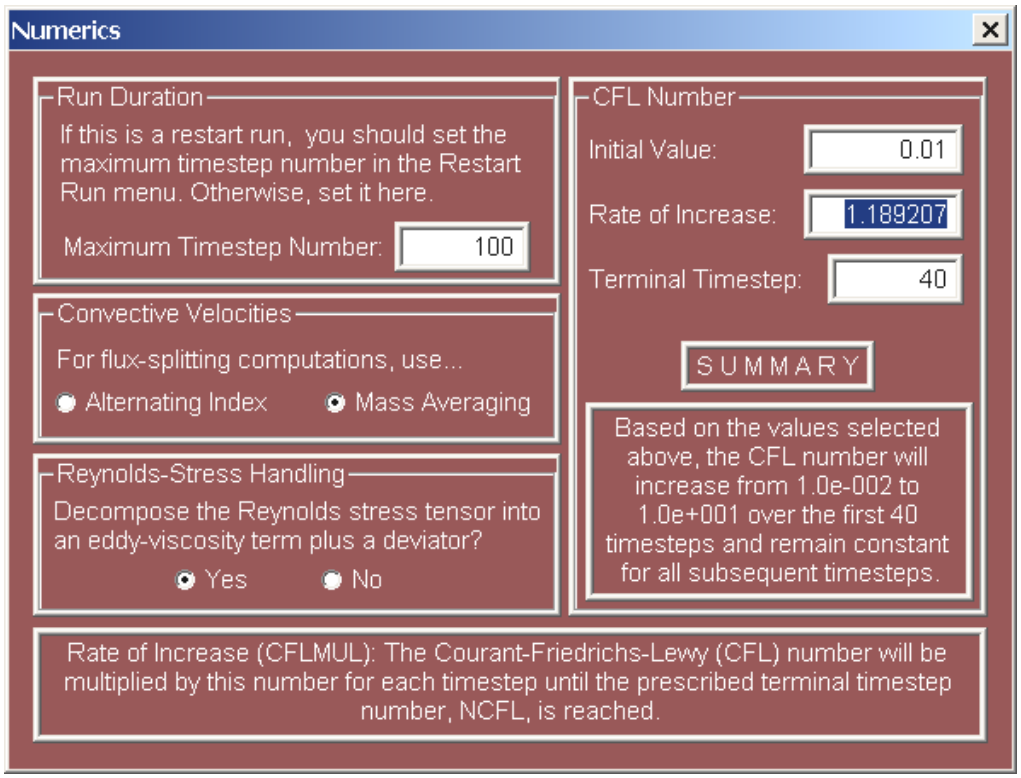

Figure 3.12: *Numerics menu of Program* **EDDY2C_DATA** *after modification.*

3.3.6 Printing Parameters

Click on the "Printing Parameters" radio button. All distances in your input-data file are expressed in meters. For the purpose of displaying printed and plotted results, it is customary for shock-wave/boundary-layer interactions to reference to the incident boundary-layer thickness, $\delta_o = 0.0163$ m. Thus, change the "Scaling Factor" to 61.35 m^{-1}, which is the reciprocal of δ_o. All displayed lengths will be multiplied by this scaling factor. Also, change the "Short-Print Modulus" to 20 so that surface conditions are displayed every 20 timesteps rather than every 10 steps.

Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. Figure 3.13 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

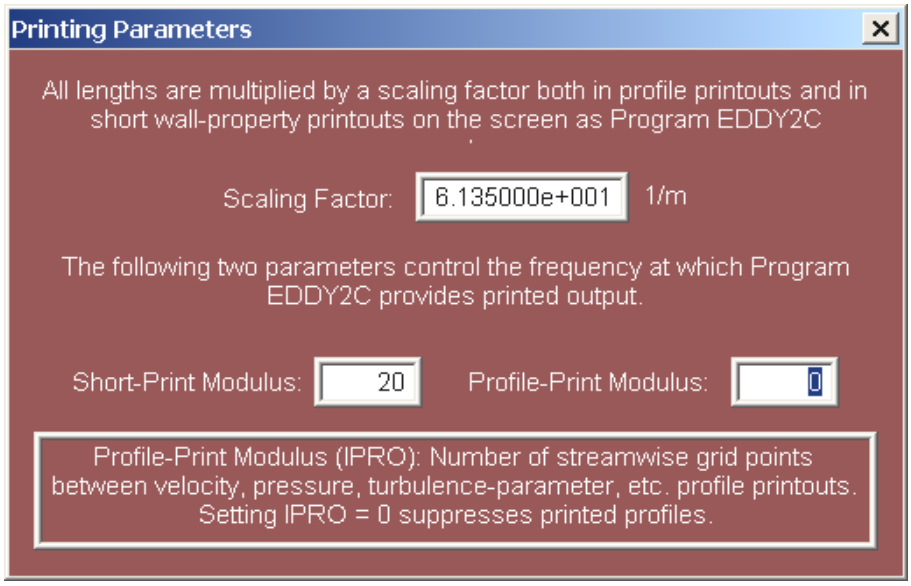

Figure 3.13: *Printing-Parameters menu of Program* **EDDY2C_DATA** *after modification.*

3.3.7 Experimental Data Sources

The final step you will take to set up this computation is to use the "Experimental Data Sources" menu. Click on its radio button, which lies in the bottom central part of the main menu. This menu permits including skin-friction and surface-pressure data that will be used in creating plots after the computation is done. This, of course, allows you to compare computed results with measurements. For the case at hand, we have data for both. Table 3.1 includes the data you will enter.

Table 3.1: *Experimental skin-friction and surface-pressure data.*

x	$10^3 c_f$	x	$10^3 c_f$	x	$10^3 c_f$	x	$10^3 c_f$
-0.1357	1.2121	-0.0595	0.2626	-0.0082	-0.4444	0.0468	0.5859
-0.1103	1.3131	-0.0415	-0.4646	0.0050	-0.4646	0.0669	1.3131
-0.0849	1.2929	-0.0288	-0.4040	0.0161	0.2424	0.0849	1.7980
-0.0706	1.0101	-0.0146	-0.3434	0.0357	0.3838	0.0981	2.0606
x	P_w/P_∞	x	P_w/P_∞	x	P_w/P_∞	x	P_w/P_∞
-0.0966	1.0032	-0.0680	1.7500	-0.0193	2.6104	0.0505	4.2175
-0.0786	1.0032	-0.0574	2.0260	-0.0082	2.9026	0.0659	4.4286
-0.0722	1.0844	-0.0463	2.1071	0.0108	3.4221	0.0870	4.6558
-0.0675	1.2468	-0.0362	2.3669	0.0315	3.8279	0.1108	4.8182

On the left-hand side of the screen, enter "Murthy and Rose" in the edit box labeled "Data Source." Next, enter the number 16 in the "Number of Points" edit box. Finally, enter the skin-friction data from Table 3.1. Note that the slider bar in the center of the menu permits scrolling up and down to make the 16 lines you need to fill in accessible.

Now, move to the right-hand side of the screen and enter "Reda and Murphy" in the "Data Source" edit box. Next, enter the number 16 in the "Number of Points" edit box. Finally, enter the surface-pressure data from Table 3.1.

Be sure to press the ENTER key, the TAB key or click on another input quantity to confirm your final input-data change. When you have finished entering all of the data points, your screen will appear as in Figure 3.14. Return to the main menu by closing this window.

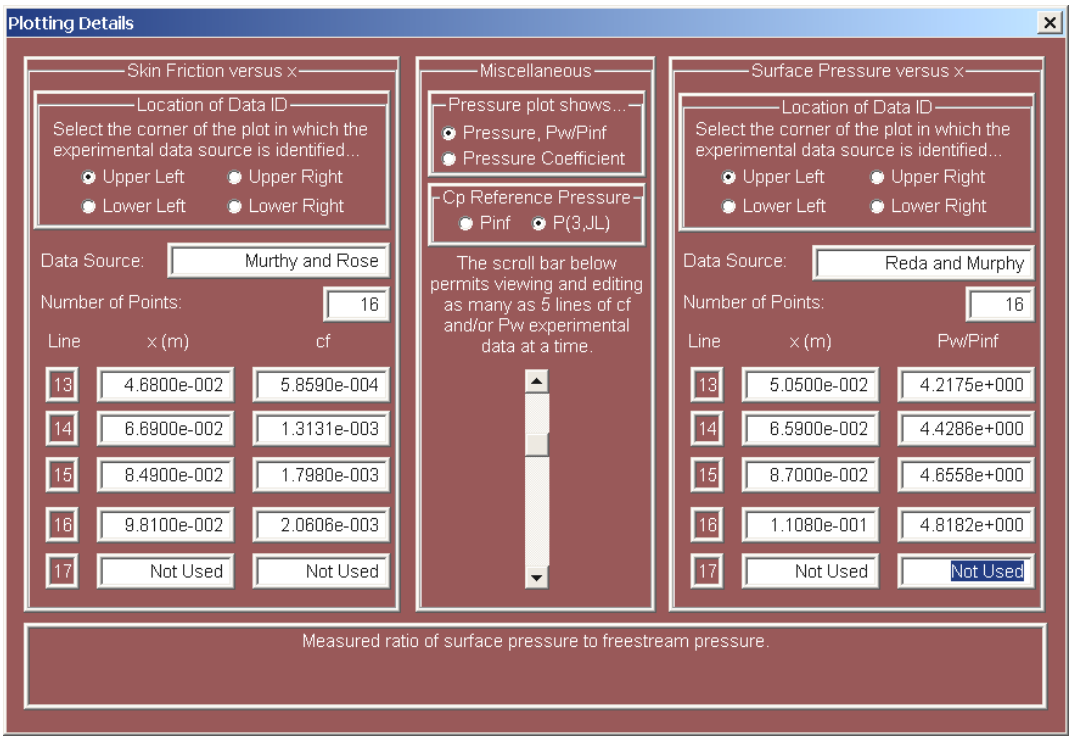

Figure 3.14: *Experimental-Data menu of Program* **EDDY2C_DATA** *after modification.*

At this point, you have made all of the input-data changes needed to perform the computation. As with the benchmark case of Section 3.2, you must first click the "Generate Grid and Initial Conditions" button. This passes control to Program **WEDGE**, which generates a finite-difference grid and initial conditions for the flowfield. The screen will appear as shown in Figure 3.5. Press the ENTER key or close the window. You will be asked if you "would like to view the grid and initial conditions in printed form." Click

on "No" to return to the main menu. Click the "Run Program EDDY2C" button. The run takes about a minute on a 3-GHz Pentium-D microcomputer. If you have made all of the input-data entries correctly, the screen should appear as in Figure 3.15.

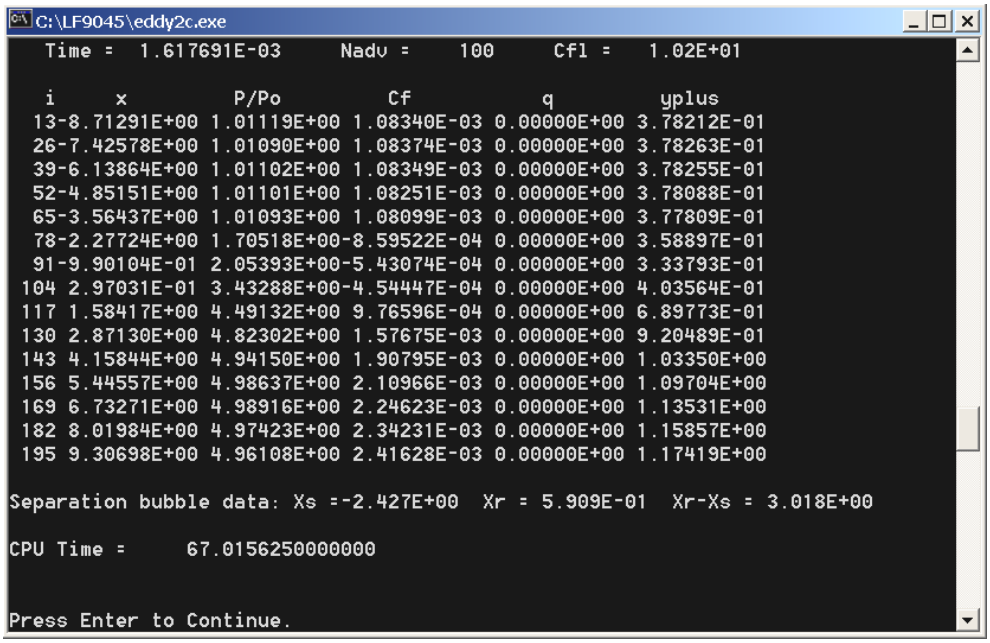

Figure 3.15: *Output from Program* **EDDY2C**.

3.3.8 Restart Run

There are a variety of reasons why it is desirable to be able to continue a Navier-Stokes computation from the point where it was terminated. For the computation just completed, inspection of the separation-bubble length, for example, indicates that it is changing and the computation clearly has not reached a steady state. Program **EDDY2C_START** provides the option to continue the computation from where you stopped it through its **Restart Run** option.

Close the output window to return to the main menu and click on the "Restart Run" radio button. First, change the "Maximum Timestep Number" to 600. Then click on the "Rename OUTPUT2C.DAT" button. Program **EDDY2C_DATA** then copies **input2c.dat** (which is the file created by **WEDGE**) to **input2c.bak** and renames **output2c.dat** as **input2c.dat**. The final output of your original run becomes input for the restart run. Figure 3.16 shows how the menu will look when you have finished making your changes. Return to the main menu by closing this window.

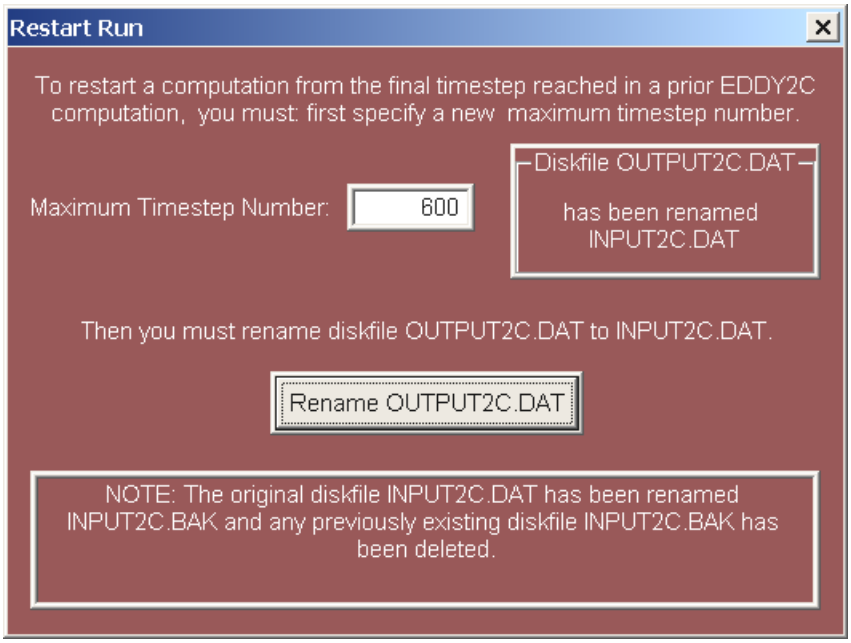

Figure 3.16: *Restart-Run menu of Program* **EDDY2C_DATA** *after modification.*

At this point, input-data files **eddy2c.dat** and **input2c.dat** have been modified as needed to continue your run from where you left off. Run Program **EDDY2C** again by clicking the "Run Program EDDY2C" button. Do not click the "Generate Grid and Initial Conditions" button — doing so would overwrite the file **input2c.dat**. The computation takes approximately 5 minutes on a 3-GHz Pentium-D microcomputer. If you have made no errors the screen will be as in Figure 3.17.

You can now compare your computed skin friction and surface pressure with the experimental data you entered. To do this, press the ENTER key or close the window to return to the main menu. Now, click on the "Plot Computed Profiles" button to display a plot of your computed results. Scroll down to display the surface-pressure profile. Your video display should appear as in Figure 3.18. You can generate hardcopy plot by clicking on the printer icon on the task bar above the plot.

To conclude the exercise, you now have sufficient information to estimate the continuum value of the separation-bubble length, Δx. For your 201 x 101 point mesh the length is $4.518\delta_o$. For the 401 x 201 point mesh, you know that $\Delta x = 4.645\delta_o$. Using Richardson extrapolation, there follows

$$\Delta x_\infty = \frac{4}{3}\Delta x_{401} - \frac{1}{3}\Delta x_{201} = 4.687\delta_o \tag{3.1}$$

Therefore the 401 x 201 point computation has a separation bubble length that is within 0.9% of the exact solution to the differential equations.

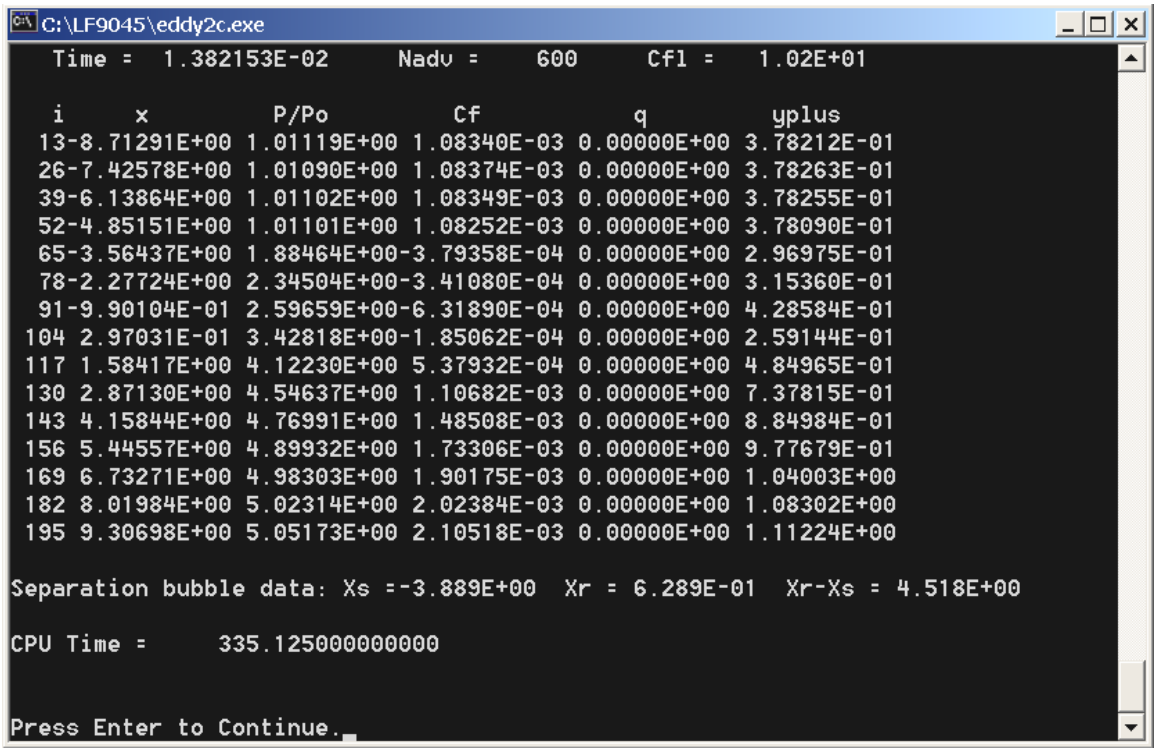

Figure 3.17: *Output from Program* **EDDY2C**.

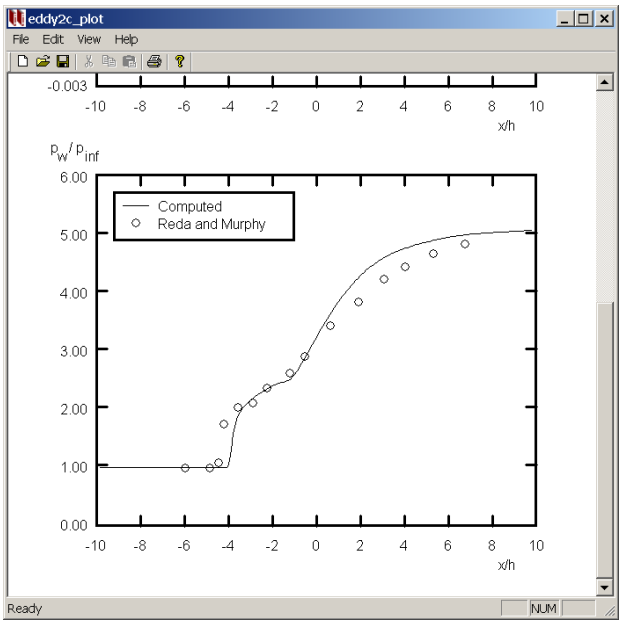

Figure 3.18: *Output from Program* **EDDY2C_PLOT**.

3.3.9 Gas Properties and Initial Conditions

Program **EDDY2C_DATA** has two additional menus that you did not use in this exercise. They are the "Gas Properties" menu (Figure 3.19) and the "Initial Conditions" menu (Figure 3.20). Note that the Gas Properties menu shown corresponds to the **EDDY2C_DATA** default selection of USCS units, not the run you just completed. As with all menus, the values and quoted units are automatically converted when units are changed on the opening menu of Program **EDDY2C_DATA**.

- The "Gas Properties" menu includes several thermodynamic properties and the turbulent Prandtl number, Pr_T. There is one relatively subtle feature regarding the viscosity law. If you wish to use a power-law viscosity relationship rather than the Sutherland law, then you are saying

$$\mu = \mu_r T^{\omega} = \mu_r \frac{T^{1+\omega}}{S+T} \qquad \text{where} \qquad S = 0 \tag{3.2}$$

  Thus, the input parameters are *SU* = 0, *VISCON* = μ_r and *VISPOW* = $1 + \omega$.

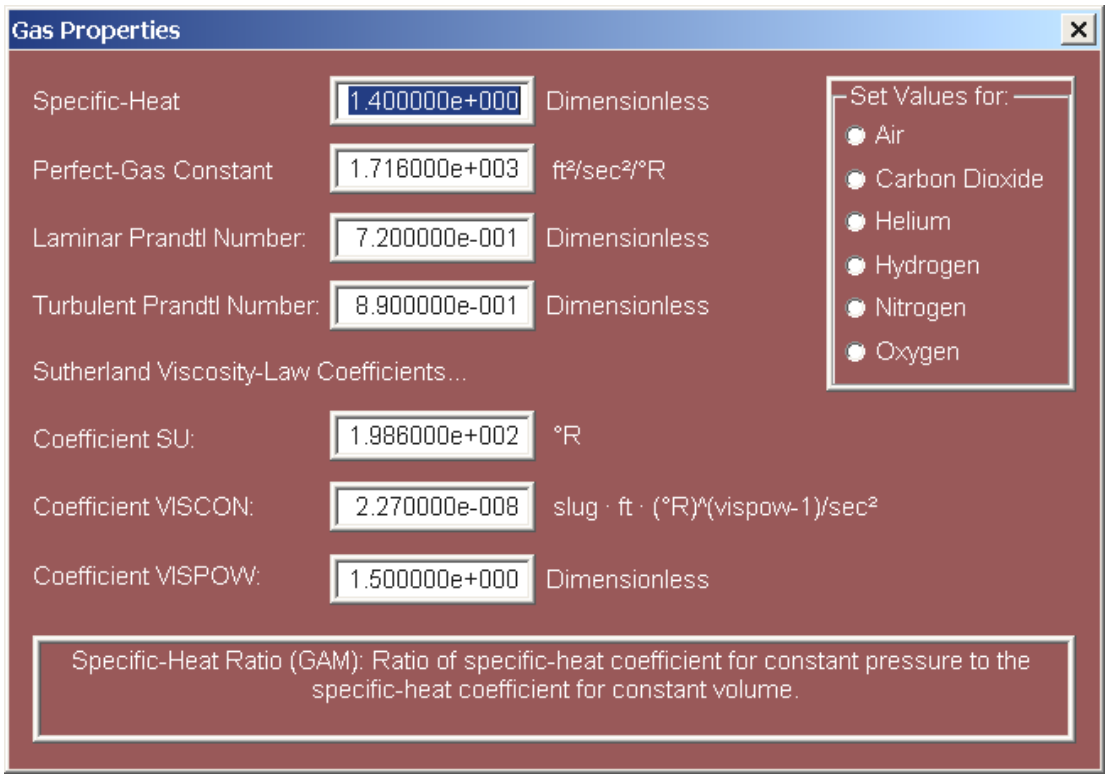

Figure 3.19: *Gas-Properties menu of Program* **EDDY2C_DATA**.

- The "Initial Conditions" menu permits selecting either uniform inlet conditions or initial conditions from a program such as **EDDYBL**. It also permits viewing the file that includes the initial conditions, viz., **profil2c.dat**.

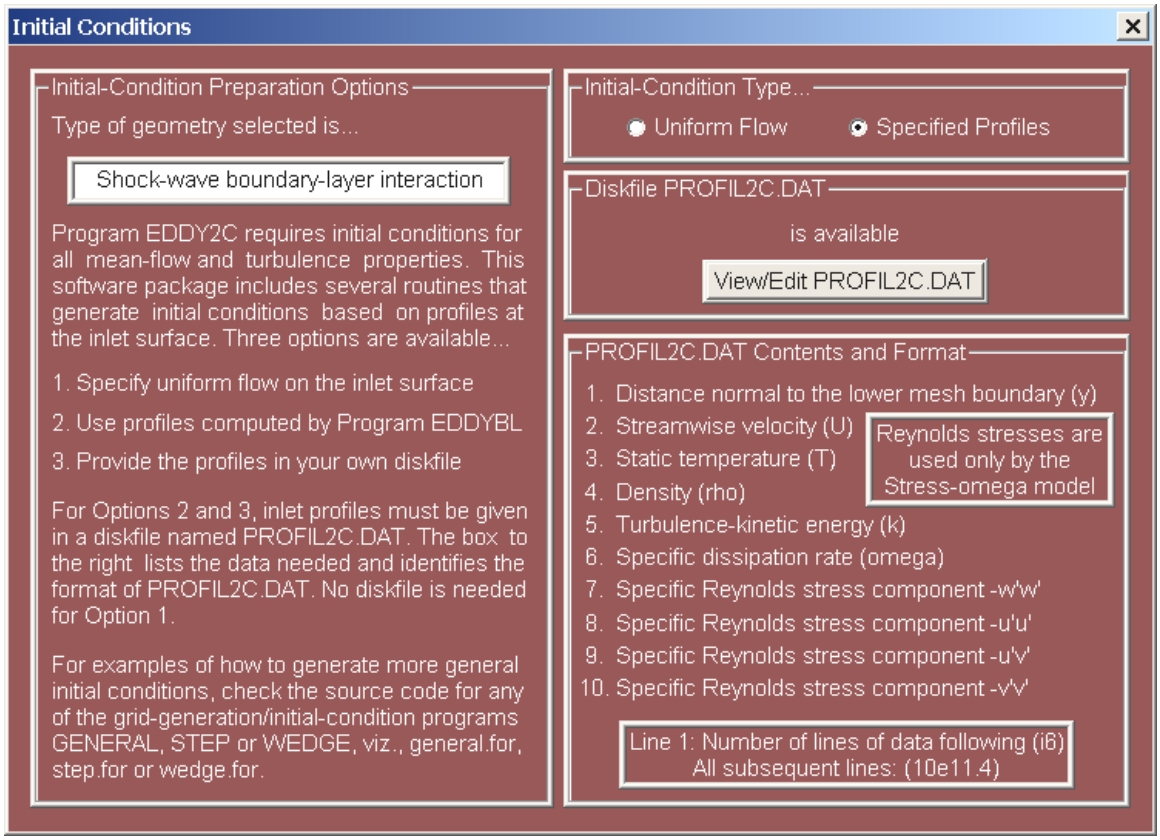

Figure 3.20: *Initial-Conditions menu of Program* **EDDY2C_DATA**.

3.4 Program Details

The companion CD includes Fortran source code for Program **EDDY2C** and its grid-generation utilities, **BUMP**, **GENERAL**, **STEP** and **WEDGE**. All can be compiled and run on virtually any computer with a Fortran compiler. The CD also includes executable input-data preparation and plotting programs designed for use on a personal computer with any version of the Microsoft Windows operating system.

3.4.1 Files Included on the CD

There are 14 files included on the companion CD for the programs described in this chapter. Programs **EDDY2C_DATA** and **EDDY2C_PLOT** have been written in Microsoft Visual C++, while Programs **EDDY2C**, **BUMP**, **GENERAL**, **STEP** and **WEDGE** are Fortran programs. The companion CD includes source code for the latter five programs. The files on the CD are as follows.

Table 3.2: *Program* **EDDY2C** *files on the CD.*

File Name	Directory	Function
eddy2c.exe	Executables	Executable Program **EDDY2C**
eddy2c_data.exe	Executables	Input-data preparation Program **EDDY2C_DATA**
eddy2c_plot.exe	Executables	Plotting Program **EDDY2C_PLOT**
bump.exe	Executables	Grid-generation Program **BUMP**
general.exe	Executables	Grid-generation Program **GENERAL**
step.exe	Executables	Grid-generation Program **STEP**
wedge.exe	Executables	Grid-generation Program **WEDGE**
come2c	Source Code	File included by all five Fortran Programs
cpuid	Source Code	File included by all five Fortran Programs
eddy2c.for	Source Code	Fortran source code for Program **EDDY2C**
bump.for	Source Code	Fortran source code for Program **BUMP**
general.for	Source Code	Fortran source code for Program **GENERAL**
step.for	Source Code	Fortran source code for Program **STEP**
wedge.for	Source Code	Fortran source code for Program **WEDGE**

The executable files for the five Fortran programs have been compiled using the Lahey Fortran-90 compiler. The programs achieve optimum performance in double precision accuracy. Also, for compatibility with the Visual C++ interface, they are compiled with the Lahey "Windows Console" option. The specific commands used in compiling and linking the programs are as follows.

```
lf90 progname -dbl -bind -winconsole
```

Additionally, Subdirectory **Input Data/Eddy2c** of the companion CD includes input-data files for many of the Navier-Stokes computations described in *Turbulence Modeling for CFD*. All of the files are provided in "Zip" format. Each file contains two input-data files named **eddy2c.dat** and **eddybl.dat**. Some also contain a file named **grid.dat**. Table 3.3 lists all of the files.

Table 3.3: **EDDY2C** *input-data files on the CD.*

File Name	Application
Driver-Cylinder.zip	Incompressible, separated cylinder flow [Driver (1991)]
Driver-Seegmiller.zip	Incompressible, High Re backstep [Driver and Seegmiller (1985)]
Holden.zip	Mach 11.3, shock-wave/boundary-layer interaction [Holden (1978)]
Johnson-Bump.zip	Mach 0.875, cylindrical bump [Bachalo and Johnson (1979)]
Jovic-LowRe.zip	Incompressible, Low Re backstep [Jovic and Driver (1994)]
Kussoy-Horstman.zip	Mach 7.05, cylinder/35^o flare [Kussoy and Horstman (1989)]
Reda-Murphy.zip	Mach 2.9, shock-wave/boundary-layer interaction [Reda and Murphy (1972)]
Samimy.zip	Mach 2.07, backstep [Samimy, Petrie and Addy (1985)]
Settles-20.zip	Mach 2.79, 20^o compression corner [Settles et al. (1976)]
Settles-24.zip	Mach 2.84, 24^o compression corner [Settles et al. (1976)]

3.4.2 Fortran Portability

In the interest of portability, **EDDY2C**, **BUMP**, **GENERAL**, **STEP** and **WEDGE** have been written so that they run on virtually any computer with a Fortran compiler. The programs all use an **include** file named **come2c** that includes all of the common blocks defining program arrays and other key program variables.

All five programs also 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, Pentium)
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 Appendix A for a detailed explanation of what is required as well as a listing of subroutine **NAMSYS**.

3.4.3 Program Input

Program **EDDY2C** requires two primary input-data files, **eddy2c.dat** for general program parameters and **profil2c.dat** for flow properties at the inlet plane. It also requires information about the finite-difference grid. This subsection focuses on **eddy2c.dat** and **profil2c.dat**. The next subsection deals with finite-difference grid issues.

EDDY2C_DATA creates a disk file named **eddy2c.dat**. The format for the quantities included varies. In the first section of the file, the format for integer quantities is (1x,a12,i6) while the format for floating-point quantities is (1x,a12,e13.6). The (1x,a12) permits entering the variable name and an equal sign. Input lines for the Mach 2.79 20° compression-corner benchmark case (see Section 3.2), for example, appear as follows.

```
2         12 15  19     26  <-- Column number
|          |  |   |      |
IUTYPE     =      0
CFL        =  1.000000e-01
CFLMUL     =  1.189207e+00
DY0        =  4.000000e-06
GAM        =  1.400000e+00
HFM        =  8.200000e-02
HMAX       =  4.920000e-01
PEXIT      =  0.000000e+00
PINF       =  5.430089e+02
           .
           .
           .
GAMMAH     =  5.014354e-01
CC1        =  1.800000e+00
XIS        =  0.000000e+00
XMT0       =  0.000000e+00
XIN        = -2.000000e-02
XOUT       =  1.600000e-01
RATEY      =  1.070000e+00
IUP        =     60
ISTEP      =     41
ICS        =      1
```

The second part of **eddy2c.dat** includes experimental-data information in mixed format as shown below.

```
1         12 15 17  22  <-- Column number
|          |  | |    |
******
 NCF       =   21
-1.2300e-01 1.0000e-03
-9.6760e-02 1.0000e-03
-7.1340e-02 8.9000e-04
-6.1500e-02 5.7000e-04
            .
            .
            .
 1.2714e-01 1.1600e-03
 1.6336e-01 1.2600e-03
 2.1729e-01 1.5800e-03
 2.7200e-01 1.7900e-03
 3.2594e-01 1.9200e-03
             Settles
 IPOSCF    =    3
 NPW       =   35
-1.6400e-01 9.9000e-01
-1.2300e-01 1.0000e+00
-8.6920e-02 1.0000e+00
-7.7080e-02 1.0300e+00
            .
            .
            .
 6.3180e-02 2.8300e+00
 1.0865e-01 3.1500e+00
 1.6336e-01 3.3200e+00
 2.1729e-01 3.4300e+00
 3.2594e-01 3.5700e+00
             Settles
 IPOSPW    =    3
```

If nonuniform initial profiles are used to specify flow properties at the inlet plane, they must be provided in a disk file named **profil2c.dat**. Program **EDDYBL** (see Chapter 2) creates such a file. The first line of the file **profil2c.dat** contains the number of mesh

points normal to the surface, *iedge*. The format for this line is (i6). The remainder of the file consists of *iedge* lines of data, format (10e12.4), containing the following boundary-layer profile data, with quantities written on each line in the order listed. Note that for the k-ϵ model, the specific dissipation rate, ω, is defined by

$$\omega = \frac{\epsilon}{C_\mu k} \tag{3.3}$$

The Reynolds stresses are used only by the Stress-ω model. Table 3.4 lists the quantities in **profil2c.dat**, including their dimensions for USCS and SI units. The contents of **profil2c.dat** can be viewed from the "Initial Conditions" menu of Program **EDDY2C_DATA** (see Figure 3.20).

Table 3.4: *Contents of input-data file* **profil2c.dat**.

Quantity	Description	Dimensions
y	Distance normal to the surface	ft (m)
$\tilde{u}$	Horizontal velocity	ft/sec (m/sec)
$\tilde{T}$	Temperature	$^\circ$R (K)
$\bar{\rho}$	Density	slug/ft^3 (kg/m^3)
k	Turbulence kinetic energy	ft^2/sec^2 (m^2/sec^2)
ω	Specific dissipation rate	sec^{-1} (sec^{-1})
$-\overline{w'w'}$	Specific Reynolds zz-normal stress	ft^2/sec^2 (m^2/sec^2)
$-\overline{u'u'}$	Specific Reynolds xx-normal stress	ft^2/sec^2 (m^2/sec^2)
$-\overline{u'v'}$	Specific Reynolds shear stress	ft^2/sec^2 (m^2/sec^2)
$-\overline{v'v'}$	Specific Reynolds yy-normal stress	ft^2/sec^2 (m^2/sec^2)

3.4.4 Grid Generation

There are four grid-generation programs on the companion CD, all of which can be executed from **EDDY2C_DATA**. The programs and the geometry to which each applies are as follows.

- **BUMP** generates a grid for the Bachalo-Johnson (1979) transonic bump flow.
- **GENERAL** reads independently-created grid points from a disk file.
- **STEP** generates a grid for a backward-facing step.
- **WEDGE** generates a grid for flow into a compression corner and for a shock-wave/boundary-layer interaction.

In addition to generating the grid, these programs also define initial and boundary conditions throughout the computational domain. With the exception of Program **GENERAL**, input-data files **eddy2c.dat** and **profil2c.dat** contain all of the information needed to run these programs.

Program BUMP: The flow of interest is a "bump" fared onto the surface of a cylinder. Using the shape of the bump, this program generates a grid consisting of vertical lines (constant x) with points fared between the bump and the horizontal upper surface. Figure 3.21 schematically depicts the structure of the grid.

Streamwise grid-point spacing is constant along the bump. Points between the upstream boundary and bump leading edge are arranged according to a geometric progression and similarly for points between the bump trailing edge and the downstream boundary.

Vertical grid points are placed in a fine mesh near the surface to facilitate accurate resolution of the boundary layer. Grid-point spacing follows a geometric progression with the point closest to the surface specified by the user. The balance of the points between the fine mesh and upper mesh boundary are also arranged in a geometric progression based on the computed cell size just below the fine-grid/outer-grid interface.

eddy2c.dat contains all of the information needed to define the grid. Table 3.5 lists the data **BUMP** uses and the **EDDY2C_DATA** menu where the quantities are located.

Table 3.5: *Input data used by Program* **BUMP**.

Quantity	Description	**EDDY2C_DATA** Menu
dy0	Height of the cell nearest the surface	Geometry and Grid
hfm	Height of the fine mesh	Geometry and Grid
hmax	Vertical height of the mesh	Geometry and Grid
il	Number of points in the x direction	Geometry and Grid
istep	Number of points on the bump	Boundary Conditions
iup	Point where the bump begins	Boundary Conditions
jl	Number of points in the y direction	Geometry and Grid
jlfm	Number of points in the fine mesh	Geometry and Grid
rl	Length of the mesh	Geometry and Grid
xin	Value of x on the inlet plane	Geometry and Grid

When you click on the "Generate Grid and Initial Conditions" button on the opening menu of **EDDY2C_DATA**, control passes to **BUMP** when the transonic-bump geometry has been selected. **BUMP** creates binary disk file **input2c.dat**, which is used by Program **EDDY2C** to define the grid and initial-flowfield properties. When it completes its operations, **BUMP** reports the computed geometric progression ratios for the parts of the grid where they are used and confirms that the grid and initial conditions have been successfully created.

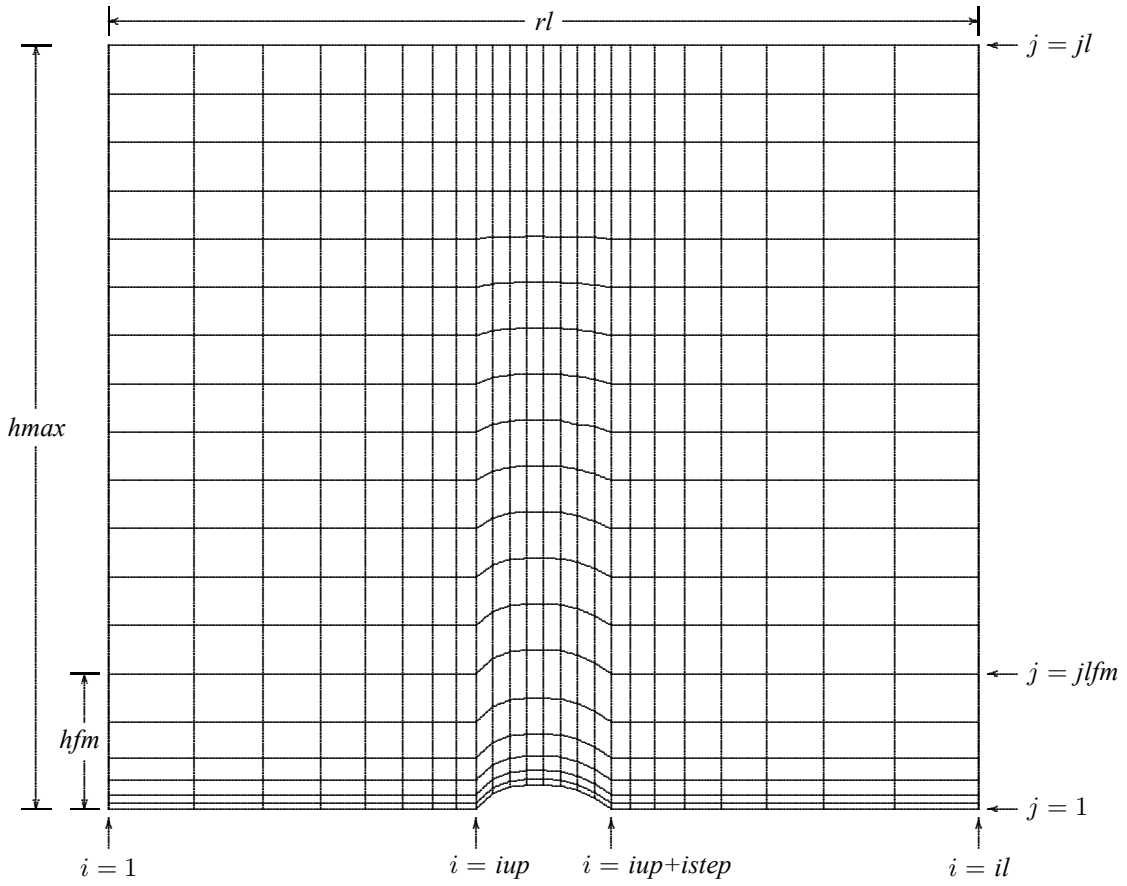

Figure 3.21: *Finite-difference grid schematic for Program* **BUMP** *(not to scale). Horizontal distance is x, vertical distance is y and the origin lies at the leading edge of the bump.*

Program GENERAL: This program reads finite-difference grid points from a user-supplied file named **grid.dat**. This permits using third-party grid-generation software with **EDDY2C**. The grid points must be exactly coincident with all mesh boundaries. Because velocities are defined at cell centers in **EDDY2C**, Program **GENERAL** adds two additional phantom cells at the upper and lower boundaries. So, if your custom grid has N points in the y direction, you must specify $jl = N + 2$ in **EDDY2C_DATA**.

Grid points must be arranged in general format with one pair per line. The following Fortran commands are used to read the grid points data from **grid.dat**.

```
jlm2 = jl-2
read(iunit,*) il,jlm2
read(iunit,*) ((x(i,j), y(i,j), j=1,jlm2)), i=1,il)
```

Table 3.6 lists the data **GENERAL** uses from **eddy2c.dat** and the **EDDY2C_DATA** menu where the quantities are located.

Table 3.6: *Input data used by Program* **GENERAL**.

Quantity	Description	**EDDY2C_DATA** Menu
il	Number of points in the x direction	Geometry and Grid
jl	Number of points in the y direction	Geometry and Grid

When you click on the "Generate Grid and Initial Conditions" button on the opening menu of **EDDY2C_DATA**, control passes to **GENERAL** when a general geometry has been selected. **GENERAL** writes binary disk file **input2c.dat**, which is used by Program **EDDY2C** to define the grid and initial flowfield properties. When it completes its operations, **GENERAL** and confirms that the grid and initial conditions have been successfully created.

Program STEP: Using the Schwarz-Christoffel transformation, this program generates a grid conforming to the shape of a backward-facing step. Figure 3.22 schematically depicts the structure of the grid.

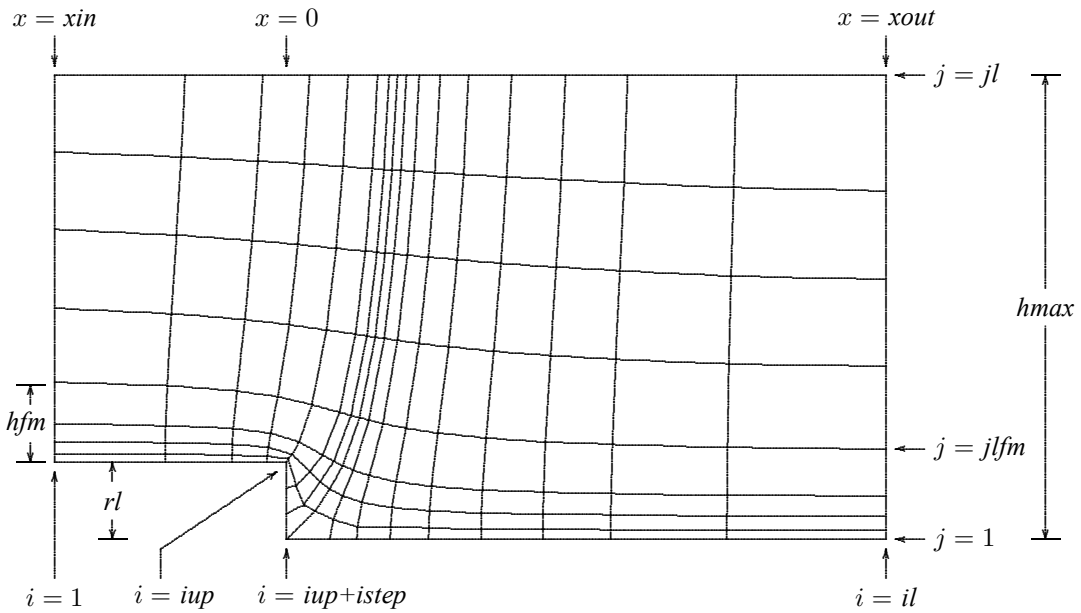

Figure 3.22: *Finite-difference grid schematic for Program* **STEP** *(not to scale). Horizontal distance is x, vertical distance is y and the origin lies at the base of the step.*

Streamwise grid-point spacing is constant along the step. Points between the upstream boundary and the top of the step are arranged according to a geometric progression and similarly for points between the bottom of the step and the downstream boundary.

Grid points normal to the surface are placed in a fine mesh to facilitate accurate resolution of the boundary layer. Grid-point spacing follows a geometric progression with the progression ratio specified by the user. The balance of the points between the fine mesh and upper mesh boundary are also arranged in a geometric progression based on the computed cell size just below the fine-mesh/upper-mesh interface.

eddy2c.dat contains all of the information needed to define the grid. Table 3.7 lists the data **STEP** uses and the **EDDY2C_DATA** menu where the quantities are located.

Table 3.7: *Input data used by Program* **STEP**.

Quantity	Description	**EDDY2C_DATA** Menu
hfm	Height of the fine mesh	Geometry and Grid
hmax	Mesh height at the outlet plane	Geometry and Grid
il	Number of points in the x direction	Geometry and Grid
istep	Number of points on the step	Boundary Conditions
iup	Point where the step begins	Boundary Conditions
jl	Number of points in the y direction	Geometry and Grid
jlfm	Number of points in the fine mesh	Geometry and Grid
ratey	Fine-mesh geometric-progression ratio	Geometry and Grid
rl	Step height	Geometry and Grid
xin	Value of x on the inlet plane	Geometry and Grid
xout	Value of x on the outlet plane	Geometry and Grid

When you click on the "Generate Grid and Initial Conditions" button on the opening menu of **EDDY2C_DATA**, control passes to **STEP** when the backward-facing step geometry has been selected. **STEP** creates binary disk file **input2c.dat**, which is used by Program **EDDY2C** to define the grid and initial flowfield properties. When it completes its operations, **STEP** reports grid parameters, including computed height of the cell closest to the surface and computed geometric progression ratios for the parts of the grid where they are used.

Program WEDGE: This program generates a grid consisting of vertical lines (constant x) with rectangular cells for shock-free flow over a flat plate or shock-wave/boundary-layer interactions and with trapezoidal cells for compression corners.

Streamwise grid-point spacing is constant. Vertical grid points are placed in a fine mesh near the surface to facilitate accurate resolution of the boundary layer. Grid-point spacing follows a geometric progression with the point closest to the surface specified by the user. The balance of the points between the fine mesh and upper mesh boundary have equal spacing.

eddy2c.dat contains all of the information needed to define the grid. Table 3.8 lists the data **WEDGE** uses and the **EDDY2C_DATA** menu where the quantities are located. Figures 3.23 and 3.24 schematically depict the structure of the grids.

Table 3.8: *Input data used by Program* **WEDGE**.

Quantity	Description	**EDDY2C_DATA** Menu
dy0	Height of the cell nearest the surface	Geometry and Grid
hfm	Height of the fine mesh	Geometry and Grid
hmax	Vertical height of the mesh	Geometry and Grid
il	Number of points in the x direction	Geometry and Grid
ishk	Point where the shock strikes the surface in an inviscid flow	Boundary Conditions
iwedge	Point where the wedge begins	Boundary Conditions
jl	Number of points in the y direction	Geometry and Grid
jlfm	Number of points in the fine mesh	Geometry and Grid
rl	Distance from the inlet to x(*ishk*) or x(*iwedge*)	Geometry and Grid

When you click on the "Generate Grid and Initial Conditions" button on the opening menu of **EDDY2C_DATA**, control passes to **WEDGE** when shock-wave/boundary-layer interaction or compression corner has been selected. **WEDGE** creates binary disk file **input2c.dat**, which is used by Program **EDDY2C** to define the grid and initial flowfield properties. When it completes its operations, **WEDGE** reports the computed fine-mesh geometric progression ratio and confirms that the grid and initial conditions have been successfully created.

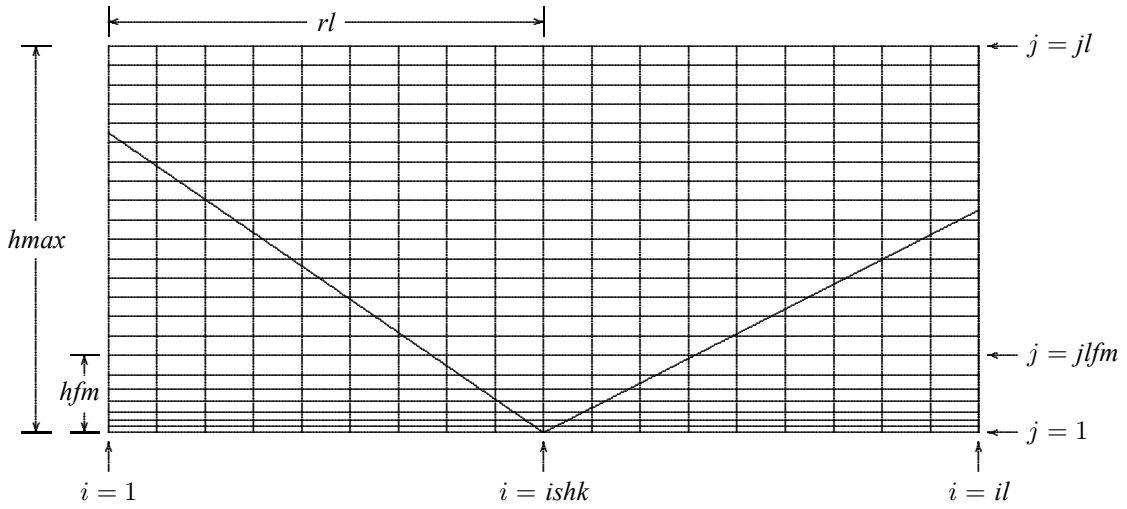

Figure 3.23: *Shock-wave/boundary-layer interaction finite-difference grid schematic for Program* **WEDGE**. *Horizontal distance is x, vertical distance is y and the origin lies at the point where the shock would impinge on the plate in an inviscid flow.*

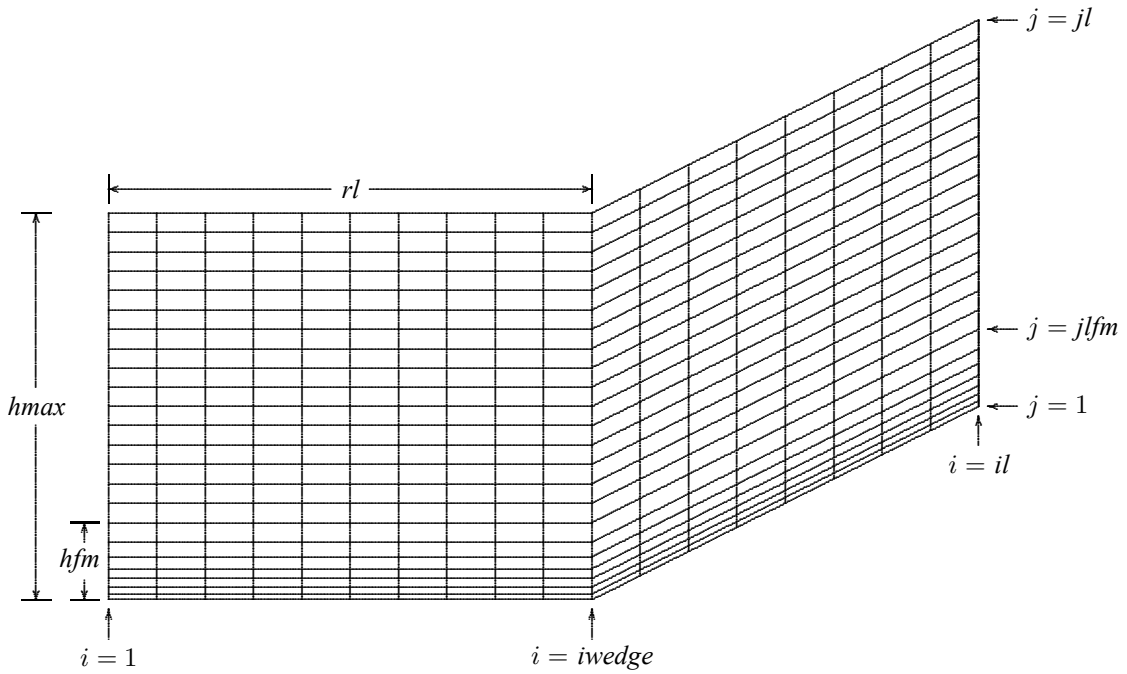

Figure 3.24: *Compression-corner grid schematic for Program* **WEDGE**. *Horizontal distance is x, vertical distance is y and the origin lies at the corner.*

3.4.5 Program Output

Program **EDDY2C** can be executed from **EDDY2C_DATA** or as a standalone program. Output from **EDDY2C** consists of the following three disk files.

- **eddy2c.prt** is a comprehensive print file detailing results of the computation. The next subsection describes the contents of the file.
- **output2c.dat** is a binary file with sufficient information to restart a run if desired.
- **wall2c.dat** is a plotting file containing surface-pressure and skin-friction data. The first line of the file has the number of lines of data in the file and the pressure flag, *ipwall*. If *ipwall* is zero, the file contains the ratio of wall to freestream static pressure, P_w/P_∞. Otherwise, pressure appears as pressure coefficient, C_p. Each line includes horizontal distance, x, either P_w/P_∞ or C_p and skin friction with format (1p3e12.4).

Plotting program **EDDY2C_PLOT** uses the two disk files **eddy2c.dat** and **wall2c.dat**, and can be run from **EDDY2C_DATA** or as a standalone program.

3.4.6 EDDY2C Output Parameters

As the run progresses, several quantities are printed on the video display and in the **EDDY2C** output file **eddy2c.prt**. Table 3.9 lists the printed quantities.

Table 3.9: *Quantities printed by Program* **EDDY2C**.

Name	Symbol/Equation	Definition
i	i	Mesh point number
x	x/L	Dimensionless streamwise distance
P/Po	P_w/P_∞	Dimensionless surface pressure
Cp	C_p	Pressure coefficient
Cf	$\tau_w/(\frac{1}{2}\rho_\infty U_\infty^2)$	Skin friction
q	$-q_w/[\rho_\infty U_\infty c_p(T_{t\infty} - T_w)]$	Heat-transfer coefficient
yplus	$u_\tau y/\nu_w$	Sublayer-scaled distance of the first grid point above the surface

Table 3.10 lists the flow properties printed in the profiles portion of **eddy2c.prt**.

Table 3.10: *Flow-property profiles printed by Program* **EDDY2C**.

Name	Symbol/Equation	Definition
j	j	Mesh point number
u/U	$\tilde{u}/U_\infty$	Dimensionless horizontal velocity
v/U	$\tilde{v}/U_\infty$	Dimensionless vertical velocity
rho	$\bar{\rho}/\rho_\infty$	Density ratio
p	P/P_∞	Pressure ratio
T	$\tilde{T}/T_\infty$	Temperature ratio
Ret	μ_t/μ	Dimensionless eddy viscosity
k/U**2	k/U_∞^2	Dimensionless turbulence kinetic energy
omega	$\nu_\infty\omega/U_\infty$	Dimensionless dissipation rate
rhotxx	$-\overline{\rho u'^2}/(\rho_\infty U_\infty^2)$	Dimensionless xx Reynolds stress
rhotxy	$-\overline{\rho u'v'}/(\rho_\infty U_\infty^2)$	Dimensionless xy Reynolds stress
rhotyy	$-\overline{\rho v'^2}/(\rho_\infty U_\infty^2)$	Dimensionless yy Reynolds stress

The quantities *k/U**2* and *omega* are not printed for the Baldwin-Lomax model. The quantities *rhotxx*, *rhotxy* and *rhotyy* are printed only for the Stress-ω model. Corresponding mesh coordinates are printed by the grid-generation program used to set up the computation.

3.4.7 Turbulence Models Implemented in EDDY2C

There are 5 turbulence models implemented in Program **EDDY2C**, which uses the two input parameters *model* and *nvisc* to make the selection. Table 3.11 lists the models implemented.

Table 3.11: *Turbulence models implemented in EDDY2C.*

Model	Type	*model*	*nvisc*
None (laminar flow)	—	-1	Not used
k-ω (high Re) - Wilcox (2006)	2-Equation	0	0
k-ω (low Re) - Wilcox (2006)	2-Equation	0	1
Stress-ω (high Re) - Wilcox (2006)	Stress-Transport	1	0
Stress-ω (high Re) - Wilcox (2006)	Stress-Transport	1	1
Baldwin-Lomax (1978)	Algebraic	2	Not used

3.5 Technical Information

The program uses the MacCormack (1985) fully-implicit flux-splitting method with Gauss-Seidel line relaxation. Five turbulence models are implemented in the program, viz., the Baldwin-Lomax (1978) algebraic model, the Wilcox (2006) two-equation k-ω model and the Wilcox (2006) stress-transport Stress-ω model. The k-ω and Stress-ω models are implemented with and without viscous modifications. The first section first presents the governing equations for mean-flow properties. Subsequent sections present the turbulence-model equations implemented in the program.

3.5.1 Mean-Flow Equations

The equations governing conservation of mass, momentum and mean energy for all models are the same. In terms of Favre-averaged flow properties, they are:

$$\frac{\partial \bar{\rho}}{\partial t} + \frac{\partial}{\partial x_i}\left(\bar{\rho}\tilde{u}_i\right) = 0 \tag{3.4}$$

$$\frac{\partial}{\partial t}\left(\bar{\rho}\tilde{u}_i\right) + \frac{\partial}{\partial x_j}\left(\bar{\rho}\tilde{u}_j\tilde{u}_i\right) = -\frac{\partial P}{\partial x_i} + \frac{\partial}{\partial x_j}\left[\bar{t}_{ji} + \bar{\rho}\tau_{ji}\right] \tag{3.5}$$

$$\begin{aligned}\frac{\partial}{\partial t}(\bar{\rho}E) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j H) &= \frac{\partial}{\partial x_j}\left[\left(\frac{\mu}{Pr_{_L}} + \frac{\mu_{_T}}{Pr_{_T}}\right)\frac{\partial \tilde{T}}{\partial x_j} + \left(\mu + \sigma^*\frac{\bar{\rho}k}{\omega}\right)\frac{\partial k}{\partial x_j}\right] \\ &+ \frac{\partial}{\partial x_j}\left[\tilde{u}_i(\bar{t}_{ij} + \bar{\rho}\tau_{ij})\right]\end{aligned} \tag{3.6}$$

The notation is as follows: t denotes time, x_i is position vector, $\tilde{u}_i$ is velocity vector, $\bar{\rho}$ is density, P is pressure, $\tilde{T}$ is temperature, $\bar{t}_{ji}$ is molecular stress tensor, τ_{ji} is Reynolds-stress tensor, $\mu_{_T}$ is eddy viscosity, $Pr_{_L}$ is laminar Prandtl number and $Pr_{_T}$ is turbulent Prandtl number. Also, k is turbulence kinetic energy and σ^* is a closure coefficient that will be defined below. For the Baldwin-Lomax model, k is set equal to zero. The perfect-gas law relates pressure, density and temperature.

$$P = \bar{\rho}R\tilde{T} \tag{3.7}$$

The quantities E and H are the total energy and total enthalpy, and include the kinetic energy of the fluctuating turbulent field, viz.,

$$E = \tilde{e} + \tfrac{1}{2}\tilde{u}_i\tilde{u}_i + k \qquad \text{and} \qquad H = \tilde{h} + \tfrac{1}{2}\tilde{u}_i\tilde{u}_i + k \tag{3.8}$$

3.5.2 Baldwin-Lomax Model

For the Baldwin-Lomax model [Baldwin and Lomax (1978)], the eddy viscosity is given by

$$\mu_T = \begin{cases} \mu_{T_i}, & y \le y_m \\ \mu_{T_o}, & y > y_m \end{cases} \tag{3.9}$$

where y denotes distance normal to the surface. The inner and outer layer viscosities, μ_{T_i} and μ_{T_o}, are defined as follows.

Inner Layer:

$$\mu_{T_i} = \bar{\rho}\ell_{mix}^2|\omega| \tag{3.10}$$

$$\ell_{mix} = \kappa y\left[1 - e^{-y^+/A_o^+}\right] \tag{3.11}$$

Outer Layer:

$$\mu_{T_o} = \alpha C_{cp}\bar{\rho}F_{wake}F_{Kleb}(y) \tag{3.12}$$

$$F_{wake} = \min\left[y_{max}F_{max}; C_{wk}y_{max}U_{dif}^2/F_{max}\right] \tag{3.13}$$

$$F_{max} = \frac{1}{\kappa}\left[\max_y(\ell_{mix}|\omega|)\right] \tag{3.14}$$

where y_{max} is the value of y at which $\ell_{mix}|\omega|$ achieves its maximum value.

Closure Coefficients:

$$\left.\begin{array}{lll} \kappa = 0.40, & \alpha = 0.0168, & A_o^+ = 26 \\ C_{cp} = 1.6, & C_{Kleb} = 0.3, & C_{wk} = 1 \end{array}\right\} \tag{3.15}$$

The function F_{Kleb} is Klebanoff's intermittency function defined by

$$F_{Kleb}(y) = \left[1 + 5.5\left(\frac{C_{Kleb}\,y}{y_{max}}\right)^6\right]^{-1} \tag{3.16}$$

Also, ω is the magnitude of the vorticity vector, i.e.,

$$\omega = \left|\frac{\partial\tilde{v}}{\partial x} - \frac{\partial\tilde{u}}{\partial y}\right| \tag{3.17}$$

The quantity U_{dif} is the maximum value of $\tilde{u}$ for boundary layers. For free shear layers, U_{dif} is the difference between the maximum velocity in the layer and the value of $\tilde{u}$ at $y = y_{max}$. For more general flows, it is defined by

$$U_{dif} = \left(\sqrt{\tilde{u}^2 + \tilde{v}^2}\right)_{max} - \left(\sqrt{\tilde{u}^2 + \tilde{v}^2}\right)_{y=y_{max}} \tag{3.18}$$

3.5.3 k-ω Model

The equations defining the high-Reynolds-number version of the Wilcox (2006) k-ω model are as follows. See Subsection 2.5.2 for viscous modifications to the model.

Molecular and Reynolds-Stress Tensors:

$$\bar{t}_{ij} = 2\mu\bar{S}_{ij}, \qquad \bar{\rho}\tau_{ij} = 2\mu_T\bar{S}_{ij} - \frac{2}{3}\bar{\rho}k\delta_{ij}, \qquad \bar{S}_{ij} = S_{ij} - \frac{1}{3}\frac{\partial\tilde{u}_k}{\partial x_k}\delta_{ij} \tag{3.19}$$

Eddy Viscosity:

$$\mu_T = \frac{\bar{\rho}k}{\tilde{\omega}}, \qquad \tilde{\omega} = \max\left\{\omega, \;\; C_{lim}\sqrt{\frac{2\bar{S}_{ij}\bar{S}_{ij}}{\beta^*}}\right\}, \qquad C_{lim} = \frac{7}{8} \tag{3.20}$$

Turbulence Kinetic Energy:

$$\frac{\partial}{\partial t}(\bar{\rho}k) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j k) = \bar{\rho}\tau_{ij}\frac{\partial\tilde{u}_i}{\partial x_j} - \beta^*\bar{\rho}k\omega + \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma^*\frac{\bar{\rho}k}{\omega}\right)\frac{\partial k}{\partial x_j}\right] \tag{3.21}$$

Specific Dissipation Rate:

$$\begin{aligned}\frac{\partial}{\partial t}(\bar{\rho}\omega) + \frac{\partial}{\partial x_j}(\bar{\rho}\tilde{u}_j\omega) &= \alpha\frac{\omega}{k}\rho\tau_{ij}\frac{\partial\tilde{u}_i}{\partial x_j} - \beta\bar{\rho}\omega^2 + \sigma_d\frac{\bar{\rho}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial\omega}{\partial x_j} \\ &+ \frac{\partial}{\partial x_j}\left[\left(\mu + \sigma\frac{\bar{\rho}k}{\omega}\right)\frac{\partial\omega}{\partial x_j}\right]\end{aligned} \tag{3.22}$$

Closure Coefficients:

$$\alpha = \frac{13}{25}, \quad \beta = \beta_o f_\beta, \quad \beta^* = \frac{9}{100}, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{3}{5}, \quad \sigma_{do} = \frac{1}{8} \tag{3.23}$$

$$\beta_o = 0.0708, \quad Pr_T = \frac{8}{9}, \quad \sigma_d = \begin{cases} 0, & \dfrac{\partial k}{\partial x_j}\dfrac{\partial\omega}{\partial x_j} \le 0 \\ \sigma_{do}, & \dfrac{\partial k}{\partial x_j}\dfrac{\partial\omega}{\partial x_j} > 0 \end{cases} \tag{3.24}$$

$$f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega}, \quad \chi_\omega \equiv \left|\frac{\Omega_{ij}\Omega_{jk}\hat{S}_{ki}}{(\beta^*\omega)^3}\right|, \quad \hat{S}_{ki} = S_{ki} - \frac{1}{2}\frac{\partial\tilde{u}_m}{\partial x_m}\delta_{ki} \tag{3.25}$$

3.5.4 Stress-ω Model

The equations defining the high-Reynolds-number version of the Wilcox (2006) Stress-ω model are as follows. See Subsection 2.5.2 for viscous modifications to the model.

Reynolds-Stress Tensor:

$$\bar{\rho}\frac{\partial \tau_{ij}}{\partial t} + \bar{\rho}\tilde{u}_k\frac{\partial \tau_{ij}}{\partial x_k} = -\bar{\rho}P_{ij} + \frac{2}{3}\beta^*\bar{\rho}\omega k\delta_{ij} - \bar{\rho}\Pi_{ij} + \frac{\partial}{\partial x_k}\left[\left(\mu + \sigma^*\mu_T\right)\frac{\partial \tau_{ij}}{\partial x_k}\right] \tag{3.26}$$

Specific Dissipation Rate:

$$\begin{aligned}\bar{\rho}\frac{\partial \omega}{\partial t} + \bar{\rho}\tilde{u}_j\frac{\partial \omega}{\partial x_j} = \alpha\frac{\bar{\rho}\omega}{k}\tau_{ij}\frac{\partial \tilde{u}_i}{\partial x_j} - \beta\bar{\rho}\omega^2 + \sigma_d\frac{\bar{\rho}}{\omega}\frac{\partial k}{\partial x_j}\frac{\partial \omega}{\partial x_j} \\ + \frac{\partial}{\partial x_k}\left[\left(\mu + \sigma\mu_T\right)\frac{\partial \omega}{\partial x_k}\right]\end{aligned} \tag{3.27}$$

Pressure-Strain Correlation:

$$\begin{aligned}\Pi_{ij} = \beta^* C_1\omega\left(\tau_{ij} + \frac{2}{3}k\delta_{ij}\right) - \hat{\alpha}\left(P_{ij} - \frac{2}{3}P\delta_{ij}\right) \\ -\hat{\beta}\left(D_{ij} - \frac{2}{3}P\delta_{ij}\right) - \hat{\gamma}k\left(S_{ij} - \frac{1}{3}S_{kk}\delta_{ij}\right)\end{aligned} \tag{3.28}$$

Auxiliary Relations:

$$\mu_T = \bar{\rho}k/\omega \tag{3.29}$$

$$P_{ij} = \tau_{im}\frac{\partial \tilde{u}_j}{\partial x_m} + \tau_{jm}\frac{\partial \tilde{u}_i}{\partial x_m}, \qquad D_{ij} = \tau_{im}\frac{\partial \tilde{u}_m}{\partial x_j} + \tau_{jm}\frac{\partial \tilde{u}_m}{\partial x_i}, \qquad P = \frac{1}{2}P_{kk} \tag{3.30}$$

Closure Coefficients:

$$\hat{\alpha} = (8 + C_2)/11, \quad \hat{\beta} = (8C_2 - 2)/11, \quad \hat{\gamma} = (60C_2 - 4)/55 \tag{3.31}$$

$$C_1 = \frac{9}{5}, \qquad C_2 = \frac{10}{19} \tag{3.32}$$

$$\alpha = \frac{13}{25}, \quad \beta = \beta_o f_\beta, \quad \beta^* = \frac{9}{100}, \quad \sigma = \frac{1}{2}, \quad \sigma^* = \frac{3}{5}, \quad \sigma_{do} = \frac{1}{8} \tag{3.33}$$

$$\beta_o = 0.0708, \quad \sigma_d = \begin{cases} 0, & \dfrac{\partial k}{\partial x_j}\dfrac{\partial \omega}{\partial x_j} \le 0 \\ \sigma_{do}, & \dfrac{\partial k}{\partial x_j}\dfrac{\partial \omega}{\partial x_j} > 0 \end{cases} \tag{3.34}$$

$$f_\beta = \frac{1 + 85\chi_\omega}{1 + 100\chi_\omega}, \quad \chi_\omega \equiv \left|\frac{\Omega_{ij}\Omega_{jk}\hat{S}_{ki}}{\left(\beta^*\omega\right)^3}\right|, \quad \hat{S}_{ki} = S_{ki} - \frac{1}{2}\frac{\partial \tilde{u}_m}{\partial x_m}\delta_{ki} \tag{3.35}$$

Appendix A

Fortran Compatibility

All of the Fortran programs described in this document include Subroutine **NAMSYS**, which returns several system-dependent and compiler-specific parameters.

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, Pentium)
- 2 VAX/VMS
- 3 SUN Fortran (SUN Workstation)
- 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 redirected to a disk file
- *'printer'* for SVS Fortran
- *'print'* for SUN Fortran
- *'formatted'* for all others

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 is a complete listing of **NAMSYS**.

```
      subroutine namsys(icpu,iin,iv,msdos,newfil,pform)
c=======================================================================
c  S E T   U P   S Y S T E M - D E P E N D E N T   P A R A M E T E R S
c=======================================================================
      character newfil*7,pform*9
c-----------------------------------------------------------------------
c  Must define the following...
c   iv     = Standard console unit number
c   msdos  = 0...Printer opened as 'prn'
c            1...Don't open 'prn'
c   pform  = 'printer' for SVS Fortran
c          = 'print' for SUN Fortran
c          = 'formatted' for all others
c   newfil = 'new' if compiler writes over an existing file
c          = 'unknown' for Ansi-77 standard operation
c-----------------------------------------------------------------------
c=======================================================================
c---------------------------------
c  Define main input data file LUN
c---------------------------------
      iin=15
c---------------------------------
c       Define console LUN
c---------------------------------
      iv=5
c---------------------------------
c  SVS Fortran (680x0 and 80x86)
c---------------------------------
      if(icpu.eq.0) then
        msdos=0
        pform='printer'
        newfil='new'
c---------------------------------
c  Lahey/Microsoft Fortran (80x86)
c---------------------------------
      elseif(icpu.eq.1) then
        msdos=0
        pform='formatted'
        newfil='unknown'
c---------------------------------
c             VAX/VMS
c---------------------------------
      elseif(icpu.eq.2) then
        msdos=1
```

```
        pform='formatted'
        newfil='new'
c---------------------------------
c   SUN Fortran...SUN Workstation
c       and Definicon SPARC
c---------------------------------
      elseif(icpu.eq.3) then
        msdos=0
        pform='print'
        newfil='unknown'
c---------------------------------
c     Cray Fortran...Unicos
c---------------------------------
      elseif(icpu.eq.4) then
        msdos=1
        pform='formatted'
        newfil='unknown'
c---------------------------------
c      Silicon Graphics Iris
c---------------------------------
      elseif(icpu.eq.5) then
        msdos=0
        pform='formatted'
        newfil='unknown'
c---------------------------------
c      Error...say so and quit
c---------------------------------
      else
        write(*,*) 'icpu = ',icpu,' is not supported!!!'
        pause
        stop
      endif
      return
      end
```

Bibliography

Andersen, P. S., Kays, W. M. and Moffat, R. J. (1972), "The Turbulent Boundary Layer on a Porous Plate: An Experimental Study of the Fluid Mechanics for Adverse Free-Stream Pressure Gradients," Report No. HMT-15, Dept. Mech. Eng., Stanford Univ.

Bachalo, W. D. and Johnson, D. A. (1979), "An Investigation of Transonic Turbulent Boundary Layer Separation Generated on an Axisymmetric Flow Model," AIAA Paper 79-1479.

Baldwin, B. S. and Lomax, H. (1978), "Thin-Layer Approximation and Algebraic Model for Separated Turbulent Flows," AIAA Paper 78-257.

Baldwin, B. S. and Barth, T. J. (1990), "A One-Equation Turbulence Transport Model for High Reynolds Number Wall-Bounded Flows," NASA TM-102847 [see also AIAA Paper 91-0610 (1991)].

Blottner, F. G. (1974), "Variable Grid Scheme Applied to Turbulent Boundary Layers," *Comput. Meth. Appl. Mech. & Eng.*, Vol. 4, No. 2, pp. 179-194.

Cebeci, T. and Smith, A. M. O. (1974), *Analysis of Turbulent Boundary Layers*, Ser. in Appl. Math. & Mech., Vol. XV, Academic Press, Orlando, FL.

Chien, K.-Y. (1982), "Predictions of Channel and Boundary-Layer Flows with a Low-Reynolds-Number Turbulence Model," *AIAA Journal*, Vol. 20, No. 1, pp. 33-38.

Coles, D. E. and Hirst, E. A. (1969), *Computation of Turbulent Boundary Layers-1968 AFOSR-IFP-Stanford Conference*, Vol. II, Thermosciences Div., Stanford University, CA.

Crank, J. and Nicolson, P. (1947), "A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type," *Proceedings of the Cambridge Philosophical Society*, Vol. 43, No. 50, pp. 50-67.

Driver, D. M. and Seegmiller, H. L. (1985), "Features of a Reattaching Turbulent Shear Layer in Divergent Channel Flow," *AIAA Journal*, Vol. 23, No. 1, pp. 163-171.

Driver, D. M. (1991), "Reynolds Shear Stress Measurements in a Separated Boundary Layer," AIAA Paper 91-1787.

Fan, S., Lakshminarayana, B. and Barnett, M. (1993), "A Low-Reynolds Number k-ϵ Model for Unsteady Turbulent Boundary Layer Flows," *AIAA Journal*, Vol. 31, No. 10, pp. 1777-1784.

Fernholz, H. H. and Finley, P. J. (1981), "A Further Compilation of Compressible Boundary Layer Data with a Survey of Turbulence Data," AGARDograph 263.

Harris, J. E. and Blanchard, D. K. (1982), "Computer Program for Solving Laminar, Transitional, or Turbulent Compressible Boundary-Layer Equations for Two-Dimensional and Axisymmetric Flow," NASA TM-83207.

Hayes, W. D. and Probstein, R. F. (1959), *Hypersonic Flow Theory*, Academic Press, New York, NY, p. 290.

Holden, M. S. (1978), "A Study of Flow Separation in Regions of Shock Wave-Boundary Layer Interaction in Hypersonic Flow," AIAA Paper 78-1168.

Johnson, D. A. and King, L. S. (1985), "A Mathematically Simple Turbulence Closure Model for Attached and Separated Turbulent Boundary Layers," *AIAA Journal*, Vol. 23, No. 11, pp. 1684-1692.

Jones, W. P. and Launder, B. E. (1972), "The Prediction of Laminarization with a Two-Equation Model of Turbulence," *International Journal of Heat and Mass Transfer*, Vol. 15, pp. 301-314.

Jovic, S. and Driver, D. (1994), "Backward-Facing Step Measurements at Low Reynolds Number," NASA TM-108870.

Kline, S. J., Cantwell, B. J. and Lilley, G. M. (1981), *1980-81 AFOSR-HTTM-Stanford Conference on Complex Turbulent Flows*, Thermosciences Division, Stanford University.

Kussoy, M. I. and Horstman, C. C. (1989), "Documentation of Two- and Three-Dimensional Hypersonic Shock Wave/Turbulent Boundary Layer Interaction Flows," NASA TM 101075.

Lam, C. K. G. and Bremhorst, K. A. (1981), "Modified Form of k-ϵ Model for Predicting Wall Turbulence," *ASME, Journal of Fluids Engineering*, Vol. 103, pp. 456-460.

Laufer, J. (1952), "The Structure of Turbulence in Fully Developed Pipe Flow," NACA Rept. 1174.

Launder, B. E. and Sharma, B. I. (1974), "Application of the Energy Dissipation Model of Turbulence to the Calculation of Flow Near a Spinning Disc," *Letters in Heat and Mass Transfer*, Vol. 1, No. 2, pp. 131-138.

MacCormack, R. W. (1985), "Current Status of Numerical Solutions of the Navier-Stokes Equations," AIAA Paper 85-32.

Mansour, N. N., Kim, J. and Moin, P. (1988), "Reynolds Stress and Dissipation Rate Budgets in Turbulent Channel Flow," *Journal of Fluid Mechanics*, Vol. 194, pp. 15-44.

Murthy, V. S. and Rose, W. C. (1978), "Wall Shear Stress Measurements in a Shock-Wave/Boundary-Layer Interaction," *AIAA Journal*, Vol. 16, No. 7, pp. 667-672.

Pope, S. B. (1978), "An Explanation of the Turbulent Round-Jet/Plane-Jet Anomaly," *AIAA Journal*, Vol. 16, No. 3, pp. 279-281.

Prandtl, L. (1925), "Über die ausgebildete Turbulenz," *ZAMM*, Vol. 5, pp. 136-139.

Price, J. M. and Harris, J. E. (1972), "Computer Program for Solving Compressible Nonsimilar-Boundary-Layer Equations for Laminar, Transitional or Turbulent Flows of a Perfect Gas," NASA TM X-2458.

Reda, D. C. and Murphy, J. D. (1972), "Shock-Wave/Turbulent-Boundary-Layer Interactions in Rectangular Channels," *AIAA Journal*, Vol. 11, No. 2, pp. 139-140; also AIAA Paper 72-715.

Rubel, A. and Melnik, R. E. (1984), "Jet, Wake and Wall Jet Solutions Using a k-ϵ Turbulence Model," AIAA Paper 84-1523, Snowmass, CO.

Samimy, M., Petrie, H. L. and Addy, A. L. (1985), "Reattachment and Redevelopment of Compressible Turbulent Free Shear Layers," *International Symposium on Laser Anemometry*, FED-Vol. 33, pp. 159-166.

Schubauer, G. B. and Klebanoff, P. S. (1955), "Contributions on the Mechanics of Boundary-Layer Transition," NASA TN 3489.

Settles, G. S., Vas, I. E. and Bogdonoff, S. M. (1976), "Details of a Shock Separated Turbulent Boundary Layer at a Compression Corner," *AIAA Journal*, Vol. 14, No. 12, pp. 1709-1715.

So, R. M. C. and Mellor, G. L. (1972), "An Experimental Investigation of Turbulent Boundary Layers Along Curved Surfaces," NASA CR-1940.

Spalart, P. R. and Allmaras, S. R. (1992), "A One-Equation Turbulence Model for Aerodynamic Flows," AIAA Paper 92-439, Reno, NV [see also *La Recherche Aerospatiale* No. 1, p. 5 (1994)].

Stratford, B. S. (1959), "An Experimental Flow with Zero Skin Friction Throughout its Region of Pressure Rise," *Journal of Fluid Mechanics*, Vol. 5, pp. 17-35.

Wilcox, D. C. (1981), "Algorithm for Rapid Integration of Turbulence Model Equations on Parabolic Regions," *AIAA Journal*, Vol. 19, No. 2, pp. 248-251.

Wilcox, D. C. (2006), *Turbulence Modeling for CFD*, Third Edition, DCW Industries, Inc., La Cañada, CA.

Yakhot, V. and Orszag, S. A. (1986), "Renormalization Group Analysis of Turbulence: 1. Basic Theory," *Journal of Scientific Computing*, Vol. 1, pp. 3-51.

Yang, Z. and Shih, T.-H. (1993), "A New Time Scale Based k-ϵ Model for Near Wall Turbulence," *AIAA Journal*, Vol. 31, No. 7, pp. 1191-1198.