Proceedings of FEDSM'98
1998 ASME Fluids Engineering Division Summer Meeting
June 21-25, 1998, Washington,DC

FEDSM98-5139

BOUNDARY LAYER CODES FOR STUDENTS IN *JAVA*

W.J. Devenport* and J.A. Schetz**
540-231-4456,540-231-9632(Fax),DEVENPORT@AOE.VT.EDU;540-231-9056,PTIGER@VT.EDU
Virginia Polytechnic Inst. and State Univ.
Blacksburg, VA

ABSTRACT

Simple *JAVA* computer codes for boundary layer problems intended for student use solving the homework problems in a text such as Boundary Layer Analysis by Schetz, 1993 and similar problems on a PC or Work Station have been developed. The notion is to make platform-independent codes conveniently available via the internet. At this point, there are four separate codes for: 1) Thwaites-Walz incompressible, laminar integral method, 2) incompressible, laminar boundary layers by an implicit numerical method, 3) Moses, incompressible turbulent integral method, and 4) incompressible turbulent boundary layers by an implicit numerical method. A brief description of each code precedes the operating instructions. Also, a default input for a typical example and results for that case are presented and discussed.

NOTATION

C_f	skin friction coefficient
$H\equiv \delta^*/\theta$	shape factor
M	Mach number
p	pressure
r(x)	body thickness distribution
Re	Reynolds number
R_o	body nose radius
s(x)	surface distance
T	temperature
K	turbulent kinetic energy
u	streamwise velocity
$U_e(x)$	edge velocity distribution
U_{inf}	freestream velocity
v	transverse velocity
x	axial and streamwise coordinate
y	transverse coordinate
$\beta\equiv(\delta^*/\tau_w)dp/dx$	Clauser pressure gradient parameter
δ	boundary layer thickness
δ^*	displacement thickness
θ	momentum thickness
$\lambda\equiv(\delta^2/\nu)dU_e/dx$	Pohlhausen pressure gradient parameter
$\Lambda\equiv(\theta^2/\nu)dU_e/dx$	Thwaites-Walz pressure gradient parameter
μ	viscosity
υ	kinematic viscosity
ρ	density
τ_w	wall shear

INTRODUCTION

Here, we present simple *JAVA* computer codes that are intended for student use solving the homework problems in a text such as Boundary Layer Analysis by Schetz, 1993 and other similar problems on a PC or Work Station. The *JAVA* language was selected so that such codes could be made conveniently available in a platform-independent form on the internet. These codes are not intended for use by working professionals in the field. The goal has been to keep the

*Associate Professor of Aerospace and Ocean Engineering, Member ASME

** J. Byron Maupin Professor of Aerospace and Ocean Engineering, Fellow ASME formulation, logic and programming as simple as possible so that the student can easily grasp the flow of the calculations. Thus, primitive variables (u,v); (x,y) are employed with no transformations.

These codes are meant only to relieve the student of the burden of writing and debugging numerous codes during a course. The hope is to thereby leave sufficient time and energy for the working of actual boundary layer problems of reasonable complexity. The student is not relieved of the burden to think. One should always estimate the answer the computer solution is expected to yield. For example, one could use a simple integral solution to estimate the level of the skin friction to be expected. The student is the analyst, not the computer, and he or she is responsible for producing and interpreting the correct answer.

At this point, there are four separate codes for the various methods covered: 1) Thwaites-Walz incompressible, laminar integral method, 2) incompressible, laminar boundary layers by an implicit numerical method, 3) Moses, incompressible turbulent integral method, and 4) incompressible turbulent boundary layers by an implicit numerical method. A brief description of each code precedes the operating instructions. Also, a default input for a typical example and results for that case are presented and discussed. Detailed development of the methods can be found in the referenced text. Earlier *FORTRAN* versions of these codes are also in that reference.

For any boundary layer problem, one must specify the fluid through density, ρ and viscosity, μ, or perhaps just kinematic viscosity,υ. In variable-density, variable-property cases, an equation of state and property variation information must also be given. The next information needed would be the freestream velocity U_{inf} and the inviscid edge velocity distribution, $U_e(x)$. For high-speed flows, Mach numbers, M_{inf} and $M_e(x)$, are needed. With heat transfer or compressible flow, the wall temperature, $T_w(x)$, or the wall heat transfer distribution and T_{inf} and $T_e(x)$ are required. Next is the matter of initial conditions. The integral methods only require initial values of quantities like θ, δ, and C_f at x_{init}. The differential methods need initial profiles of u and v at x_{init}. For turbulent cases, one must select the turbulence model, *e.g.* mixing length model, eddy viscosity model or *TKE* model with the Prandtl Energy Method. The *TKE* method requires an initial profile and boundary conditions for K.

Finally, the computational region must be selected. With an integral method, this is simply the length of interest. The differential methods require the height and the length of the computational region. The height should grow with streamwise distance to accommodate the growth in thickness of the boundary layer. The programs here are based on the untransformed equations to keep the codes simple, so the user must pick a computational region high enough at the initial station to accommodate the estimated growth of the boundary layer by the downstream end of the computational region. The last matter concerns the choice of the step size(s). The integral methods only need a choice of dx, while the differential methods need dx and dy. This is normally accomplished by picking the number of points across the height and length of the region. For simplicity, most the codes included here have fixed step sized in the x and y directions. For a laminar flow, about 20-25 points across the boundary layer are sufficient. Thus, knowing the thickness of the boundary layer at the initial station permits a reasonable choice for dy. With the boundary layer approximation, we can take $dx >> dy$, and a value of $dx \approx \delta_i/2$ is usually adequate. For a turbulent flow a stretched grid in the y direction is much more efficient, but that was not implemented in the interests of simplicity. Therefore, a large number of grid points is necessary, about 1000 points across the boundary layer.

PROGRAM *WALZ*

This program is an implementation of the Thwaites-**WALZ** integral method for incompressible, laminar boundary layer flows. Refer to Sec. 2-3-2 in Schetz (1993) for a description of the technique (and Appendix B in the same reference for the original *FORTRAN* code). A default input set is included for the following flow problem. This is the same as the original worked example.

Example Laminar Integral Method Problem: Consider 2D laminar flow of a fluid with a kinematic viscosity $\upsilon = 2.0\text{x}10^{-4}$ m^2/s at $U_{inf} = 10.0$ m/s over a surface that is a flat plate from the leading edge to $x = 1.0$ m. At that station, a ramp begins that produces an inviscid velocity distribution $U_e(x) = 10.5 - x/2$, m/s. This is an adverse pressure gradient, since U_e is decreasing so that p increases. Calculate the boundary layer development over this surface up to $x = 2.0$ m. Does the flow separate?

Solution:

We must provide input data for the kinematic viscosity as υ = 2e-4 and the freestream velocity as $U_{inf} = 10.0$. Select *NMAX* = 41 to give $dx = 0.05$.

The body thickness distribution, $r(x)$, can be specified. The code will then calculate the surface distance along the body. The default is $r(x) = 0.0$. Blunt bodies need input for $R_{o.}$

Finally, the inviscid velocity distribution is required. Since this case has a bi-linear edge velocity variation, velocities at only a few points including x_{init} and x_{fin} need to be specified to define the distribution.

The window in Fig. 1 shows the input information for the default case. On the left are two panels containing the input data for this case. The input data in the far left panel can be changed by selecting the item to be changed with the menu button, entering the new value in the slot below and pushing *SET*. The input data in the next panel to the right can be changed in a similar manner. One might wish to change the dimensionless edge velocity distribution, U_e/U_{inf} and the body shape. All that is required is to enter or modify sets of values for x, r and U_e/U_{inf} in the panel below and push *SET*. The code will fit a spline through the points used as input.

Then, press the *START* button and watch the skin friction and the integral quantities develop in the graphs on the right. The window given in Fig. 2 shows the results about halfway through the calculation. Tabular values of the output can be accessed in the panel indicated. One can scroll up and down in the table with the slider bar and left and right with the cursor. Some browsers display the table starting at the bottom, so it may be necessary to scroll up to see the column headers.

PROGRAM *ILBLI*

This code implements a numerical solution method for **I**ncompressible, **L**aminar **B**oundary **L**ayer flows with an **I**mplicit technique. See Sec. 4-7-3 in Schetz (1993) for details of the development (and the original *FORTRAN* code in Appendix B). A default case corresponding to the following example problem is shown. The equations are solved in dimensionless form using the freestream velocity, U_{inf}, a reference length, L, (taken as 1.0 for convenience), and reference viscosity and density, μ_{inf} and ρ_{inf}. The input must be in that form, and the output is in that form.

For flows with an initial stagnation point, it is convenient to use the Thwaites-Walz integral method for a short distance and then use those results to construct initial conditions for this numerical method. The integral method gives output for C_f, θ, δ and δ^*. With U_e, one can form the pressure gradient parameters Λ and λ and thus create a suitable Pohlhausen polynomial velocity profile. See Sec. 2-3 in Schetz (1993) for details. A cubic profile of this type is coded into *ILBLI*. The user must input the initial value of λ.

Example Laminar Implicit Numerical Method Problem: Consider 2D laminar flow of a fluid with a kinematic viscosity $\upsilon = 2.0\text{x}10^{-4}$ m^2/s at $U_{inf} = 10.0$ m/s over a surface that is a flat plate from the leading edge to $x = 1.0$ m. At that station, a ramp begins that produces an inviscid velocity distribution $U_e(x) = 10.5 - x/2$, m/s. Calculate the boundary layer development over this surface up to $x = 2.0$ m. Does the flow separate?

Solution:

This is the same flow problem solved with the Thwaites-Walz integral method using the code *WALZ*. Now, we can apply the implicit numerical method in code *ILBLI* to this problem for comparison in terms of accuracy of the predictions and computational effort required.

Since the first part of the problem is flow over a flat plate ($\lambda = 0$), the Blasius solution can be used to obtain "initial" conditions at $x = 1.0$. Thus, the numerical calculation will begin at $x = 1.0$ and go to $x = 2.0$. We must again provide input data for the kinematic viscosity as υ = 2e-4 and the freestream velocity as $U_{inf} = 10.0$. The Blasius solution at $x = 1.0$ gives the boundary layer thickness, *delta* = 0.0224, the displacement thickness, *delta** = 0.00771, the momentum thickness, *theta* = 0.00297 and the skin friction coefficient, $C_f = 0.00297$. Choose 21 points across the initial boundary layer ($dy = 0.00112$) and add about 80 points above that to give *MMAX* = 100. Since the implicit method is unconditionally stable, no stability criterion need be followed, and we can select the streamwise step size *dx* and the number of points *NMAX* based only on accuracy considerations. Select *NMAX* = 41 to give $dx = 0.025$, which is about the size of the initial boundary layer thickness.

Initial profiles for u and v are required. For simplicity, this default case uses a cubic polynomial velocity profile (with $\lambda = 0$ for this case) for u and $v = 0$ as adequate approximations to the exact Blasius solution which would require tabular input for the profiles.

Since this case has a linear edge velocity variation, velocities at only two points, at x_{init} and x_{fin} need be specified. Again, U_e is normalized with U_{inf} in the code.

All of this can be seen in the window shown in Fig. 3, where the input information is displayed On the left are two panels containing the input data for this case. One can change the input data in the far left panel by selecting what item is to be changed with the menu button, entering the new value in the slot below and pushing *SET*. The input data in the next panel to the right can be changed in a similar manner. In order to change the dimensionless edge velocity distribution, U_e/U_{inf}, all that is required is to enter or modify pairs of values for x/L and U_e/U_{inf} in the panel below and push *SET*.

One can then press the *START* button and watch the skin friction, integral quantities and the u/U_e velocity profile develop. The window given in Fig. 4 shows the results about halfway through the calculation. Tabular values of the output can be accessed in the panel indicated.

PROGRAM *MOSES*

Here, we present a code for the **MOSES** integral method for 2D, incompressible, turbulent boundary layers. Refer to Sec. 7-7 and Appendix B in Schetz (1993) for the theory behind the method and the original *FORTRAN* code. A default case corresponding to the example problem shown below is included. This is the same as the original worked example.

Example Turbulent Integral Method Problem: Consider 2D turbulent flow of a fluid with a kinematic viscosity $\upsilon = 1.0\text{x}10^{-5}$ m^2/s at $U_{inf} = 10.0$ m/s over a surface that is a flat plate from x = 0.0 to 5.0 m. Then, a ramp begins that produces an inviscid velocity distribution $U_e(x) = (15.0–x)$ m/s. Calculate the boundary layer to $x = 7.0$ m and determine if the flow separates.

Solution:

We must input υ = 1e-5. Since the first part of the flow is over a flat plate, the simple integral solution (see Sec. 7-7 in Schetz, 1993) can be used giving $\delta = 0.0857$ m. Also, at the initial station the pressure gradient parameter $\beta = 0$.

Pick *NMAX* = 21 corresponding to dx = 0.10 m, which is about the size of the initial boundary layer thickness.

This case has a linear edge velocity variation, so velocities can be specified at only two points, at x_{init} and x_{fin}.

The two panels on the left in Fig. 5 contain the input data for this case. The input data in the far left panel can be changed by selecting the item with the menu button, entering the new value in the slot below and pushing *SET*. The input data in the next panel to the right can be changed in a similar manner. The user can modify the dimensionless edge velocity distribution,

U_e/U_{inf}. All that is necessary is to enter or modify sets of values for x and U_e/U_{inf} in the panel below and push *SET*.

Now, press the *START* button and watch the skin friction and the integral quantities develop in the graphs on the right. The window given in Fig. 6 shows the results about halfway through the calculation. Tabular values of the output can be accessed in the panel indicated. One can scroll up and down in the table with the slider bar and left and right with the cursor.

PROGRAM *ITBL*

The fourth code implements an implicit numerical solution method for **I**ncompressible, **T**urbulent **B**oundary **L**ayer flows with three options for the turbulence model: 1) mixing length model, 2) eddy viscosity model and 3) Prandtl TKE model. See Secs. 7-9 and 7-10 in Schetz (1993) for details of the development (and the original *FORTRAN* code in Appendix B). For simplicity, a uniform grid spacing in the normal direction is employed. Thus, a large number of grid points in that direction must be used to obtain accurate results. Use at least 500 points across the initial boundary layer. A default case corresponding to the following example problem is shown. This is the same as the original worked example, recast here in *SI* units. The equations are solved in dimensionless form using the freestream velocity, U_{inf} , a reference length, L, (taken as 1.0 for convenience), and reference viscosity and density, μ_{inf} and ρ_{inf}. The input must be in that form, and the output is given in that form.

Profiles for u and v at the initial station are required. For simplicity, we have adopted the Law of the Wake velocity profile for u and $v = 0$ as approximations. The user must supply the value of the Clauser pressure gradient parameter, $\beta \equiv (\delta^*/\tau_w) dp/dx$, at the initial station so the Law of the wake can construct a suitable profile.

Example Turbulent Numerical Method Problem: Consider 2D turbulent flow of a fluid with a kinematic viscosity $\upsilon = 9.3x10^{-7}$ m^2/s ($1.0x10^{-5}$ ft^2/s) at $U_{inf} = 3.049$ m/s (10.0 ft/s) over a flat plate from $x = 1.524$ to 1.829 m (5.0 to 6.0 ft.) Use all three turbulence models and compare the results.

Solution:

The simple integral solution (see Sec. 7-7 in Schetz, 1993) can be used to obtain "initial" conditions at $x = 1.524$. Thus, the numerical calculation will begin at $x_{init} = 1.524$ and go to $x_{fin} = 1.829$ m. We must again provide input data for the kinematic viscosity as $\upsilon = 9.3e-7$ m^2/s and the freestream velocity as $U_{inf} = 3.049$ m/s. The integral solution at $x = 1.524$ m gives the boundary layer thickness, *delta* = 0.0261 m, and the Blasius turbulent skin friction law gives $C_f = 0.00267$. Choose 601 points across the initial boundary layer (*dy* = 4.35e-5) and add points above that to give *MMAX* = 999. Pick *NMAX* = 101 to give *dx* = 0.0030.

We adopt the Law of the Wake velocity profile (with $\beta = 0$ for this case) for u and $v = 0$ as adequate approximations.

Since this case is a constant edge velocity variation, velocities at only two points, at x_{init} and x_{fin} need be specified. Again, U_e is normalized with U_{inf} in the code.

Lastly, the turbulence model must be selected. The default value is 1 which indicates a mixing length model.

This is all displayed in the window in Fig. 7. To change the input data in the far left panel, select what item is to be changed with the menu botton, enter the new value in the slot below and push *SET*. It is easy to change the turbulence model and compare the results to those for the mixing length model. The input data in the next panel to the right can be changed in a similar manner as for *ILBLI*.

Press the *START* button and watch the skin friction, integral quantities and the U/U_e velocity profile develop. The window given in Fig. 8 shows the results about halfway through the calculation. The initial oscillations in skin friction are because the assumed initial velocity profiles are not exact. Tabular values of the output can be accessed in the panel.

DISCUSSION

All of these programs make use of tablular input (of edge velocity and coordinates) and tablular output (of boundary layer parameters and profiles). Data may be copied into or out of these tables using standard cut/paste/copy operations. For example, results may be pasted into *Excel* or a similar program so that the student can plot them in a presentable form.

This compatibility with the 'clipboard' and other native applications is a very important element of the programs, since it largely overcomes the limitation of *JAVA* programs not being able to read or write files on the hard disk. It also opens up the possibility of developing a suite of applets that the student can use in combination to investigate the solution to more general problems. For example, we are in the process of developing a panel code applet and an applet to determine transition location given laminar boundary layer parameters as a function of streamwise distance. Given a geometry in the form of a paneling scheme, the panel code will provide edge velocity and coordinates as table output, and these may then be pasted into the WALZ program and a laminar boundary layer computed. Results from WALZ may then be pasted into the transition calculator whose output is used to initialize one of the turbulent boundary layer applets.

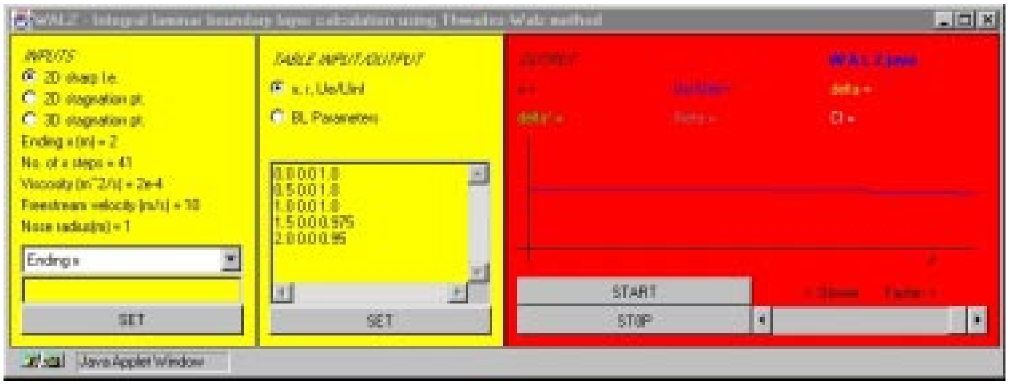
The current suite of codes can be found under Boundary Layer Applets at: http://www.engapplets.vt.edu. Primary candidates for additional codes in the near future are one for numerical solution of compressible, laminar boundary layers and one for turbulent jets and wakes.

ACKNOWLEDGEMENT

William Devenport acknowledges the support of the National Science Foundation under grant DUE-9752311

REFERENCES

Schetz, J.A., 1993, Boundary layer Analysis, Prentice Hall, Englewood Cliffs, NJ.



WALZ - Integral laminar boundary layer calculation using Thwaites-Walz method

INPUTS

● 2D sharp l.e.
○ 2D stagnation pt.
○ 3D stagnation pt.

Ending x (m) = 2
No. of x steps = 41
Viscosity (m^2/s) = 2e-4
Freestream velocity (m/s) = 10
Nose radius(m) = 1

Ending x

TABLE INPUT/OUTPUT

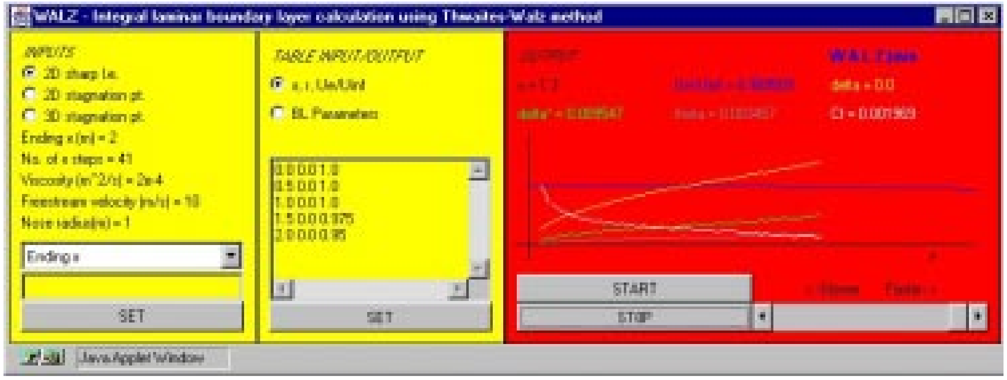
● x, r, Ue/Uinf
○ BL Parameters

```
0.0 0.0 1.0
0.5 0.0 1.0
1.0 0.0 1.0
1.5 0.0 0.975
2.0 0.0 0.95
```

OUTPUT

x =
Ue/Uinf =
delta =
delta* =
theta =
Cf =

Figure 1. Input data for the default case in the *WALZ* code.



WALZ - Integral laminar boundary layer calculation using Thwaites-Walz method

INPUTS

● 2D sharp l.e.
○ 2D stagnation pt.
○ 3D stagnation pt.

Ending x (m) = 2
No. of x steps = 41
Viscosity (m^2/s) = 2e-4
Freestream velocity (m/s) = 10
Nose radius(m) = 1

Ending x

TABLE INPUT/OUTPUT

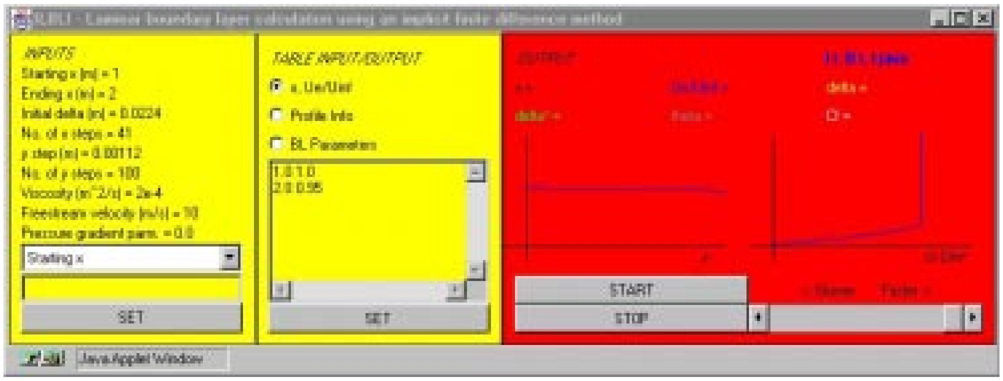
● x, r, Ue/Uinf
○ BL Parameters

```
0.0 0.0 1.0
0.5 0.0 1.0
1.0 0.0 1.0
1.5 0.0 0.975
2.0 0.0 0.95
```

OUTPUT

x = 1.2
Ue/Uinf = [illegible]
delta = 0.0
delta* = [illegible]
theta = [illegible]
Cf = 0.001969

Figure 2. Sample output for the default case from the *WALZ* code.



ILBLI - Laminar boundary layer calculation using an implicit finite difference method

INPUTS

Starting x (m) = 1
Ending x (m) = 2
Initial delta (m) = 0.0224
No. of x steps = 41
y step (m) = 0.00112
No. of y steps = 100
Viscosity (m^2/s) = 2e-4
Freestream velocity (m/s) = 10
Pressure gradient param. = 0.0

Starting x

TABLE INPUT/OUTPUT

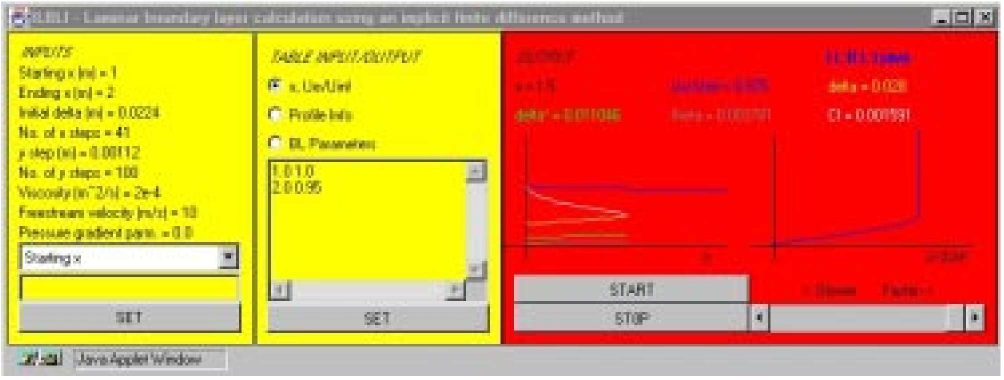
● x, Ue/Uinf
○ Profile Info
○ BL Parameters

```
1.0 1.0
2.0 0.95
```

OUTPUT

x =
Ue/Uinf =
delta =
delta* =
theta =
Cf =

Figure 3. Input data for the default case in the *ILBLI* code.



ILBLI - Laminar boundary layer calculation using an implicit finite difference method

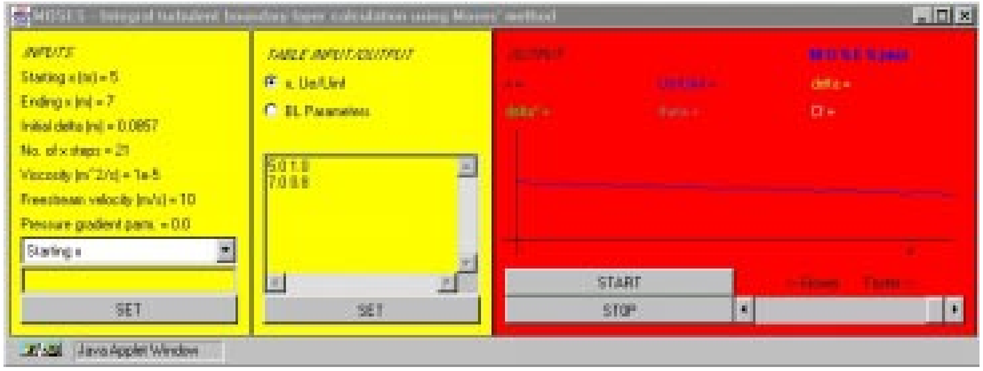
INPUTS
Starting x (m) = 1
Ending x (m) = 2
Initial delta (m) = 0.0224
No. of x steps = 41
y step (m) = 0.00112
No. of y steps = 100
Viscosity (m^2/s) = 2e-4
Freestream velocity (m/s) = 10
Pressure gradient parm. = 0.0

TABLE INPUT/OUTPUT
● x, Ue/Uinf
○ Profile Info
○ BL Parameters

```
1.0 1.0
2.0 0.95
```

OUTPUT
x = 1.5
delta = 0.028
delta* = 0.011046
Cf = 0.001591

Figure 4. Sample output for the default case from the *ILBLI* code.



MOSES - Integral turbulent boundary layer calculation using Moses' method

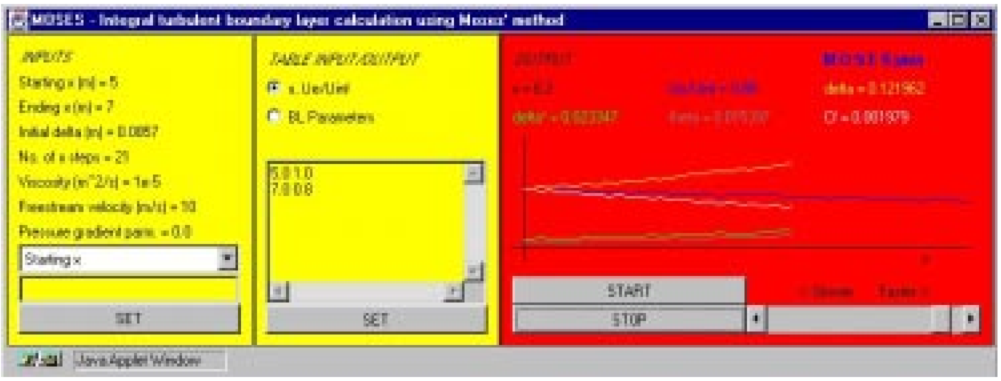
INPUTS
Starting x (m) = 5
Ending x (m) = 7
Initial delta (m) = 0.0857
No. of x steps = 21
Viscosity (m^2/s) = 1e-5
Freestream velocity (m/s) = 10
Pressure gradient parm. = 0.0

TABLE INPUT/OUTPUT
● x, Ue/Uinf
○ BL Parameters

```
5.0 1.0
7.0 0.8
```

OUTPUT
x =
Ue/Uinf =
delta =
delta* =
theta =
Cf =

Figure 5. Input data for the default case in the *MOSES* code.



MOSES - Integral turbulent boundary layer calculation using Moses' method

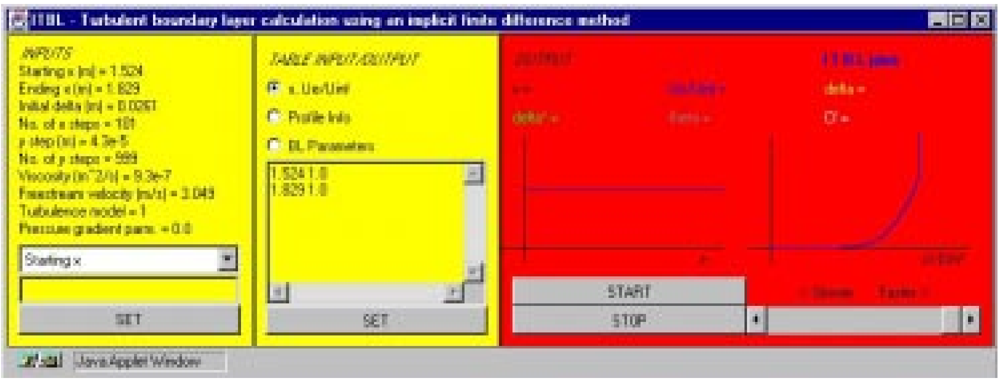
INPUTS
Starting x (m) = 5
Ending x (m) = 7
Initial delta (m) = 0.0857
No. of x steps = 21
Viscosity (m^2/s) = 1e-5
Freestream velocity (m/s) = 10
Pressure gradient parm. = 0.0

TABLE INPUT/OUTPUT
● x, Ue/Uinf
○ BL Parameters

```
5.0 1.0
7.0 0.8
```

OUTPUT
x = 6.2
delta = 0.121962
delta* = 0.022347
Cf = 0.001979

Figure 6. Sample output for the default case from the *MOSES* code.



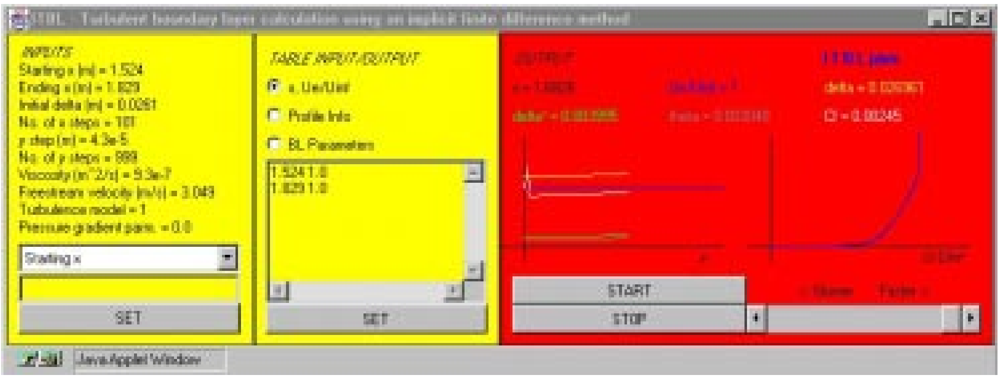
INPUTS
Starting x (m) = 1.524
Ending x (m) = 1.829
Initial delta (m) = 0.0261
No. of x steps = 101
y step (m) = 4.3e-5
No. of y steps = 999
Viscosity (m^2/s) = 9.3e-7
Freestream velocity (m/s) = 3.049
Turbulence model = 1
Pressure gradient para. = 0.0

Starting x

1.524 1.0
1.829 1.0

x =
Ue/Uinf =
delta =
delta* =
theta =
Cf =

Figure 7. Input data for the default case in the *ITBL* code.



INPUTS
Starting x (m) = 1.524
Ending x (m) = 1.829
Initial delta (m) = 0.0261
No. of x steps = 101
y step (m) = 4.3e-5
No. of y steps = 999
Viscosity (m^2/s) = 9.3e-7
Freestream velocity (m/s) = 3.049
Turbulence model = 1
Pressure gradient para. = 0.0

Starting x

1.524 1.0
1.829 1.0

x = 1.8828
Ue/Uinf = 1
delta = 0.026361
delta* = 0.003565
theta = [illegible]
Cf = 0.00245

Figure 8. Sample output for the default case from the *ITBL* code.