Proceedings of the
5th ASME/JSME Thermal Engineering Joint Conference
March 15-19, 1999, San Diego, California

AJTE99-6229

HEAT TRANSFER CODES FOR STUDENTS IN *JAVA*

W.J. Devenport,* J.A. Schetz and Yu. Wang*****
Virginia Polytechnic Inst. and State Univ.
Blacksburg, VA
devenport@aoe.vt.edu

Keywords: Heat transfer education, Web-based teaching aids, JAVA.

ABSTRACT

Simple *JAVA* computer codes for student use solving the homework problems in an undergraduate heat transfer text on a PC have been developed. *JAVA* was selected so that the codes could be made available in a platform-independent form on the Internet. At this point, there are four codes for: 1) steady, conduction in two dimensions by finite differences, 2) unsteady conduction in two dimensions by finite differences, 3) laminar boundary layers with heat transfer by an integral method, and 4) turbulent boundary layers with heat transfer by an integral 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.

NOTATION

c_p	specific heat
C_f	skin friction coefficient
$H \equiv \delta^*/\theta$	shape factor
k	thermal conductivity
Nu	Nusselt number
p	pressure
Pr	Prandtl number
q_w	wall heat flux
$r(x)$	body thickness distribution
Re	Reynolds number
$s(x)$	surface distance
St	Stanton number
T	temperature
$T_e(x)$	edge temperature distribution
$T_w(x)$	wall temperature distribution
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
θ_c	conduction thickness
μ	viscosity
ν	kinematic viscosity
ρ	density
τ_w	wall shear

*Associate Professor of Aerospace and Ocean Engineering, Member ASME
** J. Byron Maupin Professor of Aerospace and Ocean Engineering, Fellow ASME
*** Graduate Student

INTRODUCTION

In this paper, we present simple *JAVA* computer codes that are intended for student use solving the homework problems in a standard undergraduate heat transfer text such as Holman (1986) or White (1988) and other similar problems on a PC or Work Station. The *JAVA* language (see www.javasoft.com) 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 formulation, logic and programming as simple as possible so that the student can easily grasp the flow of the calculations. Thus, primitive variables (T,u,v); (x,y,t) are employed with no transformations.

At this point in the development, there are four separate codes for: 1) steady, heat conduction in two dimensions by finite differences, 2) unsteady heat conduction in two dimensions by finite differences, 3) laminar boundary layers with heat transfer by an integral method, and 4) turbulent boundary layers with heat transfer by an integral method. All the codes presume constant values of the density and thermophysical properties, so they are limited to low-speed cases with modest temperature differences. 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.

These codes are meant to relieve the student of the time consuming burden of writing, modifying and/or debugging numerous *FORTRAN* or other similar codes during a heat transfer course. The hope is to thereby leave sufficient time and energy for the working of actual heat transfer problems of reasonable complexity to build understanding and intuition. The student is not, however, relieved of the burden to think. One should always estimate the answer the computer solution is expected to yield. For example, one could use the exact solution for a simpler heat conduction problem to crudely estimate the level of the surface heat flux to be expected in a more complicated case. The student is the analyst, not the computer, and he or she is responsible for producing and interpreting the correct answer.

Conduction heat transfer problems require specification of the geometry and boundary and initial conditions. Thermophysical property values of thermal conductivity, k, specific heat, c_p ,and density, ρ, are needed. Next, the grid must be created.

For any boundary layer convection heat transfer problem, one must specify the fluid through density, ρ, conductivity, k, specific heat, c_p , and viscosity, μ, or perhaps just kinematic viscosity, ν. The next information needed would be the freestream velocity U_{inf} and the inviscid edge velocity distribution, $U_e(x)$. Also, the wall temperature, $T_w(x)$, or the wall heat transfer distribution and T_{inf} and $T_e(x)$ are required.

Finally, the computational region and grid must be selected. The differential methods require the height and the length of the computational region. The codes here are all based on untransformed and unmapped forms of the equations for simplicity. Thus, cartesian grids are used throughout. For heat conduction problems, the computational region must be at least as large as the body of interest in all dimensions. The grid spacing dx and dy must be selected so that the geometry and the variations in the boundary conditions can be accurately represented. In the interests of simplicity, we have allowed only geometries where the boundaries fall on grid points. It is possible to adequately represent very complicated geometries within that restriction by employing a fine grid. For the integral methods for heat convection included here, one need only pick the length of the flow of interest and the streamwise grid size, dx. That grid must be fine enough to accurately represent the variations in the boundary conditions such as $U_e(x)$, $T_e(x)$ and $T_w(x)$.

PROGRAM *STDYCOND*

This program treats **ST**ea**DY COND**uction heat transfer in two spatial directions by a numerical method. The thermal conductivity is taken as constant. Central finite differences are used to replace the differentials in LaPlaces Equation which governs such problems. This results in a system of algebraic equations, one at each interior grid point relating the temperature at that point to that at four neighboring points. The system is solved by the Gauss-Seidel iteration technique. See Holman (1986) or White (1988) for a description of the general approach.

The user can specify either temperature or heat flux as a boundary condition at each grid point on the boundary.

A default input set is included for the following problem.
Example Steady, 2D Heat Conduction Problem: The problem is steady heat conduction in a 2X1 ellipse of aluminum. The temperature on the top half of the boundary is 100C, and that on the bottom half is 50C. What is the temperature pattern in the interior?
Solution:

First, note that the solutions for steady heat conduction problems with constant properties are actually independent of the values of the thermophysical properties; LaPlaces equation doesn't contain any coefficients.

The 2X1 ellipse is represented in this sample input with a relatively coarse grid to keep the file size modest. Look at Fig. 1. When you launch the Applet, the input geometry and boundary conditions for the temperature are shown in the window in the lower left-hand corner. The first two entries are the grid spacings in the x and y directions. This is followed by four columns of input with x, y and T for each boundary point. The 1's in the last column indicate that this is a temperature boundary condition. (Use 2 for a heat flux boundary condition.) The region and the grid are displayed above the input panel. The solution is iterative, and the initial guess for the temperature in the interior is given below this window. Push the *COMPUTE* button at the bottom and see the results in the simple isotherm plot in the upper right-hand corner. The tabular output is in the window in the lower right-hand corner. The tabular output can also be easily copied into *EXCEL* for printing and plotting. Below that window, one can see the number of iterations required to obtain a converged solution. One can change the initial guess for the temperature in the interior and see how that affects the iterations required.

It is very easy to change the temperatures on the boundary and recompute the temperature distribution. To change the grid spacing and/or the geometry, it is necessary for the user to create an input file working offline. An *EXCEL* spreadsheet is very convenient for that task. One can then "cut and paste" the new input data into the input panel.

PROGRAM *UNSTDYCOND*

Here, we solve **UNST**ea**DY COND**uction heat transfer problems in two spatial directions by a numerical method. The thermal conductivity is taken as constant. Central finite differences are used to replace the spatial differentials in the 2D Heat Equation, which governs such problems. The time derivative is approximated with an implicit finite difference representation. Refer to Holman (1986) or White (1988) for details of the general approach.

The user can specify either temperature or heat flux as a boundary condition at each grid point on the boundary. Since these cases are unsteady, an initial temperature distribution at every grid point on the boundary and in the interior must be given.

A default input set is included for the problem described below.

Example Unsteady, 2D Heat Conduction Problem: Let us consider an unsteady equivalent of the steady problem treated above. The problem is now unsteady heat conduction in a 2X1 ellipse of aluminum. The temperature over the whole region is 100C at time $t = 0$. For $t > 0$, the temperature on the top half of the boundary is held fixed at 100C, and that on the bottom half is set to 50C. What is the temperature pattern in the interior as a function of time? How long does it take to approach steady state?

Solution:

In this case, the thermophysical properties are needed, because the thermal diffusivity, $k/c_p\rho$, enters into the Heat Equation. The values for aluminum at about 75C are shown in the column at the lower left of Fig. 2. The geometry and boundary conditions are given in the panel to the left in the same format as for the steady calculation above. The region and the grid are displayed above the input panel.

Push the *RUN* button at the bottom and watch the results develop as a function of time in the simple isotherm plot in the upper right-hand corner. The tabular output is in the window in the lower right-hand corner. The elapsed time is shown in the line below the output table. The results shown in Fig. 2 are for a time well before steady state is reached for this problem. The *FORWARD* button at the bottom advances the solution one time step at a time.

There is an analogy between iterations to find a steady solution as in the sample problem above and time steps in an unsteady problem approaching a steady-state solution as in this case. If the initial "guess" used in obtaining the steady solution is the same as the initial condition for the unsteady problem, the correspondence is close and one could watch the solutions develop iteration-by-iteration or time step by time step and compare the behavior.

PROGRAM *WALZHT*

This program is an implementation of the Thwaites-**WALZ** and Smith-Spalding integral methods for incompressible, laminar boundary layer flows with **H**eat **T**ransfer. Refer to Sec. 2-3-2 in Schetz (1993) and Smith and Spalding (1958) for a description of the technique. The method is limited to flows with constant thermophysical properties and a constant wall temperature. It can treat cases with sharp or blunt leading edges, planar or axisymmetric geometries and arbitrary inviscid velocity variations.

A default input set is included for the following flow problem.

Example Laminar Integral Method Problem: Consider 2D laminar flow of a fluid with a kinematic viscosity $\nu = 1.6\text{x}10^{-5}$ m^2/s, $c_p = 1005$ J/kg/K, $\rho = 1.2$ kg/m^3 and Prandtl number $Pr = 0.72$ at $U_{inf} = 2.0$ 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 $U_e(x) = 2.1 - x/10$, m/s. This is an adverse pressure gradient, since U_e is decreasing so that p increases. Choose the wall to freestream temperature difference, $T_w - T_e = 20$C. Calculate the boundary layer development over this surface up to $x = 2.0$ m. Does the flow separate? Note how the dimensionless wall shear, C_f, and dimensionless heat transfer, Nu, vary in the constant pressure and varying pressure regions along the surface.

Solution:

We must provide input data for the kinematic viscosity as $\nu = 1.6\text{e-}5$, Prandtl number $Pr = 0.72$, $c_p = 1005$, $\rho = 1.2$ and the freestream velocity as $U_{inf} = 2.0$. 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 $r(x) = 0.0$. Also, one must choose either a planar or axisymmetric geometry.

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 wall to edge temperature distribution $T_w - T_e$ is specified in the same way.

The window in Fig. 3 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} , the body shape and/or the temperature difference. All that is required is to enter or modify sets of values for x, r, U_e/U_{inf} and T_w-T_e 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 integral quantities and the skin friction and heat transfer results develop in the graphs on the right. Use the panel in the upper right hand corner to select which sets of results are displayed. The window given in Fig. 3 shows the results at the end of 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. The tabular output can also be easily copied into *EXCEL* for printing and plotting.

PROGRAM *MOSESHT*

Here, we present a code for the **MOSES** integral method for 2D, incompressible, turbulent boundary layers extended to include **H**eat **T**ransfer with the approximate integral energy equation method of Ambrok as described in Kays and Crawford (1980). Refer to Sec. 7-7 in Schetz (1993) for the theory behind the original Moses method.

The method is limited to flows with constant thermophysical properties. It can treat cases with arbitrary inviscid velocity and wall temperature variations.

Example Turbulent Integral Method Problem: Consider 2D turbulent flow of a fluid with a kinematic viscosity $\nu = 1.0\text{x}10^{-5}$ m^2/s, Prandtl number $Pr = 5$, $c_p = 4187$, $\rho = 1.2$ at $U_{inf} = 10.0$ m/s over a surface that is a flat plate from $x = 0.0$ to 5.0 m. Calculate the boundary layer to $x = 7.0$ m assuming a simple inviscid velocity distribution $U_e(x) = 10$ m/s = constant. The temperature difference is 10K; what is the heat transfer?

Solution:

We must input ν = 1e-5, Pr = 5, c_p = 4187, and ρ = 1.2. 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. Take St = 6.89e-4 at the initial station. 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 simple edge velocity variation, so velocities and temperature differences can be specified at only two points, at x_{init} and x_{fin}.

The two panels on the left in Fig. 4 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} and/or the temperature difference, T_w-T_e. All that is necessary is to enter or modify sets of values for x and U_e/U_{inf} and T_w-T_e in the panel below and push *SET*.

Now, press the *START* button and watch the integral quantities and the skin friction, heat transfer develop in the graphs on the right. Use the panel in the upper right hand corner to select which sets of results are displayed. The window in Fig. 4 shows the results at the end of the sample calculation. Again, tabular values of the output can be accessed in the panel indicated, and one can scroll up and down in the table with the slider bar and left and right with the cursor.

DISCUSSION

All of these programs make use of tabular input (of edge velocity and coordinates) and tabular 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 *WALZHT* program and a laminar boundary layer computed. Results from *WALZHT* 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 Heat Transfer Applets at: http://www.engapplets.vt.edu. Primary candidates for upgrades and further codes in the near future are the addition of heat generation to the conduction codes new codes for numerical solution of compressible, laminar and turbulent boundary layers and one for radiation heat transfer.

ACKNOWLEDGEMENT

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

REFERENCES

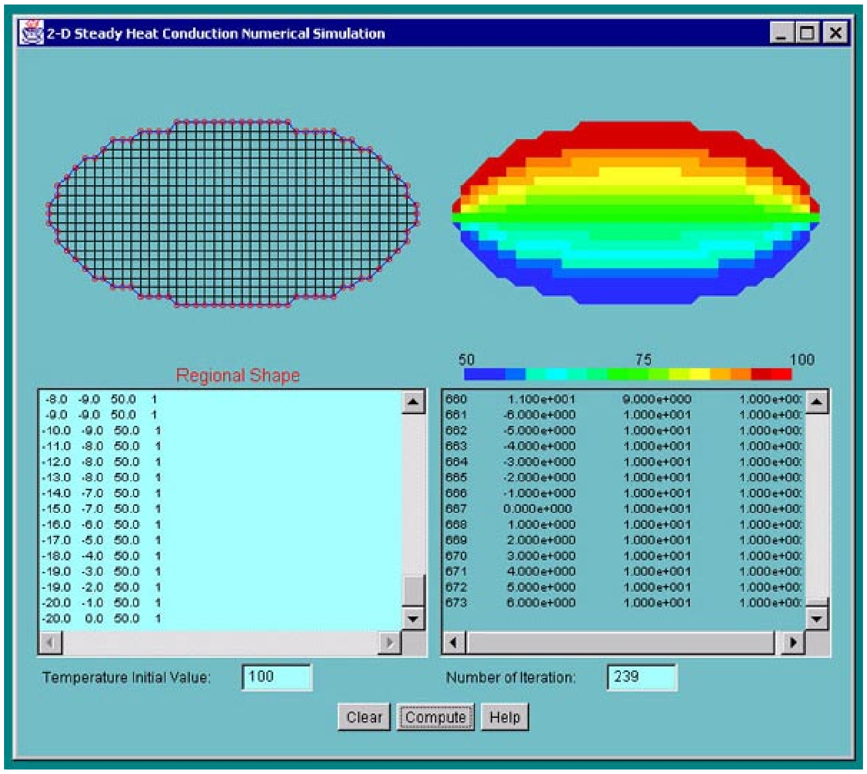
Holman, J.P., 1986, Heat Transfer, McGraw-Hill, New York, NY.

Kays, W.M. and Crawford, M.E., 1980, Convective Heat and Mass Transfer, McGraw-Hill, New York, NY.

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

Smith, A.G. and Spalding, D.B., 1958, “Heat Transfer in a Laminar Boundary Layer with Constant Properties and Constant Wall Temperature,“ Journal of the Royal Aeronautical Society, Vol. 62, pp. 60-64.

White, F.M., 1988, Heat and Mass Transfer, Addison-Wesley Publishing Co., Reading, MA.



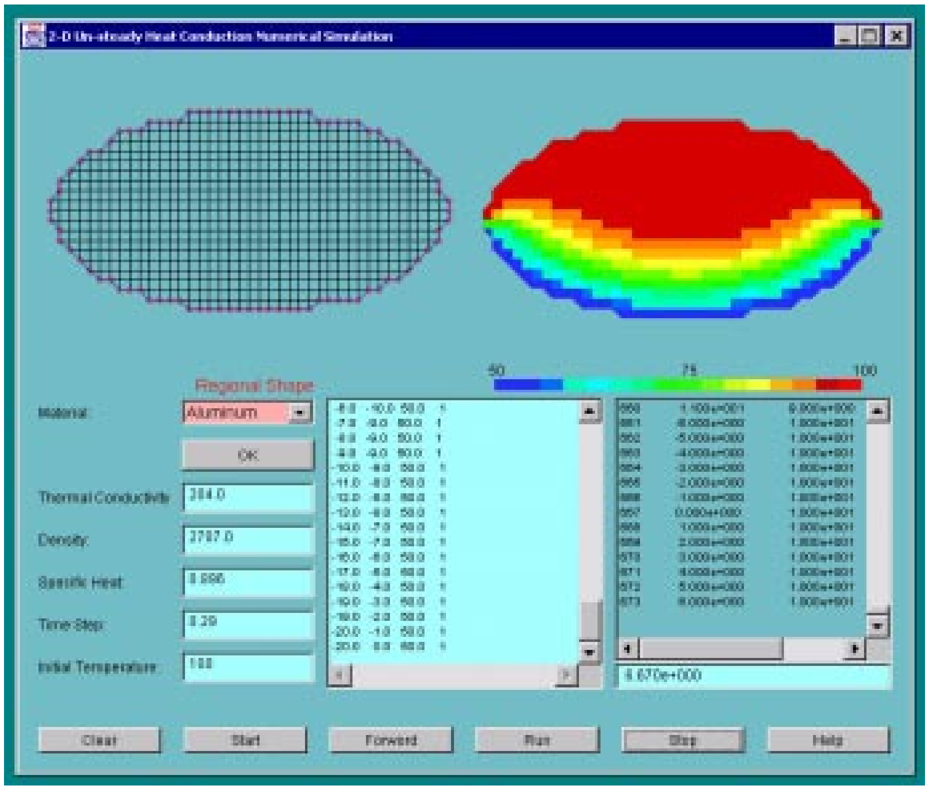
```
 -8.0  -9.0  50.0   1
 -9.0  -9.0  50.0   1
-10.0  -9.0  50.0   1
-11.0  -8.0  50.0   1
-12.0  -8.0  50.0   1
-13.0  -8.0  50.0   1
-14.0  -7.0  50.0   1
-15.0  -7.0  50.0   1
-16.0  -6.0  50.0   1
-17.0  -5.0  50.0   1
-18.0  -4.0  50.0   1
-19.0  -3.0  50.0   1
-19.0  -2.0  50.0   1
-20.0  -1.0  50.0   1
-20.0   0.0  50.0   1
```

```
660    1.100e+001   9.000e+000   1.000e+00
661   -6.000e+000   1.000e+001   1.000e+00
662   -5.000e+000   1.000e+001   1.000e+00
663   -4.000e+000   1.000e+001   1.000e+00
664   -3.000e+000   1.000e+001   1.000e+00
665   -2.000e+000   1.000e+001   1.000e+00
666   -1.000e+000   1.000e+001   1.000e+00
667    0.000e+000   1.000e+001   1.000e+00
668    1.000e+000   1.000e+001   1.000e+00
669    2.000e+000   1.000e+001   1.000e+00
670    3.000e+000   1.000e+001   1.000e+00
671    4.000e+000   1.000e+001   1.000e+00
672    5.000e+000   1.000e+001   1.000e+00
673    6.000e+000   1.000e+001   1.000e+00
```

Temperature Initial Value: 100

Number of Iteration: 239

Figure 1. Sample input and output for the default case from the *STDYCOND* code.



Material: Aluminum

Thermal Conductivity: 204.0

Density: 2707.0

Specific Heat: 0.896

Time Step: 0.29

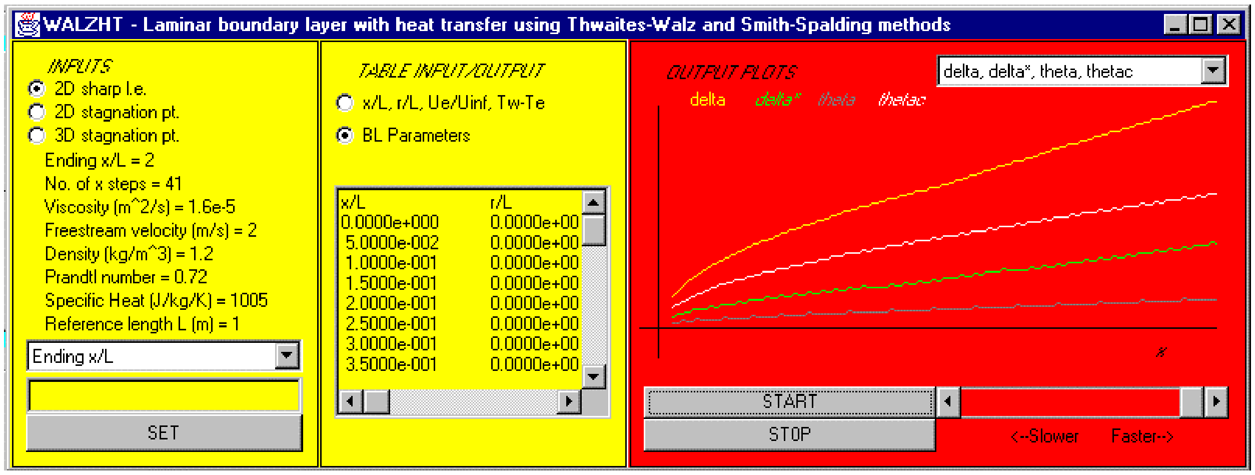
Initial Temperature: 100

```
 -6.0 -10.0  50.0   1
 -7.0  -9.0  50.0   1
 -8.0  -9.0  50.0   1
 -9.0  -9.0  50.0   1
-10.0  -9.0  50.0   1
-11.0  -8.0  50.0   1
-12.0  -8.0  50.0   1
-13.0  -8.0  50.0   1
-14.0  -7.0  50.0   1
-15.0  -7.0  50.0   1
-16.0  -6.0  50.0   1
-17.0  -5.0  50.0   1
-18.0  -4.0  50.0   1
-19.0  -3.0  50.0   1
-19.0  -2.0  50.0   1
-20.0  -1.0  50.0   1
-20.0   0.0  50.0   1
```

```
660    1.100e+001   9.000e+000
661   -6.000e+000   1.000e+001
662   -5.000e+000   1.000e+001
663   -4.000e+000   1.000e+001
664   -3.000e+000   1.000e+001
665   -2.000e+000   1.000e+001
666   -1.000e+000   1.000e+001
667    0.000e+000   1.000e+001
668    1.000e+000   1.000e+001
669    2.000e+000   1.000e+001
670    3.000e+000   1.000e+001
671    4.000e+000   1.000e+001
672    5.000e+000   1.000e+001
673    6.000e+000   1.000e+001
```

6.670e+000

Figure 2. Sample input and output for the default case from the *UNSTDYCOND* code.


WALZHT - Laminar boundary layer with heat transfer using Thwaites-Walz and Smith-Spalding methods

INPUTS

[✓ 2D sharp l.e.]
2D stagnation pt.
3D stagnation pt.
Ending x/L = 2
No. of x steps = 41
Viscosity (m^2/s) = 1.6e-5
Freestream velocity (m/s) = 2
Density (kg/m^3) = 1.2
Prandtl number = 0.72
Specific Heat (J/kg/K) = 1005
Reference length L (m) = 1

Ending x/L

TABLE INPUT/OUTPUT

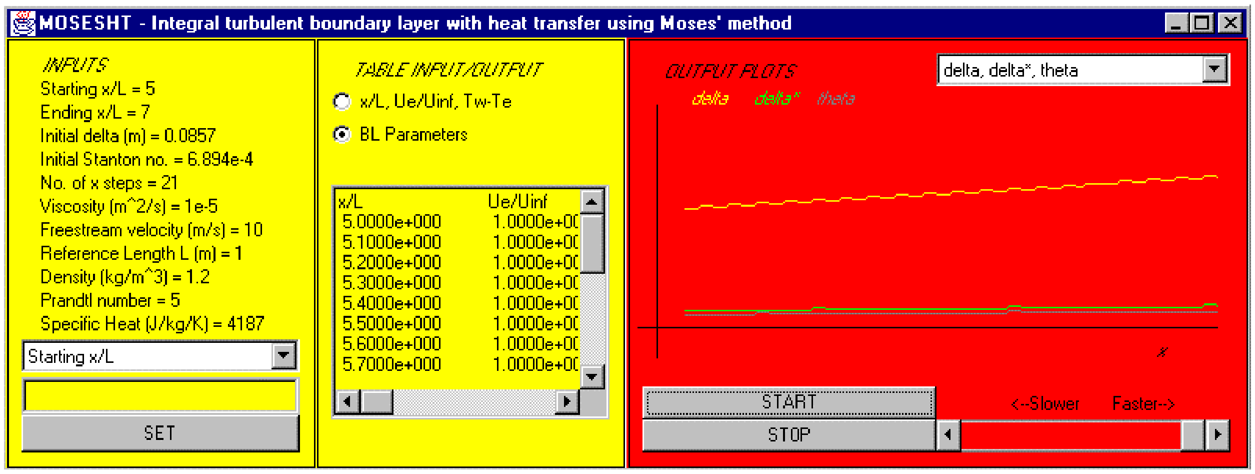
x/L, r/L, Ue/Uinf, Tw-Te
[✓ BL Parameters]

x/L	r/L
0.0000e+000	0.0000e+00
5.0000e-002	0.0000e+00
1.0000e-001	0.0000e+00
1.5000e-001	0.0000e+00
2.0000e-001	0.0000e+00
2.5000e-001	0.0000e+00
3.0000e-001	0.0000e+00
3.5000e-001	0.0000e+00

OUTPUT PLOTS

delta, delta*, theta, thetac

Figure 3. Sample input and output for the default case from the *WALZHT* code.


MOSESHT - Integral turbulent boundary layer with heat transfer using Moses' method

INPUTS

Starting x/L = 5
Ending x/L = 7
Initial delta (m) = 0.0857
Initial Stanton no. = 6.894e-4
No. of x steps = 21
Viscosity (m^2/s) = 1e-5
Freestream velocity (m/s) = 10
Reference Length L (m) = 1
Density (kg/m^3) = 1.2
Prandtl number = 5
Specific Heat (J/kg/K) = 4187

Starting x/L

TABLE INPUT/OUTPUT

x/L, Ue/Uinf, Tw-Te
[✓ BL Parameters]

x/L	Ue/Uinf
5.0000e+000	1.0000e+00
5.1000e+000	1.0000e+00
5.2000e+000	1.0000e+00
5.3000e+000	1.0000e+00
5.4000e+000	1.0000e+00
5.5000e+000	1.0000e+00
5.6000e+000	1.0000e+00
5.7000e+000	1.0000e+00

OUTPUT PLOTS

delta, delta*, theta

Figure 4. Sample input and output for the default case from the *MOSESHT* code.