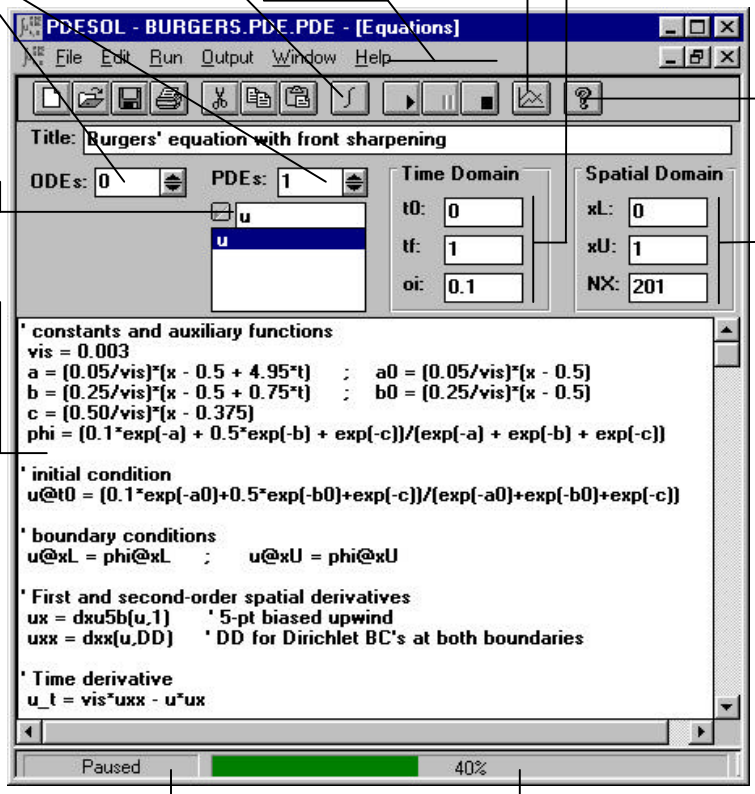
PDESOL v2.0 - Quick Reference Guide

Equation Window



Specify the number of partial differential equations (PDEs) here

Single-click access to **integrator specification window**

Single-click access to **customize output window**

Run/Resume, Pause, and Stop commands

Specify the initial time, final time, and output interval here

Specify the lower boundary, upper boundary, and number of grid points here in spatial dimensions

Specify the number of ordinary differential equations (ODEs) here

Specify the PDE dependent variables name here

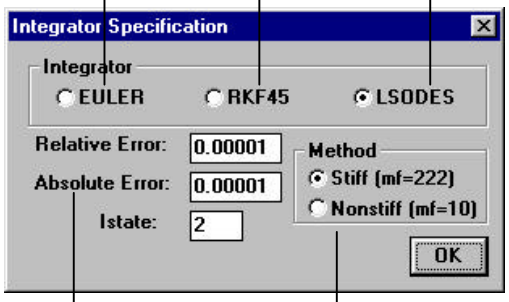
Specify the system of equations here using the exclusive PDESOL input language. Note the similarity of the statements with the mathematical formulation of the problem. There is no need to define arrays, or subscript variables. No compiler required. Thorough syntax checking prior to problem execution

On-line Help includes a tutorial, a detailed description of PDESOL commands and input language, guidelines for the specification of parameters, and a discussion of differential equations and the Method of Lines illustrated with PDESOL examples. The Help file also shows you how to easily solve numerical derivatives, integrals, and systems of nonlinear equations with PDESOL

Problem status indicator

Solution progress monitoring gauge

Integrator Specification Window



Implementation of the explicit Euler method. User specifies a fixed time step. Stability is not guaranteed. Recommended for debugging. By default, the time step is the output interval (oi). User can optionally specify intermediate Euler steps per output interval. Recommended when the use of very small output intervals becomes prohibitive.

Implementation of the Runge Kutta (RK) Fehlberg formulas. Explicit integrator with automatic adjustment of step size to satisfy user-specified error tolerances. Recommended for nonstiff problems

Implementation of backward differentiation formulas (BDFs). Implicit, variable order, and multistep integrator with automatic adjustment of step size to satisfy user-specified error tolerances. Recommended for stiff problems

Relative and absolute error tolerances (for RKF45 and LSODES only). Select them carefully. See guidelines in Help file

Stiff or nonstiff method specification (for LSODES only) See guidelines in Help file

Table Window

Tables have full spreadsheet functionality for additional post-processing of the solution. More than sixty built-in functions are available (see the Help file for details). You can have multiple views of the solution by inserting worksheets. Use Customize Table to configure the view.

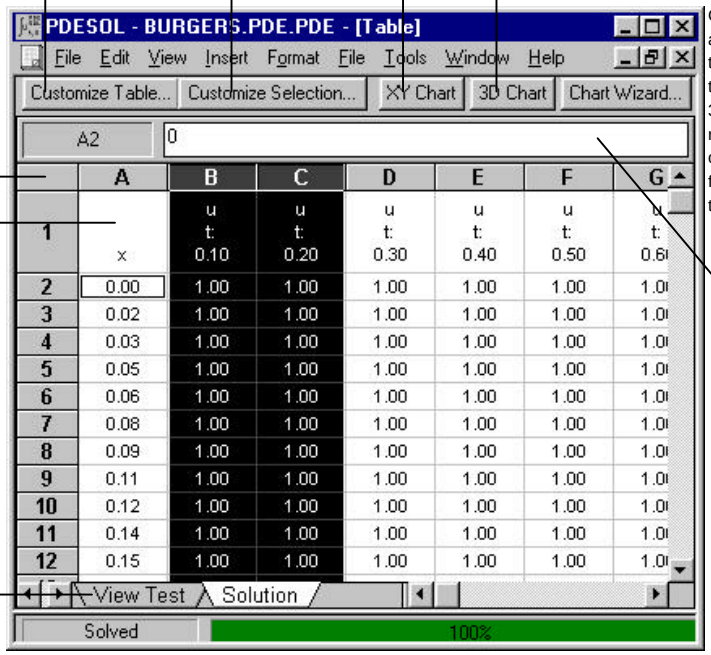
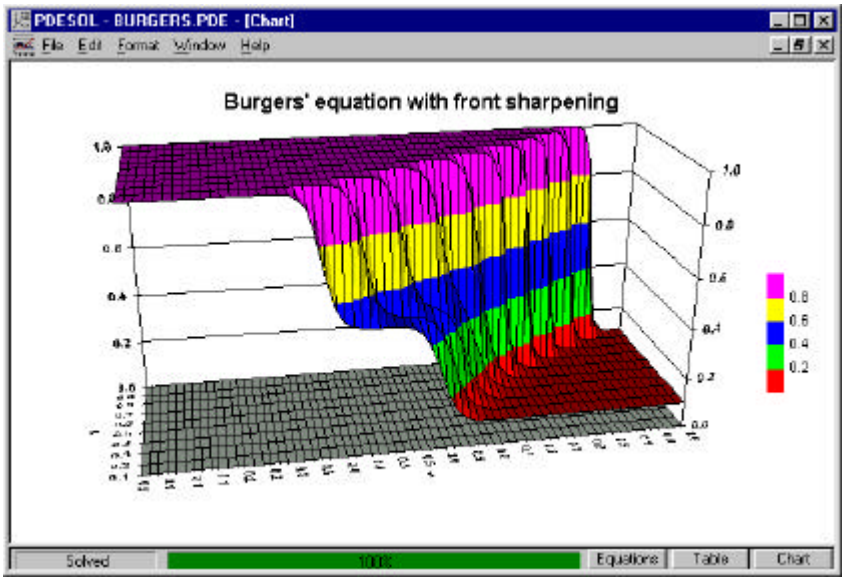


Chart Window



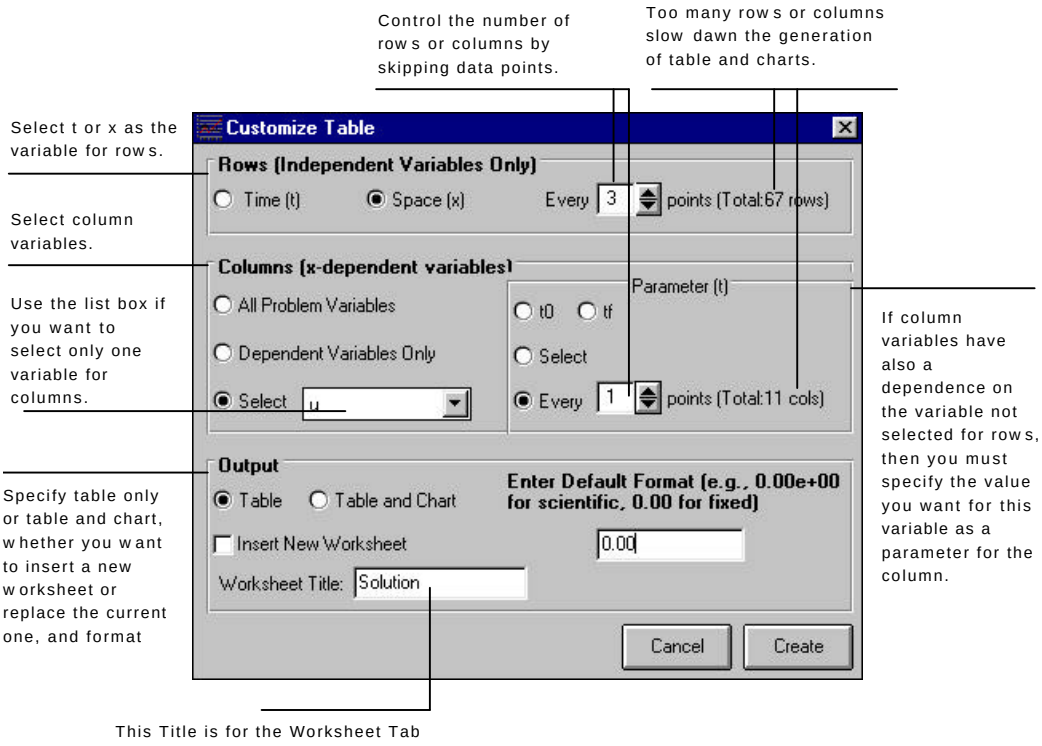
To format the chart, select items under the Format Menu or double-click a chart element to display the corresponding formatting window.

To rotate a 3D view, press CTRL -click and move the mouse.

You can also Copy and Paste charts in table worksheets.

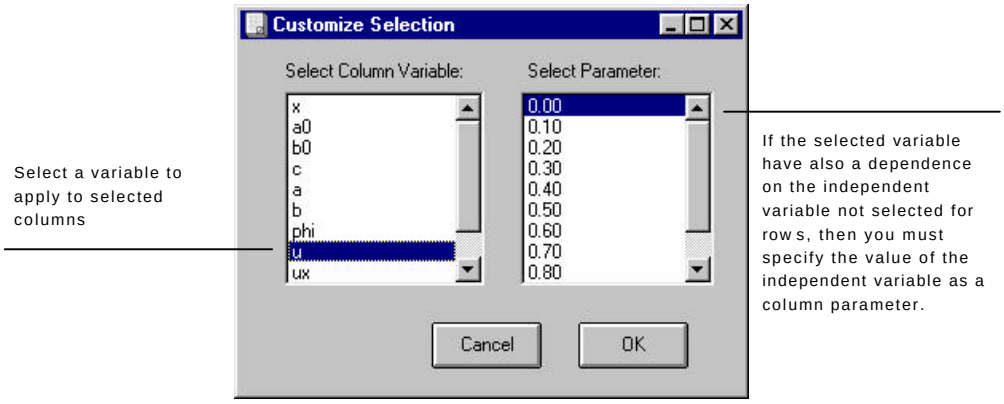
Customize Table Window

Use this window to quickly customize the output. All problem variables remain available for viewing. You can store multiple views of the solution by inserting worksheets in the tabular output window. Once you create a worksheet, you can change its configuration by using the customize selection window and by inserting or deleting columns.



Customize Table Selection Window

Use this window to quickly customize the output for selected worksheet columns. All problem variables with dependence on the rows variable are available for viewing in the selected columns.



PDESOL Input Language

Standard Operators

- **( )**, **+**, **-**, *****, **/**, **^**

Special Operators

- **@t0**: Initial condition for any dependent variable (e.g., u@t0 = u(t=t0))
- **@xL**: Lower boundary value of any variable with x-dependence (e.g., u@xL = u(x=xL))
- **@xU**: Upper boundary value of any variable with x-dependence (e.g., u@xU = u(x=xU))
- **dx**(e): five point centered approximation for first spatial (x) derivatives of any expression e with x-dependence. Other approximations are available.
- **dxu**(e,c): five point biased upwind approximation for first spatial (x) derivatives of any expression e with x dependence. Other approximations are available. Upwind approximations are recommended for strongly convective (hyperbolic) problems. c indicates the direction of upwind (a positive or negative value)
- **dxx**(e,ab): five point centered approximation for second spatial (x) derivatives of any expression e with x dependence. Other approximations are available. ab denote the type of lower and upper boundary condition. Possible values for a or b are D (for Dirichlet) and N (for Neumann). Thus, valid ab values are DD, DN, NN, or ND.
- **_t**: time(t) derivative of any dependent variable (e.g., u_t = u_t).

Intrinsic functions

- **sin**(a), **asin**(a), **sinh**(a), **cos**(a), **acos**(a), **cosh**(a), **tan**(a), **atan**(a), **tanh**(a)
- **exp**(a), **log**(a), **log10**(a)
- a>b (a<b): the result is the greater (smaller) of a,b. Similar to Max(), Min()
- **step**(a): if a > 0 the result is 1, otherwise the result is 0
- **lintc** ("*filename*", Var): linear interpolation of tabular data in *filename*. Var is the independent variable. Other interpolation functions are **lintl**() and **lintu**()

a and b can be a constant, f(t), f(x), or f(x,t)

Tip: you can build logical constructs by using the **step** function and the **>** or **<** operators

Syntax rules

- The editor is case sensitive
- Comments can be added by using an apostrophe (') preceding the comment, and can be placed to the right of the equation on a given line or anywhere on a separate line. Blank lines are also allowed
- Multiple statements in the same line are allowed, but they must be separated by a semicolon (**;**)
- **t** is the reserved name for time, and **x** is the reserved name for space
- In general, any variable in the right hand side (RHS) must be previously defined. There are no additional restrictions in the order of equations
- If the spatial derivative of a variable at either boundary is used (e.g., in a boundary condition such as u_x@xL), the spatial derivative must be previously defined (e.g., by u_x = dx(u))
- The equations should be arranged so that time derivatives appear on the left hand side (LHS) of the equation.

  Example: The equation

  $U_t + v\,U_x = 0$

  should be expressed as

  $U_t = -v\,U_x$

  and specified in PDESOL input language as

  ```
  U_t = -v*dxu(U,1)
  ```

  or

  ```
  U_x = dxu(U,1)
  U_t = -v*U_x
  ```

Tip: Initial conditions are denoted by adding the suffix @t0 to the dependent variable name (e.g., U@t0)

Tip: Boundary conditions are denoted by adding the suffix @xL or @xU to the variable name (e.g., U@xL)

PDESOL is distributed by:

NUMERICA
P.O. Box 670265
Dallas, TX 75367
USA

support@pdesol.com
www.pdesol.com