Athena Visual Studio Boundary Value Problem Tutorial

In this section we present extensions of differential-algebraic solvers from Initial-Value Problems (IVPs) to Initial-Boundary-Value Problems (IBVPs) with mixed partial differential and algebraic equations in a time like dimension t and one or more space dimensions. The technique used is the famous "method of lines", in which the IBVP is converted to an initial-value problem in the time like dimension by the discretization of the space dimension. This technique is extended to handle systems in which the boundary conditions change abruptly during the problem. Such changes may occur, for example, at known boundaries of catalytic regions along the time like coordinate, or at unknown t values defined implicitly through algebraic equations.

❖ Start **Athena Visual Studio**
❖ From the **File** menu select **New.**
❖ You are in the **Process Modeling** tab.
❖ Click **Modeling with Boundary Value Problems.**
❖ Select **A Blank Document** and click **OK.**
❖ Enter your model data, initial conditions and equations, and the Athena solver data and options as described in this tutorial.

When you are done:
❖ From the **File** menu click **Save.**
❖ Navigate to the folder where you wish to save and enter a proper filename for your model.
❖ From the **Build** menu click **Compile.**
❖ From the **Build** menu click **Build EXE.**
❖ From the **Build** menu click **Execute.**

Initial-Boundary-Value problems arise in the analysis, design and control of processes that are modeled in two dimensions, one of which can be traversed by forward integration. In this section we consider only two-dimensional systems; however, the approach described is also applicable in higher-dimensional problems. Examples can be found in chemical reactor engineering, combustion processes, atmospheric chemistry, industrial design and various biological systems. The complexity of realistic models makes it very difficult to determine the effects that small changes in their physical and chemical parameters would have on the predicted output of the process. Sensitivity analysis of such systems can reveal an abundance of information about the underlying mechanistic steps and provide information for model development, optimal experimental design and parameter estimation.

Boundary value models take the form:

$$\mathbf{F}\left(x,\mathbf{u}_x,\mathbf{u}_{xx};\mathbf{\theta}\right)=\mathbf{0} \quad \text{where} \quad \mathbf{u}_{xx}=\frac{1}{x^m}\frac{d}{dx}x^m\frac{d\mathbf{u}}{dx} \quad \text{and} \quad \alpha<x<\beta$$

Boundary Conditions

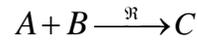$$\mathbf{F}\left(x,\mathbf{u}_x;\mathbf{\theta}\right)=\mathbf{0} \quad x=\alpha$$

$$\mathbf{F}\left(x,\mathbf{u}_x;\mathbf{\theta}\right)=\mathbf{0} \quad x=\beta$$

where $\mathbf{u}(\ )$ is a state vector of unknowns (usually temperature, pressure and composition), θ is a vector of known parameters pertinent to the process we are modeling, $\mathbf{u}_x(\ )$ represents the first order derivative of the state vector $\mathbf{u}(\ )$ with respect to the dimension x, and $\mathbf{u}_{xx}(\ )$ represents the second order derivative of the state vector $\mathbf{u}(\ )$ with respect to x (in the appropriate coordinate system). Boundary value models are ordinarily used to model steady-state reaction and diffusion. They can be solved using the Athena Visual Studio powerful damped Newton algorithm which is encoded in the **PDAPLUS** solver.

Tutorial: Reactor Modeling with Axial Dispersion

This example problem has been created to test the functionality of Athena Visual Studio in dealing with the solution of boundary value problems. Additional features such as, sensitivity analysis, parametric continuation and use of auxiliary variables will also be demonstrated.

Chemistry:

$$A + B \xrightarrow{\ \Re\ } C$$

This exothermal reaction is carried out in the liquid phase at a pressure level sufficiently high to avoid boiling. Reactant A is fed in excess, because reactant B should be totally converted at the reactor exit. The plant reactor is an adiabatic tubular reactor and its steady-state behavior can be described by two, dimensionless second order ordinary differential equations.

Plug Flow Reactor Model:

$$\Re = \exp\left[\gamma\left(1 - \frac{1}{T}\right)\right]C_B$$

$$-\upsilon\frac{\partial C_B}{\partial z} + \frac{1}{Pe_{mr}}\frac{\partial^2 C_B}{\partial z^2} - Da_r\Re = 0$$

$$-\upsilon\frac{\partial T}{\partial z} + \frac{1}{Pe_{hr}}\frac{\partial^2 T}{\partial z^2} + \Delta T_{adr}Da_r\Re = 0$$

$$z = 0 \quad C_B = C_{Bo} \quad T = T_{ro}$$

$$z = 1 \quad \frac{\partial C_B}{\partial z} = \frac{\partial T}{\partial z} = 0$$

We wish to perform the following tasks:

- ❖ Plot the concentration and reactor temperature as a function of reactor distance
- ❖ Perform a Continuation Analysis with respect to the Damköhler number
- ❖ Estimate the conversion of reactant by introducing an auxiliary variable:

$$X_B = 100 \times \frac{C_{B0} - C_B(z=1)}{C_{B0}}$$

The values and description of the parameters for this process are given in the table below:

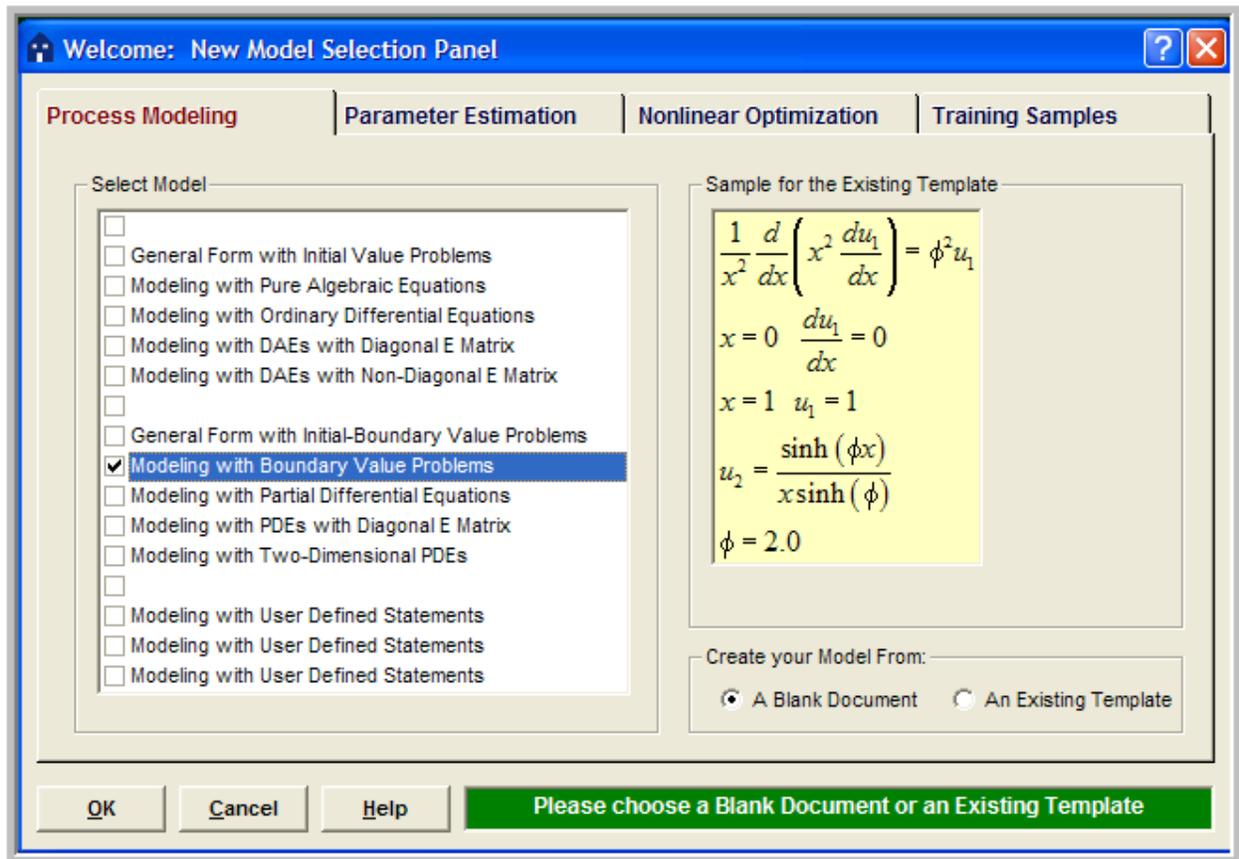| Model Parameters and Physical Properties	Description and Units
$\gamma = 20.0$	Dimensionless Activation Energy
$Da_r = 0.60 - 1.40$	Damköhler number range
$Pe_{mr} = 196.0$	Peclet number for mass dispersion
$Pe_{hr} = 42.0$	Peclet number for heat dispersions
$U_{htc} = 160.0$	Dimensionless heat transfer coefficient
$\Delta T_{adr} = 0.34$	Dimensionless adiabatic temperature rise
$\upsilon = 0.5$	Dimensionless fluid velocity
$\omega_h = 11.67$	Dimensionless heat capacity
$C_{Bo} = 0.50$	Dimensionless inlet reactant concentration
$T_{ro} = 0.95$	Dimensionless inlet reactor temperature

This sample tutorial is already precoded in Athena Visual Studio. If you do not wish to type the code on your own you may access it by doing the following:

- ❖ Open **Athena Visual Studio**
- ❖ From the **File** menu click **New**
- ❖ Select the **Training Samples** tab
- ❖ Select the **Steady-State Reactor with Axial Dispersion** sample
- ❖ Click **OK**

Implementation in Athena Visual Studio

The following step by step process describes the model implementation in Athena Visual Studio

- ❖ Open **Athena Visual Studio**.
- ❖ From the **File** menu, choose **New**.
- ❖ The *Welcome: New Model Selection Panel* window appears.

Welcome: New Model Selection Panel

Tabs: **Process Modeling** | **Parameter Estimation** | **Nonlinear Optimization** | **Training Samples**

Select Model

- ☐ General Form with Initial Value Problems
- ☐ Modeling with Pure Algebraic Equations
- ☐ Modeling with Ordinary Differential Equations
- ☐ Modeling with DAEs with Diagonal E Matrix
- ☐ Modeling with DAEs with Non-Diagonal E Matrix
- ☐ General Form with Initial-Boundary Value Problems
- ☑ Modeling with Boundary Value Problems
- ☐ Modeling with Partial Differential Equations
- ☐ Modeling with PDEs with Diagonal E Matrix
- ☐ Modeling with Two-Dimensional PDEs
- ☐ Modeling with User Defined Statements
- ☐ Modeling with User Defined Statements
- ☐ Modeling with User Defined Statements

Sample for the Existing Template

$$\frac{1}{x^2}\frac{d}{dx}\left(x^2\frac{du_1}{dx}\right) = \phi^2 u_1$$

$$x = 0 \quad \frac{du_1}{dx} = 0$$

$$x = 1 \quad u_1 = 1$$

$$u_2 = \frac{\sinh(\phi x)}{x\sinh(\phi)}$$

$$\phi = 2.0$$

Create your Model From:
- ⦿ A Blank Document   ○ An Existing Template

OK   Cancel   Help   **Please choose a Blank Document or an Existing Template**

- ❖ Select the *Process Modeling* tab
- ❖ Select the *Modeling with Boundary Value Problems* option.
- ❖ Choose *A Blank Document* and click **OK**.

Type your source code in the new window. The source code contains standard modeling sections (see description in the next sections below); it may also contain calls to the available math and engineering procedures as well as user-defined procedures.

Writing the Source Code

You must enter a minimum of three sections in order to create the boundary value model. The first section labeled **@Initial Conditions** is used to insert initial values for the state variables vector. The second section labeled **@Model Equations** is used to enter the model equations. The third section labeled **@Boundary Conditions** is used to enter the boundary conditions. A data section not labeled by Athena Visual Studio is used to enter all the data pertinent to the model. The data section also contains the declaration statements for all model variables, parameters and constants. This section, if used, must be the first one. The declaration of the model variables, parameters and constants must be done in accordance the Athena Visual Studio syntax rules detailed below:

Data Section

In the data section (which may not be needed and is not labeled by Athena Visual Studio) the user simply enters the problem data and various parameters and constants as shown below. In this example the user enters values for the Peclet and Damköhler numbers, the heat transfer coefficients and other parameters and constants. The Athena interpreter treats any line that begins with the exclamation mark ! as a comment. It is mandatory and strongly recommended that the users declare all the problem variables, parameters and constants. All variables in Athena are either real double or single precision or integer long (4 bytes). Character and Logical variables are also allowed. The following source code may be entered for the Data Section of this sample:

```
! Declarations and Model Constants
!==============================
 Global vi,Pehr,Uhtc,Dar,DeltaT,gamma,Pemr As Real
 Global Tro,Cbo,Rate As Real

 gamma=20.0      ! Dimensionless activation energy
 Dar=1.00        ! Damkohler number
 Pemr=196.0      ! Peclet number for mass dispersion
 Pehr=42.0       ! Peclet number for heat dispersion
 Uhtc=160.0      ! Dimensionless heat transfer coefficient
 DeltaT=0.34     ! Dimensionless adiabatic temperature rise
 vi=0.50         ! Dimensionless fluid velocity

 Cbo=0.50        ! Dimensionless inlet concentration of reactant
 Tro=0.95        ! Dimensionless inlet reactor temperature
```

Declaration of Variables

Global Variables: To declare global variables in the Athena Visual Studio environment you must use the **Global** keyword as the examples below illustrate:

> **Global**   *x, y, z, krate*   **As Real**
> **Global**   *Skount, Ncc*   **As Integer**
> **Global**   *myName* **As Character**
> **Global**   *myDecision* **As Logical**

In the above statements the variables *x, y, z, krate* will be treated as double precision and will be accessible by all modeling sections. Similarly the variables *Skount,Ncc* will be treated as integer and be accessible by all modeling sections. Character variables are assigned as Character*132 from the Athena Visual Studio parser. Single precision variables cannot be declared **Global**. Vectors and matrices can be declared in a similar manner. The array size, type and number of dimensions are declared with the **Global** statement. The elements of the array can be referenced by an integer index number, which runs from one (or zero) to the maximum number declared in the **Global** statement:

> **Global**   *y(10), c(0:5), a(4,50), b(2,4,6)*   **As Real**
> **Global**   *istate(5)*   **As Integer**

Local Variables: To declare local variables in the Athena Visual Studio environment you must use the **Dim** keyword as the examples below illustrate:

> **Dim**   *Temp, Pres*   **As Real**
> **Dim**   *TotalFlow*   **As Single**
> **Dim**   *i*   **As Integer**

In the above statements the variables *Temp, Pres* will be treated as double precision, where as the variable *TotalFlow* will be treated as single precision; these variables will be accessible only at the section where they have been declared. Similarly the variable *i* will be treated as integer and will be accessible only by the corresponding modeling section where it has been declared. Vectors and matrices can be declared in a similar manner. The array size, type and number of dimensions are declared with the **Dim** statement. The elements of the array are referenced by an integer index, which runs from one(or zero) to the number declared in the **Dim** statement:

> **Dim**   *c(10), p(4,50)*   **As Real**
> **Dim**   *streamEnthalpy(10)*   **As Single**
> **Dim**   *irow(5)*   **As Integer**

Parameter Statement: Use the **Parameter** keyword to define named constants as the examples below illustrate:

> **Parameter** *y=2.0, z=4.0* **As Real**
> **Parameter** *Skount=1, Ncc=4*   **As Integer**

In the above statements the variables *y, z* will be treated as double precision and their numerical values will be accessible by all parts of the modeling code. Similarly the variables *Skount,Ncc* will be treated as integer and their numerical values will be accessible through out all the modeling sections. The **Parameter** keyword is only allowed in the data section of the Athena Visual Studio modeling code. If it is used in the other modeling sections it will be ignored. You may view the generated Fortran code to see how the parser interprets the **Parameter** keyword.

Important Note: Always remember to declare all of your variables. Athena treats **Real** variables as double precision, **Integer** variables as 4-byte integers, **Character** variables as Character*132 and **Logical** variables as **.True.** or **.False.** Single precision variables are only allowed if are declared as local with the **Dim** keyword.

Fortran 95 Declaration Statements: You can insert Fortran 95 declaration statements by prefixing them with the double dollar sign. Below please see a list of Fortran 95 declaration statements that you can insert in your Athena code. Consult your Fortran 95 manual for the syntax rules of variable and constant declarations:

> **$$Integer, Parameter::** dp=Kind(1.0D0)
> **$$Integer, Parameter::** sp=Kind(1.0)
> **$$Real(Kind=dp)::** v1,v2
> **$$Real(Kind=sp), Dimension(3)::** a1,a2
> **$$Integer::** I1, I2
> **$$Character(Len=3)::** s2,s3
> **$$Character(Len=10), Dimension(2)::** s1
> **$$Logical::** Done
> **$$Real(Kind=dp), Dimension(:), Allocatable::** w

We are now going to describe in detail the various steps involved in writing the algebraic model for this example in the Athena Visual Studio environment. The modeling code is NOT case sensitive.

Initial Conditions

In the Initial Conditions section the user must enter the initial values for the unknown state vector. The initial values are required by the algorithm in PDAPLUS to start the iteration. The user must do the selection of the unknown state variables. The user must also make sure that he/she has a well-defined system where the number of equations is equal to the number of unknowns. The unknown state vector is represented by the variable **U( )** in Athena. For our example we choose **U(1)** to represent the dimensionless concentration of the reactant B, **U(2)** to represent the dimensionless reactor temperature. To enter the heading for the Initial Conditions section for our example:

- ❖ From the **Model** menu choose **Initial Conditions** (or **Hit F11**)
- ❖ Enter the source code as shown below for our example.

```
@Initial Conditions
 U(1)=Cbo
 U(2)=Tro
```

Model Equations

In the Model Equations section the user must enter the functions that describe the physical process. For example these functions may simply indicate the rate of change of the concentration of miscellaneous chemical components. The vector **F( )** is reserved in the Athena environment to represent the values of these functions. For our example **F(1)** is used to represent the material balance equation for the reactant B, while **F(2)** is used to represent the reactor energy balance. In this section the user may make use of temporary variables to calculate intermediate variables such as, for example, the reaction rates. This facilitates the model writing process and it is also a sign of good programming skills. To enter the Model Equations section for our example

- ❖ From the **Model** menu choose **Model Equations** (or **Hit F11**)
- ❖ Enter the source code as shown below for our example.

```
@Model Equations
 Rate=exp(gamma*(1.0-1.0/U(2)))*U(1)
 F(1)=-vi*Ux(1)+1.0/Pemr*Uxx(1)-Dar*Rate
 F(2)=-vi*Ux(2)+1.0/Pehr*Uxx(2)+DeltaT*Dar*Rate
```

Boundary Conditions

In the Boundary Conditions section the user must enter the functions that describe the physical process at the boundaries of the space domain. For example these functions may simply represent flux conditions, state vector values or a mixture of both. The vector **F( )** is reserved in the Athena environment to represent the values of these functions. For example **F(1)** may be used to represent the boundary conditions of the first model equation, **F(2)** of the second equation and so on. The vector **Ux( )** is used here to represent the first order spatial derivative and the symbol **X** is used to indicate the value of the space variable on the boundary. The variable **LEFT** is reserved in Athena to indicate the left boundary location, and the variable **RIGHT** is reserved to indicate the right boundary location. To enter the heading for the Boundary Conditions Section:

- ❖ From the **Model** menu choose **Boundary Conditions** (or **Hit F11**)
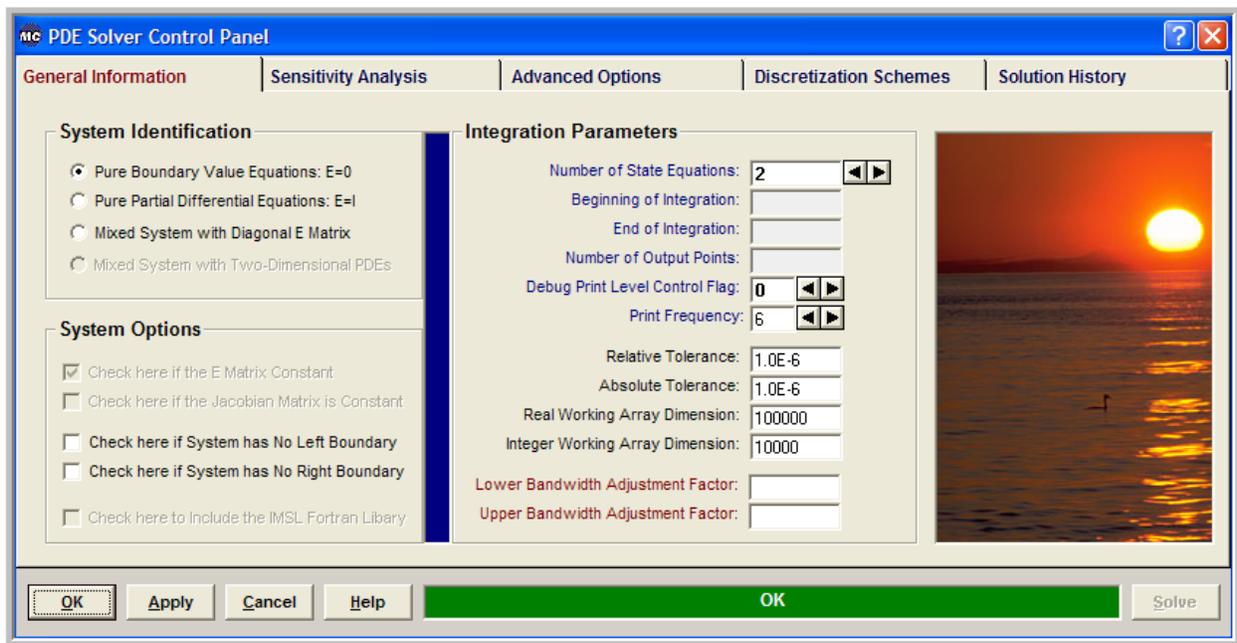- ❖ Enter the source code as shown below for our example.

```
@Boundary Conditions
 If(LEFT)Then
  F(1)=U(1)-Cbo
  F(2)=U(2)-Tro
 Else
  F(1)=Ux(1)
  F(2)=Ux(2)
 EndIf
```
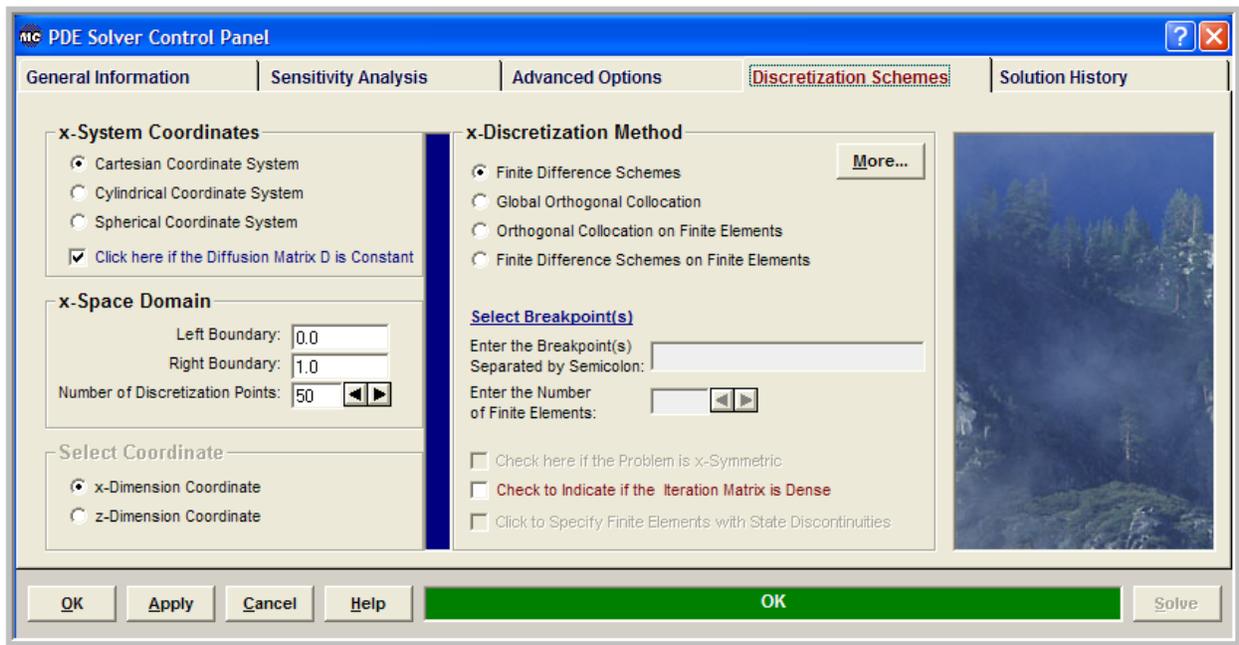
The PDAPLUS Solver

It is now time to access the Athena Visual Studio solver for Boundary Value Problems in order to enter information about the system of equations we wish to solve and various other parameters that control the integration algorithm, To do that:

- ❖ From the **Model** menu choose **Load Solver (or Hit F12)**
- ❖ The **PDE Solver Control Panel** window appears
- ❖ Enter the solver parameters as shown below for our example

In the **System Identification** group you will see that the option *Pure Boundary Value Equations E=0* has already been selected for you. From the **Integration Parameters** group enter the *Number of State Equations* and optionally change the *Debug Print Level Control Flag*, the *Print Frequency* (determines how many discretization points are printed) and the *Relative and Absolute State Tolerance* fields. The Real and Integer Working Array Dimension fields are indicative of the size of the problem. If the default values are not large enough the solver will return with the message indicating the space requirements for your problem. You may also need to examine if your model does not have a left or right boundary, in which case you will have to check the appropriate options in the **System Options** group. After you make all your selections click **OK.**

Next click on the **Discretization Schemes** tab. The following display appears

In the **x-System Coordinates** group select the Cartesian Coordinate System (or whatever is proper for your model) and check the Check here if the Diffusion Matrix D is Constant to indicate that for our example, the Peclet numbers for heat and mass transfer do not change along the reactor length. From the **x-Discretization Method** group select the *Finite Difference Schemes* method and optionally click on the command button [More...] to select *Central Differences*, *Upwind* or *Downwind Differences* (that might be appropriate for hyperbolic partial differential equations; also you may change the *Non-Uniform Grid Attenuation* factor that controls the uniformity of the discretization points distribution along the spatial direction). Finally, from the **x-Space Domain** group enter the *Left Boundary* value, the *Right Boundary* value and the *Number of Discretization Points*. After you make all your selections click **OK.**

Saving and Running

You are now ready to save your model and run it. New files are labeled UNTITLED until they are saved. Keep in mind that the maximum number of characters in a line is 132; the maximum number of lines in a file is infinity. In order to save your project:

❖ From the *File* menu, choose *Save*. The *Save As* dialog box appears. This action will save your model as a text file, and also create the Fortran code.
❖ In the Directories box, double-click a directory where you want to store the source file of your project.
❖ Type a filename in the File Name box, then choose **OK**. The default extension is **avw**
❖ To view the Fortran code that you have just created from the *View* menu choose *Fortran Code*.

You may now choose to compile, build and execute your project; to do that:

❖ From the *Build* menu choose *Compile* (or **Hit F2**)
❖ From the *Build* menu choose *Build EXE* (or **Hit F4**)
❖ From the *Build* menu choose *Execute* (or **Hit F5**)

Numerical Results

If everything goes well the results window will appear. In this window you can see the solution of your problem as well as various statistics pertaining to the solution process

```
Number of State Equations.......................    2
Number of Sensitivity Parameters................    0
Number of Discretization Points.................   52
Number of User Specified Iterations.............   30

EXIT PDAPLUS: SOLUTION FOUND

  SPACE          U(1)           U(2)
 0.00000E+00    5.00000E-01    9.50000E-01
 1.17647E-01    4.50946E-01    9.68922E-01
 2.35294E-01    3.79998E-01    9.97035E-01
 3.52941E-01    2.68014E-01    1.03889E+00
 4.70588E-01    1.23885E-01    1.08336E+00
 5.88235E-01    3.45855E-02    1.10650E+00
 7.05882E-01    7.56673E-03    1.11301E+00
 8.23529E-01    1.55139E-03    1.11443E+00
 9.41176E-01    3.13692E-04    1.11472E+00
 1.00000E+00    1.84655E-04    1.11473E+00

Number of Newton Iterations.....................    8
Number of Function Evaluations..................   69
Number of Jacobian Evaluations..................    8
Number of Jacobian Factorizations...............    8
```
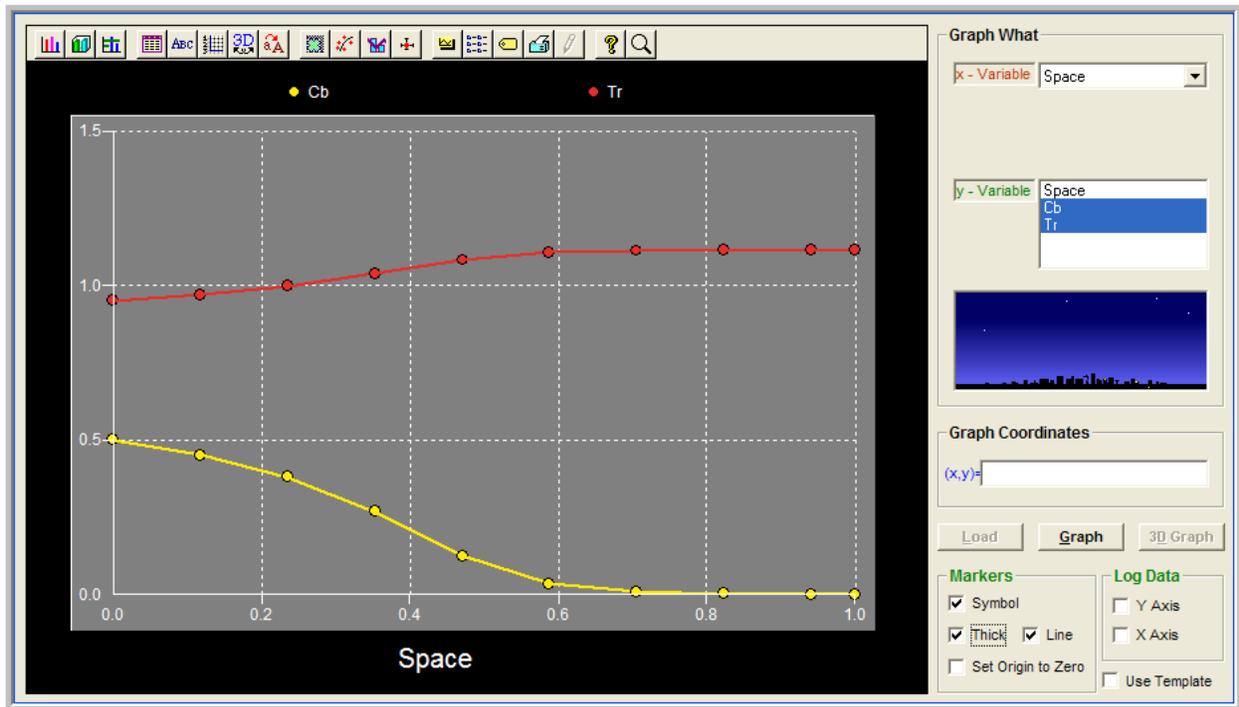
Graphical Results

If you wish to see the space profiles for the reactant concentration and reactor temperature from the **View** menu choose **Solution Graphs**, or click ![icon] The Athena Visual Studio graphics control panel appears:
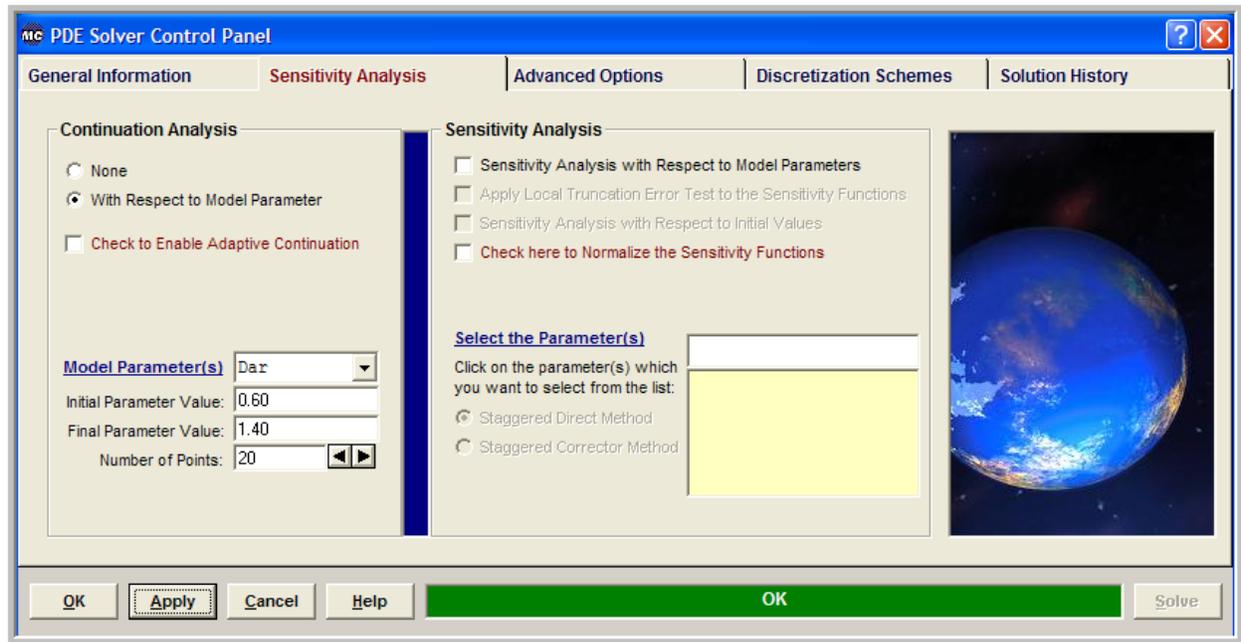

In this window fist we click **Load** to load the numerical results. Then in the **Graph What** group we select the **x-variable** (here **Space**) and the **y-variable** (here the two state variables by dragging the mouse) and click **Graph**. You should see the graph that appears above. You may now click on the graph toolbar and modify the type, title, symbol, the style and miscellaneous other properties of the graph. You may also select to graph, any one or more state variables by holding the **Ctrl** key down and clicking with your mouse on the variable or variables you wish to plot.
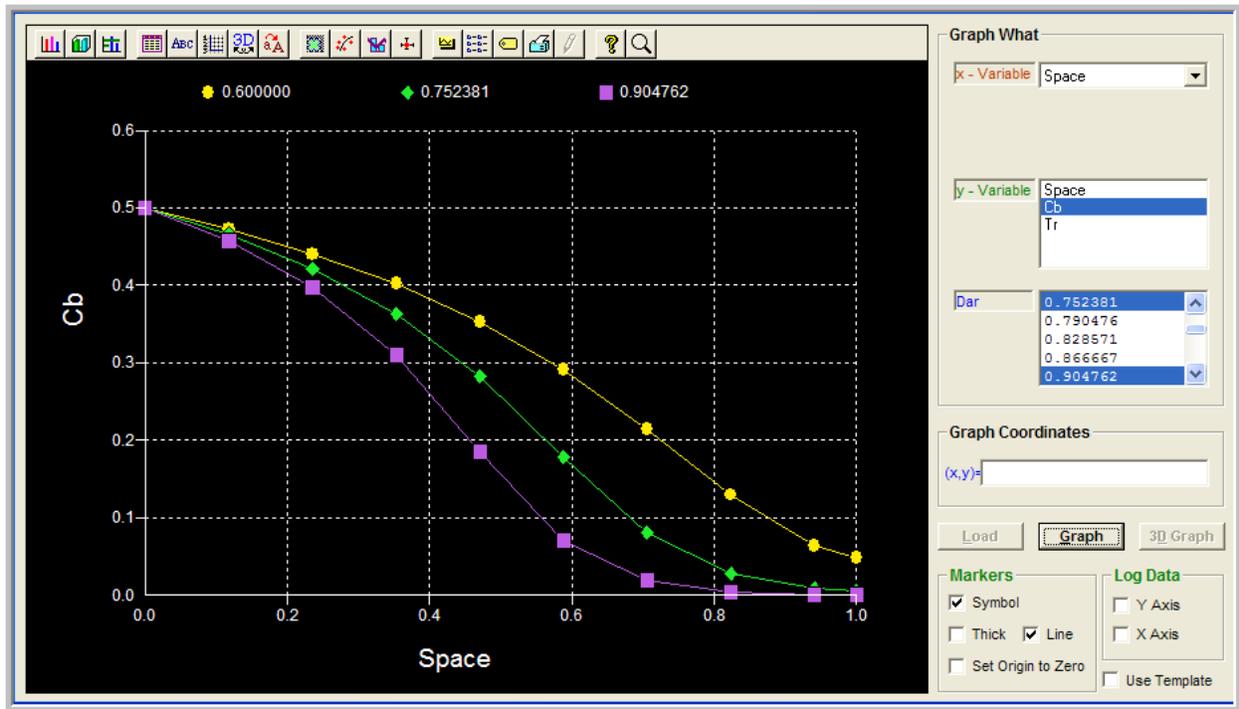
State Variable Names

You might have noticed from the Graphics Control Panel shown above, that the names of the state variables are, Cb and Tr. In order to enter these names as well as modify them, you must load the solver (**Hit F12**). After you do that, select the Solution History tab:

In the **Display Options** group enter the Number of Variables using the spin control and then, in the adjacent spreadsheet change the names of the state variables. Also from the **Show Solution** group click on the State Grid Location spin control to display the solution at various grid locations. If the spin control is disabled click **Solve** to enable it. Notice that in this tab, you have miscellaneous other Solution Options, such as Saving the Final Solution, or Restarting from the Saved Solution. These options allow you to perform dynamic studies of steady-state systems by starting from the steady-state solution. You may also perform reactor shut-down scenarios as well as investigate the effect of control variables to the reactor performance.

Continuation Analysis

Athena Visual Studio allows for convenient and efficient parametric studies. Suppose for instance, we wish to determine the effect of the Damköhler number of the reactant concentration and reactor temperatures. In order to do that, first we must load the solver (**Hit F12**), and click the Sensitivity and Continuation tab.

In the **Continuation Analysis** group click With Respect to Model Parameter and subsequently select the parameters **Dar** from the drop down list; enter the *Initial Parameter Value*, the *Final Parameter Value* and the *Number of Points* for the continuation analysis

Now choose **OK** or click **Apply**. From the **Build** menu select **Execute** (or **Hit F5**). From the **View** menu we select **Solution Graphs** and the following panel is displayed:

In this window fist we click **Load** to load the numerical results. Then in the **Graph What** group we select the **x-variable** (here **Space**), the **y-variable** (here concentration, **Cb**) and also a number of values of the Damköhler number; then click **Graph**. You should see the graph that appears above. You may now click on the graph toolbar and modify the type, title, symbol, the style and miscellaneous other properties of the graph.

Implicit Auxiliary Variables

Athena Visual Studio allows for convenient and efficient calculation of derived quantities by the introduction of auxiliary variables. Assume for example that we wish to estimate the conversion of the reactant B in the plug flow reactor. We introduce an auxiliary variable $U(3)=X_B$ that represents the conversion and the associated equation that is obviously the definition of the conversion, i.e.

$$F(3) = U(3) - \frac{C_{B0} - C_B(z=1)}{C_{B0}} \times 100$$

We can implement this in Athena in a very straightforward manner. First we load the PDAPLUS solver (**Hit F12**) and increase the Number of State Equations to 3. Then we write the source code that corresponds to the introduction of the new variable and its associated equation. The new source code might look like the code displayed below:

```
@Initial Conditions
 U(1)=Cbo
 U(2)=Tro
 U(3)=0.0

@Model Equations
 Rate=exp(gamma*(1.0-1.0/U(2)))*U(1)
 F(1)=-vi*Ux(1)+1.0/Pemr*Uxx(1)-Dar*Rate
 F(2)=-vi*Ux(2)+1.0/Pehr*Uxx(2)+DeltaT*Dar*Rate
 F(3)=U(3)-(Cbo-U(1))/Cbo*100.0

@Boundary Conditions
 If(LEFT)Then
  F(1)=U(1)-Cbo
  F(2)=U(2)-Tro
 Else
  F(1)=Ux(1)
  F(2)=Ux(2)
 EndIf
 F(3)=U(3)-(Cbo-U(1))/Cbo*100.0
```
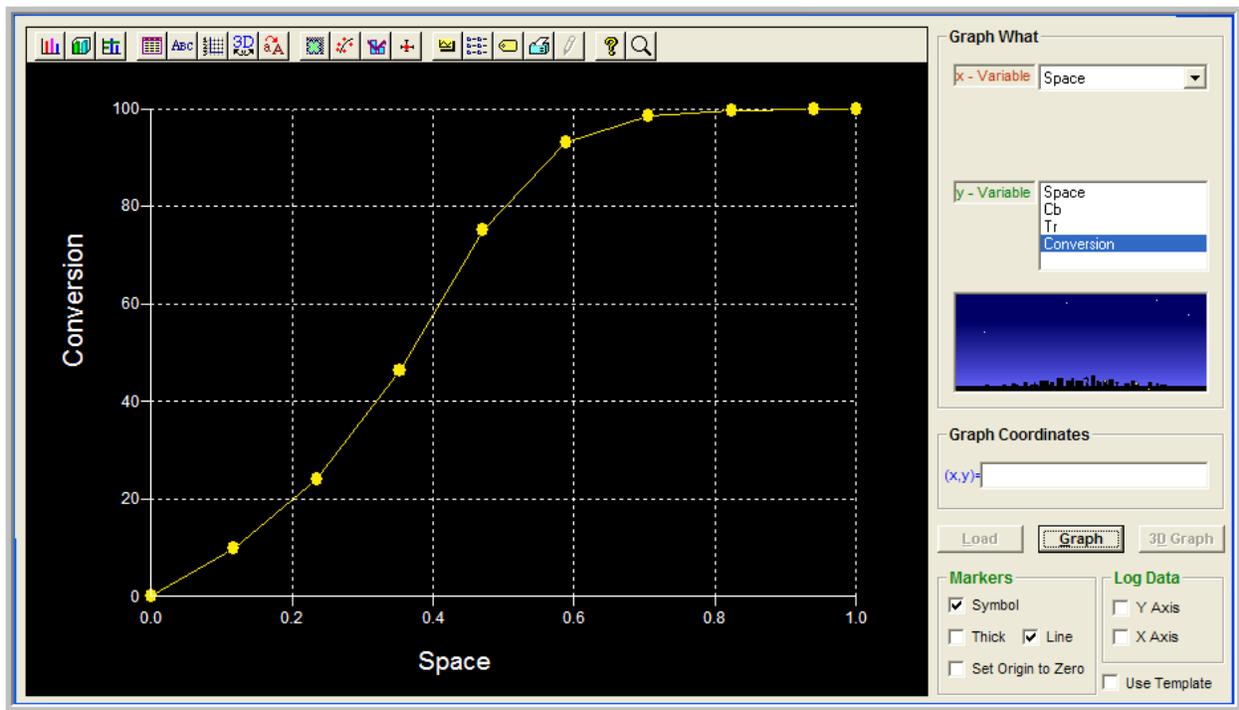
From the **Build** menu select **Execute** (or **Hit F5**). From the **View** menu we select **Solution Graphs** and the following panel is displayed:

In this window fist we click **Load** to load the numerical results. Then in the **Graph What** group we select the **x-variable** (here **Space**), and the **y-variable** (here conversion, **Conversion**); then click **Graph**. You should see the graph that appears above. You may now click on the graph toolbar and modify the type, title, symbol, the style and miscellaneous other properties of the graph.