AthenaVisual Plus

Process Modeling and Nonlinear Parameter Estimation.
Model Discrimination and Optimal Experimental Design.

User Programming Guide

Michael Caracotsios, Ph.D.

AthenaVisual, Inc.

Table of Contents

1.0 Introduction

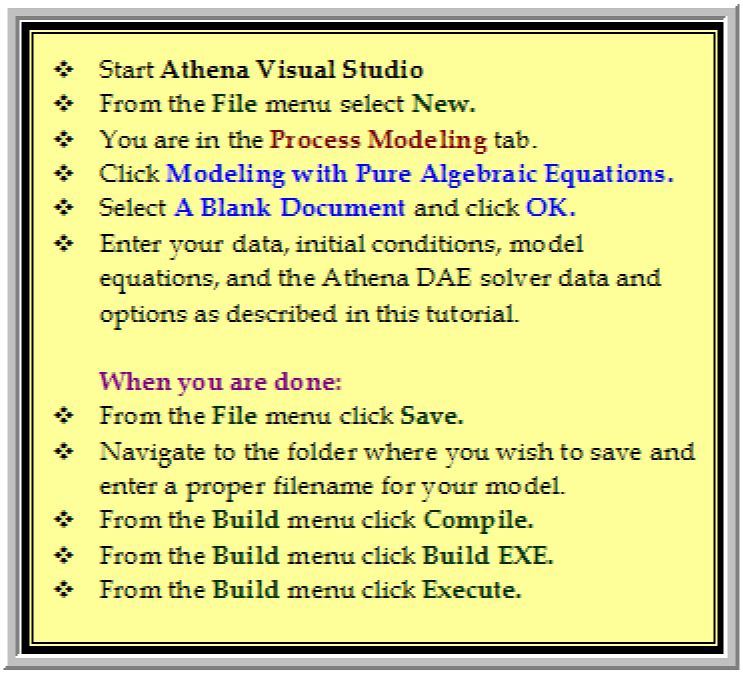
As an engineer or scientist in a manufacturing facility, researcher in a pilot plant or laboratory, graduate or undergraduate student, you frequently come face to face with some of the most challenging tasks of science and engineering: the mathematical modeling of an existing or conceptualized process, and the estimation of the unknown chemical and physical parameters that you have decided to include in the model or models you are about to investigate.

You decide to begin your project. Suddenly your world becomes a complicated and unfriendly place. Unless you happen to be an expert on all phases of your task, deriving the models, solving the mathematical equations efficiently and successfully, designing an optimal set of experiments, gathering supporting evidence, estimating the unknown parameters in your models, and discriminating amongst rival models, you will be faced with the daunting task of selecting the appropriate tools to produce, gather, and analyze the information needed to carry out your modeling project. So you start sorting through software. You find one to solve your models, another to estimate your model parameters, and yet another to design a good set of experiments. In addition, you begin to look for physical property data, thermodynamic methods to describe the state of the mixtures being dealt with, and a whole host of other necessities. If you are like most scientists and engineers, you will also begin looking at and discussing the works of your peers around you. Soon you realize that there is a myriad of choices regarding how to write models of chemical systems, what software to use to solve the models and estimate parameters, and what language to write your models in. How easy is it? How much money you have to spend on the appropriate tools? What training will be required to use said tools? It can be overwhelming.

If all these sound too familiar then I invite you to experience **AthenaVisual Plus**, a unique software that offers an integrated environment for the modeling, estimation, optimal experimental design, model discriminating and graphical interpretation of chemically reactive and non-reactive systems. Athena gives the user the freedom to develop his or her own models, but alleviates the tasks associated with solving the underlying equations and estimating the model adjustable parameters. It offers an advanced graphical interface to a set of powerful solvers for the robust and efficient handling of lumped and distributed parameter systems both dynamic and steady-state. In addition, it offers an easy, direct and seamless link of all these systems with powerful parameter estimation and optimization software, which allows for the analysis of single, and multi-response experiments, model discrimination and optimal experimental design. A large number of graphical capabilities allow the users to interpret and disseminate the acquired information in a useful and effective manner.

Michael Caracotsios, Ph.D.
President
AthenaVisual, Inc.

2.0 Analysis of Nonlinear Algebraic Models



- Start **Athena Visual Studio**
- From the **File** menu select **New**.
- You are in the **Process Modeling** tab.
- Click **Modeling with Pure Algebraic Equations**.
- Select **A Blank Document** and click **OK**.
- Enter your data, initial conditions, model equations, and the Athena DAE 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**.

Nonlinear Algebraic Models that can be solved with AthenaVisual Plus take the form $\mathbf{F}(\mathbf{u};\boldsymbol{\theta}) = \mathbf{0}$ where u is a vector of unknowns, such as the temperature, pressure and composition, and $\boldsymbol{\theta}$ is a vector of parameters pertinent to the process we are modeling. Algebraic models are ordinarily used to describe steady-state processes such as CSTRs (Continuous Stirred Tank Reactors); they are also used to describe steady-state phase equilibrium, such as single-stage and multi-stage flash calculations, chemical equilibrium and the discretized forms of distributed parameter systems such as steady-state plug flow and fixed bed reactors. The Algebraic Equations can be solved using the AthenaVisual Plus powerful damped Newton algorithm which is encoded in the DDAPLUS solver. The example problem below has been created to illustrate the functionality of AthenaVisual Plus in dealing with the solution of nonlinear algebraic equations. Additional features such as sensitivity analysis, parametric continuation and the use of auxiliary variables will also be demonstrated.

2.1 Continuous Stirred Tank Reactor (CSTR) Modeling

The following chemical reactions take place in a Continuous Stirred Tank Reactor (CSTR) under isothermal and isobaric conditions:

$$A \xrightleftharpoons{r_1} B \xrightarrow{r_2} C$$
$$A \xrightarrow{r_3} D$$

The mathematical model that describes the reactor material balance is given by the following system of nonlinear algebraic equations. The last column in Table 1. shows the unknown variables (component molar concentrations) in AthenaVisual Plus notation:

Table 1. CSTR Material Balance Equations

$C_{A0} + \tau(-r_1 - r_3) - C_A = 0$	$r_1 = \dfrac{k_1 K_A}{1 + K_A C_A + K_B C_B}\left(C_A - \dfrac{C_B}{K_{eql}}\right)$	$U(1) := C_A$
$C_{B0} + \tau(r_1 - r_2) - C_B = 0$	$r_2 = \dfrac{k_2 K_B}{1 + K_A C_A + K_B C_B} C_B$	$U(2) := C_B$
$C_{C0} + \tau r_2 - C_C = 0$	$r_3 = \dfrac{k_3 K_A}{1 + K_A C_A + K_B C_B} C_A$	$U(3) := C_C$
$C_{D0} + \tau r_3 - C_D = 0$		$U(4) := C_D$

We wish to perform the following tasks:

Estimate the concentration of the reaction components in the effluent stream
Estimate the sensitivity of the concentrations to changes in rate constants k_1, k_2
Estimate the conversion of component *A* by introducing an auxiliary variable, X_A
Create a plot of Residence Time vs. Conversion using Continuation Analysis

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

MODEL PARAMETERS	**Description and Units**
$\tau = 90.0$	Residence Time (s)
$C_{A0} = 0.35$	Feed Molar Concentration (mol/m3)
$k_1 = 0.01$	Reaction Rate Constant (mol/m3 s)
$k_2 = 0.01$	Reaction Rate Constant (mol/m3 s)
$k_3 = 0.001$	Reaction Rate Constant (mol/m3 s)
$K_A = \exp\left(\dfrac{35{,}000}{R_g T} - \dfrac{91}{R_g}\right)$	Adsorption Constant (m3/mol)
$K_B = \exp\left(\dfrac{20{,}000}{R_g T} - \dfrac{53}{R_g}\right)$	Adsorption Constant (m3/mol)
$K_{eql} = 4.29 \times 10^{-4} \dfrac{K_A}{K_B} \exp\left(\dfrac{30{,}000}{R_g T}\right)$	Equilibrium Constant(Unitless)
$R_g = 8.314$	Universal Gas Constant (J/mol K)
$T = 330.0$	Reactor Temperature (K)

The conversion of the component *A* is defined via the auxiliary variable:

$$X_A = \frac{C_{A0} - C_A}{C_{A0}} \times 100$$

The sensitivity functions of the components with respect to the reaction rate constants $\{k_1, k_2\}$ are given by the first order differentials as indicated below (notice the AthenaVisual Plus nomenclature):

$$\left[\begin{array}{c|c:c} \mathrm{U(1,1)} & \mathrm{U(1,2)} & \mathrm{U(1,3)} \\ \hdashline \mathrm{U(2,1)} & \mathrm{U(2,2)} & \mathrm{U(2,3)} \\ \hdashline \mathrm{U(3,1)} & \mathrm{U(3,2)} & \mathrm{U(3,3)} \\ \hdashline \mathrm{U(4,1)} & \mathrm{U(4,2)} & \mathrm{U(4,3)} \end{array}\right] = \left[\begin{array}{c|c:c} C_A & \frac{\partial C_A}{\partial k_1} & \frac{\partial C_A}{\partial k_2} \\ \hdashline C_B & \frac{\partial C_B}{\partial k_1} & \frac{\partial C_B}{\partial k_2} \\ \hdashline C_C & \frac{\partial C_c}{\partial k_1} & \frac{\partial C_C}{\partial k_2} \\ \hdashline C_D & \frac{\partial C_D}{\partial k_1} & \frac{\partial C_C}{\partial k_2} \end{array}\right]$$

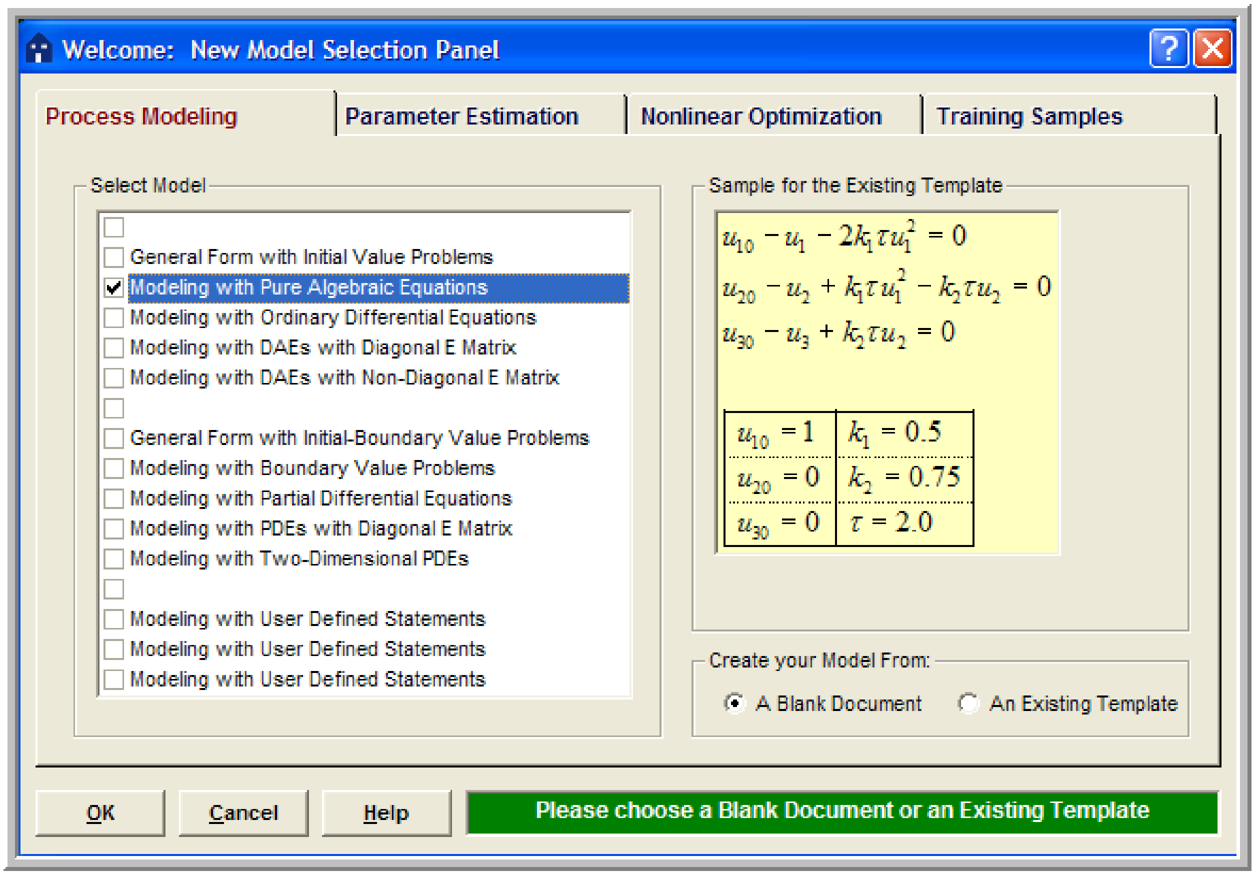
This example tutorial is already precoded in AthenaVisual Plus. If you do not wish to type the code on your own you may access it by doing the following:

- Open **AthenaVisual Plus**
- From the **File** menu click **New**
- Select the **Training Samples** tab
- Select the **Steady-State Isothermal CSTR** sample
- Click **OK**

2.2 Implementation in AthenaVisual Plus

The following step by step process describes the model implementation in AthenaVisual Plus

- Open **AthenaVisual Plus**. From the ***File*** menu, choose ***New***. The *Welcome: New Model Selection Panel* window appears.



Welcome: New Model Selection Panel

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

$$u_{10} - u_1 - 2k_1 \tau u_1^2 = 0$$

$$u_{20} - u_2 + k_1 \tau u_1^2 - k_2 \tau u_2 = 0$$

$$u_{30} - u_3 + k_2 \tau u_2 = 0$$

$u_{10} = 1$	$k_1 = 0.5$
$u_{20} = 0$	$k_2 = 0.75$
$u_{30} = 0$	$\tau = 2.0$

Create your Model From:

● A Blank Document ○ An Existing Template

Please choose a Blank Document or an Existing Template

- You are in the *Process Modeling* tab
- Select the *Modeling with Pure Algebraic Equations* option.
- Choose *A Blank Document* and click **OK**.

Type your source code in the new window. The source code contains data and modeling sections (see detailed description below); it may also contain calls to the available AthenaVisual Plus math and engineering procedures as well as user-defined subroutines and/or functions.

2.2.1 Writing the Source Code for Algebraic Models

The user must enter a minimum of two sections in order to create the algebraic model. The first section labeled **@Initial Conditions** is used to insert initial guesses for the unknown vector. These initial guesses are used by the Newton method to start the iterative algorithm. The second section labeled **@Model Equations** is used to enter the model equations. A data section not labeled by AthenaVisual Plus 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 AthenaVisual Plus syntax rules detailed below:

2.2.1.1 Data Section

In the data section the user simply enters the problem data and various constants. In this example the user enters values for residence time, reactor temperature, initial feed concentration and miscellaneous other reaction parameters. 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. Character and Logical variables are also allowed. The following source code may be entered for this example (If you use the Windows Copy and Paste commands to enter this code into your AthenaVisual Plus project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```
! Declarations and Model Constants
!=================================
 Global k1,k2,k3,Keql,Ka,Kb As Real
 Global Temp,Rg,Tau As Real
 Global CAo,CBo,CCo,CDo As Real
 Global CA,CB,CC,CD As Real

 Rg=8.314    ! Universal Gas Constant (J/mole K)
 Tau=90.0    ! Residence Time (s)
 Temp=330.0  ! Temperature (K)

 CAo=0.35    ! Initial Concentration of A (mol/m3)
 CBo=0.0     ! Initial Concentration of B (mol/m3)
 CCo=0.0     ! Initial Concentration of C (mol/m3)
 CDo=0.0     ! Initial Concentration of D (mol/m3)

 k1=0.01     ! Reaction rate coefficient (mol/m3 s)
 k2=0.01     ! Reaction rate coefficient (mol/m3 s)
 k3=0.001    ! Reaction rate coefficient (mol/m3 s)

 Ka=exp(35000.0/Rg/Temp-91.0/Rg)          ! Adsorption constant (m3/mol)
 Kb=exp(20000.0/Rg/Temp-53.0/Rg)          ! Adsorption constant (m3/mol)
 Keql=4.29E-4*Ka/Kb*exp(30000.0/Rg/Temp)  ! Equilibrium constant
```

2.2.1.2 Declaration of Variables in AthenaVisual Plus

Global Variables: To declare global variables in the AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus environment. The modeling code is NOT case sensitive.

2.2.1.3 Initial Conditions Section

In the Initial Conditions section the user must enter the initial guess for the unknown state vector. The initial guess is required by the Newton algorithm in **DDAPLUS** to start the iterations. The user must do the selection of the unknown state variables and 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 molar concentration of component *A*, **U(2)** to represent the molar concentration of component *B* and so forth. 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)=CAo  ! [Ca]
 U(2)=CBo  ! [Cb]
 U(3)=CCo  ! [Cc]
 U(4)=CDo  ! [Cd]
```

2.2.1.4 Model Equations Section

In the Model Equations section the user must enter the functions that describe the physical process. For example these functions may be simply the component balances. The value of these functions will eventually be driven to zero by the Newton algorithm. At the initial guess these functions will have values, which are indicative of how good our guess is. The closer to zero we are, the better our initial guess is. The vector **F( )** is reserved in the Athena environment to represent the values of these functions (often know as residuals). For example **F(1)** may be used to represent the component *A* material balance, **F(2)** may be used to represent the component *B* material balance and so forth. 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

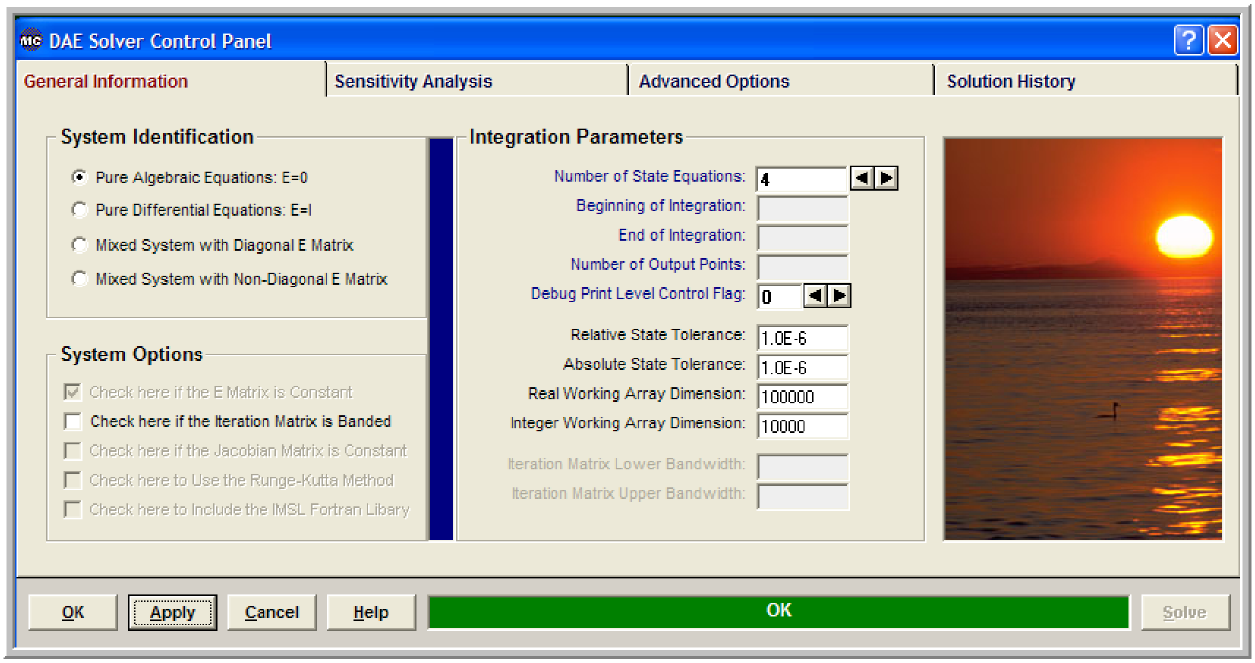
- From the ***Model*** menu choose ***Model Equations*** (or **Hit F11)**
- Enter the source code as shown below for our example.

```
@Model Equations
 Dim R1, R2, R3, R4, Denom As Real

 CA = U(1)
 CB = U(2)
 CC = U(3)
 CD = U(4)

 Denom = 1.0 + Ka * CA + Kb * CB
 R1 = k1 * Ka * (CA - CB / Keql) / Denom
 R2 = k2 * Kb * CB / Denom
 R3 = k3 * Ka * CA / Denom

 F(1) = CAo - CA + (-R1 - R3) * Tau
 F(2) = CBo - CB + (R1 - R2) * Tau
 F(3) = CCo - CC + R2 * Tau
 F(4) = CDo - CD + R3 * Tau
```

2.2.2 Entering the Information about the DDAPLUS Solver

It is now time to access the AthenaVisual Plus solver for Differential/Algebraic Equations in order to enter information about the system of equations we wish to solve and various other parameters that control the Newton algorithm. To do that:

- From the ***Model*** menu choose ***Load Solver*** (or **Hit F12**)
- Enter the solver parameters as shown below for our example



System Identification

● Pure Algebraic Equations: E=0
○ Pure Differential Equations: E=I
○ Mixed System with Diagonal E Matrix
○ Mixed System with Non-Diagonal E Matrix

System Options

☑ Check here if the E Matrix is Constant
□ Check here if the Iteration Matrix is Banded
□ Check here if the Jacobian Matrix is Constant
□ Check here to Use the Runge-Kutta Method
□ Check here to Include the IMSL Fortran Libary

Integration Parameters

Number of State Equations:	4
Beginning of Integration:	
End of Integration:	
Number of Output Points:	
Debug Print Level Control Flag:	0
Relative State Tolerance:	1.0E-6
Absolute State Tolerance:	1.0E-6
Real Working Array Dimension:	100000
Integer Working Array Dimension:	10000
Iteration Matrix Lower Bandwidth:	
Iteration Matrix Upper Bandwidth:	

The ***DAE Solver Control Panel*** window appears. In the **System Identification** group you will see that the option *Pure Algebraic Equations E=0* has already been selected for you. From the **Integration Parameters** group enter the *Number of State Equations* and optionally change *Debug Print Level Control Flag*, 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 has a banded structure in which case you will have to check the appropriate options in the **System Options** group. Then choose **OK** or click **Apply.**

2.2.3 Saving and Running your Model

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 saves your model and creates the Fortran code.
- In the Directories box, double-click a directory where you want to store the source file.
- Type a filename (a filename cannot contain the following characters: \ / : * ? “ < > |) 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**)

2.2.4 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:

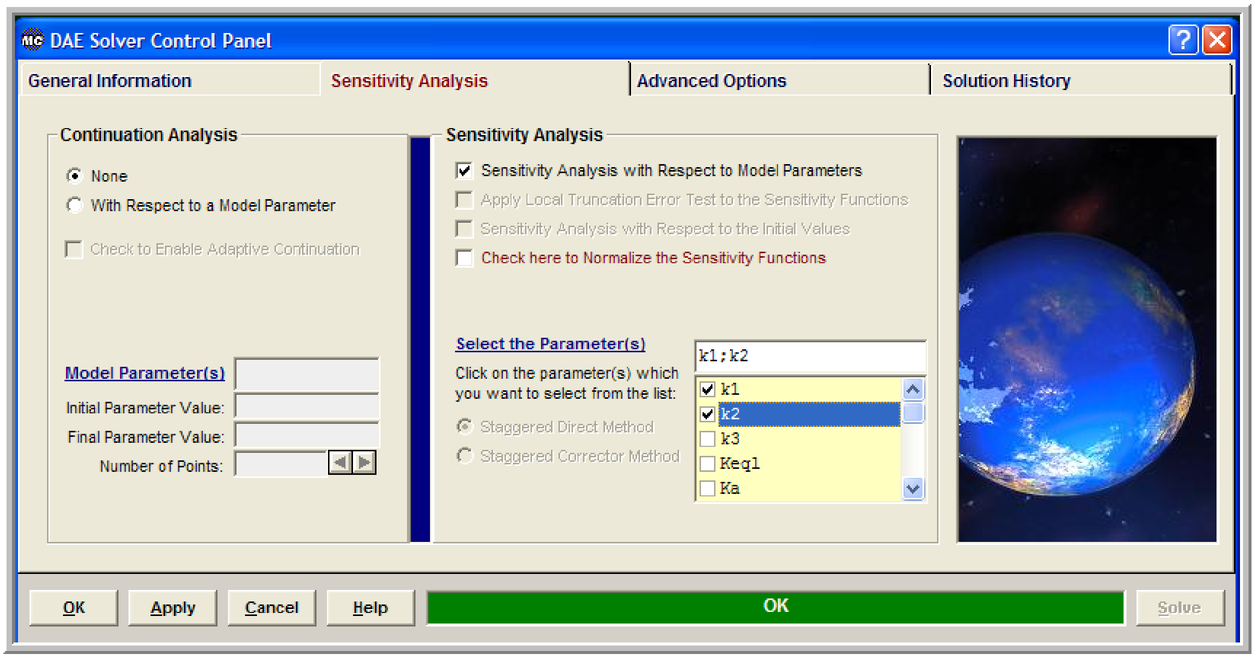
```
Number of Algebraic Equations.................     4
Number of Sensitivity Parameters..............     0
Number of User Specified Iterations...........    30

EXIT DDAPLUS: SOLUTION FOUND

VARIABLE  INITIAL GUESS      RESIDUALS    FINAL VALUE
 U(1)       3.50000E-01    3.02720E-15    8.00051E-02
 U(2)       0.00000E+00   -5.91150E-15    1.07519E-01
 U(3)       0.00000E+00    3.10212E-15    1.37412E-01
 U(4)       0.00000E+00   -2.59450E-16    2.50643E-02

Number of Newton Iterations.....................     5
Number of Function Evaluations..................    26
Number of Jacobian Evaluations..................     5
Number of Jacobian Factorizations...............     5
```

2.2.5 Sensitivity Analysis Calculations

AthenaVisual Plus allows for convenient and efficient calculation of the first order sensitivity functions given by:

$$\mathbf{W}(\boldsymbol{\theta}) = \frac{\partial \mathbf{u}}{\partial \boldsymbol{\theta}}$$

Suppose that we wish to investigate the sensitivity of the reactor effluent composition to small perturbations in the reaction rate constants k_1 and k_2. In order to do that, first we must load the solver (**Hit F12**), and click the *Sensitivity Analysis* tab.



Continuation Analysis

● None
○ With Respect to a Model Parameter
□ Check to Enable Adaptive Continuation

Model Parameter(s)
Initial Parameter Value:
Final Parameter Value:
Number of Points:

Sensitivity Analysis

☑ Sensitivity Analysis with Respect to Model Parameters
□ Apply Local Truncation Error Test to the Sensitivity Functions
□ Sensitivity Analysis with Respect to the Initial Values
□ Check here to Normalize the Sensitivity Functions

Select the Parameter(s)

Click on the parameter(s) which you want to select from the list:

● Staggered Direct Method
○ Staggered Corrector Method

k1;k2

☑ k1
☑ k2
□ k3
□ Keq1
□ Ka

In the **Sensitivity Analysis** group click *Sensitivity Analysis with respect to Model Parameters* and subsequently select the parameters k_1, k_2 from the drop down list; optionally you may wish to click on the *Check here to Normalize the Sensitivity Functions* in which case the following first order coefficients will be calculated:

$$\mathbf{W}(\boldsymbol{\theta}) = \frac{\partial \mathbf{u}}{\partial \ln \boldsymbol{\theta}}$$

Now choose **OK** or click **Apply**. From the ***Build*** menu select ***Execute*** (or **Hit F5**). You should see the following results:

```
Number of Algebraic Equations..................     4
Number of Sensitivity Parameters...............     2
Number of User Specified Iterations............    30

EXIT DDAPLUS: SOLUTION FOUND

VARIABLE  INITIAL GUESS      RESIDUALS    FINAL VALUE        SENSITIVITY MATRIX
 U(1)       3.50000E-01    3.02720E-15   8.00051E-02  ||  -6.68166E+00   -7.99267E-01
 U(2)       0.00000E+00   -5.91150E-15   1.07519E-01  ||   2.44028E+00   -6.41993E+00
 U(3)       0.00000E+00    3.10212E-15   1.37412E-01  ||   5.83854E+00    7.17137E+00
 U(4)       0.00000E+00   -2.59450E-16   2.50643E-02  ||  -1.59715E+00    4.78335E-02

Number of Newton Iterations.....................    5
Number of Function Evaluations..................   28
Number of Jacobian Evaluations..................    5
Number of Jacobian Factorizations...............    5
```

An interpretation of these results is given in the following table using the nomenclature in our example:

$C_A = 0.080005$	$\frac{\partial C_A}{\partial k_1} = -6.68166$	$\frac{\partial C_A}{\partial k_2} = -0.799267$
$C_B = 0.107519$	$\frac{\partial C_B}{\partial k_1} = 2.44028$	$\frac{\partial C_B}{\partial k_2} = -6.41993$
$C_C = 0.137412$	$\frac{\partial C_C}{\partial k_1} = 5.83854$	$\frac{\partial C_C}{\partial k_2} = 7.17137$
$C_D = 0.025064$	$\frac{\partial C_D}{\partial k_1} = -1.59715$	$\frac{\partial C_D}{\partial k_2} = 0.047833$

2.2.6 Auxiliary Variables

AthenaVisual Plus 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 component *A* in the continuous stirred tank reactor. We introduce an auxiliary variable $U(5) = X_A$ that represents the conversion and the associated equation that is obviously the definition of the conversion, i.e.

$$F(5) = U(5) - \frac{C_{A0} - C_A}{C_{A0}} \times 100$$

We can implement this in Athena in a very straightforward manner. First we load the DDAPLUS solver (**Hit F12**) and increase the *Number of State Equations* to 5. 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:

```
! Declarations and Model Constants
!================================
 Global k1,k2,k3,Keql,Ka,Kb As Real
 Global Temp,Rg,Tau As Real
 Global CAo,CBo,CCo,CDo As Real
 Global CA,CB,CC,CD As Real

 Rg=8.314    ! Universal Gas Constant (J/mole K)
 Tau=90.0    ! Residence Time (s)
 Temp=330.0  ! Temperature (K)

 CAo=0.35    ! Initial Concentration of A (mol/m3)
 CBo=0.0     ! Initial Concentration of B (mol/m3)
 CCo=0.0     ! Initial Concentration of C (mol/m3)
 CDo=0.0     ! Initial Concentration of D (mol/m3)

 k1=0.01     ! Reaction rate coefficient (mol/m3 s)
 k2=0.01     ! Reaction rate coefficient (mol/m3 s)
 k3=0.001    ! Reaction rate coefficient (mol/m3 s)

 Ka=exp(35000.0/Rg/Temp-91.0/Rg)          ! Adsorption constant (m3/mol)
 Kb=exp(20000.0/Rg/Temp-53.0/Rg)          ! Adsorption constant (m3/mol)
 Keql=4.29E-4*Ka/Kb*exp(30000.0/Rg/Temp)  ! Equilibrium constant

@Initial Conditions
 U(1)=CAo  ! [Ca]
 U(2)=CBo  ! [Cb]
 U(3)=CCo  ! [Cc]
 U(4)=CDo  ! [Cd]
 U(5)=0.0  ! Conversion

@Model Equations
 Dim R1, R2, R3, R4, Denom As Real

 CA = U(1)
 CB = U(2)
 CC = U(3)
 CD = U(4)

 Denom = 1.0 + Ka * CA + Kb * CB
 R1 = k1 * Ka * (CA - CB / Keql) / Denom
 R2 = k2 * Kb * CB / Denom
 R3 = k3 * Ka * CA / Denom

 F(1) = CAo - CA + (-R1 - R3) * Tau
 F(2) = CBo - CB + (R1 - R2) * Tau
 F(3) = CCo - CC + R2 * Tau
 F(4) = CDo - CD + R3 * Tau
 F(5) = U(5) – (CAo – CA)/CAo * 100.0
```

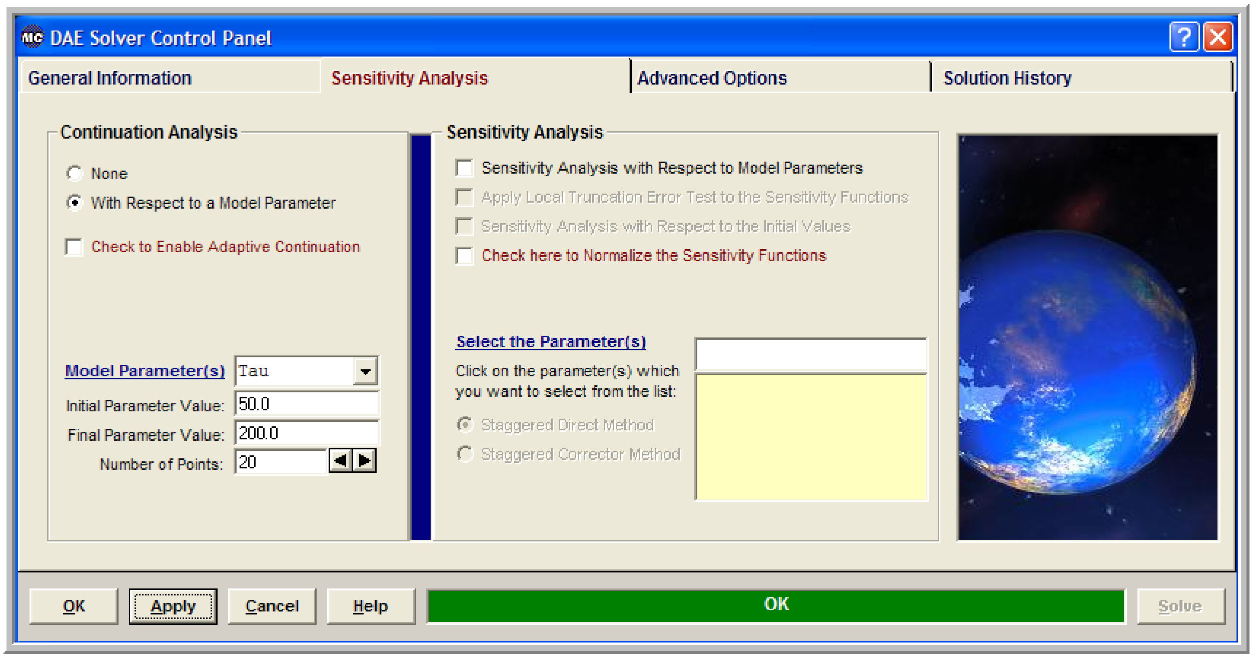
Now from the ***Build*** menu select ***Execute*** (or **Hit F5**). You should see the following:

```
Number of Algebraic Equations..................     5
Number of Sensitivity Parameters...............     0
Number of User Specified Iterations............    30

EXIT DDAPLUS: SOLUTION FOUND

VARIABLE  INITIAL GUESS      RESIDUALS     FINAL VALUE
 U(1)       3.50000E-01    3.02720E-15    8.00051E-02
 U(2)       0.00000E+00   -5.91150E-15    1.07519E-01
 U(3)       0.00000E+00    3.10212E-15    1.37412E-01
 U(4)       0.00000E+00   -2.59450E-16    2.50643E-02
 U(5)       0.00000E+00   -8.27047E-14    7.71414E+01

Number of Newton Iterations.....................     5
Number of Function Evaluations..................    31
Number of Jacobian Evaluations..................     5
Number of Jacobian Factorizations...............     5
```

From these results we observe that the conversion is $X_A = 77.14\%$

2.2.7 Homotopy or Continuation Analysis

Now we can use Athena to conduct a continuation study. Our goal is to investigate the variation of conversion with respect to the reactor residence time. To do that we load the solver (**Hit F12**),


DAE Solver Control Panel

Continuation Analysis

○ None
● With Respect to a Model Parameter
□ Check to Enable Adaptive Continuation

Model Parameter(s): Tau
Initial Parameter Value: 50.0
Final Parameter Value: 200.0
Number of Points: 20

Sensitivity Analysis

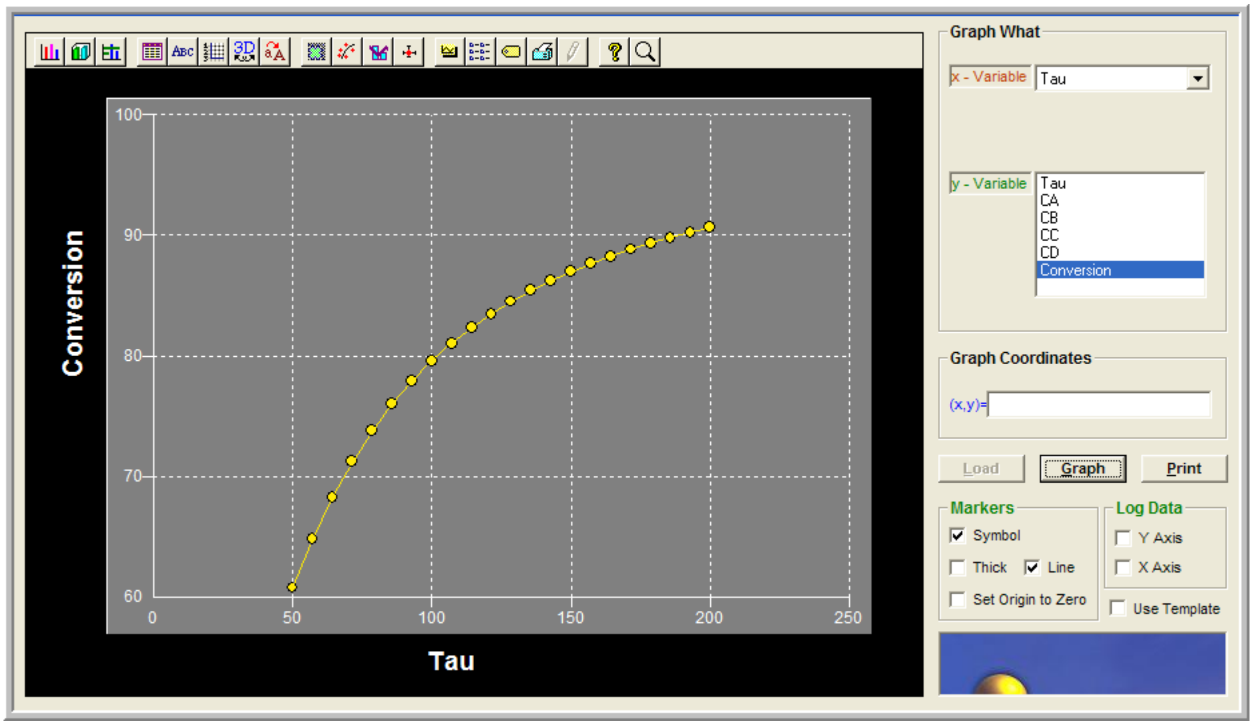
□ Sensitivity Analysis with Respect to Model Parameters
□ Apply Local Truncation Error Test to the Sensitivity Functions
□ Sensitivity Analysis with Respect to the Initial Values
□ Check here to Normalize the Sensitivity Functions

Select the Parameter(s)
Click on the parameter(s) which you want to select from the list:
● Staggered Direct Method
○ Staggered Corrector Method

OK

and click on the *Sensitivity Analysis* tab. In the **Continuation Analysis** group we click the option with caption *With respect to a Model Parameter* and select the parameter **Tau** from the drop

down list. We then input the *Initial Parameter Value* and the *Final Parameter Value* as well as the *Number of Points* and click **OK.**
Now from the ***Build*** menu select ***Execute***(or **Hit F5**). Instead of looking at the file with the large number of numerical results from the ***View*** menu we select ***Solution Graphs*** and the following panel is displayed:



Graph What

x - Variable: Tau

y - Variable: Tau, CA, CB, CC, CD, Conversion

Graph Coordinates

(x,y)=

Load | Graph | Print

Markers: ☑ Symbol; □ Thick; ☑ Line; □ Set Origin to Zero

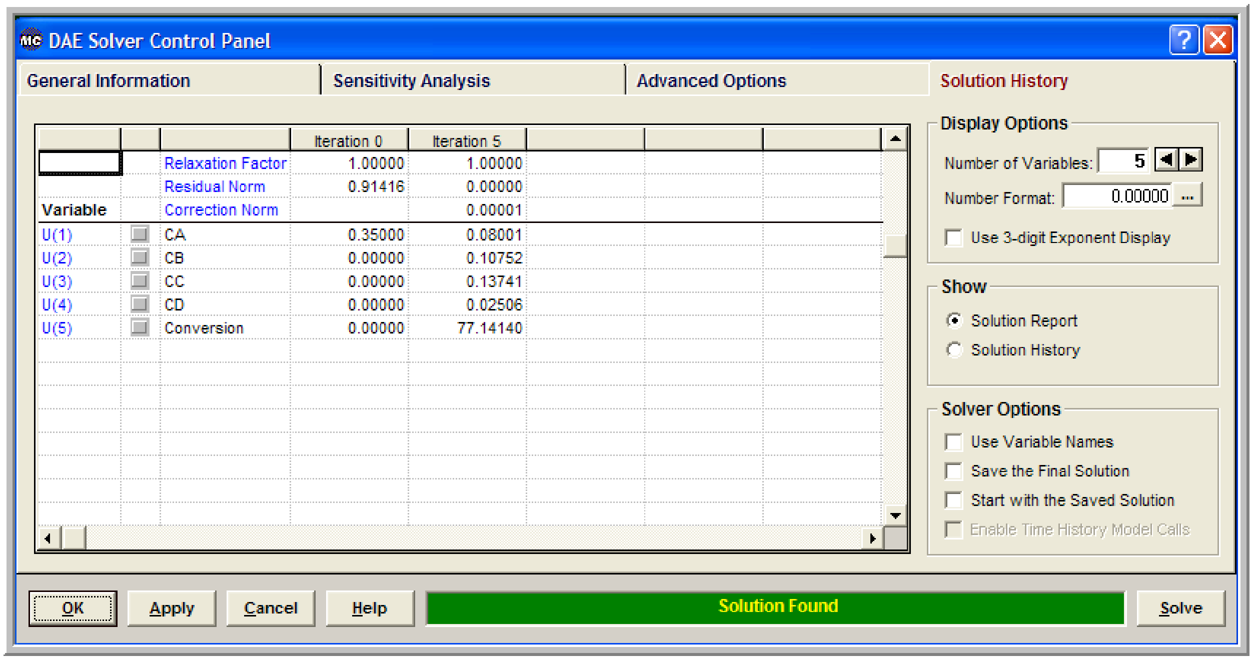
Log Data: □ Y Axis; □ X Axis

□ Use Template

In this window fist we click ***Load*** to load the numerical results. Then in the **Graph What** group we select the **x-variable** (here **Tau**) and the **y-variable** (here **Conversion**) 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

2.2.8 State Variable Names

You might have noticed from the Graphics Control Panel shown above, the names of the independent variables being, CA, CB, CC, CD and Conversion. In order to enter these names as well as modify them, you must load the solver (**Hit F12**). Now, first disable the continuation analysis option (you can always enable it again) by selecting the *Sensitivity Analysis* tab, and then in the **Continuation Analysis** group by clicking the option with caption *None*. After you do that, select the *Solution History* tab:



			Iteration 0	Iteration 5
		Relaxation Factor	1.00000	1.00000
		Residual Norm	0.91416	0.00000
Variable		Correction Norm		0.00001
U(1)	□	CA	0.35000	0.08001
U(2)	□	CB	0.00000	0.10752
U(3)	□	CC	0.00000	0.13741
U(4)	□	CD	0.00000	0.02506
U(5)	□	Conversion	0.00000	77.14140

Display Options

Number of Variables: 5

Number Format: 0.00000

□ Use 3-digit Exponent Display

Show

● Solution Report

○ Solution History

Solver Options

□ Use Variable Names

□ Save the Final Solution

□ Start with the Saved Solution

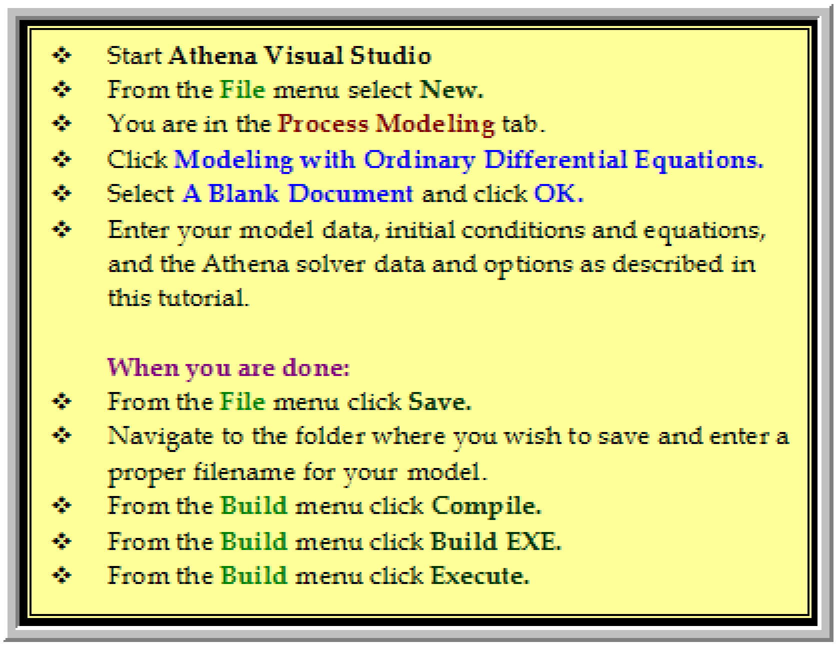
□ Enable Time History Model Calls

Solution Found

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** group select *Solution Report*. Then click *Apply* and then *Solve*. You should see the results as shown in the image above. Notice also, other types of solution information appearing in the spreadsheet. If you wish to enter analytical derivatives with respect to one of the independent variables you must click the corresponding cell in the CheckBox column. This can be helpful especially for the auxiliary variables, such as the conversion in our example, since the derivatives of all functions with respect to conversion are exactly equal to zero, except the last equation, in which of course the derivative with respect to conversion is equal to one. Should you decide to do that for the auxiliary variable, you just enter the following segment of code in your project:

```
@Model Equations
 Dim R1, R2, R3, R4, Denom As Real

 CA = U(1)
 CB = U(2)
 CC = U(3)
 CD = U(4)

 Denom = 1.0 + Ka * CA + Kb * CB
 R1 = k1 * Ka * (CA - CB / Keql) / Denom
 R2 = k2 * Kb * CB / Denom
 R3 = k3 * Ka * CA / Denom

 F(1) = CAo - CA + (-R1 - R3) * Tau
 F(2) = CBo - CB + (R1 - R2) * Tau
 F(3) = CCo - CC + R2 * Tau
 F(4) = CDo - CD + R3 * Tau
 F(5) = U(5) – (CAo – CA)/CAo *100

@Jacobian Matrix
 dF(5,5) = 1.0
```

3.0 Analysis of Nonlinear Initial Value Problems



- Start **Athena Visual Studio**
- From the **File** menu select **New**.
- You are in the **Process Modeling** tab.
- Click **Modeling with Ordinary Differential Equations.**
- 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-Value Problems or Differential/Algebraic models in AthenaVisual Plus take the form:

$$\mathbf{E}(t,\mathbf{u};\boldsymbol{\theta})\frac{d\mathbf{u}}{dt}=\mathbf{F}(t,\mathbf{u};\boldsymbol{\theta})$$

where **u** is a state vector of unknown variables (usually temperature, pressure, and composition), $\boldsymbol{\theta}$ is a vector of known parameters pertinent to the process we are modeling and *t* is a time-like dimension. $\mathbf{E}(t,\mathbf{u};\boldsymbol{\theta})$ is in general a non-diagonal matrix. This matrix does not have to be of full rank; zero rows of this matrix correspond to pure algebraic equations in the original mixed DAE system. Differential and Algebraic models are ordinarily used to model unsteady-state processes such as the start-up or shut-down of Continuous Stirred Tank Reactors as well as batch or semi-batch reactors; they are also used to describe steady-state distributed parameter systems such as plug flow and fixed bed reactors. They can be solved using **DDAPLUS,** a powerful modified Newton algorithm in conjunction with a fixed leading coefficient backward difference formula for the approximation of the first order derivative.

3.1 Batch Conversion of Glucose to Gluconic Acid

The mathematical model for the fermentation of the bacterium Pseudomonas Ovalis, that produces gluconic acid, is described by the following system of differential equations. The last column shows the unknown variables in Athena notation.

Rate of cell growth	$\frac{dC_1}{dt} = k_1 C_1 \left(1 - \frac{C_1}{k_2}\right)$	$U(1) := C_1$
Rate of gluconolactone formation	$\frac{dC_2}{dt} = \frac{k_3 C_1 C_4}{k_4 + C_4} - a k_5 C_2$	$U(2) := C_2$
Rate of gluconic acid formation	$\frac{dC_3}{dt} = k_5 C_2$	$U(3) := C_3$
Rate of glucose consumption	$\frac{dC_4}{dt} = -b \frac{k_3 C_1 C_4}{k_4 + C_4}$	$U(4) := C_4$

We wish to perform the following tasks:

- Plot the concentration of the reaction components as a function of time for $0 \le t \le 10\,hr$
- Estimate the sensitivity of these concentrations to changes in reaction rate constants k_3, k_4 at the end of the reaction.
- Estimate the conversion of glucose X_{Glu} by introducing an auxiliary variable, $U(5)$
- Estimate the glucose X_{Glu} conversion assuming that the overall cell growth rate was depressed by 50% after 5 hours from the beginning of the reaction and 100% after 8 hours, i.e.,

$$k_1 = \begin{cases} 0.950 & 0 \le t \le 5 \\ 0.425 & 5 < t \le 8 \\ 0.000 & 8 < t \le 10 \end{cases}$$

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

MODEL PARAMETERS	INITIAL CONDITIONS
$a = 0.9082$	$C_1(0) = 0.5$
$b = 1.011$	$C_2(0) = 0.0$
$k_1 = 0.95$	$C_3(0) = 0.0$
$k_2 = 3.5$	$C_4(0) = 50.0$
$k_3 = 18.00$	
$k_4 = 37.50$	
$k_5 = 1.10$	

The conversion of the glucose $\left(C_4\right)$ is defined via the auxiliary variable:

$$X_{\text{Glu}} = \frac{C_4\left(t=0\right) - C_4\left(t\right)}{C_4\left(t=0\right)} \times 100$$

The sensitivity functions of the components with respect to the reaction rate constants $\left\{k_3, k_4\right\}$ are given by the first order differentials as indicated below (notice the AthenaVisual Plus nomenclature):

$$\left[\begin{array}{c|c:c}
\text{U(1,1)} & \text{U(1,2)} & \text{U(1,3)} \\ \hdashline
\text{U(2,1)} & \text{U(2,2)} & \text{U(2,3)} \\ \hdashline
\text{U(3,1)} & \text{U(3,2)} & \text{U(3,3)} \\ \hdashline
\text{U(4,1)} & \text{U(4,2)} & \text{U(4,3)}
\end{array}\right] = \left[\begin{array}{c|c:c}
C_1 & \dfrac{\partial C_1}{\partial k_3} & \dfrac{\partial C_1}{\partial k_4} \\ \hdashline
C_2 & \dfrac{\partial C_2}{\partial k_3} & \dfrac{\partial C_2}{\partial k_4} \\ \hdashline
C_3 & \dfrac{\partial C_3}{\partial k_3} & \dfrac{\partial C_3}{\partial k_4} \\ \hdashline
C_4 & \dfrac{\partial C_4}{\partial k_3} & \dfrac{\partial C_4}{\partial k_4}
\end{array}\right]$$

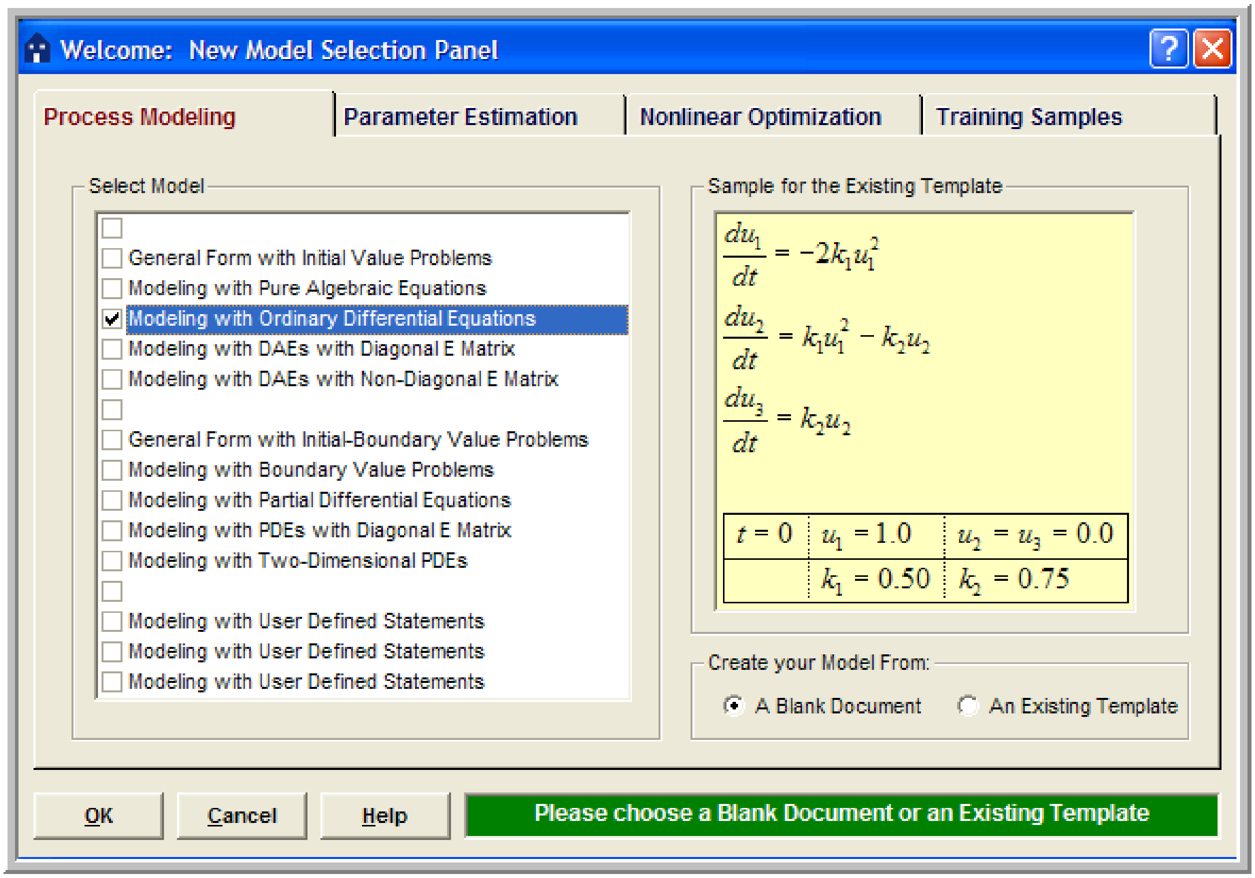
This example tutorial is already precoded in AthenaVisual Plus. If you do not wish to type the code on your own you may access it by doing the following:

- Open **AthenaVisual Plus**
- From the **File** menu click **New**
- Select the **Training Samples** tab
- Select the **Batch Isothermal Reactor** sample
- Click **OK**

3.2 Implementation in AthenaVisual Plus

The following step by step process describes the model implementation in AthenaVisual Plus

- Open **AthenaVisual Plus**. From the ***File*** menu, choose ***New***. The *Welcome: New Model Selection Panel* window appears.



□
□ 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

$$\frac{du_1}{dt} = -2k_1u_1^2$$

$$\frac{du_2}{dt} = k_1u_1^2 - k_2u_2$$

$$\frac{du_3}{dt} = k_2u_2$$

$t = 0$	$u_1 = 1.0$	$u_2 = u_3 = 0.0$
	$k_1 = 0.50$	$k_2 = 0.75$

● A Blank Document ○ An Existing Template

Please choose a Blank Document or an Existing Template

- Select the *Process Modeling* tab
- Select the *Modeling with Ordinary Differential Equations* 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.

3.2.1 Writing the Source Code for Initial-Value Problems

You must enter a minimum of two sections in order to create the differential model. The first section labeled **@Initial Conditions** is used to insert initial values for the unknown vector. The second section labeled **@Model Equations** is used to enter the model equations. A third section labeled **@Coefficient Matrix** is optional and may be used to enter the matrix $\mathbf{E}\left(t,\mathbf{u};\boldsymbol{\theta}\right)$ of DAE systems with diagonal or non-diagonal matrix. A data section not labeled by AthenaVisual Plus may also be used to enter all the data pertinent to the model. The data section may also contain declaration statements for all model variables, parameters and constants. This section, if used, must be the first one in the model. The declaration of the model variables, parameters and constants must be done in accordance the AthenaVisual Plus syntax rules shown below:

3.2.1.1 Data Section

In the data section the user simply enters the problem data and various constants as shown below. In this example the user enters values for reaction rate constants and the system kinetic parameters. 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. Character and Logical variables are also allowed. The following source code may be entered for this example (If you use the Windows Copy and Paste commands to enter this code into your AthenaVisual Plus project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```
! Declarations and Model Constants
!=================================
 Global a,b As Real
 Global k1,k2,k3,k4,k5 As Real

 a = 0.9082
 b = 1.011

 k1 = 0.95
 k2 = 3.5
 k3 = 18.0
 k4 = 37.5
 k5 = 1.1
```

3.2.1.2 Declaration of Variables in AthenaVisual Plus

Global Variables: To declare global variables in the AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 differential model for this example in the AthenaVisual Plus environment. The modeling code is NOT case sensitive.

3.2.1.3 Initial Conditions Section

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 DDAPLUS to start the integration. 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 concentration of cell, **U(2)** to represent the concentration of gluconolactone, **U(3)** to represent the concentration of gluconic acid and **U(4)** to represent the concentration of glucose. 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) = 0.56
 U(2) = 1.28
 U(3) = 0.16
 U(4) = 45.0
```

3.2.1.4 Model Equations Section

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 example **F(1)** may be used to represent the rate of change of the cell concentration, **F(2)** may be used to represent the rate of change of the gluconolactone concentration and so forth. 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

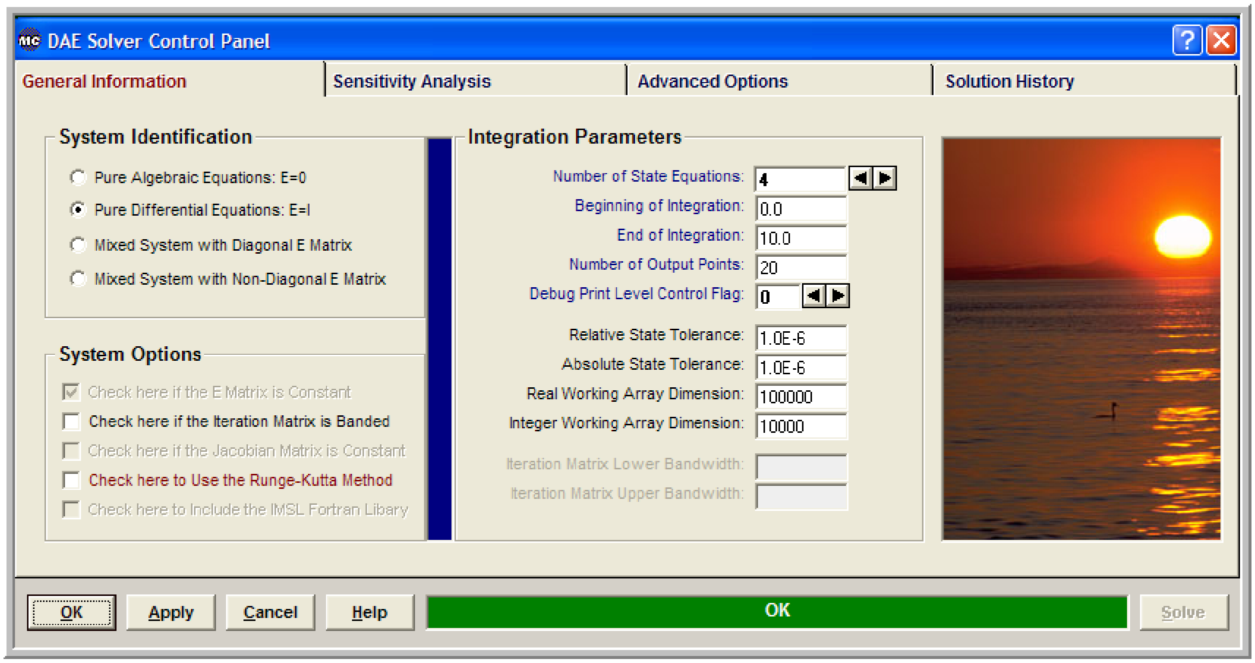
- From the ***Model*** menu choose ***Model Equations*** (or **Hit F11**)
- Enter the source code as shown below for our example.

```
@Model Equations
 F(1) = k1 * U(1) * (1.0 - U(1) / k2)
 F(2) = k3 * U(1) * U(4) / (k4 + U(4)) - a * k5 * U(2)
 F(3) = k5 * U(2)
 F(4) = -b * k3 * U(1) * U(4) / (k4 + U(4))
```

3.2.2 Entering the Information about the DDAPLUS Solver

It is now time to load the AthenaVisual Plus solver for Differential/Algebraic Equations in order to enter data about the system of equations and the integration algorithm. To do that:

- From the ***Model*** menu choose ***Load Solver*** (or **Hit F12**)
- Enter the solver parameters as shown below for our example



System Identification

○ Pure Algebraic Equations: E=0
● Pure Differential Equations: E=I
○ Mixed System with Diagonal E Matrix
○ Mixed System with Non-Diagonal E Matrix

System Options

☑ Check here if the E Matrix is Constant
□ Check here if the Iteration Matrix is Banded
□ Check here if the Jacobian Matrix is Constant
□ Check here to Use the Runge-Kutta Method
□ Check here to Include the IMSL Fortran Libary

Integration Parameters

Number of State Equations:	4
Beginning of Integration:	0.0
End of Integration:	10.0
Number of Output Points:	20
Debug Print Level Control Flag:	0
Relative State Tolerance:	1.0E-6
Absolute State Tolerance:	1.0E-6
Real Working Array Dimension:	100000
Integer Working Array Dimension:	10000
Iteration Matrix Lower Bandwidth:	
Iteration Matrix Upper Bandwidth:	

The ***DAE Solver Control Panel*** window appears. In the **System Identification** group you will see that the option *Pure Differential Equations E=I* has already been selected for you. From the **Integration Parameters** group enter the *Number of State Equations* the *Beginning* and *End of Integration*, the *Number of Output Points* (that controls the granularity of the graphs) and optionally change *Debug Print Level Control Flag*, 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 has a banded structure in which case you will have to check the appropriate options in the **System Options** group. Then choose **OK** or click **Apply.**

3.2.3 Saving and Running your Model

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 saves your model and creates 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 (a filename cannot contain the following characters: \ / : * ? “ < > |) in the File Name box, then choose **OK**. The default extension is **avw**
- To view the Fortran code that you 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**)

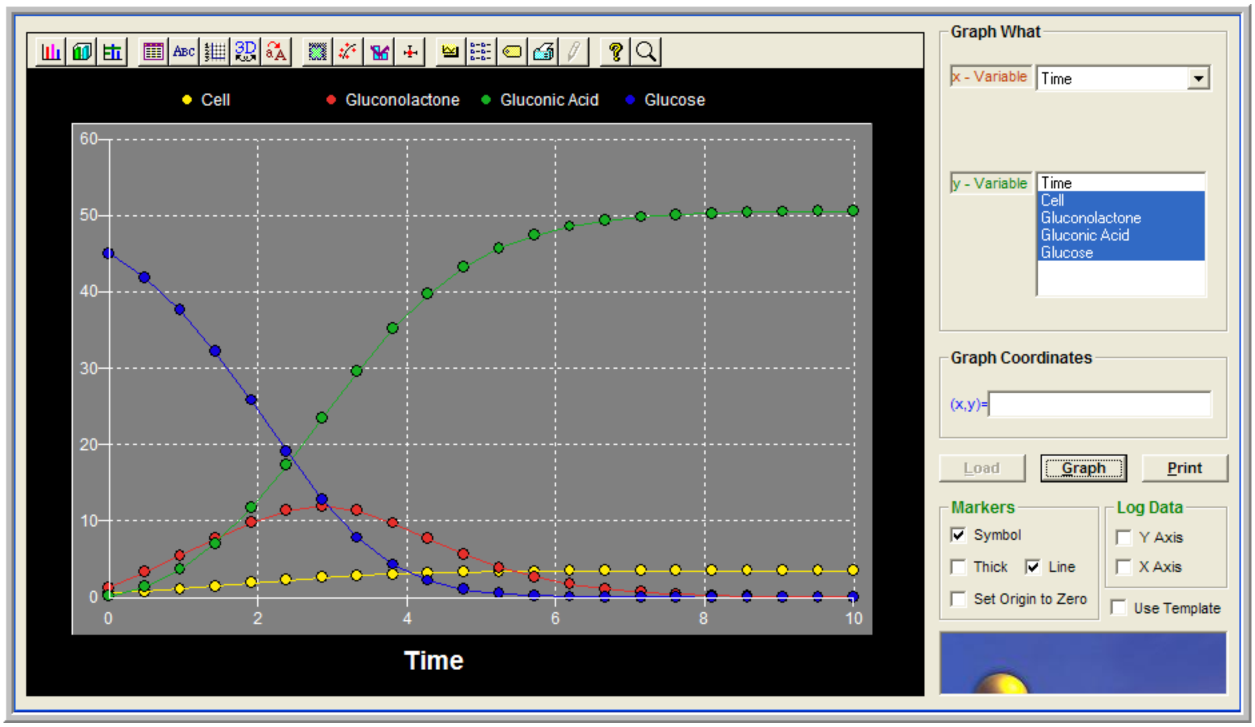
3.2.4 Numerical Results

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.......................     4
Number of Sensitivity Parameters................     0
Number of Integration Output Points.............    22

 TIME           U(1)           U(2)           U(3)           U(4)
 0.00000E+00    5.60000E-01    1.28000E+00    1.60000E-01    4.50000E+01
 4.76190E-01    8.06528E-01    3.30736E+00    1.35857E+00    4.18498E+01
 9.52381E-01    1.12023E+00    5.45140E+00    3.64626E+00    3.75817E+01
 1.42857E+00    1.48852E+00    7.69245E+00    7.08777E+00    3.21560E+01
 1.90476E+00    1.88213E+00    9.80013E+00    1.16829E+01    2.58060E+01
 2.38095E+00    2.26274E+00    1.13652E+01    1.72601E+01    1.91027E+01
 2.85714E+00    2.59678E+00    1.19523E+01    2.34172E+01    1.28558E+01
 3.33333E+00    2.86590E+00    1.13510E+01    2.95714E+01    7.81295E+00
 3.80952E+00    3.06818E+00    9.75887E+00    3.51337E+01    4.31537E+00
 4.28571E+00    3.21240E+00    7.68674E+00    3.97113E+01    2.20720E+00
 4.76190E+00    3.31141E+00    5.64883E+00    4.31938E+01    1.06986E+00
 5.23810E+00    3.37764E+00    3.94536E+00    4.56888E+01    5.01224E-01
 5.71429E+00    3.42116E+00    2.65892E+00    4.74006E+01    2.30068E-01
 6.19048E+00    3.44944E+00    1.74831E+00    4.85402E+01    1.04339E-01
 6.66667E+00    3.46767E+00    1.13022E+00    4.92832E+01    4.69835E-02
 7.14286E+00    3.47936E+00    7.22125E-01    4.97608E+01    2.10670E-02
 7.61905E+00    3.48684E+00    4.57634E-01    5.00647E+01    9.42209E-03
 8.09524E+00    3.49162E+00    2.88366E-01    5.02567E+01    4.20744E-03
 8.57143E+00    3.49466E+00    1.80978E-01    5.03775E+01    1.87750E-03
 9.04762E+00    3.49660E+00    1.13260E-01    5.04532E+01    8.37320E-04
 9.52381E+00    3.49784E+00    7.07377E-02    5.05005E+01    3.73302E-04
 1.00000E+01    3.49863E+00    4.41169E-02    5.05301E+01    1.66392E-04

EXIT DDAPLUS: SOLUTION FOUND

Number of Steps Taken Thus Far...................      128
Number of Function Evaluations...................      284
Number of Jacobian Evaluations...................        6
Number of Jacobian Factorizations................        6
```

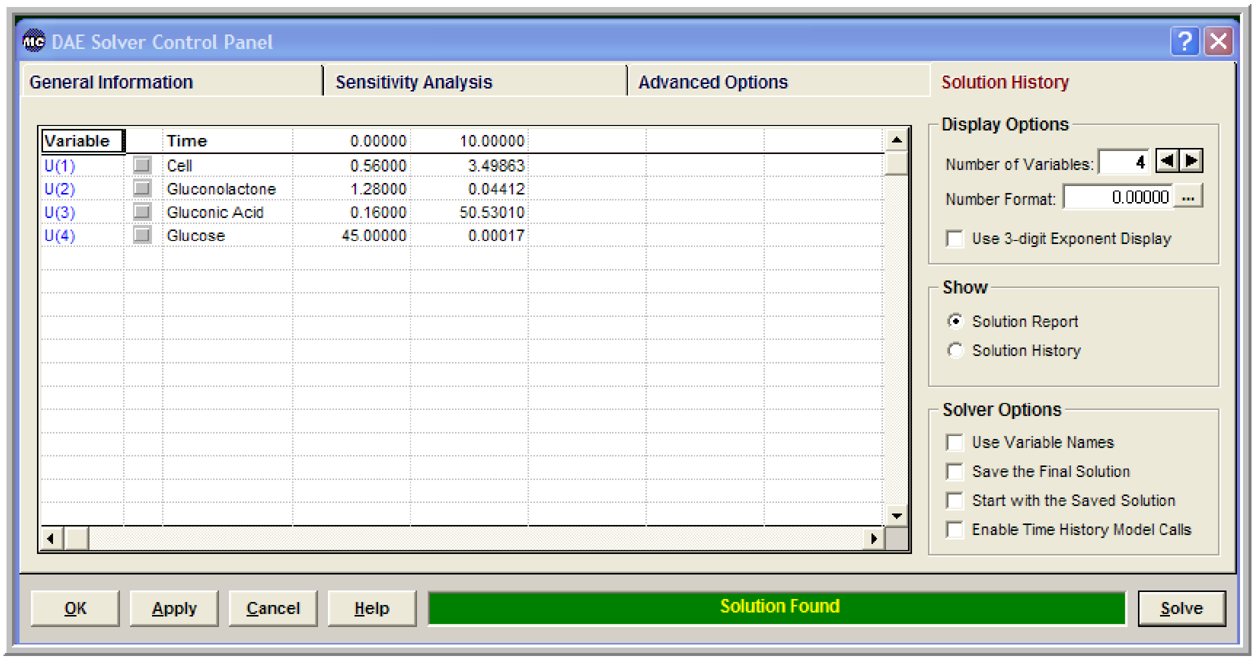
3.2.5 Solution Graphs

If you wish to see the time profiles for all the species that participate in this fermentation process from the ***View*** menu choose ***Solution Graphs***. The AthenaVisual Plus 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 **Time**) and the **y-variable** (here all the 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.

Note: You might have noticed from the Graphics Control Panel shown above, that the names of the state variables are, *Cell*, *Gluconolactone*, *Gluconic Acid* and *Glucose*. 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:



Variable		Time	0.00000	10.00000
U(1)	□	Cell	0.56000	3.49863
U(2)	□	Gluconolactone	1.28000	0.04412
U(3)	□	Gluconic Acid	0.16000	50.53010
U(4)	□	Glucose	45.00000	0.00017

Display Options
Number of Variables: 4
Number Format: 0.00000
□ Use 3-digit Exponent Display

Show
● Solution Report
○ Solution History

Solver Options
□ Use Variable Names
□ Save the Final Solution
□ Start with the Saved Solution
□ Enable Time History Model Calls

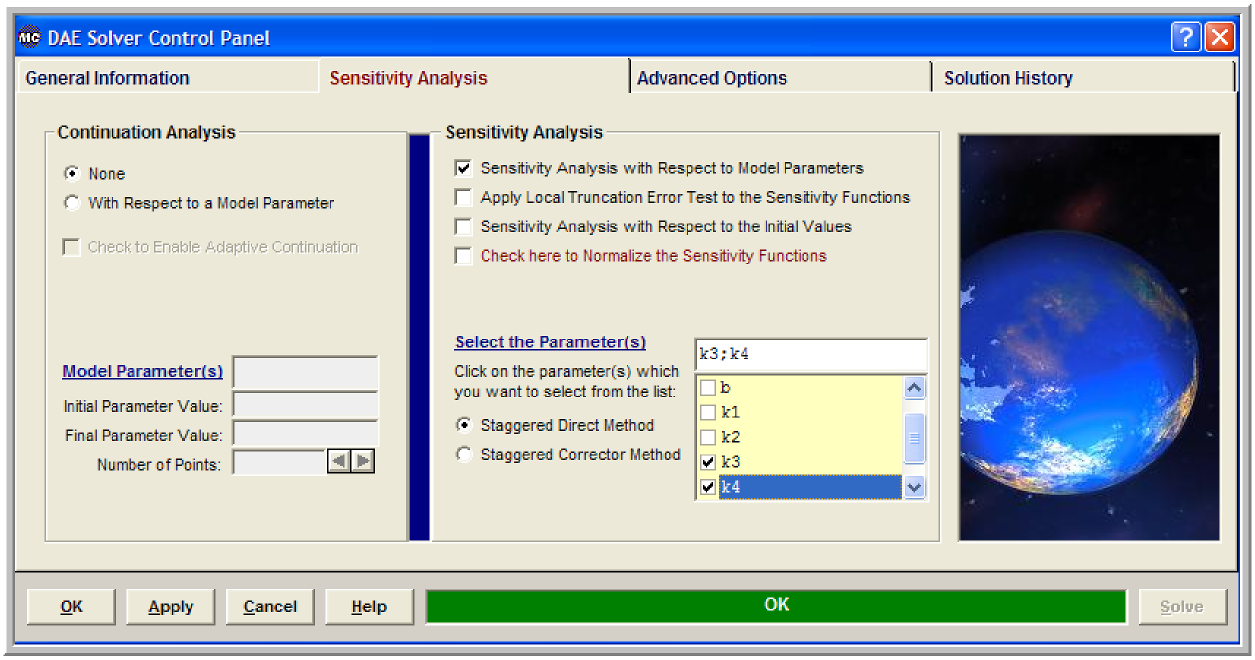
In the **Display Options** group enter the *Number of Variables* using the spin control and then, in the adjacent spreadsheet change the names of the state variables. Also from the **Show** group select *Solution Report*. Then click *Apply* and then *Solve*. You should see the results as shown in the image above. The final values correspond to the solution at the end of the integration. Notice also, other types of solution information appearing in the spreadsheet. If you wish to enter analytical derivatives with respect to one of the independent variables you must click the corresponding cell in the CheckBox column. We may exercise this option when we calculate the conversion of glucose as a function of reaction time.

3.2.6 Sensitivity Analysis Calculations

AthenaVisual Plus allows for convenient and efficient calculation of the first order sensitivity functions given by:

$$\mathbf{W}(t;\boldsymbol{\theta}) = \frac{\partial \mathbf{u}(t)}{\partial \boldsymbol{\theta}}$$

Suppose that we wish to investigate the sensitivity of the reaction mixture composition to small perturbations in the reaction rate constants k_3 and k_4. In order to do that, first we must load the solver (**Hit F12**), and click the *Sensitivity and Continuation* tab.



DAE Solver Control Panel

General Information | Sensitivity Analysis | Advanced Options | Solution History

Continuation Analysis

● None
○ With Respect to a Model Parameter
□ Check to Enable Adaptive Continuation

Model Parameter(s)
Initial Parameter Value:
Final Parameter Value:
Number of Points:

Sensitivity Analysis

☑ Sensitivity Analysis with Respect to Model Parameters
□ Apply Local Truncation Error Test to the Sensitivity Functions
□ Sensitivity Analysis with Respect to the Initial Values
□ Check here to Normalize the Sensitivity Functions

Select the Parameter(s): k3;k4
Click on the parameter(s) which you want to select from the list:

□ b
□ k1
□ k2
☑ k3
☑ k4

● Staggered Direct Method
○ Staggered Corrector Method

In the **Sensitivity Analysis** group click *Sensitivity Analysis with respect to Model Parameters* and subsequently select the parameters k_3, k_4 from the drop down list; optionally you may wish to click on the *Check here to Normalize the Sensitivity Functions* in which case the following first order coefficients will be calculated:

$$\mathbf{W}(t;\boldsymbol{\theta}) = \frac{\partial \mathbf{u}(t)}{\partial \ln \boldsymbol{\theta}}$$

There are two methods of calculating the sensitivity functions. The *Staggered Direct Method* and the *Staggered Corrector Method*. In the first method, the Jacobian matrix of the system is updated and decomposed at every integration step prior to calculating the sensitivity functions. The calculation then, is carried out by applying the Newton method once since the sensitivity equations are linear. In the second method the Jacobian matrix of the systems is updated at every step, but the LU decomposition of the last successful state integration step is used. This option is helpful for very large systems of equations. The calculation of sensitivities is then carried out by executing the Newton method a number of times selected by the user (minimum default is 2).

Now choose **OK** or click **Apply**. From the ***Build*** menu select ***Execute*** (or **Hit F5**). You should see the following results (*only a partial snapshot of the results is shown here*):

```
Number of State Equations.......................     4
Number of Sensitivity Parameters................     2
Number of Integration Output Points.............    22

 TIME         U(1,1)       U(1,2)       U(1,3)       U(2,1)       U(2,2)       U(2,3)
 0.00000E+00  5.60000E-01  0.00000E+00  0.00000E+00  1.28000E+00  0.00000E+00  0.00000E+00
 4.76190E-01  8.06528E-01  0.00000E+00  0.00000E+00  3.30736E+00  1.36996E-01 -3.05263E-02
 9.52381E-01  1.12023E+00  0.00000E+00  0.00000E+00  5.45140E+00  2.61417E-01 -6.02549E-02
 1.42857E+00  1.48852E+00  0.00000E+00  0.00000E+00  7.69245E+00  3.66565E-01 -8.89308E-02
 1.90476E+00  1.88213E+00  0.00000E+00  0.00000E+00  9.80014E+00  4.26869E-01 -1.11581E-01
 2.38095E+00  2.26274E+00  0.00000E+00  0.00000E+00  1.13652E+01  4.04263E-01 -1.17913E-01
 2.85714E+00  2.59678E+00  0.00000E+00  0.00000E+00  1.19523E+01  2.72800E-01 -9.60814E-02
 3.33333E+00  2.86590E+00  0.00000E+00  0.00000E+00  1.13510E+01  5.67522E-02 -4.39940E-02
 3.80952E+00  3.06818E+00  0.00000E+00  0.00000E+00  9.75886E+00 -1.61601E-01  2.07664E-02
 4.28571E+00  3.21240E+00  0.00000E+00  0.00000E+00  7.68673E+00 -3.00792E-01  7.14077E-02
 4.76190E+00  3.31141E+00  0.00000E+00  0.00000E+00  5.64883E+00 -3.40144E-01  9.40931E-02
 5.23810E+00  3.37764E+00  0.00000E+00  0.00000E+00  3.94536E+00 -3.09819E-01  9.28120E-02
 5.71429E+00  3.42116E+00  0.00000E+00  0.00000E+00  2.65891E+00 -2.49443E-01  7.85921E-02
 6.19048E+00  3.44944E+00  0.00000E+00  0.00000E+00  1.74831E+00 -1.85871E-01  6.06293E-02
 6.66667E+00  3.46767E+00  0.00000E+00  0.00000E+00  1.13022E+00 -1.31582E-01  4.40031E-02
 7.14286E+00  3.47936E+00  0.00000E+00  0.00000E+00  7.22124E-01 -8.99269E-02  3.06301E-02
 7.61905E+00  3.48684E+00  0.00000E+00  0.00000E+00  4.57633E-01 -5.99486E-02  2.07020E-02
 8.09524E+00  3.49162E+00  0.00000E+00  0.00000E+00  2.88366E-01 -3.92529E-02  1.36972E-02
 8.57143E+00  3.49466E+00  0.00000E+00  0.00000E+00  1.80978E-01 -2.53661E-02  8.92209E-03
 9.04762E+00  3.49660E+00  0.00000E+00  0.00000E+00  1.13260E-01 -1.62333E-02  5.74467E-03
 9.52381E+00  3.49784E+00  0.00000E+00  0.00000E+00  7.07379E-02 -1.03134E-02  3.66685E-03
 1.00000E+01  3.49863E+00  0.00000E+00  0.00000E+00  4.41171E-02 -6.51675E-03  2.32532E-03

EXIT DDAPLUS: SOLUTION FOUND

Number of Steps Taken Thus Far...................      126
Number of Function Evaluations...................     1142
Number of Jacobian Evaluations...................      127
Number of Jacobian Factorizations................      127
```

An interpretation of these results at the end of the reaction time is given in the following table using the nomenclature in our example (*again only partial results are shown*):

$C_1\left(t=10\right)=3.49863$	$\left.\frac{\partial C_1}{\partial k_3}\right\vert_{t=10}=0.0$	$\left.\frac{\partial C_1}{\partial k_4}\right\vert_{t=10}=0.0$
$C_2\left(t=8\right)=0.441171$	$\left.\frac{\partial C_2}{\partial k_3}\right\vert_{t=10}=-0.00651675$	$\left.\frac{\partial C_2}{\partial k_4}\right\vert_{t=10}=0.00232532$

Notice that the number of Jacobian evaluations (127) is the same as the number of Jacobian factorizations (127), since we chose the *Staggered Direct Method*.

3.2.7 Implicit Auxiliary Variables

AthenaVisual Plus 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 glucose $\left(C_4\right)$ in the batch tank reactor. We introduce an auxiliary variable $U(5) = X_{\text{Glu}}$ that represents the conversion and the associated equation that is obviously the definition of the conversion, i.e.

$$F(5) = U(5) - \frac{C_4\left(t=0\right) - C_4\left(t\right)}{C_4\left(t=0\right)} \times 100$$

We can implement this in Athena in a very straightforward manner. First we load the DDAPLUS solver (**Hit F12**) and increase the *Number of State Equations* to 5. 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:

```
! Declarations and Model Constants
!=================================
 Global a,b As Real
 Global k1,k2,k3,k4,k5 As Real

 a = 0.9082
 b = 1.011

 k1 = 0.95
 k2 = 3.5
 k3 = 18.0
 k4 = 37.5
 k5 = 1.1

@Initial Conditions
 U(1) = 0.56
 U(2) = 1.28
 U(3) = 0.16
 U(4) = 45.0

 U(5) = 0.0    ! Conversion

@Model Equations
 F(1) = k1 * U(1) * (1.0 - U(1) / k2)
 F(2) = k3 * U(1) * U(4) / (k4 + U(4)) - a * k5 * U(2)
 F(3) = k5 * U(2)
 F(4) = -b * k3 * U(1) * U(4) / (k4 + U(4))

 F(5) = U(5) - (45.0 - U(4))/45.0 * 100.0

@Coefficient Matrix
 E(1:4) = 1.0
 E(5) = 0.0
```

Notice the new section that supplies the elements of the matrix of the coefficients for the time derivatives. This is necessary, because the conversion equation is algebraic. Therefore our systems is no longer a set of pure ordinary differential equations and therefore it must be changed. In order to change the type of system we must load the solver (**Hit F12**) and from the **System Identification** group we select *Mixed System with Diagonal E Matrix*. Then we **Hit F11** and AthenaVisual Plus will insert the heading for the Coefficient matrix. We then enter the elements of the coefficient matrix as shown in the code above.

Now from the ***Build*** menu select ***Execute*** (or **Hit F5**). You should see the following results:

```
 Number of State Equations.......................     5
 Number of Sensitivity Parameters................     0
 Number of Integration Output Points.............    22

  TIME          U(1)          U(2)          U(3)          U(4)          U(5)
  0.00000E+00   5.60000E-01   1.28000E+00   1.60000E-01   4.50000E+01   0.00000E+00
  4.76190E-01   8.06528E-01   3.30736E+00   1.35857E+00   4.18498E+01   7.00039E+00
  9.52381E-01   1.12023E+00   5.45140E+00   3.64626E+00   3.75817E+01   1.64852E+01
  1.42857E+00   1.48852E+00   7.69245E+00   7.08777E+00   3.21560E+01   2.85422E+01
  1.90476E+00   1.88213E+00   9.80013E+00   1.16829E+01   2.58060E+01   4.26534E+01
  2.38095E+00   2.26274E+00   1.13652E+01   1.72601E+01   1.91027E+01   5.75495E+01
  2.85714E+00   2.59678E+00   1.19523E+01   2.34172E+01   1.28558E+01   7.14315E+01
  3.33333E+00   2.86590E+00   1.13510E+01   2.95714E+01   7.81295E+00   8.26379E+01
  3.80952E+00   3.06818E+00   9.75888E+00   3.51337E+01   4.31538E+00   9.04103E+01
  4.28571E+00   3.21240E+00   7.68674E+00   3.97113E+01   2.20721E+00   9.50951E+01
  4.76190E+00   3.31141E+00   5.64883E+00   4.31938E+01   1.06986E+00   9.76225E+01
  5.23810E+00   3.37764E+00   3.94536E+00   4.56888E+01   5.01222E-01   9.88862E+01
  5.71429E+00   3.42116E+00   2.65892E+00   4.74006E+01   2.30067E-01   9.94887E+01
  6.19048E+00   3.44944E+00   1.74831E+00   4.85402E+01   1.04338E-01   9.97681E+01
  6.66667E+00   3.46767E+00   1.13022E+00   4.92832E+01   4.69835E-02   9.98956E+01
  7.14286E+00   3.47936E+00   7.22124E-01   4.97608E+01   2.10671E-02   9.99532E+01
  7.61905E+00   3.48684E+00   4.57634E-01   5.00647E+01   9.42218E-03   9.99791E+01
  8.09524E+00   3.49162E+00   2.88366E-01   5.02567E+01   4.20773E-03   9.99906E+01
  8.57143E+00   3.49466E+00   1.80978E-01   5.03775E+01   1.87738E-03   9.99958E+01
  9.04762E+00   3.49660E+00   1.13260E-01   5.04532E+01   8.37149E-04   9.99981E+01
  9.52381E+00   3.49784E+00   7.07375E-02   5.05005E+01   3.73156E-04   9.99992E+01
  1.00000E+01   3.49863E+00   4.41167E-02   5.05301E+01   1.66291E-04   9.99996E+01

 EXIT DDAPLUS: SOLUTION FOUND

 Number of Steps Taken Thus Far..................       136
 Number of Function Evaluations..................       323
 Number of Jacobian Evaluations..................         9
 Number of Jacobian Factorizations...............         9
```

From these results we observe that the conversion is $X_4 = 99.99\%$. We can also choose to plot the glucose conversion as a function of reaction time, by loading the Graphics Control Panel, and following the exact same procedure outlined above for plotting the time profiles of the system state variables.

3.2.8 Explicit Auxiliary Variables

Auxiliary variables can also be implemented in an alternative (explicit) way, without having to increase the dimensionality of the original system of equations. In order to do that we have to make use of two **AthenaVisual Plus** options; the first allows us to enter custom code right after a call to the integrator, and the second allows us to append auxiliary variables to the state vector for printing purposes.

As an example, consider the introduction of two auxiliary variables, *x_glucose* and *s_gluAcid* that represent the glucose conversion and the selectivity of glucose to gluconic acid respectively; these auxiliary variables are therefore defined by the following equations:

$$x\_glucose = \frac{45.0 - U(4)}{45.0} \times 100.0$$

$$s\_gluAcid = \frac{U(3)}{U(4)}$$

The code below illustrates how to do the implementation:

```
@Initial Conditions
 U(1) = 0.56
 U(2) = 1.28
 U(3) = 0.16
 U(4) = 45.0

@Model Equations
 F(1) = k1 * U(1) * (1.0 - U(1) / k2)
 F(2) = k3 * U(1) * U(4) / (k4 + U(4)) - a * k5 * U(2)
 F(3) = k5 * U(2)
 F(4) = -b * k3 * U(1) * U(4) / (k4 + U(4))

@After Calling Solver
 Dim x_glucose, x_gluAcid As Real

 x_glucose=(45.0-U(4))/45.0*100.0  ! Glucose Conversion
 x_gluAcid=U(3)/U(4)               ! Gluconic Acid Selectivity

@Solver Options
 Tend=10                           ! End of Integration
 Npts=15                           ! Number of Output Points
 Append=x_glucose;x_gluAcid        ! Auxiliary Variables
```

Notice above in the Solver Options section, the **Tend** keyword specifies the end of the integration of the system of differential equations while the **Npts** keyword specifies the number of output points for printing purposes. The **Append** keyword instructs the solver to append the auxiliary variables to the state vector so that they can be printed and therefore used in the graphics server. To view all Solver Options form the **View** menu in AthenaVisual Plus select **Solver Options**.

Now from the ***Build*** menu select ***Execute*** (or **Hit F5**). You should see the following results:

```
Number of State Equations.......................    4
Number of Sensitivity Parameters................    0
Number of Integration Output Points.............   17

 TIME           U(1)           U(2)           U(3)           U(4)           x_glucose      x_gluAcid
 0.00000E+00    5.60000E-01    1.28000E+00    1.60000E-01    4.50000E+01    0.00000E+00    3.55556E-03
 6.25000E-01    8.97591E-01    3.96182E+00    1.95333E+00    4.06421E+01    9.68431E+00    4.80618E-02
 1.25000E+00    1.34555E+00    6.85062E+00    5.65930E+00    3.43187E+01    2.37362E+01    1.64904E-01
 1.87500E+00    1.85751E+00    9.67947E+00    1.13640E+01    2.62207E+01    4.17317E+01    4.33398E-01
 2.50000E+00    2.35164E+00    1.16178E+01    1.87656E+01    1.74650E+01    6.11889E+01    1.07447E+00
 3.12500E+00    2.75660E+00    1.17575E+01    2.69195E+01    9.83687E+00    7.81403E+01    2.73660E+00
 3.75000E+00    3.04631E+00    9.99600E+00    3.44869E+01    4.66950E+00    8.96233E+01    7.38556E+00
 4.37500E+00    3.23401E+00    7.28951E+00    4.04467E+01    1.93356E+00    9.57032E+01    2.09183E+01
 5.00000E+00    3.34793E+00    4.74602E+00    4.45530E+01    7.34679E-01    9.83674E+01    6.06428E+01
 5.62500E+00    3.41435E+00    2.86930E+00    4.71292E+01    2.66531E-01    9.94077E+01    1.76824E+02
 6.25000E+00    3.45218E+00    1.65685E+00    4.86516E+01    9.44648E-02    9.97901E+01    5.15024E+02
 6.87500E+00    3.47343E+00    9.30183E-01    4.95186E+01    3.30918E-02    9.99265E+01    1.49640E+03
 7.50000E+00    3.48528E+00    5.13226E-01    5.00012E+01    1.15236E-02    9.99744E+01    4.33903E+03
 8.12500E+00    3.49185E+00    2.80120E-01    5.02660E+01    4.00057E-03    9.99911E+01    1.25647E+04
 8.75000E+00    3.49550E+00    1.51851E-01    5.04101E+01    1.38705E-03    9.99969E+01    3.63433E+04
 9.37500E+00    3.49751E+00    8.19616E-02    5.04881E+01    4.80521E-04    9.99989E+01    1.05069E+05
 1.00000E+01    3.49863E+00    4.41169E-02    5.05301E+01    1.66392E-04    9.99996E+01    3.03681E+05

EXIT DDAPLUS: SOLUTION FOUND

Number of Steps Taken Thus Far...................      128
Number of Function Evaluations...................      284
Number of Jacobian Evaluations...................        6
Number of Jacobian Factorizations................        6
```

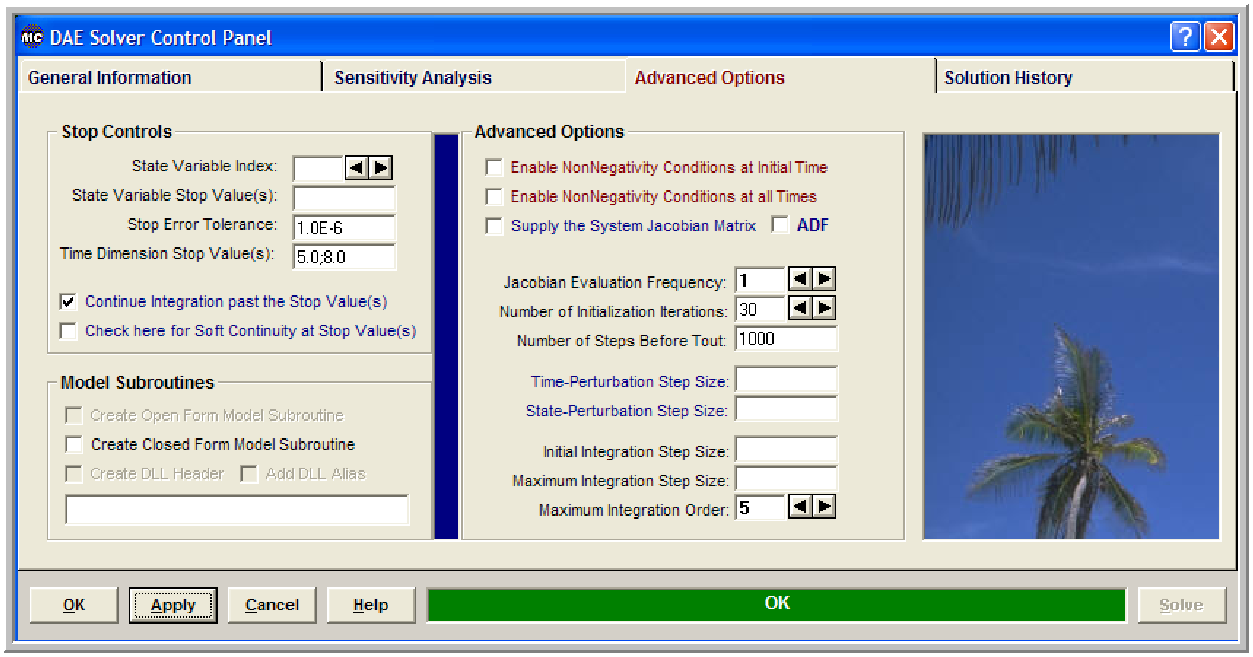
From the above table we clearly see how the AthenaVisual Plus solver augmented the auxiliary variables to the state vector for printing purposes. One can now access the Graphics control panel to graph the auxiliary variables along with the state variables.

3.2.9 Integration with Implicit or Explicit Discontinuities

AthenaVisual Plus offers a convenient way to integrate systems of Differential/Algebraic equations that exhibit implicit or explicit discontinuities in the independent variable. To illustrate this capability assume that for our example, the rate of cell growth formation abruptly changes according to the following equations:

$$k_1 = \begin{cases} 0.950 & 0 \le t \le 5 \\ 0.425 & 5 < t \le 8 \\ 0.000 & 8 < t \le 10 \end{cases}$$

First we must specify in the AthenaVisual Plus solver the location of the explicit discontinuities defined with the above equations. To do that we first load the solver (**Hit F12**).



DAE Solver Control Panel

Stop Controls

State Variable Index:

State Variable Stop Value(s):

Stop Error Tolerance: 1.0E-6

Time Dimension Stop Value(s): 5.0;8.0

☑ Continue Integration past the Stop Value(s)

□ Check here for Soft Continuity at Stop Value(s)

Model Subroutines

□ Create Open Form Model Subroutine

□ Create Closed Form Model Subroutine

□ Create DLL Header □ Add DLL Alias

Advanced Options

□ Enable NonNegativity Conditions at Initial Time

□ Enable NonNegativity Conditions at all Times

□ Supply the System Jacobian Matrix □ ADF

Jacobian Evaluation Frequency: 1

Number of Initialization Iterations: 30

Number of Steps Before Tout: 1000

Time-Perturbation Step Size:

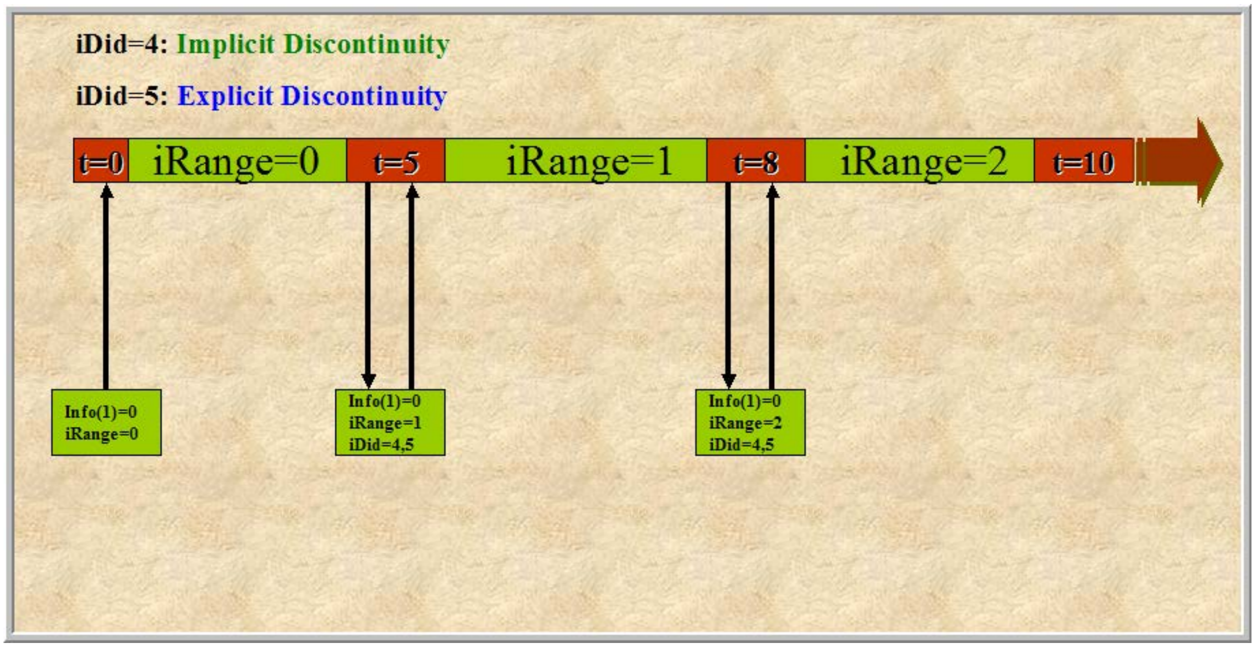
State-Perturbation Step Size:

Initial Integration Step Size:

Maximum Integration Step Size:

Maximum Integration Order: 5

We first click the *Advanced Options* tab. In the **Stop Controls** group, we enter the values 5.0 and 8.0 separated by semicolon(;) in the *Time Dimension Stop Values(s)* text box. We also click the option *Continue the Integration Past the Stop Value(s)*, so that the integrator will span the entire reaction time from zero to 10 hours. Otherwise it will stop after 5 hours. The click **OK**. When we do that the solver divides the domain of integration in three ranges (zones) identified by the internal variable **iRange** as indicated in the pictorial below:



We can then use the **iRange** variable to enter the information about the reaction rate constant as indicated in the code below:

```
@Model Equations

 Select Case (iRange)
  Case (0)
   k1=0.95

  Case (1)
   k1=0.425

  Case Default
   k1=Zero

 End Select

 F(1) = k1 * U(1) * (1.0 - U(1) / k2)
 F(2) = k3 * U(1) * U(4) / (k4 + U(4)) - a * k5 * U(2)
 F(3) = k5 * U(2)
 F(4) = -b * k3 * U(1) * U(4) / (k4 + U(4))

 F(5) = U(5) - (45.0 - U(4))/45.0 * 100.0

@Coefficient Matrix
 E(1:4) = 1.0
 E(5) = 0.0
```

3.2.10 Root Finding and Implicit Discontinuities

AthenaVisual Plus offers a convenient way to find roots (values of the independent variable t) of prescribed event functions of the form:

$$h_i\left(t_i^*,\mathbf{x},\mathbf{u};\boldsymbol{\theta}\right)=0 \qquad i=1,2,\ldots m$$

When the Athena solver locates the root(s), it divides the domain of integration domain, in two or more ranges (depending on the number of the event functions $h_i\left(t_i^*\right)$) identified by the internal variable **iRange**. For instance in the case of one event function, the value of **iRange=0** denotes the first zone from $0\le t\le t^*$, the value **iRange=1** denotes the second zone from $t^*\le t\le T$ where T indicates the end of the integration.

To illustrate these concepts, consider as an example the problem of finding the time (t^*) at which the concentration of the gluconolactone attains its maximum value. The equation that describes this point in time is given by:

$$h\left(t^*,\mathbf{x},\mathbf{u};\boldsymbol{\theta}\right)=\frac{k_3C_1C_4}{k_4+C_4}-ak_5C_2=0$$

To solve this problem, we introduce an auxiliary variable $U(5)$ that represents the above event function and also the associated auxiliary residual, which, at the point where the concentration of gluconolactone attains its maximum value, is given by:

$$F(5)=U(5)-\frac{k_3C_1C_4}{k_4+C_4}-ak_5C_2$$

We can implement this in AthenaVisual Plus in a very straightforward manner. To begin, we load the solver by hitting **Hit F12** . When the **DAE Solver Control Panel** window appears we increase the *Number of State Equations* to 5. Since the auxiliary equation is algebraic, our system is no longer a set of pure ordinary differential equations and therefore it must be changed. To do that, from the **System Identification** group we select *Mixed System with Diagonal E Matrix*. We also select from the **System Options** *Check here if the E matrix is constant* since this is the case for our example. Then click **OK.** Now we **Hit F11** to insert the heading for the **@Coefficient Matrix** and enter the code as shown below:

```
@Initial Conditions
 U(1) = 0.56  ! Cell
 U(2) = 1.28  ! Gluconolactone
 U(3) = 0.16  ! Gluconic Acid
 U(4) = 45.0  ! Glucose

@Model Equations
 F(1) = k1 * U(1) * (1.0 - U(1) / k2)
 F(2) = k3 * U(1) * U(4) / (k4 + U(4)) - a * k5 * U(2)
 F(3) = k5 * U(2)
 F(4) = -b * k3 * U(1) * U(4) / (k4 + U(4))

 F(5)=U(5)-F(2)

@Coefficient Matrix
 E(1:4)=1.0
 E(5)=0.0
```

Next we load the solver again and select the **Advanced Options** tab; in the **Stop Controls** group, we enter the value 0.0 in the *State Variable Stop Values(s)* text box and using the spin control we select state variable U(**5**). These actions instruct the integrator to locate the value of the independent variable *t* for which the state variable U(5) becomes equal to zero. Click **OK** to accept the changes.

Now from the ***Build*** menu select ***Execute*** (or **Hit F5**). You should see the following results:

```
 Number of State Equations.......................     5
 Number of Sensitivity Parameters................     0
 Number of Integration Output Points.............    22

  TIME          U(1)          U(2)          U(3)          U(4)          U(5)
  0.00000E+00   5.60000E-01   1.28000E+00   1.60000E-01   4.50000E+01   4.21944E+00
  4.76190E-01   8.06528E-01   3.30736E+00   1.35857E+00   4.18498E+01   4.35255E+00
  9.52381E-01   1.12023E+00   5.45140E+00   3.64627E+00   3.75817E+01   4.64700E+00
  1.42857E+00   1.48852E+00   7.69245E+00   7.08777E+00   3.21560E+01   4.68400E+00
  1.90476E+00   1.88213E+00   9.80013E+00   1.16829E+01   2.58060E+01   4.01961E+00
  2.38095E+00   2.26274E+00   1.13652E+01   1.72601E+01   1.91027E+01   2.39159E+00
  2.85579E+00   2.59591E+00   1.19523E+01   2.33993E+01   1.28722E+01   2.50569E-08

 EXIT DDAPLUS: SOLUTION FOUND

 Number of Steps Taken Thus Far..................       56
 Number of Function Evaluations..................      173
 Number of Jacobian Evaluations..................       11
 Number of Jacobian Factorizations...............       11
```

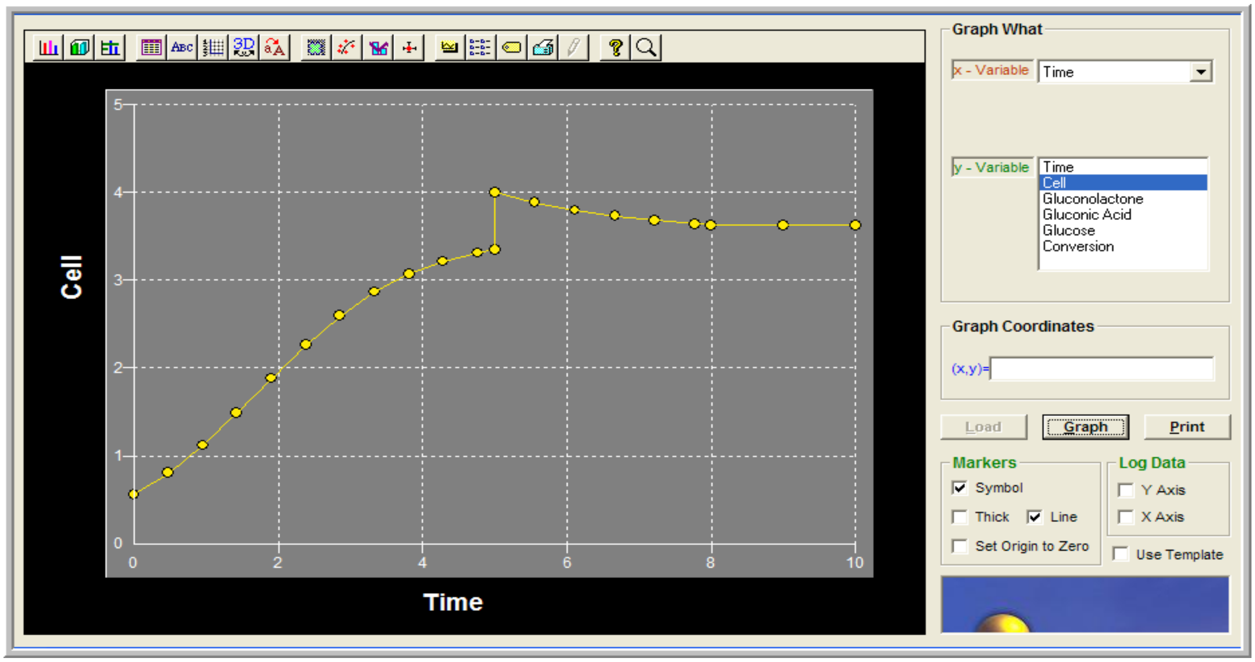
We can see from these results that the integrator stops at $t^* = 2.86$ at which point the value of the state variable U(5) which is equal to the rate of change of gluconolactone becomes equal to zero, i.e., the concentration of gluconolactone at this time attains it’s maximum value. If we wish to continue the integration past the root of the algebraic equations we must click the option *Continue Integration Past the Stop Values(s)* in the **Advanced Options** tab of the Athena **DDAPLUS** solver.

3.2.11 Changing Initial Conditions at Discontinuities

This option in AthenaVisual Plus is very powerful since not only can we change the values of system constants, but also change the form of equations or even enter new initial value at each different range. Suppose for example that since the cell growth was depressed we decide to inject more cells after 5 hours so that the effective cell concentration was $\mathbf{U}(1) = C_1(t=5) = 4.0$. In order to implement this initial condition in our problems we use the option of inserting special code prior to calling the integrator. To do that from the ***Model*** menu we select ***Add Custom Code*** and then click ***Before Calling Solver***. We than enter the code that is indicated below:

```
@Before Calling Solver
 If(iRange==1 .AND. Info(1)==0)U(1)=4.0
```

In the above code, **Info(1)** is a reserved variable that is set by Athena; in the very first call to the integrator the value of **Info(1)** is set equal to zero; in the subsequent calls the value of **Info(1)** is set equal to one. When the integration of a new range begins Athena resets **Info(1)** to zero again. If we build and run our model now, and plot the cell concentration as a function of time we will see the following graph:



x - Variable: Time

y - Variable: Time, **Cell** (selected), Gluconolactone, Gluconic Acid, Glucose, Conversion

☑ Symbol

□ Thick ☑ Line

□ Set Origin to Zero

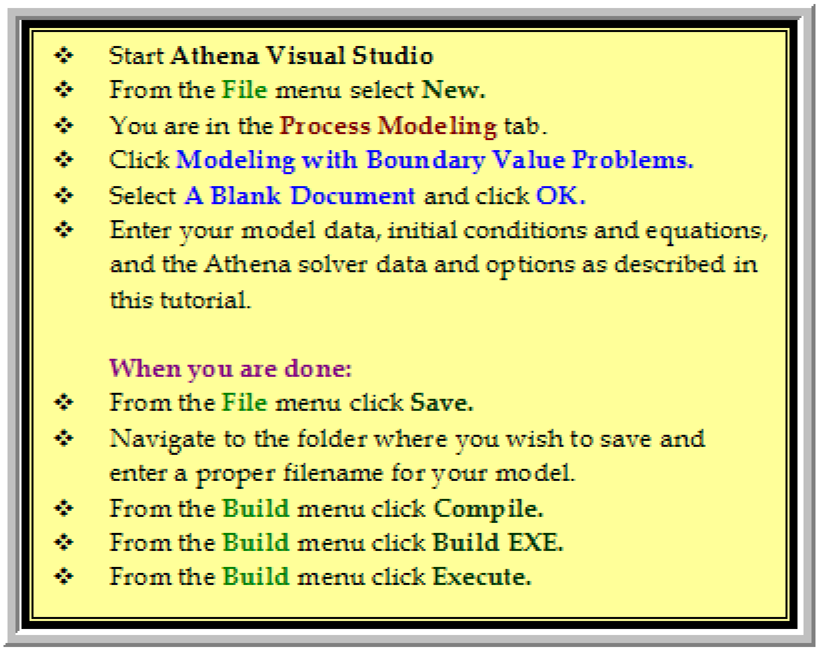
□ Y Axis

□ X Axis

□ Use Template

We can clearly see the three different ranges $\{[0,5]\cup[5,8]\cup[8,10]\}$, and also the new initial conditions in the beginning of the second range $5 < t \le 8$. Information about the use of the **Info(1)** flag, that is used for the specification of the new initial conditions, can be found in the **Athena Solvers Help** manual. From the AthenaVisual Plus ***Help*** menu select ***Athena Solvers*** and click on the **Info( )** array subject.

4.0 Analysis of Boundary Value Problems



- 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.**

Boundary Value problems arise in the steady-state 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:

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

Boundary Conditions

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

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

where **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 **u( )** with respect to the dimension *x*, and $\mathbf{u}_{xx}$**( )** represents the second order derivative of the state vector **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 AthenaVisual Plus powerful damped Newton algorithm which is encoded in the **PDAPLUS** solver.

4.1 Reactor Modeling with Axial Dispersion

This example problem has been created to test the functionality of AthenaVisual Plus 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

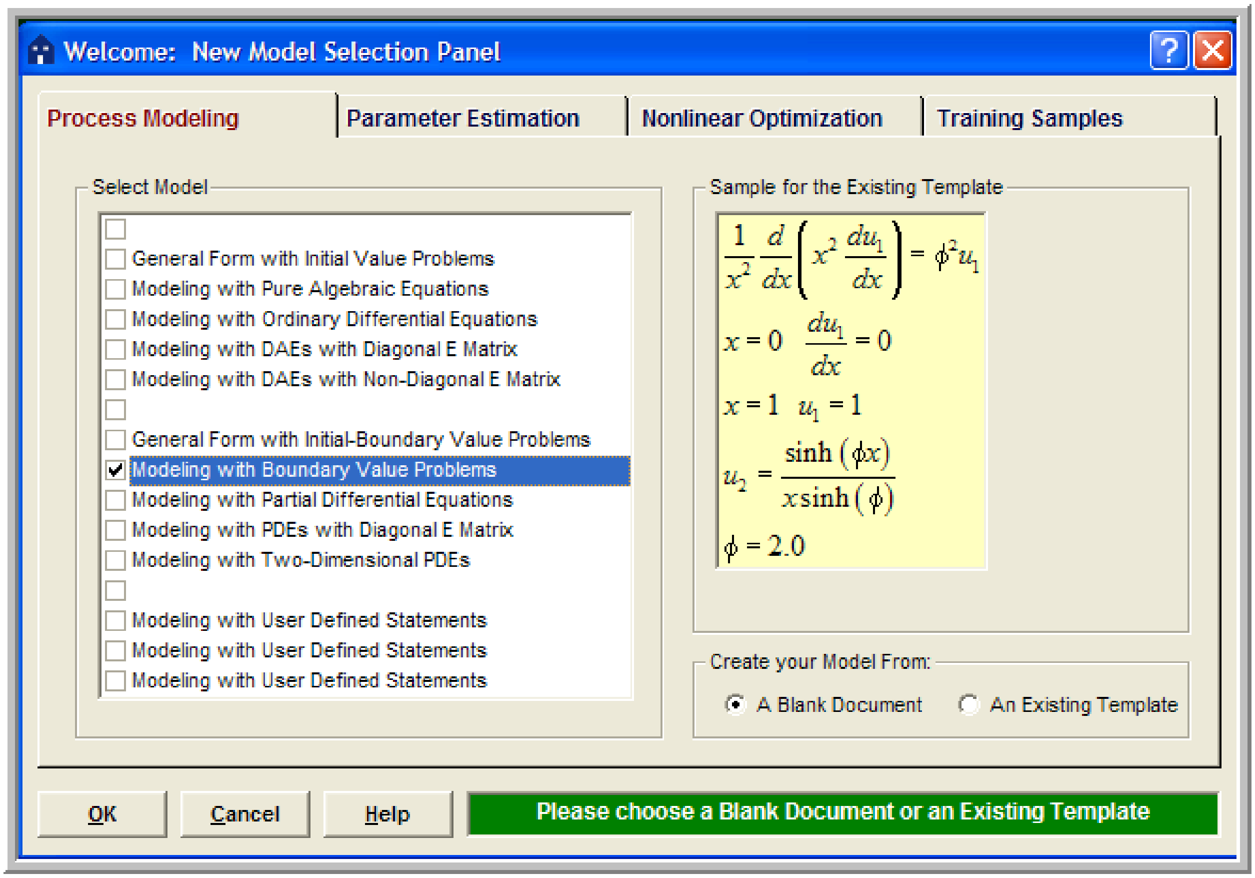
This sample tutorial is already precoded in AthenaVisual Plus. If you do not wish to type the code on your own you may access it by doing the following:

- Open **AthenaVisual Plus**
- From the **File** menu click **New**
- Select the **Training Samples** tab
- Select the **Steady-State Reactor with Axial Dispersion** sample
- Click **OK**

4.2 Implementation in AthenaVisual Plus

The following step by step process describes the model implementation in AthenaVisual Plus

- Open **AthenaVisual Plus**. From the ***File*** menu, choose ***New***. The *Welcome: New Model Selection Panel* window appears.



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

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.

4.2.1 Writing the Source Code for Boundary Value Problems

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 AthenaVisual Plus 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 AthenaVisual Plus syntax rules detailed below:

4.2.1.1 Data Section

In the data section (which may not be needed and is not labeled by AthenaVisual Plus) 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      ! Damköhler 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
```

4.2.1.2 Declaration of Variables

Global Variables: To declare global variables in the AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus environment. The modeling code is NOT case sensitive.

4.2.1.3 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
```

4.2.1.4 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
```

4.2.1.5 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:

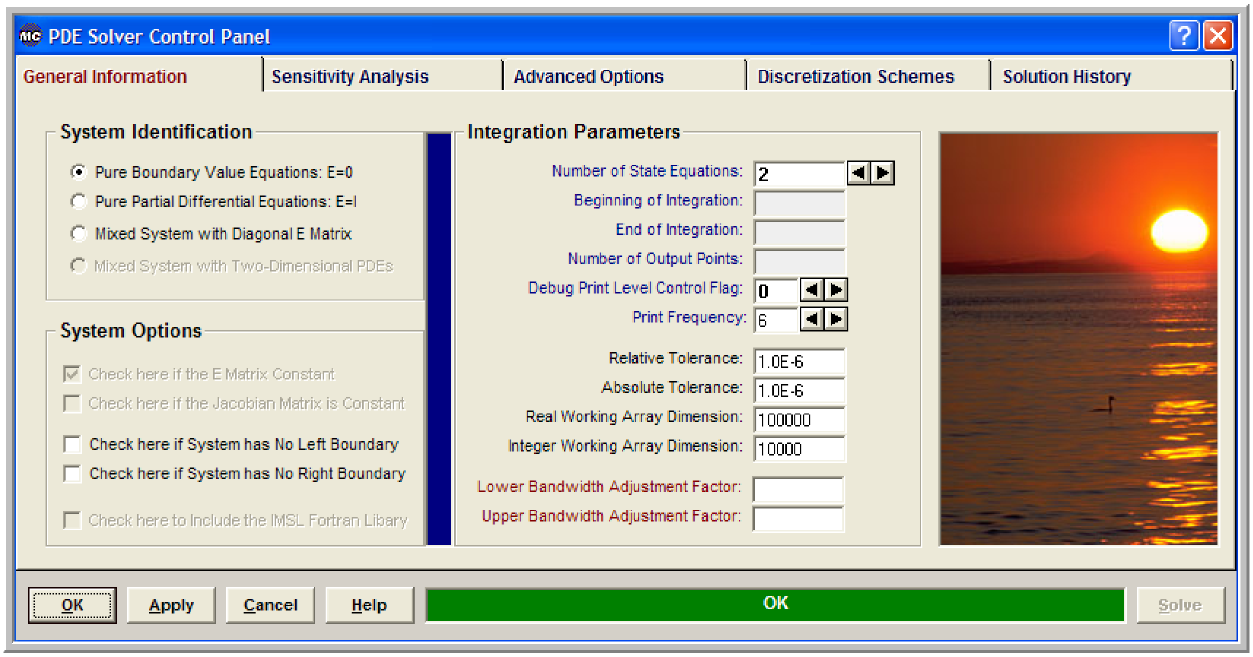
- From the **Model** menu choose **Boundary Conditions (**or **Hit F11)**
- Enter the source code as shown below for our example.

```
@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
```

4.2.2 Entering the Information about the PDAPLUS Solver

It is now time to access the AthenaVisual Plus 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**)
- Enter the solver parameters as shown below for our example



System Identification

- ● Pure Boundary Value Equations: E=0
- ○ Pure Partial Differential Equations: E=I
- ○ Mixed System with Diagonal E Matrix
- ○ Mixed System with Two-Dimensional PDEs

System Options

- ☑ Check here if the E Matrix Constant
- □ Check here if the Jacobian Matrix is Constant
- □ Check here if System has No Left Boundary
- □ Check here if System has No Right Boundary
- □ Check here to Include the IMSL Fortran Libary

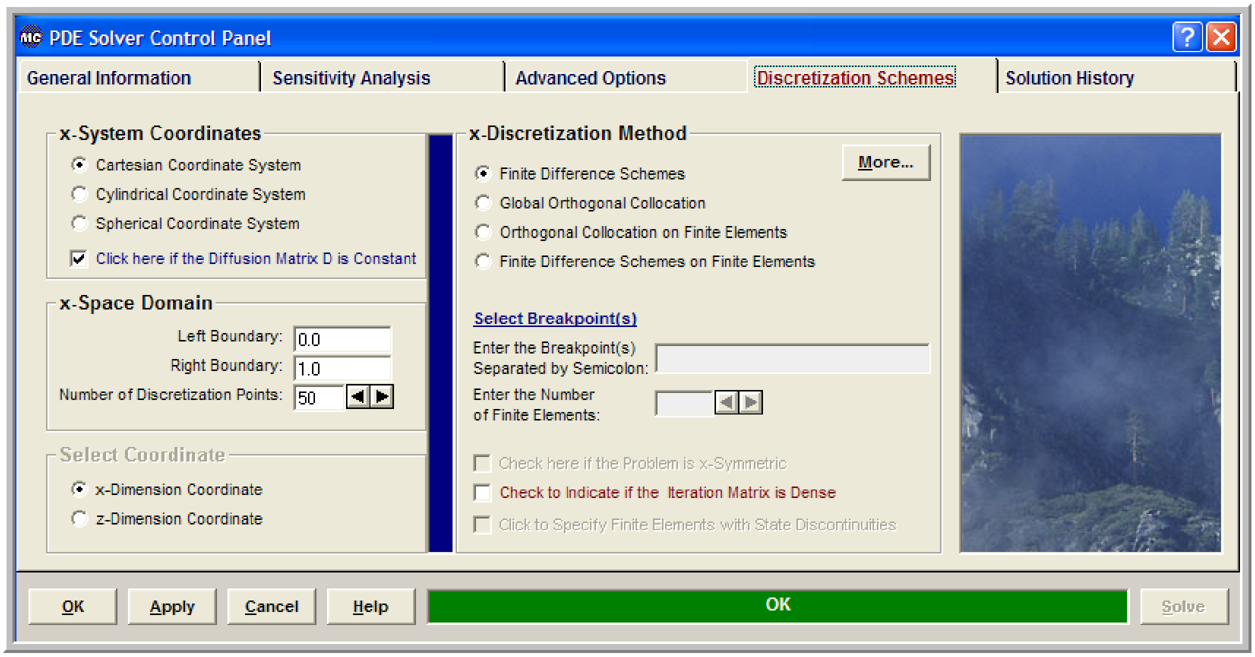
Integration Parameters

Number of State Equations:	2
Beginning of Integration:	
End of Integration:	
Number of Output Points:	
Debug Print Level Control Flag:	0
Print Frequency:	6
Relative Tolerance:	1.0E-6
Absolute Tolerance:	1.0E-6
Real Working Array Dimension:	100000
Integer Working Array Dimension:	10000
Lower Bandwidth Adjustment Factor:	
Upper Bandwidth Adjustment Factor:	

OK

The **PDE Solver Control Panel** window appears. 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



x-System Coordinates

● Cartesian Coordinate System
○ Cylindrical Coordinate System
○ Spherical Coordinate System
☑ Click here if the Diffusion Matrix D is Constant

x-Space Domain

Left Boundary: 0.0
Right Boundary: 1.0
Number of Discretization Points: 50

Select Coordinate

● x-Dimension Coordinate
○ z-Dimension Coordinate

x-Discretization Method

● Finite Difference Schemes
○ Global Orthogonal Collocation
○ Orthogonal Collocation on Finite Elements
○ Finite Difference Schemes on Finite Elements

Select Breakpoint(s)

Enter the Breakpoint(s) Separated by Semicolon:
Enter the Number of Finite Elements:

□ Check here if the Problem is x-Symmetric
□ Check to Indicate if the Iteration Matrix is Dense
□ Click to Specify Finite Elements with State Discontinuities

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.**

4.2.3 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**)

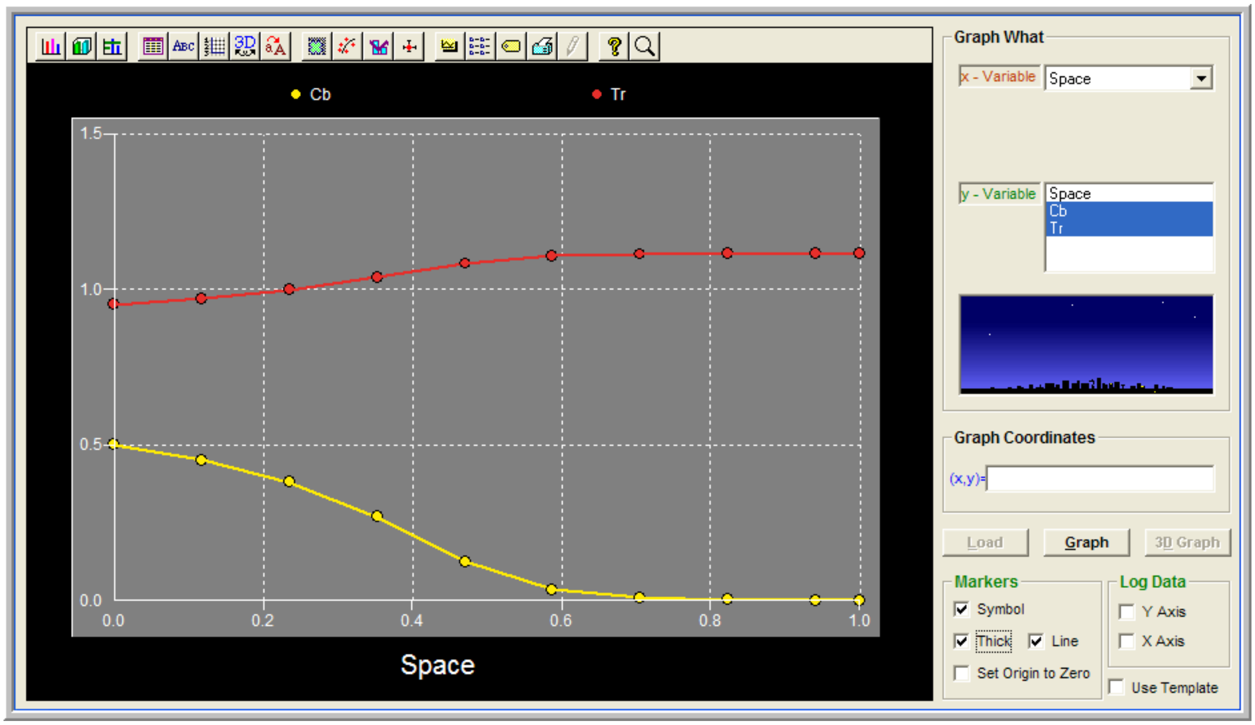
4.2.4 Numerical Results

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
```

4.2.5 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 The AthenaVisual Plus graphics control panel appears:



x - Variable: Space

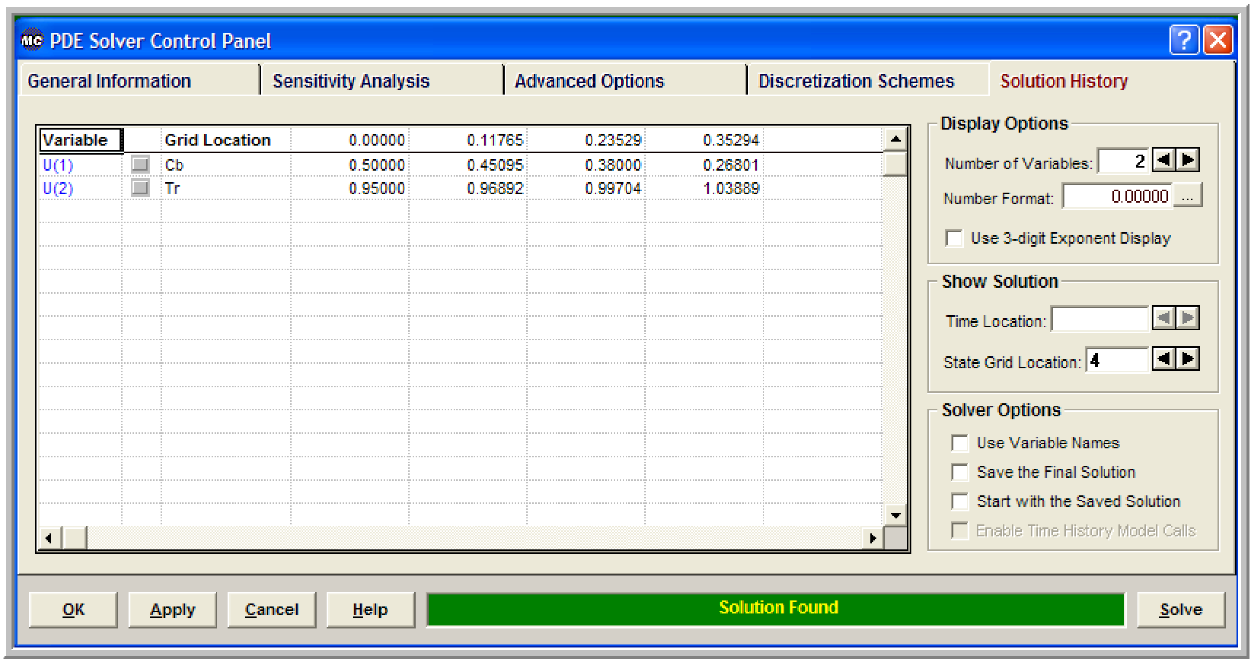
y - Variable: Space, Cb, Tr

☑ Symbol
☑ Thick ☑ Line
☐ Set Origin to Zero
☐ Y Axis
☐ X Axis
☐ Use Template

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.

4.2.6 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:



Variable		Grid Location	0.00000	0.11765	0.23529	0.35294
U(1)	□	Cb	0.50000	0.45095	0.38000	0.26801
U(2)	□	Tr	0.95000	0.96892	0.99704	1.03889

Display Options

Number of Variables: 2

Number Format: 0.00000

□ Use 3-digit Exponent Display

Show Solution

Time Location:

State Grid Location: 4

Solver Options

□ Use Variable Names

□ Save the Final Solution

□ Start with the Saved Solution

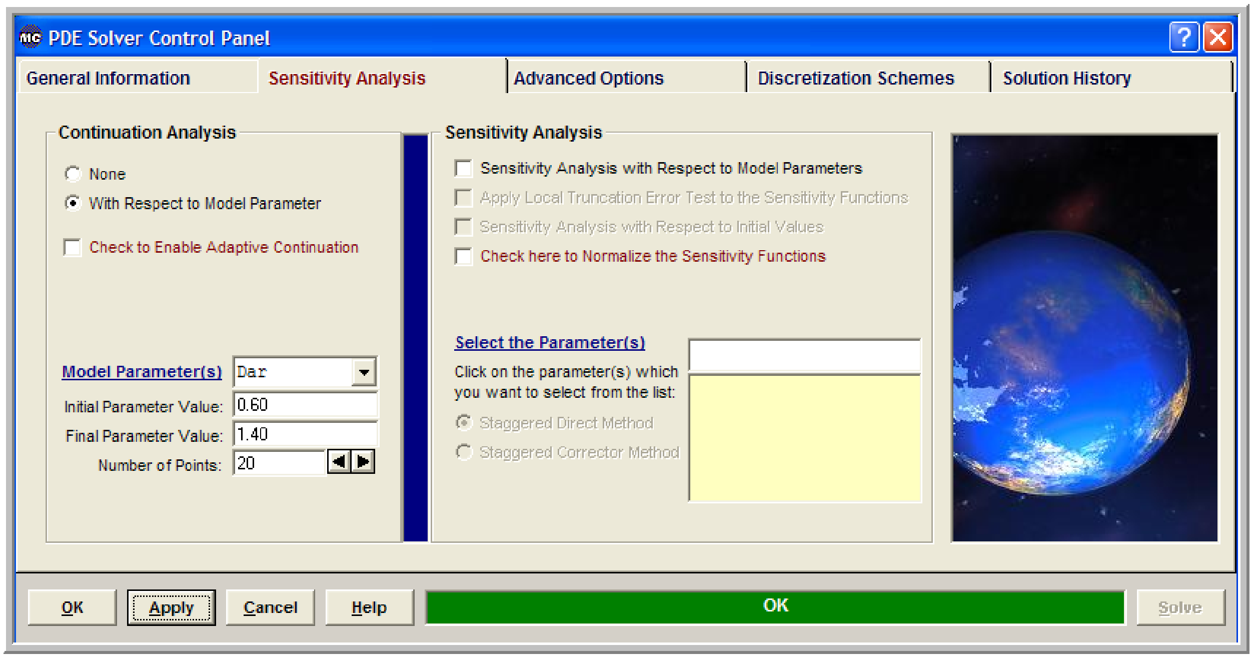
□ Enable Time History Model Calls

Solution Found

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.

4.2.7 Continuation Analysis

AthenaVisual Plus 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.



PDE Solver Control Panel

Continuation Analysis

○ None

● With Respect to Model Parameter

□ Check to Enable Adaptive Continuation

Model Parameter(s): Dar

Initial Parameter Value: 0.60

Final Parameter Value: 1.40

Number of Points: 20

Sensitivity Analysis

□ Sensitivity Analysis with Respect to Model Parameters

□ Apply Local Truncation Error Test to the Sensitivity Functions

□ Sensitivity Analysis with Respect to Initial Values

□ Check here to Normalize the Sensitivity Functions

Select the Parameter(s)

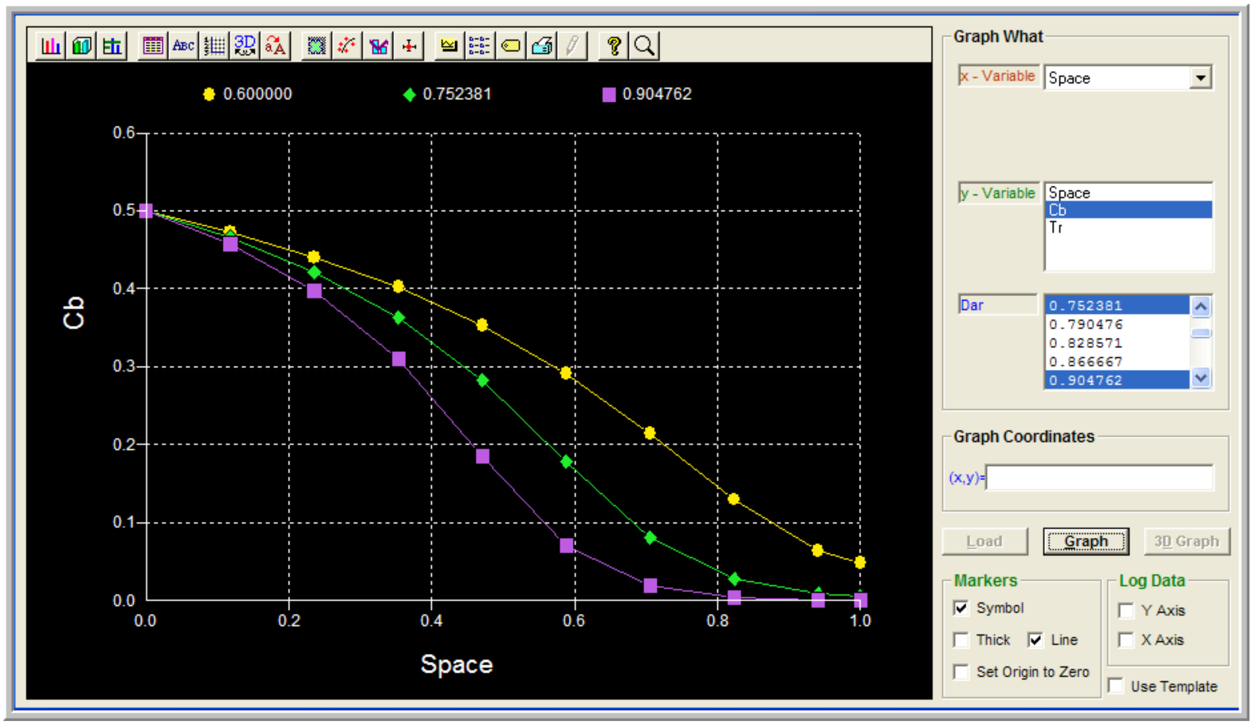
Click on the parameter(s) which you want to select from the list:

● Staggered Direct Method

○ Staggered Corrector Method

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:



Graph What

x - Variable: Space

y - Variable: Space, **Cb**, Tr

Dar: **0.752381**, 0.790476, 0.828571, 0.866667, **0.904762**

Graph Coordinates

(x,y)=

Markers

☑ Symbol

□ Thick ☑ Line

□ Set Origin to Zero

Log Data

□ Y Axis

□ X Axis

□ Use Template

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

4.2.8 Implicit Auxiliary Variables

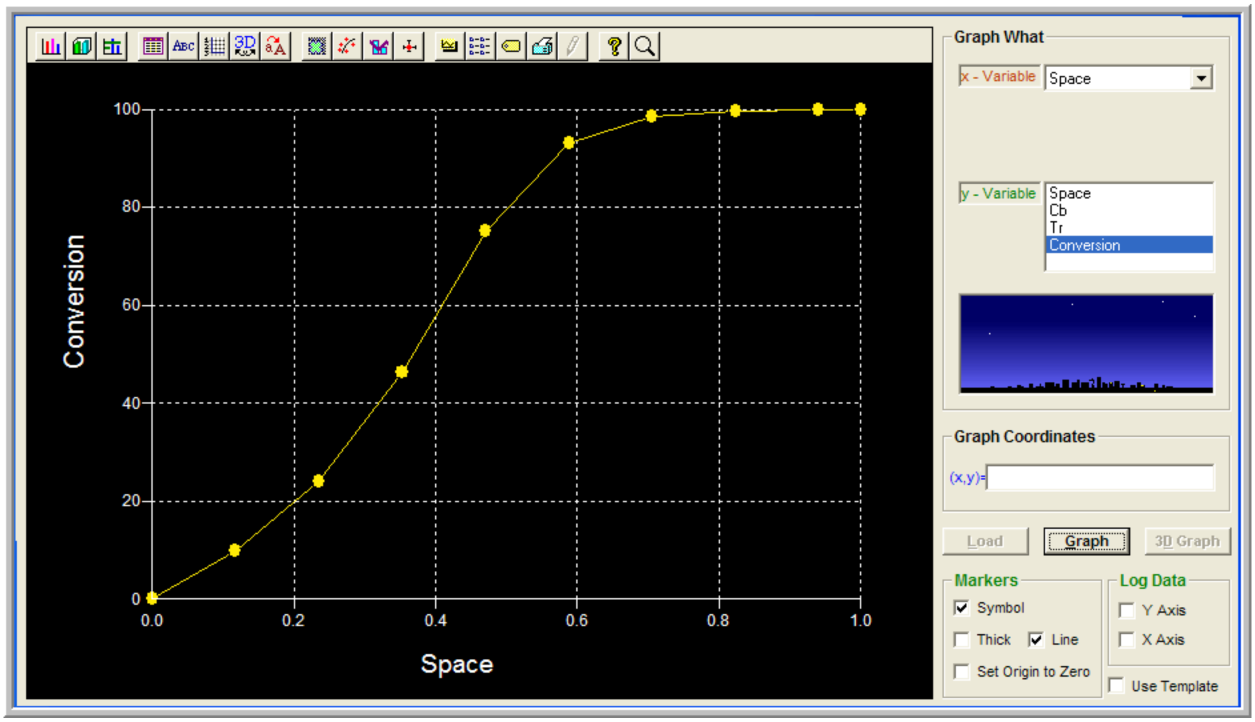
AthenaVisual Plus allows for convenient and efficient calculation of derived quantities by the introduction of auxiliary variables. Assume for example that we wish to estimate the conversion of the reactant B in the plug flow reactor. We introduce an auxiliary variable $U(3)=X_B$ that represents the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

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

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:

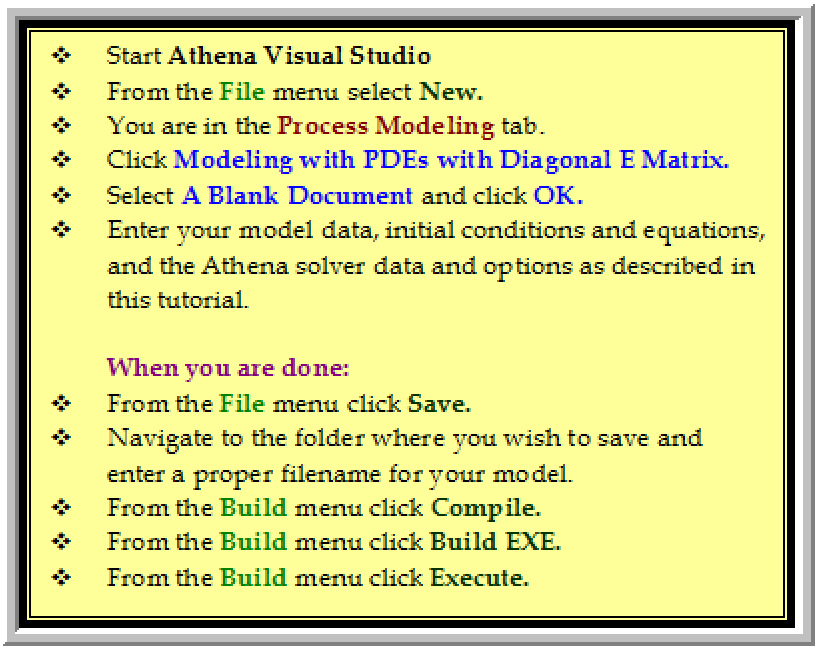
```
@Initial Conditions
 U(1)=Cbo
 U(2)=Tro
 U(3)=0.0

@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.

5.0 Analysis of Partial Differential Equations



- Start **Athena Visual Studio**
- From the **File** menu select **New.**
- You are in the **Process Modeling** tab.
- Click **Modeling with PDEs with Diagonal E Matrix.**
- 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.**

Partial Differential Equations 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 cases of higher dimensional problems.

Examples can be found in chemical reactor engineering, various 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. Partial Differential Equations models take the form:

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

Initial Conditions **Boundary Conditions**

$$\mathbf{u}\left(t_0, x\right) = \mathbf{u}_0\left(x\right) \qquad \mathbf{E}\frac{\partial \mathbf{u}}{\partial t} = \mathbf{F}\left(t, \alpha, \mathbf{u}_x; \boldsymbol{\theta}\right) \qquad \mathbf{E}\frac{\partial \mathbf{u}}{\partial t} = \mathbf{F}\left(t, \beta, \mathbf{u}_x; \boldsymbol{\theta}\right)$$

where **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 **u( )** with respect to the dimension *x*, and $\mathbf{u}_{xx}$**( )** represents the second order derivative of the state vector **u( )** with respect to *x* (in the appropriate coordinate system). Initial-Boundary value models are ordinarily used to model unsteady-state reaction and diffusion as well as steady-state fixed bed reactors with significant gradients in the radial direction. They can be solved using **PDAPLUS,** that combines a powerful modified Newton algorithm with a fixed leading coefficient backward difference formula for the approximation of the first order time derivative and various discretization schemes (such as Finite Differences, Global Orthogonal Collocation and Collocation on Finite Elements) for the spatial derivatives.

5.1 Diffusion in Nylon 12 Food Packaging

This example tutorial has been created to test the functionality of AthenaVisual Plus in dealing with the solution of partial differential equations. An additional feature, the simultaneous solution of an ordinary differential equation, which is valid only on the right boundary is also demonstrated.

The transport of a migrant from polymer to food is described by the following set of partial and ordinary differential equations:

$$\frac{\partial C_P}{\partial t} = D_P \frac{\partial^2 C_P}{\partial x^2}$$

$$\frac{\partial C_F}{\partial t} = -D_p \alpha_\upsilon \left.\frac{\partial C_P}{\partial x}\right|_{x=L_P} \qquad \alpha_\upsilon = \frac{A}{V_F}$$

$$t = 0 \quad C_P = \rho C_{Po} \quad C_F = 0$$

$$x = 0 \quad \frac{\partial C_P}{\partial x} = 0$$

$$x = L_P \quad C_P = K_{PF} C_F$$

We wish to solve for the concentration of the migrant in the food C_F as function of time. Also we wish to plot the migrant concentration in the polymer C_P as a function of time at different spatial locations.

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

Model Parameters and Physical Properties	**Description and Units**
$D_P = 3.0\times10^{-14}\ \mathtt{\frac{m^2}{s}}$	Diffusion coefficient in polymer
$K_{PF} = 1.4$	Partitioning coefficient
$L_P = 10^{-4}\ \mathtt{m}$	Polymer thickness
$V_F = 10^{-4}\ \mathtt{m^3}$	Solution volume
$A = 2.0\times10^{-2}\ \mathtt{m^2}$	Contact area
$\rho_P = 1000.0\frac{\mathtt{kg}}{\mathtt{m^3}}$	Polymer density
$C_{P0} = 2300.0\frac{\mathtt{mg}}{\mathtt{kg}}$	Migrant initial concentration

Let’s define the following variables:

$$\xi = \frac{x}{L_P} \quad D_V = \frac{D_P}{L_P^2} \quad D_S = \frac{D_P A}{L_P V_F} \quad u_1 = \frac{C_P}{\rho C_{P0}} \quad u_2 = \frac{C_F}{\rho C_{P0}}$$

Then the model equations can be written as:

$$\frac{\partial u_1}{\partial t} = D_V \frac{\partial^2 u_1}{\partial \xi^2} \quad \frac{\partial u_2}{\partial t} = -D_S \left.\frac{\partial u_1}{\partial \xi}\right|_{\xi=1}$$

$$t = 0 \quad u_1 = 1 \qquad u_2 = 0$$

$$\xi = 0 \quad \frac{\partial u_1}{\partial \xi} = 0$$

$$\xi = 1 \quad u_1 = K_{PF} u_2$$

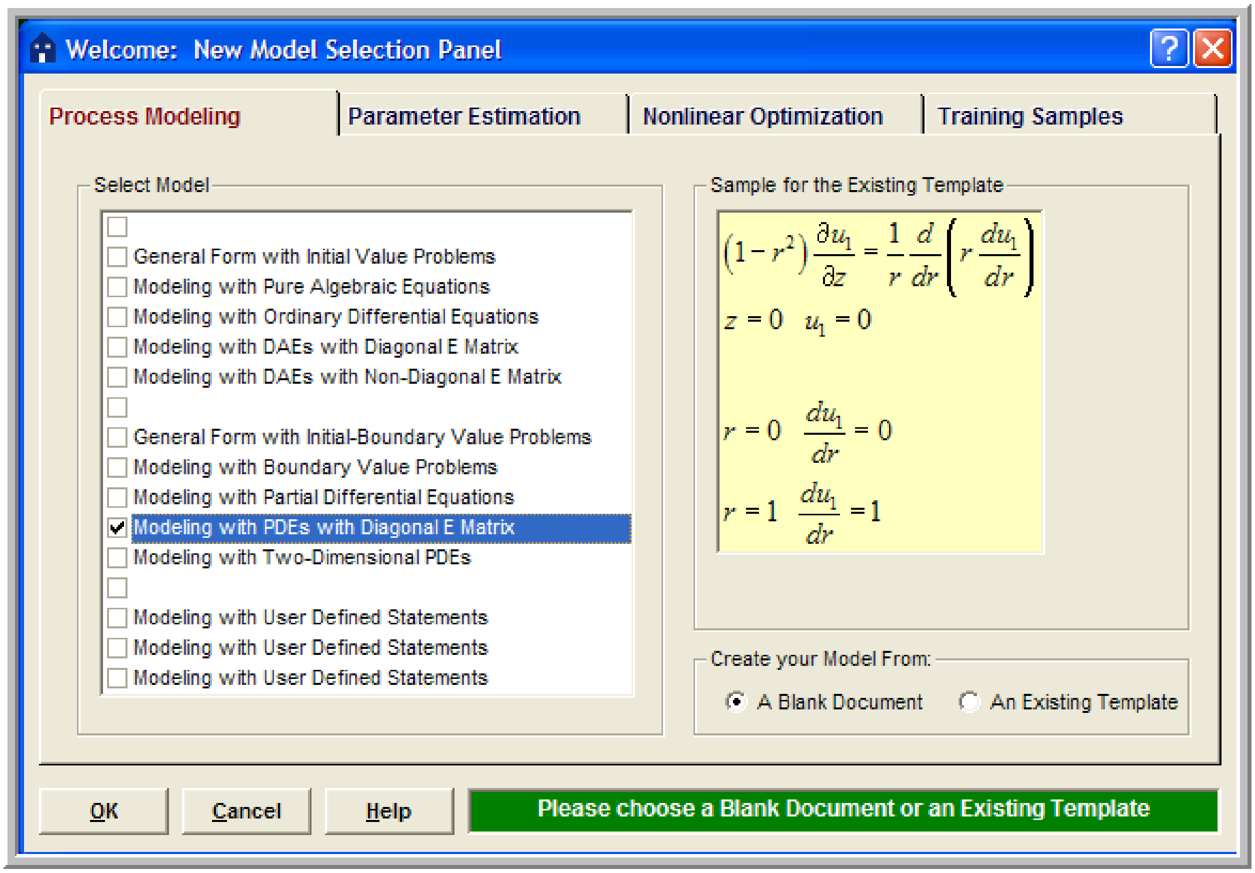
This example tutorial is already precoded in AthenaVisual Plus. If you do not wish to type the code on your own you may access it by doing the following:

- ❖ Open **AthenaVisual Plus**
- ❖ From the **File** menu click **New**
- ❖ Select the **Training Samples** tab
- ❖ Select the **Dynamic Diffusion with Equilibrium Adsorption** sample
- ❖ Click **OK**

5.2 Implementation in AthenaVisual Plus

The following step by step process describes the model implementation in AthenaVisual Plus

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



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

$$\left(1-r^2\right)\frac{\partial u_1}{\partial z}=\frac{1}{r}\frac{d}{dr}\left(r\frac{du_1}{dr}\right)$$

$$z=0 \quad u_1=0$$

$$r=0 \quad \frac{du_1}{dr}=0$$

$$r=1 \quad \frac{du_1}{dr}=1$$

Create your Model From:

● A Blank Document ○ An Existing Template

Please choose a Blank Document or an Existing Template

- ❖ Select the *Process Modeling* tab
- ❖ Select the *Modeling with PDEs with Diagonal E Matrix* 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.

5.2.1 Writing Source Code for Partial Differential Equations

You must enter a minimum of four sections in order to create the partial differential equations model with diagonal **E**( ) matrix. 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. The fourth section labeled **@Coefficient Matrix** is used to enter the diagonal elements of the **E**( ) matrix. A data section not labeled by AthenaVisual Plus may also be used to enter all the data pertinent to the model. The data section may also contain declaration statements for all model variables, parameters and constants. This section, if used, must be the first one in the model. The declaration of the model variables, parameters and constants must be done in accordance the AthenaVisual Plus syntax rules shown below:

5.2.1.1 Data Section

In the data section (which may not be needed and is not labeled by AthenaVisual Plus) the user simply enters the problem data and various constants as shown below. In this example the user enters values for the diffusion coefficient in the polymer, the thickness of the polymer, the contact area and the solvent volume, as well as the partition coefficient and the initial concentration and polymer density.. 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. Character and Logical variables are also allowed. The following source code may be entered for this example (If you use the Windows Copy and Paste commands to enter this code into your AthenaVisual Plus project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```
! Declarations and Model Constants
!--------------------------------
 Global Kpf, Dp, Vf, Ac, Rhop, Cpo, Lp As Real
 Global Dv,Ds As Real

 Kpf=1.4                   ! Partition coefficient
 Dp=3.0D-14                ! Diffusion coefficient in polymer, m2/s
 Lp=1.0D-4                 ! Polymer film thickness, m
 Ac=2E-2                   ! Total contact area, m2
 Vf=100E-6                 ! Volume solvent, m3
 Rhop=1000.0               ! Density of polymer, kg/m3
 Cpo=2300.0                ! Initial concentration in mg/kg

 Dv=Dp/Lp^2*3600.0         ! 1/hr
 Ds=Dv*Ac*Lp/Vf            ! 1/hr
```

5.2.1.2 Declaration of Variables in AthenaVisual Plus

Global Variables: To declare global variables in the AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 differential model for this example in the AthenaVisual Plus environment. The modeling code is NOT case sensitive.

5.2.1.3 Initial Conditions Section

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 integration. 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 in the polymer and **U(2)** to represent the dimensionless concentration in the food. 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)=1.0                 ! Dimensionless concentration in polymer
 U(2)=0.0                 ! Dimensionless initial concentration in solvent
```

5.2.1.4 Model Equations Section

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 concentration of the migrant in the polymer, while **F(2)** is used to represent the material balance equation for the concentration of the migrant in the solvent. 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
 F(1) = Dv * Uxx(1)
 F(2) = 0.0
```

5.2.1.5 Boundary Conditions Section

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. 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) = Ux(1)
  F(2) = 0.0
 Else
  F(1) = U(1) - Kpf * U(2)
  F(2) = -Ds * Ux(1)
 EndIf
```

5.2.1.6 Coefficient Matrix Section

Finally in the Coefficient Matrix section the user must enter the diagonal elements of the **E**( ) matrix on the left and right boundary as well as in the interior of the space domain. The vector **E( )** is reserved in the Athena environment to represent the values of these elements. The variable **LEFT** is reserved in Athena to indicate the left boundary location, the variable **RIGHT** is reserved to indicate the right boundary location and the variable **INTERIOR** is reserved to indicate the interior of the space domain. To enter the heading for the Coefficient Matrix Section:

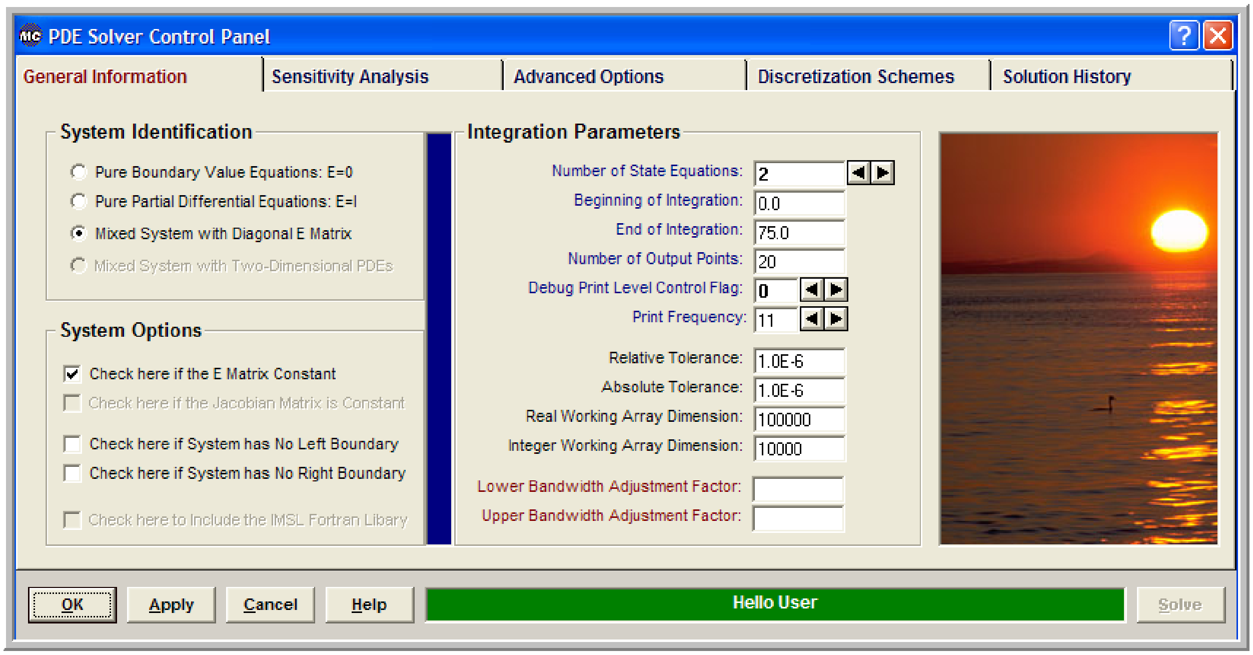
- From the ***Model*** menu choose ***Coefficient Matrix*** (or **Hit F11**)
- Enter the source code as shown below for our example.

```
@Coefficient Matrix
 E(2) = 1.0
 If (INTERIOR) Then
  E(1) = 1.0
 EndIf
```

5.2.2 Entering the Information about the PDAPLUS Solver

It is now time to access the AthenaVisual Plus solver for Partial Differential Equations 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**
- Enter the solver parameters as shown below for our example



System Identification

○ Pure Boundary Value Equations: E=0

○ Pure Partial Differential Equations: E=I

● Mixed System with Diagonal E Matrix

○ Mixed System with Two-Dimensional PDEs

System Options

☑ Check here if the E Matrix Constant

□ Check here if the Jacobian Matrix is Constant

□ Check here if System has No Left Boundary

□ Check here if System has No Right Boundary

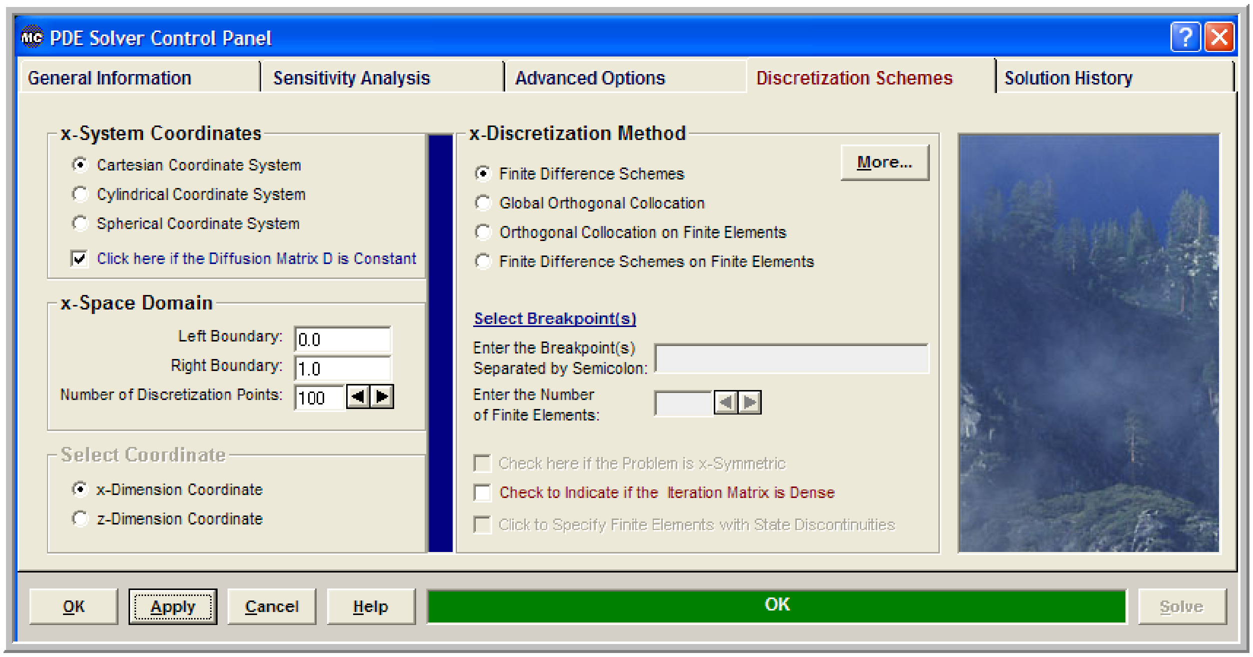
□ Check here to Include the IMSL Fortran Libary

Integration Parameters

Number of State Equations:	2
Beginning of Integration:	0.0
End of Integration:	75.0
Number of Output Points:	20
Debug Print Level Control Flag:	0
Print Frequency:	11
Relative Tolerance:	1.0E-6
Absolute Tolerance:	1.0E-6
Real Working Array Dimension:	100000
Integer Working Array Dimension:	10000
Lower Bandwidth Adjustment Factor:	
Upper Bandwidth Adjustment Factor:	

The **PDE Solver Control Panel** window appears. In the **System Identification** group you will see that the option *Mixed System with Diagonal E Matrix* has already been selected for you. From the **Integration Parameters** group enter the Number of State Equations, the Beginning of Integration, the End of Integration and the Number of Output Points(control the granularity of graphs). You may also 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



PDE Solver Control Panel

x-System Coordinates

- ● Cartesian Coordinate System
- ○ Cylindrical Coordinate System
- ○ Spherical Coordinate System
- ☑ Click here if the Diffusion Matrix D is Constant

x-Space Domain

- Left Boundary: 0.0
- Right Boundary: 1.0
- Number of Discretization Points: 100

Select Coordinate

- ● x-Dimension Coordinate
- ○ z-Dimension Coordinate

x-Discretization Method

- ● Finite Difference Schemes
- ○ Global Orthogonal Collocation
- ○ Orthogonal Collocation on Finite Elements
- ○ Finite Difference Schemes on Finite Elements

Select Breakpoint(s)

- Enter the Breakpoint(s) Separated by Semicolon:
- Enter the Number of Finite Elements:
- □ Check here if the Problem is x-Symmetric
- □ Check to Indicate if the Iteration Matrix is Dense
- □ Click to Specify Finite Elements with State Discontinuities

OK

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); you may also 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.**

5.2.3 Saving and Running your Model

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:

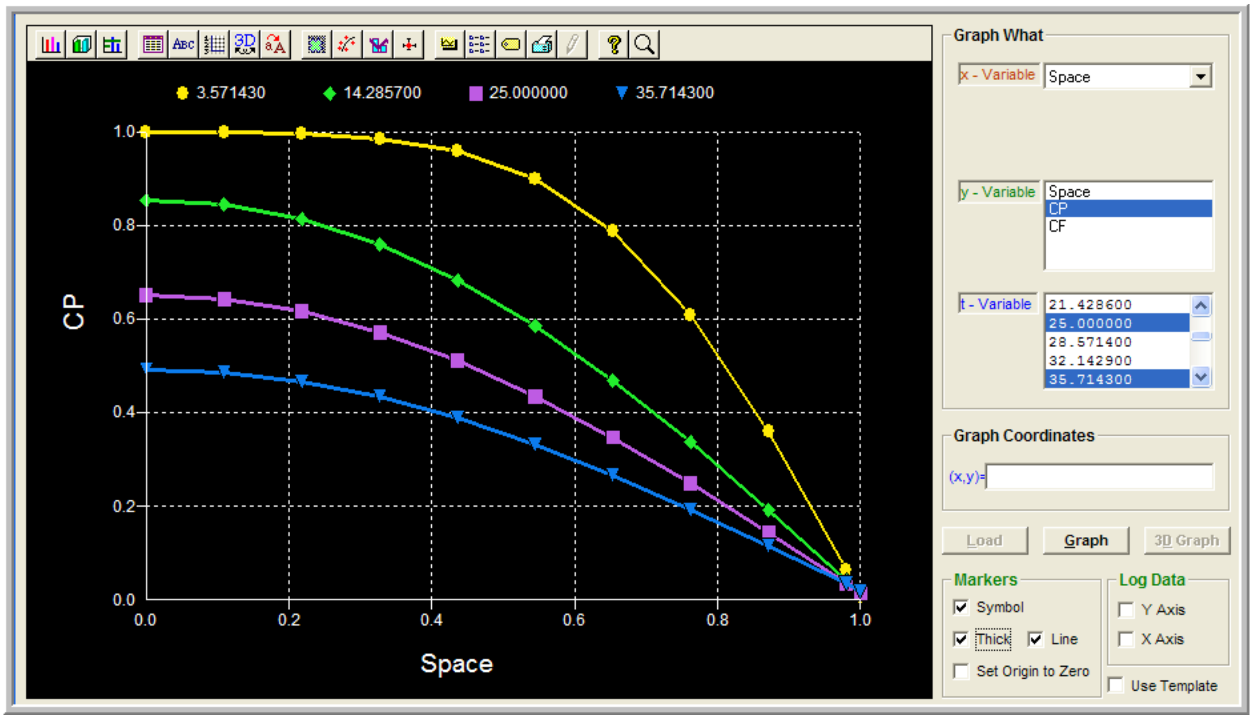
- From the ***File*** menu, choose ***Save***. The *Save As* dialog box appears. This action saves your model and creates the Fortran code.
- In the Directories box, double-click a directory where you want to store the source file of your project.
- Type a filename (a filename cannot contain the following characters: \ / : * ? " < > |) in the File Name box, then choose **OK**. The default extension is **avw**
- To view the Fortran code that you 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**)

5.2.4 Numerical and Graphical 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. If you wish to see the time or space profiles for the concentration in the polymer and solution phases, from the **View** menu choose **Solution Graphs**, or click The graphics control panel appears:



Graph What

x - Variable: Space

y - Variable: Space, CP (selected), CF

t - Variable: 21.428600, 25.000000 (selected), 28.571400, 32.142900, 35.714300 (selected)

Graph Coordinates

(x,y)=

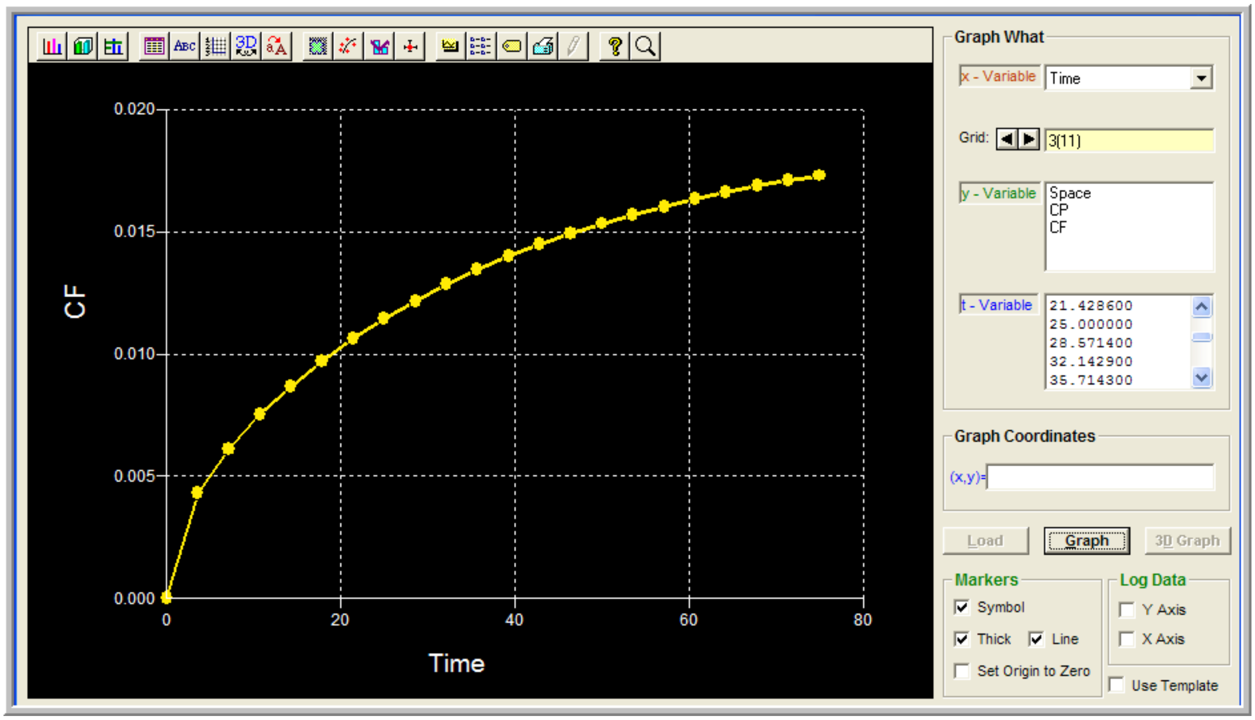
Markers: ☑ Symbol ☑ Thick ☑ Line □ Set Origin to Zero

Log Data: □ Y Axis □ X Axis

□ Use Template

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 polymer concentration **CP**) and in the **t-variable** select a number of times at which you wish to plot the concentration; 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.

Suppose you wish to see the concentration of the migrant in the solution as a function of time. To do that in the **x-variable** select **Time** from the drop down list.



Graph What

x - Variable: Time

Grid: ◀ ▶ 3(11)

y - Variable:
Space
CP
CF

t - Variable:
21.428600
25.000000
28.571400
32.142900
35.714300

Graph Coordinates

(x,y)=

Markers

☑ Symbol

☑ Thick ☑ Line

□ Set Origin to Zero

Log Data

□ Y Axis

□ X Axis

□ Use Template

In the **y-variable** double-click on the **CF** (migrant concentration in solution) variable and then press the **Grid** spin control until you reach the discretization point at the right boundary (in this case point number 12). Notice that this discretization point is not equal to 102 because of the selected *Printing Frequency* input number. Then click **Graph**; you should see the graph above that shows the dimensionless concentration of the migrant as a function of time.

5.2.5 Explicit Auxiliary Variables

Auxiliary variables can be implemented in an explicit way, without having to increase the dimensionality of the original system of equations. In order to do that we have to make use of two **AthenaVisual Plus** options; the first allows us to enter custom code right after a call to the integrator, and the second allows us to append auxiliary variables to the state vector for printing purposes.

As an example, consider the introduction of the auxiliary variable *Flux* that represents the flux of the migrant at the particle surface; this auxiliary variable is therefore defined by the following equation:

$$Flux = -D_s \frac{\partial u_1}{\partial x} = -D_s \frac{u_{1,Ngrid} - u_{1,Ngrid-1}}{x_{Ngrid} - x_{Ngrid-1}}$$

The code below illustrates how to do the implementation:

```
@Initial Conditions
 U(1)=1.0                 ! Dimensionless concentration in polymer
 U(2)=0.0                 ! Dimensionless initial concentration in solvent

@Model Equations
 F(1) = Dv * Uxx(1)
 F(2) = 0.0

@Boundary Conditions
 If (LEFT) Then
  F(1) = Ux(1)
  F(2) = 0.0
 Else
  F(1) = U(1) - Kpf * U(2)
  F(2) = -Ds * Ux(1)
 End If

@Coefficient Matrix
 E(2) = 1.0
 If (INTERIOR) Then
  E(1) = 1.0
 EndIf

@After Calling Solver
 Global Flux(Mgrid) As Real
 Flux=-Ds*(U(1,Ngrid)-U(1,Ngrid-1))/(xGrid(Ngrid)-xGrid(Ngrid-1))

@Solver Options
 Tend=75.0
 Npts=1
 Append=Flux
```

Notice above in the Solver Options section, the **Tend** keyword specifies the end of the integration of the system of differential equations while the **Npts** keyword specifies the number of output points for printing purposes. The **Append** keyword instructs the solver to append the auxiliary variable to the state vector so that they can be printed and therefore used in the graphics server. To view all Solver Options form the **View** menu in Athena select **Solver Options**.

Now from the ***Build*** menu select ***Execute*** (or **Hit F5**). You should see the following results:

```
Number of State Equations.......................     2
Number of Sensitivity Parameters................     0
Number of Discretization Points.................   102
Number of Integration Output Points.............     3

 TIME          SPACE        U(1)         U(2)         Flux
 0.00000E+00   0.00000E+00  1.00000E+00  0.00000E+00  2.18160E-02
               1.08911E-01  1.00000E+00  0.00000E+00  2.18160E-02
               2.17822E-01  1.00000E+00  0.00000E+00  2.18160E-02
               3.26733E-01  1.00000E+00  0.00000E+00  2.18160E-02
               4.35644E-01  1.00000E+00  0.00000E+00  2.18160E-02
               5.44554E-01  1.00000E+00  0.00000E+00  2.18160E-02
               6.53465E-01  1.00000E+00  0.00000E+00  2.18160E-02
               7.62376E-01  1.00000E+00  0.00000E+00  2.18160E-02
               8.71287E-01  1.00000E+00  0.00000E+00  2.18160E-02
               9.80198E-01  1.00000E+00  0.00000E+00  2.18160E-02
               1.00000E+00  0.00000E+00  0.00000E+00  2.18160E-02
 3.75000E+01   0.00000E+00  4.70675E-01  0.00000E+00  1.53043E-04
               1.08911E-01  4.64598E-01  0.00000E+00  1.53043E-04
               2.17822E-01  4.45338E-01  0.00000E+00  1.53043E-04
               3.26733E-01  4.13471E-01  0.00000E+00  1.53043E-04
               4.35644E-01  3.69953E-01  0.00000E+00  1.53043E-04
               5.44554E-01  3.16090E-01  0.00000E+00  1.53043E-04
               6.53465E-01  2.53502E-01  0.00000E+00  1.53043E-04
               7.62376E-01  1.84075E-01  0.00000E+00  1.53043E-04
               8.71287E-01  1.09901E-01  0.00000E+00  1.53043E-04
               9.80198E-01  3.32213E-02  0.00000E+00  1.53043E-04
               1.00000E+00  1.91909E-02  1.37078E-02  1.53043E-04
 7.50000E+01   0.00000E+00  1.85093E-01  0.00000E+00  5.45159E-05
               1.08911E-01  1.82926E-01  0.00000E+00  5.45159E-05
               2.17822E-01  1.76057E-01  0.00000E+00  5.45159E-05
               3.26733E-01  1.64694E-01  0.00000E+00  5.45159E-05
               4.35644E-01  1.49179E-01  0.00000E+00  5.45159E-05
               5.44554E-01  1.29980E-01  0.00000E+00  5.45159E-05
               6.53465E-01  1.07676E-01  0.00000E+00  5.45159E-05
               7.62376E-01  8.29386E-02  0.00000E+00  5.45159E-05
               8.71287E-01  5.65142E-02  0.00000E+00  5.45159E-05
               9.80198E-01  2.91992E-02  0.00000E+00  5.45159E-05
               1.00000E+00  2.42014E-02  1.72867E-02  5.45159E-05

EXIT PDAPLUS: SOLUTION FOUND

Number of Steps Taken Thus Far...................      246
Number of Function Evaluations...................      736
Number of Jacobian Evaluations...................       32
Number of Jacobian Factorizations................       32
```

From the above table we clearly see how the **AthenaVisual Plus** solver augmented the auxiliary variable to the state vector for printing purposes. One can now access the Graphics control panel to graph the auxiliary variables along with the state variables.

6.0 Parameter Estimation from Single-Response Data

The development of a process model typically goes through several stages, considered further in this section. These stages include model formulation, collection of data from designed experiments or from existing sources, model testing and discrimination, and extensions of the database with sequentially designed experiments. Computational aids can expedite several of these tasks, but a prudent investigator will use data plots, physical and chemical clues whenever possible in formulating candidate models.

The statistical investigation of a model begins with the estimation of its parameters from observations. For single-response observations with independent Normal error distributions and given relative variances, Bayes' theorem leads to the famous Method of Least Squares.

Least squares was introduced by Legendre in 1085 as a curve-fitting method, and by Gauss in 1809 as a Bayesian procedure for estimating parameters from data with independent Normal distributions of error. A fascinating account of these discoveries and related events is given by Stigler (1986). Gauss included models nonlinear in the parameters, and weighted the observations according to their precision. He gave an efficient solution scheme for the *normal equations* of the least-squares problem and showed how to calculate the variances of the resulting parameter estimates. Many later workers have built on this foundation, refining the posterior density function and adding interval estimates, hypothesis tests, model discrimination methods, and efficient procedures for design of experiments.

7.1 The Parameter Estimation Problem

The first task in this section is the estimation of the parameter vector $\boldsymbol{\theta} = \left\{\theta_1, \theta_2, \ldots \theta_p\right\}$ in a mathematical model of the form

$$y_u = f_u\left(\mathbf{x}_u, \boldsymbol{\theta}\right) + \varepsilon_u = f_u\left(\boldsymbol{\theta}\right) + \varepsilon_u \qquad u = 1, \ldots, n \tag{1}$$

for a data vector $\mathbf{y} := \left\{y_1, \ldots, y_n\right\}^T$ of single-response observations taken at experimental settings $\left\{x_u^1, x_u^2 \ldots, x_u^k\right\}$. Each observation y_u is modeled as an expectation function $f_u\left(\mathbf{x}_u, \boldsymbol{\theta}\right)$ [postulated to be true] plus an independent random error ε_u with probability density

$$p\left(\varepsilon_u \middle| \sigma_u\right) = \frac{1}{\sqrt{2\pi\sigma_u}} \exp\left[-\frac{\varepsilon_u^2}{2\sigma_u^2}\right] \quad u = 1,\ldots,n \tag{2}$$

based on the Central Limit Theorem, and on the assumption that the data have been corrected for any systematic errors. Hence, the predictive probability density for any observation y_u is:

$$p\left(y_u \middle| \boldsymbol{\theta}, \sigma_u\right) = \frac{1}{\sqrt{2\pi\sigma_u}} \exp\left\{-\frac{\left[y_u - f_u\left(\boldsymbol{\theta}\right)\right]^2}{2\sigma_u^2}\right\} \quad u = 1,\ldots,n \tag{3}$$

This density times $d\varepsilon$ is the probability that a replicate observation at $\mathbf{x}_u$ would fall within $\pm d\varepsilon/2$ of the given y_u. The probability density for the full data vector $\mathbf{y}$ is the product of these independent functions:

$$p\left(\mathbf{y} \middle| \boldsymbol{\theta}, \sigma_1,\ldots,\sigma_n\right) = \left[\prod_{u=1}^{n} \frac{1}{\sqrt{2\pi\sigma_u}}\right] \exp\left\{-\sum_{u=1}^{n} \frac{\left[y_u - f_u\left(\boldsymbol{\theta}\right)\right]^2}{2\sigma_u^2}\right\} \tag{4}$$

To describe the expected relative precisions of observations at the various conditions $\mathbf{x}_u$ it is customary to assign a vector $\boldsymbol{\omega}$ of numerical weights

$$\omega_u = \sigma^2 / \sigma_u^2 \quad u = 1,\ldots,n \tag{5}$$

in which σ^2 is the variance of observations of unit weight. The weights may be assigned on the basis of a precision analysis, or more realistically via replicate observations. Equation (4) then takes the form:

$$p\left(\mathbf{y} \middle| \boldsymbol{\theta}, \sigma, \boldsymbol{\omega}\right) = \left[\prod_{u=1}^{n} \frac{\sqrt{\omega_u}}{\sqrt{2\pi\sigma}}\right] \exp\left[-\frac{S\left(\boldsymbol{\theta}\right)}{2\sigma^2}\right] \tag{6}$$

in which

$$\begin{aligned} S\left(\boldsymbol{\theta}\right) &= \sum_{u=1}^{n} \left[\sqrt{\omega_u}\, y_u - \sqrt{\omega_u}\, f_u\left(\boldsymbol{\theta}\right)\right]^2 \\ &= \sum_{u=1}^{n} \left[Y_u - F_u\left(\boldsymbol{\theta}\right)\right]^2 = \sum_{u=1}^{n} E_u\left(\boldsymbol{\theta}\right)^2 \end{aligned} \tag{7}$$

Here the notations

$$Y_u = \sqrt{\omega_u}\, y_u; \quad F_u(\boldsymbol{\theta}) = \sqrt{\omega_u}\, f_u(\boldsymbol{\theta}); \quad E_u(\theta) = Y_u - F_u(\boldsymbol{\theta}) \tag{8}$$

have reduced $S(\boldsymbol{\theta})$ to a simple sum of squares of the weighted errors $E_u(\boldsymbol{\theta})$. Correspondingly, the predictive probability density for a weighted observation $\boldsymbol{Y_u}$ is

$$p\left(Y_u \middle| \theta,\sigma\right) = \left[\frac{p\left(y_u \middle| \boldsymbol{\theta},\sigma\right)}{\sqrt{\omega_u}}\right] = \left(\sqrt{2\pi\sigma}\right)^{-1} \exp\left[-\frac{E_u^2(\boldsymbol{\theta})}{2\sigma^2}\right] \quad u = 1,\ldots,n \tag{9}$$

and that for a prospective vector $Y_1 \ldots, Y_n$ is

$$p\left(\mathbf{Y} \middle| \boldsymbol{\theta},\sigma\right) = \left(\sqrt{2\pi\sigma}\right)^{-n} \exp\left[-\frac{S(\boldsymbol{\theta})}{2\sigma^2}\right]. \tag{10}$$

When **Y** is given instead of $\boldsymbol{\theta}$ and σ we will call this function the *likelihood*, $\ell\left(\boldsymbol{\theta},\sigma \middle| \mathbf{Y}\right)$.

To apply Bayes' theorem, we need a prior density for the unknowns $\boldsymbol{\theta}$ and σ. Treating $\boldsymbol{\theta}$ and σ as independent *a priori*, and $p(\boldsymbol{\theta})$ as uniform over the permitted range of $\boldsymbol{\theta}$, we obtain the joint prior density

$$p(\boldsymbol{\theta},\sigma) \propto \begin{cases} \sigma^{-1} & \text{for permitted values of } \boldsymbol{\theta} \\ 0 & \text{for forbidden values of } \boldsymbol{\theta} \end{cases} \tag{11}$$

consistent with the Jeffries prior. Multiplication of $\ell\left(\boldsymbol{\theta},\sigma \middle| \mathbf{Y}\right)$ by $p(\boldsymbol{\theta},\sigma)$ in accordance with Bayes' Theorem then gives the posterior density for permitted values of $\boldsymbol{\theta}$ as

$$p\left(\boldsymbol{\theta},\sigma \middle| \mathbf{Y}\right) \propto \sigma^{-(n+1)} \exp\left[-\frac{S(\boldsymbol{\theta})}{2\sigma^2}\right]. \tag{12}$$

This probability density takes its maximum at the minimum sum of squares $S(\boldsymbol{\theta})$ encountered in the permitted range of $\boldsymbol{\theta}$. In practice, we need not only this least-squares value (the *point estimate*), but also various *interval estimates*, regions of *highest posterior density*, and integrals over part or all of the range of $\boldsymbol{\theta}$..

Another rationale for minimizing $S(\boldsymbol{\theta})$ is to maximize the likelihood function $\ell(\boldsymbol{\theta},\sigma|\mathbf{Y})$ obtained from Equation (10). This *maximum likelihood* approach, which was advocated by Fisher (1925), gives the same point estimate of $\boldsymbol{\theta}$ as does the posterior density function.

The permitted region of $\boldsymbol{\theta}$ can take various forms. We prefer to use a continuous rectangular region given by the inequalities:

$$\ell_i \le \theta_i \le u_i \quad i = 1,\ldots,p \tag{13}$$

for a model containing *p* adjustable parameters.

Many methods are available for least-squares calculations. Models linear in $\boldsymbol{\theta}$ allow direct solutions; other models require iteration. The choice of method depends on what is wanted. For a simple curve-fit of data with a nonlinear model, a direct search procedure such as that of Powell or of Nelder and Mead [or of several current software packages] may suffice. But to make statistical inferences, a method based on local expansions of $S(\boldsymbol{\theta})$ is essential. The Gauss-Newton method is such a method and a variant of it has been implemented in AthenaVisual Plus.

7.2 References

M. Caracotsios, *Model Parametric Sensitivity Analysis and Nonlinear Parameter Estimation. Theory and Applications.* Ph. D. thesis, Dept. of Chemical Engineering, University of Wisconsin, Madison (1986).

W. E. Stewart, M. Caracotsios, and J. P. Sorensen, *Parameter estimation from multiresponse data*, AIChE J., 38, 641-650 (1992); Errata, 38, 1302 (1992).

W. E. Stewart, and M. Caracotsios, *Computer-Aided Modeling of Reactive Systems*, John Wiley and Sons, Inc. (2008).

7.3 Parameter Estimation with an Explicit Model

- Start **Athena Visual Studio**
- From the **File** menu select **New.**
- You are in the **Process Modeling** tab.
- Select the **Parameter Estimation** tab.
- Click **Estimation with User Defined Explicit Models.**
- 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.**

Nonlinear Parameter estimation problems with user defined explicit models must be self-contained. Specifically they should not rely on the solvers that are in AthenaVisual Plus. The user however, may make calls to the available math and engineering subroutine as well as user defined procedures. This test problem has been design to illustrate the functionality of AthenaVisual Plus in dealing with single response parameter estimation with user explicit models.

This example problem has been created to test the functionality of AthenaVisual Plus in dealing with parameter estimation from single-response data with explicit mechanistic models.

7.4 Rate Constants Estimation in CH_3OH Chemistry

The Methanol production chemistry can be represented by the simple chemical reaction:

$$CO + 2H_2 \xrightarrow{\Re} CH_3OH$$

The following model has been postulated as plausible candidate to describe the observed reactor rate of the product methanol based on the available experimental data:

$$\Re_1 = \frac{k(T)K_1K_2^2\left(P_CP_H^2 - P_M / K_{eq}\right)}{\left(1+K_1P_C+K_2P_H+K_3P_M\right)^3}$$

$$k = \exp\left(\ln k_B + \frac{E}{RT_B}\left(1-\frac{T_B}{T}\right)\right)$$

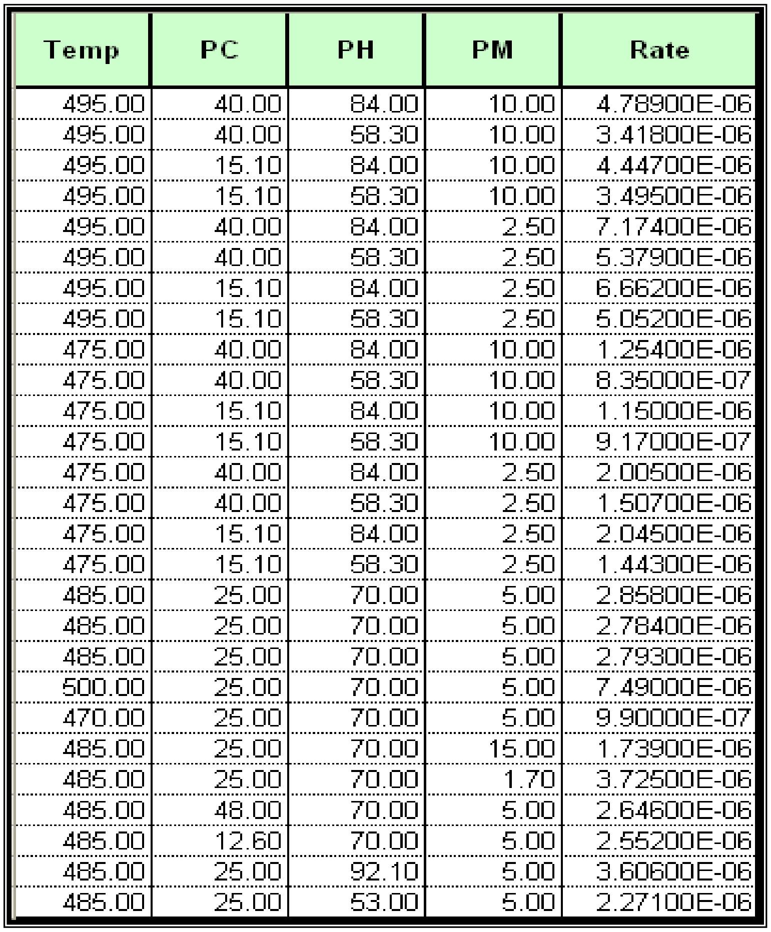
$$\log K_{eq} = \frac{3914}{T} - 7.536\,\log T + 0.001766\log T + 9.388$$

- Estimate the parameters $\{k_B, E, K_1, K_2, K_3\}$
- Make parameter transformations by use of the **Phi** functions
- Investigate the use of partial analytical derivative information
- Conduct an interpolation and an extrapolation study to assess the model accuracy

The parameter initial guess for this example is given in the table below:

Initial Values of Parameters	
$\ln k_B = -5.0$	$K_1 = 0.1$
$\dfrac{E}{RT_B} = 10.0$	$K_2 = 0.1$
$T_B = 485.0$	$K_3 = 0.1$

The experimental data for this example are given below:

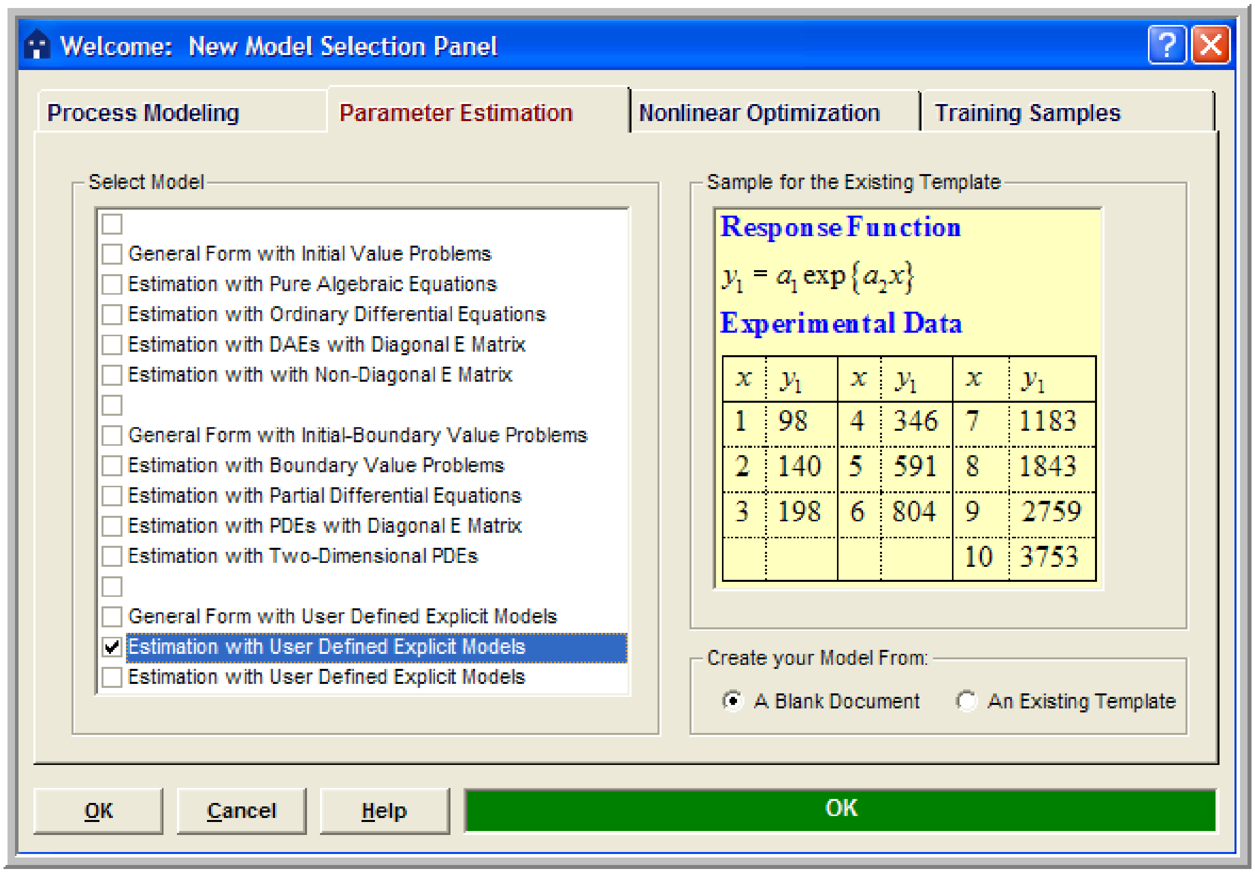


Temp	PC	PH	PM	Rate
495.00	40.00	84.00	10.00	4.78900E-06
495.00	40.00	58.30	10.00	3.41800E-06
495.00	15.10	84.00	10.00	4.44700E-06
495.00	15.10	58.30	10.00	3.49500E-06
495.00	40.00	84.00	2.50	7.17400E-06
495.00	40.00	58.30	2.50	5.37900E-06
495.00	15.10	84.00	2.50	6.66200E-06
495.00	15.10	58.30	2.50	5.05200E-06
475.00	40.00	84.00	10.00	1.25400E-06
475.00	40.00	58.30	10.00	8.35000E-07
475.00	15.10	84.00	10.00	1.15000E-06
475.00	15.10	58.30	10.00	9.17000E-07
475.00	40.00	84.00	2.50	2.00500E-06
475.00	40.00	58.30	2.50	1.50700E-06
475.00	15.10	84.00	2.50	2.04500E-06
475.00	15.10	58.30	2.50	1.44300E-06
485.00	25.00	70.00	5.00	2.85800E-06
485.00	25.00	70.00	5.00	2.78400E-06
485.00	25.00	70.00	5.00	2.79300E-06
500.00	25.00	70.00	5.00	7.49000E-06
470.00	25.00	70.00	5.00	9.90000E-07
485.00	25.00	70.00	15.00	1.73900E-06
485.00	25.00	70.00	1.70	3.72500E-06
485.00	48.00	70.00	5.00	2.64600E-06
485.00	12.60	70.00	5.00	2.55200E-06
485.00	25.00	92.10	5.00	3.60600E-06
485.00	25.00	53.00	5.00	2.27100E-06

7.5 Implementation in AthenaVisual Plus

The following step by step process describes the model implementation in AthenaVisual Plus

- Open **AthenaVisual Plus**. From the ***File*** menu, choose ***New***. The *Welcome: New Model Selection Panel* window appears.



Select Model

□
□ General Form with Initial Value Problems
□ Estimation with Pure Algebraic Equations
□ Estimation with Ordinary Differential Equations
□ Estimation with DAEs with Diagonal E Matrix
□ Estimation with with Non-Diagonal E Matrix
□
□ General Form with Initial-Boundary Value Problems
□ Estimation with Boundary Value Problems
□ Estimation with Partial Differential Equations
□ Estimation with PDEs with Diagonal E Matrix
□ Estimation with Two-Dimensional PDEs
□
□ General Form with User Defined Explicit Models
☑ Estimation with User Defined Explicit Models
□ Estimation with User Defined Explicit Models

Sample for the Existing Template

Response Function

$y_1 = a_1 \exp\{a_2 x\}$

Experimental Data

x	y_1	x	y_1	x	y_1
1	98	4	346	7	1183
2	140	5	591	8	1843
3	198	6	804	9	2759
				10	3753

Create your Model From:

● A Blank Document ○ An Existing Template

- Select the **Parameter Estimation** tab
- Select the *Estimation with User Defined Explicit Models* 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.

7.5.1 Writing Source Code for Estimation with Explicit Models

You must enter a minimum of one section in order to create a parameter estimation model with a user defined (or explicit) stand-alone model. This section labeled **@Response Model** is used to define the experimental responses (observations). The vector **Y( )** is reserved in Athena to define these responses. A section labeled **@Gradient Vector** is optional and may be used to enter the derivatives of the responses with respect to the adjustable parameters. The matrix **dY(i,k)** is reserved in Athena to define these derivatives. A data section not labeled by AthenaVisual Plus may also be used to enter all the data pertinent to the model. The data section may also contain declaration statements for all model variables, parameters and constants. This section, if used, must be the first one in the model. The declaration of the model variables, parameters and constants must be done in accordance the AthenaVisual Plus syntax rules shown below:. In addition you must enter information about the experimental observations and the parameters you are about to estimate.

7.5.1.1 Declaration of Variables in AthenaVisual Plus

Global Variables: To declare global variables in the AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 an implicit model for parameter estimation in the AthenaVisual Plus environment. The modeling code is NOT case sensitive.

7.5.1.2 Data Section

In the data section the user simply enters the problem data and various constants. The data section also contains the declarations of problem variables, parameters and constants. For our example the user enters the experimental base temperature and the universal gas constant as shown in the code below. The Athena interpreter treats any line that begins with an exclamation mark **!** as a comment. It is mandatory and strongly recommended that the users declare all the problem parameters and constants. All variables in Athena are either real double or single precision or integer long. Character and Logical variables are also allowed. The following source code may be entered for this example (If you use the Windows Copy and Paste commands to enter this code into your AthenaVisual Plus project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```
! Declarations and Model Constants
!---------------------------------
 Global Tref,Temp,Rg As Real
 Global PC,PM,PH As Real
 Global K1,K2,K3,KEQ,RK As Real

 Rg=8.314     ! Universal Gas Constant (J/mol k)
 Tref=485.0   ! Reference Temperature (K)
```

7.5.1.3 Response Model

In the Response Model Equations section the user must enter the responses. The vector **Par( )** is reserved to access the adjustable parameters, and the vector **Xu( )** is reserved to access the experimental settings. As mentioned above the vector **Y( )** is reserved to enter the responses. For example **Y(1)** represents the first measured response, **Y(2)** the second (if present) and so on. As mentioned above, **Xu( )** is reserved for the experimental settings. For example **Xu(1)** represents the first setting, **Xu(2)** the second if present and so on. To enter the responses for your model:

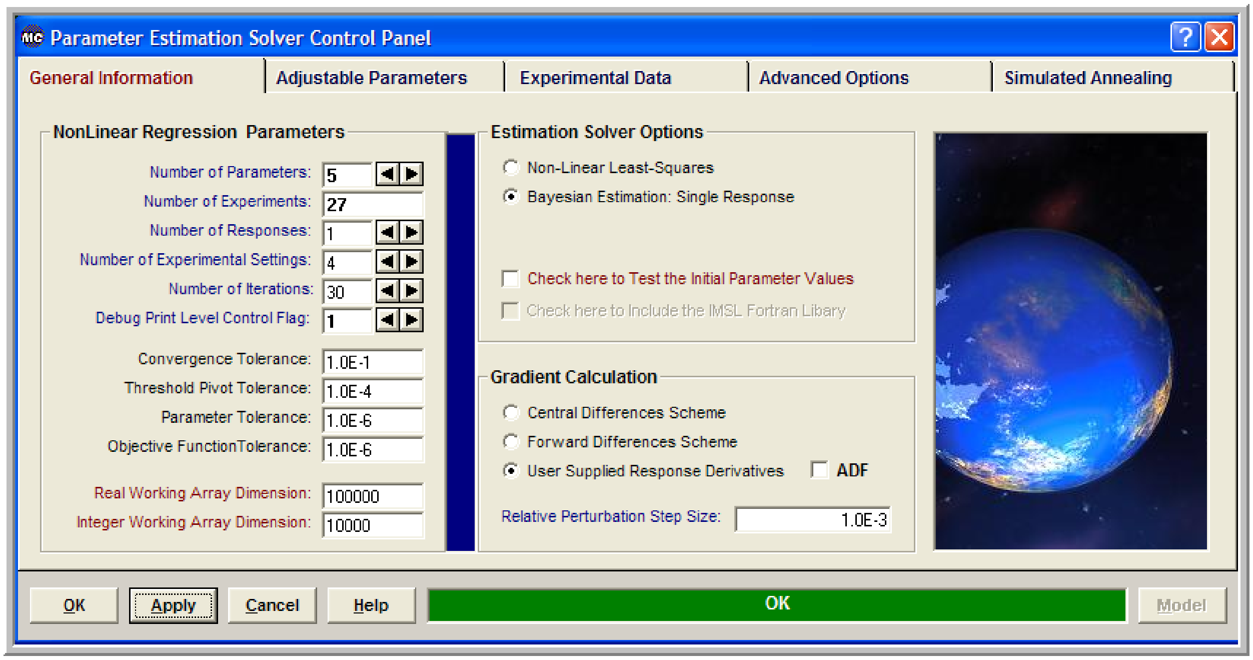
- From the ***Model*** menu choose ***Response Model*** (or **Hit F11)**
- Enter the source code as shown below for our example.

```
@Response Model
 Temp=Xu(1)
 PC=Xu(2)
 PH=Xu(3)
 PM=Xu(4)

 RK=exp(Par(1)+Par(2)*(1.0-Tref/Temp))
 K1=Par(3)
 K2=Par(4)
 K3=Par(5)
 KEQ=10^(3914.0/Temp-7.536*log10(Temp)+0.001766*Temp+9.388)

 Y(1)=RK*(PC*PH**2-PM/KEQ)*K1*K2**2/(1.0+K1*PC+K2*PH+K3*PM)**3
```

7.5.1.4 Gradient Vector

The **Gradient Vector** section is optional. The user may decide to enter the derivatives of the response vector with respect to the parameter vector analytically; this may help situations where errors in the numerical evaluation of the response function derivatives may hinder the progress of the solution algorithm. The matrix **dY( )** is reserved in the Athena environment to represent these derivative elements. Thus **dY(1,3)** holds the derivative of the response **Y(1)** with respect to the parameter **Par(3).** To enter the Response Model Derivative section:

❖ From the ***Model*** menu choose ***Load Solver***. The **Parameter Estimation Solver Control Panel** appears.



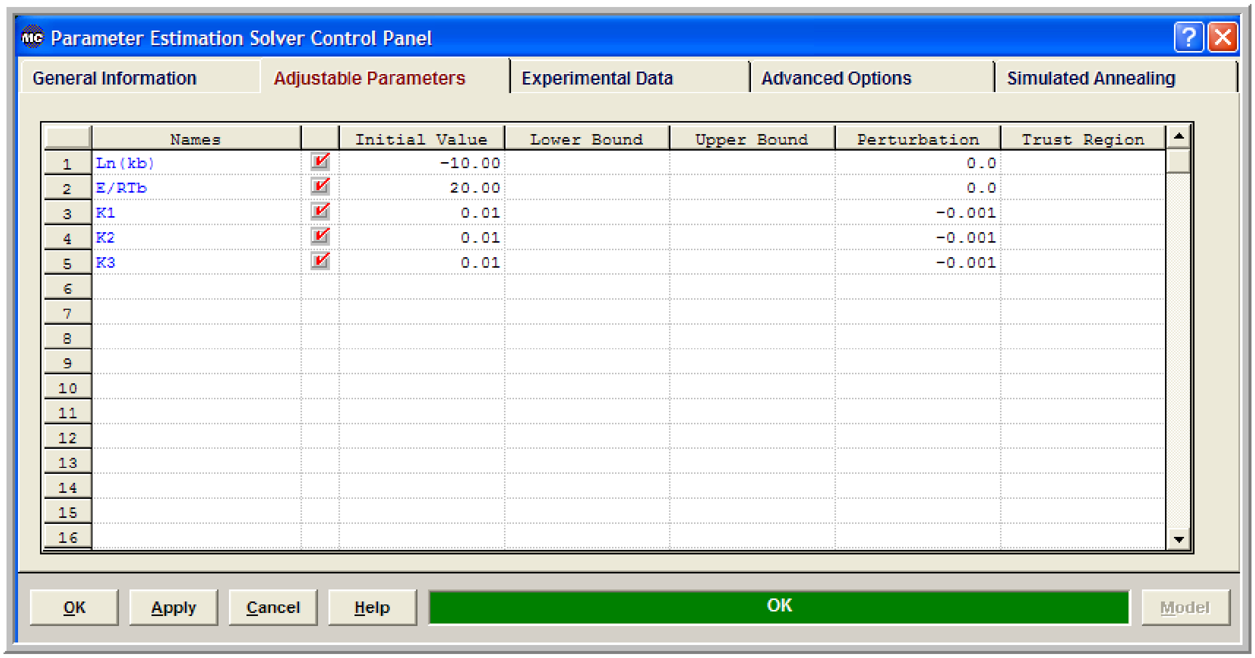
NonLinear Regression Parameters
Number of Parameters: 5
Number of Experiments: 27
Number of Responses: 1
Number of Experimental Settings: 4
Number of Iterations: 30
Debug Print Level Control Flag: 1
Convergence Tolerance: 1.0E-1
Threshold Pivot Tolerance: 1.0E-4
Parameter Tolerance: 1.0E-6
Objective FunctionTolerance: 1.0E-6
Real Working Array Dimension: 100000
Integer Working Array Dimension: 10000

Estimation Solver Options
○ Non-Linear Least-Squares
● Bayesian Estimation: Single Response
□ Check here to Test the Initial Parameter Values
□ Check here to Include the IMSL Fortran Libary

Gradient Calculation
○ Central Differences Scheme
○ Forward Differences Scheme
● User Supplied Response Derivatives □ ADF
Relative Perturbation Step Size: 1.0E-3

❖ From the **Derivatives Calculation** group choose the *User Supplied Model Derivatives* option, then choose **OK**.
❖ From the ***Model*** menu choose ***Gradient Vector*** (or **Hit F11**)
❖ Enter the source code for the derivatives of the responses with respect to the parameters

Alternatively if you wish to enter selective derivatives, in the **Parameter Estimation Solver Control Panel** choose the **Adjustable Parameters** tab and enter zero for *Perturbation* for the parameters you will be supplying analytical derivatives.



	Names		Initial Value	Lower Bound	Upper Bound	Perturbation	Trust Region
1	Ln(kb)	☑	-10.00			0.0	
2	E/RTb	☑	20.00			0.0	
3	K1	☑	0.01			-0.001	
4	K2	☑	0.01			-0.001	
5	K3	☑	0.01			-0.001	
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

Then choose **OK** and enter the source code for the selected derivatives as shown below:

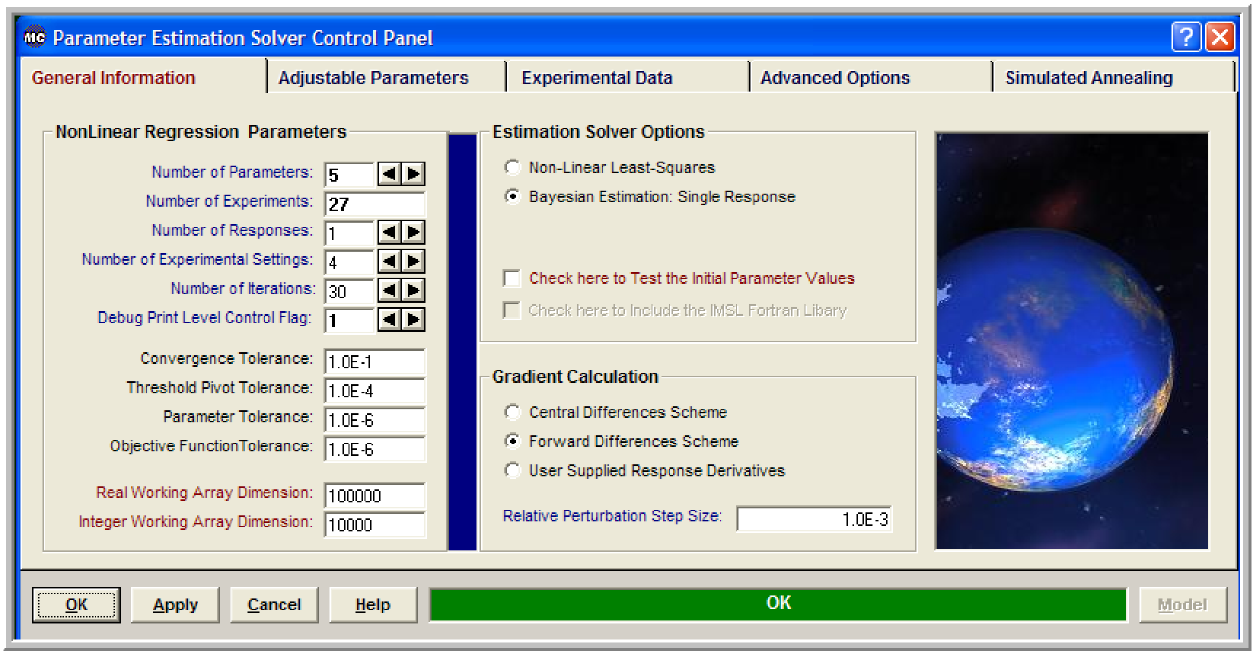
```
@Gradient Vector
 dY(1,1)=Y(1)
 dY(1,2)=Y(1)*(1.0-Tref/Temp)
```

This option is useful, when the model contains sets of parameters with respect to which the derivatives are very simple to derive and implement. For the parameters with respect to which the derivative derivation is complex, the user leaves the task to the parameter estimation solver. In this case the derivatives are calculated by finite differences. It is worth pointing out that the estimation solver in Athena contains technology that optimizes the perturbation step size, in order to minimize the truncation and round-off error. This guarantees the best possible estimation of the gradient vector which is crucial in the minimization process.

7.5.2 The Parameter Estimation Solver Control Panel

It is now time to access the AthenaVisual Plus solver for Parameter Estimation in order to enter information about the adjustable parameters, the experimental observations and various other parameters that control the estimation algorithm, To do that:

- From the ***Model*** menu choose ***Load Solver,*** or **Hit F12**
- Enter the solver parameters as shown below for our example



NonLinear Regression Parameters

Number of Parameters: 5
Number of Experiments: 27
Number of Responses: 1
Number of Experimental Settings: 4
Number of Iterations: 30
Debug Print Level Control Flag: 1
Convergence Tolerance: 1.0E-1
Threshold Pivot Tolerance: 1.0E-4
Parameter Tolerance: 1.0E-6
Objective FunctionTolerance: 1.0E-6
Real Working Array Dimension: 100000
Integer Working Array Dimension: 10000

Estimation Solver Options

○ Non-Linear Least-Squares
● Bayesian Estimation: Single Response
□ Check here to Test the Initial Parameter Values
□ Check here to Include the IMSL Fortran Libary

Gradient Calculation

○ Central Differences Scheme
● Forward Differences Scheme
○ User Supplied Response Derivatives

Relative Perturbation Step Size: 1.0E-3

From the **Nonlinear Regression Parameters** group enter:

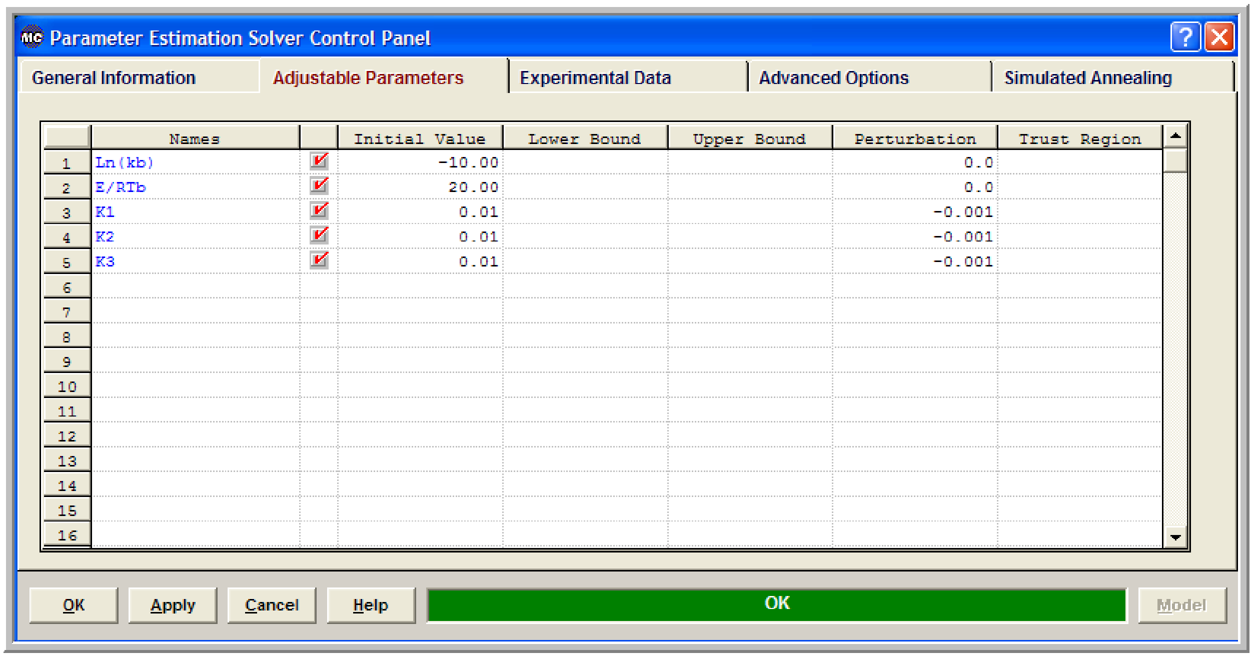
- The number of unknown parameters **(5)**
- The number of experiments or datasets **(27)**
- The number of responses or dependent variables **(1)** and
- The number of experimental settings or predictor variables **(4)**

Optionally you may change the *Number of Iterations*, the *Convergence and Parameter Tolerance*, the *Debug Print Level Control Flag* and the Real and Integer working arrays space requirements. From the **Estimation Solver Options** group choose the type of estimation you are going to be using and optionally request a diagonal covariance if the Bayesian estimation has been chosen, or *Click here to perform a Test Call to Model*. The test call to model is a very useful option since it allows you to see how good is your initial parameter guess before you proceed with the estimation process. You also have the option to use a relative weighting factor of the observation vector in the estimation process. From the **Derivatives Calculation** group choose the method for the objective function gradient calculation and optionally enter the

Relative Perturbation Step Size. Should you choose User Supplied Model Derivatives you must enter the section **@Gradient Vector** when you enter the source code.

7.5.2.1 The Adjustable Parameters

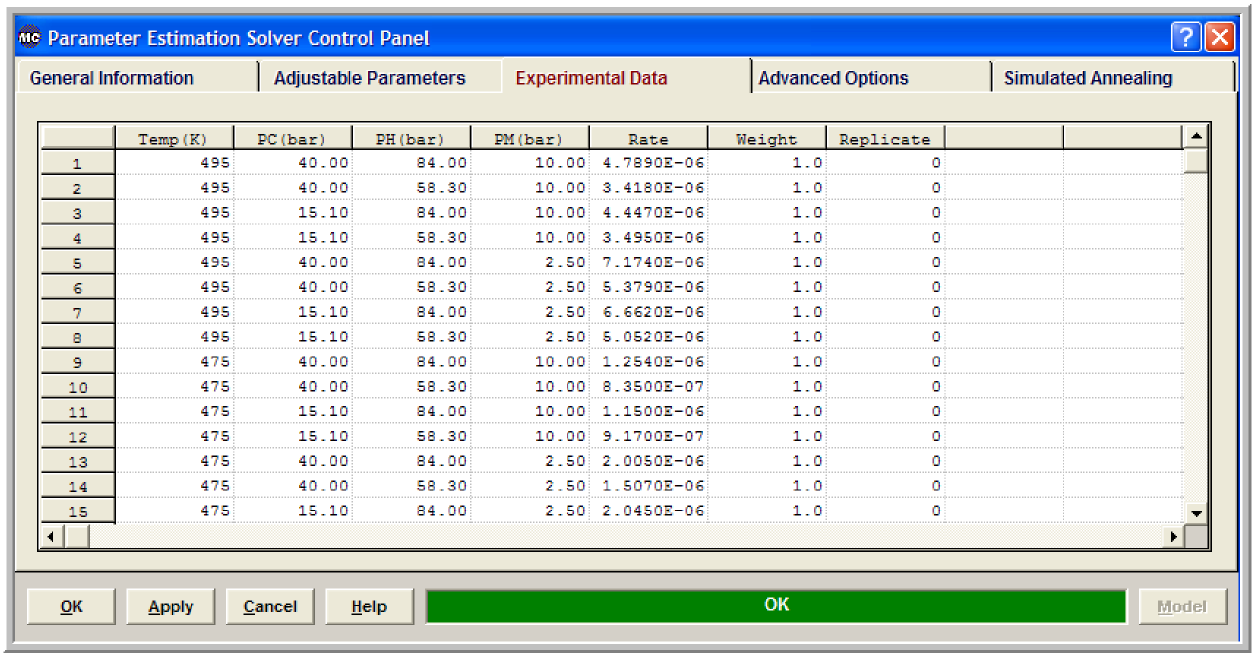
Next select the **Adjustable Parameters** tab on the Parameter Estimation Control Panel and enter the names of the parameters you wish to estimate, their initial guesses and optionally their lower and upper bounds, the relative perturbation step size for gradient computation and the size of the trust region. Notice that if a parameter is not checked in the check box next to it, will remain fixed at its initial value during the estimation. The following data have been inserted for our example:



	Names		Initial Value	Lower Bound	Upper Bound	Perturbation	Trust Region
1	Ln(kb)	☑	-10.00			0.0	
2	E/RTb	☑	20.00			0.0	
3	K1	☑	0.01			-0.001	
4	K2	☑	0.01			-0.001	
5	K3	☑	0.01			-0.001	
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							

7.5.2.2 The Experimental Data

Next select the **Experimental Data** tab on the Parameter Estimation Control Panel and enter the observations and experimental settings. Optionally you may enter weights for each observation. Otherwise these weights are set equal to one by the Parameter Estimation solver. Windows Copy and Paste functions can be used to transfer your data from, say, an Excel spreadsheet. The following data have been inserted for our example: (partial list shown)



	Temp (K)	PC (bar)	PH (bar)	PM (bar)	Rate	Weight	Replicate
1	495	40.00	84.00	10.00	4.7890E-06	1.0	0
2	495	40.00	58.30	10.00	3.4180E-06	1.0	0
3	495	15.10	84.00	10.00	4.4470E-06	1.0	0
4	495	15.10	58.30	10.00	3.4950E-06	1.0	0
5	495	40.00	84.00	2.50	7.1740E-06	1.0	0
6	495	40.00	58.30	2.50	5.3790E-06	1.0	0
7	495	15.10	84.00	2.50	6.6620E-06	1.0	0
8	495	15.10	58.30	2.50	5.0520E-06	1.0	0
9	475	40.00	84.00	10.00	1.2540E-06	1.0	0
10	475	40.00	58.30	10.00	8.3500E-07	1.0	0
11	475	15.10	84.00	10.00	1.1500E-06	1.0	0
12	475	15.10	58.30	10.00	9.1700E-07	1.0	0
13	475	40.00	84.00	2.50	2.0050E-06	1.0	0
14	475	40.00	58.30	2.50	1.5070E-06	1.0	0
15	475	15.10	84.00	2.50	2.0450E-06	1.0	0

Finally you may specify replicate experiments for performing lack-of-fit analysis and model discrimination. Choose **OK** and continue

7.5.3 Saving and Running your Model

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***. This will save your model and create the Fortran code that will access the Parameter Estimation solver. The ***Save As*** dialog box appears.
- ➢ In the Directories box, double-click a directory where you want to store the source file (or down a directories path to the appropriate directory.)
- ➢ Type a filename (a filename cannot contain the following characters: \ / : * ? “ < > |) in the File Name box, then choose OK. The default extension given to a file is **AVW**.
- ➢ To view the Fortran code that you have just created from the ***View*** menu choose ***Fortran Code***.

New files are labeled UNTITLED until they are saved. The maximum number of characters in a line is 132; the maximum number of lines in a file is infinity. Before you can save or close a window it must be active. To make a window active, either switch to the window (by clicking anywhere in it) or choose the window name or number from the Window menu.

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**)

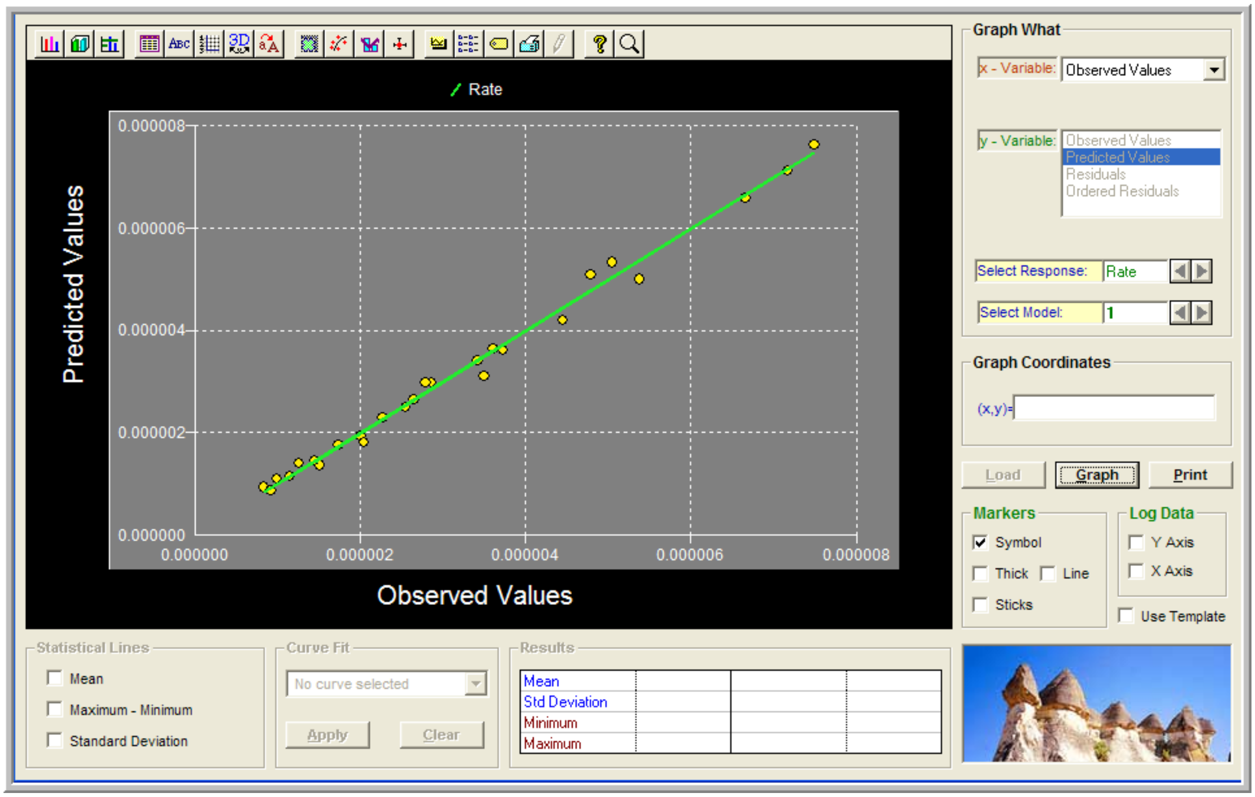
7.5.4 Numerical Results

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 Experiments..........................   27
Number of Parameters...........................    5
Number of Responses............................    1
Number of Settings.............................    4

EXIT GREGPLUS: SOLUTION FOUND. MODEL No. 1

STATISTICAL ANALYSIS

OBJECTIVE FUNCTION.......................  7.67167E-13
SUM OF SQUARES OF RESIDUALS..............  7.67167E-13
ESTIMATED PARAMETERS FROM DATA...........            5
TOTAL OBSERVATIONS DATA COUNT IS.........           27

              OPTIMAL
PARAMETER    ESTIMATES       95% MARGINAL HPD INTERVALS      PARAMETER STATUS
 PAR( 1)  -9.709381E+00     -9.709381E+00 +- 2.248E-01      Estimated
 PAR( 2)   3.149077E+01      3.149077E+01 +- 1.595E+00      Estimated
 PAR( 3)   6.562781E-02      6.562781E-02 +- 1.755E-02      Estimated
 PAR( 4)   2.904504E-02      2.904504E-02 +- 1.019E-02      Estimated
 PAR( 5)   9.838805E-02      9.838805E-02 +- 2.965E-02      Estimated

EVENT      OBSERVED         PREDICTED         RESIDUAL
  1       4.7890E-06       5.0927E-06      -3.0373E-07
  2       3.4180E-06       3.4141E-06       3.8616E-09
  3       4.4470E-06       4.2071E-06       2.3986E-07
  4       3.4950E-06       3.1181E-06       3.7688E-07
  5       7.1740E-06       7.1222E-06       5.1815E-08
  6       5.3790E-06       4.9984E-06       3.8058E-07
  7       6.6620E-06       6.5932E-06       6.8788E-08
  8       5.0520E-06       5.3324E-06      -2.8040E-07
  9       1.2540E-06       1.3934E-06      -1.3940E-07
 10       8.3500E-07       9.3725E-07      -1.0225E-07
 11       1.1500E-06       1.1570E-06      -7.0059E-09
 12       9.1700E-07       8.6525E-07       5.1752E-08
 13       2.0050E-06       1.9442E-06       6.0819E-08
 14       1.5070E-06       1.3656E-06       1.4143E-07
 15       2.0450E-06       1.8021E-06       2.4293E-07
 16       1.4430E-06       1.4607E-06      -1.7668E-08
 17       2.8580E-06       2.9767E-06      -1.1869E-07
 18       2.7840E-06       2.9767E-06      -1.9269E-07
 19       2.7930E-06       2.9767E-06      -1.8369E-07
 20       7.4900E-06       7.6282E-06      -1.3818E-07
 21       9.9000E-07       1.0916E-06      -1.0159E-07
 22       1.7390E-06       1.7522E-06      -1.3198E-08
 23       3.7250E-06       3.6249E-06       1.0010E-07
 24       2.6460E-06       2.6532E-06      -7.1546E-09
 25       2.5520E-06       2.5005E-06       5.1482E-08
 26       3.6060E-06       3.6315E-06      -2.5525E-08
 27       2.2710E-06       2.3007E-06      -2.9742E-08


NUMBER OF ITERATIONS..............     6
NUMBER OF FUNCTION CALLS..........    52
```

7.5.5 Graphical Results

If you wish to see various graphs of the estimation process from the ***View*** menu choose ***Solution Graphs***, or click The AthenaVisual Plus graphics control panel appears:



x - Variable: Observed Values

y - Variable: Observed Values / **Predicted Values** / Residuals / Ordered Residuals

Select Response: Rate

Select Model: 1

(x,y)=

☑ Symbol ☐ Thick ☐ Line ☐ Sticks

☐ Y Axis ☐ X Axis ☐ Use Template

☐ Mean ☐ Maximum - Minimum ☐ Standard Deviation

No curve selected

Mean			
Std Deviation			
Minimum			
Maximum			

In this window fist we click ***Load*** to load the numerical results. Then in the **Graph What** group we select the **x-variable** (here **Observed Values**) and the **y-variable** (here **Predicted Values**) and click ***Graph***. You should see the parity 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 other graphs, like plotting of residuals, plotting observed and predicted values against the event number, or against any selected settings.

The following residual plots are also available in order to aid you in the model criticism and improvement process:

- **Scatter Plot**. Choose any Experimental Setting as the **x-variable** and the Residuals as the **y-Variable**. Plots in which the residuals do not exhibit any systematic structure indicate that the model fits the data well. Plots of the residuals that exhibit any systematic structure indicate that the model can be improved.

- **Run Order (or Sequence) Plot**. Choose the Event Number as the **x-variable** and the Residuals as the **y-Variable**. This plots are used to check for drift in the process. This plot is useful, however, only if data have been collected in a randomized order.
- **Lag Plot**. Choose the Residuals as the **x-variable** and the Residuals as the **y-Variable**. This type of plot suggests whether or not the observation errors are independent. If the errors are not independent, then the estimate of the error standard deviation will be biased, and that may lead to improper inferences about the process. The lag plot is constructed by plotting each residual value versus the value of the successive residual; thus the first residual is plotted versus the second, the second versus the third and so on.

- **Normal Probability Plot**. Choose the Normal Statistic Medians as the **x-variable** and the Ordered Residuals as the **y-Variable**. These plots are used to check whether or not it is reasonable for us to assume that the random errors inherent in the process we are modeling have been drawn from a normal probability distribution. This normality assumption is necessary for the error magnitude we are willing to accept when making decisions about the model. If the random errors are not from a normal distribution, then incorrect decisions will be made.

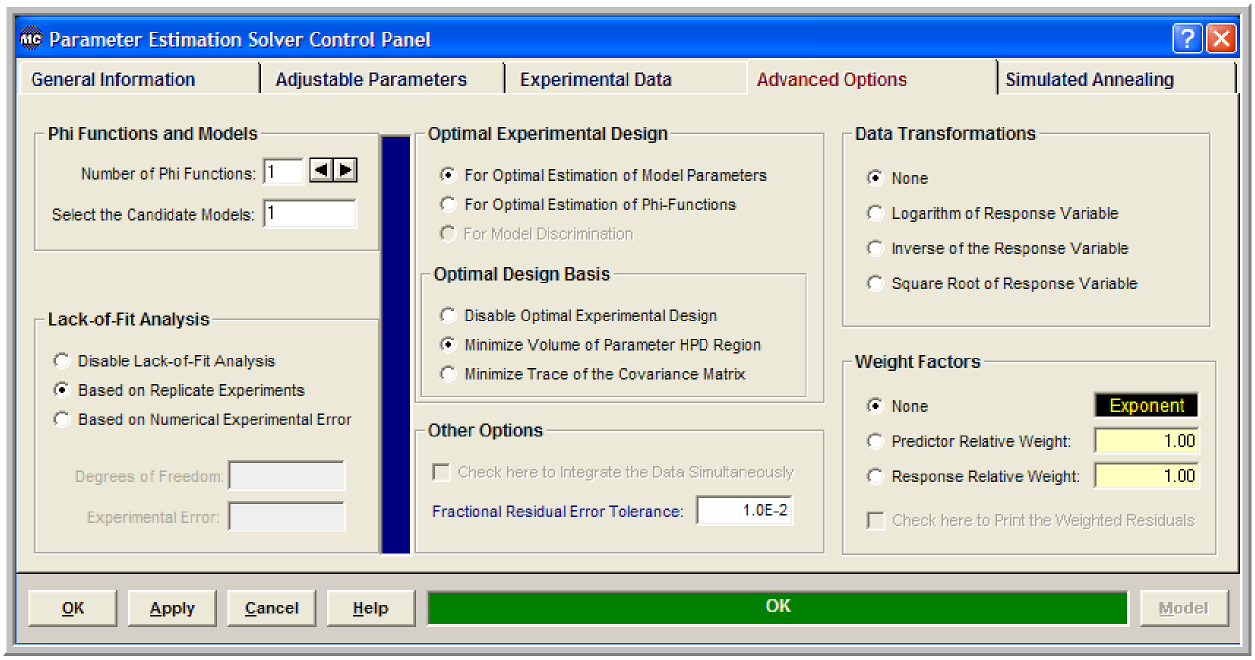
7.5.6 Parameter Transformations – The Phi Functions

In the example above, the second parameter **Par**(2) corresponds to the dimensionless activation energy defined as: $\mathbf{Par}(2)=\dfrac{E}{RT_{\text{ref}}}$. Suppose that we are also interested in the estimation of the actual activation energy, and its Highest Posterior Density (HPD) interval. To accomplish this task we use the concept of the **Phi** functions. These functions represent explicit and implicit parameter transformations of any kind and are used by the estimation solver to calculate point values and HPD intervals. There is no limit in the number of the Phi functions that the user can implement. For our example, suppose that we assign the activation energy to the first Phi function according to the equation: $E \equiv \text{Phi}(1)$, then the following equation defines this Phi function:

$$\mathbf{Phi}(1)=\mathbf{Par}(2)*RT_{\text{ref}}$$

To implement this concept:

- From the ***Model*** menu choose ***Load Solver.*** The **Parameter Estimation Solver Control Panel** appears:



Phi Functions and Models

Number of Phi Functions: 1

Select the Candidate Models: 1

Lack-of-Fit Analysis

○ Disable Lack-of-Fit Analysis
● Based on Replicate Experiments
○ Based on Numerical Experimental Error

Degrees of Freedom:

Experimental Error:

Optimal Experimental Design

● For Optimal Estimation of Model Parameters
○ For Optimal Estimation of Phi-Functions
○ For Model Discrimination

Optimal Design Basis

○ Disable Optimal Experimental Design
● Minimize Volume of Parameter HPD Region
○ Minimize Trace of the Covariance Matrix

Other Options

□ Check here to Integrate the Data Simultaneously

Fractional Residual Error Tolerance: 1.0E-2

Data Transformations

● None
○ Logarithm of Response Variable
○ Inverse of the Response Variable
○ Square Root of Response Variable

Weight Factors

● None — Exponent
○ Predictor Relative Weight: 1.00
○ Response Relative Weight: 1.00

□ Check here to Print the Weighted Residuals

- ❖ Click the **Advanced Options** tab.
- ❖ In the **Phi Functions and Models** group use the spin control button to enter the *Number of Phi Functions* (in our case 1), then choose **OK**.
- ❖ From the ***Model*** menu choose ***Add Phi Functions***, then ***Parameter Transformations*** and enter the code as shown below:

```
@Phi Functions
 Phi(1)=Par(2)*Rg*Tref       ! Activation Energy
```

You may now choose to compile, build and execute your project; to do that. From the ***Build*** menu choose ***Compile*** then choose ***Build EXE*** and finally ***Execute***

In the results file, scroll down and you will see the following output:

```
EXIT GREGPLUS: USER PREDICTED FUNCTIONS

 FUNCTIONS           PREDICTED        95% MARGINAL HPD INTERVALS
  PHI( 1)          1.269799E+05     1.269799E+05   +- 6.433E+03
```

From the above output we can clearly see that the value of the activation energy in kJ/mol and its HPD interval is equal to $E = 126.9 \pm 6.4$.

7.5.7 Making Model Predictions – The Phi Functions

The concept of the **Phi** functions can be extended to make model predictions and also calculate the prediction HPD intervals of the predicted values in a straightforward manner. Suppose for example, that we wish to calculate the Methanol production rate and its HPD interval at the following conditions of the experimental settings: (some of these settings correspond to extrapolation and some to interpolation):

$$\begin{array}{llll} T = 480\,K & P_C = 30\,bar & P_H = 70\,bar & P_M = 6\,bar \\ T = 600\,K & P_C = 30\,bar & P_H = 70\,bar & P_M = 6\,bar \end{array}$$

Following the logic above we load the solver and increase the number of Phi functions to 4, using the spin control button. Then we enter the following code that is shown below:

```
@Phi Functions
 Phi(1)=Par(2)*Rg*Tref        ! Activation Energy
 Phi(2)=exp(Par(1)+Par(2))    ! Frequency Factor

! Calculation of Reaction Rate for a User Specified Set of Conditions
! Selected Temperature is within the Range of Experimental Conditions
!--------------------------------------------------------------------
 Temp=480.0
 PC=30.0
 PH=70.0
 PM=6.0

 RK=exp(Par(1)+Par(2)*(1.0-Tref/Temp))
 K1=Par(3)
 K2=Par(4)
 K3=Par(5)
 KEQ=10^(3914.0/Temp-7.536*log10(Temp)+0.001766*Temp+9.388)
 Phi(3)=RK*(PC*PH**2-PM/KEQ)*K1*K2**2/(1.0+K1*PC+K2*PH+K3*PM)**3

! Calculation of Reaction Rate for a User Specified Set of Conditions
! Selected Temperature is outside the Range of Experimental Conditions
!---------------------------------------------------------------------
 Temp=600.0
 RK=exp(Par(1)+Par(2)*(1.0-Tref/Temp))
 KEQ=10^(3914.0/Temp-7.536*log10(Temp)+0.001766*Temp+9.388)
 Phi(4)=RK*(PC*PH**2-PM/KEQ)*K1*K2**2/(1.0+K1*PC+K2*PH+K3*PM)**3
```

We run the model and in the results file, if we scroll down we can see the following output:

```
EXIT GREGPLUS: USER PREDICTED FUNCTIONS

 FUNCTIONS          PREDICTED        95% MARGINAL HPD INTERVALS
  PHI( 1)       1.269799E+05      1.269799E+05   +- 6.433E+03
  PHI( 2)       2.880970E+09      2.880970E+09   +- 4.584E+09
  PHI( 3)       2.029647E-06      2.029647E-06   +- 9.782E-08
  PHI( 4)       7.361967E-04      7.361967E-04   +- 2.074E-04

 NORMALIZED COVARIANCE MATRIX
   1.000
   0.990  1.000
  -0.866 -0.863  1.000
   0.996  0.985 -0.820  1.000
```

From the above output we can clearly see that the value of the methanol production rates and their HPD interval are equal to:

$$\Re_{T=480} \times 10^{6} = 2.03 \pm 0.098$$

and

$$\Re_{T=600} \times 10^{4} = 7.36 \pm 2.074$$

Observe how the confidence intervals change for the extrapolated values. This is to be expected since the uncertainty of the predictions increases as we move away from the region where the experiments were conducted.

8.0 Parameter Estimation from Multi-Response Data

Multiresponse experimentation is important in studies of complex systems, and of systems observable by multiple methods. Chemical engineers and chemists use such data to investigate chemical reactions, mixtures and processes; similar structures of data occur in other fields of science and engineering. Here we study modern methods for investigating models with such data. Bayes' Theorem is the key to parameter estimation, and Jeffreys' rule takes greater prominence here.

Least-squares methods are not appropriate for multiresponse investigations, unless the various responses have known relative precisions and independent, unbiased Normal distributions of error. Nonlinear multiresponse models were treated under these special assumptions by Gauss (1809) and Deming (1943). Aitken (1935) generalized linear weighted least squares to multiple responses with a specified error covariance matrix; his method was extended to nonlinear parameter estimation by Bard (1968, 1974) and by Klaus and Rippin (1979). Programs for nonlinear least squares are also adaptable by expert users to per form such calculations. This approach to multiresponse estimation is rather subjective, however, since the covariance matrix is seldom known.

Bayes' theorem (Bayes 1763; Box and Tiao 1973) permits estimation of the error covariance matrix $\boldsymbol{\Sigma}$ from a multiresponse data set, along with the parameter vector $\boldsymbol{\theta}$ of a predictive model. It is also possible, in simple cases, to avoid explicit calculation of $\boldsymbol{\Sigma}$ and estimate only $\boldsymbol{\theta}$. Methods for such calculations began to appear in the 1960's and are now quite far developed. This chapter summarizes these modern Bayesian methods.

Jeffreys (1961) made a major advance in Bayesian estimation theory by giving a noninformative prior probability density for the parameters in any suitably differentiable model. The naturalness and objectivity of his analysis led to renewed interest in Bayesian methods.

Box and Draper (1965) took another major step by deriving a posterior density function $p\left(\theta|Y\right)$ for estimating a parameter vector $\boldsymbol{\theta}$ from a full matrix $\boldsymbol{Y}$ of multiresponse observations. The errors in the observations were assumed to be Normally distributed with an unknown full covariance matrix $\boldsymbol{\Sigma}$. Michael Box and Norman Draper (1972) gave a corresponding density function for treating a data matrix $\boldsymbol{Y}$ of independent blocks of responses, and applied this function to sequential design of experiments. Later workers have considered more flexible data structures and error models.

The posterior density function $p\left(\theta|Y\right)$ found by Box and Draper (1965) is a power of the determinant $\left|\boldsymbol{v}\left(\theta\right)\right|$. These authors used contour plots of $p\left(\theta|Y\right)$ to determine approximate 95%

probability regions for a two-parameter model. This technique is useful for studying models with small parameter sets.
Multiparameter models require digital optimization methods. Early workers minimized $\left|\boldsymbol{v}(\theta)\right|$ by search techniques, which proved to be slow and gave only point estimates of $\boldsymbol{\theta}$. Newton like algorithms for minimization of $\left|\boldsymbol{v}(\theta)\right|$, and for interval estimation of $\boldsymbol{\theta}$, were given by Stewart and Sørensen (1976, 1981) and by Bates and Watts (1985, 1987). Related algorithms for likelihood-based estimation were developed by Bard (1968, 1974), and were extended by Klaus and Rippin (1979) and Steiner *et al*. (1986).

Several generalizations of the problem considered by Box and Draper (1965) have been analyzed in the literature. The theory has been extended to more general models and data structures, and some interesting applications have appeared. These results are described in this chapter along with available software for further applications.

8.1 The Parameter Estimation Problem

The main theme of this section is the estimation of the parameter vector $\boldsymbol{\theta}$ (and explicitly or implicitly the covariance matrix $\boldsymbol{\Sigma}$ in a model

$$Y_{ui} = F_i\left(\boldsymbol{x}_u, \theta\right) + E_{ui} \qquad \left(\boldsymbol{u} = 1, \ldots, \boldsymbol{n}; \boldsymbol{i} = 1, \ldots, \boldsymbol{m}\right) \tag{5.2-1}$$

for the elements $Y_{ui} = \sqrt{\boldsymbol{w}_u}\boldsymbol{y}_{ui}$ of a multiresponse weighted data matrix $\boldsymbol{Y}$. Each integer $\boldsymbol{u}$ from 1 to $\boldsymbol{n}$ denotes an independent event, in which $\boldsymbol{m}_u \leq \boldsymbol{m}$ responses are observed. A weight $\boldsymbol{w}_u$ is considered given for each event to express its reliability relative to some standard. The weighted function $F_i\left(\boldsymbol{x}_u, \theta\right) := \sqrt{\boldsymbol{w}_u}\boldsymbol{f}_i\left(\boldsymbol{x}_u\theta\right)$ is an expectation model for response $\boldsymbol{i}$ at the experimental design point $\boldsymbol{x}_u$ and is assumed to be differentiable with respect to each parameter θ_r.

The weighted errors $E_{ui} := \sqrt{\boldsymbol{w}_u}\varepsilon_{ui}$ are modeled by an m-dimensional Normal distribution, with expected values of zero and unknown covariances $\sigma_{ij} := E\left(E_{ui}E_{uj}\right)$. Use of this distribution with Eq. (5.2-1) yields a density function $p\left(Y\middle|\theta, \Sigma\right)$ for observations to be taken at planned test conditions $\left\{x_1, \ldots, x_n\right\}$. This function can take various forms, discussed below, depending on the locations that are occupied in the observation matrix Y and in the covariance matrix Σ.

Once data ***Y*** are available, the likelihood function $l\left(\theta,\Sigma\middle|Y\right)$ is constructed in the manner of Fisher (1922) as the function $p\left(Y\middle|\theta,\Sigma\right)$, with ***Y*** now given whereas $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ are free to vary. Multiplication of the likelihood by a suitable prior density $p\left(\theta,\Sigma\right)$, in accordance with Bayes' theorem gives the posterior density function $p\left(\theta,\Sigma\middle|Y\right)$, which contains all current information on $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$. These constructions are summarized below for several problem types.

Box and Draper (1965) derived a posterior density function for estimating the expectation parameter vector $\boldsymbol{\theta}$ from a full data array ***Y*** with full unknown covariance matrix $\boldsymbol{\Sigma}$. With this type of data, every event ***u*** has a full set of $\boldsymbol{m}$ responses. The density function for prospective data ***Y***, consistent with Eqs. (5.2-1) is

$$
\begin{aligned}
p\left(Y\middle|\theta,\Sigma\right) &= \prod_{u=1}^{n} p\left(Y_u\middle|\theta,\Sigma\right)\\
&= \left|2\pi\Sigma\right|^{-n/2}\exp\left\{-\tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\sigma^{ij}\sum_{u=1}^{n}\left[Y_{ui}-F_i\left(x_u,\theta\right)\right]\left[Y_{uj}-F_j\left(x_u,\theta\right)\right]\right\}\\
&= \left|2\pi\Sigma\right|^{-n/2}\exp\left\{-\tfrac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\sigma^{ij}\boldsymbol{v}_{ji}\left(\theta\right)\right\} \qquad (5.2\text{-}2)\\
&= \left|2\pi\Sigma\right|^{-n/2}\exp\left\{-\tfrac{1}{2}\sum_{i=1}^{m}\left[\Sigma^{-1}\boldsymbol{v}\left(\theta\right)\right]_{\boldsymbol{ii}}\right\}\\
&= \left|2\pi\Sigma\right|^{-n/2}\exp\left\{-\tfrac{1}{2}\,\mathrm{tr}\left[\Sigma^{-1}\boldsymbol{v}\left(\theta\right)\right]\right\}
\end{aligned}
$$

The matrices Σ^{-1} and $\boldsymbol{v}\left(\theta\right)$ are symmetric, with elements σ^{ij} and

$$
v_{ij}\left(\theta\right)=\sum_{u=1}^{n}\left[Y_{ui}-F_i\left(x_u,\theta\right)\right]\left[Y_{uj}-F_j\left(x_u,\theta\right)\right] \qquad (5.2\text{-}3)
$$

respectively; also **tr** denotes the trace of a matrix: tr $\boldsymbol{A}=\Sigma_{\boldsymbol{i}}\boldsymbol{a}_{\boldsymbol{ii}}$

When data ***Y*** are available, one interprets the right-hand member of Eq. (5.2-2) as a likelihood function

$$
\boldsymbol{l}\left(\theta,\Sigma\middle|Y\right)\propto\left|\Sigma\right|^{-n/2}\exp=\left\{-\tfrac{1}{2}\,\mathrm{tr}\left[\Sigma^{-1}\boldsymbol{v}\left(\theta\right)\right]\right\} \qquad (5.2\text{-}4)
$$

for the postulated model and the given data, in the manner of Fisher (1922). Eq. (5.2-4) is meaningful for all positive definite values of $\boldsymbol{\Sigma}$, provided that $\boldsymbol{v}(\boldsymbol{\theta})$ is nonsingular over the permitted range of $\boldsymbol{\theta}$. In practice, $\boldsymbol{v}(\boldsymbol{\theta})$ can always be made nonsingular by proper choice of the working responses; this is done automatically in the subroutine package GREGPLUS of the software **AthenaVisual Plus**.

A noninformative prior density $\boldsymbol{p}(\boldsymbol{\Sigma})$ may be derived from Eq. (5.2-2) by the method of Jeffreys. The result,

$$p(\Sigma) \propto \lfloor \Sigma \rfloor^{-(m+1)/2}, \qquad (5.2\text{-}5)$$

was given by Box and Draper (1965) and a derivation is given on page 475 of Box and Tiao (1973). This result holds when the elements σ_{ij} for $i \geq j$ are all regarded as independent unknowns. A uniform prior density is assumed for $\boldsymbol{\theta}$ over the range of appreciable $\ell(\theta,\Sigma|Y)$; then the joint prior $p(\theta,\Sigma)$ also has the form in Eq. (5.2-5).

Multiplying the likelihood by the prior in accordance with Bayes' theorem, Box and Draper obtained the posterior density function

$$p(\theta,\Sigma|Y) \propto |\Sigma|^{-(n+m+1)/2} \exp\left\{-\tfrac{1}{2}\operatorname{tr}\left[\Sigma^{-1}\boldsymbol{v}(\theta)\right]\right\} \qquad (5.2\text{-}6)$$

for the parameters of the expectation and error models. This formula gives all the information obtainable from the data regarding the unknown vector $\boldsymbol{\theta}$ and matrix $\boldsymbol{\Sigma}$.

Box and Draper chose to remove $\boldsymbol{\Sigma}$ from the problem by integrating Eq. (5.2-6) over the positive definite range of $\boldsymbol{\Sigma}^{-1}$, just as one removes the variance $\boldsymbol{\sigma}$ from single-response problems by suitable integration. In this way they found

$$p(\theta|Y) \propto |\boldsymbol{v}(\theta)|^{-n/2} \qquad (5.2\text{-}7)$$

as the *marginal* posterior density for the parameters of the expectation model. The mode of this function occurs at the minimum of the determinant $|\boldsymbol{v}(\theta)|$ thus providing a multiresponse generalization of least squares.

Another way of reducing Eq. (5.2-6) is to use the conditional maximum-density estimate of $\boldsymbol{\Sigma}$ at each value of $\boldsymbol{\theta}$. This gives the modified posterior density function

$$\tilde{\boldsymbol{p}}\left(\theta\middle|Y\right) := \boldsymbol{p}\left(\theta,\tilde{\Sigma}(\theta)\middle|Y\right) \propto \left|\boldsymbol{v}(\theta)\right|^{-(n+m+1)/2} \qquad (5.2\text{-}8)$$

which has the same modal $\boldsymbol{\theta}$ value as Eqs. (5.2-6) and (5.2-7). This density function is steeper than the one in Eq. (5.2-7), and thus gives sharper estimates of $\boldsymbol{\theta}$.

Equation (5.2-6) gives the fullest information from this type of data, and was used by Stewart and Sørensen (1981) to obtain modal and interval estimates of $\boldsymbol{\theta}$ and $\boldsymbol{\Sigma}$ for a sample problem. The modal $\boldsymbol{\theta}$ occurs at the minimum of $\left|\boldsymbol{v}(\theta)\right|$, and the modal covariance estimate is

$$\hat{\Sigma} = \boldsymbol{v}\left(\hat{\theta}\right)\Big/(n+m+1) \qquad (5.2-9)$$

Bard (1974) gave formulas for parameter estimation based on the likelihood function in Equation (5.2-4). The antilogarithm of his “concentrated likelihood” function $\tilde{\mathcal{L}}(\theta)$ corresponds to $\boldsymbol{l}\left(\theta,\tilde{\Sigma}(\theta)|Y\right)$ which is proportional to the density function given in Eq. (5.2-7) and thus is not as steep as Eq. (5.2-8). It follows that Bayesian estimation, with the prior of Eq. (5.2-5), is preferable for its sharpness as well as for the consistency properties of the Jeffreys prior.

8.2 References

M. Caracotsios, *Model Parametric Sensitivity Analysis and Nonlinear Parameter Estimation. Theory and Applications*. Ph. D. thesis, Dept. of Chemical Engineering, University of Wisconsin, Madison (1986).

W. E. Stewart, M. Caracotsios, and J. P. Sorensen, *Parameter estimation from multiresponse data*, AIChE J., 38, 641-650 (1992); Errata, 38, 1302 (1992).

W. E. Stewart, and M. Caracotsios, *Computer-Aided Modeling of Reactive Systems*, John Wiley and Sons, Inc. (2008).

8.3 Parameter Estimation with an Implicit Model

- Start **Athena Visual Studio**
- From the **File** menu select **New.**
- You are in the **Process Modeling** tab.
- Select the **Parameter Estimation** tab.
- Click **Estimation with Ordinary Differential Equations.**
- 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.**

Nonlinear Model Parameter Estimation with implicit or explicit models may include nonlinear systems of algebraic equations, mixed systems of differential and algebraic equations, as well as boundary value problems or partial differential equations. This example problem has been created to test the functionality of AthenaVisual Plus in dealing with the nonlinear parameter estimation of reaction rate constants from multi-response data at different temperature levels with systems of ordinary differential equations. Experimental data at different temperature levels allows to estimate both the reaction activation energy and the reaction rate constant at a base temperature.

This example problem has been created to test the functionality of AthenaVisual Plus in dealing with parameter estimation from multi-response data with systems of differential equations.

8.4 Rate Constants Estimation in Glucose Chemistry

The mathematical model for the fermentation of the bacterium Pseudomonas Ovalis, that produces gluconic acid, is described by the following system of differential equations:

Rate of cell growth	$\frac{dC_1}{dt} = k_1 C_1 \left(1 - \frac{C_1}{k_2}\right)$
Rate of gluconolactone formation	$\frac{dC_2}{dt} = \frac{k_3 C_1 C_4}{k_4 + C_4} - \alpha k_5 C_2$
Rate of gluconic acid formation	$\frac{dC_3}{dt} = k_5 C_2$
Rate of glucose consumption	$\frac{dC_4}{dt} = -\beta \frac{k_3 C_1 C_4}{k_4 + C_4}$
Reaction Rate Constant Form	$k_i = k_{iB} \exp\left[\frac{E_i}{RT_B}\left(1 - \frac{T_B}{T}\right)\right]$

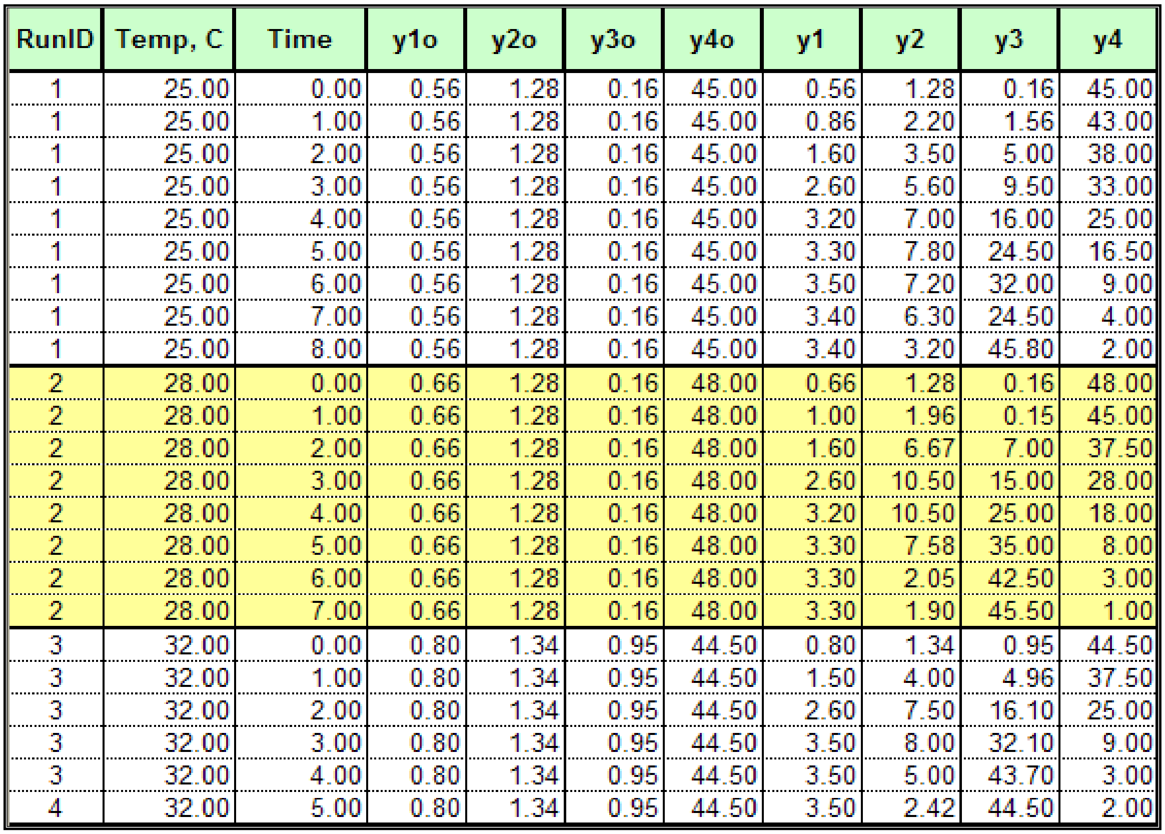
Experiments at three different temperature levels were conducted for various initial concentrations. From these experiments we wish to perform the following tasks:

- Estimate the parameters $\{k_{1B}, k_{2B}, k_{3B}, k_{4B}, k_{5B}, E_{1B}, E_{2B}, E_{3B}, E_{4B}, E_{5B}\}$ and their Highest Posterior Density Intervals
- Make parameter transformations by use of the **Phi** functions
- Investigate the use of partial analytical derivative information
- Conduct an interpolation and an extrapolation study to assess the model accuracy

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

Model Parameters	**Athena Variables**	**Initial Conditions**	**Athena Variables**	**Nomenclature**
$\alpha = 0.9082$	a	$C_1(0) = \{0.56, 0.66, 0.80\}$	U(1)	Cell Concentration
$\beta = 1.0110$	b	$C_2(0) = \{1.28, 1.28, 1.34\}$	U(2)	Glunolactone Concentration
$\{k_{1B}, E_{1B}\} = \{0.1, 0.0\}$	k1b, E1b	$C_3(0) = \{0.16, 0.16, 0.95\}$	U(3)	Gluconic Acid Concentration
$\{k_{1B}, E_{1B}\} = \{0.1, 0.0\}$	k2b, E2b	$C_4(0) = \{45.0, 48.0, 44.5\}$	U(4)	Glucose Concentration
$\{k_{1B}, E_{1B}\} = \{0.1, 0.0\}$	k3b, E3b			
$\{k_{1B}, E_{1B}\} = \{0.1, 0.0\}$	k4b, E4b			
$\{k_{1B}, E_{1B}\} = \{0.1, 0.0\}$	k5b, E5b			

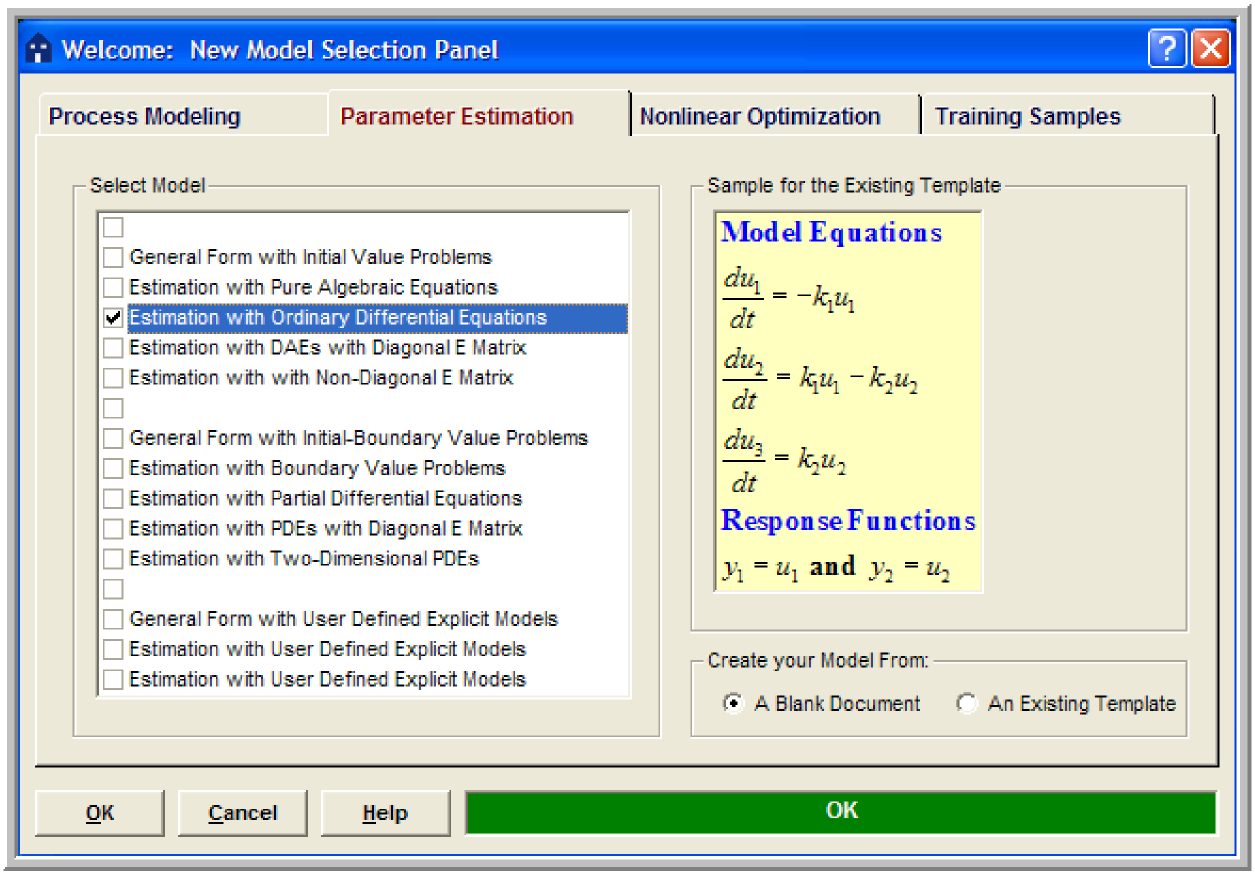
The experimental data at different levels of temperature and reaction time for this example are:



RunID	Temp, C	Time	y1o	y2o	y3o	y4o	y1	y2	y3	y4
1	25.00	0.00	0.56	1.28	0.16	45.00	0.56	1.28	0.16	45.00
1	25.00	1.00	0.56	1.28	0.16	45.00	0.86	2.20	1.56	43.00
1	25.00	2.00	0.56	1.28	0.16	45.00	1.60	3.50	5.00	38.00
1	25.00	3.00	0.56	1.28	0.16	45.00	2.60	5.60	9.50	33.00
1	25.00	4.00	0.56	1.28	0.16	45.00	3.20	7.00	16.00	25.00
1	25.00	5.00	0.56	1.28	0.16	45.00	3.30	7.80	24.50	16.50
1	25.00	6.00	0.56	1.28	0.16	45.00	3.50	7.20	32.00	9.00
1	25.00	7.00	0.56	1.28	0.16	45.00	3.40	6.30	24.50	4.00
1	25.00	8.00	0.56	1.28	0.16	45.00	3.40	3.20	45.80	2.00
2	28.00	0.00	0.66	1.28	0.16	48.00	0.66	1.28	0.16	48.00
2	28.00	1.00	0.66	1.28	0.16	48.00	1.00	1.96	0.15	45.00
2	28.00	2.00	0.66	1.28	0.16	48.00	1.60	6.67	7.00	37.50
2	28.00	3.00	0.66	1.28	0.16	48.00	2.60	10.50	15.00	28.00
2	28.00	4.00	0.66	1.28	0.16	48.00	3.20	10.50	25.00	18.00
2	28.00	5.00	0.66	1.28	0.16	48.00	3.30	7.58	35.00	8.00
2	28.00	6.00	0.66	1.28	0.16	48.00	3.30	2.05	42.50	3.00
2	28.00	7.00	0.66	1.28	0.16	48.00	3.30	1.90	45.50	1.00
3	32.00	0.00	0.80	1.34	0.95	44.50	0.80	1.34	0.95	44.50
3	32.00	1.00	0.80	1.34	0.95	44.50	1.50	4.00	4.96	37.50
3	32.00	2.00	0.80	1.34	0.95	44.50	2.60	7.50	16.10	25.00
3	32.00	3.00	0.80	1.34	0.95	44.50	3.50	8.00	32.10	9.00
3	32.00	4.00	0.80	1.34	0.95	44.50	3.50	5.00	43.70	3.00
4	32.00	5.00	0.80	1.34	0.95	44.50	3.50	2.42	44.50	2.00

8.5 Implementation in AthenaVisual Plus

❖ Open **AthenaVisual Plus**. From the ***File*** menu, choose ***New***. The *Welcome: New Model Selection Panel* window appears.



Select Model

☐

☐ General Form with Initial Value Problems

☐ Estimation with Pure Algebraic Equations

☑ Estimation with Ordinary Differential Equations

☐ Estimation with DAEs with Diagonal E Matrix

☐ Estimation with with Non-Diagonal E Matrix

☐

☐ General Form with Initial-Boundary Value Problems

☐ Estimation with Boundary Value Problems

☐ Estimation with Partial Differential Equations

☐ Estimation with PDEs with Diagonal E Matrix

☐ Estimation with Two-Dimensional PDEs

☐

☐ General Form with User Defined Explicit Models

☐ Estimation with User Defined Explicit Models

☐ Estimation with User Defined Explicit Models

Sample for the Existing Template

Model Equations

$$\frac{du_1}{dt} = -k_1 u_1$$

$$\frac{du_2}{dt} = k_1 u_1 - k_2 u_2$$

$$\frac{du_3}{dt} = k_2 u_2$$

Response Functions

$y_1 = u_1$ **and** $y_2 = u_2$

Create your Model From:

● A Blank Document ○ An Existing Template

❖ Select the **Parameter Estimation** tab

❖ Select the *Estimation with Ordinary Differential Equations* 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.

8.5.1 Writing Source Code for Estimation with Implicit Models

You must enter a minimum of four sections in order to create the parameter estimation model with a system of differential equations. 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 implicit model equations. The third required section labeled **@Response Model** is used to define the experimental responses(observations). The vector **Y(i)** is reserved in Athena to define these responses A section labeled **@Gradient Vector** is optional and may be used to enter the derivatives of the responses with respect to the adjustable parameters. The matrix **dY(i,k)** is reserved in Athena to define these derivatives . The fourth and final section labeled **@Connect Parameters** is used to connect the adjustable parameters and settings with the differential model parameters and constants. A data section not labeled by AthenaVisual Plus may also be used to enter all the data pertinent to the model. The data section may also contain declaration statements for all model variables, parameters and constants. This section, if used, must be the first one in the model. The declaration of the model variables, parameters and constants must be done in accordance the AthenaVisual Plus syntax rules shown below:

8.5.1.1 Declaration of Variables in AthenaVisual Plus

Global Variables: To declare global variables in the AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 an implicit model for parameter estimation in the AthenaVisual Plus environment. The modeling code is NOT case sensitive.

8.5.1.2 Data Section

In the data section the user simply enters the problem data and various constants. The data section also contains the declarations of problem variables, parameters and constants. For our example the user enters the experimental base temperature, as well as problem specific constants as shown in the code below. The Athena interpreter treats any line that begins with an exclamation mark **!** as a comment. It is mandatory and strongly recommended that the users declare all the problem parameters and constants. All variables in Athena are either real double or single precision or integer long. Character and Logical variables are also allowed. The following source code may be entered for this example (If you use the Windows Copy and Paste commands to enter this code into your AthenaVisual Plus project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```
! Parameter Estimation: Conversion of Glucose to Gluconic Acid
!--------------------------------------------------------------
 Global a,b,Temp,Tref,RxnTime As Real
 Global kB(Nrx),EB(Nrx),kRATE(Nrx) As Real

 Parameter Nrx=5 As Integer    ! Number of Chemical Reactions
 Parameter Rg=8.31447 As Real  ! Universal Gas Constant, J/(mol K)

! Constant Data and Parameters
!----------------------------
 a=0.9082
 b=1.011
 Tref=27.8+273.15      ! Reference Temperature, deg K
```

8.5.1.3 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 DDAPLUS to start the integration. 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, **U(1)** represents the concentration of cell, **U(2)** the concentration of gluconolactone, **U(3)** the concentration of gluconic acid and **U(4)** the concentration of glucose. Since the initial conditions change for the experiments at different temperature levels we use the experimental setting to pass these conditions as shown below: To enter 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(1:4)=Xu(4:7)
```

The variable **Xu( )** vector will be discussed later in the **Connect Parameters** and in the Experimental Data tab of the Parameter Estimation Solver Control Panel sections; it is used define the experimental settings.

8.5.1.4 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 example **F(1)** represents the rate of change of the cell concentration, **F(2)** the rate of change of the gluconolactone concentration and so forth. 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
 Dim C1,C2,C3,C4 As Real
 Dim Alpha As Real

 C1=U(1)
 C2=U(2)
 C3=U(3)
 C4=U(4)

 Alpha=1.0-Tref/Temp
 kRATE(1:5)=kB*exp(EB*Alpha)

 F(1)=kRATE(1)*C1*(1.0-C1/kRATE(2))
 F(2)=kRATE(3)*C1*C4/(kRATE(4)+C4)-a*kRATE(5)*C2
 F(3)=kRATE(5)*C2
 F(4)=-b*kRATE(3)*C1*C4/(kRATE(4)+C4)
```

Observe how the reaction rate constants are calculated from the base constants and the activation energies. The variable *Temp* that defines the temperature will be defined later at the **Connect Parameters** section where the experimental settings are passed to the estimator.

8.5.1.5 Response Model

In the Response Model section the user must enter the responses. The vector **Y( )** is reserved to enter the responses. For example **Y(1)** represents the first measured response, **Y(2)** the second (if present) and so on. To enter the responses for your model:

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

```
@Response Model
 Y(1:4)=U(1:4)
```

8.5.1.6 Connect Parameters

In the Connect Parameters section the user must connect the adjustable parameters and experimental settings with the model parameters and constants. The vector **Par( )** is reserved to access the adjustable parameters, and the vector **Xu( )** is reserved to access the experimental settings. As mentioned above the vector **Y( )** is reserved to enter the responses. For example **Y(1)** represents the first measured response, **Y(2)** the second (if present) and so on. As mentioned above, **Xu( )** is reserved for the experimental settings. For example **Xu(1)** represents the first setting, **Xu(2)** the second if present and so on. To enter the Connect Parameters section for our model:

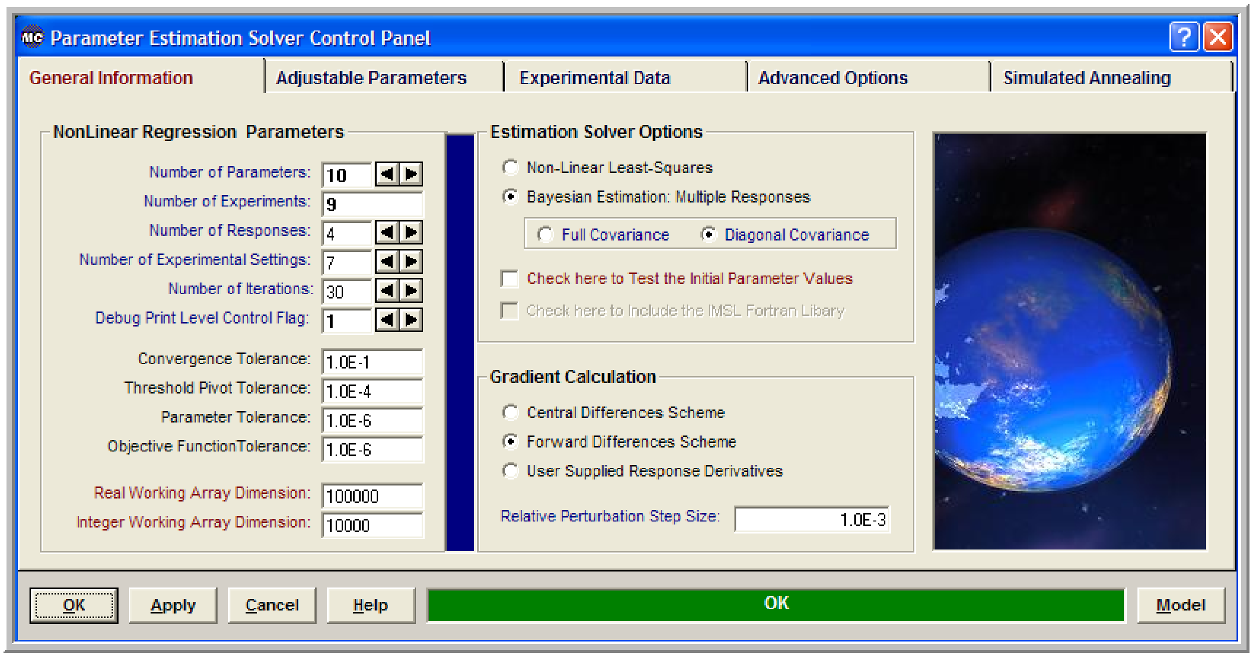
- From the ***Model*** menu choose ***Connect Parameters*** (or **Hit F11**)
- Enter the source code as shown below for our example.

```
@Connect Variables
 RxnTime=Xu(3)          ! Reaction Time, s
 Temp=Xu(2)+273.15      ! Reaction Temperature, deg K

 kB(1:5)=Par(1:5)
 EB(1:5)=Par(6:10)
```

8.5.2 The Parameter Estimation Solver

It is now time to access the AthenaVisual Plus solver for Parameter Estimation in order to enter information about the adjustable parameters, the experimental observations and various other parameters that control the estimation algorithm. To do that:

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



NonLinear Regression Parameters

Number of Parameters: 10
Number of Experiments: 9
Number of Responses: 4
Number of Experimental Settings: 7
Number of Iterations: 30
Debug Print Level Control Flag: 1

Convergence Tolerance: 1.0E-1
Threshold Pivot Tolerance: 1.0E-4
Parameter Tolerance: 1.0E-6
Objective FunctionTolerance: 1.0E-6

Real Working Array Dimension: 100000
Integer Working Array Dimension: 10000

Estimation Solver Options

○ Non-Linear Least-Squares
● Bayesian Estimation: Multiple Responses
○ Full Covariance ● Diagonal Covariance
□ Check here to Test the Initial Parameter Values
□ Check here to Include the IMSL Fortran Libary

Gradient Calculation

○ Central Differences Scheme
● Forward Differences Scheme
○ User Supplied Response Derivatives

Relative Perturbation Step Size: 1.0E-3

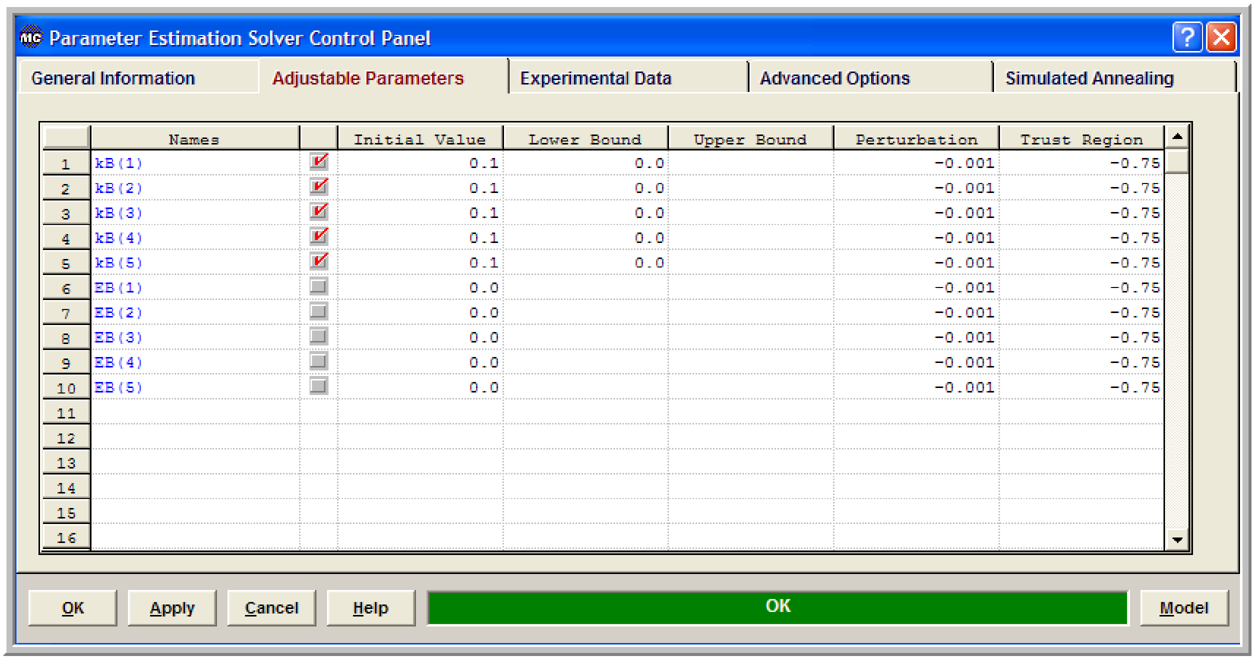
From the **Nonlinear Regression Parameters** group enter:

- ❑ The number of unknown parameters **(10)**
- ❑ The number of experiments or datasets **(**Actual number is **23. I**nsert **9** to estimate only the base reaction rate constants **kB(1:5)** from the first **9** isothermal data as shown in sections below**)**
- ❑ The number of responses or dependent variables **(4)** and
- ❑ The number of experimental settings or predictor variables **(7)**

Optionally you may change the *Number of Iterations*, the *Convergence and Parameter Tolerance*, the *Debug Print Level Control Flag* and the Real and Integer working arrays space requirements. From the **Estimation Solver Options** group choose the type of estimation you are going to be using and optionally request a diagonal covariance if the Bayesian estimation has been chosen, or *Perform a Test Call to Model*. The test call to model is a very useful option since it allows you to see how good is your initial parameter guess before you proceed with the estimation process. You also have the option to use a relative weighting factor of the observation vector in the estimation process. From the **Derivatives Calculation** group choose the method for the objective function gradient calculation and optionally enter the *Relative Perturbation Step Size*. Should you choose User Supplied Model Derivatives you must enter the section **Gradient Vector** when you enter the source code.

8.5.2.1 The Adjustable Parameters

Next select the **Adjustable Parameters** tab on the Parameter Estimation Control Panel and enter the names of the parameters you wish to estimate, their initial guesses and optionally their lower and upper bounds, the relative perturbation step size for gradient computation and the size of the trust region. Notice that if a parameter is not checked in the check box next to it, will remain fixed at its initial value during the estimation. The following data have been inserted for our example:

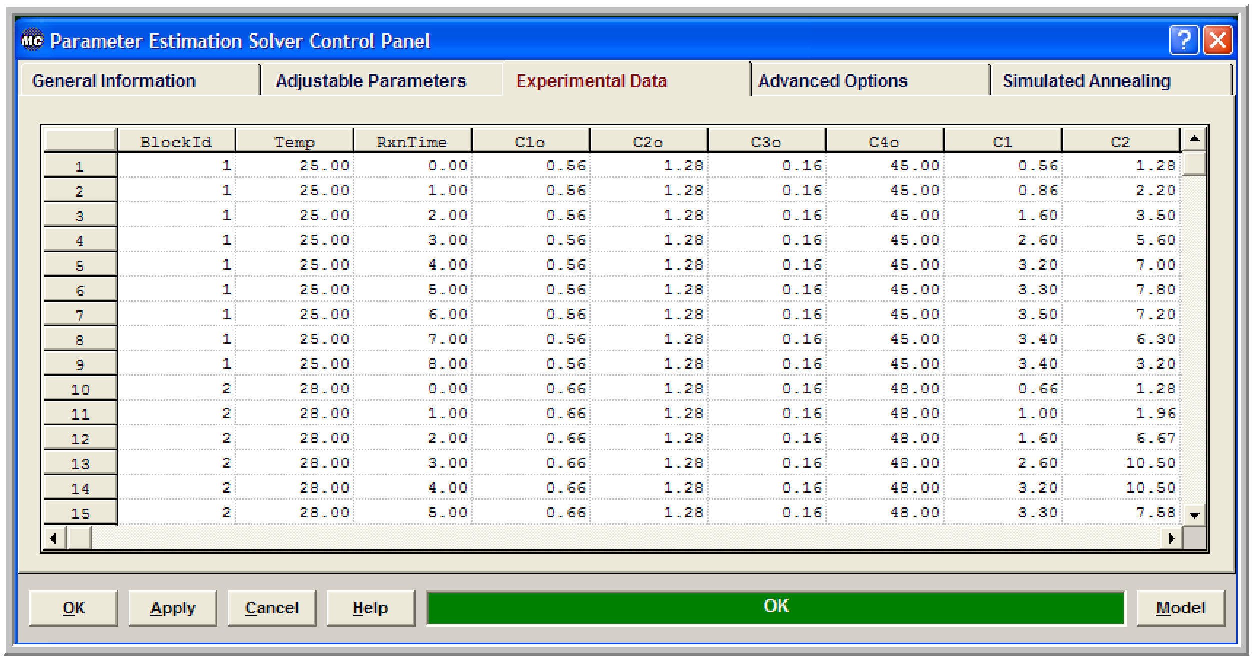


	Names		Initial Value	Lower Bound	Upper Bound	Perturbation	Trust Region
1	kB(1)	☑	0.1	0.0		-0.001	-0.75
2	kB(2)	☑	0.1	0.0		-0.001	-0.75
3	kB(3)	☑	0.1	0.0		-0.001	-0.75
4	kB(4)	☑	0.1	0.0		-0.001	-0.75
5	kB(5)	☑	0.1	0.0		-0.001	-0.75
6	EB(1)	□	0.0			-0.001	-0.75
7	EB(2)	□	0.0			-0.001	-0.75
8	EB(3)	□	0.0			-0.001	-0.75
9	EB(4)	□	0.0			-0.001	-0.75
10	EB(5)	□	0.0			-0.001	-0.75
11							
12							
13							
14							
15							
16							

Important Note: Observe that we first wish to estimate only the base values of the reaction rate constants **kB(1:5)** by fixing the activation energy values to zero. For this step we use only the first **9** experiments from a database of **23** experiments. This process is called *staged parameter estimation*. The idea is to estimate the parameters sequentially in the presence of data at different temperature levels. Once the base reaction rate constants have been estimated we can update the solution vector and proceed to estimate the activation energies by bringing in the experiments at all temperature levels.

8.5.2.2 The Experimental Data

Next select the **Experimental Data** tab on the Parameter Estimation Control Panel and enter the observations and experimental settings. Optionally you may enter weights for each observation. Otherwise these weights are set equal to one by the Parameter Estimation solver. Please keep in mind that Windows *Copy* and *Paste* functions can be used to transfer your data from, say, an Excel spreadsheet. The following data have been inserted for our example: (partial list shown)



	BlockId	Temp	RxnTime	C1o	C2o	C3o	C4o	C1	C2
1	1	25.00	0.00	0.56	1.28	0.16	45.00	0.56	1.28
2	1	25.00	1.00	0.56	1.28	0.16	45.00	0.86	2.20
3	1	25.00	2.00	0.56	1.28	0.16	45.00	1.60	3.50
4	1	25.00	3.00	0.56	1.28	0.16	45.00	2.60	5.60
5	1	25.00	4.00	0.56	1.28	0.16	45.00	3.20	7.00
6	1	25.00	5.00	0.56	1.28	0.16	45.00	3.30	7.80
7	1	25.00	6.00	0.56	1.28	0.16	45.00	3.50	7.20
8	1	25.00	7.00	0.56	1.28	0.16	45.00	3.40	6.30
9	1	25.00	8.00	0.56	1.28	0.16	45.00	3.40	3.20
10	2	28.00	0.00	0.66	1.28	0.16	48.00	0.66	1.28
11	2	28.00	1.00	0.66	1.28	0.16	48.00	1.00	1.96
12	2	28.00	2.00	0.66	1.28	0.16	48.00	1.60	6.67
13	2	28.00	3.00	0.66	1.28	0.16	48.00	2.60	10.50
14	2	28.00	4.00	0.66	1.28	0.16	48.00	3.20	10.50
15	2	28.00	5.00	0.66	1.28	0.16	48.00	3.30	7.58

Finally you may wish to specify the replicate experiments(if any) for performing lack-of-fit analysis and model discrimination. The first seven columns of the Experimental Data correspond to the variable **Xu(1:7)**; specifically **Xu(1)** contains the Block ID (not used in the estimation), **Xu(2)** contains the data for the reaction temperature, **Xu(3)** the data for the reaction time and **Xu(4:7)** contain the initial conditions for the experiments.

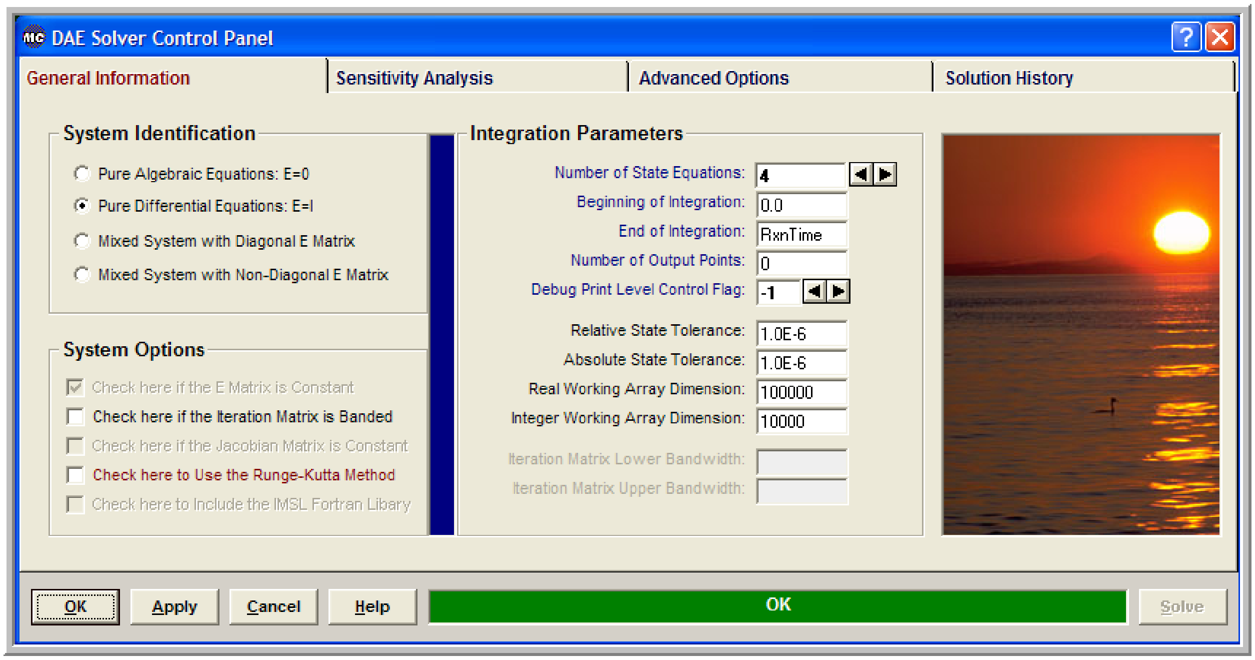
Important Note: You may notice the description of the headers of the experimental data. An easy way to enter the headers is by using the keyword Headers in the Solver Options of your code as indicated below: To enter the Connect Parameters section for our model:

- From the ***Model*** menu choose ***Solver Options***
- Enter the source code as shown below for our example.

```
@Solver Options
 Headers=BlockId;Temp;RxnTime;C1o;C2o;C3o;C4o;C1;C2;C3;C4;w1;w2;w3;w4;Replicate
```

8.5.2.3 The Kinetic Model

Now click the **Model** control button to enter the information on the differential equations for the kinetic model. The ***DAE Solver Control Panel*** window will appear:



System Identification

○ Pure Algebraic Equations: E=0
● Pure Differential Equations: E=I
○ Mixed System with Diagonal E Matrix
○ Mixed System with Non-Diagonal E Matrix

System Options

☑ Check here if the E Matrix is Constant
□ Check here if the Iteration Matrix is Banded
□ Check here if the Jacobian Matrix is Constant
□ Check here to Use the Runge-Kutta Method
□ Check here to Include the IMSL Fortran Libary

Integration Parameters

Number of State Equations:	4
Beginning of Integration:	0.0
End of Integration:	RxnTime
Number of Output Points:	0
Debug Print Level Control Flag:	-1
Relative State Tolerance:	1.0E-6
Absolute State Tolerance:	1.0E-6
Real Working Array Dimension:	100000
Integer Working Array Dimension:	10000
Iteration Matrix Lower Bandwidth:	
Iteration Matrix Upper Bandwidth:	

In the **System Identification** group you will see that the option *Pure Differential Equations E=I* has already been selected for you. From the **Integration Parameters** group enter the *Number of State Equations* the *Beginning* and *End of Integration* (notice that the End of the Integration is equal **to RxnTime=Xu(3)**, the third experimental setting), set *Number of Output Points* equal to zero and change *Debug Print Level Control Flag* to –1 to suppress the printing from the DAE solver during the estimation process. Optionally you may change 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 has a banded structure in which case you will have to check the appropriate options in the **System Options** group. After you make all your selections click **OK.** When the Parameter Estimation control panel appears click **OK** and proceed to save and run your model.

8.5.3 Saving and Running your Model

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***. This will save your model and create the Fortran code that will access the Parameter Estimation solver. The ***Save As*** dialog box appears.
- ➢ In the Directories box, double-click a directory where you want to store the source file (or down a directories path to the appropriate directory.)
- ➢ Type a filename (a filename cannot contain the following characters: \ / : * ? “ < > |) in the File Name box, then choose OK. The default extension given to a file is **AVW**.
- ➢ To view the Fortran code that you have just created from the ***View*** menu choose ***Fortran Code***.

New files are labeled UNTITLED until they are saved. The maximum number of characters in a line is 132; the maximum number of lines in a file is infinity. Before you can save or close a window it must be active. To make a window active, either switch to the window (by clicking anywhere in it) or choose the window name or number from the Window menu.

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**)

8.5.4 Numerical Results for Base Reaction Rate Constants

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:

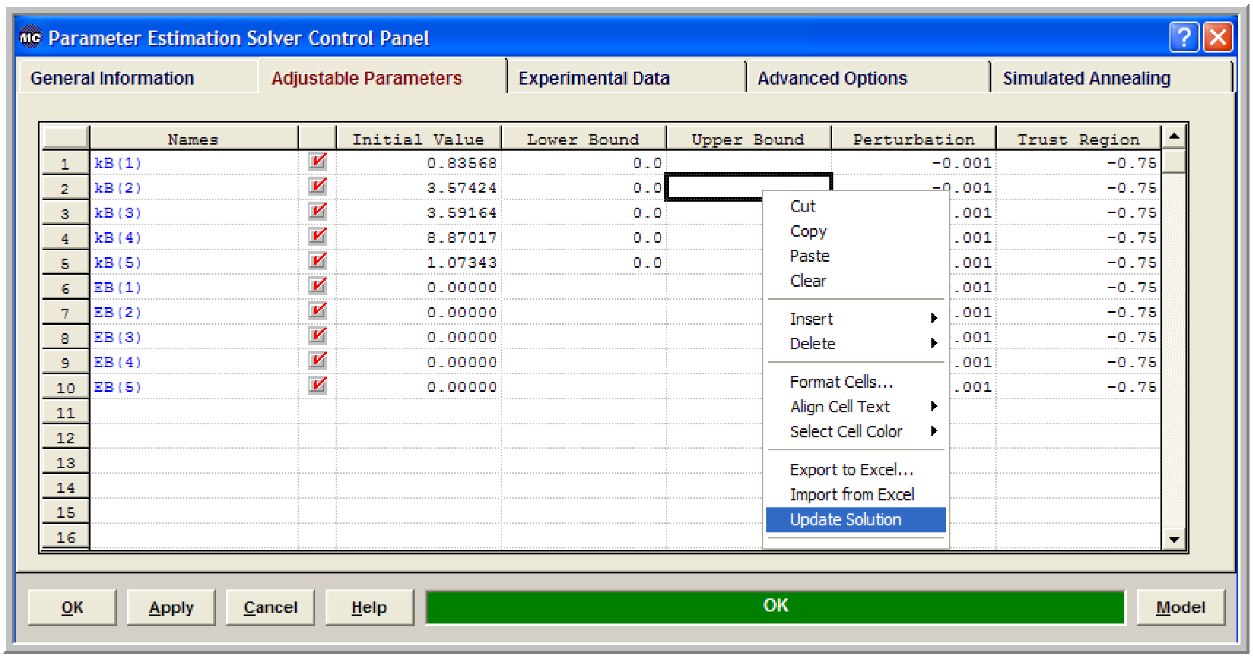
```
Number of Experiments..........................    9
Number of Parameters...........................   10
Number of Responses............................    4
Number of Settings.............................    7

EXIT GREGPLUS: SOLUTION FOUND. MODEL No. 1

STATISTICAL ANALYSIS

OBJECTIVE FUNCTION.......................  4.26509E+01
SUM OF SQUARES OF RESIDUALS..............  2.31027E+02
SUM OF SQUARES OF WEIGHTED RESIDUALS.....  2.31027E+02
ESTIMATED PARAMETERS FROM DATA...........            5
TOTAL OBSERVATIONS DATA COUNT IS.........           36

              OPTIMAL
PARAMETER    ESTIMATES      95% MARGINAL HPD INTERVALS      PARAMETER STATUS
 PAR(  1)   8.356830E-01     8.356830E-01 +- 8.132E-02      Estimated
 PAR(  2)   3.574236E+00     3.574236E+00 +- 1.552E-01      Estimated
 PAR(  3)   3.591635E+00     3.591635E+00 +- 4.073E-01      Estimated
 PAR(  4)   8.870168E+00     8.870168E+00 +- 2.928E+00      Estimated
 PAR(  5)   1.073433E+00     1.073433E+00 +- 3.895E-02      Estimated
 PAR(  6)   0.000000E+00                                    Fixed
 PAR(  7)   0.000000E+00                                    Fixed
 PAR(  8)   0.000000E+00                                    Fixed
 PAR(  9)   0.000000E+00                                    Fixed
 PAR( 10)   0.000000E+00                                    Fixed

NORMALIZED PARAMETER COVARIANCE MATRIX
  1.000
 -0.647  1.000
 -0.812  0.483  1.000
 -0.646  0.552  0.932  1.000
  0.040 -0.019 -0.229 -0.308  1.000
  0.000 -0.000 -0.000 -0.000  0.000  0.000
  0.000 -0.000 -0.000 -0.000  0.000  0.000  0.000
  0.000 -0.000 -0.000 -0.000  0.000  0.000  0.000  0.000
  0.000 -0.000 -0.000 -0.000  0.000  0.000  0.000  0.000  0.000
  0.000 -0.000 -0.000 -0.000  0.000  0.000  0.000  0.000  0.000  0.000
```

Only partial results are shown here. Observe from the data above that the parameters 6 though 8 are indicated as **Fixed** at their initial value equal to zero.

8.5.5 Estimation of the Activation Energy Constants

We are now ready to proceed with the estimation of the activation energy values and update the base values of the reaction rate constants by using the full set of the experimental data. To do that first we bring up the parameter estimation solver control panel **(Hit F12)**:

We select the **Adjustable Parameters** tab and then we perform a right mouse click and select ***Update Solution***. With this action the reaction rate base constants will be set to their optimal values. We then click the parameters that correspond to the activation energies so that they can participate in the estimation. When we are done the estimation solver panel looks as follow:



	Names		Initial Value	Lower Bound	Upper Bound	Perturbation	Trust Region
1	kB(1)	☑	0.83568	0.0		-0.001	-0.75
2	kB(2)	☑	3.57424	0.0		-0.001	-0.75
3	kB(3)	☑	3.59164	0.0		.001	-0.75
4	kB(4)	☑	8.87017	0.0		.001	-0.75
5	kB(5)	☑	1.07343	0.0		.001	-0.75
6	EB(1)	☑	0.00000			.001	-0.75
7	EB(2)	☑	0.00000			.001	-0.75
8	EB(3)	☑	0.00000			.001	-0.75
9	EB(4)	☑	0.00000			.001	-0.75
10	EB(5)	☑	0.00000			.001	-0.75
11							
12							
13							
14							
15							
16							

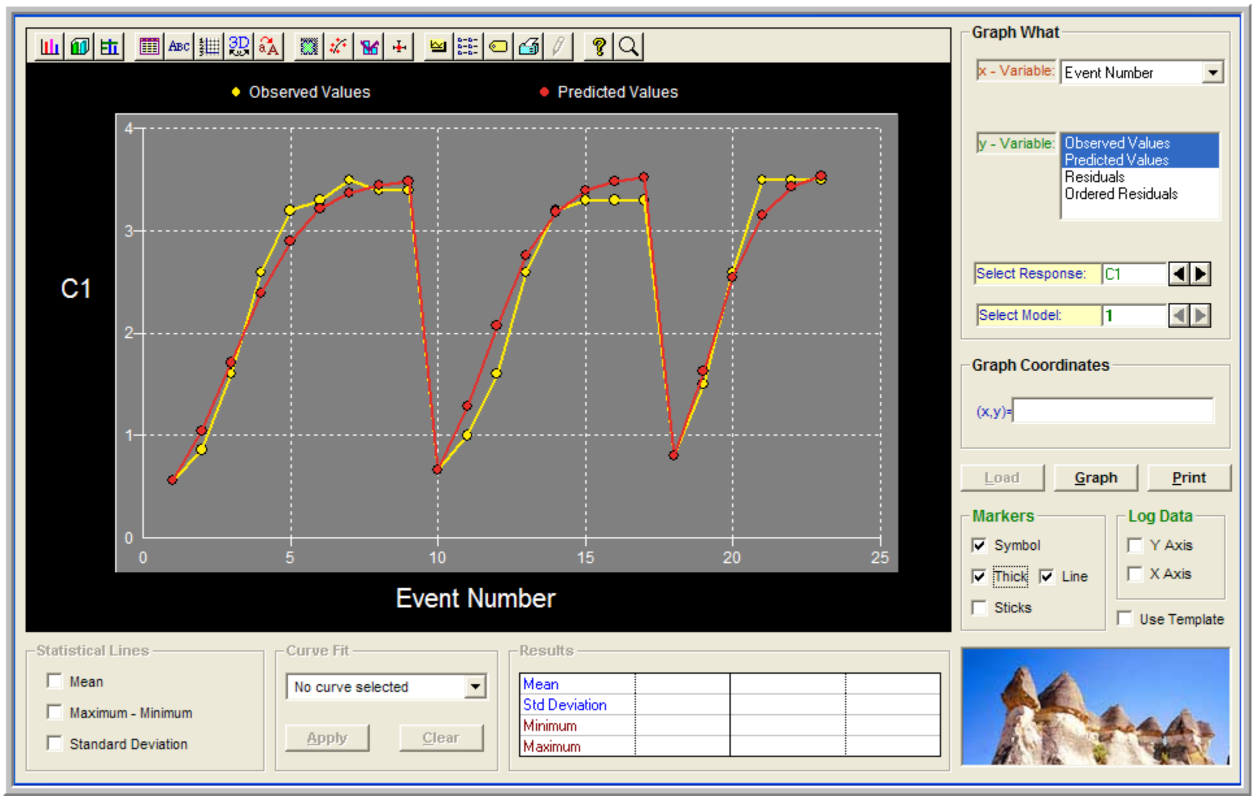
We are almost ready to perform the full parameter estimation. Before choosing **OK** to unload the Parameter Estimation Control Panel, we select the **General Information** tab and enter the full number of experiments (in our case **23**). Then click **OK** and run the estimation. The following results appear (only partial list is shown here):

```
Number of Experiments.........................    23
Number of Parameters..........................    10
Number of Responses...........................     4
Number of Settings............................     7

EXIT GREGPLUS: SOLUTION FOUND. MODEL No. 1

STATISTICAL ANALYSIS

OBJECTIVE FUNCTION.......................  2.85824E+02
SUM OF SQUARES OF RESIDUALS..............  3.06309E+02
SUM OF SQUARES OF WEIGHTED RESIDUALS.....  3.06309E+02
ESTIMATED PARAMETERS FROM DATA...........           10
TOTAL OBSERVATIONS DATA COUNT IS.........           92
```

```
                 OPTIMAL
PARAMETER       ESTIMATES        95% MARGINAL HPD INTERVALS        PARAMETER STATUS
 PAR(  1)     9.034596E-01      9.034596E-01 +- 8.633E-02        Estimated
 PAR(  2)     3.549381E+00      3.549381E+00 +- 1.451E-01        Estimated
 PAR(  3)     5.237616E+00      5.237616E+00 +- 6.918E-01        Estimated
 PAR(  4)     1.167732E+01      1.167732E+01 +- 4.182E+00        Estimated
 PAR(  5)     1.232811E+00      1.232811E+00 +- 1.035E-01        Estimated
 PAR(  6)     1.201063E+01      1.201063E+01 +- 1.081E+01        Estimated
 PAR(  7)     1.097267E+00      1.097267E+00 +- 4.667E+00        Estimated
 PAR(  8)     3.445240E+01      3.445240E+01 +- 1.402E+01        Estimated
 PAR(  9)     1.072950E+01      1.072950E+01 +- 3.660E+01        Estimated
 PAR( 10)     2.358907E+01      2.358907E+01 +- 9.179E+00        Estimated

NORMALIZED PARAMETER COVARIANCE MATRIX
  1.000
 -0.662  1.000
 -0.616  0.403  1.000
 -0.403  0.379  0.943  1.000
  0.012 -0.001 -0.139 -0.158  1.000
  0.133 -0.085 -0.008  0.054  0.009  1.000
 -0.100  0.059  0.026 -0.033 -0.006 -0.691  1.000
  0.023 -0.008  0.010 -0.039  0.014 -0.596  0.404  1.000
  0.094 -0.071 -0.056 -0.102  0.037 -0.362  0.352  0.936  1.000
 -0.000 -0.002  0.003  0.021 -0.058  0.004  0.002 -0.144 -0.159  1.000

EVENT      OBSERVED         PREDICTED         RESIDUAL
  1       5.6000E-01       5.6000E-01       0.0000E+00
  2       8.6000E-01       1.0471E+00      -1.8710E-01
  3       1.6000E+00       1.7124E+00      -1.1236E-01
  4       2.6000E+00       2.3908E+00       2.0919E-01
  5       3.2000E+00       2.9049E+00       2.9512E-01
  6       3.3000E+00       3.2135E+00       8.6506E-02
  7       3.5000E+00       3.3736E+00       1.2642E-01
  8       3.4000E+00       3.4504E+00      -5.0355E-02
  9       3.4000E+00       3.4858E+00      -8.5786E-02
 10       6.6000E-01       6.6000E-01       0.0000E+00
 11       1.0000E+00       1.2856E+00      -2.8564E-01
 12       1.6000E+00       2.0782E+00      -4.7822E-01
 13       2.6000E+00       2.7637E+00      -1.6366E-01
 14       3.2000E+00       3.1865E+00       1.3500E-02
 15       3.3000E+00       3.3955E+00      -9.5509E-02
 16       3.3000E+00       3.4875E+00      -1.8754E-01
 17       3.3000E+00       3.5260E+00      -2.2599E-01
 18       8.0000E-01       8.0000E-01       0.0000E+00
 19       1.5000E+00       1.6338E+00      -1.3379E-01
 20       2.6000E+00       2.5467E+00       5.3318E-02
 21       3.5000E+00       3.1525E+00       3.4748E-01
 22       3.5000E+00       3.4335E+00       6.6550E-02
 23       3.5000E+00       3.5420E+00      -4.2009E-02

NUMBER OF ITERATIONS..............    8
NUMBER OF FUNCTION CALLS..........  110
```

8.5.6 Graphical Results

If you wish to see various graphs of the estimation process from the ***View*** menu choose ***Solution Graphs***, or click . The AthenaVisual Plus graphics control panel appears:



Graph What

x - Variable: Event Number

y - Variable: Observed Values, Predicted Values, Residuals, Ordered Residuals

Select Response: C1

Select Model: 1

Graph Coordinates

(x,y)=

Load Graph Print

Markers: ☑ Symbol ☑ Thick ☑ Line □ Sticks

Log Data: □ Y Axis □ X Axis □ Use Template

Statistical Lines: □ Mean □ Maximum - Minimum □ Standard Deviation

Curve Fit: No curve selected

Apply Clear

Results			
Mean			
Std Deviation			
Minimum			
Maximum			

In this window fist we click ***Load*** to load the numerical results. Then in the **Graph What** group we select the **x-variable** (here **Event Number**) and the **y-variables** (here **Observed** and **Predicted Values**) and click ***Graph***. You should see the monitoring 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 other graphs, like plotting of residuals, plotting observed vs. predicted values, or against any selected settings.

9.0 Model Discrimination and Optimal Design

Test patterns for models linear in the parameters can be planned in advance, and are highly developed for polynomial models. Models nonlinear in parameters can be studied more economically in a sequential fashion, once an exploratory test pattern has been run. Computer simulations based on the current information can then be used to select additional test conditions from a set of feasible alternatives. The resulting gains of efficiency can be substantial, once the ultimate model is in the trial set.

9.1 Discrimination and Design Criteria

There is a risk in starting model-based planning too soon. These strategies concentrate the selected tests in regions dictated by the models that are known, and thus may exclude experiments that could lead to better models. To minimize this risk, one could use a regular grid of candidate experiments, and stipulate that no candidate can be re-elected. Hybrid strategies are also possible, in which the set of candidate experiments is varied as the work proceeds.

The selection strategy should be tailored to the goals of the investigation. Many workers choose one or more of the following goals:

- ❖ Optimal estimation of parameters.
- ❖ Optimal estimation of functions.
- ❖ Selection of a preferred model from a given set. This process is known as *discrimination*.
- ❖ Finding good models and then choosing the best.

Once a goal is defined, a strategy can be chosen. Some strategies for particular goals are described below.

The parameter estimates are determined by the posterior density function, which we approximated via the Hessian matrix or its inverse. The expected values of these matrices after any further observations are computable after appending simulations of those observations to the data set. The desired criterion of estimation can be calculated for each experimental plan, and the optimal plan can then be selected.

Determinant criteria aim to minimize the volume of a highest-posterior-density region in parameter space. Experimental designs selected in this way are called *D-optimal*; the literature on them is extensive.

Two *shape criteria* for the HPD region have been proposed for planning experiments for parameter estimation. Hosten (1974) advocated selection to maximize the smallest eigenvalue of ***A***, thus giving the HPD region a rounder shape. Pritchard and Bacon (1975) selected the experiments to reduce the correlations between the parameter estimates. Reparameterization reduces the correlations more effectively as shown by Agarwal and Brisk (1985).
A *trace criterion* for reducing the HPD region was introduced by Pinto *et al*. (1990). Here one selects the experiments to minimize the trace of the inverse Hessian matrix. This quantity is the sum of the eigenvalues and is proportional to the sum of the variances of the parameter estimates. This criterion is simpler than Hosten's, and is preferable because it includes every eigenvalue. Finally, the trace criterion is readily applied to any subset θ_a of the parameters by including just the corresponding diagonal terms of A^{ee} in the trace calculation.

The determinant criterion and the trace criterion are well regarded, and both are provided as options in GREGPLUS. Procedures and results for the determinant criterion in least squares are included in the GREGPLUS user instructions.

9.2 References

M. Caracotsios, *Model Parametric Sensitivity Analysis and Nonlinear Parameter Estimation. Theory and Applications.* Ph. D. thesis, Dept. of Chemical Engineering, University of Wisconsin, Madison (1986).

W. E. Stewart, M. Caracotsios, and J. P. Sorensen, *Parameter estimation from multiresponse data*, AIChE J., 38, 641-650 (1992); Errata, 38, 1302 (1992).

W. E. Stewart, and M. Caracotsios, *Computer-Aided Modeling of Reactive Systems*, John Wiley and Sons, Inc. (2008).

J. C. Pinto, M. W. Lobao, and J. L. Monteiro, Sequential experimental design for parameter estimation: analysis of relative deviations, *Chem. Eng. Sci*., **46**, 3129--3138 (1991).

L. H. Hosten, A sequential experimental procedure for precise parameter estimation based on the shape of the joint confidence region, *Chem. Eng. Sci*., **29**, 2247--2252 (1974).

A. K. Agarwal, and M. L. Brisk, Sequential experimental design for precise parameter estimation. 1. Use of Reparameterization. 2. Design criteria, *Ind. Eng. Chem Process Des. Dev*., **24**, 203--207, 207--210 (1985).

D. J. Pritchard, and D. W. Bacon, Prospects for reducing correlations among parameter estimates in kinetic models, *Chem. Eng. Sci*., **33**, 1539--1543 (1978).

9.3 Model Discrimination with an Explicit Model

- Start **Athena Visual Studio**
- From the **File** menu select **New.**
- You are in the **Process Modeling** tab.
- Select the **Parameter Estimation** tab.
- Click **Estimation with User Defined Explicit Models.**
- 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.**

Nonlinear Parameter estimation and model discrimination problems with user defined explicit models must be self-contained. Specifically they should not rely on the solvers that are in AthenaVisual Plus. The user however, may make calls to the available math and engineering subroutine as well as user defined procedures. This test problem has been design to illustrate the functionality of AthenaVisual Plus in dealing with single response parameter estimation and model discrimination with user explicit models.

This example problem has been created to test the functionality of AthenaVisual Plus in dealing with parameter estimation and model discrimination and criticism from single-response data with explicit mechanistic models.

9.4 Model Discrimination in CH_3OH Chemistry

The Methanol production chemistry can be represented by the simple chemical reaction:

$$CO + 2H_2 \xrightarrow{\Re} CH_3OH$$

The following two models have been postulated as plausible candidates to describe the observed reaction rate of the product methanol based on the available experimental data:

$$\Re_1 = \frac{k(T)K_1K_2^2\left(P_CP_H^2 - P_M / K_{\text{eq}}\right)}{\left(1 + K_1P_C + K_2P_H + K_3P_M\right)^3} \quad \text{and} \quad \Re_2 = k(T)P_C^{n_1}P_H^{n_2}P_M^{n_3}$$

$$k = \exp\left(\ln k_B + \frac{E}{RT_B}\left(1 - \frac{T_B}{T}\right)\right)$$

$$\log K_{\text{eq}} = \frac{3914}{T} - 7.536\ \log T + 0.001766 \log T + 9.388$$

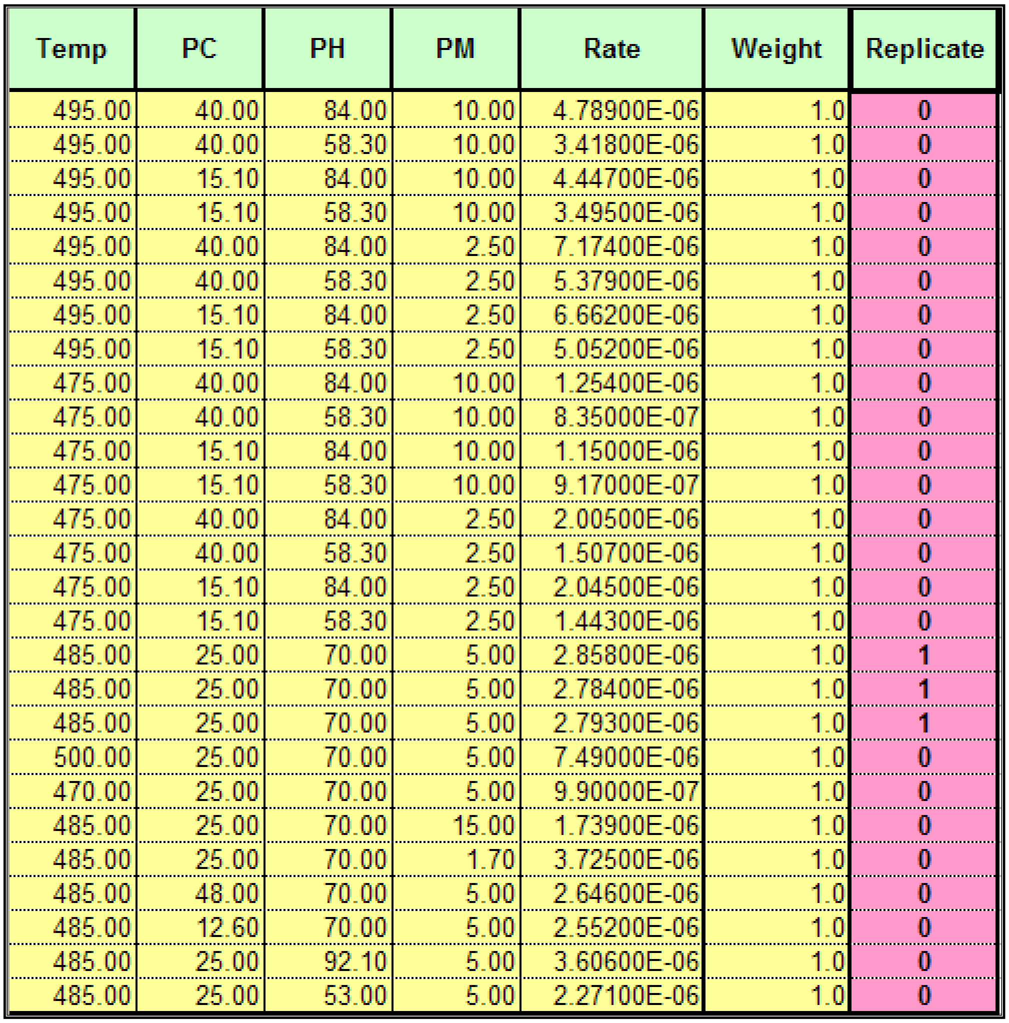
- Estimate the parameters $\{k_B, E, K_1, K_2, K_3, n_1, n_2, n_3\}$
- Determine which of the models fits the data better
- Perform a Lack-of-Fit or Goodness-of-Fit Analysis to assess the validity of the best model
- Propose additional experiments to improve the estimation of the parameters in the range,

$$\begin{aligned} 10 &\le P_C \le 50 \\ 50 &\le P_H \le 100 \\ 0 &\le P_M \le 15 \\ 470 &\le T \le 500 \end{aligned}$$

The parameter initial guess for this example is given in the table below:

Initial Values of Parameters		
$\ln k_B = -5.0$	$K_1 = 0.1$	$n_1 = 0$
$\dfrac{E}{RT_B} = 10.0$	$K_2 = 0.1$	$n_2 = 0$
$T_B = 485.0$	$K_3 = 0.1$	$n_3 = 0$

The experimental data for this example are given in the Table below:

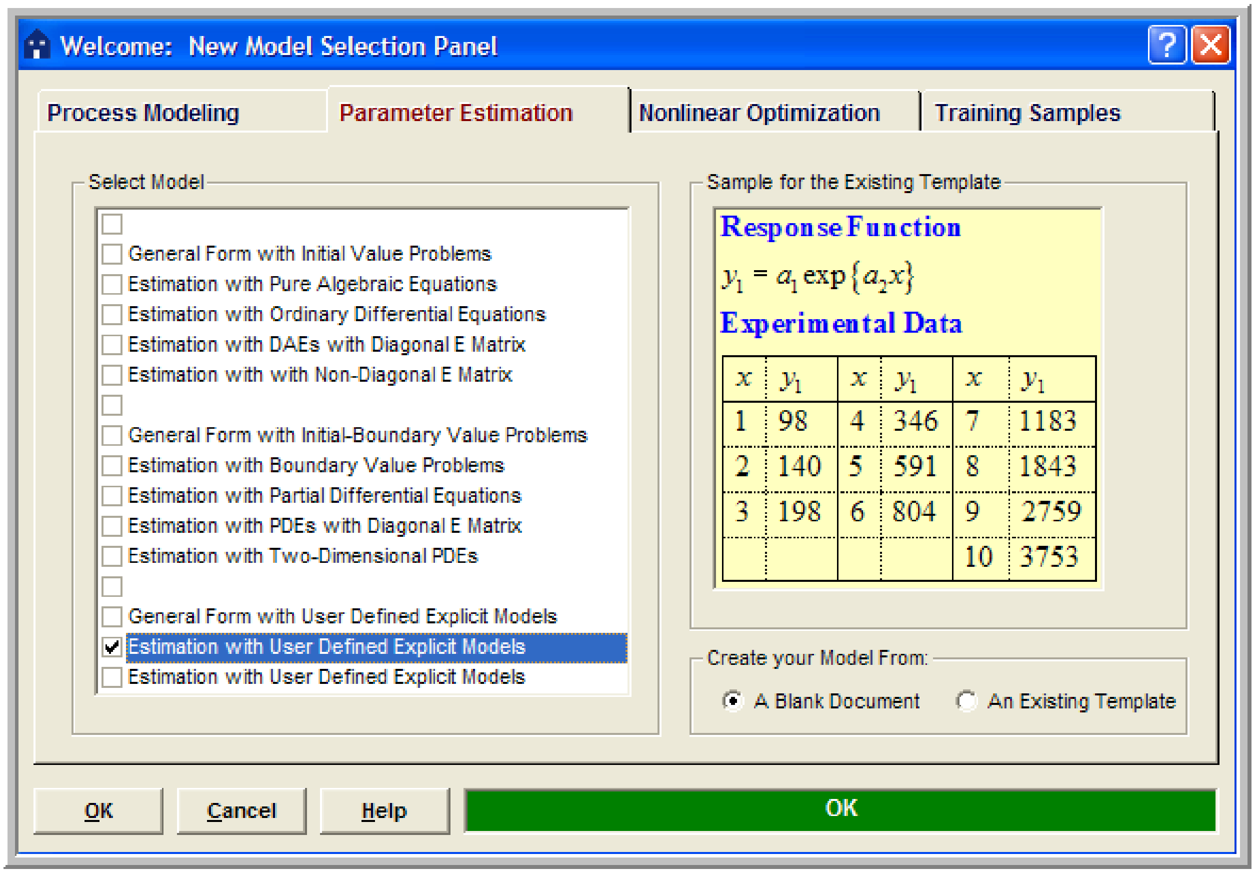


Temp	PC	PH	PM	Rate	Weight	Replicate
495.00	40.00	84.00	10.00	4.78900E-06	1.0	0
495.00	40.00	58.30	10.00	3.41800E-06	1.0	0
495.00	15.10	84.00	10.00	4.44700E-06	1.0	0
495.00	15.10	58.30	10.00	3.49500E-06	1.0	0
495.00	40.00	84.00	2.50	7.17400E-06	1.0	0
495.00	40.00	58.30	2.50	5.37900E-06	1.0	0
495.00	15.10	84.00	2.50	6.66200E-06	1.0	0
495.00	15.10	58.30	2.50	5.05200E-06	1.0	0
475.00	40.00	84.00	10.00	1.25400E-06	1.0	0
475.00	40.00	58.30	10.00	8.35000E-07	1.0	0
475.00	15.10	84.00	10.00	1.15000E-06	1.0	0
475.00	15.10	58.30	10.00	9.17000E-07	1.0	0
475.00	40.00	84.00	2.50	2.00500E-06	1.0	0
475.00	40.00	58.30	2.50	1.50700E-06	1.0	0
475.00	15.10	84.00	2.50	2.04500E-06	1.0	0
475.00	15.10	58.30	2.50	1.44300E-06	1.0	0
485.00	25.00	70.00	5.00	2.85800E-06	1.0	1
485.00	25.00	70.00	5.00	2.78400E-06	1.0	1
485.00	25.00	70.00	5.00	2.79300E-06	1.0	1
500.00	25.00	70.00	5.00	7.49000E-06	1.0	0
470.00	25.00	70.00	5.00	9.90000E-07	1.0	0
485.00	25.00	70.00	15.00	1.73900E-06	1.0	0
485.00	25.00	70.00	1.70	3.72500E-06	1.0	0
485.00	48.00	70.00	5.00	2.64600E-06	1.0	0
485.00	12.60	70.00	5.00	2.55200E-06	1.0	0
485.00	25.00	92.10	5.00	3.60600E-06	1.0	0
485.00	25.00	53.00	5.00	2.27100E-06	1.0	0

9.5 Implementation in AthenaVisual Plus

The following step by step process describes the model implementation in AthenaVisual Plus

- Open **AthenaVisual Plus**. From the ***File*** menu, choose ***New***. The *Welcome: New Model Selection Panel* window appears.



Select Model

□
□ General Form with Initial Value Problems
□ Estimation with Pure Algebraic Equations
□ Estimation with Ordinary Differential Equations
□ Estimation with DAEs with Diagonal E Matrix
□ Estimation with with Non-Diagonal E Matrix
□
□ General Form with Initial-Boundary Value Problems
□ Estimation with Boundary Value Problems
□ Estimation with Partial Differential Equations
□ Estimation with PDEs with Diagonal E Matrix
□ Estimation with Two-Dimensional PDEs
□
□ General Form with User Defined Explicit Models
☑ Estimation with User Defined Explicit Models
□ Estimation with User Defined Explicit Models

Sample for the Existing Template

Response Function

$y_1 = a_1 \exp\{a_2 x\}$

Experimental Data

x	y_1	x	y_1	x	y_1
1	98	4	346	7	1183
2	140	5	591	8	1843
3	198	6	804	9	2759
				10	3753

Create your Model From:

● A Blank Document ○ An Existing Template

- Select the **Parameter Estimation** tab
- Select the *Estimation with User Defined Explicit Models* 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.

9.5.1 Writing Source Code for Estimation with Explicit Models

You must enter a minimum of one section in order to create a parameter estimation model with a user defined (or explicit) stand-alone model. This section labeled **@Response Model** is used to define the experimental responses (observations). The vector **Y( )** is reserved in Athena to define these responses. A section labeled **@Gradient Vector** is optional and may be used to enter the derivatives of the responses with respect to the adjustable parameters. The matrix **dY(i,k)** is reserved in Athena to define these derivatives. A data section not labeled by AthenaVisual Plus may also be used to enter all the data pertinent to the model. The data section may also contain declaration statements for all model variables, parameters and constants. This section, if used, must be the first one in the model. The declaration of the model variables, parameters and constants must be done in accordance the AthenaVisual Plus syntax rules shown below:. In addition you must enter information about the experimental observations and the parameters you are about to estimate.

9.5.1.1 Declaration of Variables in AthenaVisual Plus

Global Variables: To declare global variables in the AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 AthenaVisual Plus 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 an implicit model for parameter estimation in the AthenaVisual Plus environment. The modeling code is NOT case sensitive.

9.5.1.2 Data Section

In the data section the user simply enters the problem data and various constants. The data section also contains the declarations of problem variables, parameters and constants. For our example the user enters the experimental base temperature and the universal gas constant as shown in the code below. The Athena interpreter treats any line that begins with an exclamation mark **!** as a comment. It is mandatory and strongly recommended that the users declare all the problem parameters and constants. All variables in Athena are either real double or single precision or integer long. Character and Logical variables are also allowed. The following source code may be entered for this example (If you use the Windows Copy and Paste commands to enter this code into your AthenaVisual Plus project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```
! Model Discrimination and Lack-of-Fit Analysis
!----------------------------------------------
 Global k,K1,K2,K3,KEQ As Real
 Global n1,n2,n3 As Real
 Global Temp,PC,PH,PM As Real
 Global Tb,Rg As Real

 Tb=485.0  ! Base Temperature, K
 Rg=8.314  ! Universal Gas Constant, J/mol K

 If(iModel==1)Then
  CHMAX(6:8)=0.0
 ElseIf(iModel==2)Then
  CHMAX(3:5)=0.0
 EndIf
```

Here in this data section we have introduced two new Athena variables called **iModel** and **CHMAX**. The **iModel** variable is used as a counter for the number of models that we wish to process. In our example **iModel** takes the values 1 and 2, since we are investigating two models. We order the full set of parameters as it is indicated here: $\{k_B, E, K_1, K_2, K_3, n_1, n_2, n_3\}$. This order implies that for Model #1, parameters 6 through 8 do not participate in the estimation process, while for Model #2 parameters 3 through 5 do not participate in the estimation process. The variable **CHMAX** which is a vector is used to specify at run time if a parameter participates or not in the estimation process. For instance, a value **CHMAX(3)=0** indicates that parameter 3 remains fixed at its initial value during the estimation process.

9.5.1.3 Response Model

In the Response Model Equations section the user must enter the responses. The vector **Par( )** is reserved to access the adjustable parameters, and the vector **Xu( )** is reserved to access the experimental settings. As mentioned above the vector **Y( )** is reserved to enter the responses. For example **Y(1)** represents the first measured response, **Y(2)** the second (if present) and so on. As mentioned above, **Xu( )** is reserved for the experimental settings. For example **Xu(1)** represents the first setting, **Xu(2)** the second if present and so on. To enter the responses for your model:

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

```
@Response Model
 Temp=Xu(1)
 PC=Xu(2)
 PH=Xu(3)
 PM=Xu(4)

 k=exp(Par(1)+Par(2)*(1.0-Tb/Temp))
 K1=Par(3)
 K2=Par(4)
 K3=Par(5)

 n1=Par(6)
 n2=Par(7)
 n3=Par(8)

 KEQ=10.0^(3914.0/Temp-7.536*log10(Temp)+0.001766*Temp+9.388)

 If(iModel==1)Then
  Y(1)=k*K1*K2^2*(PC*PH^2-PM/KEQ)/(1+K1*PC+K2*PH+K3*PM)^3
 ElseIf(iModel==2)Then
  Y(1)=k*PC^n1*PH^n2*PM^n3
 EndIf
```

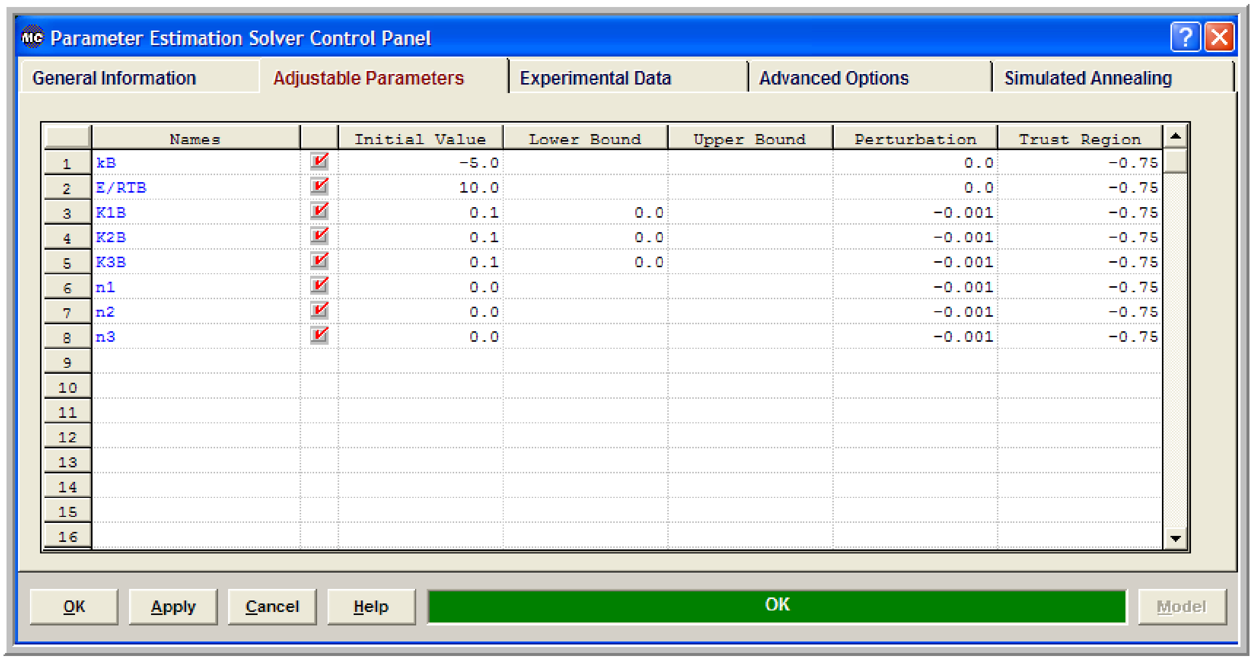
Notice how we use the IF logical construct to enter both models into the estimator.

9.5.1.4 Gradient Vector

The Gradient Vector section is optional. The user may decide to enter the derivatives of the response vector with respect to the parameter vector analytically; this may help situations where errors in the numerical evaluation of the response function derivatives may hinder the progress of the solution algorithm. The matrix **dY( )** is reserved in the Athena environment to represent these derivative elements. Thus **dY(1,3)** holds the derivative of the response **Y(1)** with respect to the parameter **Par(3).** To enter the Response Model Derivative section:

- From the ***Model*** menu choose ***Load Solver***. The **Parameter Estimation Solver Control Panel** appears.
- From the **Derivatives Calculation** group choose the *User Supplied Response Derivatives* option, then choose **OK**.
- From the ***Model*** menu choose ***Gradient Vector*** (or **Hit F11**)

Alternatively if you wish to enter selective derivatives, in the **Parameter Estimation Solver Control Panel** choose the **Adjustable Parameters** tab and enter zero for *Perturbation* for the parameters you will be supplying analytical derivatives as shown below:



	Names		Initial Value	Lower Bound	Upper Bound	Perturbation	Trust Region
1	kB	☑	-5.0			0.0	-0.75
2	E/RTB	☑	10.0			0.0	-0.75
3	K1B	☑	0.1	0.0		-0.001	-0.75
4	K2B	☑	0.1	0.0		-0.001	-0.75
5	K3B	☑	0.1	0.0		-0.001	-0.75
6	n1	☑	0.0			-0.001	-0.75
7	n2	☑	0.0			-0.001	-0.75
8	n3	☑	0.0			-0.001	-0.75
9							
10							
11							
12							
13							
14							
15							
16							

Then choose **OK** and enter the source code for the selected derivatives as shown below:

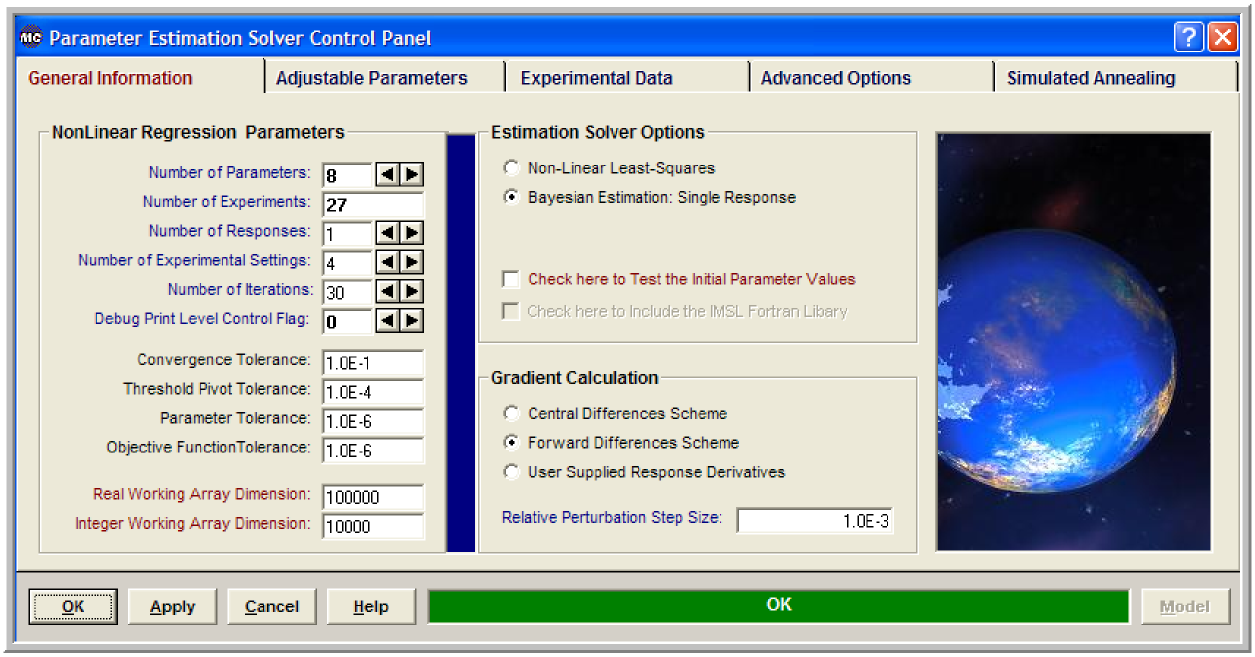
```
@Gradient Vector
 dY(1,1)=Y(1)
 dY(1,2)=Y(1)*(1.0-Tb/Temp)
```

This option is useful, when the model contains sets of parameters with respect to which the derivatives are very simple to derive and implement. For the parameters with respect to which the derivative derivation is complex, the user leaves the task to the parameter estimation solver. In this case the derivatives are calculated by finite differences. It is worth pointing out that the estimation solver in Athena contains technology that optimizes the perturbation step size, in order to minimize the truncation and round-off error. This guarantees the best possible estimation of the gradient vector which is crucial in the minimization process.

9.5.2 The Parameter Estimation Solver

It is now time to access the AthenaVisual Plus solver for Parameter Estimation in order to enter information about the adjustable parameters, the experimental observations and various other parameters that control the estimation algorithm, To do that:

- From the ***Model*** menu choose ***Load Solver*** (or **Hit F12**)
- Enter the solver parameters as shown below for our example



NonLinear Regression Parameters

Number of Parameters:	8
Number of Experiments:	27
Number of Responses:	1
Number of Experimental Settings:	4
Number of Iterations:	30
Debug Print Level Control Flag:	0
Convergence Tolerance:	1.0E-1
Threshold Pivot Tolerance:	1.0E-4
Parameter Tolerance:	1.0E-6
Objective FunctionTolerance:	1.0E-6
Real Working Array Dimension:	100000
Integer Working Array Dimension:	10000

Estimation Solver Options

○ Non-Linear Least-Squares
● Bayesian Estimation: Single Response

□ Check here to Test the Initial Parameter Values
□ Check here to Include the IMSL Fortran Libary

Gradient Calculation

○ Central Differences Scheme
● Forward Differences Scheme
○ User Supplied Response Derivatives

Relative Perturbation Step Size: 1.0E-3

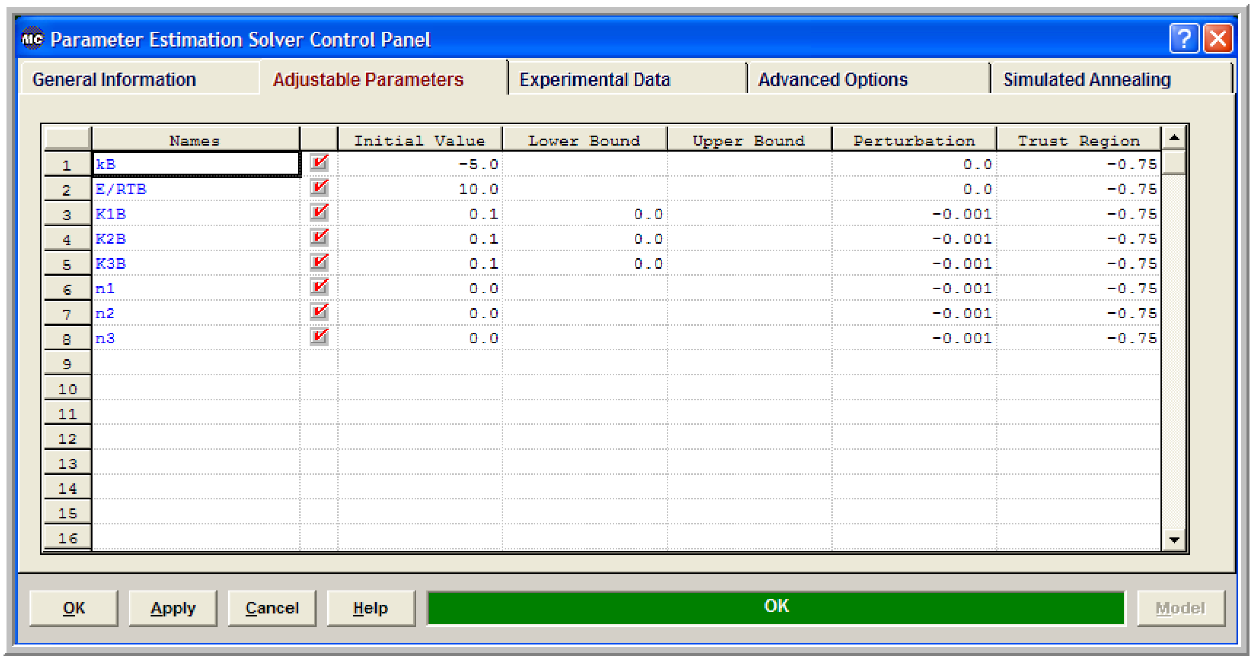
From the **Nonlinear Regression Parameters** group enter:

- The number of the full set of unknown parameters **(8)**
- The number of experiments or datasets **(27)**
- The number of responses or dependent variables **(1)** and
- The number of experimental settings or predictor variables **(4)**

Optionally you may change the *Number of Iterations*, the *Convergence and Parameter Tolerance*, the *Debug Print Level Control Flag* and the Real and Integer working arrays space requirements. From the **Estimation Solver Options** group choose the type of estimation you are going to be using and optionally request a diagonal covariance if the Bayesian estimation has been chosen, or *Click here to perform a Test Call to Model*. The test call to model is a very useful option since it allows you to see how good is your initial parameter guess before you proceed with the estimation process. You also have the option to use a relative weighting factor of the observation vector in the estimation process. From the **Derivatives Calculation** group choose the method for the objective function gradient calculation and optionally enter the *Relative Perturbation Step Size*. Should you choose User Supplied Model Derivatives you must enter the section **@Gradient Vector** when you enter the source code.

9.5.2.1 The Adjustable Parameters

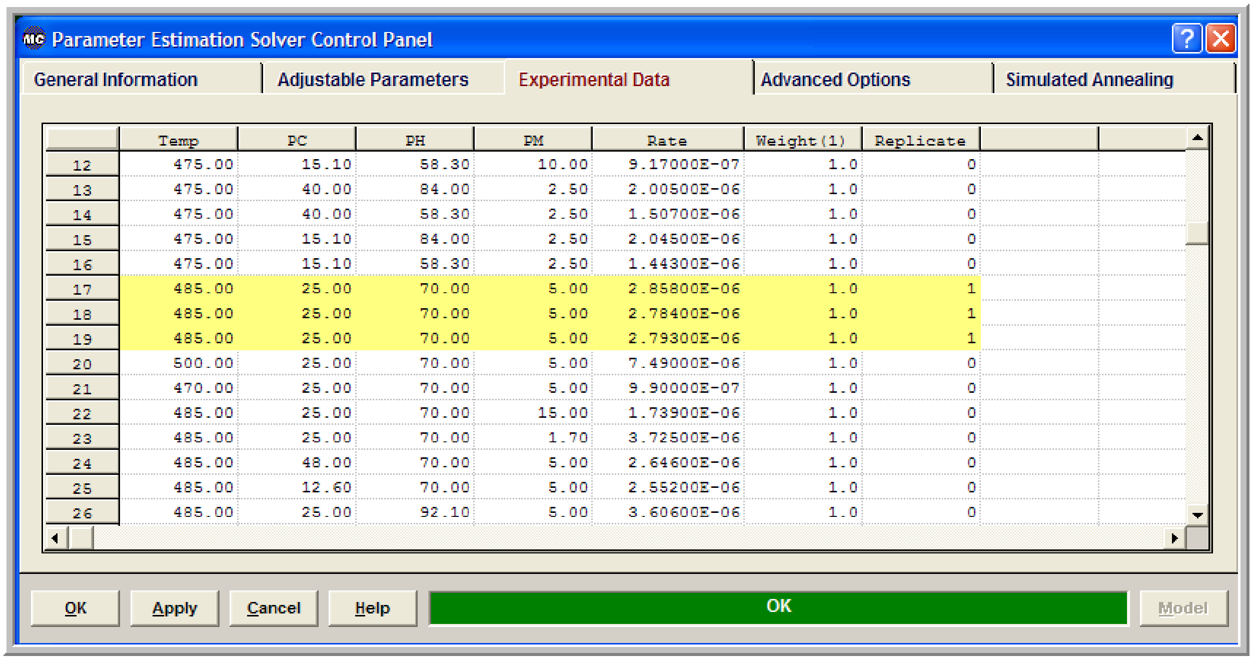
Next select the **Adjustable Parameters** tab on the Parameter Estimation Control Panel and enter the names of the parameters you wish to estimate, their initial guesses and optionally their lower and upper bounds, the relative perturbation step size for gradient computation and the size of the trust region. Notice that if a parameter is not checked in the check box next to it, will remain fixed at its initial value during the estimation. The following data have been inserted for our example:



	Names		Initial Value	Lower Bound	Upper Bound	Perturbation	Trust Region
1	kB	☑	-5.0			0.0	-0.75
2	E/RTB	☑	10.0			0.0	-0.75
3	K1B	☑	0.1	0.0		-0.001	-0.75
4	K2B	☑	0.1	0.0		-0.001	-0.75
5	K3B	☑	0.1	0.0		-0.001	-0.75
6	n1	☑	0.0			-0.001	-0.75
7	n2	☑	0.0			-0.001	-0.75
8	n3	☑	0.0			-0.001	-0.75
9							
10							
11							
12							
13							
14							
15							
16							

9.5.2.2 The Experimental Data

Next select the **Experimental Data** tab on the Parameter Estimation Control Panel and enter the observations and experimental settings. Optionally you may enter weights for each observation. Otherwise these weights are set equal to one by the Parameter Estimation solver. Windows Copy and Paste functions can be used to transfer your data from, say, an Excel spreadsheet. The following data have been inserted for our example: (partial list shown)

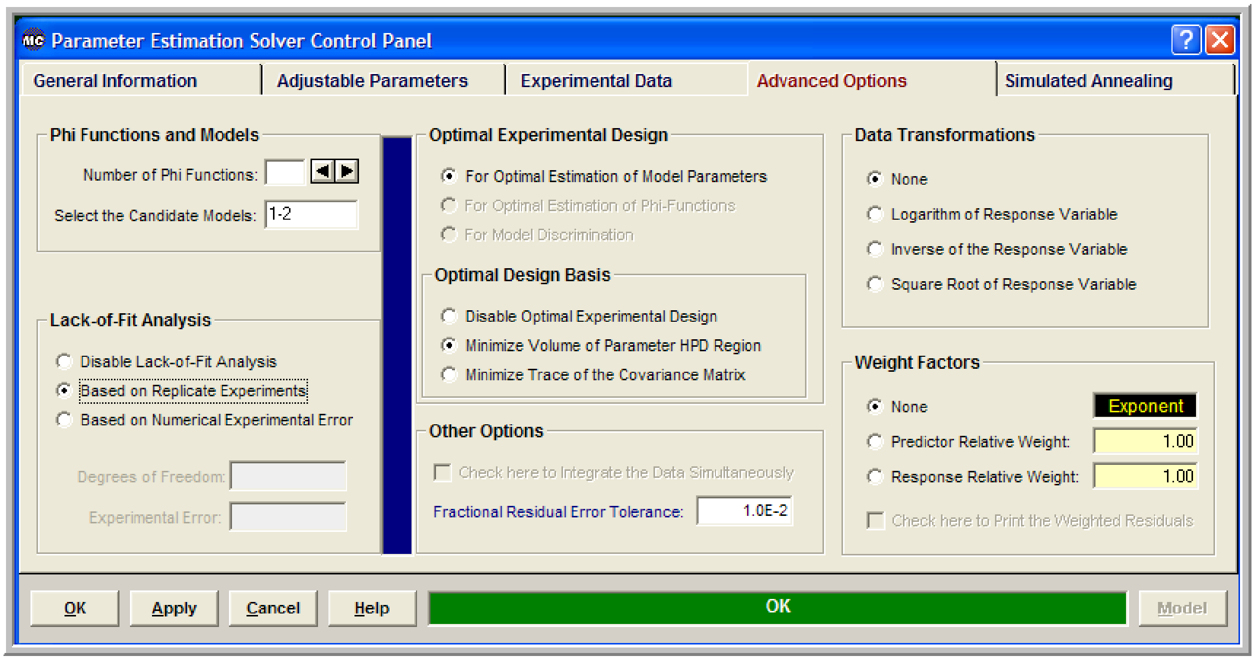


	Temp	PC	PH	PM	Rate	Weight(1)	Replicate
12	475.00	15.10	58.30	10.00	9.17000E-07	1.0	0
13	475.00	40.00	84.00	2.50	2.00500E-06	1.0	0
14	475.00	40.00	58.30	2.50	1.50700E-06	1.0	0
15	475.00	15.10	84.00	2.50	2.04500E-06	1.0	0
16	475.00	15.10	58.30	2.50	1.44300E-06	1.0	0
17	485.00	25.00	70.00	5.00	2.85800E-06	1.0	1
18	485.00	25.00	70.00	5.00	2.78400E-06	1.0	1
19	485.00	25.00	70.00	5.00	2.79300E-06	1.0	1
20	500.00	25.00	70.00	5.00	7.49000E-06	1.0	0
21	470.00	25.00	70.00	5.00	9.90000E-07	1.0	0
22	485.00	25.00	70.00	15.00	1.73900E-06	1.0	0
23	485.00	25.00	70.00	1.70	3.72500E-06	1.0	0
24	485.00	48.00	70.00	5.00	2.64600E-06	1.0	0
25	485.00	12.60	70.00	5.00	2.55200E-06	1.0	0
26	485.00	25.00	92.10	5.00	3.60600E-06	1.0	0

Finally you may specify replicate experiments for performing lack-of-fit analysis and model discrimination. If there was another group of replicates they would have been identified by the number 2 and so on. Choose **OK** and continue

9.5.3 Select the Models for Estimation

Next we must select which models we would like to process. To that we load the Parameter Estimation Solver and select the Advanced Options tab as show below:



Parameter Estimation Solver Control Panel

General Information | Adjustable Parameters | Experimental Data | Advanced Options | Simulated Annealing

Phi Functions and Models

Number of Phi Functions:

Select the Candidate Models: 1-2

Lack-of-Fit Analysis

○ Disable Lack-of-Fit Analysis
● Based on Replicate Experiments
○ Based on Numerical Experimental Error

Degrees of Freedom:

Experimental Error:

Optimal Experimental Design

● For Optimal Estimation of Model Parameters
○ For Optimal Estimation of Phi-Functions
○ For Model Discrimination

Optimal Design Basis

○ Disable Optimal Experimental Design
● Minimize Volume of Parameter HPD Region
○ Minimize Trace of the Covariance Matrix

Other Options

□ Check here to Integrate the Data Simultaneously

Fractional Residual Error Tolerance: 1.0E-2

Data Transformations

● None
○ Logarithm of Response Variable
○ Inverse of the Response Variable
○ Square Root of Response Variable

Weight Factors

● None — Exponent
○ Predictor Relative Weight: 1.00
○ Response Relative Weight: 1.00

□ Check here to Print the Weighted Residuals

OK | Apply | Cancel | Help | OK | Model

In the **Phi Functions and Models** group we enter 1-2 in the *Select the Candidate Models* field. We may also choose to process one model at a time to debug any possible errors. In that case we just simply enter the model number we wish to run. In the **Lack-of-Fit Analysis** group we select to perform this test by choosing *Based on Replicate Experiments*. This option will help us discriminate amongst the models and choose the best candidate as we will describe later. We now choose **OK** and the save our project. Then from the ***Build*** menu we then choose ***Execute***.

9.5.4 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 (partial results):

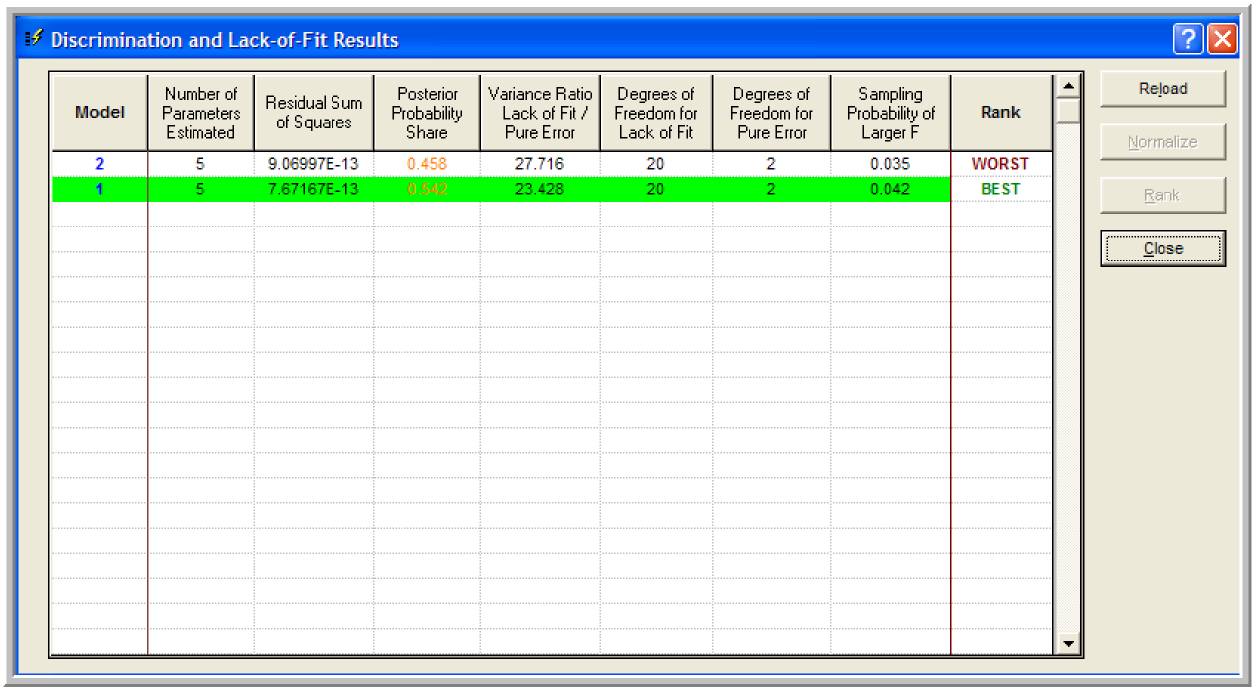
```
Number of Experiments..........................   27
Number of Parameters...........................    8
Number of Responses............................    1
Number of Settings.............................    4

EXIT GREGPLUS: SOLUTION FOUND. MODEL No. 1

STATISTICAL ANALYSIS

OBJECTIVE FUNCTION.......................  7.67167E-13
SUM OF SQUARES OF RESIDUALS..............  7.67167E-13
ESTIMATED PARAMETERS FROM DATA...........            5
TOTAL OBSERVATIONS DATA COUNT IS.........           27

                OPTIMAL
PARAMETER     ESTIMATES        95% MARGINAL HPD INTERVALS      PARAMETER STATUS
 PAR(  1)  -9.709487E+00    -9.709487E+00 +- 2.249E-01      Estimated
 PAR(  2)   3.149089E+01     3.149089E+01 +- 1.597E+00      Estimated
 PAR(  3)   6.563378E-02     6.563378E-02 +- 1.752E-02      Estimated
 PAR(  4)   2.904900E-02     2.904900E-02 +- 1.018E-02      Estimated
 PAR(  5)   9.839118E-02     9.839118E-02 +- 2.958E-02      Estimated
 PAR(  6)   0.000000E+00                                    Fixed
 PAR(  7)   0.000000E+00                                    Fixed
 PAR(  8)   0.000000E+00                                    Fixed

EXIT GREGPLUS: MODEL DISCRIMINATION AND CRITICISM

SOURCE OF VARIANCE      SUM OF SQUARES   DEG. OF FREEDOM   MEAN SQUARE
Residuals                  7.67167E-13                22
Lack of Fit                7.63906E-13                20      3.82E-14
Experimental               3.26067E-15                 2      1.63E-15

Variance Ratio (Lack-of-Fit/Experimental)                       23.428
Sampling Probability of Greater Ratio                            0.042
Log10 of Posterior Unnormalized Probability Density             11.363
```

```
EXIT GREGPLUS: SOLUTION FOUND. MODEL No. 2

STATISTICAL ANALYSIS

OBJECTIVE FUNCTION.......................  9.06997E-13
SUM OF SQUARES OF RESIDUALS..............  9.06997E-13
ESTIMATED PARAMETERS FROM DATA...........            5
TOTAL OBSERVATIONS DATA COUNT IS.........           27

              OPTIMAL
PARAMETER    ESTIMATES      95% MARGINAL HPD INTERVALS      PARAMETER STATUS
 PAR(  1)  -1.582727E+01    -1.582727E+01 +- 6.345E-01      Estimated
 PAR(  2)   3.159274E+01     3.159274E+01 +- 1.779E+00      Estimated
 PAR(  3)   1.000000E-01                                    Fixed
 PAR(  4)   1.000000E-01                                    Fixed
 PAR(  5)   1.000000E-01                                    Fixed
 PAR(  6)   5.421410E-02     5.421410E-02 +- 5.241E-02      Estimated
 PAR(  7)   7.731606E-01     7.731606E-01 +- 1.417E-01      Estimated
 PAR(  8)  -2.901408E-01    -2.901408E-01 +- 3.805E-02      Estimated

EXIT GREGPLUS: MODEL DISCRIMINATION AND CRITICISM

SOURCE OF VARIANCE      SUM OF SQUARES   DEG. OF FREEDOM   MEAN SQUARE
Residuals                  9.06997E-13                22
Lack of Fit                9.03736E-13                20      4.52E-14
Experimental               3.26067E-15                 2      1.63E-15

Variance Ratio (Lack-of-Fit/Experimental)                       27.716
Sampling Probability of Greater Ratio                            0.035
Log10 of Posterior Unnormalized Probability Density             11.290
```

We can clearly see the results of the estimation of the two models as well as the Lack-of-Fit analysis tests.

9.5.4.1 Model Discrimination Results

To see the Model Discrimination results in a tabular from the ***View*** menu select ***Discrimination and Lack-of-Fit***. In the window that appears click first **Normalize** and then **Rank** to see which of the two models is the best as shown in color coded window below:

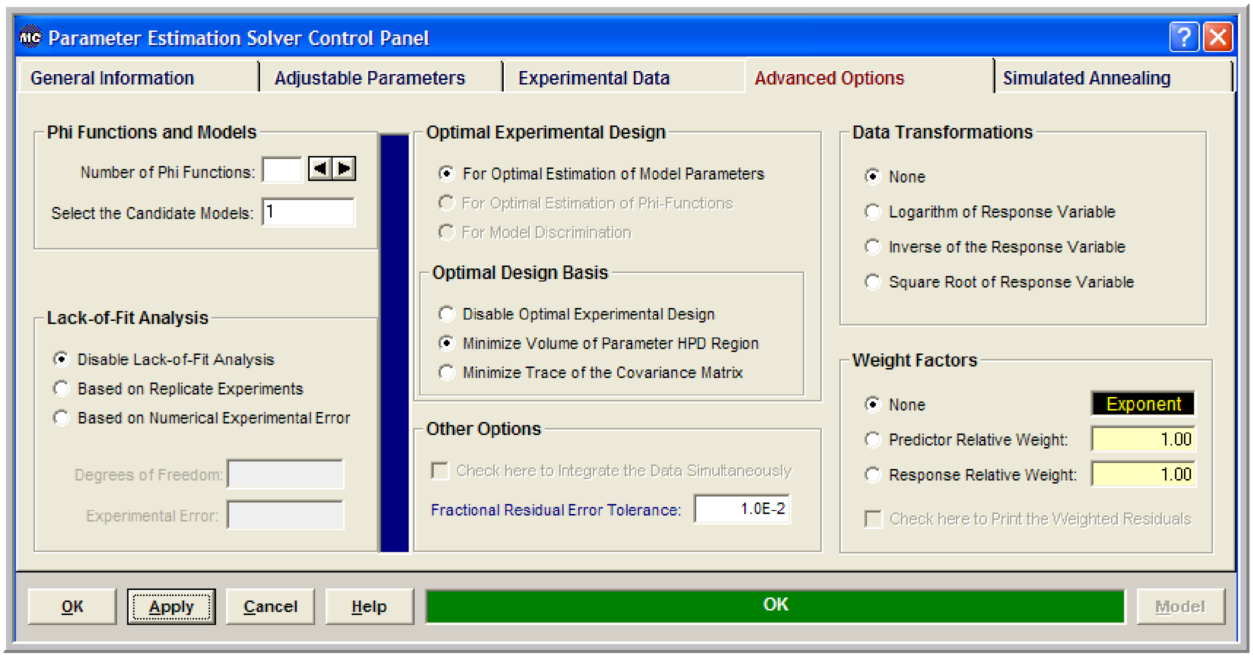


Model	Number of Parameters Estimated	Residual Sum of Squares	Posterior Probability Share	Variance Ratio Lack of Fit / Pure Error	Degrees of Freedom for Lack of Fit	Degrees of Freedom for Pure Error	Sampling Probability of Larger F	Rank
2	5	9.06997E-13	0.458	27.716	20	2	0.035	WORST
1	5	7.67167E-13	0.542	23.428	20	2	0.042	BEST

Notice that this table contains a wealth of statistical information about the two models. A deep knowledge of statistics is required for complete understanding of this table. The user is referred to the various references in the previous sections for information regarding model discrimination and lack-of-fit.

9.5.5 Optimal Experimental Design

Suppose now that we choose the best model as outlined in the previous section and we wish to perform a few experiments in order to improve the parameter estimates. To do that first we load the parameter estimation solver:



Phi Functions and Models

Number of Phi Functions:

Select the Candidate Models: 1

Lack-of-Fit Analysis

- ● Disable Lack-of-Fit Analysis
- ○ Based on Replicate Experiments
- ○ Based on Numerical Experimental Error

Degrees of Freedom:

Experimental Error:

Optimal Experimental Design

- ● For Optimal Estimation of Model Parameters
- ○ For Optimal Estimation of Phi-Functions
- ○ For Model Discrimination

Optimal Design Basis

- ○ Disable Optimal Experimental Design
- ● Minimize Volume of Parameter HPD Region
- ○ Minimize Trace of the Covariance Matrix

Other Options

□ Check here to Integrate the Data Simultaneously

Fractional Residual Error Tolerance: 1.0E-2

Data Transformations

- ● None
- ○ Logarithm of Response Variable
- ○ Inverse of the Response Variable
- ○ Square Root of Response Variable

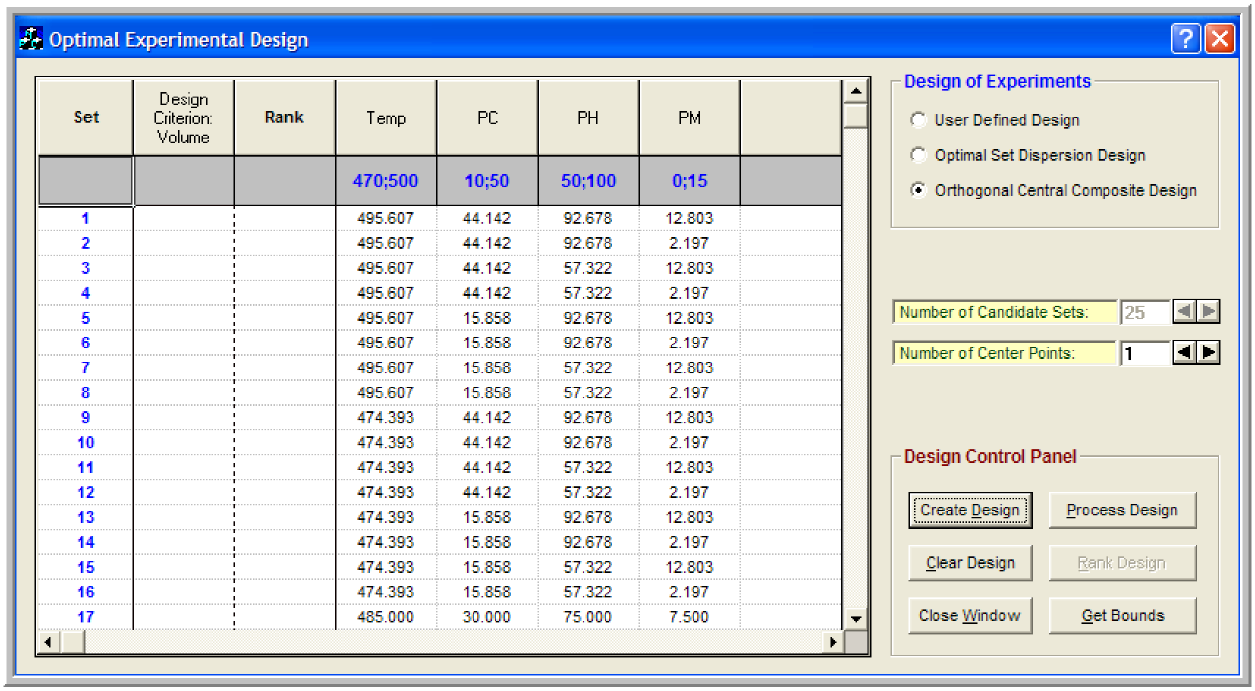
Weight Factors

- ● None — Exponent
- ○ Predictor Relative Weight: 1.00
- ○ Response Relative Weight: 1.00

□ Check here to Print the Weighted Residuals

In the **Phi Functions and Models** group we enter 1(our best of two models) in the *Select the Candidate Models* field. In the **Lack-of-Fit Analysis** group we disable this statistical test and in the **Optional Design Basis** group we choose to *Minimize Volume of Parameter HPD Region*. We now choose **OK** and the save our project. Then from the ***View*** menu we then choose ***Optimal Experimental Design***.

The **Optimal Experimental Design** window appears. In this panel we first enter the lower and upper bounds of the experimental settings (or click **Get Bounds** to insert the maximum and minimum values of the corresponding experimental settings) and then select *Orthogonal Central Composite* design with the *Number of Center Points* equal to 1. Please observe that you may wish to enter your own design by selecting the appropriate option in the **Design of Experiments** group or use a different method based on minimum dispersion. We now click **Create Design** and the Athena solver generates 25 experiments as shown below:



Optimal Experimental Design

Set	Design Criterion: Volume	Rank	Temp	PC	PH	PM	
			470;500	10;50	50;100	0;15	
1			495.607	44.142	92.678	12.803	
2			495.607	44.142	92.678	2.197	
3			495.607	44.142	57.322	12.803	
4			495.607	44.142	57.322	2.197	
5			495.607	15.858	92.678	12.803	
6			495.607	15.858	92.678	2.197	
7			495.607	15.858	57.322	12.803	
8			495.607	15.858	57.322	2.197	
9			474.393	44.142	92.678	12.803	
10			474.393	44.142	92.678	2.197	
11			474.393	44.142	57.322	12.803	
12			474.393	44.142	57.322	2.197	
13			474.393	15.858	92.678	12.803	
14			474.393	15.858	92.678	2.197	
15			474.393	15.858	57.322	12.803	
16			474.393	15.858	57.322	2.197	
17			485.000	30.000	75.000	7.500	

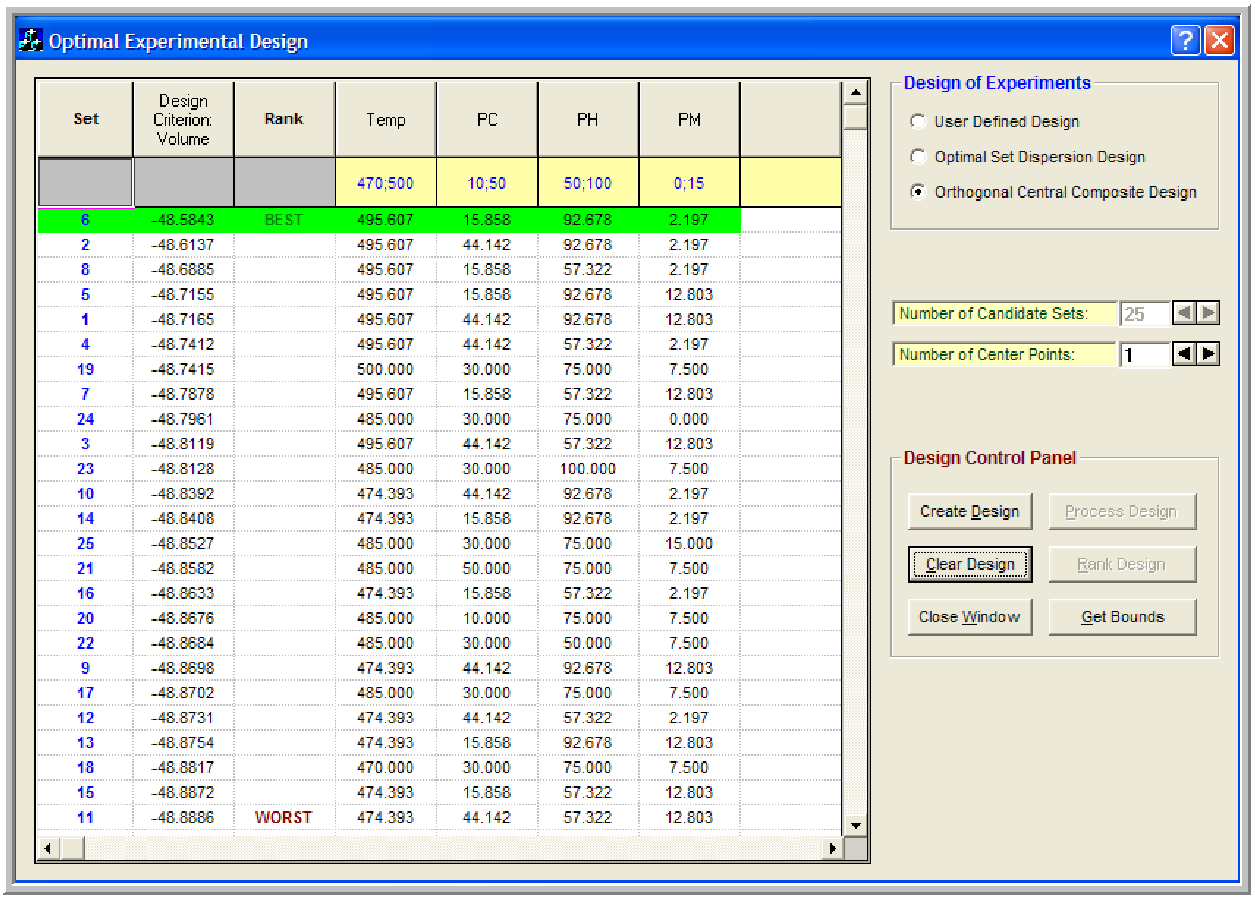
Design of Experiments

○ User Defined Design
○ Optimal Set Dispersion Design
● Orthogonal Central Composite Design

Number of Candidate Sets: 25
Number of Center Points: 1

We then click **Process Design**. The Parameter Estimation Solver control panel appears. Click **OK** to generate the proper information and run your model.

From the ***View*** menu select ***Optimal Experimental Design***. The **Optimal Experimental Design** control panel appears again, but now with all the information necessary to rank this design. In order to do that click **Rank Design** and the Athena solver will rank all the proposed experiments sequentially from the best to the worst as shown below. The experiment highlighted in green will provide (if executed) the most information required to improve the parameter estimates. You may now clear this design and investigate a different design with other criteria or different lower and upper bound on the experimental settings.



Set	Design Criterion: Volume	Rank	Temp	PC	PH	PM
			470;500	10;50	50;100	0;15
6	-48.5843	BEST	495.607	15.858	92.678	2.197
2	-48.6137		495.607	44.142	92.678	2.197
8	-48.6885		495.607	15.858	57.322	2.197
5	-48.7155		495.607	15.858	92.678	12.803
1	-48.7165		495.607	44.142	92.678	12.803
4	-48.7412		495.607	44.142	57.322	2.197
19	-48.7415		500.000	30.000	75.000	7.500
7	-48.7878		495.607	15.858	57.322	12.803
24	-48.7961		485.000	30.000	75.000	0.000
3	-48.8119		495.607	44.142	57.322	12.803
23	-48.8128		485.000	30.000	100.000	7.500
10	-48.8392		474.393	44.142	92.678	2.197
14	-48.8408		474.393	15.858	92.678	2.197
25	-48.8527		485.000	30.000	75.000	15.000
21	-48.8582		485.000	50.000	75.000	7.500
16	-48.8633		474.393	15.858	57.322	2.197
20	-48.8676		485.000	10.000	75.000	7.500
22	-48.8684		485.000	30.000	50.000	7.500
9	-48.8698		474.393	44.142	92.678	12.803
17	-48.8702		485.000	30.000	75.000	7.500
12	-48.8731		474.393	44.142	57.322	2.197
13	-48.8754		474.393	15.858	92.678	12.803
18	-48.8817		470.000	30.000	75.000	7.500
15	-48.8872		474.393	15.858	57.322	12.803
11	-48.8886	WORST	474.393	44.142	57.322	12.803

Design of Experiments

○ User Defined Design
○ Optimal Set Dispersion Design
● Orthogonal Central Composite Design

Number of Candidate Sets: 25

Number of Center Points: 1

Design Control Panel

Index