

Athena Visual Studio Differential Model Tutorial

- ❖ 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**.

Differential/Algebraic models in Athena take the form:

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

where \mathbf{u} is a state vector of unknowns (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.

Tutorial: 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 $0 \leq t \leq 10 \text{ hr}$
- ❖ 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.,

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

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 (C_4) is defined via the auxiliary variable:

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

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

$$\begin{bmatrix} U(1,1) & U(1,2) & U(1,3) \\ U(2,1) & U(2,2) & U(2,3) \\ U(3,1) & U(3,2) & U(3,3) \\ U(4,1) & U(4,2) & U(4,3) \end{bmatrix} = \begin{bmatrix} C_1 & \frac{\partial C_1}{\partial k_3} & \frac{\partial C_1}{\partial k_4} \\ C_2 & \frac{\partial C_2}{\partial k_3} & \frac{\partial C_2}{\partial k_4} \\ C_3 & \frac{\partial C_3}{\partial k_3} & \frac{\partial C_3}{\partial k_4} \\ C_4 & \frac{\partial C_4}{\partial k_3} & \frac{\partial C_4}{\partial k_4} \end{bmatrix}$$

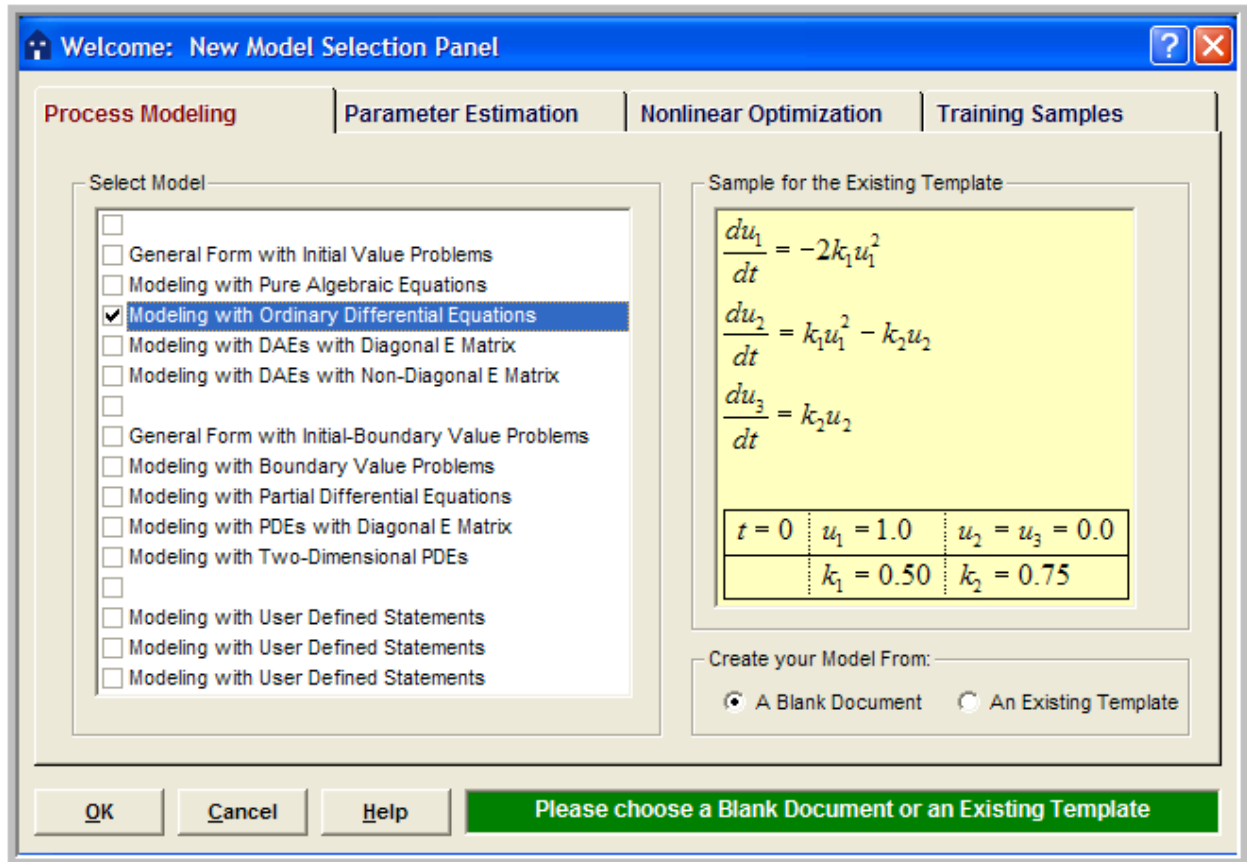
This example tutorial is already precoded in Athena Visual Studio. If you do not wish to type the code on your own you may access it by doing the following:

- ❖ Open **Athena Visual Studio**
- ❖ From the **File** menu click **New**
- ❖ Select the **Training Samples** tab
- ❖ Select the **Batch Isothermal Reactor** sample
- ❖ Click **OK**

Implementation in Athena Visual Studio

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

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



- ❖ 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.

Writing the Source Code

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 $\mathbf{E}(t, \mathbf{u}; \boldsymbol{\theta})$ of DAE systems with diagonal or non-diagonal matrix. A data section not labeled by Athena Visual Studio 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 Athena Visual Studio syntax rules shown below:

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 Athena Visual Studio 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
```

Declaration of Variables

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

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

In the above statements the variables *x, y, z, krate* will be treated as double precision and will be accessible by all modeling sections. Similarly the variables *Skount, Ncc* will be treated as integer and be accessible by all modeling sections. Character variables are assigned as Character*132 from the Athena Visual Studio parser. Single precision variables cannot be declared **Global**.

Vectors and matrices can be declared in a similar manner. The array size, type and number of dimensions are declared with the **Global** statement. The elements of the array can be referenced by an integer index number, which runs from one (or zero) to the maximum number declared in the **Global** statement:

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

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

```
Dim Temp, Pres As Real  
Dim TotalFlow As Single  
Dim i As Integer
```

In the above statements the variables *Temp, Pres* will be treated as double precision, where as the variable *TotalFlow* will be treated as single precision; these variables will be accessible only at the section where they have been declared. Similarly the variable *i* will be treated as integer and will be accessible only by the corresponding modeling section where it has been declared.

Vectors and matrices can be declared in a similar manner. The array size, type and number of dimensions are declared with the **Dim** statement. The elements of the array are referenced by an integer index, which runs from one(or zero) to the number declared in the **Dim** statement:

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

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

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

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

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

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

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

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

Initial Conditions

In the Initial Conditions section the user must enter the initial values for the unknown state vector. The initial values are required by the algorithm in 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
```

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 $F(1)$ 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)
```

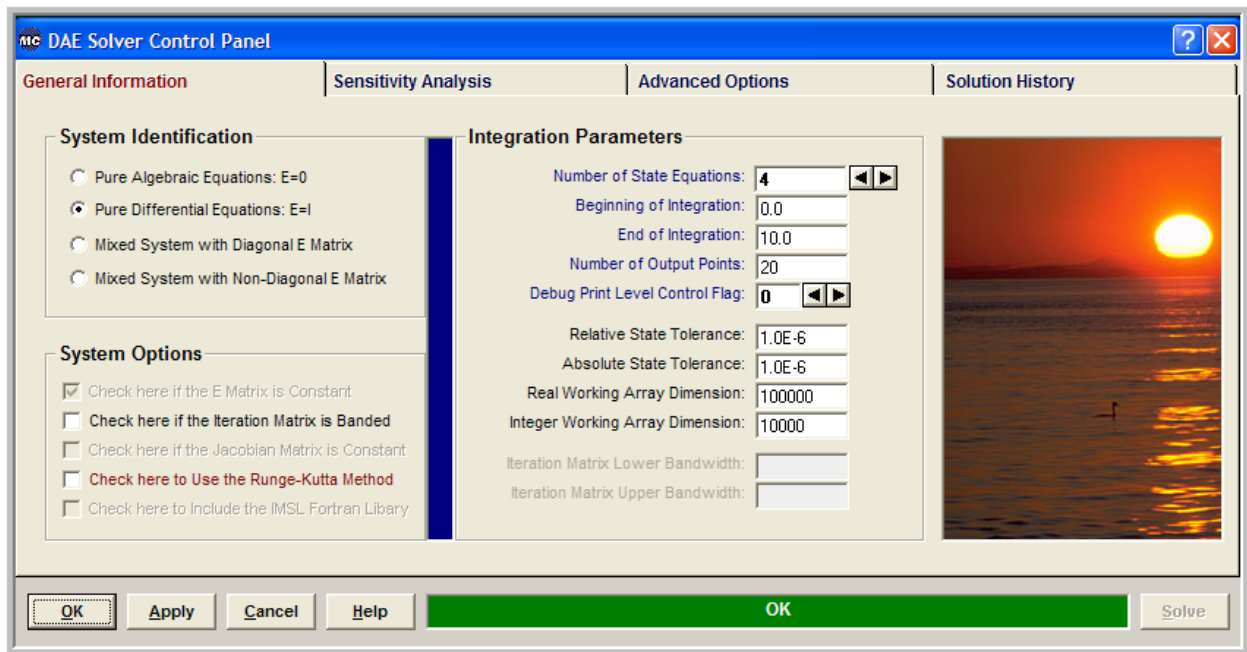
```
F(3) = k5 * U(2)
```

```
F(4) = -b * k3 * U(1) * U(4) / (k4 + U(4))
```


The DDAPLUS Solver

It is now time to load the Athena Visual Studio 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**)
- ❖ The *DAE Solver Control Panel* window appears
- ❖ Enter the solver parameters as shown below for our example



In the **System Identification** group you will see that the option *Pure 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**.

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 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**)

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
```

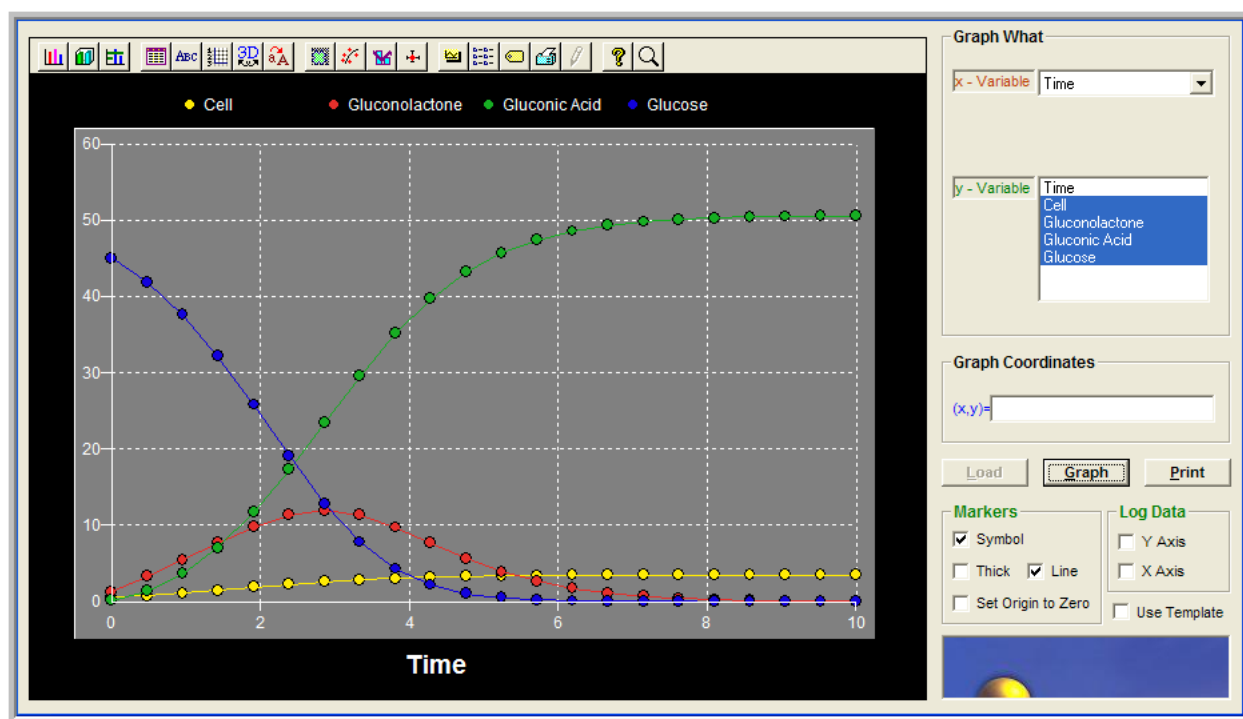
TIME	U(1)	U(2)	U(3)	U(4)
0.00000E+00	5.60000E-01	1.28000E+00	1.60000E-01	4.50000E+01
4.76190E-01	8.06528E-01	3.30736E+00	1.35857E+00	4.18498E+01
9.52381E-01	1.12023E+00	5.45140E+00	3.64626E+00	3.75817E+01
1.42857E+00	1.48852E+00	7.69245E+00	7.08777E+00	3.21560E+01
1.90476E+00	1.88213E+00	9.80013E+00	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
```

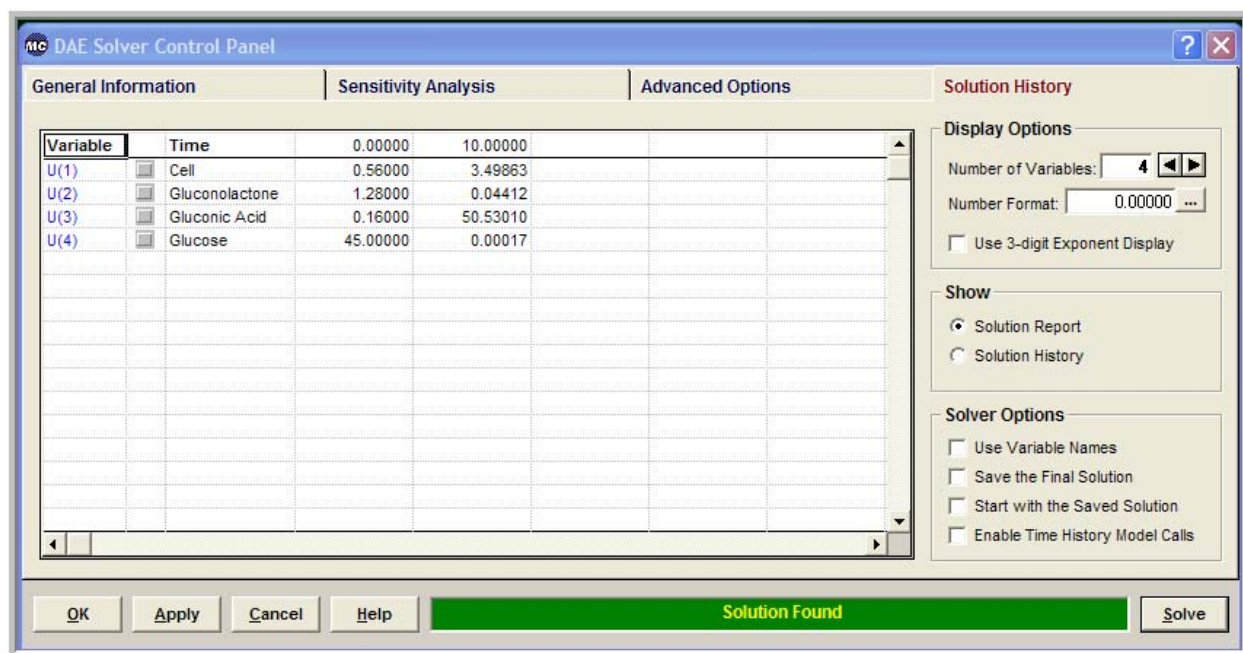
Graphical Results

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 Athena Visual Studio graphics control panel appears:



In this window first 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:



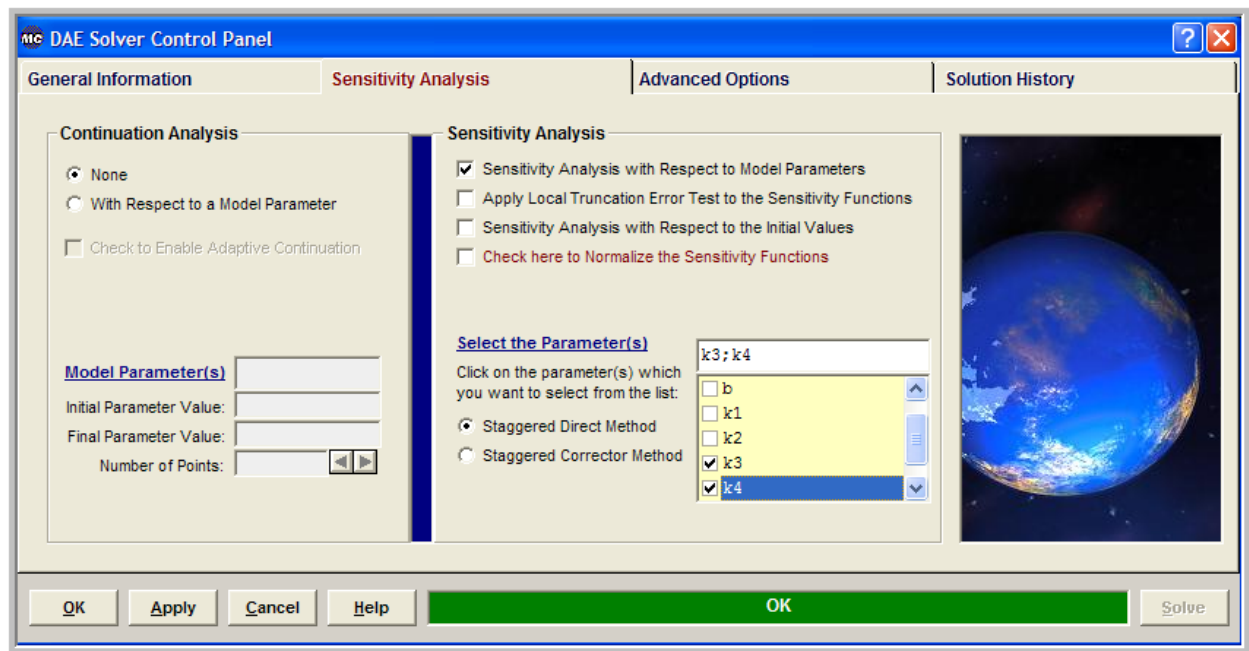
In the **Display Options** group enter the *Number of Variables* using the spin control and then, in the adjacent spreadsheet change the names of the state variables. Also from the **Show** group select *Solution Report*. Then click *Apply* and then *Solve*. You should see the results as shown in the 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.

Sensitivity Analysis

Athena Visual Studio 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.



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*):

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

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*.

Implicit Auxiliary Variables

Athena Visual Studio allows for convenient and efficient calculation of derived quantities by the introduction of auxiliary variables. Assume for example that we wish to estimate the conversion of the glucose (C_4) 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.

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

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 Athena Visual Studio 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.

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 Athena Visual Studio 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 Athena Visual Studio 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

```

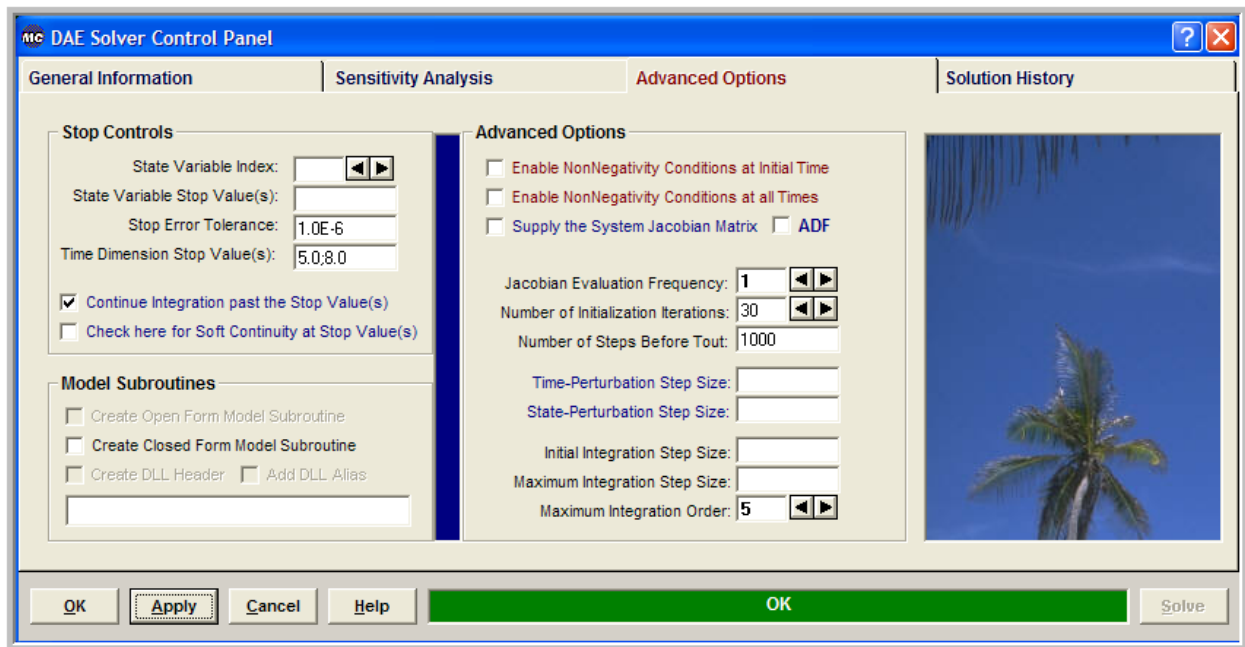
From the above table we clearly see how the Athena Visual Studio solver augmented the auxiliary variables to the state vector for printing purposes. One can now access the Graphics control panel to graph the auxiliary variables along with the state variables.

Integration with Implicit or Explicit Discontinuities

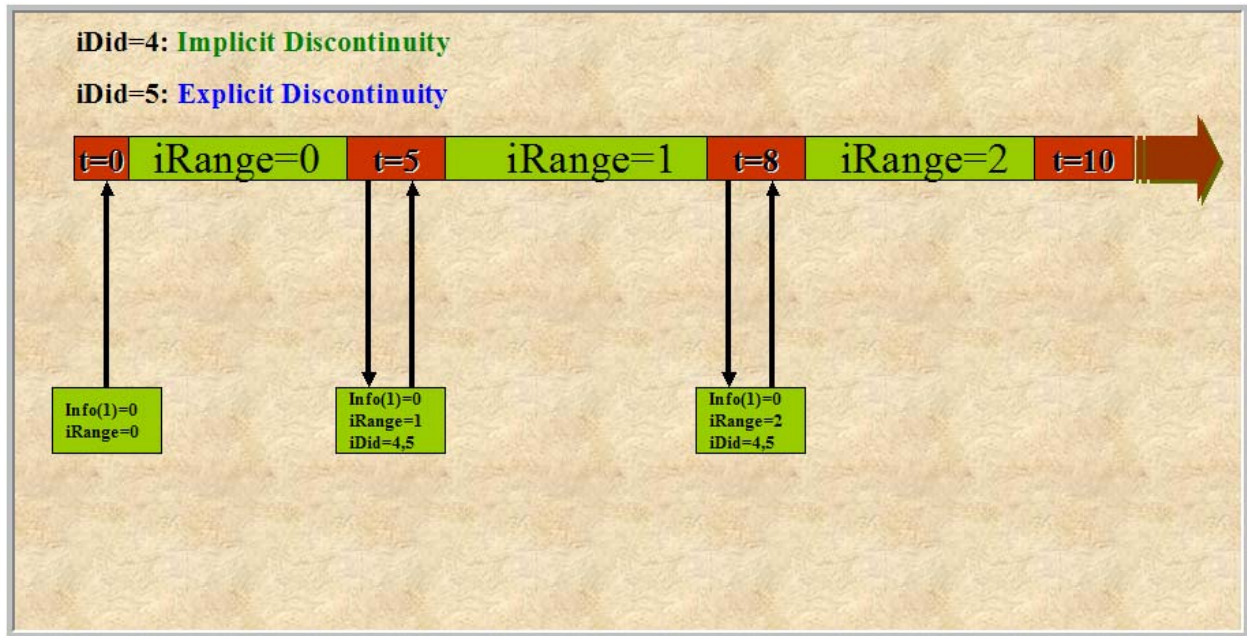
Athena Visual Studio offers a convenient way to integrate systems of Differential/Algebraic equations that exhibit implicit or explicit discontinuities in the independent variable. To illustrate this capability assume that for our example, the rate of cell growth formation abruptly changes according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

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

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



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. We 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
```

Root Finding and Implicit Discontinuities

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

$$h_i(t_i^*, \mathbf{x}, \mathbf{u}; \boldsymbol{\theta}) = 0 \quad i = 1, 2, \dots, m$$

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 $h_i(t_i^*)$) 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 $0 \leq t \leq t^*$, the value **iRange=1** denotes the second zone from $t^* \leq t \leq T$ 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:

$$h(t^*, \mathbf{x}, \mathbf{u}; \boldsymbol{\theta}) = \frac{k_3 C_1 C_4}{k_4 + C_4} - ak_5 C_2 = 0$$

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:

$$F(5) = U(5) - \frac{k_3 C_1 C_4}{k_4 + C_4} - ak_5 C_2$$

We can implement this in Athena Visual Studio 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.

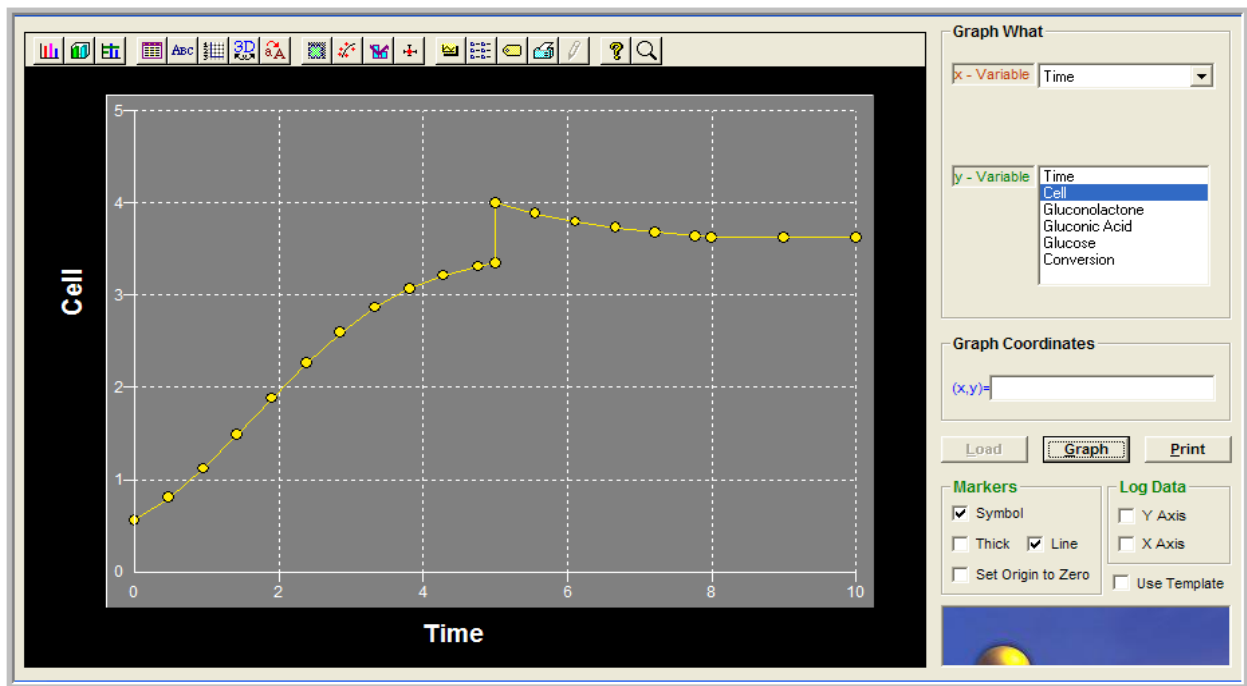
Integration with Changing Initial Conditions at Discontinuities

This option in Athena Visual Studio 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 $U(1) = C_1(t = 5) = 4.0$.

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 then 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:



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 $5 < t \leq 8$. 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 Athena Visual Studio *Help* menu select *Athena Solvers* and click on the **Info()** array subject.