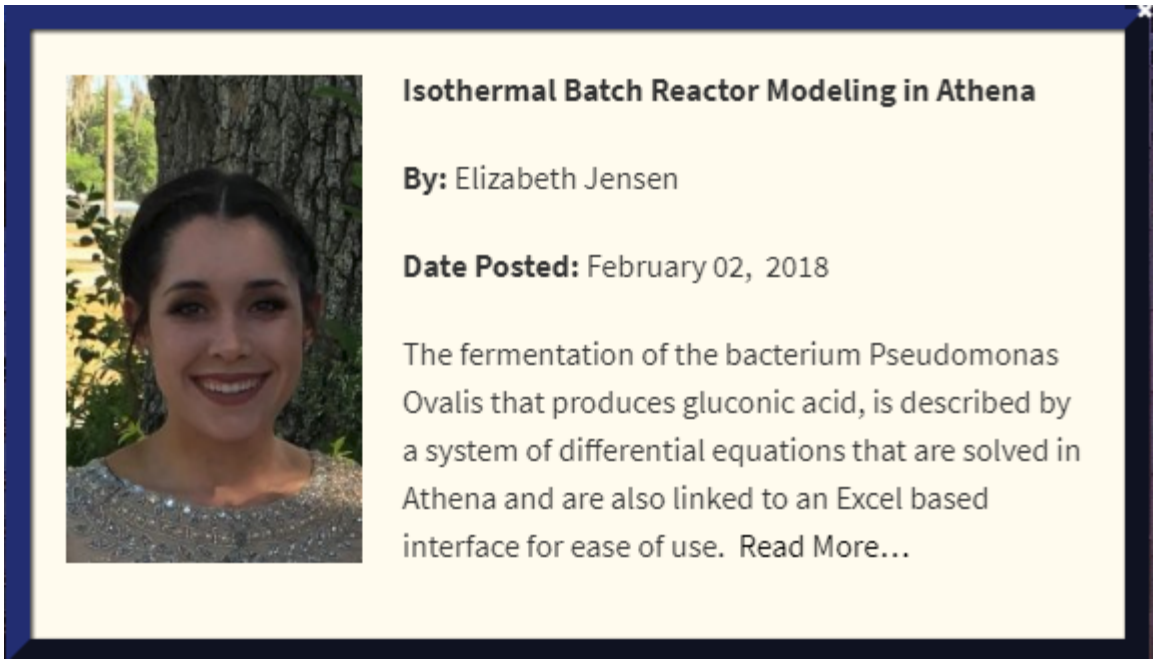


Instructions for use of the tutorial:

Download the compressed file **Example1.zip** and store it on a folder of your choice on your desktop, or in a location where you have rights to read and write. Open the compressed folder and extract all files in the same location.

Non-Athena Users:

If you are not a holder of Athena Visual Studio license you will be unable to open the Athena project **Example1.avw** and recompile it to create an executable or a dynamic link library. You can still view the Athena file with your favorite editor like Notepad or Notepad++. You can run the example from the Excel interface implementation in the file **Example1.xlsm**, but you will only be able to change inputs to the model from the Excel interface. To run the model from the Excel interface, first make sure that the dynamic link library (DLL) that contains the model **Example1.dll** is in the same folder as the Excel file **Example1.xlsm.**

Athena Users:

If you are a holder of Athena Visual Studio Academic or Commercial License you will be able to open the Athena file **Example1.avw** and modify it to meet your needs or to investigate different situations. If you want to run the model stand-alone and not through Excel you must set the flag **ClosedModel=0** in the **@Solver Options** and then recompile and execute your model. Setting the same flag back to the value **ClosedModel =1** will allow you again to recreate the Dynamic Link Library (DLL) which you can call from the Excel interface. Please keep in mind that you may have to close Excel in order to make modifications to the Athena code in order to recreate the DLL or the EXE files.

Model Description

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 by introducing an implicit auxiliary variable, $U(5)$

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

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

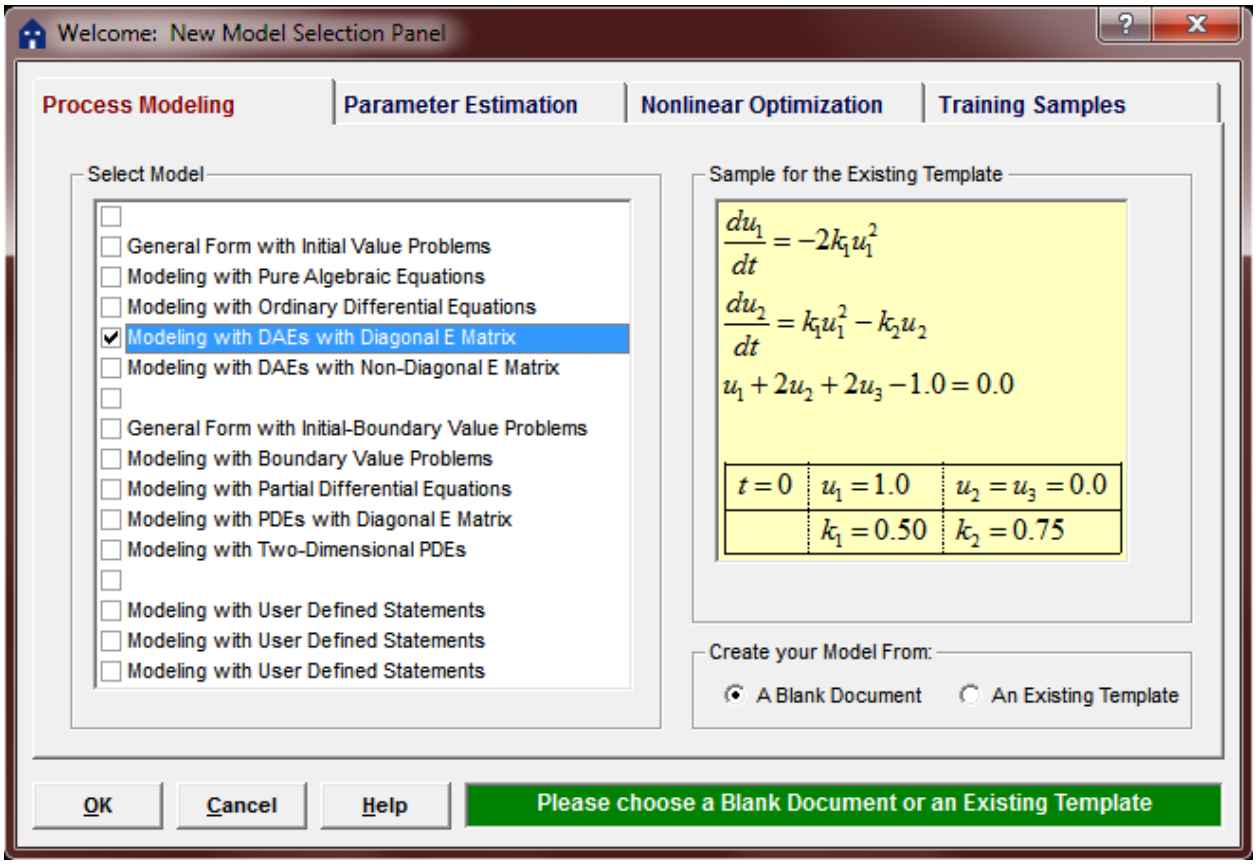
The sensitivity functions of the components with respect to the reaction rate constants $\{k_3, k_4\}$ are given by the first order differentials as indicated below (notice the Athena Visual Studio nomenclature):

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

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 DAEs 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.

Writing the Source Code

You must enter a minimum of three sections in order to create the differential/algebraic 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 require to enter the matrix $\mathbf{E}(t,\mathbf{u};\theta)$ of DAE systems with a 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.

```
Global a,b As Real
a=0.9082    !Fermentation parameter
b=1.0110    !Fermentation parameter

Global C1o,C2o,C3o,C4o As Real
C1o=0.5    !Cell concentration
C2o=0.0    !Gluconolactone concentration
C3o=0.0    !Gluconic acid concentration
C4o=50.0   !Glucose concentration

Global k1,k2,k3,k4,k5 As Real
k1=0.95
k2=3.50
k3=18.0
k4=37.5
k5=1.10

Global RxnTime As Real
RxnTime=10.0    !Reaction time, hr
Npts=22         !Number of profile points
```

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 Declaration Statements: You can insert Fortran declaration statements by prefixing them with the double dollar sign. Below please see a list of Fortran declaration statements that you can insert in your Athena code. Consult your Fortran 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*
$$Real(Kind=dp**), Dimension(:), Allocatable::** *w*
$$Type *myObject*
$$ Real*8:: *k1*
$$ Real*8:: *k2*
$$End Type *myObject*

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. The variable **U(5)** is used to hold the conversion 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)=C1o    !Initial cell concentration
 U(2)=C2o    !Initial gluconolactone concentration
 U(3)=C3o    !Initial gluconic acid
 U(4)=C4o    !Initial glucose concentration
 U(5)=0.0    !Variable for glucose conversion
```

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
 Dim C1,C2,C3,C4 As Real
 C1=U(1)
 C2=U(2)
 C3=U(3)
 C4=U(4)

 F(1)=k1*C1*(1.0-C1/k2)          !Cell balance
 F(2)=k3*C1*C4/(k4+C4)-a*k5*C2   !Gluconolactone balance
 F(3)=k5*C2                      !Gluconic acid balance
 F(4)=-b*k3*C1*C4/(k4+C4)        !Glucose balance
 F(5)=U(5)-(C4o-C4)/C4o*100.0    !Glucose conversion auxiliary equation
```

The fifth equation above is an algebraic equation that defines the conversion of glucose according to the well known formula:

$$x_{\text{Glucose}} = U(5) = \frac{C_{4o} - U(4)}{C_{4o}} \times 100 \Rightarrow F(5) = U(5) - \frac{C_{4o} - U(4)}{C_{4o}} \times 100$$

where C_{4o} is the initial concentration of glucose.

Coefficient Matrix

In the Coefficient Matrix section the user must enter the values of the coefficient of the time derivatives of the state variables. The vector E( ) is reserved in the Athena environment to enter these coefficients. For our example it is obvious that the first four values of the matrix E( ) are equal to one, whereas the last value of the coefficient matrix is zero, since it corresponds to the conversion equation which is an algebraic equation. To enter the Coefficient Matrix section for our example

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

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

Solver Options

In the Solver Options section the user must enter the values for parameters that control the Athena Solvers, and also the structure of the Dynamic Link Library that will be called from the Excel interface. The use of objects such as **ModelInput** and **ModelResults** facilitates the communication between Excel and Athena. Changes can be easily made to have fewer or more input/output variables. The **ClosedModel** parameter is used to toggle between creating an Executable file and a Dynamic Link Library. The Solver Options section is accessed from the ***Model*** menu

After Calling Solver

The After Calling Solver section is used to populate the arrays that will be passed to the Excel interface for plotting and display purposes. The After Calling Solver section is accessed from the ***Model*** menu

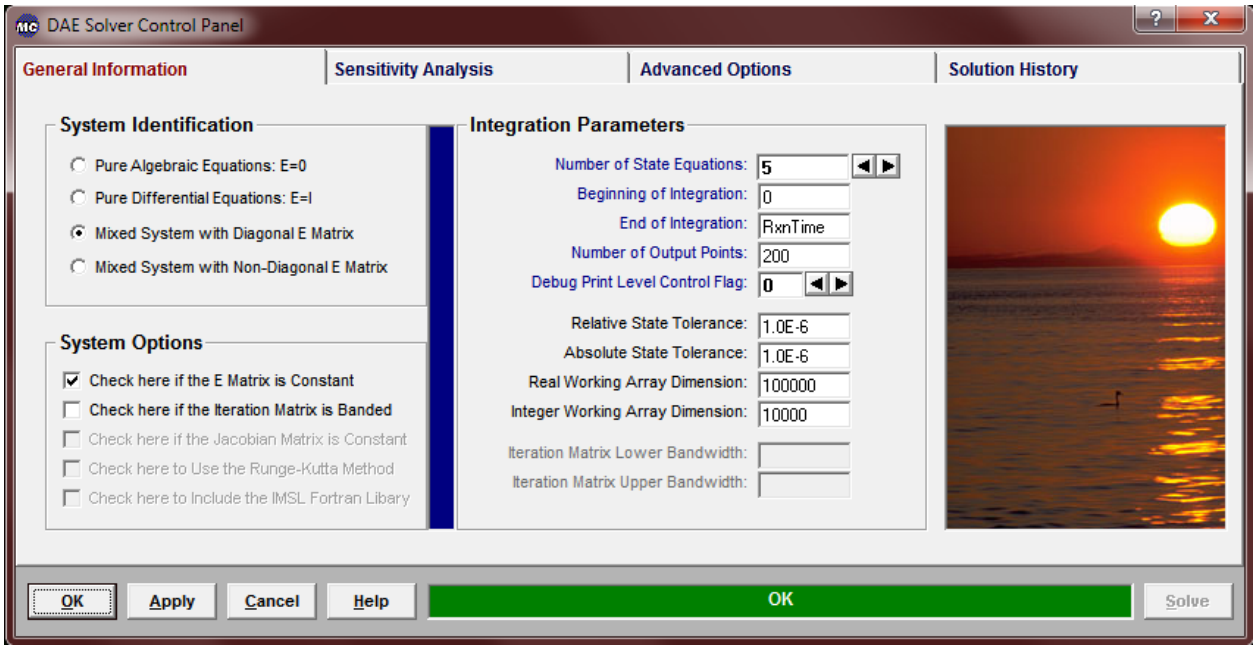
User Output

The User Output section is used to calculate other quantities of interest that need to be passed to the Excel interface for display. The User Output section is executed at the end of the integration. The User Output section is accessed from the ***Model*** menu

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 *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* 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**)

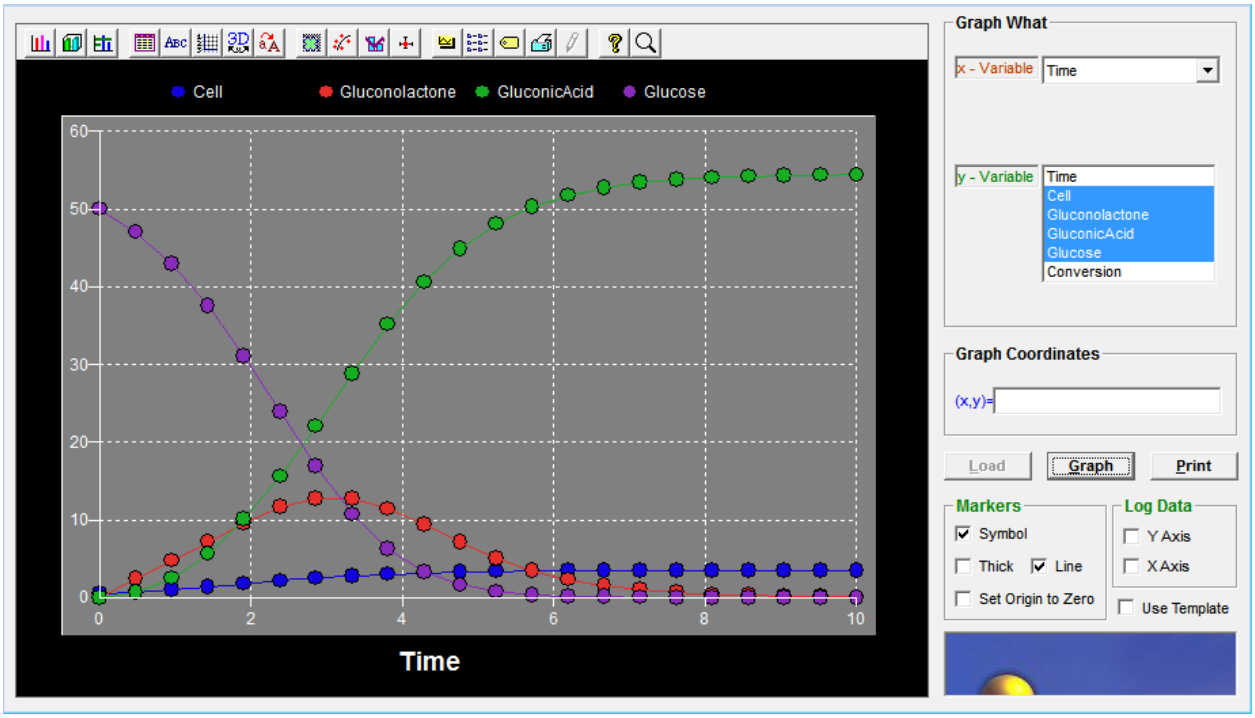
Numerical Results

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.....................     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   5.00000E-01   0.00000E+00   0.00000E+00   5.00000E+01   0.00000E+00
 4.76190E-01   7.26643E-01   2.36985E+00   6.25395E-01   4.70299E+01   5.94029E+00
 9.52381E-01   1.02105E+00   4.73938E+00   2.48375E+00   4.29279E+01   1.41441E+01
 1.42857E+00   1.37559E+00   7.19745E+00   5.60724E+00   3.75749E+01   2.48503E+01
 1.90476E+00   1.76554E+00   9.60998E+00   1.00173E+01   3.10865E+01   3.78270E+01
 2.38095E+00   2.15396E+00   1.16244E+01   1.56061E+01   2.39183E+01   5.21633E+01
 2.85714E+00   2.50445E+00   1.27746E+01   2.20438E+01   1.68444E+01   6.63112E+01
 3.33333E+00   2.79360E+00   1.27055E+01   2.87739E+01   1.07348E+01   7.85304E+01
 3.80952E+00   3.01503E+00   1.14229E+01   3.51398E+01   6.18646E+00   8.76271E+01
 4.28571E+00   3.17513E+00   9.34991E+00   4.06017E+01   3.26712E+00   9.34658E+01
 4.76190E+00   3.28612E+00   7.07742E+00   4.49011E+01   1.61695E+00   9.67661E+01
 5.23810E+00   3.36085E+00   5.04896E+00   4.80604E+01   7.66909E-01   9.84662E+01
 5.71429E+00   3.41019E+00   3.45281E+00   5.02671E+01   3.54468E-01   9.92911E+01
 6.19048E+00   3.44233E+00   2.29312E+00   5.17543E+01   1.61377E-01   9.96772E+01
 6.66667E+00   3.46310E+00   1.49259E+00   5.27322E+01   7.28250E-02   9.98544E+01
 7.14286E+00   3.47643E+00   9.58161E-01   5.33643E+01   3.26945E-02   9.99346E+01
 7.61905E+00   3.48497E+00   6.09209E-01   5.37682E+01   1.46330E-02   9.99707E+01
 8.09524E+00   3.49043E+00   3.84757E-01   5.40242E+01   6.53713E-03   9.99869E+01
 8.57143E+00   3.49390E+00   2.41862E-01   5.41855E+01   2.91721E-03   9.99942E+01
 9.04762E+00   3.49612E+00   1.51534E-01   5.42867E+01   1.30111E-03   9.99974E+01
 9.52381E+00   3.49753E+00   9.47190E-02   5.43500E+01   5.80061E-04   9.99988E+01
 1.00000E+01   3.49843E+00   5.91072E-02   5.43896E+01   2.58524E-04   9.99995E+01

EXIT DDAPLUS: SOLUTION FOUND

Number of Steps Taken Thus Far...................      133
Number of Function Evaluations...................      317
Number of Jacobian Evaluations...................        9
Number of Jacobian Factorizations................        9
```

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:



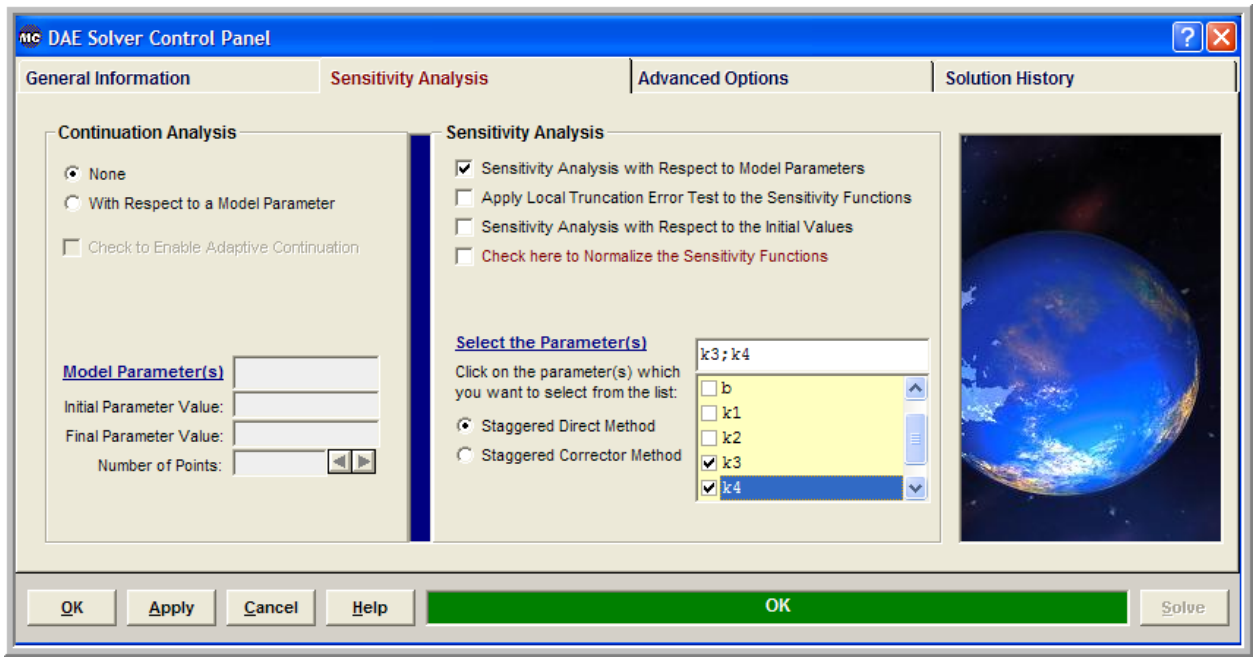
In this window fist we click ***Load*** to load the numerical results. Then in the **Graph What** group we select the **x-variable** (here **Time**) and the **y-variable** (here all the state variables by dragging the mouse) and click ***Graph***. You should see the graph that appears above. You may 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.

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.....................     5
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   5.00000E-01   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00   0.00000E+00
 4.76190E-01   7.26643E-01   0.00000E+00   0.00000E+00   2.36985E+00   1.29746E-01  -2.71933E-02
 9.52381E-01   1.02105E+00   0.00000E+00   0.00000E+00   4.73938E+00   2.52706E-01  -5.45917E-02
 1.42857E+00   1.37559E+00   0.00000E+00   0.00000E+00   7.19744E+00   3.66086E-01  -8.28280E-02
 1.90476E+00   1.76554E+00   0.00000E+00   0.00000E+00   9.60997E+00   4.48927E-01  -1.08684E-01
 2.38095E+00   2.15396E+00   0.00000E+00   0.00000E+00   1.16244E+01   4.63298E-01  -1.23715E-01
 2.85714E+00   2.50445E+00   0.00000E+00   0.00000E+00   1.27746E+01   3.70073E-01  -1.15223E-01
 3.33333E+00   2.79360E+00   0.00000E+00   0.00000E+00   1.27055E+01   1.65195E-01  -7.40161E-02
 3.80952E+00   3.01503E+00   0.00000E+00   0.00000E+00   1.14229E+01  -8.80989E-02  -7.98792E-03
 4.28571E+00   3.17513E+00   0.00000E+00   0.00000E+00   9.34992E+00  -2.89005E-01   5.63026E-02
 4.76190E+00   3.28612E+00   0.00000E+00   0.00000E+00   7.07743E+00  -3.79578E-01   9.50176E-02
 5.23810E+00   3.36085E+00   0.00000E+00   0.00000E+00   5.04896E+00  -3.73015E-01   1.04117E-01
 5.71429E+00   3.41019E+00   0.00000E+00   0.00000E+00   3.45281E+00  -3.14076E-01   9.35412E-02
 6.19048E+00   3.44233E+00   0.00000E+00   0.00000E+00   2.29312E+00  -2.40729E-01   7.48739E-02
 6.66667E+00   3.46310E+00   0.00000E+00   0.00000E+00   1.49259E+00  -1.73616E-01   5.56810E-02
 7.14286E+00   3.47643E+00   0.00000E+00   0.00000E+00   9.58161E-01  -1.20169E-01   3.94126E-02
 7.61905E+00   3.48497E+00   0.00000E+00   0.00000E+00   6.09209E-01  -8.08254E-02   2.69547E-02
 8.09524E+00   3.49043E+00   0.00000E+00   0.00000E+00   3.84757E-01  -5.32613E-02   1.79874E-02
 8.57143E+00   3.49390E+00   0.00000E+00   0.00000E+00   2.41862E-01  -3.45787E-02   1.17905E-02
 9.04762E+00   3.49612E+00   0.00000E+00   0.00000E+00   1.51535E-01  -2.22046E-02   7.62698E-03
 9.52381E+00   3.49753E+00   0.00000E+00   0.00000E+00   9.47195E-02  -1.41430E-02   4.88535E-03
 1.00000E+01   3.49843E+00   0.00000E+00   0.00000E+00   5.91078E-02  -8.95347E-03   3.10617E-03

EXIT DDAPLUS: SOLUTION FOUND

Number of Steps Taken Thus Far...................      137
Number of Function Evaluations...................     1388
Number of Jacobian Evaluations...................      139
Number of Jacobian Factorizations................      139
```

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*):

$C_1(t=10)=3.49863$	$\left.\frac{\partial C_1}{\partial k_3}\right\|_{t=10}=0.0$	$\left.\frac{\partial C_1}{\partial k_4}\right\|_{t=10}=0.0$
$C_2(t=10)=0.05911$	$\left.\frac{\partial C_2}{\partial k_3}\right\|_{t=10}=-0.0089535$	$\left.\frac{\partial C_2}{\partial k_4}\right\|_{t=10}=0.0031062$

Notice that the number of Jacobian evaluations (139) is the same as the number of Jacobian factorizations (139), since we chose the *Staggered Direct Method*.