Athena Visual Studio

Process Modeling, Parameter Estimation and Optimization

Programming Guide

Athena Visual Software, Inc.
# Table of Contents

ATHENA VISUAL STUDIO ........................................................................................................ 1

1.0 INTRODUCTION .................................................................................................................. 6

2.0 ANALYSIS OF NONLINEAR ALGEBRAIC MODELS ................................................................. 7

2.1 CONTINUOUS STIRRED TANK REACTOR (CSTR) MODELING ........................................ 7
2.2 IMPLEMENTATION IN ATHENA VISUAL STUDIO .............................................................. 10
   2.2.1 Writing the Source Code for Algebraic Models .......................................................... 11
   2.2.1.1 Data Section ........................................................................................................ 11
   2.2.1.2 Declaration of Variables in Athena Visual Studio .............................................. 12
   2.2.1.3 Initial Conditions Section ................................................................................... 14
   2.2.1.4 Model Equations Section .................................................................................... 15
   2.2.2 Entering the Information about the DDAPLUS Solver ............................................. 16
   2.2.3 Saving and Running your Model ............................................................................. 17
   2.2.4 Numerical Results ................................................................................................... 17
   2.2.5 Sensitivity Analysis Calculations .......................................................................... 18
   2.2.6 Auxiliary Variables .................................................................................................. 19
   2.2.7 Homotopy or Continuation Analysis ...................................................................... 21
   2.2.8 State Variable Names .............................................................................................. 22

2.3 ANALYSIS OF NONLINEAR INITIAL VALUE PROBLEMS ................................................... 24

3.1 BATCH CONVERSION OF GLUCOSE TO GLUCONIC ACID .................................................. 24
3.2 IMPLEMENTATION IN ATHENA VISUAL STUDIO .............................................................. 27
   3.2.1 Writing the Source Code for Initial-Value Problems ................................................ 28
   3.2.1.1 Data Section ........................................................................................................ 28
   3.2.1.2 Declaration of Variables in Athena Visual Studio .............................................. 29
   3.2.1.3 Initial Conditions Section ................................................................................... 31
   3.2.1.4 Model Equations Section .................................................................................... 31
   3.2.2 Entering the Information about the DDAPLUS Solver ............................................. 32
   3.2.3 Saving and Running your Model ............................................................................. 33
   3.2.4 Numerical Results ................................................................................................... 34
   3.2.5 Solution Graphs ....................................................................................................... 35
   3.2.6 Sensitivity Analysis Calculations .......................................................................... 36
   3.2.7 Implicit Auxiliary Variables .................................................................................... 39
   3.2.8 Explicit Auxiliary Variables ..................................................................................... 41
   3.2.9 Integration with Implicit or Explicit Discontinuities ................................................. 43
   3.2.10 Root Finding and Implicit Discontinuities ............................................................... 45
   3.2.11 Changing Initial Conditions at Discontinuities ...................................................... 47

4.0 ANALYSIS OF BOUNDARY VALUE PROBLEMS .................................................................... 48

4.1 REACTOR MODELING WITH AXIAL DISPERSION .................................................................. 49

© Athena Visual Software, Inc. 2
4.2 IMPLEMENTATION IN ATHENA VISUAL STUDIO .......................................................... 51
  4.2.1 Writing the Source Code for Boundary Value Problems ........................................ 52
    4.2.1.1 Data Section ........................................................................................................ 52
    4.2.1.2 Declaration of Variables ..................................................................................... 53
    4.2.1.3 Initial Conditions ................................................................................................. 55
    4.2.1.4 Model Equations ................................................................................................. 55
    4.2.1.5 Boundary Conditions ......................................................................................... 56
  4.2.2 Entering the Information about the PDAPLUS Solver ........................................... 57
  4.2.3 Saving and Running ................................................................................................. 59
  4.2.4 Numerical Results .................................................................................................. 60
  4.2.5 Graphical Results ................................................................................................... 61
  4.2.6 State Variable Names ............................................................................................. 62
  4.2.7 Continuation Analysis ............................................................................................ 63
  4.2.8 Implicit Auxiliary Variables ................................................................................... 65

5.0 ANALYSIS OF PARTIAL DIFFERENTIAL EQUATIONS ................................................ 67
  5.1 DIFFUSION IN NYLON 12 FOOD PACKAGING .......................................................... 68
  5.2 IMPLEMENTATION IN ATHENA VISUAL STUDIO .................................................... 70
    5.2.1 Writing Source Code for Partial Differential Equations ........................................ 71
      5.2.1.1 Data Section ....................................................................................................... 71
      5.2.1.2 Declaration of Variables in Athena Visual Studio .............................................. 72
      5.2.1.3 Initial Conditions Section ................................................................................... 74
      5.2.1.4 Model Equations Section ................................................................................... 74
      5.2.1.5 Boundary Conditions Section .......................................................................... 75
      5.2.1.6 Coefficient Matrix Section ............................................................................... 75
    5.2.2 Entering the Information about the PDAPLUS Solver ......................................... 76
    5.2.3 Saving and Running your Model ........................................................................... 78
    5.2.4 Numerical and Graphical Results ......................................................................... 79
    5.2.5 Explicit Auxiliary Variables .................................................................................. 81

6.0 ATHENA VISUALKINETICS™ ................................................................................... 83
  6.1 MODELING OF REACTION NETWORKS: BZ OXIDATION .......................................... 84
  6.2 IMPLEMENTATION IN ATHENA VISUAL STUDIO .................................................... 85
    6.2.1 Component Selection .............................................................................................. 85
    6.2.2 Reaction Mechanism ............................................................................................ 86
    6.2.3 Reaction Rate Data ............................................................................................... 87
    6.2.4 Operating Data ..................................................................................................... 88
    6.2.5 Auxiliary File for the Subroutine Rates ................................................................. 88
    6.2.6 Creating the Subroutine Rates ............................................................................. 90
    6.2.7 Calling the Subroutine Rates ............................................................................... 90
  6.3 REACTOR MODELING: BZ OXIDATION IN BATCH REACTOR .................................. 91
  6.4 IMPLEMENTATION IN ATHENA VISUAL STUDIO .................................................... 92
    6.4.1 Component Selection ............................................................................................. 92
    6.4.2 Reaction Mechanism ............................................................................................ 93
    6.4.3 Reaction Rate Data ............................................................................................... 94
8.5.6 Graphical Results........................................................................................................ 144

9.0 MODEL DISCRIMINATION AND OPTIMAL DESIGN............................................... 145

9.1 DISCRIMINATION AND DESIGN CRITERIA .............................................................. 145
9.2 REFERENCES .............................................................................................................. 146
9.3 MODEL DISCRIMINATION WITH AN EXPLICIT MODEL ........................................ 147
9.4 MODEL DISCRIMINATION IN CH$_3$OH CHEMISTRY ............................................. 148
9.5 IMPLEMENTATION IN ATHENA VISUAL STUDIO.................................................... 150
9.5.1 Writing Source Code for Estimation with Explicit Models..................................... 151
  9.5.1.1 Declaration of Variables in Athena Visual Studio.............................................. 152
  9.5.1.2 Data Section ...................................................................................................... 154
  9.5.1.3 Response Model ............................................................................................... 155
  9.5.1.4 Gradient Vector ............................................................................................... 156
9.5.2 The Parameter Estimation Solver ............................................................................ 157
  9.5.2.1 The Adjustable Parameters ............................................................................. 158
  9.5.2.2 The Experimental Data .................................................................................. 159
9.5.3 Select the Models for Estimation .......................................................................... 160
9.5.4 Numerical Results ................................................................................................. 161
  9.5.4.1 Model Discrimination Results ........................................................................ 163
  9.5.5 Optimal Experimental Design ............................................................................. 164

10.0 NONLINEAR OPTIMIZATION PROBLEMS............................................................. 167

10.1 ALKYLATION PROCESS OPTIMIZATION ............................................................... 169
10.2 IMPLEMENTATION IN ATHENA VISUAL STUDIO.................................................... 171
  10.2.1 Writing Source Code for Optimization Problems ............................................. 172
    10.2.1.1 Declaration of Variables in Athena Visual Studio........................................ 173
    10.2.1.2 Data Section ................................................................................................ 175
    10.2.1.3 Optimization Model .................................................................................... 175
  10.2.2 Entering the Information for the Optimization Solver ......................................... 176
  10.2.3 Entering the Information for the Independent Variables ...................................... 177
  10.2.4 Entering the Information for the Dependent Variables ........................................ 178
  10.2.5 Saving and Running your Model ......................................................................... 179
  10.2.6 Numerical Results .............................................................................................. 180

INDEX ..................................................................................................................................... 181
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 [Athena Visual Studio](#), 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.

Dr. Michael Caracotsios
Naperville, ILLINOIS
USA
2.0 Analysis of Nonlinear Algebraic Models

Nonlinear Algebraic Models that can be solved with Athena Visual Studio take the form $F(u; \theta) = 0$ where $u$ is a vector of unknowns, such as the temperature, pressure and composition, and $\theta$ 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 Athena Visual Studio powerful damped Newton algorithm which is encoded in the DDPLUS solver. The example problem below has been created to illustrate the functionality of Athena Visual Studio 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:

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

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 Athena Visual Studio notation:
Table 1. CSTR Material Balance Equations

<table>
<thead>
<tr>
<th>Equation</th>
<th>Description</th>
<th>Parameter List</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{A0} + \tau(-r_1 - r_2) - C_A = 0$</td>
<td>$r_1 = \frac{k_1 K_A}{1 + K_A C_A + K_B C_B} \left( C_A - \frac{C_B}{K_{eq}} \right)$</td>
<td>$U(1) := C_A$</td>
<td></td>
</tr>
<tr>
<td>$C_{B0} + \tau(r_1 - r_2) - C_B = 0$</td>
<td>$r_2 = \frac{k_1 K_B}{1 + K_A C_A + K_B C_B} C_B$</td>
<td>$U(2) := C_B$</td>
<td></td>
</tr>
<tr>
<td>$C_{C0} + \tau r_2 - C_C = 0$</td>
<td>$r_3 = \frac{k_1 K_A}{1 + K_A C_A + K_B C_B} C_A$</td>
<td>$U(3) := C_C$</td>
<td></td>
</tr>
<tr>
<td>$C_{D0} + \tau r_3 - C_D = 0$</td>
<td></td>
<td>$U(4) := C_D$</td>
<td></td>
</tr>
</tbody>
</table>

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:

<table>
<thead>
<tr>
<th>MODEL PARAMETERS</th>
<th>Description and Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\tau = 90.0$</td>
<td>Residence Time (s)</td>
</tr>
<tr>
<td>$C_{A0} = 0.35$</td>
<td>Feed Molar Concentration (mol/m3)</td>
</tr>
<tr>
<td>$k_1 = 0.01$</td>
<td>Reaction Rate Constant (mol/m3 s)</td>
</tr>
<tr>
<td>$k_2 = 0.01$</td>
<td>Reaction Rate Constant (mol/m3 s)</td>
</tr>
<tr>
<td>$k_3 = 0.001$</td>
<td>Reaction Rate Constant (mol/m3 s)</td>
</tr>
<tr>
<td>$K_A = \exp \left( \frac{35,000}{Rg T} - \frac{91}{Rg} \right)$</td>
<td>Adsorption Constant (m3/mol)</td>
</tr>
<tr>
<td>$K_B = \exp \left( \frac{20,000}{Rg T} - \frac{53}{Rg} \right)$</td>
<td>Adsorption Constant (m3/mol)</td>
</tr>
<tr>
<td>$K_{eq} = 4.29 \times 10^{-4} \frac{K_A}{K_B} \exp \left( \frac{30,000}{Rg T} \right)$</td>
<td>Equilibrium Constant(Unitless)</td>
</tr>
<tr>
<td>$R_g = 8.314$</td>
<td>Universal Gas Constant (J/mol K)</td>
</tr>
<tr>
<td>$T = 330.0$</td>
<td>Reactor Temperature (K)</td>
</tr>
</tbody>
</table>
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 Athena Visual Studio nomenclature):

$$
\begin{bmatrix}
U(1,1) & U(1,2) & U(1,3) \\
U(2,1) & U(2,2) & U(2,3) \\
U(3,1) & U(3,2) & U(3,3) \\
U(4,1) & U(4,2) & U(4,3)
\end{bmatrix}
\begin{bmatrix}
\frac{\partial C_A}{\partial k_1} & \frac{\partial C_A}{\partial k_2} \\
\frac{\partial C_B}{\partial k_1} & \frac{\partial C_B}{\partial k_2} \\
\frac{\partial C_C}{\partial k_1} & \frac{\partial C_C}{\partial k_2} \\
\frac{\partial C_D}{\partial k_1} & \frac{\partial C_D}{\partial k_2}
\end{bmatrix}
$$

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 Steady-State Isothermal CSTR sample
- Click OK
2.2 Implementation in Athena Visual Studio

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


- 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 Athena Visual Studio 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 Athena Visual Studio is used to enter all the data pertinent to the model. The data section also contains the declaration statements for all model variables, parameters and constants. This section, if used, must be the first one. The declaration of the model variables, parameters and constants must be done in accordance the Athena Visual Studio syntax rules detailed below:

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 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 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 Athena Visual Studio

**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 throughout all the modeling sections. The **Parameter** keyword is only allowed in the data section of the Athena Visual Studio modeling code. If it is used in the other modeling sections it will be ignored. You may view the generated Fortran code to see how the parser interprets the **Parameter** keyword.

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

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

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

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

```plaintext
@Initial Conditions
U(1)=Ca  ! [Ca]
U(2)=Cb  ! [Cb]
U(3)=Cc  ! [Cc]
U(4)=Cd  ! [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 known 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.

```plaintext
@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
```

© Athena Visual Software, Inc. 15
2.2.2 Entering the Information about the DDAPLUS Solver

It is now time to access the Athena Visual Studio 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

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:

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Algebraic Equations</td>
<td>4</td>
</tr>
<tr>
<td>Number of Sensitivity Parameters</td>
<td>0</td>
</tr>
<tr>
<td>Number of User Specified Iterations</td>
<td>30</td>
</tr>
</tbody>
</table>

**EXIT DDAPLUS: SOLUTION FOUND**

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>INITIAL GUESS</th>
<th>RESIDUALS</th>
<th>FINAL VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>U(1)</td>
<td>3.50000E-01</td>
<td>3.02720E-15</td>
<td>8.00051E-02</td>
</tr>
<tr>
<td>U(2)</td>
<td>0.00000E+00</td>
<td>-5.91150E-15</td>
<td>1.07519E-01</td>
</tr>
<tr>
<td>U(3)</td>
<td>0.00000E+00</td>
<td>3.10212E-15</td>
<td>1.37412E-01</td>
</tr>
<tr>
<td>U(4)</td>
<td>0.00000E+00</td>
<td>-2.59450E-16</td>
<td>2.50643E-02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Statistics</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Newton Iterations</td>
<td>5</td>
</tr>
<tr>
<td>Number of Function Evaluations</td>
<td>26</td>
</tr>
<tr>
<td>Number of Jacobian Evaluations</td>
<td>5</td>
</tr>
<tr>
<td>Number of Jacobian Factorizations</td>
<td>5</td>
</tr>
</tbody>
</table>
2.2.5 Sensitivity Analysis Calculations

Athena Visual Studio allows for convenient and efficient calculation of the first order sensitivity functions given by:

\[ W(\theta) = \frac{\partial u}{\partial \theta} \]

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.

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:

\[ W(\theta) = \frac{\partial u}{\partial \ln \theta} \]
Now choose **OK** or click **Apply**. From the **Build** menu select **Execute** (or **Hit F5**). You should see the following results:

<table>
<thead>
<tr>
<th>Number of Algebraic Equations</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sensitivity Parameters</td>
<td>2</td>
</tr>
<tr>
<td>Number of User Specified Iterations</td>
<td>30</td>
</tr>
</tbody>
</table>

**EXIT DDAPLUS: SOLUTION FOUND**

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

| 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:

<table>
<thead>
<tr>
<th>$C_A = 0.080005$</th>
<th>$\frac{\partial C_A}{\partial k_1} = -6.68166$</th>
<th>$\frac{\partial C_A}{\partial k_2} = -0.799267$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_B = 0.107519$</td>
<td>$\frac{\partial C_B}{\partial k_1} = 2.44028$</td>
<td>$\frac{\partial C_B}{\partial k_2} = -6.41993$</td>
</tr>
<tr>
<td>$C_C = 0.137412$</td>
<td>$\frac{\partial C_C}{\partial k_1} = 5.83854$</td>
<td>$\frac{\partial C_C}{\partial k_2} = 7.17137$</td>
</tr>
<tr>
<td>$C_D = 0.025064$</td>
<td>$\frac{\partial C_D}{\partial k_1} = -1.59715$</td>
<td>$\frac{\partial C_D}{\partial k_2} = 0.047833$</td>
</tr>
</tbody>
</table>

### 2.2.6 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 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.

$$F(5) = U(5) - \frac{C_{A_0} - C_A}{C_{A_0}} \times 100$$
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:

```plaintext
! 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:

<table>
<thead>
<tr>
<th>Number of Algebraic Equations</th>
<th>............... 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Sensitivity Parameters</td>
<td>............... 0</td>
</tr>
<tr>
<td>Number of User Specified Iterations</td>
<td>............... 30</td>
</tr>
</tbody>
</table>

**EXIT DDAPLUS: SOLUTION FOUND**

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>INITIAL GUESS</th>
<th>RESIDUALS</th>
<th>FINAL VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>U(1)</td>
<td>3.50000E-01</td>
<td>3.02720E-15</td>
<td>8.00051E-02</td>
</tr>
<tr>
<td>U(2)</td>
<td>0.00000E+00</td>
<td>-5.91150E-15</td>
<td>1.07519E-01</td>
</tr>
<tr>
<td>U(3)</td>
<td>0.00000E+00</td>
<td>3.10212E-15</td>
<td>1.37412E-01</td>
</tr>
<tr>
<td>U(4)</td>
<td>0.00000E+00</td>
<td>-2.59450E-16</td>
<td>2.50643E-02</td>
</tr>
<tr>
<td>U(5)</td>
<td>0.00000E+00</td>
<td>-8.27047E-14</td>
<td>7.71414E+01</td>
</tr>
</tbody>
</table>

| 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**),

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 Panel](image)

In this window first 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:
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:

```plaintext
@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) = CC0 - 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

Initial-Value Problems or Differential/Algebraic models in Athena Visual Studio take the form:

\[ E(t, u; \theta) \frac{du}{dt} = F(t, u; \theta) \]

where \( u \) is a state vector of unknown variables (usually temperature, pressure, and composition), \( \theta \) is a vector of known parameters pertinent to the process we are modeling and \( t \) is a time-like dimension. \( E(t, u; \theta) \) 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.
We wish to perform the following tasks:

- Plot the concentration of the reaction components as a function of time for \(0 \leq t \leq 10\ hr\)
- 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 \(X_{\text{Glu}}\) by introducing an auxiliary variable, \(U(5)\)
- Estimate the glucose \(X_{\text{Glu}}\) 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:

<table>
<thead>
<tr>
<th>MODEL PARAMETERS</th>
<th>INITIAL CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a = 0.9082)</td>
<td>(C_1(0) = 0.5)</td>
</tr>
<tr>
<td>(b = 1.011)</td>
<td>(C_2(0) = 0.0)</td>
</tr>
<tr>
<td>(k_1 = 0.95)</td>
<td>(C_3(0) = 0.0)</td>
</tr>
<tr>
<td>(k_2 = 3.5)</td>
<td>(C_4(0) = 50.0)</td>
</tr>
<tr>
<td>(k_3 = 18.00)</td>
<td></td>
</tr>
<tr>
<td>(k_4 = 37.50)</td>
<td></td>
</tr>
<tr>
<td>(k_5 = 1.10)</td>
<td></td>
</tr>
</tbody>
</table>
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):

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

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**
3.2 Implementation in Athena Visual Studio

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

- 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 $E(t,u;\theta)$ 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:

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 Athena Visual Studio project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```plaintext
! 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 Athena Visual Studio

**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 throughout 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.
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 $U(\cdot)$ 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 $F(\cdot)$ 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 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**)  
- Enter the solver parameters as shown below for our example

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:

<table>
<thead>
<tr>
<th>TIME</th>
<th>U(1)</th>
<th>U(2)</th>
<th>U(3)</th>
<th>U(4)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00000E+00</td>
<td>5.60000E-01</td>
<td>1.28000E+00</td>
<td>1.60000E-01</td>
<td>4.50000E+01</td>
</tr>
<tr>
<td>4.76190E-01</td>
<td>8.06528E-01</td>
<td>3.30736E+00</td>
<td>1.35857E+00</td>
<td>4.18498E+01</td>
</tr>
<tr>
<td>9.52381E-01</td>
<td>1.12023E+00</td>
<td>5.45140E+00</td>
<td>3.64626E+00</td>
<td>3.75817E+01</td>
</tr>
<tr>
<td>1.42857E+00</td>
<td>1.48852E+00</td>
<td>7.69245E+00</td>
<td>7.08777E+00</td>
<td>3.21560E+01</td>
</tr>
<tr>
<td>1.90476E+00</td>
<td>1.88213E+00</td>
<td>9.80013E+00</td>
<td>1.16829E+01</td>
<td>2.58060E+01</td>
</tr>
<tr>
<td>2.38095E+00</td>
<td>2.26274E+00</td>
<td>1.13652E+01</td>
<td>1.72601E+01</td>
<td>1.91027E+01</td>
</tr>
<tr>
<td>2.85714E+00</td>
<td>2.59678E+00</td>
<td>1.19523E+01</td>
<td>2.34172E+01</td>
<td>1.28558E+01</td>
</tr>
<tr>
<td>3.33333E+00</td>
<td>2.86590E+00</td>
<td>1.29310E+01</td>
<td>2.95714E+01</td>
<td>7.81295E+00</td>
</tr>
<tr>
<td>3.80952E+00</td>
<td>3.06818E+00</td>
<td>9.75887E+00</td>
<td>3.51337E+01</td>
<td>4.31537E+00</td>
</tr>
<tr>
<td>4.28571E+00</td>
<td>3.21240E+00</td>
<td>7.68674E+00</td>
<td>3.97113E+01</td>
<td>2.20720E+00</td>
</tr>
<tr>
<td>4.76190E+00</td>
<td>3.31141E+00</td>
<td>5.64883E+00</td>
<td>4.31938E+01</td>
<td>1.06986E+00</td>
</tr>
<tr>
<td>5.23810E+00</td>
<td>3.37764E+00</td>
<td>3.94536E+00</td>
<td>4.56888E+01</td>
<td>5.01224E-01</td>
</tr>
<tr>
<td>5.71429E+00</td>
<td>3.42116E+00</td>
<td>2.65892E+00</td>
<td>4.74006E+01</td>
<td>2.30068E-01</td>
</tr>
<tr>
<td>6.19048E+00</td>
<td>3.44944E+00</td>
<td>1.74831E+00</td>
<td>4.85402E+01</td>
<td>1.04339E-01</td>
</tr>
<tr>
<td>6.66667E+00</td>
<td>3.46767E+00</td>
<td>1.13022E+00</td>
<td>4.92832E+01</td>
<td>4.69835E-02</td>
</tr>
<tr>
<td>7.14286E+00</td>
<td>3.47936E+00</td>
<td>7.22125E-01</td>
<td>4.97608E+01</td>
<td>2.10670E-02</td>
</tr>
<tr>
<td>7.61905E+00</td>
<td>3.48684E+00</td>
<td>4.57634E-01</td>
<td>5.00647E+01</td>
<td>9.42209E-03</td>
</tr>
<tr>
<td>8.09524E+00</td>
<td>3.49162E+00</td>
<td>2.88366E-01</td>
<td>5.02567E+01</td>
<td>4.20744E-03</td>
</tr>
<tr>
<td>8.57143E+00</td>
<td>3.49666E+00</td>
<td>1.80978E-01</td>
<td>5.03775E+01</td>
<td>1.87750E-03</td>
</tr>
<tr>
<td>9.04762E+00</td>
<td>3.49660E+00</td>
<td>1.13260E-01</td>
<td>5.04532E+01</td>
<td>8.37320E-04</td>
</tr>
<tr>
<td>9.52381E+00</td>
<td>3.49784E+00</td>
<td>7.07377E-02</td>
<td>5.05005E+01</td>
<td>3.73302E-04</td>
</tr>
<tr>
<td>1.00000E+01</td>
<td>3.49863E+00</td>
<td>4.41169E-02</td>
<td>5.05301E+01</td>
<td>1.66392E-04</td>
</tr>
</tbody>
</table>

**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 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.

### 3.2.6 Sensitivity Analysis Calculations

Athena Visual Studio allows for convenient and efficient calculation of the first order sensitivity functions given by:

\[
W(t; \theta) = \frac{\partial u(t)}{\partial \theta}
\]

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:

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

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):

<table>
<thead>
<tr>
<th>TIME</th>
<th>U(1,1)</th>
<th>U(1,2)</th>
<th>U(1,3)</th>
<th>U(2,1)</th>
<th>U(2,2)</th>
<th>U(2,3)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00000E+00</td>
<td>5.60000E-01</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.28000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>4.76190E-01</td>
<td>8.06528E-01</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>3.30736E+00</td>
<td>1.36996E-01</td>
<td>-3.05263E-02</td>
</tr>
<tr>
<td>9.52381E-01</td>
<td>1.12023E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>5.45140E+00</td>
<td>2.61417E-01</td>
<td>-6.02549E-02</td>
</tr>
<tr>
<td>1.42857E+00</td>
<td>1.48852E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>7.69245E+00</td>
<td>3.66565E-01</td>
<td>-8.89308E-02</td>
</tr>
<tr>
<td>1.90476E+00</td>
<td>1.86213E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>9.80014E+00</td>
<td>4.26869E-01</td>
<td>-1.11581E-01</td>
</tr>
<tr>
<td>2.38095E+00</td>
<td>2.26274E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.13652E+00</td>
<td>4.04263E-01</td>
<td>-1.17913E-01</td>
</tr>
<tr>
<td>2.85714E+00</td>
<td>2.59678E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.19523E+00</td>
<td>2.72800E-01</td>
<td>-9.60814E-02</td>
</tr>
<tr>
<td>3.33333E+00</td>
<td>2.86590E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.13510E+00</td>
<td>5.67222E-01</td>
<td>-4.39940E-02</td>
</tr>
<tr>
<td>3.80952E+00</td>
<td>3.06818E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>9.75868E+00</td>
<td>-1.61601E+00</td>
<td>2.07664E-02</td>
</tr>
<tr>
<td>4.28571E+00</td>
<td>3.21240E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>7.68673E+00</td>
<td>-3.00792E+00</td>
<td>7.14077E-02</td>
</tr>
<tr>
<td>4.76190E+00</td>
<td>3.31141E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>5.64883E+00</td>
<td>-3.40144E-01</td>
<td>9.40931E-02</td>
</tr>
<tr>
<td>5.23810E+00</td>
<td>3.37764E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>3.94536E+00</td>
<td>-3.09819E+00</td>
<td>9.28102E-02</td>
</tr>
<tr>
<td>5.71429E+00</td>
<td>3.42116E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>2.65891E+00</td>
<td>-2.49443E-01</td>
<td>7.85921E-02</td>
</tr>
<tr>
<td>6.19048E+00</td>
<td>3.44944E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.74831E+00</td>
<td>-1.85871E+00</td>
<td>6.06293E-02</td>
</tr>
<tr>
<td>6.66667E+00</td>
<td>3.46767E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.13022E+00</td>
<td>-1.31582E+00</td>
<td>4.40031E-02</td>
</tr>
<tr>
<td>7.14286E+00</td>
<td>3.47936E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>7.22124E-01</td>
<td>-8.99269E-02</td>
<td>3.60301E-02</td>
</tr>
<tr>
<td>7.61050E+00</td>
<td>3.48684E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>4.57633E-01</td>
<td>-5.99486E-02</td>
<td>2.07020E-02</td>
</tr>
<tr>
<td>8.09524E+00</td>
<td>3.49162E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>2.88366E-01</td>
<td>-3.92529E-02</td>
<td>1.36972E-02</td>
</tr>
<tr>
<td>8.57143E+00</td>
<td>3.49466E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.80978E-01</td>
<td>-2.53661E-02</td>
<td>8.92290E-03</td>
</tr>
<tr>
<td>9.04762E+00</td>
<td>3.49660E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.13260E-01</td>
<td>-1.62333E-02</td>
<td>5.74467E-03</td>
</tr>
<tr>
<td>9.52381E+00</td>
<td>3.49784E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>7.07379E-02</td>
<td>-1.03134E-02</td>
<td>3.66658E-03</td>
</tr>
<tr>
<td>1.00000E+00</td>
<td>3.49863E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>4.41171E-02</td>
<td>-6.51675E-03</td>
<td>2.35323E-03</td>
</tr>
</tbody>
</table>

EXIT DDAPLUS: SOLUTION FOUND

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):

<table>
<thead>
<tr>
<th>C₁ (t = 10) = 3.49863</th>
<th>( \frac{\partial C_1}{\partial k_4} ) (_{t=10} = 0.0 )</th>
<th>( \frac{\partial C_1}{\partial k_4} ) (_{t=10} = 0.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>C₂ (t = 8) = 0.441171</td>
<td>( \frac{\partial C_2}{\partial k_2} ) (_{t=10} = -0.00651675 )</td>
<td>( \frac{\partial C_2}{\partial k_4} ) (_{t=10} = 0.00232532 )</td>
</tr>
</tbody>
</table>

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

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:

<table>
<thead>
<tr>
<th>TIME</th>
<th>U (1)</th>
<th>U (2)</th>
<th>U (3)</th>
<th>U (4)</th>
<th>U (5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000E+00</td>
<td>5.6000E-01</td>
<td>1.2800E+00</td>
<td>1.6000E-01</td>
<td>4.5000E+01</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>4.7619E-01</td>
<td>8.0652E-01</td>
<td>3.3073E+00</td>
<td>1.3587E+00</td>
<td>4.1849E+01</td>
<td>7.0003E+00</td>
</tr>
<tr>
<td>9.5238E-01</td>
<td>1.1202E+00</td>
<td>5.4514E+00</td>
<td>3.6426E+00</td>
<td>3.7581E+01</td>
<td>1.6485E+01</td>
</tr>
<tr>
<td>1.4286E+00</td>
<td>1.4885E+00</td>
<td>7.6924E+00</td>
<td>7.0877E+00</td>
<td>3.2156E+01</td>
<td>2.8542E+01</td>
</tr>
<tr>
<td>1.9047E+00</td>
<td>1.8821E+00</td>
<td>9.8001E+00</td>
<td>1.1682E+01</td>
<td>2.5806E+01</td>
<td>4.2653E+01</td>
</tr>
<tr>
<td>2.3809E+00</td>
<td>2.2627E+00</td>
<td>1.1352E+01</td>
<td>1.7260E+01</td>
<td>1.9027E+01</td>
<td>5.7549E+01</td>
</tr>
<tr>
<td>2.8571E+00</td>
<td>2.5967E+00</td>
<td>1.1952E+01</td>
<td>2.3417E+01</td>
<td>1.2855E+01</td>
<td>7.1432E+01</td>
</tr>
<tr>
<td>3.3333E+00</td>
<td>2.8659E+00</td>
<td>1.1351E+01</td>
<td>2.9571E+01</td>
<td>7.8129E+01</td>
<td>8.2638E+01</td>
</tr>
<tr>
<td>3.8095E+00</td>
<td>3.0681E+00</td>
<td>9.7588E+00</td>
<td>3.5133E+01</td>
<td>4.3153E+01</td>
<td>9.0410E+01</td>
</tr>
<tr>
<td>4.2857E+00</td>
<td>3.2124E+00</td>
<td>7.6867E+00</td>
<td>3.9711E+01</td>
<td>2.2072E+01</td>
<td>9.5095E+01</td>
</tr>
<tr>
<td>4.7619E+00</td>
<td>3.3114E+00</td>
<td>5.6488E+00</td>
<td>4.3193E+01</td>
<td>1.0696E+01</td>
<td>9.7622E+01</td>
</tr>
<tr>
<td>5.2381E+00</td>
<td>3.3776E+00</td>
<td>3.9453E+00</td>
<td>4.5688E+01</td>
<td>5.0122E+01</td>
<td>9.8862E+01</td>
</tr>
<tr>
<td>5.7143E+00</td>
<td>3.4211E+00</td>
<td>2.6589E+00</td>
<td>4.7400E+01</td>
<td>2.3007E+01</td>
<td>9.9488E+01</td>
</tr>
<tr>
<td>6.1904E+00</td>
<td>3.4494E+00</td>
<td>1.7483E+00</td>
<td>4.8540E+01</td>
<td>1.0433E+01</td>
<td>9.9768E+01</td>
</tr>
<tr>
<td>6.6667E+00</td>
<td>3.4676E+00</td>
<td>1.1302E+00</td>
<td>4.9283E+01</td>
<td>4.6935E+02</td>
<td>9.9896E+01</td>
</tr>
<tr>
<td>7.1429E+00</td>
<td>3.4793E+00</td>
<td>7.2212E+00</td>
<td>4.9760E+01</td>
<td>2.1067E+02</td>
<td>9.9953E+01</td>
</tr>
<tr>
<td>7.6190E+00</td>
<td>3.4868E+00</td>
<td>4.5763E+00</td>
<td>5.0064E+01</td>
<td>9.4221E+03</td>
<td>9.9979E+01</td>
</tr>
<tr>
<td>8.0952E+00</td>
<td>3.4916E+00</td>
<td>2.8836E+00</td>
<td>5.0256E+01</td>
<td>4.2077E+03</td>
<td>9.9990E+01</td>
</tr>
<tr>
<td>8.5714E+00</td>
<td>3.4966E+00</td>
<td>1.8097E+00</td>
<td>5.0377E+01</td>
<td>1.8773E+03</td>
<td>9.9995E+01</td>
</tr>
<tr>
<td>9.0476E+00</td>
<td>3.4960E+00</td>
<td>1.3260E+00</td>
<td>5.0453E+01</td>
<td>8.3714E+04</td>
<td>9.9998E+01</td>
</tr>
<tr>
<td>9.5238E+00</td>
<td>3.4978E+00</td>
<td>7.0737E+00</td>
<td>5.0500E+01</td>
<td>3.7315E+04</td>
<td>9.9999E+01</td>
</tr>
<tr>
<td>1.0000E+01</td>
<td>3.4986E+00</td>
<td>4.4116E-02</td>
<td>5.0530E+01</td>
<td>1.6629E-04</td>
<td>9.9999E+01</td>
</tr>
</tbody>
</table>

EXIT DDAPLUS: SOLUTION FOUND

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 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, \( x_{\text{glucose}} \) and \( s_{\text{gluAcid}} \) that represent the glucose conversion and the selectivity of glucose to gluconic acid respectively; these auxiliary variables are therefore defined by the following equations:

\[
\begin{align*}
    x_{\text{glucose}} &= \frac{45.0 - U(4)}{45.0} \times 100.0 \\
    s_{\text{gluAcid}} &= \frac{U(3)}{U(4)}
\end{align*}
\]

The code below illustrates how to do the implementation:

```plaintext
@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:

<table>
<thead>
<tr>
<th>TIME</th>
<th>U(1)</th>
<th>U(2)</th>
<th>U(3)</th>
<th>U(4)</th>
<th>x_glucose</th>
<th>x_gluAcid</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00000E+00</td>
<td>5.60000E-01</td>
<td>1.28000E+00</td>
<td>1.60000E-01</td>
<td>4.50000E+01</td>
<td>0.00000E+00</td>
<td>3.55556E-03</td>
</tr>
<tr>
<td>6.25000E-01</td>
<td>8.97591E-01</td>
<td>3.96182E+00</td>
<td>1.95333E+00</td>
<td>4.06421E+01</td>
<td>9.68431E+00</td>
<td>4.80618E-02</td>
</tr>
<tr>
<td>1.25000E+00</td>
<td>1.34555E+00</td>
<td>6.8562E+00</td>
<td>5.6593E+00</td>
<td>3.43187E+01</td>
<td>2.37362E-01</td>
<td>1.64904E-01</td>
</tr>
<tr>
<td>1.87500E+00</td>
<td>1.85751E+00</td>
<td>9.67947E+00</td>
<td>1.13640E+01</td>
<td>2.62207E+01</td>
<td>4.17317E+01</td>
<td>4.33398E-01</td>
</tr>
<tr>
<td>2.50000E+00</td>
<td>2.35164E+00</td>
<td>1.16178E+01</td>
<td>1.87656E+01</td>
<td>1.74650E+01</td>
<td>6.11889E+00</td>
<td>1.07447E+00</td>
</tr>
<tr>
<td>3.12500E+00</td>
<td>2.75660E+00</td>
<td>1.17575E+01</td>
<td>2.69195E+01</td>
<td>9.83687E+00</td>
<td>7.81403E+00</td>
<td>2.73660E+00</td>
</tr>
<tr>
<td>3.75000E+00</td>
<td>3.04631E+00</td>
<td>9.99600E+00</td>
<td>3.44869E+01</td>
<td>4.66950E+00</td>
<td>8.96233E+00</td>
<td>7.38556E+00</td>
</tr>
<tr>
<td>4.37500E+00</td>
<td>3.23401E+00</td>
<td>5.27951E+00</td>
<td>4.04467E+01</td>
<td>1.93356E+00</td>
<td>9.57032E+00</td>
<td>2.09183E+01</td>
</tr>
<tr>
<td>5.00000E+00</td>
<td>3.34793E+00</td>
<td>4.74602E+00</td>
<td>4.45530E+01</td>
<td>7.34679E+00</td>
<td>9.83674E+00</td>
<td>6.06428E+01</td>
</tr>
<tr>
<td>5.62500E+00</td>
<td>3.41335E+00</td>
<td>2.86930E+00</td>
<td>4.71292E+01</td>
<td>2.66531E+01</td>
<td>9.94077E+00</td>
<td>1.76924E+02</td>
</tr>
<tr>
<td>6.25000E+00</td>
<td>3.45218E+00</td>
<td>1.65685E+00</td>
<td>4.86516E+01</td>
<td>9.44648E-02</td>
<td>9.7901E+00</td>
<td>5.15024E+02</td>
</tr>
<tr>
<td>6.87500E+00</td>
<td>3.47343E+00</td>
<td>9.30183E-01</td>
<td>4.95186E+01</td>
<td>3.3091E+02</td>
<td>9.99265E+00</td>
<td>1.49640E+03</td>
</tr>
<tr>
<td>7.50000E+00</td>
<td>3.48528E+00</td>
<td>5.13226E-01</td>
<td>5.00012E+01</td>
<td>1.15236E+02</td>
<td>9.99744E+00</td>
<td>4.33903E+03</td>
</tr>
<tr>
<td>8.12500E+00</td>
<td>3.49185E+00</td>
<td>2.80120E+01</td>
<td>5.02660E+01</td>
<td>4.00057E+03</td>
<td>9.99911E+01</td>
<td>1.25647E+04</td>
</tr>
<tr>
<td>8.75000E+00</td>
<td>3.49550E+00</td>
<td>1.51851E+01</td>
<td>5.04101E+01</td>
<td>1.38705E+03</td>
<td>9.99969E+01</td>
<td>3.63433E+04</td>
</tr>
<tr>
<td>9.37500E+00</td>
<td>3.49751E+00</td>
<td>8.19616E-02</td>
<td>5.04881E+01</td>
<td>4.80521E-04</td>
<td>9.99989E+01</td>
<td>1.05069E+05</td>
</tr>
<tr>
<td>1.00000E+01</td>
<td>3.49863E+00</td>
<td>4.41169E-02</td>
<td>5.05301E+01</td>
<td>1.66392E+04</td>
<td>9.99996E+01</td>
<td>3.03681E+05</td>
</tr>
</tbody>
</table>

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.
3.2.9 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:

\[
k_i = \begin{cases} 
0.950 & 0 \leq t \leq 5 \\
0.425 & 5 < t \leq 8 \\
0.000 & 8 < t \leq 10 
\end{cases}
\]

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. 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

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

\[
h_i\left(t^*, x, u; \theta\right) = 0 \quad i = 1, 2, \ldots m
\]

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 \( h_i\left(t^*\right) \)) 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 \( 0 \leq t \leq t^* \), 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:

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

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:

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

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 its 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 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 than enter the code that is indicated below:

```plaintext
@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 \(5 \leq t \leq 8\). 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.
4.0 Analysis of Boundary Value Problems

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:

\[ F(x, u_x, u_{xx}; \theta) = 0 \]

where \( u_x \) represents the first order derivative of the state vector \( u(\cdot) \) with respect to the dimension \( x \), and \( u_{xx} \) represents the second order derivative of the state vector \( u(\cdot) \) with respect to \( x \) (in the appropriate coordinate system). Boundary value models are ordinarily used to model steady-state reaction and diffusion. They can be solved using the Athena Visual Studio powerful damped Newton algorithm which is encoded in the PDAPLUS solver.
4.1 Reactor Modeling with Axial Dispersion

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

Chemistry:

\[ A + B \xrightarrow{r} C \]

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:

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

\[-v \frac{\partial C_B}{\partial z} + \frac{1}{P_{er}} \frac{\partial^2 C_B}{\partial z^2} = Da_1 \mathcal{R} = 0 \]

\[-v \frac{\partial T}{\partial z} + \frac{1}{P_{er}} \frac{\partial^2 T}{\partial z^2} + \Delta T_{abs} Da_2 \mathcal{R} = 0 \]

\[ z = 0 \quad C_B = C_{B0} \quad T = T_{ro} \]

\[ z = 1 \quad \frac{\partial C_B}{\partial z} |_{z=1} = \frac{\partial T}{\partial z} |_{z=1} = 0 \]
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:

\[ X_B = 100 \times \frac{C_{B_0} - C_B(z=1)}{C_{B_0}} \]

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

<table>
<thead>
<tr>
<th>Model Parameters and Physical Properties</th>
<th>Description and Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \gamma = 20.0 )</td>
<td>Dimensionless Activation Energy</td>
</tr>
<tr>
<td>( Da_r = 0.60 - 1.40 )</td>
<td>Damköhler number range</td>
</tr>
<tr>
<td>( Pe_{mr} = 196.0 )</td>
<td>Peclet number for mass dispersion</td>
</tr>
<tr>
<td>( Pe_{hr} = 42.0 )</td>
<td>Peclet number for heat dispersions</td>
</tr>
<tr>
<td>( U_{htc} = 160.0 )</td>
<td>Dimensionless heat transfer coefficient</td>
</tr>
<tr>
<td>( \Delta T_{adr} = 0.34 )</td>
<td>Dimensionless adiabatic temperature rise</td>
</tr>
<tr>
<td>( \nu = 0.5 )</td>
<td>Dimensionless fluid velocity</td>
</tr>
<tr>
<td>( \omega_h = 11.67 )</td>
<td>Dimensionless heat capacity</td>
</tr>
<tr>
<td>( C_{B_0} = 0.50 )</td>
<td>Dimensionless inlet reactant concentration</td>
</tr>
<tr>
<td>( T_{ro} = 0.95 )</td>
<td>Dimensionless inlet reactor temperature</td>
</tr>
</tbody>
</table>

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

- Open Athena Visual Studio
- From the File menu click New
- Select the Training Samples tab
- Select the Steady-State Reactor with Axial Dispersion sample
- Click OK
4.2 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 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 Athena Visual Studio is used to enter all the data pertinent to the model. The data section also contains the declaration statements for all model variables, parameters and constants. This section, if used, must be the first one. The declaration of the model variables, parameters and constants must be done in accordance the Athena Visual Studio syntax rules detailed below:

4.2.1.1 Data Section

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

```plaintext
! 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
```

© Athena Visual Software, Inc.
4.2.1.2 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 throughout 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:

\[
\begin{align*}
\text{\$Integer, Parameter::: } & \text{ dp=Kind(1.0D0)} \\
\text{\$Integer, Parameter::: } & \text{ sp=Kind(1.0)} \\
\text{\$Real(Kind=dp):: } & \text{ v1,v2} \\
\text{\$Real(Kind=sp), Dimension(3):: } & \text{ a1,a2} \\
\text{\$Integer::: } & \text{ I1, I2} \\
\text{\$Character(Len=3):: } & \text{ s2,s3} \\
\text{\$Character(Len=10), Dimension(2):: } & \text{ s1} \\
\text{\$Logical::: } & \text{ Done} \\
\text{\$Real(Kind=dp), Dimension::, Allocatable::: } & \text{ w}
\end{align*}
\]

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

```plaintext
@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 Athena Visual Studio solver for Boundary Value Problems in order to enter information about the system of equations we wish to solve and various other parameters that control the integration algorithm. To do that:

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

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

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 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**

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

- **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 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 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:

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.
Athena Visual Studio Programming Guide

4.2.7 Continuation Analysis

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

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

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

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

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

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

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

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

@Boundary Conditions
If(LEFT) Then
F(1)=U(1)-Cbo
F(2)=U(2)-Tro
Else
F(1)=Ux(1)
F(2)=Ux(2)
EndIf
F(3)=U(3)-(Cbo-U(1))/Cbo*100.0

From the Build menu select Execute (or Hit F5). From the View menu we select Solution Graphs and the following panel is displayed:
In this window fist we click Load to load the numerical results. Then in the Graph What group we select the x-variable (here Space), and the y-variable (here conversion, Conversion); then click Graph. You should see the graph that appears above. You may now click on the graph toolbar and modify the type, title, symbol, the style and miscellaneous other properties of the graph.
5.0 Analysis of Partial Differential Equations

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:

\[ E \frac{\partial u}{\partial t} = F(t, x, u_x, u_{xx}; \theta) \]

where \( u(\cdot) \) is a state vector of unknowns (usually temperature, pressure and composition), \( \theta \) is a vector of known parameters pertinent to the process we are modeling, \( u_x(\cdot) \) represents the first order derivative of the state vector \( u(\cdot) \) with respect to the dimension \( x \), and \( u_{xx}(\cdot) \) represents the second order derivative of the state vector \( u(\cdot) \) 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 Athena Visual Studio 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:

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

\[
\frac{\partial C_F}{\partial t} = -D_P \alpha_v \frac{\partial C_P}{\partial x} \bigg|_{x=L_P}, \quad \alpha_v = \frac{A}{V_F}
\]

\[
t = 0 \quad C_P = \rho C_{P_0} \quad C_F = 0
\]

\[
x = 0 \quad \frac{\partial C_P}{\partial x} = 0
\]

\[
x = L_P \quad C_P = K_{pF} C_F
\]

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:

<table>
<thead>
<tr>
<th>Model Parameters and Physical Properties</th>
<th>Description and Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>(D_P = 3.0 \times 10^{-14} \text{ m}^2 \text{ s}^{-1})</td>
<td>Diffusion coefficient in polymer</td>
</tr>
<tr>
<td>(K_{pF} = 1.4)</td>
<td>Partitioning coefficient</td>
</tr>
<tr>
<td>(L_P = 10^{-4} \text{ m})</td>
<td>Polymer thickness</td>
</tr>
<tr>
<td>(V_F = 10^{-4} \text{ m}^3)</td>
<td>Solution volume</td>
</tr>
<tr>
<td>(A = 2.0 \times 10^{-2} \text{ m}^2)</td>
<td>Contact area</td>
</tr>
<tr>
<td>(\rho_P = 1000.0 \frac{\text{ kg}}{\text{ m}^3})</td>
<td>Polymer density</td>
</tr>
<tr>
<td>(C_{P_0} = 2300.0 \frac{\text{ mg}}{\text{ kg}})</td>
<td>Migrant initial concentration</td>
</tr>
</tbody>
</table>
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:

\[ \frac{\partial u_1}{\partial t} = D_v \frac{\partial^2 u_1}{\partial \xi^2} \quad \frac{\partial u_2}{\partial t} = -D_s \frac{\partial u_1}{\partial \xi} \bigg|_{\xi=1} \]

\[ t = 0 \quad u_1 = 1 \quad u_2 = 0 \]

\[ \xi = 0 \quad \frac{\partial u_1}{\partial \xi} = 0 \]

\[ \xi = 1 \quad u_1 = K_{pF} u_2 \]

This example tutorial is already precoded in 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 Dynamic Diffusion with Equilibrium Adsorption sample
- Click OK
5.2 Implementation in Athena Visual Studio

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


- 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 \textbf{@Initial Conditions} is used to insert initial values for the state variables vector. The second section labeled \textbf{@Model Equations} is used to enter the model equations. The third section labeled \textbf{@Boundary Conditions} is used to enter the boundary conditions. The fourth section labeled \textbf{@Coefficient Matrix} is used to enter the diagonal elements of the \( E(\ ) \) 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:

5.2.1.1 Data Section

In the data section (which may not be needed and is not labeled by Athena Visual Studio) the user simply enters the problem data and various 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 Athena Visual Studio project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```plaintext
! 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
```

© Athena Visual Software, Inc.
5.2.1.2 Declaration of Variables in Athena Visual Studio

**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 throughout 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.
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 $F(\cdot)$ is reserved in the Athena environment to represent the values of these functions. The vector $Ux(\cdot)$ 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.

```plaintext
@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 $E(\cdot)$ matrix on the left and right boundary as well as in the interior of the space domain. The vector $E(\cdot)$ 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.

```plaintext
@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 Athena Visual Studio 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

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

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:

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.

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 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 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:

```plaintext
@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:

<table>
<thead>
<tr>
<th>TIME</th>
<th>SPACE</th>
<th>U(1)</th>
<th>U(2)</th>
<th>Flux</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
</tr>
<tr>
<td>1.08911E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>2.17822E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>3.26733E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>4.35644E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>5.44554E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>6.53465E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>7.62376E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>8.71287E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>9.80198E-01</td>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>1.00000E+00</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
<td>2.18160E-02</td>
<td></td>
</tr>
<tr>
<td>3.75000E+01</td>
<td>0.00000E+00</td>
<td>4.70675E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
</tr>
<tr>
<td>1.08911E-01</td>
<td>4.64598E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>2.17822E-01</td>
<td>4.45338E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>3.26733E-01</td>
<td>4.13471E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>4.35644E-01</td>
<td>3.69953E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>5.44554E-01</td>
<td>3.16090E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>6.53465E-01</td>
<td>2.53502E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>7.62376E-01</td>
<td>1.84075E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>8.71287E-01</td>
<td>1.09901E-01</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>9.80198E-01</td>
<td>3.22133E-02</td>
<td>0.00000E+00</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>1.00000E+00</td>
<td>1.37078E-02</td>
<td>1.37078E-02</td>
<td>1.53043E-04</td>
<td></td>
</tr>
<tr>
<td>7.50000E+01</td>
<td>0.00000E+00</td>
<td>1.85093E-01</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
</tr>
<tr>
<td>1.08911E-01</td>
<td>1.82962E-01</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>2.17822E-01</td>
<td>1.76057E-01</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>3.26733E-01</td>
<td>1.64694E-01</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>4.35644E-01</td>
<td>1.49179E-01</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>5.44554E-01</td>
<td>1.29908E-01</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>6.53465E-01</td>
<td>1.07676E-01</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>7.62376E-01</td>
<td>8.29386E-02</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>8.71287E-01</td>
<td>5.65142E-02</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>9.80198E-01</td>
<td>2.91992E-02</td>
<td>0.00000E+00</td>
<td>5.45159E-05</td>
<td></td>
</tr>
<tr>
<td>1.00000E+00</td>
<td>2.42014E-02</td>
<td>1.72867E-02</td>
<td>5.45159E-05</td>
<td></td>
</tr>
</tbody>
</table>

**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 **Athena Visual Studio** 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 Athena VisualKinetics™

VisualKinetics™ is an integrated tool within the Athena Visual Studio software environment, which allows scientists and engineers to simulate the dynamic behavior of homogeneous and heterogeneous chemical reactions. Chemical reactions may include but not limited to, microbial growth kinetics, pharmacokinetics, food preparation kinetics, enzymatic reactions, and combustion reactions and many more. In addition VisualKinetics™ allows the user to perform sensitivity analysis and compare simulation results with experimental data in order to estimate unknown parameters. Rigorous statistical methods based on Bayes’ theorem and implemented in VisualKinetics™ allow for model criticism and lack-of-fit analysis, rival model discrimination as well as optimal experimental design. More specifically, the following three tasks are implemented in VisualKinetics™:

Modeling of Complex Reaction Networks: Allows the user to design and build reaction network, by entering species, reactions, kinetic and adsorption parameters and miscellaneous other operating conditions either via a graphical user interface or a text file. Athena compiles the user information and creates a subroutine named by the user; this subroutine is written in Fortran 95 and it is callable form compatible environments.

Reactor Modeling: Allows the user to design and build reactor models, by entering species, reactions, kinetic and adsorption parameters and miscellaneous other operating conditions either via a graphical user interface or a text file. The reactor models available through the interface are Batch, Semi-Batch, Continuous stirred tank reactors, Plug Flow and Fixed bed reactors. The reactor models can be isothermal, adiabatic or non-isothermal.

Simulation: Once the kinetic model is implemented and the proper data entered, VisualKinetics™ generates the required interface to the Athena Visual Studio computational engines in order to perform various simulation studies and investigate the dynamic and steady-state behavior of the chemical species. In this mode the user can exploit the rich environment of Athena Visual Studio and perform sensitivity analysis and parametric continuation studies.

Statistical Analysis and Nonlinear Parameter Estimation: This unique feature of VisualKinetics™ permits the user to set up parameter estimation problems via the graphical interface. In this setup the user brings the experimental data, and easily selects the model responses and parameters that need to be estimated. VisualKinetics™ then creates the proper interface to Athena Visual Studio’s statistical analysis engines, such as weighted Least Squares and Bayesian Estimation techniques. A large number of tools are now available to the user, for statistical inferences on the model parameters, model discrimination and lack-of-fit as well optimal experimental design.
6.1 Modeling of Reaction Networks: BZ Oxidation

The production of maleic anhydride by the oxidation of benzene (BZ) in the presence of excess air is given by the following reaction mechanism:

\[
\begin{align*}
&\text{Benzene Oxidation: } C_6H_6 + 4.5O_2 \xrightarrow{k_1} C_4H_2O_3 + 2CO_2 + 2H_2O \\
&\text{Maleic Anhydride Cracking: } C_4H_2O_3 + 3O_2 \xrightarrow{k_2} 4CO_2 + 1H_2O \\
&\text{Benzene Cracking: } C_6H_6 + 7.5O_2 \xrightarrow{k_3} 6CO_2 + 3H_2O
\end{align*}
\]

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

<table>
<thead>
<tr>
<th>MODEL PARAMETERS</th>
<th>INITIAL CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k_1 = 4280.0 \times \exp\left(-\frac{12660}{T}\right) )</td>
<td>( C_6H_6(0) = 10.0 )</td>
</tr>
<tr>
<td>( k_2 = 70100.0 \times \exp\left(-\frac{15000}{T}\right) )</td>
<td>( C_4H_2O_3(0) = 0.0 )</td>
</tr>
<tr>
<td>( k_3 = 26.0 \times \exp\left(-\frac{10800}{T}\right) )</td>
<td>( O_2(0) = 40.0 )</td>
</tr>
<tr>
<td>( K_{C_6H_6} = 0.5 )</td>
<td>( CO_2(0) = 0.0 )</td>
</tr>
<tr>
<td>( )</td>
<td>( H_2O(0) = 0.0 )</td>
</tr>
</tbody>
</table>

We wish to develop a subroutine called Rates that would accept input by the user and return the component rates and the reaction rates as described in the Table below:

<table>
<thead>
<tr>
<th>SUBROUTINE RATES (VTEMP,VPRES,VCON,VRC,VRX)</th>
<th>INPUT ARGUMENTS</th>
<th>OUTPUT ARGUMENTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reaction Temperature</td>
<td>vTemp</td>
<td></td>
</tr>
<tr>
<td>Reaction Pressure</td>
<td>vPres</td>
<td></td>
</tr>
<tr>
<td>Molar Component Concentrations</td>
<td>vCon (Ncc)</td>
<td></td>
</tr>
<tr>
<td>Component Reaction Rates</td>
<td>vRc (Ncc)</td>
<td></td>
</tr>
<tr>
<td>Reaction Rates</td>
<td>vRx (Nrx)</td>
<td></td>
</tr>
</tbody>
</table>
6.2 Implementation in Athena Visual Studio

The following step-by-step process describes the Benzene Oxidation Example implementation in Athena Visual Studio

6.2.1 Component Selection

- Open Athena Visual Studio. From the *VisualKinetics* menu, choose *Reaction Rates*…
- Enter a *File name* in the *New Model File* Dialog box and then Click *Save*. The *Visual Kinetics Control Panel* appears. You are in the *Component Selection* tab:

  ![Visual Kinetics Control Panel](image)

  - **Search** for and **Add**… the reaction components \([\text{C}_6\text{H}_6, \text{C}_4\text{H}_2\text{O}_3, \text{O}_2, \text{CO}_2, \text{H}_2\text{O}]\). Click **New**… if you wish to add a user component that is not in the Athena Database.
  - When you complete the selection of the reaction components click **Submit**. This action activates the Reaction Mechanism and Reaction Rate Data tabs. The Operating Data tab also is enabled in order to enter miscellaneous component properties and the reaction temperature and pressure.
6.2.2 Reaction Mechanism

Select the Reaction Mechanism tab.

- Formulate and Add the chemical reactions by selecting the Reactants and the Products from the drop down lists. Select blank from the drop down list if you wish to delete a component from a particular reaction. Use the Insert, Modify and Remove command buttons to edit your reaction scheme.
- Modify, if necessary, the stoichiometric coefficients (as indicated in this example). By default Athena assumes elementary reactions and therefore the stoichiometric coefficients can be the integers 1, 2 or 3. Should you wish to modify these coefficients to accommodate overall reaction mechanisms, such as cracking to miscellaneous components, you may do so by exporting your model into a data file, modifying the reaction rates and then re-importing the mechanism back into Athena.
- Use the Clear command button if something went wrong and you wish to start from the beginning.
6.2.3 Reaction Rate Data

Select the Reaction Rate Data tab.

From the Reaction Rate Constant Input Options group, select the form of the reaction rate constants (in our example, Frequency Factor and E/R) and enter their numerical values. You may wish to enter values for the equilibrium constant if you have selected a reaction that is reversible; you may also wish to enter the adsorption terms for a heterogeneous reaction. Recall that the general form of the reaction rate constants, equilibrium constants and species adsorption constants in Athena is given by the following equations:

\[
\begin{align*}
    k &= k_b \exp\left[n \ln\left(\frac{T}{T_b}\right) + \frac{E}{RT_b} \left(1 - \frac{T_b}{T}\right)\right] \\
    k &= k_0 T^n \exp\left[-\frac{E}{RT}\right] \\
    K_{eq} &= K_{eq,b} \exp\left[\frac{\Delta G}{RT_b} \left(1 - \frac{T_b}{T}\right)\right] \\
    K_{eq} &= K_{eq,0} \exp\left[-\frac{\Delta G}{RT}\right] \\
    K_{ads} &= K_{ads,b} \exp\left[-\frac{\Delta H}{RT_b} \left(1 - \frac{T_b}{T}\right)\right] \\
    K_{ads} &= K_{ads,0} \exp\left[+\frac{\Delta H}{RT}\right]
\end{align*}
\]
Modify, if necessary, the reaction rate form (as indicated in this example) by clicking on the reaction rate you wish to modify, entering the corrections in the yellow text box and clicking **OK** to accept the modifications.

### 6.2.4 Operating Data

Select the **Operating Data** tab.

- Enter the Initial Concentrations of all chemical species of the reaction mixture.
- Enter, if necessary, the numerical values of the species adsorption constants.
- Enter the Reaction Temperature and Reaction Pressure

### 6.2.5 Auxiliary File for the Subroutine Rates

Select the **Component Selection** tab and click **Aux** to enter the necessary information in order to create the subroutine: **Subroutine Rates** (*vTemp,vPres,vCon,vRc,vRx)*.

When you click **Aux**, a window will appear with a textbox, where you must enter the information required to generate the Rates subroutine. In the very beginning of the textbox you must declare the subroutine arguments. Based on our selection of FORTRAN variables for the input and output arguments of this subroutine we enter the following code as indicated below:
In the code above the variables \texttt{Ncc} and \texttt{Nrx} have already been declared by Athena Visual Studio and they contain the numerical values of the number of components in the reaction mixture and the number of reactions. Similarly, the variables \texttt{Temp}, \texttt{Pres} and \texttt{V(Ncc)} are already declared and represent the reaction temperature, pressure and the component concentrations respectively. With the code then, as indicated above, the user values \texttt{vTemp}, \texttt{vPres} and \texttt{vCon} are passed to the local variables for the reaction rate execution. The code presented here, is only a sample. The experienced user will be able to modify the code, the argument list and write more specific code relevant to his/her situation. The statements here are meant to serve only as a simple example.

In the \texttt{@Solver Options} section, the user instructs Athena to execute a number of steps in order to create the reaction rates subroutine. The instructions are in the form of keywords. The keyword \texttt{Sub} informs Athena Visual Studio what is the name of the reaction rates subroutine (in our case: \texttt{Rates}) and what are the arguments of the subroutine (in our case: \texttt{vTemp, vPres, vCon, vRc, vRx}). The keyword \texttt{ClosedModel}, when given the value 1, instructs Athena to actually form the subroutine. If a value of zero is given Athena will simply form a Main program that calculates the reaction and component rates. Finally the keyword \texttt{Prt} when given the value -1, instruct Athena to suppress printing.

In the \texttt{@User Output} section, we simply pass the calculated reaction rates and component reactions rates from the local Athena variables to the user subroutine selected variables.

In the \texttt{@Function vRates} section, we enter the evaluation of the adsorption term. Notice that the Athena statements for the adsorption term are precedent by the \$ sign. This instructs the Athena parser to place these statements prior to the calculation of the reaction rates.
6.2.6 Creating the Subroutine Rates

You are now ready to create the reaction subroutine and use it in order to calculate the reaction rates and the component rates. Click OK, close the Visual Kinetics Control Panel and:

- From the File menu, choose Save. This action saves your model and creates the Fortran code.
- To view the Fortran code that you created from the View menu choose Fortran Code.

You may now choose to compile your subroutine to make sure that everything is OK; to do that: From the Build menu choose Compile

6.2.7 Calling the Subroutine Rates

If everything goes well close your current application, and create a new one in order to call your subroutine. The following is a sample of Athena code that calls the rates subroutine; these examples can be found in the C:\Program Files\Athena\Samples\VisualKinetics folder that was created during the installation of Athena Visual Studio.
6.3 Reactor Modeling: BZ Oxidation in Batch Reactor

The production of maleic anhydride by the oxidation of benzene (BZ) in the presence of excess air is given by the following reaction mechanism:

<table>
<thead>
<tr>
<th>Reaction</th>
<th>Reaction Equation</th>
<th>Rate Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benzene Oxidation</td>
<td>[ C_6H_6 + 4.5O_2 \rightarrow k \rightarrow C_4H_2O_3 + 2CO_2 + 2H_2O ]</td>
<td>[ r_1 = \frac{k_1 C_{C_6H_6}}{1 + K_{C_6H_6} C_{C_6H_6}} ]</td>
</tr>
<tr>
<td>Maleic Anhydride Cracking</td>
<td>[ C_4H_2O_3 + 3O_2 \rightarrow k \rightarrow 4CO_2 + 1H_2O ]</td>
<td>[ r_2 = \frac{k_2 C_{C_4H_2O_3}}{1 + K_{C_4H_2O_3} C_{C_4H_2O_3}} ]</td>
</tr>
<tr>
<td>Benzene Cracking</td>
<td>[ C_6H_6 + 7.5O_2 \rightarrow k \rightarrow 6CO_2 + 3H_2O ]</td>
<td>[ r_3 = \frac{k_3 C_{C_6H_6}}{1 + K_{C_6H_6} C_{C_6H_6}} ]</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>MODEL PARAMETERS</th>
<th>INITIAL CONDITIONS</th>
</tr>
</thead>
<tbody>
<tr>
<td>[ k_1 = 4280.0 \times \exp \left( -\frac{12660}{T} \right) ]</td>
<td>[ C_6H_6(0) = 10.0 ]</td>
</tr>
<tr>
<td>[ k_2 = 70100.0 \times \exp \left( -\frac{15000}{T} \right) ]</td>
<td>[ C_4H_2O_3(0) = 0.0 ]</td>
</tr>
<tr>
<td>[ k_3 = 26.0 \times \exp \left( -\frac{10800}{T} \right) ]</td>
<td>[ O_2(0) = 40.0 ]</td>
</tr>
<tr>
<td>[ K_{C_6H_6} = 0.5 ]</td>
<td>[ CO_2(0) = 0.0 ]</td>
</tr>
<tr>
<td>[ K_{C_4H_2O_3} = 0.5 ]</td>
<td>[ H_2O(0) = 0.0 ]</td>
</tr>
</tbody>
</table>

We wish to develop a reactor model called **VKBatchReactor** that would accept input by the user and perform a dynamic simulation of the process, yielding the species concentration profiles as a function of time.
6.4 Implementation in Athena Visual Studio

The following step-by-step process describes the Benzene Oxidation Batch Reactor Example implementation in Athena Visual Studio:

6.4.1 Component Selection

- Enter a File name in the New Model File Dialog box and then Click Save. The Visual Kinetics Control Panel appears. You are in the Component Selection tab.

- **Search** for and **Add...** the reaction components \([\text{C}_6\text{H}_6, \text{C}_4\text{H}_2\text{O}_3, \text{O}_2, \text{CO}_2, \text{H}_2\text{O}]\). Click **New...** if you wish to add a user component that is not in the Athena Database.
- When you complete the selection of the reaction components click **Submit**. This action activates the Reaction Mechanism and Reaction Rate Data tabs. The Operating Data tab also is enabled in order to enter miscellaneous component properties and the reaction temperature and pressure.
6.4.2 Reaction Mechanism

Select the Reaction Mechanism tab.

- Formulate and Add the chemical reactions by selecting the Reactants and the Products from the drop down lists. Select blank from the drop down list if you wish to delete a component from a particular reaction. Use the Insert, Modify and Remove command buttons to edit your reaction scheme.

- Modify, if necessary, the stoichiometric coefficients (as indicated in this example). By default Athena assumes elementary reactions and therefore the stoichiometric coefficients can be the integers 1, 2 or 3. Should you wish to modify these coefficients to accommodate overall reaction mechanisms, such as cracking to miscellaneous components, you may do so by exporting your model into a data file, modifying the reaction rates and then re-importing the mechanism back into Athena.

- Use the Clear command button if something went wrong and you wish to start from the beginning.
6.4.3 Reaction Rate Data

Select the **Reaction Rate Data** tab.

From the **Reaction Rate Constant Input Options** group, select the form of the reaction rate constants (in our example, Frequency Factor and E/R) and enter their numerical values. You may wish to enter values for the equilibrium constant if you have selected a reaction that is reversible; you may also wish to enter the adsorption terms for a heterogeneous reaction. Recall that the general form of the reaction rate constants, equilibrium constants and species adsorption constants in Athena is given by the following equations:

\[
\begin{align*}
  k &= k_b \exp \left[ n \ln \left( \frac{T}{T_b} \right) + \frac{E}{RT_b} \left( 1 - \frac{T_b}{T} \right) \right] \\
  k &= k_0 T^n \exp \left[ -\frac{E}{RT} \right] \\
  K_{eq} &= K_{eq,b} \exp \left[ \frac{\Delta G}{RT_b} \left( 1 - \frac{T_b}{T} \right) \right] \\
  K_{eq} &= K_{eq,0} \exp \left[ -\frac{\Delta G}{RT} \right] \\
  K_{ads} &= K_{ads,b} \exp \left[ -\frac{\Delta H}{RT_b} \left( 1 - \frac{T_b}{T} \right) \right] \\
  K_{ads} &= K_{ads,0} \exp \left[ +\frac{\Delta H}{RT} \right]
\end{align*}
\]
Modify, if necessary, the reaction rate form (as indicated in this example) by clicking on the reaction rate you wish to modify, entering the corrections in the yellow text box and clicking OK to accept the modifications.

### 6.4.4 Batch Reactor Data

Select the **Batch Reactor** tab.

- Enter the Initial Concentrations of all chemical species of the reaction mixture.
- Enter, if necessary, the numerical values of the species adsorption constants.
- Enter the Reaction Time, Temperature and Reaction Pressure
- Click OK
6.4.5 Simulate the Batch Reactor

You are now ready to save your Batch Reactor model and run it. 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**)

6.4.6 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:

<table>
<thead>
<tr>
<th>Time</th>
<th>( U(1) )</th>
<th>( U(2) )</th>
<th>( U(3) )</th>
<th>( U(4) )</th>
<th>( U(5) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.00000E+00</td>
<td>1.00000E+01</td>
<td>0.00000E+00</td>
<td>4.00000E+01</td>
<td>0.00000E+00</td>
<td>0.00000E+00</td>
</tr>
<tr>
<td>5.71429E+00</td>
<td>9.78822E+00</td>
<td>2.05853E+00</td>
<td>3.79727E+01</td>
<td>5.96931E-02</td>
<td>1.67225E-01</td>
</tr>
<tr>
<td>1.98764E+00</td>
<td>1.10339E+01</td>
<td>4.17147E+00</td>
<td>3.87542E+01</td>
<td>9.25789E-01</td>
<td>5.04082E+00</td>
</tr>
<tr>
<td>2.85714E+00</td>
<td>9.10339E+00</td>
<td>6.82739E+00</td>
<td>3.89641E+01</td>
<td>9.63134E-01</td>
<td>9.05299E-01</td>
</tr>
<tr>
<td>3.33333E+00</td>
<td>9.05000E+00</td>
<td>8.17241E+00</td>
<td>3.89748E+01</td>
<td>9.60104E-01</td>
<td>9.05299E-01</td>
</tr>
<tr>
<td>4.28571E+00</td>
<td>8.99691E+00</td>
<td>9.00500E+00</td>
<td>3.89748E+01</td>
<td>9.60104E-01</td>
<td>9.05299E-01</td>
</tr>
<tr>
<td>5.14286E+00</td>
<td>8.94359E+00</td>
<td>9.00500E+00</td>
<td>3.89748E+01</td>
<td>9.60104E-01</td>
<td>9.05299E-01</td>
</tr>
</tbody>
</table>
EXIT DDAPLUS: SOLUTION FOUND

Number of Steps Taken Thus Far.......................... 18
Number of Function Evaluations.......................... 72
Number of Jacobian Evaluations.......................... 6
Number of Jacobian Factorizations....................... 6

6.4.7 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 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 two of the state variables by holding the Ctrl key down and clicking with your mouse on the two variables) 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.
7.0 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 1805 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 \( \theta = \{ \theta_1, \theta_2, \ldots, \theta_p \} \) in a mathematical model of the form

\[
y_u = f_u(x_u; \theta) + \epsilon_u = f_u(\theta) + \epsilon_u \quad u = 1, \ldots, n
\]

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

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

This density times \( d\varepsilon \) is the probability that a replicate observation at \( x_u \) would fall within \( \pm d\varepsilon/2 \) of the given \( y_u \). The probability density for the full data vector \( y \) is the product of these independent functions:

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

To describe the expected relative precisions of observations at the various conditions \( x_u \) it is customary to assign a vector \( \omega \) of numerical weights

\[
\omega_u = \sigma_u^2 / \sigma_u^2 \quad u = 1, \ldots, n
\]  

in which \( \sigma_u^2 \) 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:

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

in which

\[
S(\theta) = \sum_{u=1}^{n} \left[ \sqrt{\omega_u} y_u - \sqrt{\omega_u} f_u(\theta) \right]^2
\]

\[
= \sum_{u=1}^{n} \left[ Y_u - F_u(\theta) \right]^2 = \sum_{u=1}^{n} E_u(\theta)^2
\]
Here the notations

\[ Y_u = \sqrt{\omega_u} y_u; \quad F_u(\theta) = \sqrt{\omega_u} f_u(\theta); \quad E_u(\theta) = Y_u - F_u(\theta) \]  

(8)

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

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

(9)

and that for a prospective vector \( Y_1 \ldots Y_n \) is

\[ p(Y|\theta, \sigma) = \left( \frac{2\pi\sigma}{\omega_u} \right)^{-n} \exp \left[ -\frac{S(\theta)}{2\sigma^2} \right]. \]  

(10)

When \( Y \) is given instead of \( \theta \) and \( \sigma \) we will call this function the likelihood, \( \ell(\theta, \sigma|Y) \).

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

\[ p(\theta, \sigma) \propto \sigma^{-1} \quad \text{for permitted values of } \theta \]
\[ 0 \quad \text{for forbidden values of } \theta \]  

(11)

consistent with the Jeffries prior. Multiplication of \( \ell(\theta, \sigma|Y) \) by \( p(\theta, \sigma) \) in accordance with Bayes’ Theorem then gives the posterior density for permitted values of \( \theta \) as

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

(12)

This probability density takes its maximum at the minimum sum of squares \( S(\theta) \) encountered in the permitted range of \( \theta \). In practice, we need not only this least-squares value (the \textit{point estimate}), but also various \textit{interval estimates}, regions of \textit{highest posterior density}, and integrals over part or all of the range of \( \theta \).
Another rationale for minimizing $S(\theta)$ is to maximize the likelihood function $\ell(\theta, \sigma | Y)$ obtained from Equation (10). This maximum likelihood approach, which was advocated by Fisher (1925), gives the same point estimate of $\theta$ as does the posterior density function.

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

$$\ell_i \leq \theta_i \leq u_i \quad i = 1, \ldots, p$$

(13)

for a model containing $p$ adjustable parameters.

Many methods are available for least-squares calculations. Models linear in $\theta$ 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 $S(\theta)$ is essential. The Gauss-Newton method is such a method and a variant of it has been implemented in Athena Visual Studio.

7.2 References


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.

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

7.4 Rate Constants Estimation in CH₃OH Chemistry

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

\[ CO + 2H₂ \rightarrow CH₃OH \]

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:

\[ \mathcal{R}_1 = \frac{k(T) K_1K_2^2 (P_c P_H^2 - P_M / K_{eq})}{(1 + K_1 P_c + K_2 P_H + K_3 P_M)^3} \]

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

\[ \log K_{eq} = \frac{3914}{T} - 7.536 \log T + 0.001766 \log T + 9.388 \]
Estimate the parameters \( \{k_B, E, K_1, K_2, K_3\} \)

Make parameter transformations by use of the \textbf{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:

<table>
<thead>
<tr>
<th>Initial Values of Parameters</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( \ln k_B = -5.0 )</td>
<td>( K_1 = 0.1 )</td>
</tr>
<tr>
<td>( \frac{E}{RT_B} = 10.0 )</td>
<td>( K_2 = 0.1 )</td>
</tr>
<tr>
<td>( T_B = 485.0 )</td>
<td>( K_3 = 0.1 )</td>
</tr>
</tbody>
</table>

The experimental data for this example are given below:

<table>
<thead>
<tr>
<th>Temp</th>
<th>PC</th>
<th>PH</th>
<th>PM</th>
<th>Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>84.00</td>
<td>10.00</td>
<td>4.789E-06</td>
</tr>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>56.30</td>
<td>10.00</td>
<td>3.418E-06</td>
</tr>
<tr>
<td>495.00</td>
<td>15.10</td>
<td>84.00</td>
<td>10.00</td>
<td>4.447E-06</td>
</tr>
<tr>
<td>495.00</td>
<td>15.10</td>
<td>56.30</td>
<td>10.00</td>
<td>3.495E-06</td>
</tr>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>84.00</td>
<td>2.50</td>
<td>7.174E-06</td>
</tr>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>56.30</td>
<td>2.50</td>
<td>5.379E-06</td>
</tr>
<tr>
<td>496.00</td>
<td>15.10</td>
<td>84.00</td>
<td>2.50</td>
<td>6.662E-06</td>
</tr>
<tr>
<td>496.00</td>
<td>15.10</td>
<td>56.30</td>
<td>2.50</td>
<td>5.052E-06</td>
</tr>
<tr>
<td>496.00</td>
<td>40.00</td>
<td>84.00</td>
<td>10.00</td>
<td>1.254E-06</td>
</tr>
<tr>
<td>475.00</td>
<td>40.00</td>
<td>56.30</td>
<td>10.00</td>
<td>8.350E-07</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>84.00</td>
<td>10.00</td>
<td>1.150E-06</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>56.30</td>
<td>10.00</td>
<td>9.170E-07</td>
</tr>
<tr>
<td>475.00</td>
<td>40.00</td>
<td>84.00</td>
<td>2.50</td>
<td>2.006E-06</td>
</tr>
<tr>
<td>475.00</td>
<td>40.00</td>
<td>56.30</td>
<td>2.50</td>
<td>1.507E-06</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>84.00</td>
<td>2.50</td>
<td>2.045E-06</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>56.30</td>
<td>2.50</td>
<td>1.443E-06</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.658E-06</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.784E-06</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.793E-06</td>
</tr>
<tr>
<td>500.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>7.490E-06</td>
</tr>
<tr>
<td>470.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>9.900E-07</td>
</tr>
<tr>
<td>480.00</td>
<td>25.00</td>
<td>70.00</td>
<td>15.00</td>
<td>1.739E-06</td>
</tr>
<tr>
<td>486.00</td>
<td>26.00</td>
<td>70.00</td>
<td>1.70</td>
<td>3.726E-06</td>
</tr>
<tr>
<td>486.00</td>
<td>48.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.648E-06</td>
</tr>
<tr>
<td>486.00</td>
<td>12.60</td>
<td>70.00</td>
<td>5.00</td>
<td>2.562E-06</td>
</tr>
<tr>
<td>486.00</td>
<td>26.00</td>
<td>92.10</td>
<td>5.00</td>
<td>3.686E-06</td>
</tr>
<tr>
<td>486.00</td>
<td>25.00</td>
<td>53.00</td>
<td>5.00</td>
<td>2.271E-06</td>
</tr>
</tbody>
</table>
7.5 Implementation in Athena Visual Studio

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

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

  ![Welcome: New Model Selection Panel](image)

- 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 $Y(\cdot)$ 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 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:. 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 Athena Visual Studio

**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 throughout 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 an implicit model for parameter estimation in the Athena Visual Studio 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 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 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 \( \text{Par}(\) ) is reserved to access the adjustable parameters, and the vector \( \text{Xu}(\) ) is reserved to access the experimental settings. As mentioned above the vector \( \text{Y}(\) ) is reserved to enter the responses. For example \( \text{Y}(1) \) represents the first measured response, \( \text{Y}(2) \) the second (if present) and so on. As mentioned above, \( \text{Xu}(\) ) is reserved for the experimental settings. For example \( \text{Xu}(1) \) represents the first setting, \( \text{Xu}(2) \) 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 $\mathbf{dY}(\cdot)$ is reserved in the Athena environment to represent these derivative elements. Thus $\mathbf{dY}(1,3)$ holds the derivative of the response $Y(1)$ with respect to the parameter $\text{Par}(3)$. 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 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.
Then choose **OK** and enter the source code for the selected derivatives as shown below:

```plaintext
@Gradient Vector
\[ dY(1,1) = Y(1) \]
\[ dY(1,2) = Y(1) \times (1.0 - \text{Tref}/\text{Temp}) \]
```

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 Athena Visual Studio 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

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**)
- 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:

![Parameter Estimation Solver Control Panel](image)

<table>
<thead>
<tr>
<th>Names</th>
<th>Initial Value</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
<th>Perturbation</th>
<th>Trust Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-10.00</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20.00</td>
<td>0.0</td>
<td>0.0</td>
<td>-0.001</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.01</td>
<td>-0.001</td>
<td>-0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.01</td>
<td>-0.001</td>
<td>-0.001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>-0.001</td>
<td>-0.001</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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)

![Parameter Estimation Solver Control Panel](image)

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:

<table>
<thead>
<tr>
<th>Number of Experiments</th>
<th>27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Parameters</td>
<td>5</td>
</tr>
<tr>
<td>Number of Responses</td>
<td>1</td>
</tr>
<tr>
<td>Number of Settings</td>
<td>4</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>OPTIMAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PARAMETER</td>
</tr>
<tr>
<td>PAR( 1)</td>
</tr>
<tr>
<td>PAR( 2)</td>
</tr>
<tr>
<td>PAR( 3)</td>
</tr>
<tr>
<td>PAR( 4)</td>
</tr>
<tr>
<td>PAR( 5)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>EVENT</th>
<th>OBSERVED</th>
<th>PREDICTED</th>
<th>RESIDUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4.7890E-06</td>
<td>5.0927E-06</td>
<td>-3.0373E-07</td>
</tr>
<tr>
<td>2</td>
<td>3.4180E-06</td>
<td>3.4141E-06</td>
<td>3.8616E-09</td>
</tr>
<tr>
<td>3</td>
<td>4.4470E-06</td>
<td>4.2071E-06</td>
<td>2.3986E-07</td>
</tr>
<tr>
<td>4</td>
<td>3.4950E-06</td>
<td>3.1181E-06</td>
<td>3.7688E-07</td>
</tr>
<tr>
<td>5</td>
<td>7.1740E-06</td>
<td>7.1222E-06</td>
<td>5.1815E-08</td>
</tr>
<tr>
<td>6</td>
<td>5.3790E-06</td>
<td>4.9984E-06</td>
<td>3.8058E-07</td>
</tr>
<tr>
<td>7</td>
<td>6.6620E-06</td>
<td>6.5932E-06</td>
<td>6.8788E-08</td>
</tr>
<tr>
<td>8</td>
<td>5.0520E-06</td>
<td>5.3324E-06</td>
<td>2.8040E-07</td>
</tr>
<tr>
<td>9</td>
<td>1.2540E-06</td>
<td>1.3934E-06</td>
<td>1.3940E-07</td>
</tr>
<tr>
<td>10</td>
<td>8.3500E-07</td>
<td>9.3725E-07</td>
<td>1.0225E-07</td>
</tr>
<tr>
<td>11</td>
<td>1.1500E-06</td>
<td>1.1570E-06</td>
<td>7.0059E-09</td>
</tr>
<tr>
<td>12</td>
<td>9.1700E-07</td>
<td>8.6525E-07</td>
<td>5.1752E-08</td>
</tr>
<tr>
<td>13</td>
<td>2.0050E-06</td>
<td>1.9442E-06</td>
<td>6.0819E-08</td>
</tr>
<tr>
<td>14</td>
<td>1.5070E-06</td>
<td>1.3656E-06</td>
<td>1.4143E-07</td>
</tr>
<tr>
<td>15</td>
<td>2.0450E-06</td>
<td>1.8021E-06</td>
<td>2.4293E-07</td>
</tr>
<tr>
<td>16</td>
<td>1.4430E-06</td>
<td>1.4607E-06</td>
<td>1.7668E-08</td>
</tr>
<tr>
<td>17</td>
<td>2.8580E-06</td>
<td>2.9767E-06</td>
<td>1.8690E-07</td>
</tr>
<tr>
<td>18</td>
<td>2.7840E-06</td>
<td>2.9767E-06</td>
<td>1.9269E-07</td>
</tr>
<tr>
<td>19</td>
<td>2.7930E-06</td>
<td>2.9767E-06</td>
<td>1.8369E-07</td>
</tr>
<tr>
<td>20</td>
<td>7.4900E-06</td>
<td>7.6282E-06</td>
<td>1.3818E-07</td>
</tr>
<tr>
<td>21</td>
<td>9.9000E-07</td>
<td>1.0916E-06</td>
<td>1.0159E-07</td>
</tr>
<tr>
<td>22</td>
<td>1.7390E-06</td>
<td>1.7522E-06</td>
<td>1.3198E-08</td>
</tr>
<tr>
<td>23</td>
<td>3.7250E-06</td>
<td>3.6249E-06</td>
<td>1.0010E-07</td>
</tr>
<tr>
<td>24</td>
<td>2.6460E-06</td>
<td>2.6532E-06</td>
<td>7.1546E-09</td>
</tr>
<tr>
<td>25</td>
<td>2.5520E-06</td>
<td>2.5005E-06</td>
<td>5.1482E-08</td>
</tr>
<tr>
<td>26</td>
<td>3.6060E-06</td>
<td>3.6315E-06</td>
<td>2.5525E-08</td>
</tr>
<tr>
<td>27</td>
<td>2.2710E-06</td>
<td>2.3007E-06</td>
<td>2.9742E-08</td>
</tr>
</tbody>
</table>

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 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 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 \textit{x-variable} and the Residuals as the \textit{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 \textit{x-variable} and the Residuals as the \textit{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 \textit{x-variable} and the Ordered Residuals as the \textit{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 \( \text{Par}(2) \) corresponds to the dimensionless activation energy defined as: \( \text{Par}(2) = \frac{E}{RT_{\text{ref}}} \). 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 \textbf{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: \( E \equiv \Phi(1) \), then the following equation defines this Phi function:

\[
\Phi(1) = \text{Par}(2) \times RT_{\text{ref}}
\]

To implement this concept:

- From the \textit{Model} menu choose \textit{Load Solver}. The \textbf{Parameter Estimation Solver Control Panel} appears:
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:

```plaintext
@Phi Functions
Ph(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**

<table>
<thead>
<tr>
<th>FUNCTIONS</th>
<th>PREDICTED 95% MARGINAL HPD INTERVALS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHI(1)</td>
<td>1.269799E+05</td>
</tr>
</tbody>
</table>

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):

\[
\begin{align*}
T &= 480 \text{ } K & P_C &= 30 \text{ bar} & P_H &= 70 \text{ bar} & P_M &= 6 \text{ bar} \\
T &= 600 \text{ } K & P_C &= 30 \text{ bar} & P_H &= 70 \text{ bar} & P_M &= 6 \text{ bar}
\end{align*}
\]

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**

<table>
<thead>
<tr>
<th>FUNCTIONS</th>
<th>PREDICTED</th>
<th>95% MARGINAL HPD INTERVALS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PHI( 1)</td>
<td>1.269799E+05</td>
<td>1.269799E+05 ± 6.433E+03</td>
</tr>
<tr>
<td>PHI( 2)</td>
<td>2.880970E+09</td>
<td>2.880970E+09 ± 4.584E+09</td>
</tr>
<tr>
<td>PHI( 3)</td>
<td>2.029647E-06</td>
<td>2.029647E-06 ± 9.782E-08</td>
</tr>
<tr>
<td>PHI( 4)</td>
<td>7.361967E-04</td>
<td>7.361967E-04 ± 2.074E-04</td>
</tr>
</tbody>
</table>

**NORMALIZED COVARIANCE MATRIX**

<table>
<thead>
<tr>
<th></th>
<th>1.000</th>
<th>0.990</th>
<th>-0.866</th>
<th>-0.863</th>
<th>1.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.990</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.866</td>
<td></td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.863</td>
<td></td>
<td></td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.996</td>
<td>0.985</td>
<td>-0.820</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 perform 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 $\Sigma$ from a multiresponse data set, along with the parameter vector $\theta$ of a predictive model. It is also possible, in simple cases, to avoid explicit calculation of $\Sigma$ and estimate only $\theta$. 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 $p(\theta|Y)$ for estimating a parameter vector $\theta$ from a full matrix $Y$ of multiresponse observations. The errors in the observations were assumed to be Normally distributed with an unknown full covariance matrix $\Sigma$. Michael Box and Norman Draper (1972) gave a corresponding density function for treating a data matrix $Y$ 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 $p(\theta|Y)$ found by Box and Draper (1965) is a power of the determinant $|v(\theta)|$. These authors used contour plots of $p(\theta|Y)$ 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 $|\mathbf{v}(\mathbf{\theta})|$ by search techniques, which proved to be slow and gave only point estimates of $\mathbf{\theta}$. Newton-like algorithms for minimization of $|\mathbf{v}(\mathbf{\theta})|$, and for interval estimation of $\mathbf{\theta}$, 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 $\mathbf{\theta}$ (and explicitly or implicitly the covariance matrix $\mathbf{\Sigma}$) in a model

$$Y_u = F_i(x_u, \mathbf{\theta}) + E_u \quad (u = 1, \ldots, n; i = 1, \ldots, m)$$

(5.2-1)

for the elements $Y_u = \sqrt{w_u} y_u$ of a multiresponse weighted data matrix $Y$. Each integer $u$ from 1 to $n$ denotes an independent event, in which $m_u \leq m$ responses are observed. A weight $w_u$ is considered given for each event to express its reliability relative to some standard. The weighted function $F_i(x_u, \mathbf{\theta}) := \sqrt{w_u} f_i(x_u, \mathbf{\theta})$ is an expectation model for response $i$ at the experimental design point $x_u$ and is assumed to be differentiable with respect to each parameter $\theta_r$.

The weighted errors $E_u := \sqrt{w_u} e_u$ are modeled by an $m$-dimensional Normal distribution, with expected values of zero and unknown covariances $\sigma_{ij} := E(E_{ui} E_{uj})$. Use of this distribution with Eq. (5.2-1) yields a density function $p(Y|\mathbf{\theta}, \mathbf{\Sigma})$ for observations to be taken at planned test conditions $\{x_1, \ldots, x_n\}$. 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 $\mathbf{\Sigma}$.
Once data $Y$ are available, the likelihood function $l(\theta, \Sigma|Y)$ is constructed in the manner of Fisher (1922) as the function $p(Y|\theta, \Sigma)$, with $Y$ now given whereas $\theta$ and $\Sigma$ are free to vary. Multiplication of the likelihood by a suitable prior density $p(\theta, \Sigma)$, in accordance with Bayes’ theorem gives the posterior density function $p(\theta, \Sigma|Y)$, which contains all current information on $\theta$ and $\Sigma$. These constructions are summarized below for several problem types.

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

$$p(Y|\theta, \Sigma) = \prod_{u=1}^{n} p(Y_u|\theta, \Sigma)$$

$$= |2\pi\Sigma|^{-n/2} \exp \left\{ -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \sigma^{ij} \sum_{u=1}^{n} \left[ Y_{ui} - F_i(x_u, \theta) \right] \left[ Y_{uj} - F_j(x_u, \theta) \right] \right\}$$

$$= |2\pi\Sigma|^{-n/2} \exp \left\{ -\frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \sigma^{ij} v_{ij}(\theta) \right\}$$

$$= |2\pi\Sigma|^{-n/2} \exp \left\{ -\frac{1}{2} \text{tr} \left[ \Sigma^{-1} v(\theta) \right] \right\}$$

The matrices $\Sigma^{-1}$ and $v(\theta)$ are symmetric, with elements $\sigma^{ij}$ and

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

respectively; also $\text{tr}$ denotes the trace of a matrix: $\text{tr} A = \Sigma_i a_{ii}$

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

$$l(\theta, \Sigma|Y) \propto |\Sigma|^{-n/2} \exp \left\{ -\frac{1}{2} \text{tr} \left[ \Sigma^{-1} v(\theta) \right] \right\}$$

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 $\Sigma$, provided that $v(\theta)$ is nonsingular over the
permitted range of $\theta$. In practice, $v(\theta)$ can always be made nonsingular by proper choice of the working responses; this is done automatically in the subroutine package GREGPLUS of the software Athena Visual Studio.

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

$$p(\Sigma) \propto |\Sigma|^{-(m+1)/2}, \quad (5.2-5)$$

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 $\sigma_{ij}$ for $i \geq j$ are all regarded as independent unknowns. A uniform prior density is assumed for $\theta$ over the range of appreciable $Y_{\theta \Sigma}$; 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

$$p(\theta, \Sigma|Y) \propto |\Sigma|^{-(n+m+1)/2} \exp\{-\frac{1}{2} \text{tr}[[\Sigma \cdot v(\theta)]^{2}]\} \quad (5.2-6)$$

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

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

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

as the marginal posterior density for the parameters of the expectation model. The mode of this function occurs at the minimum of the determinant $|v(\theta)|$ 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 $\Sigma$ at each value of $\theta$. This gives the modified posterior density function

$$\tilde{p}(\theta|Y) := p(\theta, \tilde{\Sigma}(\theta)|Y) \propto |v(\theta)|^{-(n+m+1)/2}$$

(5.2-8)

which has the same modal $\theta$ 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 $\theta$.

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 $\theta$ and $\Sigma$ for a sample problem. The modal $\theta$ occurs at the minimum of $|v(\theta)|$, and the modal covariance estimate is

$$\hat{\Sigma} = v(\hat{\theta})/(n + m + 1)$$

(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 $\tilde{L}(\theta)$ corresponds to $l(\theta, \tilde{\Sigma}(\theta)|Y)$ 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


8.3 Parameter Estimation with an Implicit Model

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 Athena Visual Studio 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 Athena Visual Studio 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 \[ \frac{dC_2}{dt} = \frac{k_3 C_4 C_2}{k_4 + C_4} - \alpha k_2 C_2 \]

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:

<table>
<thead>
<tr>
<th>Model Parameters</th>
<th>Athena Variables</th>
<th>Initial Conditions</th>
<th>Athena Variables</th>
<th>Nomenclature</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha = 0.9082 )</td>
<td>a</td>
<td>( C_1(0) = {0.56, 0.66, 0.80} )</td>
<td>U(1)</td>
<td>Cell Concentration</td>
</tr>
<tr>
<td>( \beta = 1.0110 )</td>
<td>b</td>
<td>( C_2(0) = {1.28, 1.28, 1.34} )</td>
<td>U(2)</td>
<td>Glunolactone Concentration</td>
</tr>
<tr>
<td>( {k_{1B}, E_{1B}} = {0.1, 0.0} )</td>
<td>k1b, E1b</td>
<td>( C_3(0) = {0.16, 0.16, 0.95} )</td>
<td>U(3)</td>
<td>Gluconic Acid Concentration</td>
</tr>
<tr>
<td>( {k_{2B}, E_{2B}} = {0.1, 0.0} )</td>
<td>k2b, E2b</td>
<td>( C_4(0) = {45.0, 48.0, 44.5} )</td>
<td>U(4)</td>
<td>Glucose Concentration</td>
</tr>
<tr>
<td>( {k_{3B}, E_{3B}} = {0.1, 0.0} )</td>
<td>k3b, E3b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( {k_{4B}, E_{4B}} = {0.1, 0.0} )</td>
<td>k4b, E4b</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( {k_{5B}, E_{5B}} = {0.1, 0.0} )</td>
<td>k5b, E5b</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The experimental data at different levels of temperature and reaction time for this example are:

<table>
<thead>
<tr>
<th>RunID</th>
<th>Temp, C</th>
<th>Time</th>
<th>y1o</th>
<th>y2o</th>
<th>y3o</th>
<th>y4o</th>
<th>y1</th>
<th>y2</th>
<th>y3</th>
<th>y4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>25.00</td>
<td>0.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>1.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>0.86</td>
<td>2.20</td>
<td>1.56</td>
<td>43.00</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>2.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>1.60</td>
<td>3.50</td>
<td>5.00</td>
<td>38.00</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>3.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>2.60</td>
<td>5.60</td>
<td>9.50</td>
<td>33.00</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>4.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>3.20</td>
<td>7.00</td>
<td>16.00</td>
<td>25.00</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>5.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>3.30</td>
<td>7.80</td>
<td>24.50</td>
<td>16.50</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>6.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>3.50</td>
<td>7.20</td>
<td>32.00</td>
<td>9.00</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>7.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>3.40</td>
<td>6.30</td>
<td>24.50</td>
<td>4.00</td>
</tr>
<tr>
<td>1</td>
<td>25.00</td>
<td>8.00</td>
<td>0.56</td>
<td>1.28</td>
<td>0.16</td>
<td>45.00</td>
<td>3.40</td>
<td>3.20</td>
<td>46.80</td>
<td>2.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>0.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>1.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>1.00</td>
<td>1.96</td>
<td>0.15</td>
<td>45.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>2.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>1.60</td>
<td>6.67</td>
<td>7.00</td>
<td>37.50</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>3.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>2.60</td>
<td>10.50</td>
<td>15.00</td>
<td>28.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>4.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>3.20</td>
<td>10.50</td>
<td>25.00</td>
<td>18.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>5.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>3.30</td>
<td>7.58</td>
<td>35.00</td>
<td>8.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>6.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>3.30</td>
<td>2.05</td>
<td>42.50</td>
<td>3.00</td>
</tr>
<tr>
<td>2</td>
<td>28.00</td>
<td>7.00</td>
<td>0.66</td>
<td>1.28</td>
<td>0.16</td>
<td>48.00</td>
<td>3.30</td>
<td>1.90</td>
<td>46.50</td>
<td>1.00</td>
</tr>
<tr>
<td>3</td>
<td>32.00</td>
<td>0.00</td>
<td>0.80</td>
<td>1.34</td>
<td>0.95</td>
<td>44.50</td>
<td>0.80</td>
<td>1.34</td>
<td>0.95</td>
<td>44.50</td>
</tr>
<tr>
<td>3</td>
<td>32.00</td>
<td>1.00</td>
<td>0.80</td>
<td>1.34</td>
<td>0.95</td>
<td>44.50</td>
<td>1.50</td>
<td>4.00</td>
<td>4.96</td>
<td>37.50</td>
</tr>
<tr>
<td>3</td>
<td>32.00</td>
<td>2.00</td>
<td>0.80</td>
<td>1.34</td>
<td>0.95</td>
<td>44.50</td>
<td>2.60</td>
<td>7.50</td>
<td>16.10</td>
<td>25.00</td>
</tr>
<tr>
<td>3</td>
<td>32.00</td>
<td>3.00</td>
<td>0.80</td>
<td>1.34</td>
<td>0.95</td>
<td>44.50</td>
<td>3.60</td>
<td>8.00</td>
<td>32.10</td>
<td>9.00</td>
</tr>
<tr>
<td>3</td>
<td>32.00</td>
<td>4.00</td>
<td>0.80</td>
<td>1.34</td>
<td>0.95</td>
<td>44.50</td>
<td>3.50</td>
<td>5.00</td>
<td>43.70</td>
<td>3.00</td>
</tr>
<tr>
<td>4</td>
<td>32.00</td>
<td>5.00</td>
<td>0.80</td>
<td>1.34</td>
<td>0.95</td>
<td>44.50</td>
<td>3.50</td>
<td>2.42</td>
<td>44.50</td>
<td>2.00</td>
</tr>
</tbody>
</table>
8.5 Implementation in Athena Visual Studio


- 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 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 with the Athena Visual Studio syntax rules shown below:
8.5.1.1 Declaration of Variables in Athena Visual Studio

**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 throughout 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:

```
$$\text{Integer, Parameter::: dp=Kind(1.0D0)}$
$$\text{Integer, Parameter::: sp=Kind(1.0)}$
$$\text{Real(Kind=dp):: v1,v2}$
$$\text{Real(Kind=sp), Dimension(3):: a1,a2}$
$$\text{Integer::: I1, I2}$
$$\text{Character(Len=3):: s2,s3}$
$$\text{Character(Len=10), Dimension(2):: s1}$
$$\text{Logical::: Done}$
$$\text{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 Athena Visual Studio 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 Athena Visual Studio project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```plaintext
! 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.

```plaintext
@Initial Conditions
U(1:4)=Xu(4:7)
```
The variable \( \mathbf{X}_u(\cdot) \) 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 to 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 \( \mathbf{F}(\cdot) \) is reserved in the Athena environment to represent the values of these functions. For example \( \mathbf{F}(1) \) represents the rate of change of the cell concentration, \( \mathbf{F}(2) \) 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.

```plaintext
@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 \( \text{Temp} \) 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.

```v
@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 $\text{Par}(\ )$ is reserved to access the adjustable parameters, and the vector $\text{Xu}(\ )$ 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, $\text{Xu}(\ )$ is reserved for the experimental settings. For example $\text{Xu}(1)$ represents the first setting, $\text{Xu}(2)$ 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.

```v
@Connect Variables
RxnTime=\text{Xu}(3) \quad \text{! Reaction Time, s}
Temp=\text{Xu}(2)+273.15 \quad \text{! Reaction Temperature, deg K}
kB(1:5)=\text{Par}(1:5)
EB(1:5)=\text{Par}(6:10)
```
8.5.2 The Parameter Estimation Solver

It is now time to access the Athena Visual Studio 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

From the Nonlinear Regression Parameters group enter:

- The number of unknown parameters (10)
- The number of experiments or datasets (Actual number is 23. Insert 9 to estimate only the base reaction rate constants $k_B(1:5)$ 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:

![Parameter Estimation Solver Control Panel](image)

<table>
<thead>
<tr>
<th>Name</th>
<th>Initial Value</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
<th>Perturbation</th>
<th>Trust Region</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>2</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>3</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>4</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>7</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>8</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>9</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
<tr>
<td>10</td>
<td>0.0</td>
<td>0.0</td>
<td>1.0</td>
<td>0.001</td>
<td>0.7%</td>
</tr>
</tbody>
</table>

**Important Note:** Observe that we first wish to estimate only the base values of the reaction rate constants $k_B(1:5)$ 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)

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 \( \text{Xu}(1:7) \); specifically \( \text{Xu}(1) \) contains the Block ID (not used in the estimation), \( \text{Xu}(2) \) contains the data for the reaction temperature, \( \text{Xu}(3) \) the data for the reaction time and \( \text{Xu}(4:7) \) 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
```

© Athena Visual Software, Inc.
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:

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=$X_u(3)$, 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:

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

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

### NORMALIZED PARAMETER COVARIANCE MATRIX

<table>
<thead>
<tr>
<th></th>
<th>1.000</th>
<th>-0.662 1.000</th>
<th>-0.403 0.403 1.000</th>
<th>-0.012 -0.379 -0.008 0.009 1.000</th>
<th>1.000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.000</td>
<td></td>
<td>-0.662 1.000</td>
<td>-0.403 0.403 1.000</td>
<td>-0.012 -0.379 -0.008 0.009 1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>-0.662</td>
<td></td>
<td>-0.662 1.000</td>
<td>-0.403 0.403 1.000</td>
<td>-0.012 -0.379 -0.008 0.009 1.000</td>
<td>1.000</td>
</tr>
<tr>
<td>1.000</td>
<td></td>
<td>-0.403 0.403 1.000</td>
<td>-0.012 -0.379 -0.008 0.009 1.000</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>-0.403</td>
<td></td>
<td>-0.403 0.403 1.000</td>
<td>-0.012 -0.379 -0.008 0.009 1.000</td>
<td>1.000</td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td></td>
<td>-0.012 -0.379 -0.008 0.009 1.000</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.012</td>
<td></td>
<td>-0.012 -0.379 -0.008 0.009 1.000</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.379</td>
<td></td>
<td>-0.379 -0.008 0.009 1.000</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-0.008</td>
<td></td>
<td>-0.008 0.009 1.000</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.009</td>
<td></td>
<td>0.009 1.000</td>
<td>1.000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td></td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### EVENT  OBSERVED  PREDICTED  RESIDUAL

<table>
<thead>
<tr>
<th>EVENT</th>
<th>OBSERVED</th>
<th>PREDICTED</th>
<th>RESIDUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.600E-01</td>
<td>5.600E-01</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>2</td>
<td>8.600E-01</td>
<td>1.047E+00</td>
<td>-1.871E-01</td>
</tr>
<tr>
<td>3</td>
<td>1.600E+00</td>
<td>1.712E+00</td>
<td>-1.123E-01</td>
</tr>
<tr>
<td>4</td>
<td>2.600E+00</td>
<td>2.390E+00</td>
<td>2.091E-01</td>
</tr>
<tr>
<td>5</td>
<td>3.200E+00</td>
<td>2.904E+00</td>
<td>2.951E-01</td>
</tr>
<tr>
<td>6</td>
<td>3.300E+00</td>
<td>3.213E+00</td>
<td>8.650E-02</td>
</tr>
<tr>
<td>7</td>
<td>3.500E+00</td>
<td>3.373E+00</td>
<td>1.264E-01</td>
</tr>
<tr>
<td>8</td>
<td>3.400E+00</td>
<td>3.450E+00</td>
<td>-5.035E-02</td>
</tr>
<tr>
<td>9</td>
<td>3.400E+00</td>
<td>3.485E+00</td>
<td>-8.578E-02</td>
</tr>
<tr>
<td>10</td>
<td>6.600E-01</td>
<td>6.600E-01</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>11</td>
<td>1.000E+00</td>
<td>1.285E+00</td>
<td>-2.856E-01</td>
</tr>
<tr>
<td>12</td>
<td>1.600E+00</td>
<td>2.078E+00</td>
<td>-4.782E-01</td>
</tr>
<tr>
<td>13</td>
<td>2.600E+00</td>
<td>2.763E+00</td>
<td>-1.636E-01</td>
</tr>
<tr>
<td>14</td>
<td>3.200E+00</td>
<td>3.186E+00</td>
<td>1.350E-02</td>
</tr>
<tr>
<td>15</td>
<td>3.300E+00</td>
<td>3.395E+00</td>
<td>-9.550E-02</td>
</tr>
<tr>
<td>16</td>
<td>3.300E+00</td>
<td>3.487E+00</td>
<td>-1.875E-01</td>
</tr>
<tr>
<td>17</td>
<td>3.300E+00</td>
<td>3.526E+00</td>
<td>-2.259E-01</td>
</tr>
<tr>
<td>18</td>
<td>8.000E-01</td>
<td>8.000E-01</td>
<td>0.0000E+00</td>
</tr>
<tr>
<td>19</td>
<td>1.500E+00</td>
<td>1.633E+00</td>
<td>-1.337E-01</td>
</tr>
<tr>
<td>20</td>
<td>2.600E+00</td>
<td>2.546E+00</td>
<td>5.331E-02</td>
</tr>
<tr>
<td>21</td>
<td>3.500E+00</td>
<td>3.152E+00</td>
<td>3.474E-01</td>
</tr>
<tr>
<td>22</td>
<td>3.500E+00</td>
<td>3.433E+00</td>
<td>6.655E-02</td>
</tr>
<tr>
<td>23</td>
<td>3.500E+00</td>
<td>3.542E+00</td>
<td>-4.209E-02</td>
</tr>
</tbody>
</table>

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 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 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 \textit{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 \textit{trace criterion} for reducing the HPD region was introduced by Pinto \textit{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 $\Theta_a$ 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.

\section*{9.2 References}


9.3 Model Discrimination with an Explicit Model

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 Athena Visual Studio. 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 Athena Visual Studio 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 Athena Visual Studio in dealing with parameter estimation and model discrimination and criticism from single-response data with explicit mechanistic models.
9.4 Model Discrimination in CH₃OH Chemistry

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

\[
CO + 2H₂ \xrightarrow{\mathcal{R}} CH₃OH
\]

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:

\[
\mathcal{R}_1 = \frac{k(T)K_1K_2^2\left(P_C^2P_H^2 - P_M / K_{eq}\right)}{(1 + K_1P_C + K_2P_H + K_3P_M)^3} \quad \text{and} \quad \mathcal{R}_2 = k(T)P_C^n P_H^n P_M^n
\]

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

\[
\log K_{eq} = \frac{3914}{T} - 7.536 \log T + 0.001766 \log T + 9.388
\]

- Estimate the parameters \(\{k_B, E, K_1, K_2, K_3, n_1, n_2, n_3\}\)
- 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,

\[
10 \leq P_C \leq 50 \\
50 \leq P_H \leq 100 \\
0 \leq P_M \leq 15 \\
470 \leq T \leq 500
\]

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

<table>
<thead>
<tr>
<th>Initial Values of Parameters</th>
<th>(K_1 = 0.1)</th>
<th>(n_1 = 0)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\ln k_B = -5.0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(\frac{E}{RT_B} = 10.0)</td>
<td>(K_2 = 0.1)</td>
<td>(n_2 = 0)</td>
</tr>
<tr>
<td>(T_B = 485.0)</td>
<td>(K_3 = 0.1)</td>
<td>(n_3 = 0)</td>
</tr>
</tbody>
</table>
The experimental data for this example are given in the Table below:

<table>
<thead>
<tr>
<th>Temp</th>
<th>PC</th>
<th>PH</th>
<th>PM</th>
<th>Rate</th>
<th>Weight</th>
<th>Replicate</th>
</tr>
</thead>
<tbody>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>84.00</td>
<td>10.00</td>
<td>4.7890E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>58.30</td>
<td>10.00</td>
<td>3.4180E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>495.00</td>
<td>15.10</td>
<td>84.00</td>
<td>10.00</td>
<td>4.4470E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>495.00</td>
<td>15.10</td>
<td>58.30</td>
<td>10.00</td>
<td>3.4950E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>84.00</td>
<td>2.50</td>
<td>7.1740E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>495.00</td>
<td>40.00</td>
<td>58.30</td>
<td>2.50</td>
<td>5.3790E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>495.00</td>
<td>15.10</td>
<td>84.00</td>
<td>2.50</td>
<td>6.6620E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>495.00</td>
<td>15.10</td>
<td>58.30</td>
<td>2.50</td>
<td>5.0520E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>40.00</td>
<td>84.00</td>
<td>10.00</td>
<td>1.2540E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>40.00</td>
<td>58.30</td>
<td>10.00</td>
<td>8.3500E-07</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>84.00</td>
<td>10.00</td>
<td>1.1500E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>58.30</td>
<td>10.00</td>
<td>9.1700E-07</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>40.00</td>
<td>84.00</td>
<td>2.50</td>
<td>2.0050E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>40.00</td>
<td>58.30</td>
<td>2.50</td>
<td>1.5070E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>84.00</td>
<td>2.50</td>
<td>2.0450E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>475.00</td>
<td>15.10</td>
<td>58.30</td>
<td>2.50</td>
<td>1.4430E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.8580E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.7840E-06</td>
<td>1.0</td>
<td>1</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.7930E-06</td>
<td>1.0</td>
<td>1</td>
</tr>
<tr>
<td>500.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>7.4900E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>470.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>9.9000E-07</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>15.00</td>
<td>1.7390E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>1.70</td>
<td>3.7250E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>70.00</td>
<td>5.00</td>
<td>2.6460E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>12.60</td>
<td>70.00</td>
<td>5.00</td>
<td>2.5520E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>92.10</td>
<td>5.00</td>
<td>3.6060E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
<tr>
<td>485.00</td>
<td>25.00</td>
<td>53.00</td>
<td>5.00</td>
<td>2.2710E-06</td>
<td>1.0</td>
<td>0</td>
</tr>
</tbody>
</table>
9.5 Implementation in Athena Visual Studio

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


- 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 \texttt{@Response Model} is used to define the experimental responses (observations). The vector $Y()$ is reserved in Athena to define these responses. A section labeled \texttt{@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 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:. 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 Athena Visual Studio

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 \( \text{Skount}, \text{Ncc} \) will be treated as integer and their numerical values will be accessible throughout 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 they 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:

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

We are now going to describe in detail the various steps involved in writing an implicit model for parameter estimation in the Athena Visual Studio 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 Athena Visual Studio project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```plaintext
! 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 \texttt{iModel} and \texttt{CHMAX}. The \texttt{iModel} variable is used as a counter for the number of models that we wish to process. In our example \texttt{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 \texttt{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 \texttt{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 \( \text{Par}(\) \) is reserved to access the adjustable parameters, and the vector \( \text{Xu}(\) \) is reserved to access the experimental settings. As mentioned above the vector \( \text{Y}(\) \) is reserved to enter the responses. For example \( \text{Y}(1) \) represents the first measured response, \( \text{Y}(2) \) the second (if present) and so on. As mentioned above, \( \text{Xu}(\) \) is reserved for the experimental settings. For example \( \text{Xu}(1) \) represents the first setting, \( \text{Xu}(2) \) the second if present and so on. To enter the responses for your model:

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

```plaintext
@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 $dY(\cdot)$ 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 $\text{Par}(3)$. 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:

![Parameter Estimation Solver Control Panel](image)

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

```plaintext
@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 Athena Visual Studio 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

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:
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)

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:

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

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>ESTIMATES</th>
<th>95% MARGINAL HPD INTERVALS</th>
<th>PARAMETER STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAR( 1)</td>
<td>-9.709487E+00</td>
<td>-9.709487E+00 +- 2.249E-01</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 2)</td>
<td>3.149089E+01</td>
<td>3.149089E+01 +- 1.597E+00</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 3)</td>
<td>6.563378E-02</td>
<td>6.563378E-02 +- 1.752E-02</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 4)</td>
<td>2.904900E-02</td>
<td>2.904900E-02 +- 1.018E-02</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 5)</td>
<td>9.839118E-02</td>
<td>9.839118E-02 +- 2.958E-02</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 6)</td>
<td>0.000000E+00</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>PAR( 7)</td>
<td>0.000000E+00</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>PAR( 8)</td>
<td>0.000000E+00</td>
<td></td>
<td>Fixed</td>
</tr>
</tbody>
</table>

EXIT GREGPLUS: MODEL DISCRIMINATION AND CRITICISM

<table>
<thead>
<tr>
<th>SOURCE OF VARIANCE</th>
<th>SUM OF SQUARES</th>
<th>DEG. OF FREEDOM</th>
<th>MEAN SQUARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residuals</td>
<td>7.67167E-13</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>7.63906E-13</td>
<td>20</td>
<td>3.82E-14</td>
</tr>
<tr>
<td>Experimental</td>
<td>3.26076E-15</td>
<td>2</td>
<td>1.63E-15</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>ESTIMATES</th>
<th>95% MARGINAL HPD INTERVALS</th>
<th>PARAMETER STATUS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PAR( 1)</td>
<td>-1.582727E+01</td>
<td>-1.582727E+01 +- 6.345E-01</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 2)</td>
<td>3.159274E+01</td>
<td>3.159274E+01 +- 1.779E+00</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 3)</td>
<td>1.000000E-01</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>PAR( 4)</td>
<td>1.000000E-01</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>PAR( 5)</td>
<td>1.000000E-01</td>
<td></td>
<td>Fixed</td>
</tr>
<tr>
<td>PAR( 6)</td>
<td>5.421410E-02</td>
<td>5.421410E-02 +- 5.241E-02</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 7)</td>
<td>7.731606E-01</td>
<td>7.731606E-01 +- 1.417E-01</td>
<td>Estimated</td>
</tr>
<tr>
<td>PAR( 8)</td>
<td>-2.901408E-01</td>
<td>-2.901408E-01 +- 3.805E-02</td>
<td>Estimated</td>
</tr>
</tbody>
</table>

EXIT GREGPLUS: MODEL DISCRIMINATION AND CRITICISM

<table>
<thead>
<tr>
<th>SOURCE OF VARIANCE</th>
<th>SUM OF SQUARES</th>
<th>DEG. OF FREEDOM</th>
<th>MEAN SQUARE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Residuals</td>
<td>9.06997E-13</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>Lack of Fit</td>
<td>9.03736E-13</td>
<td>20</td>
<td>4.52E-14</td>
</tr>
<tr>
<td>Experimental</td>
<td>3.26067E-15</td>
<td>2</td>
<td>1.63E-15</td>
</tr>
</tbody>
</table>

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 form 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:

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:

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](image)

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.
10.0 Nonlinear Optimization Problems

The basic elements of an optimization problem are listed below:

1. The objective [or merit] function
2. The equality constraints
3. The inequality constraints
4. The lower and upper bounds on the independent variables
5. A set of independent or decision variables
6. A set of known parameters pertinent to the process under investigation

The conventional mathematical formulations for various types of optimization problems including both constrained and unconstrained are given in the tables below. In these tables we differentiate between explicit and implicit models. Implicit models frequently contains mixed systems of differential and algebraic equations, that need to be solved efficiently during the optimization process. Differential equations can be both ordinary and partial, thus describing a wide range of lumped as well distributed parameter systems, such as continuous stirred tank reactors, fixed bed and plug flow reactors both dynamic and steady-state.
where

- **x**
  Multi-dimensional vector of unknown variables also known as independent variables or decision variables

- **u**
  Multi-dimensional vector of state variables

- **θ**
  Multi-dimensional vector of known parameters

- **f(x;θ)**
  The [scalar] objective function
  - Profit or cost financial objective
  - Process objective

- **h(x;θ) = 0**
  Multi-dimensional vector of equality constraints
  - Material and Energy balance equations
  - Design specifications

- **c(x;θ) ≥ 0**
  Multi-dimensional vector of inequality constraints
  - Design and Equipment constraints
  - Process constraints

- **x^l, x^u**
  Lower and upper bounds on the independent variables

A point **x** that satisfies the equality and inequality constraints is called a **feasible point**. The set of all points that are feasible is called the **feasible region**.
Objective Function: The objective function usually is a profit or cost function. Frequently the objective function represents a process variable, such as, production rate.

Equality Constraints: The equality constraints typically consist of the material and energy balance, and miscellaneous design and operating specifications.

Inequality Constraints: The inequality constraints represent design and operation limitations as well as operating regions.

Lower and Upper Bounds: Lower and upper bounds on the independent variables represent feasible operating regimes.

10.1 Alkylation Process Optimization

The following nonlinear model including the objective function and process constraints has been postulated for the optimization of the alkylation process:

maximize the process profit
\[
\Phi(x) = c_1x_4x_7 - c_2x_4 - c_3x_2 - c_4x_3 - c_5x_5
\]
subject to process constraints
\[
x_4 = x_1(1.12 + 0.12167x_8 - 0.0067x_8^2)
\]
\[
x_5 = 1.22x_4 - x_i
\]
\[
x_6(x_4x_9 + 1000x_5) = 98000x_5
\]
\[
x_7 = 86.35 + 1.098x_4 - 0.038x_2^2 + 0.325(x_6 - 89)
\]
\[
x_8x_1 = x_2 + x_5
\]
\[
x_9 = 35.82 - 0.222x_{10}
\]
\[
x_{10} = 3x_7 - 133
\]

The profit function is calculated based on cost data given in the table below along with the definitions of the Alkylation process variables.

<table>
<thead>
<tr>
<th>Process Variable</th>
<th>( C_i )</th>
<th>( x_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alkylate Product Value</td>
<td>0.063</td>
<td>( x_1 )</td>
</tr>
<tr>
<td>Olefin Feed Cost</td>
<td>5.04</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>Isobutane Feed Cost</td>
<td>3.36</td>
<td>( x_3 )</td>
</tr>
<tr>
<td>Isobutane Recycle Cost</td>
<td>0.035</td>
<td>( x_4 )</td>
</tr>
<tr>
<td>Acid Addition Cost</td>
<td>10.0</td>
<td>( x_10 )</td>
</tr>
<tr>
<td>Isobutane Makeup</td>
<td></td>
<td>( x_9 )</td>
</tr>
<tr>
<td>Acid Strength</td>
<td></td>
<td>( x_5 )</td>
</tr>
<tr>
<td>Isobutane to Olefin Ratio</td>
<td></td>
<td>( x_6 )</td>
</tr>
<tr>
<td>Motor Octane Number</td>
<td></td>
<td>( x_7 )</td>
</tr>
<tr>
<td>Acid Addition Rate</td>
<td></td>
<td>( x_8 )</td>
</tr>
<tr>
<td>External Isobutane to Olefin Ratio</td>
<td></td>
<td>( x_9 )</td>
</tr>
<tr>
<td>Acid Dilution Factor</td>
<td></td>
<td>( x_8 )</td>
</tr>
<tr>
<td>F-4 Performance Number of Alkylate</td>
<td></td>
<td>( x_{10} )</td>
</tr>
</tbody>
</table>

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 Nonlinear Constrained Optimization sample
Click OK
10.2 Implementation in Athena Visual Studio

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

- Select the Nonlinear Optimization tab
- Select the Optimization 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.
10.2.1 Writing Source Code for Optimization Problems

You must enter a minimum of one section in order to create the nonlinear optimization model with user defined explicit models. This section is labeled `@Optimization Model` and is used to insert the objective function and the constraints. 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:
10.2.1.1 Declaration of Variables in Athena Visual Studio

**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 throughout 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 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 explicit model for nonlinear optimization in the Athena Visual Studio environment. The modeling code is NOT case sensitive.
10.2.1.2 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 cost data required for the calculation of the objective function. 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 Athena Visual Studio project, please beware that invisible format symbols may also be copied and cause the compilation of your code to fail):

```plaintext
! Declarations and Model Constants
!---------------------------------                   
Global C1,C2,C3,C4,C5 As Real
C1=0.063  ! alkylate product value/octane-barrel
C2=5.04   ! olefin feed cost/barrel
C3=0.035  ! isobutane recycle costs/barrel
C4=10.0   ! acid addition cost/1000 lbs
C5=3.36   ! isobutane make-up cost/barrel
```

10.2.1.3 Optimization Model

In the Optimization Model section the user must enter the objective function and the constraints(if any). The vector \( \mathbf{F}( ) \) is reserved to enter these functions. For example \( \mathbf{F}(1) \) represents the objective function, \( \mathbf{F}(2) \) represents the equality constraint that describes the alkylate yield and so on. To enter the objective and constraints for your model:

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

```plaintext
@Optimization Model
F(1)=C1*X(4)*X(7)-C2*X(1)-C3*X(2)-C4*X(3)-C5*X(5)
F(2)=X(4)-X(1)*(1.12+0.12167*X(8)-0.00667*X(8)^2)
F(3)=X(5)+X(1)-1.22*X(4)
F(4)=X(6)*(X(4)*X(9)+1000.0*X(3))-98000.0*X(3)
F(5)=X(7)-1.098*X(8)+0.0038*X(8)^2-0.325*(X(6)-89.0)
F(6)=X(8)*X(1)-X(2)-X(5)
F(7)=X(9)+0.222*X(10)
F(8)=3.0*X(7)-X(10)
```

© Athena Visual Software, Inc.
10.2.2 Entering the Information for the Optimization Solver

It is now time to access the Athena Visual Studio solver for Nonlinear Optimization in order to enter information about the independent variables, the objective function and constraints and various other parameters that control the optimization algorithm, To do that:

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

From the Optimization Parameters group enter:

- The number of Independent Variables (10)
- The number of Functions (Objective and Constraints) (8)

Optionally you may change the Number of Iterations, the Optimality Tolerance, the Constraints Functions or Independent Variables Tolerance, the Print Level Control Flag and the Real and Integer Working Arrays space requirements. From the Optimization Problem Type group choose Non-Linear with General Constraints, and from the Optimization Solver Tasks group choose Maximize Objective Function. You may also consider to Check here to Test the Initial Values. The test call to model is a very useful option since it allows you to see how good is your initial guess before you proceed with the optimization process. Finally you may consider entering a more appropriate value for the Perturbation Step Size. Keep in mind that if your functions are calculated down to machine precision, then a good value for the perturbation step size is the square root of the machine precision (this value is around 1.0E-8)
10.2.3 Entering the Information for the Independent Variables

Next select the **Independent Variables** tab on the Optimization Solver Control Panel and enter the names of the variables you wish to optimize, their initial guesses and optionally their lower and upper bounds, the scale factor to improve the optimization and the size of the trust region. Notice that if a variable is not checked in the *State* column next to it, will remain fixed at its initial value during the optimization. The following data have been inserted for our example:

![Optimization Solver Control Panel](image-url)

<table>
<thead>
<tr>
<th>Names</th>
<th>State</th>
<th>Value</th>
<th>Lower Bound</th>
<th>Upper Bound</th>
<th>Scale Factor</th>
<th>Trust Region</th>
<th>Ana</th>
</tr>
</thead>
<tbody>
<tr>
<td>Olefin Feed</td>
<td></td>
<td>1745.00</td>
<td>0.0</td>
<td>2000.0</td>
<td>1000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Isobutene Recycle</td>
<td></td>
<td>12000.0</td>
<td>0.0</td>
<td>16000.0</td>
<td>1000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acid Addition Rate</td>
<td></td>
<td>110.00</td>
<td>0.0</td>
<td>120.0</td>
<td>100.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Alkylate Yield</td>
<td></td>
<td>3048.00</td>
<td>0.0</td>
<td>6000.0</td>
<td>10000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Isobutene Makeup</td>
<td></td>
<td>1974.00</td>
<td>0.0</td>
<td>2000.0</td>
<td>1000.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acid Strength</td>
<td></td>
<td>89.20</td>
<td>88.0</td>
<td>93.0</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Motor Octane Number</td>
<td></td>
<td>92.80</td>
<td>90.0</td>
<td>95.0</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Isobutene to Olefin</td>
<td></td>
<td>8.00</td>
<td>8.0</td>
<td>12.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acid Dilution</td>
<td></td>
<td>5.60</td>
<td>1.2</td>
<td>4.0</td>
<td>1.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Performance Factor</td>
<td></td>
<td>146.00</td>
<td>145.0</td>
<td>162.0</td>
<td>10.0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
10.2.4 Entering the Information for the Dependent Variables

Next select the **Objective and Constraints** tab on the Optimization Solver Control Panel and enter the names and the types of the model functions. To enter the type perform a right mouse click in the **Type** column next the function you wish to specify and then click **Select Constraint**. The **Select Function Type** window will show up (see image below). Select the **Function Type** and then click **OK**. Next enter the lower and upper bounds for the constraints making sure that these bounds are identical for the equality constraints. For **Less Than** type constraints enter only the upper bound (the lower bound is taken as negative infinity). For **Greater Than** type constraints enter only the lower bound (the upper bound is taken as infinity). Optionally you may enter scale factors for the constraints and objective as well as overwrite the global tolerances for convergence by specifying an individual tolerance for each function. The following data have been inserted for our example:
10.2.5 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 Nonlinear Optimization 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)
10.2.6 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 run statistics pertaining to the solution process:

Number of Independent Variables....................... 10
Number of Dependent Variables......................... 8
Number of User Specified Iterations.................... 30

EXIT OPTIPLUS: SOLUTION FOUND

<table>
<thead>
<tr>
<th>VARIABLE</th>
<th>STATE</th>
<th>LOWER BOUND</th>
<th>UPPER BOUND</th>
<th>INITIAL VALUE</th>
<th>FINAL VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Olefin F</td>
<td>1 FR</td>
<td>0.000000E+00</td>
<td>2.000000E+03</td>
<td>1.745000E+03</td>
<td>1.917843E+03</td>
</tr>
<tr>
<td>Isobutan</td>
<td>1 FR</td>
<td>0.000000E+00</td>
<td>1.600000E+04</td>
<td>1.200000E+04</td>
<td>1.596152E+04</td>
</tr>
<tr>
<td>Acid Add</td>
<td>1 FR</td>
<td>0.000000E+00</td>
<td>1.200000E+02</td>
<td>1.100000E+02</td>
<td>4.359033E+01</td>
</tr>
<tr>
<td>Alkylate</td>
<td>1 FR</td>
<td>0.000000E+00</td>
<td>5.000000E+03</td>
<td>3.048000E+03</td>
<td>3.211347E+03</td>
</tr>
<tr>
<td>Isobutan</td>
<td>1 UL</td>
<td>0.000000E+00</td>
<td>2.000000E+03</td>
<td>1.974000E+03</td>
<td>2.000000E+03</td>
</tr>
<tr>
<td>Acid Str</td>
<td>1 LL</td>
<td>8.500000E+01</td>
<td>9.300000E+01</td>
<td>8.920000E+01</td>
<td>8.500000E+01</td>
</tr>
<tr>
<td>Motor Oc</td>
<td>1 UL</td>
<td>9.000000E+01</td>
<td>9.500000E+01</td>
<td>9.280000E+01</td>
<td>9.500000E+01</td>
</tr>
<tr>
<td>Isobutan</td>
<td>1 FR</td>
<td>3.000000E+00</td>
<td>1.200000E+01</td>
<td>8.000000E+00</td>
<td>9.365489E+00</td>
</tr>
<tr>
<td>Acid Dil</td>
<td>1 FR</td>
<td>1.200000E+00</td>
<td>4.000000E+00</td>
<td>3.600000E+00</td>
<td>2.076000E+00</td>
</tr>
<tr>
<td>Performa</td>
<td>1 FR</td>
<td>1.450000E+02</td>
<td>1.620000E+02</td>
<td>1.450000E+02</td>
<td>1.520000E+02</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FUNCTION</th>
<th>STATE</th>
<th>LOWER BOUND</th>
<th>UPPER BOUND</th>
<th>INITIAL VALUE</th>
<th>FINAL VALUE</th>
<th>LAGRANGE MULT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Objectiv</td>
<td>O</td>
<td>NONE</td>
<td>NONE</td>
<td>8.723872E+02</td>
<td>1.839425E+03</td>
<td></td>
</tr>
<tr>
<td>Alkylate</td>
<td>E OK</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>1.399924E+02</td>
<td>5.022278E-05</td>
<td>-6.708493E-01</td>
</tr>
<tr>
<td>Isobutan</td>
<td>E OK</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>4.400000E-01</td>
<td>-3.601563E-13</td>
<td>4.244614E+00</td>
</tr>
<tr>
<td>Acid Str</td>
<td>E OK</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>1.077376E+04</td>
<td>-3.862624E-09</td>
<td>-7.692394E-04</td>
</tr>
<tr>
<td>Motor Oc</td>
<td>E OK</td>
<td>8.635000E+01</td>
<td>8.635000E+01</td>
<td>8.419420E+01</td>
<td>8.365000E+01</td>
<td>-6.955878E+01</td>
</tr>
<tr>
<td>Isobutan</td>
<td>E --</td>
<td>0.000000E+00</td>
<td>0.000000E+00</td>
<td>-1.400000E+01</td>
<td>1.891684E-02</td>
<td>-3.500650E-02</td>
</tr>
<tr>
<td>Acid Dil</td>
<td>E OK</td>
<td>3.582000E+01</td>
<td>3.582000E+01</td>
<td>3.579000E+01</td>
<td>3.582000E+01</td>
<td>2.099756E+02</td>
</tr>
<tr>
<td>F4 Perfo</td>
<td>E OK</td>
<td>1.330000E+02</td>
<td>1.330000E+02</td>
<td>1.330000E+02</td>
<td>1.330000E+02</td>
<td>4.661460E+01</td>
</tr>
</tbody>
</table>

Number of Function Evaluations.................... 21
Number of Optimization Iterations.................. 21
Number of Line Search Evaluations.................. 0

From the above results we see that the optimization process took 21 iterations. No line search was performed during this process; this means that the full Newton step was taken at every single iteration. It is instructive to see that the optimizer reports the Isobutane constraint as violated(--). This is due to scaling of the problem. Choosing the proper scaling factor can expedite the solution process and also lead to better results. The choice of scaling factors however, is an art, not a science, and sometimes it may lead to failure instead of success.
Index

Adjustable Parameters, 109, 112, 137, 142, 156, 158
Bayes’ theorem, 98, 100
Candidate Models, 160, 164

CHMAX, 154

Connect Parameters, 134, 135, 137, 138, 139

DDAPLUS, 133
Determinant criteria, 145
Excel, 113, 138, 159
Goodness-of-Fit, 148

Gradient Vector, 109, 112, 137, 156, 158
Gradient Vector Section, 156
Graphical Results, 116, 144, 166

iModel, 154
Implicit Model, 126, 147
Implicit or Explicit Discontinuities, 45

Initial Conditions, 127, 133

Integration Parameters, 139
Lack-of-Fit, 148
Lack-of-Fit Analysis, 160, 164

Lag Plot, 117

likelihood, 100, 101
Making Predictions, 119
maximum likelihood, 101
Model Discrimination, 145

Model Equations, 108, 134, 155
Multi-Response Data, 121

Nonlinear Regression Parameters, 111, 136, 157
Normal Probability Plot, 117

Optimal Experimental Design, 145, 164, 165, 166

Ordinary Differential Equations, 24, 129
Parameter Estimation, 1, 98, 101, 104, 109, 111, 112, 113, 117, 122, 125, 129, 136, 137, 138, 139, 146, 150, 156, 157, 158, 159
Parameter Section, 133, 154
parameter transformations, 103, 117, 127
parity graph, 116
Partial Differential Equations, 82
perturbation step size, 110, 112, 137, 157, 158

Phi functions, 103, 127
Phi Functions, 117, 118, 119, 160, 164
Replicate Experiments, 160
Response Model, 105, 108, 109, 135, 151, 155, 156
Response Model Section, 108, 109, 155
Run Order (or Sequence) Plot, 117

Scatter Plot, 116
shape criteria, 146
staged parameter estimation, 137

System Identification, 45, 139

System Options, 139

Update Solution, 142
VisualKinetics, 83