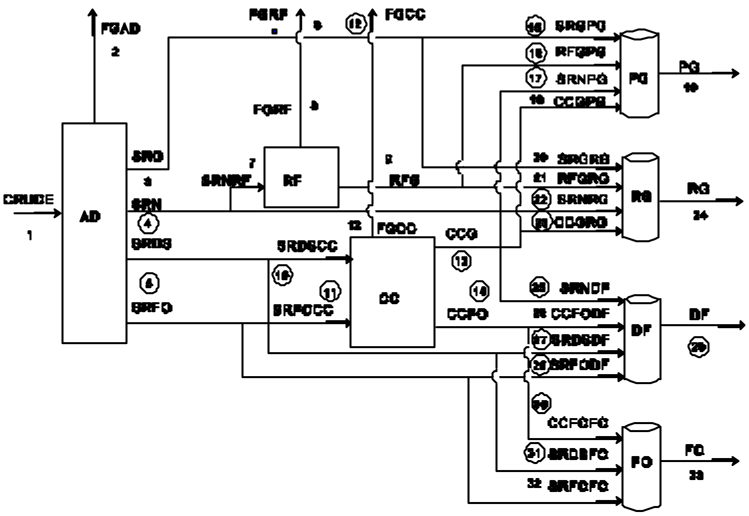
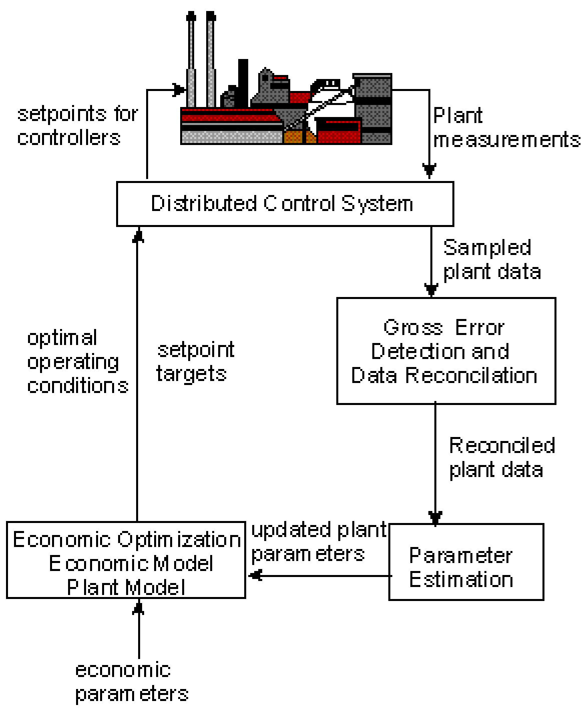


Mineral Processing Research Institute
Louisiana State University

Interactive On-Line Optimization System



User's Manual
and
Tutorial
For Simple Refinery

Xueyu Chen
Ralph W. Pike
Janardhana R. Punuru

March 1,2001

TABLE OF CONTENTS

Disclaimer

Mineral Processing Research Institute(MPRI) makes no warranties, express or implied, including without limitation the implied warranties of merchantability and fitness for particular purpose, regarding the MPRI software. MPRI does not warrant, guarantee or make any representation regarding the use or the results of the use of the MPRI software in terms of its correctness, accuracy, reliability, currentness or otherwise. The entire risk as to the results and performance of the MPRI software is assumed by you.

In no event will MPRI, its director, officers, employees or agents be liable to you for any consequential, incidental or indirect damages (including damage for loss of business profits, business interruption, loss of business information, and the like) arising out of the use or inability to use the MPRI software even if MPRI has been advised of the possibility of such damages.

I. INTRODUCTION

On-line optimization is an effective approach for economic improvement and source reduction in chemical plants. On-line optimization uses an automated system which adjusts the operation of a plant based on product scheduling and production control to maximize profit and minimize emissions by providing optimal set points to the distributed control system.

On-line optimization includes three nonlinear optimization problems, which are gross error detection and data reconciliation, parameter estimation, and economic optimization as shown in Figure 1. The procedure to conduct on-line optimization for a plant requires that these three optimization problems be solved in sequence. First, the plant data is extracted from the distributed control system. Then gross errors are rectified and the data is reconciled to satisfy process material and energy balances. This reconciled set of data is then used to estimate the current values of the process parameters. This updates the process model to have plant-model matching, and then the updated model is used for economic optimization to generate the optimal set points that will maximize the plant's profit and will satisfy the constraints in the process model. The system guides the engineer to enter information required for on-line optimization. Also, the engineer can select to solve any one of the three optimization problems separately, gross error detection and data reconciliation, parameter estimation, or economic optimization. An example for the simple refinery (Pike, 1986) is used to demonstrate entering the process and economic models and obtaining optimal solution using the interactive on-line optimization system. Models for the aniline process and sulfuric acid contact plant are included, also.

The interactive on-line optimization system provides a mechanism where all information needed to solve the three nonlinear programming problems is provided by the process engineer through a window's interface. The three optimization problems involved in the on-line optimization system share and transfer information. The process simulation and economic models, raw material availability and product demand data be input through the interface program to generate the optimization programs. The system then extracts plant data from the database of the distributed control system, performs gross error detection and data reconciliation, parameter estimation, economic optimization, and sends the optimal set points to the distributed control system. The interactive on-line optimization system guides the process engineer to enter the necessary information to complete the process. The process engineer does not need to understand the details of the methodology of on-line optimization.

GAMS (General Algebraic Modeling System) is used to solve the three nonlinear optimization problems of on-line optimization. It is an optimization-simulation language, which was developed to make the formulation and solution of large scale mathematical programming problems more straightforward and comprehensible to the user. The GAMS program was developed at the World Bank and has been used successfully with large economic models of industrial sectors. GAMS has a number of linear and nonlinear solvers, such as MINOS and CONOPT for choices of users. GAMS/MINOS and GAMS/CONOPT are the built-in solvers for the optimization in the interactive on-line optimization system.

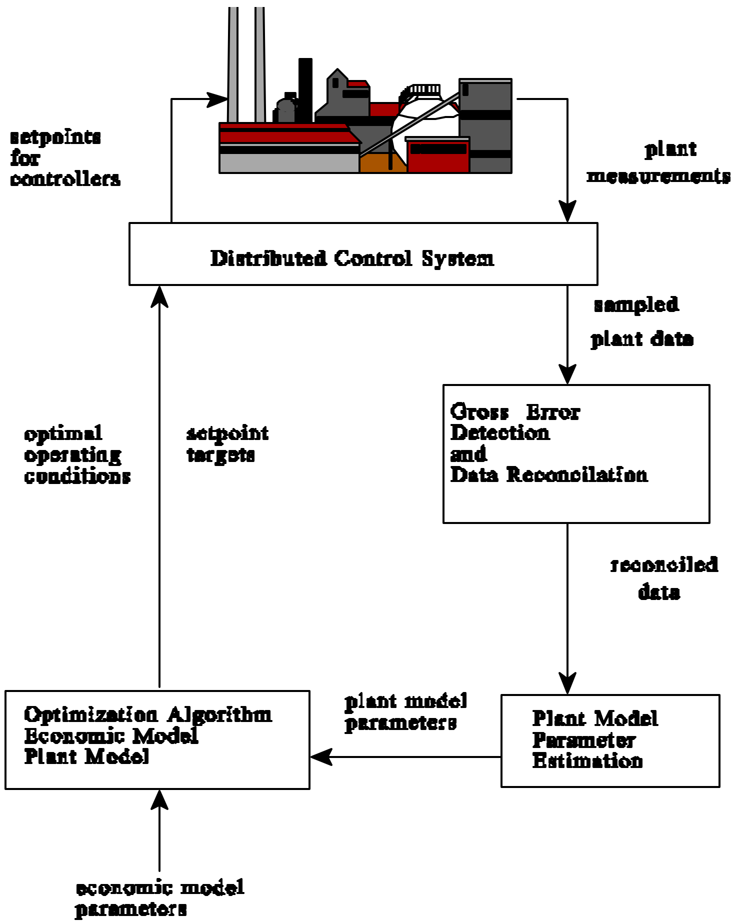


Figure 1 Simplified Structure of On-Line Optimization

Microsoft's Visual Basic 5.0 is a programming language used to develop the interactive on-line optimization system. It provides an efficient way to create User Access Windows as an interface where the process information (data and equations of plant model) can be entered easily. This system only requires that the process engineer provide the plant model, economic model, and plant data from the distributed control system. Then, the interactive On-line Optimization system transfers the input information into a GAMS program, links with GAMS, has GAMS solve the optimization problems, and presents the optimization solution in windows to be reviewed by the process engineer. The process engineer does not need to know the methodology of on-line optimization and the specific GAMS programs for the three optimization problems as the interface program writes these GAMS programs.

II. INSTALLATION

The OnlineOptimization System can be installed using a setup program. The setup program will install the Interactive On - Line Optimization system and the Help system to the user's hard disk.

A. Hardware and System Requirements

To run the Interactive On-line Optimization System, you must have certain hardware and software installed on your computer. The system requirements include:

- Any IBM compatible machine with 80486 processor or higher
- 16 megabytes available space(minimum)
- 3.5-inch disk drive
- Any display supported by Windows 95
- A mouse or other suitable pointing device
- Windows 95 or later, or Windows NT™ 4.0 or later.

B. Installation Procedure

The Interactive On-Line Optimization System must be installed under Windows 95 or Windows NT. The procedure to install Interactive On-Line Optimization System is described as follows:

1) Insert CD-ROM in drive in CD-ROM Drive and run 'onlineopt.exe' program under Windows 95.
2) The default destination directory is 'C:\program files\ioo' into which the program and the help file will be copied when the setup program is run.
3) The setup program installs GAMS in the Agams25@ subdirectory of the installation directory.
4) Run the program 'Online.exe' in the Installation directory.

III. GETTING STARTED

The On-line Optimization instruction screen appears as soon as the application begins as shown in Figure 2. The instruction screen provides the user a choice between an existing model or a new one. If the user does not want this screen to be displayed in future, he/she can check the check box provided for this purpose. The "Cancel" button in this screen, when clicked, hides this screen, and the main window alone is displayed as in Figure 3. The "Help" button when clicked invokes the associated help file (described in detail in the next section).

The On-line Optimization main window shown in Figure 3 is displayed as the user starts the application. The file pull down menu can be used to retrieve an existing model or to create a new model as shown in Figure 4.

CTo display the file menu, point the mouse to the File on the menu bar at the top of the screen and click the left mouse button.
$The File menu displays a list of commands, which the user can invoke. The new command is highlighted. The toolbar can also be used by the user to invoke some of the commands. The ToolTip text displayed when the mouse is moved on top of the buttons describes what that particular button does when clicked. The new command is used to create a new model.

The user may select the highlighted box by either using the up and down arrow keys on your keyboard or by pointing the mouse to the corresponding commands in the file menu. For this introductory section, we will use the Open command to load an existing process model (e.g. a simple refinery).

CTo open an existing process model, point and click the left mouse button on the Open command on the File menu. The Open dialog box appears as shown in Figure 5.
CAn existing file named refinery.ioo that is stored in the Examples directory can be opened by:

1. Changing the current directory to the Examples subdirectory.
2. Selecting the file named **refinery.ioo**
3. Clicking on Open or double clicking on the filename

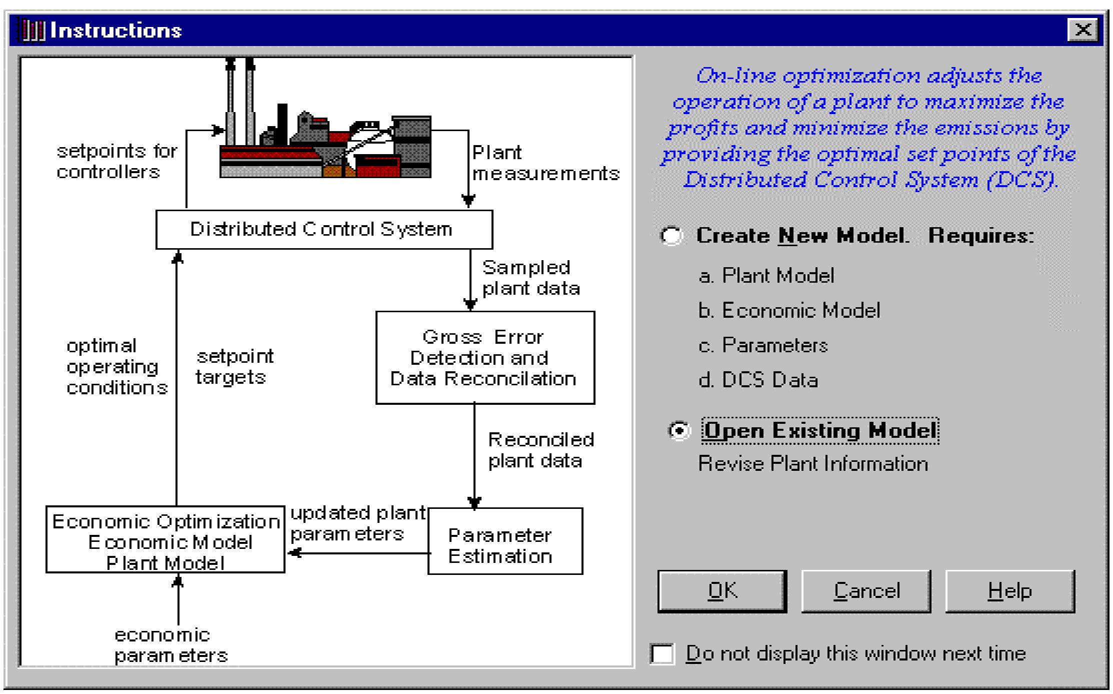


Figure2 The Introduction Window for The Interactive On-Line Optimization System

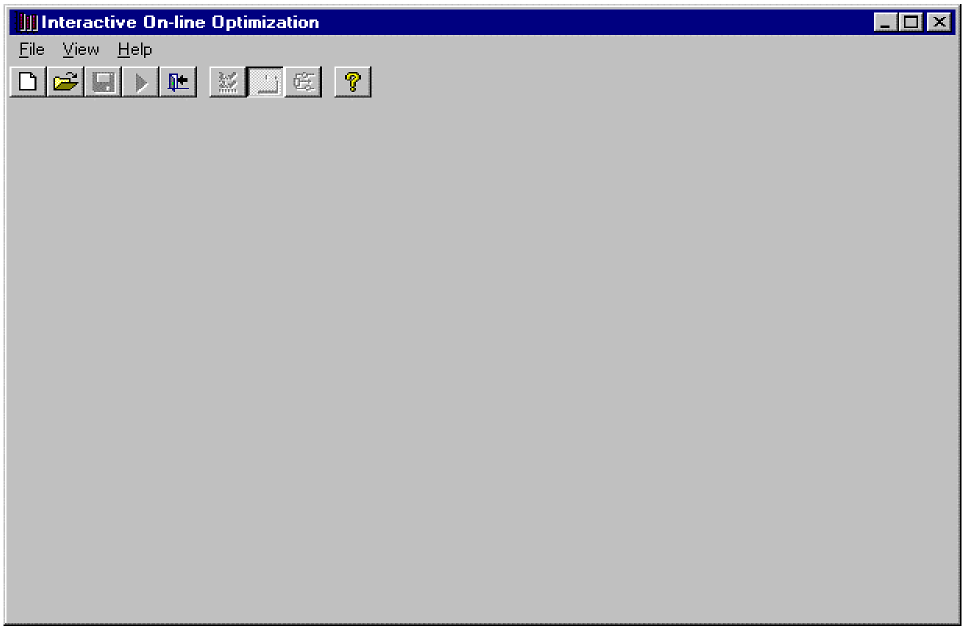


Figure 3 Main Window of the On-Line Optimization Program

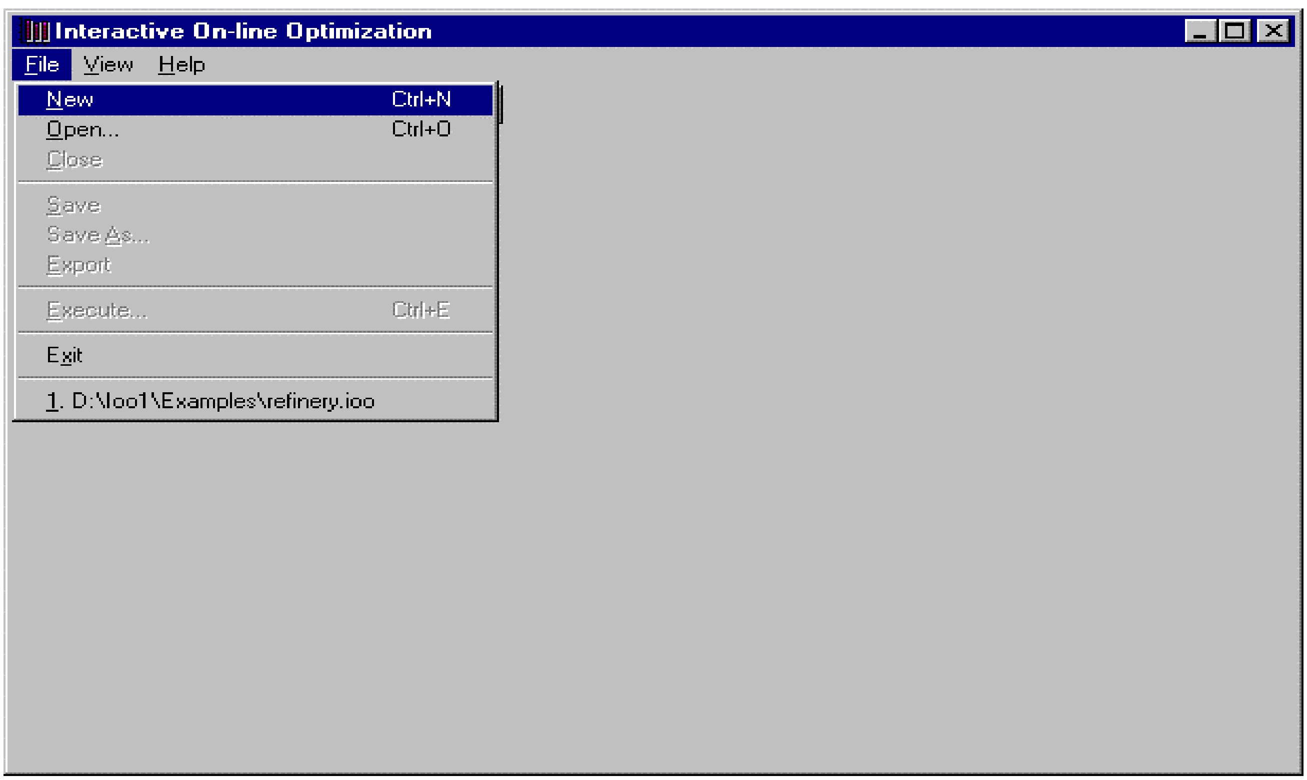


Figure 4 File Menu of the Main Window

While the model is being opened, the mouse cursor changes to an hourglass, which indicates that you are waiting for the application to finish an operation. When the operation is over, the mouse cursor returns to the Select arrow shape.

The “Save” and “Save As” options in the “File” menu are used to save the changes made to an existing model or to save a new model respectively. The “Save As” option can also be used to save an existing model under a different name. Clicking on the “Save As” option opens a Save dialog box as shown in Figure 6. After the user enters a name and clicks on “Save”, the database is saved under the given name with extension “ioo”.

If users want to save the database tables in Excel for viewing or printing, they can export the current loading table as an ”xls” file. The user should navigate to the window they want to export and then click on the “Export” option in the 'File' menu. This opens the Save dialog box. After the user enters a name, the details entered in the corresponding input window are saved as an Excel file under the given name.

Clicking on the ”Execute” button in the toolbar or by clicking on the “Execute” option in the File menu opens the “Model Summary and Execute window” to conduct the execution of On-line Optimization.

The Current Model can be closed by clicking on the “Close” option in the “File” Menu. Then the user can open an existing model or create a new one. Clicking 'Close' closes the current model but not the application. Exiting from the application can be done through the “Exit” button or through clicking on the “Exit” option in the 'File' menu.

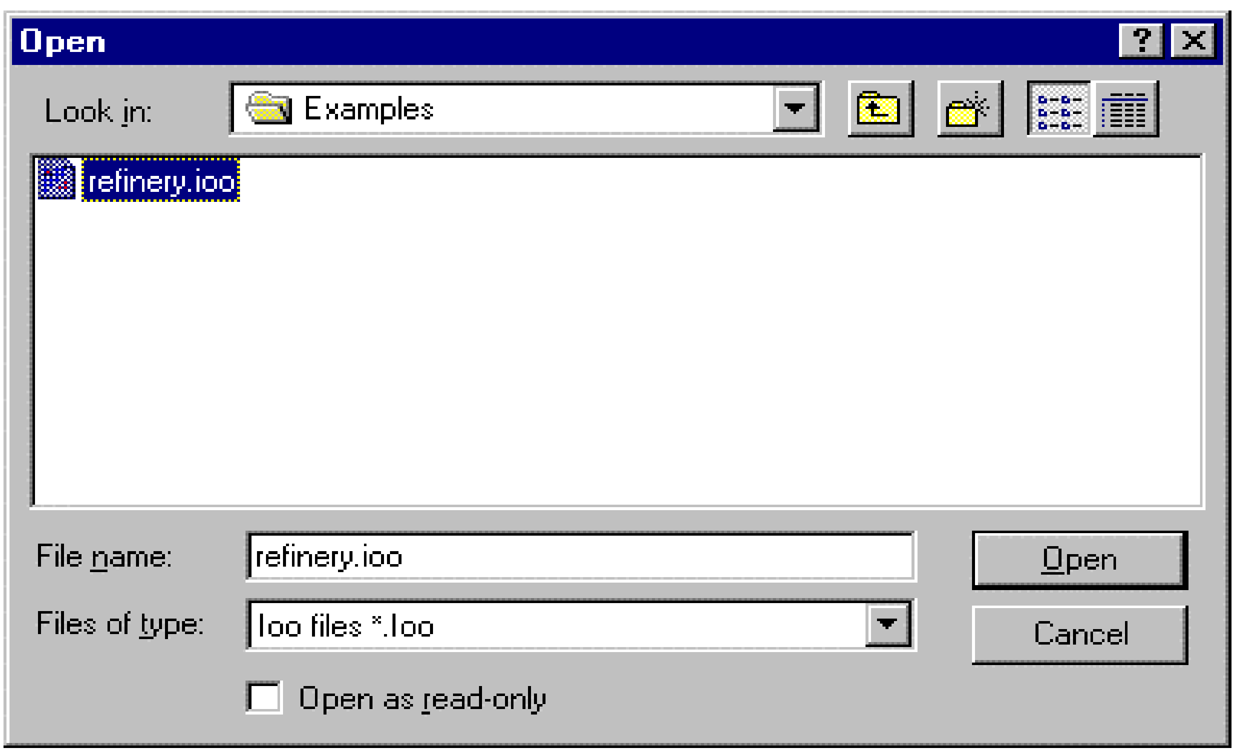


Figure 5 Open Window for an Existing On-Line Optimization Model

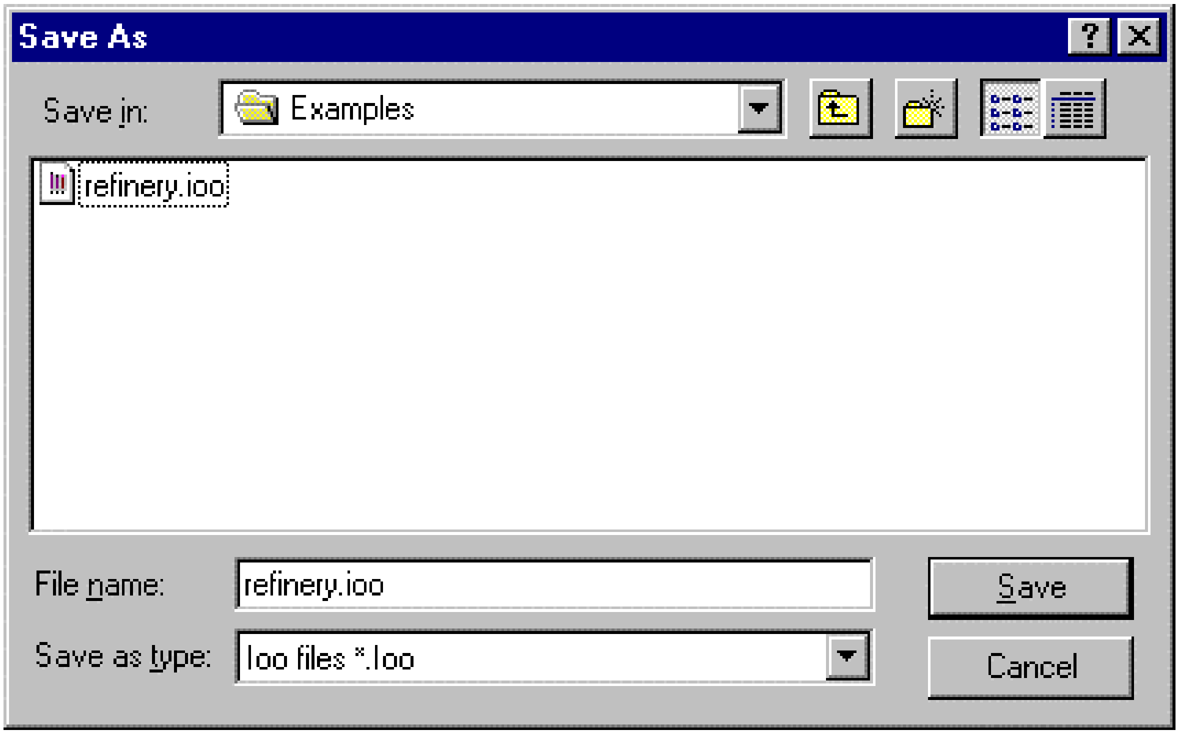


Figure 6 Save Window for On-Line Optimization Model

A. All Information Mode

When the user clicks on the View menu in the main window (Figure 3), a pull down menu is displayed as shown in Figure 7. The View menu includes the Optimization Algorithms, the All Information mode and Flowsheet Diagram. The All Information mode is used to switch model enter procedure. The Flowsheet Diagram is used to draw the process flow sheet diagram.

The All Information mode displays the different input windows combined together into one switchable window as shown in the Figure 8. The titles for each input window are listed in the text boxes at the top of the window. Engineers can switch between the input windows by highlighting the corresponding title. The model description window can be used to change the optimization objective from "On-line Optimization" to "Data Validation", "Parameter Estimation", "Economic Optimization" or "Parameter Estimation and Economic Optimization". This can be done by highlighting the title "General Description" and changing the option from the dropdown list of "Optimization Objective".

The user can also draw a flow sheet diagram and store the diagram as a part of the database. This facility can be invoked by clicking on the "Flowsheet" option in the "View" menu or by clicking on the "Flowsheet" button on the toolbar. A detailed description for drawing a flowsheet diagram is described in the section 'Drawing a Flow Sheet Diagram'.

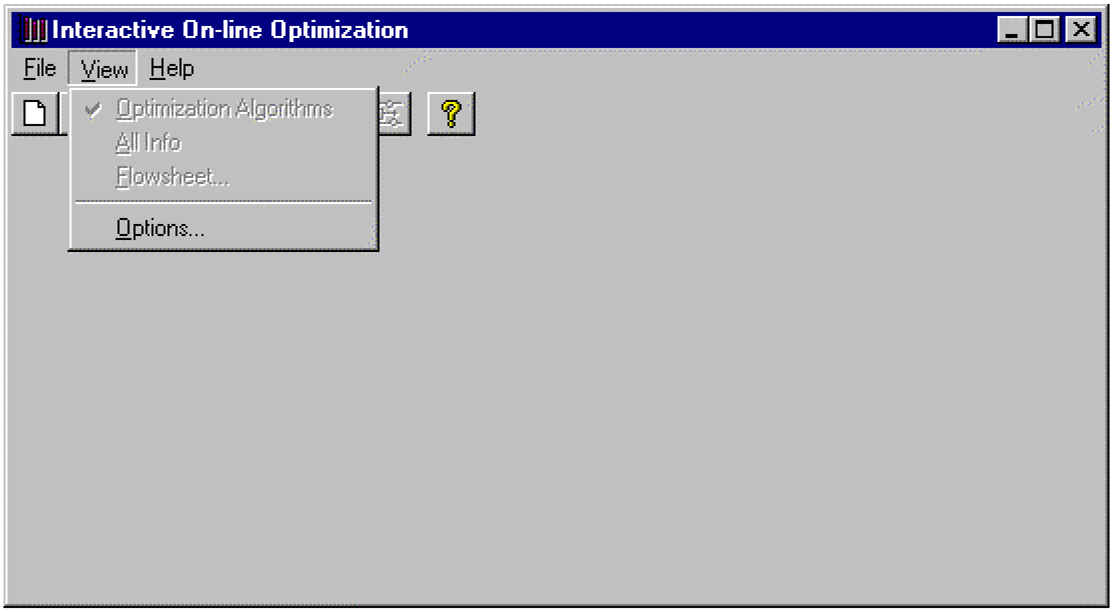


Figure 7 The View Menu of Main Window

If the flowsheet diagram already exists as a part of the database, it is automatically loaded. The user can then make and save any desired changes.

The Engineer can switch from one input window to another to make any modification. Clicking the “Help” button in the toolbar provides On-line assistance for the user. The HELP system is described in detail in the next section, “How to use Help”. The “Help” menu includes “Contents” which displays the index of topics for which help is available, 'Search' which allows the user to search for a keyword in the 'Help' file, and 'About' which displays the Application details such as the system information and the version number.

B. Introduction to Grids

Grids are controls, which are used for displaying data from a database. Data can be entered and edited in a grid. Resizing the width of a grid can be done by pointing the mouse to the line dividing the columns at the column header. When the cursor changes to the resize cursor, drag the mouse by holding the mouse left button down. The same procedure can be used for resizing the height of a grid, i.e., point the mouse to the line dividing the rows at the row head and drag the mouse by holding the left mouse button down. For deleting a row or a column, move the cursor on the column header or the left side of the grid. This changes the cursor to an arrow shape. After positioning the cursor, click on the row or column to be selected (this selects the entire row or column), then press the delete key to delete the selected row or column.

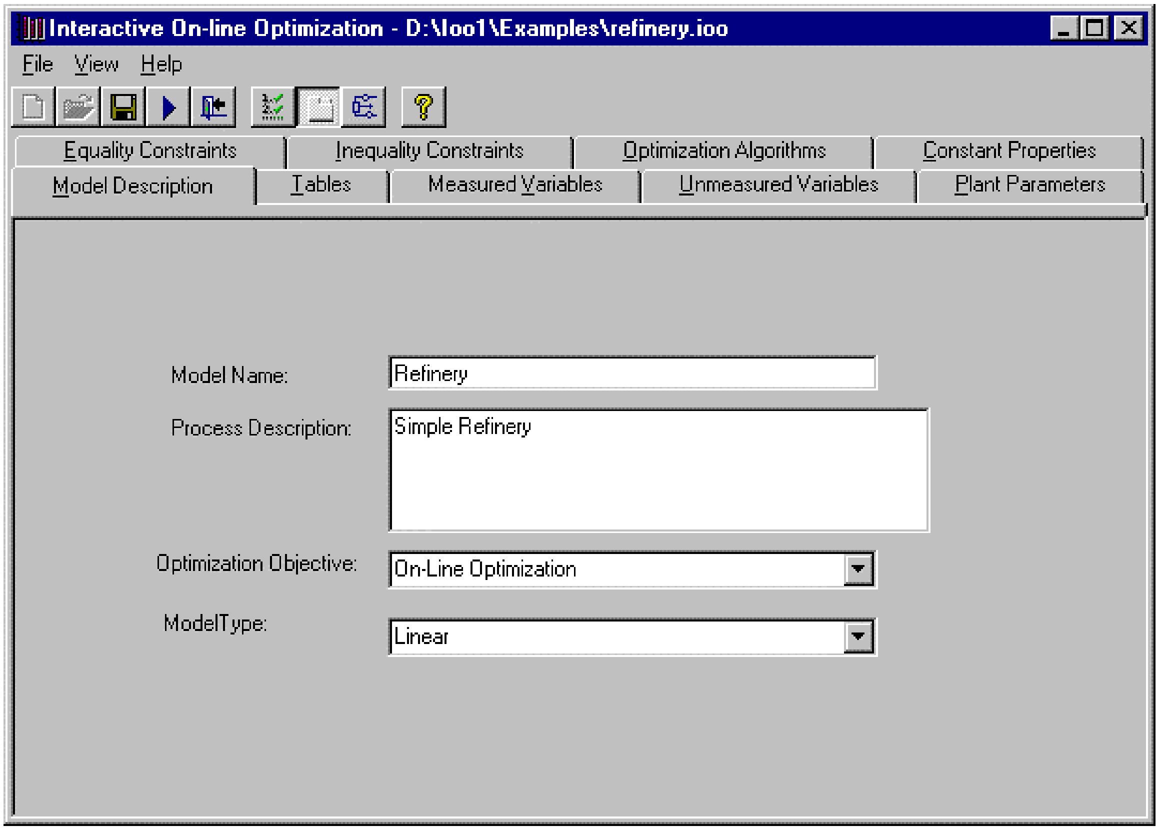


Figure 8 Model Description Window in All Information Mode

IV. HOW TO USE HELP

During the application of the on-line optimization system, the HELP system is available at any time to assist the process engineer entering information. There are two types of help systems provided by the on-line optimization system. The first one is obtained by pressing the AF1" key, which opens a HELP window. The complete HELP information is provided in "Table of Content" format. The HELP information is given in a sequence as the development of the on-line optimization model. However, the engineer can quickly access the desired topic by using a keyword index search. Clicking a key word opens a HELP window that provides detailed information on the subject related to the key word. The second type of HELP is "What=s this ?" , which is active only to the specific highlighted object in the window. To use "What=s this?" HELP, the user must first highlight an object, then click the right mouse button to bring out a drop-down menu with the feature "What=s this ? ". Clicking "What=s this ?" opens a window that gives an explanation about the definition and function of that object.

The AF1@ key for HELP is a general method, which can be used to find specific information through keyword index search. However, 'What=s this?' HELP only corresponds to the highlighted object in a specific input window.

V. STRUCTURE OF THE INTERACTIVE ON-LINE OPTIMIZATION SYSTEM

The structure of on-line optimization was given in Figure 1, and the information and elements required to conduct on-line optimization are shown in Table 1. Also, Table 1 gives the elements of the on-line optimization system. First, process and economic models are entered, and algorithms for data validation and parameter estimation are specified. Then, the system constructs GAMS source codes for three optimization problems. In the execution section the system uses GAMS to solve these three optimization problems in sequence. In the solution summary section, the system generates a final report file that summarizes important results. GAMS also generates and displays three detailed output files corresponding to the three optimization problems. These files include detailed information about the programs and solutions.

Table 1 The Structure of On-Line Optimization

- Elements of On-Line Optimization
  - Plant Model
    - Model characteristics
    - Measured and unmeasured variable declaration
    - Plant data and standard deviation (S.D.)
    - Process parameters
    - Constraints (equality and inequality)
    - Constant tables
    - Variable bounds and initial points of variables
    - Scaling factors for variables and equations
  - Economic Model
  - Algorithms of Data Validation and Parameter Estimation
- Execution of On-Line Optimization
- Result for the Distributed Control System

A. Process and Economic Models

To apply the on-line optimization to a process, the information about the process must be provided by the user for the interactive on-line optimization system to develop the programs for the three optimization problems. As listed in Table 1, the input information required to conduct on-line optimization consists of plant and economic models, as well as algorithms for gross error detection and data reconciliation and parameter estimation. The plant model includes general information about the model (description and model type), measured and unmeasured variable declaration, plant data sampled from distributed control system and standard deviation, process parameters, constraints (equality and inequality), constant tables, bound and initial point of variables, and scaling factors for variables and equations. The details of the input procedure through the system are illustrated in the following sections.

Upon entering the system, the process engineer is asked to choose whether to build a new process model or edit an existing one as shown in Figure 2. If a new process model (ANew Model@) is chosen, then the plant information must be entered as discussed in section A-1 to A-4. If an existing model (AOpen Model@) is selected, the previously saved model can be reloaded to the windows. From there, the process engineer can make any modifications needed.

A-1. Process Model:

The information about a plant includes process model description, measured and unmeasured variables, process parameters, constraints (equality and inequality), bounds and initial points for the variables. The system will guide the process engineer through all of the corresponding input windows in sequence so he/she can complete the plant model. The following describes the input of the on-line optimization model using a simple refinery example (Pike, 1986) to illustrate the procedure. The detailed process description about this refinery is given in Section VII, Example.

Model Description:

If the “New Model” in Figure 2 or the “New” option in the “File” menu is chosen, the Model Description window is provided for process engineers to enter the general information about the models shown in Figure 8. It includes Model Name, Process Description, Optimization Objective, and Model Type. Model Name identifies the process to the on-line optimization system. The name of the process model is entered in a text box (the blank square block) beside the caption entitled “Model Name”. A descriptive name for the plant is recommended because it makes further editing easier. However, it must be less than ten characters without spaces among the characters. An optional description of the process may also be entered in the text box situated across from the caption entitled “Process Description”.

The optimization objective can be selected from the drop-down list of “Optimization Objective”. Five selections are “On-Line Optimization”, “Data Validation”, “Parameter

Estimation", "Economic Optimization", and "Parameter Estimation and Economic Optimization". They are described below.

- When "On-line Optimization" is chosen, the three problems (data validation, parameter estimation, economic optimization) are executed in sequence. When this objective is chosen, the user must enter the following information: plant data for measured variables, equality constraints, plant parameters and its initial values, algorithms for data validation and parameter estimation and the economic objective. Also, additional information such as tables and scalars may be entered if they exist in the equations. Tables and scalars are used to define the constants in equations and unmeasured variables

- When only "data validation" is chosen, only 'data validation' is executed. For this case, the required information is plant data for measured variables, equality constraints, and algorithms for data validation. Plant parameters must be given the initial values if the plant model has parameters, otherwise, they are not required. Also, additional information such as tables, scalars, and unmeasured variables may be entered if they exist in equations.

- When only Aparameter estimation@ is chosen, the GAMS program uses the plant data for the measured variables instead of using the reconciled data from data validation. The required information is plant data for measured variables, equality constraints, plant parameters, and the algorithm for parameter estimation. Also additional information such as tables, scalars, unmeasured variables may be entered if they exist in equations.

- When only Aeconomic optimization@ is chosen, the initial values of the plant parameters are used instead of the estimated values from parameter estimation in the GAMS program for economic optimization. The required information are equality constraints and/or inequality constraints, plant parameters and their values (if they exist in the plant model), the economic model, and measured variables (and/or unmeasured variables).

- When AParameter Estimation and Economic Optimization@ is chosen, then the user must enter the following information: measured variables and plant data, equality constraints and/or inequality constraints, plant parameters, algorithms for parameter estimation, and economic model. The economic optimization program uses the estimated value of the plant parameters from Parameter Estimation.

Finally, the Model Type of the plant model must be specified. Either Linear or Nonlinear is selected from the drop-down list. The type of the plant model should agree with the model formulation (constraint equations). If the model has nonlinear constraints, then the ANonlinear@ option should be chosen. Otherwise, the ALinear@ option is selected. Only one type can be chosen for each model. The default type for the plant model is ANonlinear@. For the simple refinery, the name of the plant model is AREFINERY@ and all process constraints are linear as shown in Figure 8.

When the information for this form is completed, users can use the All Information mode to move to any other input window.

Constant Tables:

A convenient option is given by the creation of the Tables window. The constant coefficients used in the constraints equations can be defined in the Tables window. These constant coefficients can be grouped in sets, and they can be defined using concise names which refer to their values in the equations before an equation definition. Then, the names of constants will appear in the equations replacing the numerical values. This avoids typing errors from reentering the numbers, and it makes the program more concise as well as easier to read and to check. A common example is to use a table to define the constant coefficients in a polynomial function for enthalpy. The enthalpy of a stream usually is expressed as a polynomial function of the temperature and flow rate. This function appears repeatedly in the plant model with the same coefficients, which have different numerical values for each chemical component. An example is:

$$h_i = a_{1i} + a_{2i}\,T + a_{3i}\,T^2 + a_{4i}\,T^3 + a_{5i}\,T^4$$

where the table will contain the five coefficients, a_{1i} to a_{5i}, for component i.

To create a table, first click AAdd New@ in the Tables window (Figure 9.a.) This activates the input window for entering general information. As soon as the AAdd New@ button is clicked, the caption of the AAdd New@ button changes to ASave@ and that of ADelete@ changes to ACancel@ and the edit button is disabled as shown in Figure 9.b. Then the general information of a table: the name of the table, rows and columns as well as the dimensions of columns, must be entered from the input window as shown in Figure 9.b. The name of the table stands for the name of the coefficient group. The names of rows and columns are the set names of the sub-components. For the coefficients of enthalpy equations, the table can be defined by using ACoef@ as the table name, AComp@ as the name of the component group (row name), and AEnt_Coe@ as the name of the constant enthalpy coefficient group for each component (column name). Also, the corresponding dimension of each column is the number of constant coefficients in an enthalpy equation for a component, which is 5 in this example.

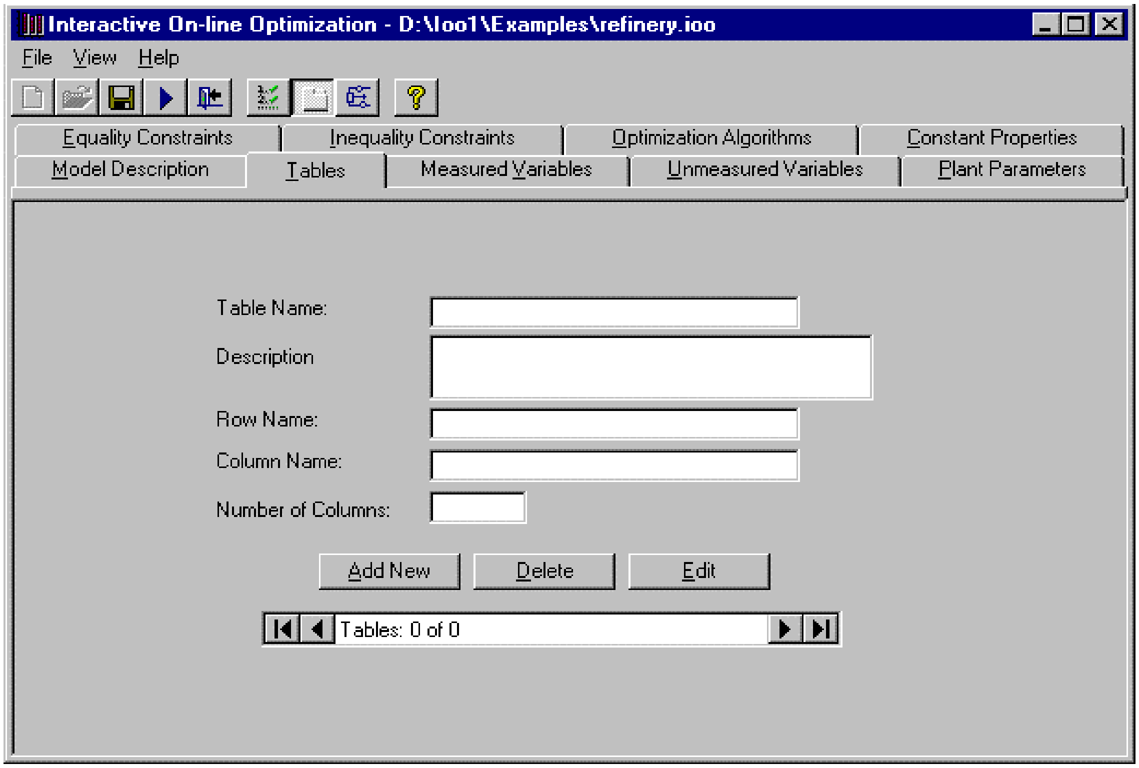


Figure 9.a. The Creation of Tables Window

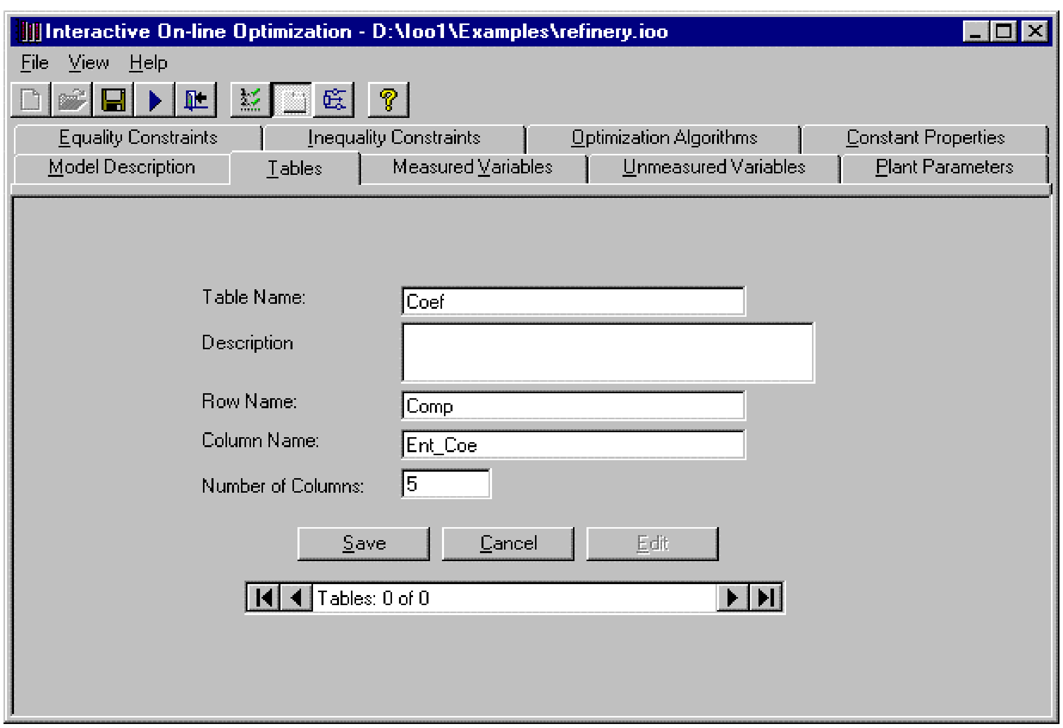


Figure 9.b The Creation of Tables Window

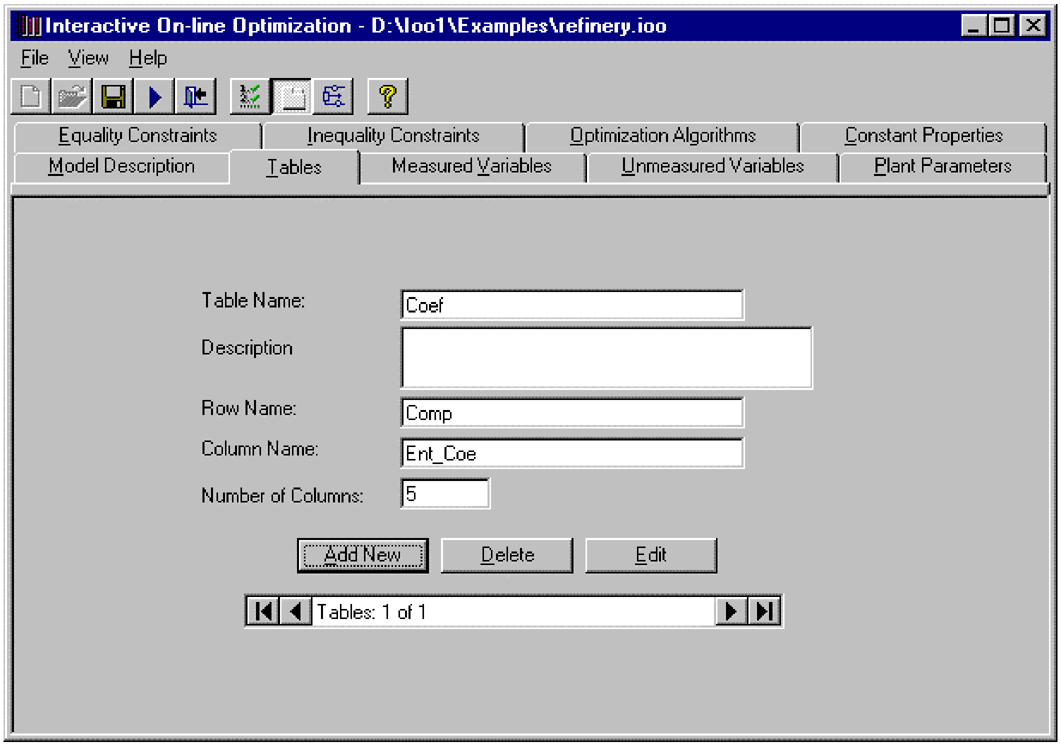


Figure 9.c. The Creation of Tables Window

After entering the table information, the user needs to click the ASave@ button to activate the AEdit@ button for editing the tables or click ACancel@ to discard the information just entered and return to the window shown in Figure 9.c.

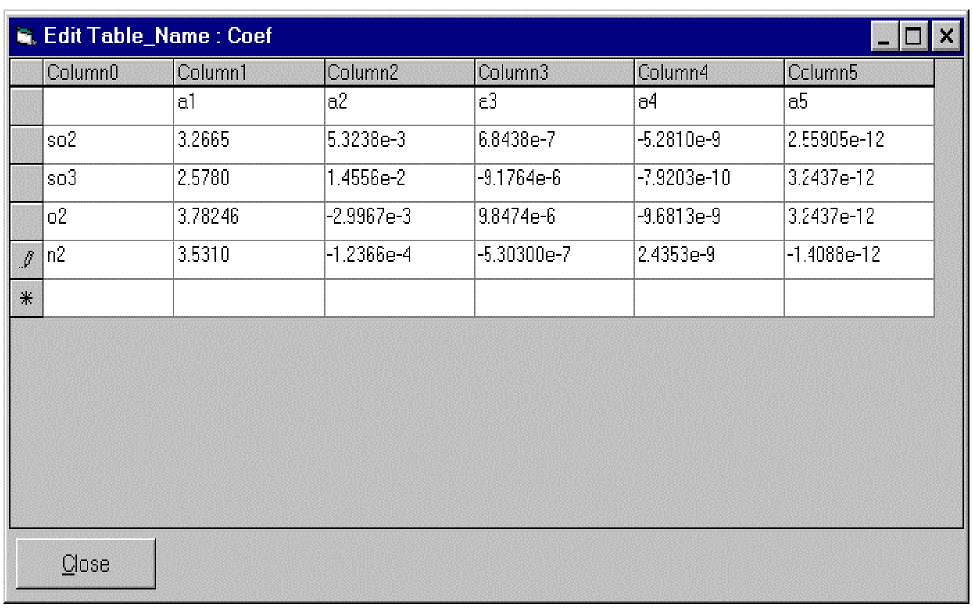
The table content window is opened by clicking on "Edit". The table content window is used for entering the names and numerical values of the constant coefficients shown in Figure 10. The table content window has the window title as the table name specified in the previous Tables window for the corresponding table. As shown in Figure 10, the table content window for table ACoef@ has the window title as ACoef@ which is the name of the table. This table has 6 columns (the defined column number + 1) and unlimited rows. The user must enter the names of each enthalpy coefficient in the first row. Beginning with the second row, the name of a component must be entered in the first cell of each row. All other cells must contain the numerical values of the constant coefficients for the corresponding components. When the table is completed, clicking AClose@ will update the table and bring users to the previous Tables window (Figure 9).

The user can create multiple tables sequentially by following the above procedure, i.e., first clicking AAdd New@ to enable the Table input form (Figure 9.c.) for entering general information about the table (name of the table, rows and columns, as well as the dimension of columns), then clicking ASave@ to activate the AEdit@ button as shown in Figure 9.b. Finally, The user may click on AEdit@ to open the table content window (Figure 10) for entering the names and numerical values of the constants coefficients. An existing table can be edited or deleted by selecting the table and then clicking AEdit@ or ADelete@. The Scroll Bar at the bottom of the window (Figure 9) is provided to browse and select tables. The user can use the Scroll Bar to select a table for editing. Clicking AEdit@ opens the corresponding table content window for editing. Since there are no Tables in the Simple Refiney , Tables window of Dsulfuric process shown in figure 10.

Measured and Unmeasured Variables:

The variables existing in constraints must be declared in variable declaration windows. For a process model, variables include temperatures, flow rates, compositions, and pressures. These variables are either measured or unmeasured. The measured variables have data sampled from the distributed control system or from the plant =s analytical laboratory. Unmeasured variables are the variables in the constraint equations for which no measurements are available. Two separate windows for declaring the measured and unmeasured variables are shown in Figure 11 and 12 respectively.

As shown in Figure 11, the Measured Variables window is an eleven column table used for entering variable names, plant data sampled from the distributed control system, standard deviation, initial points, lower and upper descriptions about the variables, and scaling factors. The information shown in Figure 11 is for the simple refinery, and it was taken from Table 20 of Section VII, Example, Simple Refinery.



Column0	Column1	Column2	Column3	Column4	Column5
	a1	a2	a3	a4	a5
so2	3.2665	5.3238e-3	6.8438e-7	-5.2810e-9	2.55905e-12
so3	2.5780	1.4556e-2	-9.1764e-6	-7.9203e-10	3.2437e-12
o2	3.78246	-2.9967e-3	9.8474e-6	-9.6813e-9	3.2437e-12
n2	3.5310	-1.2366e-4	-5.30300e-7	2.4353e-9	-1.4088e-12

Figure 10 Specification Window for the Table Content

The variable name, plant sampled data, and standard deviation for all measured variables are required for the optimization objective of on-line optimization, data validation, parameter estimation, or parameter estimation and economic optimization. The plant data is obtained from the distributed control system or control laboratory. The standard deviation of the plant data are determined using historical plant data. For conducting economic optimization alone, the plant data and standard deviations are not required. The upper and lower bounds, scaling factors, description, stream number and Process Unit ID of the variable are optional. The upper and lower bounds of a variable specify the range of values allowed for the variable. The upper bound and lower bound are the largest and smallest values for the variable. If upper and lower bounds are not specified, the default values are used which are negative infinity (-INF) for lower bound and positive infinity (+ INF) for upper bound.

The variable ‘s initial point provides the starting point for the optimization. If the initial points are not specified, the default initial point will be used. The default initial points for the measured variables are the plant data sampled from the distributed control system. The default initial points for the unmeasured variables are zero. The plant data provides accurate initial points for measured variables and is used automatically as the initial points for measured variables by the system.

For linear optimization problems, initial points are not necessary for the unmeasured variables, but they will improve the speed to reach the optimal solution for very large optimization problems. For nonlinear optimization problems, an initial point is necessary for solving optimization problems. They give the starting points that are close to the optimal solution, which aids in reaching the optimal solution. Also, the default initial point for unmeasured variables is zero which may cause an execution error (division by zero error) when GAMS linearizes the nonlinear equations. Although the bounds and initial points for variables are optional, it is recommended that this information be provided for more optimal solutions.

Optimization programs need all of the variables in the same numerical range, so it may be necessary to scale the variables by adjusting the units. In most cases, the units of a variable are selected to have a value around unity. If some of the variables are not well scaled, i.e., their values are out of the range of 0.01 to 100, then these variables should be scaled using the Scaling factor column to specify the value of the scaling factor.

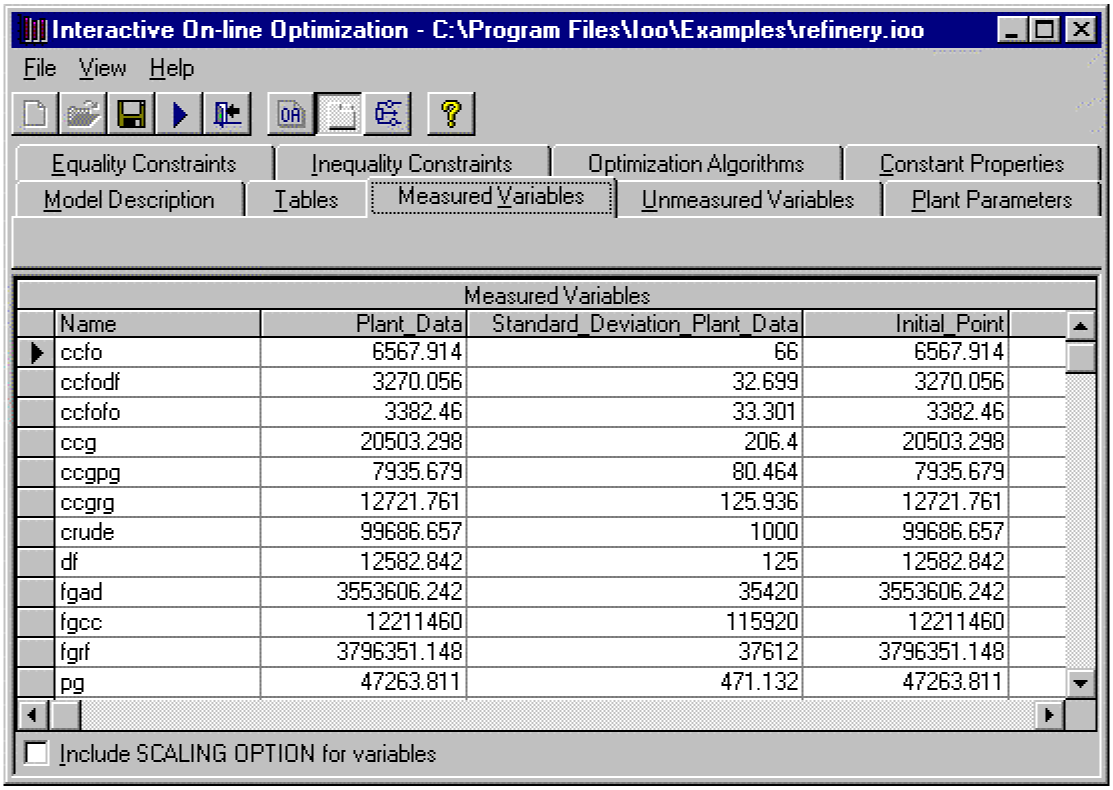


Figure 11 Declaration Window for Measured Variables

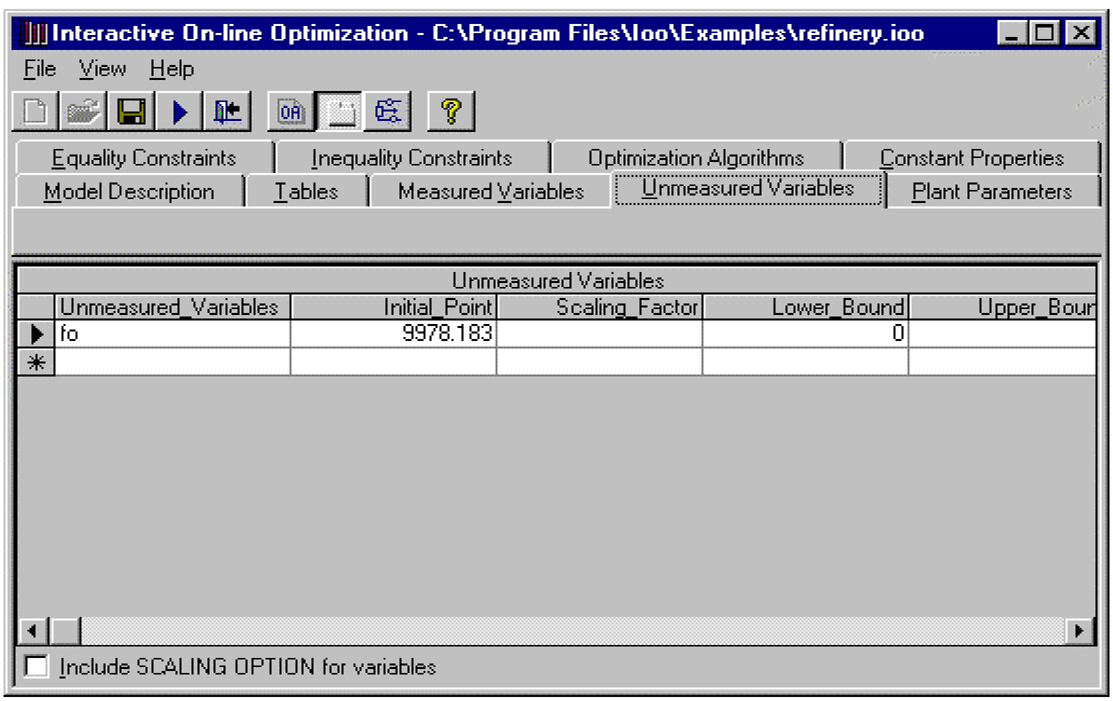


Figure 12 Declaration Window for Unmeasured Variables

To scale variables using the Scaling Option provided by the system, the scale factor for the variable that needs to be scaled must be entered in the corresponding cell, and the icon of 'Include Scaling Option for variables' at the bottom of the window must be highlighted. The details for the determination of scaling factors are discussed in the "Scaling Option for Variables and Equations" of the Optimization Solver section.

The stream number of a variable is the name of the stream to which the variable belongs, and the Process UnitID is the name of the Process Unit to which the variable is associated. Although the stream number and Process UnitID of a variable are optional, their information will be used by the system to organize the solution presentation in the Final Report and Flowsheet Diagram. If the user specifies the Stream number or Process UnitID of the variable, then the solution of this variable can be viewed in the stream summary and the Flowsheet diagram of Final Report through the stream number or Process UnitID search. Otherwise, the solution of this variable will not be available in the Flowsheet presentation and cannot be searched by a stream number or a Process UnitID. Therefore, it is recommended to give the stream number or the Process UnitID for important variables such as flow rate, temperature, pressure, and composition.

The Unit of the process variable and description are provided for the user to give the unit of the variable and a brief description for the variable. They are optional.

In Figure 12, the Unmeasured Variables window is shown. It includes nine columns for entering the variable name, initial point, scaling factor, lower and upper bounds, stream number, Process UnitID, Unit of the Process Variable, and a description for each unmeasured variable. If some of the variables in the constraints are unmeasured, then they must be declared in this window. The procedure for entering the unmeasured variables in this window is the same for entering measured variables. The initial point, scaling factor, bounds, stream number, and description for an unmeasured variable are all optional as they are for measured variables. Also, if there is no unmeasured variable in the plant model, this window is not required, and the ANext@ button should be pressed.

For the simple refinery process, the measured and unmeasured process variables were given in Table 20 of Section VII, Example. The plant sampled data and standard deviation for measured variables were also given in this table. This information is entered in the interactive on-line optimization system and is shown in Figure 11 and Figure 12. For the simple refinery process, the measured variables should be scaled. So the box next to AInclude Scaling Option for variables@ should be checked as shown in Figure 11.

Parameters in the Process Model:

Parameters are unmeasurable and vary slowly in the plant model. Some examples are the heat transfer coefficient, the catalyst activity coefficient, and the tray efficiency. Their values change slowly with time, so they are considered constants over the time intervals for optimization. The Parameters window includes six columns for entering information about names, initial values, lower and upper bounds of plant parameters, the Process Unit ID (to which the parameter is associated with), and the unit of the parameter as shown in Figure 13. The names and initial values for the parameters must be provided by the user which are estimated using reconciled data from data validation. The bounds of the parameters are optional. If the plant model does not have any parameters to be estimated, then this input window can be left empty, and the A Next @ button should be clicked to move to the next window.

The parameters of the simple refinery were listed in Table 22 of Section VII, Examples. These are entered in the interactive on-line optimization system as shown in Figure 13. In this figure, the first row of the parameter list table is A vfgad 35.42 0 100 AD BBL/BBL@, which means that the parameter has the name vfgad, initial value 35.42, lower bound of 0 and upper bound of 100. This parameter is associated with the Process unit AD, and the unit of this parameter is BBL/BBL

Equality and Inequality Equations :

The constraint equations are the plant simulation which include equality and inequality constraints. These equations describe the relationship among process variables. The equality constraints include material and energy balances, reaction rate equations, and equilibrium relations. Inequality constraints provide the limits on equipment capacities, raw material availabilities, and demand for products.

For each equation input window, two columns are provided as shown in Figure 14 and

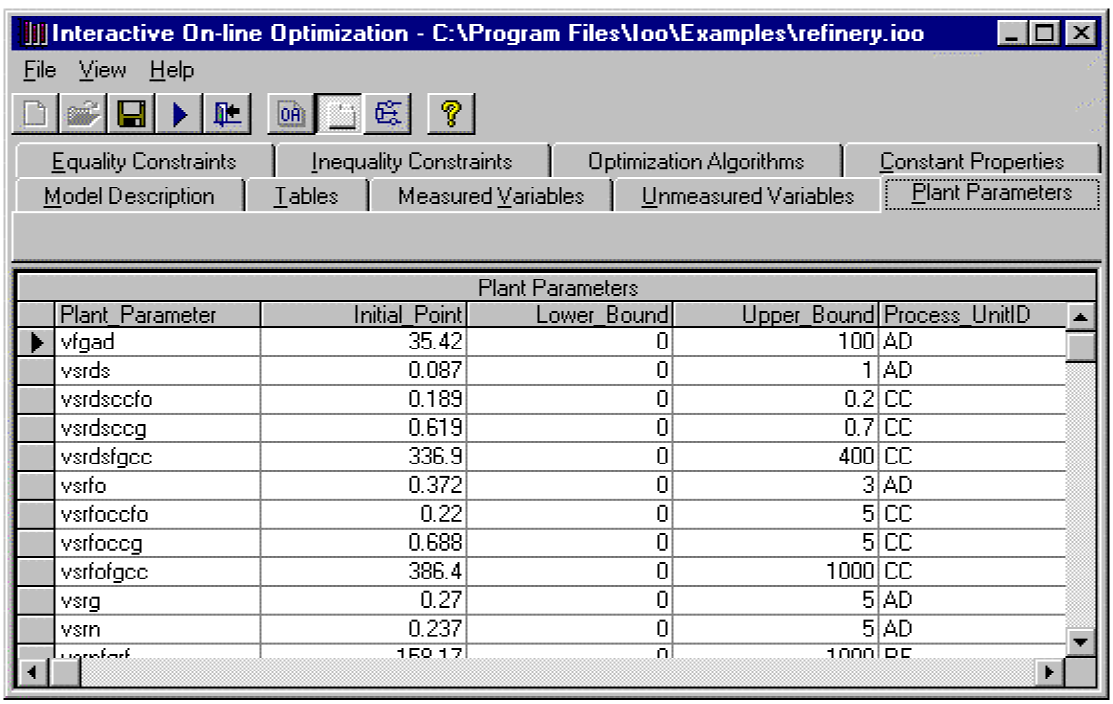


Figure 13 Declaration Window for Parameters in the Plant Model

Figure 15. The first column is used to enter the equation and the second column is used to specify the value of the scale factor for this equation. The equation can be directly entered and modified in the Equation column one by one.

The formulation of the constraint equations required by the system is similar to the mathematical formulations in the general program application. The only difference is that it uses A =E=@ to represent the equality sign instead of A=@, A=L=@ for the less than or equal sign instead of A#@, and A=G=@ for the greater than or equal sign instead of A$@. An additional description of the equation format is given in the Optimization Solver section.

The scale factors for equations are not required if the coefficients in the equations are well scaled, i.e., their values are around 1.0. If the coefficients of an equation are out of the range of 0.01 to 100, a scale factor should be given to scale the coefficients to the appropriate range and is entered in the Scaling Factor column. To scale an equation, first the scale factor for the equation must be entered in the cell corresponding to the equation in the Scale Factor column, then the icon A Include Scaling Option for Equation @ must be highlighted. Details about the Scaling Option are found in the optimization solver section.

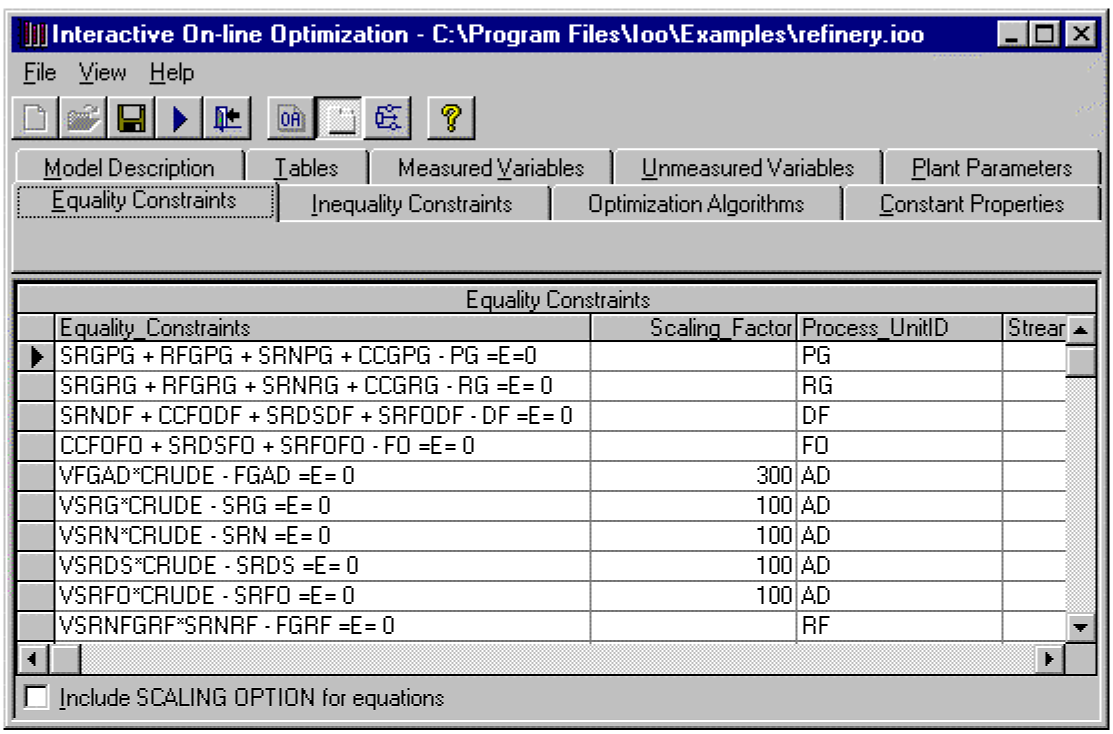


Figure 14 Declaration Window for Equality Constraints

The equality constraints are entered in the Equality Constraints window shown in Figure 14. Inequality constraints are entered in the Inequality Constraints window shown in Figure 15. These constraints are included only in the economic optimization program. The constraints for the simple refinery are entered in the interactive on-line optimization system, which are shown in Figure 14 and 15. These equations are given in Table 23 and Table 24 of Section VII, Example. For the simple refinery, the A Include Scaling Option for Equations @ box in the equality constraints window should be checked as shown in Figure 14.

The next step is selecting algorithms for conducting gross error detection, data reconciliation (data validation), and parameter estimation.

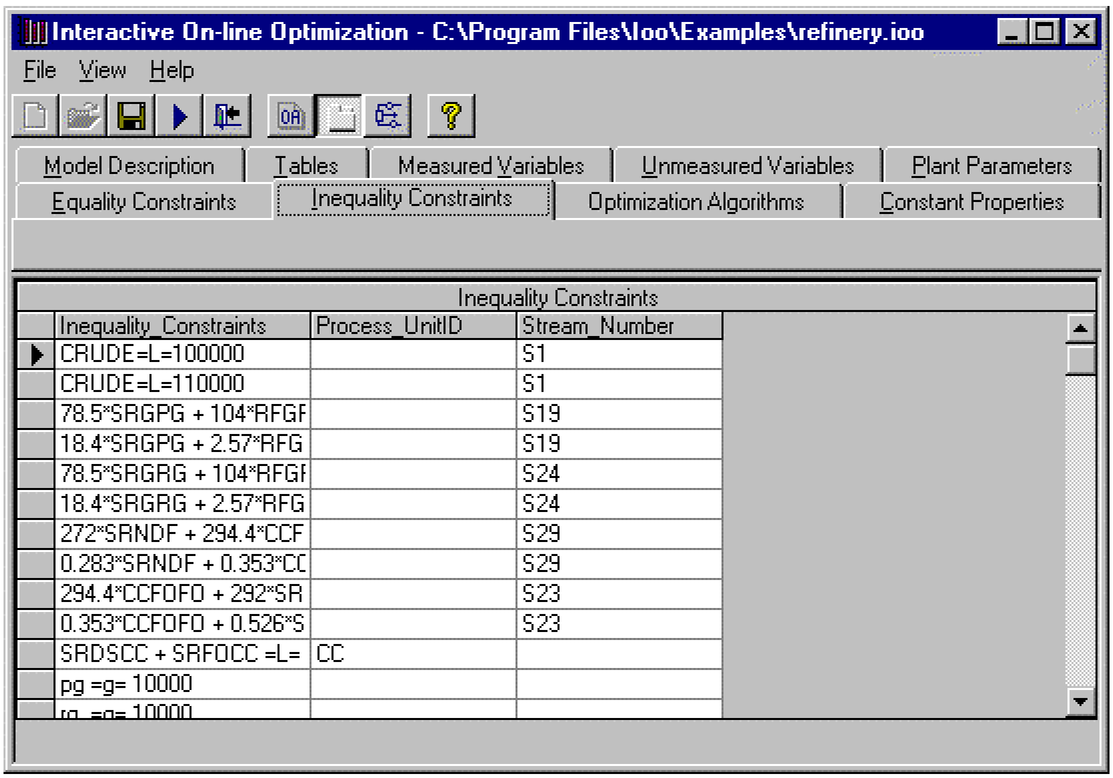


Figure 15 Declaration Window for Inequality Constraints

A-2. Algorithms for Gross Error Detection and Data Reconciliation and for Parameter Estimation

This section describes the selection of algorithms for conducting combined gross error detection, data reconciliation (data validation), and simultaneous data reconciliation and parameter estimation (parameter estimation). When A On-Line Optimization @ is selected as the optimization objective, two algorithms must be specified. One algorithm is specified for gross error detection and data reconciliation; the other is specified for simultaneous data reconciliation and parameter estimation. One of three alternative algorithms must be selected from a drop-down list as shown in Figure 16. The three alternative algorithms are the Least Squares Method, the T-B Method, and the Robust Function Method. The default options are for Tjoa-Biegler=s method for data validation and the Least Squares method for parameter estimation. The objective function for the specified algorithm will be formulated automatically by the system.

The selection of algorithms is based on the character of the plant data. The Least Squares Method should be chosen for gross error detection and data reconciliation when plant measurements contain random and small gross errors. The T-B (Tjoa-Biegler) Method is preferred when measurements contain random and moderate size gross errors (5-30 times of the standard deviation). The Robust Function Method is preferred when measurements contain random and very large gross errors (larger than 30 times of the standard deviation). For parameter estimation, the system constructs a set of reconciled plant data containing only random errors. Therefore, the Least Squares Method should be chosen for parameter estimation instead of the T-B or Robust Function Methods. These methods are described in detail by Chen, 1998.

A-3. Economic Model:

The economic model must be entered directly in the text box following the caption “Objective Function for Economic Optimization” shown in Figure 16. In the Economic Optimization, the objective function can be a profit function that is a simple value added model, or it can be more elaborate and include a range of manufacturing costs. Waste reduction may also be incorporated into the economic optimization model in various ways.

The formulation of the objective function includes the same mathematical format as the constraints, except an equal sign is not used. This is shown in Figure 16 using the profit function for the simple refinery. This function may be located in the examples section under equation (1), section 7. Also, the optimization direction (either maximizing or minimizing) and the economic model type (either linear or nonlinear) must be specified in the corresponding drop-down lists shown in Figure 16. The optimization direction specifies either maximizing or minimizing the economic model. The economic model is either linear or nonlinear depending on the formulation of the economic model. When the user does not specify the optimization direction or the economic model type, the system takes the default values “maximizing” and “nonlinear”. After completing both process and economic models, the “Execute” button in the toolbar or the Execute option in the File menu must be clicked to execute the On-line Optimization program.

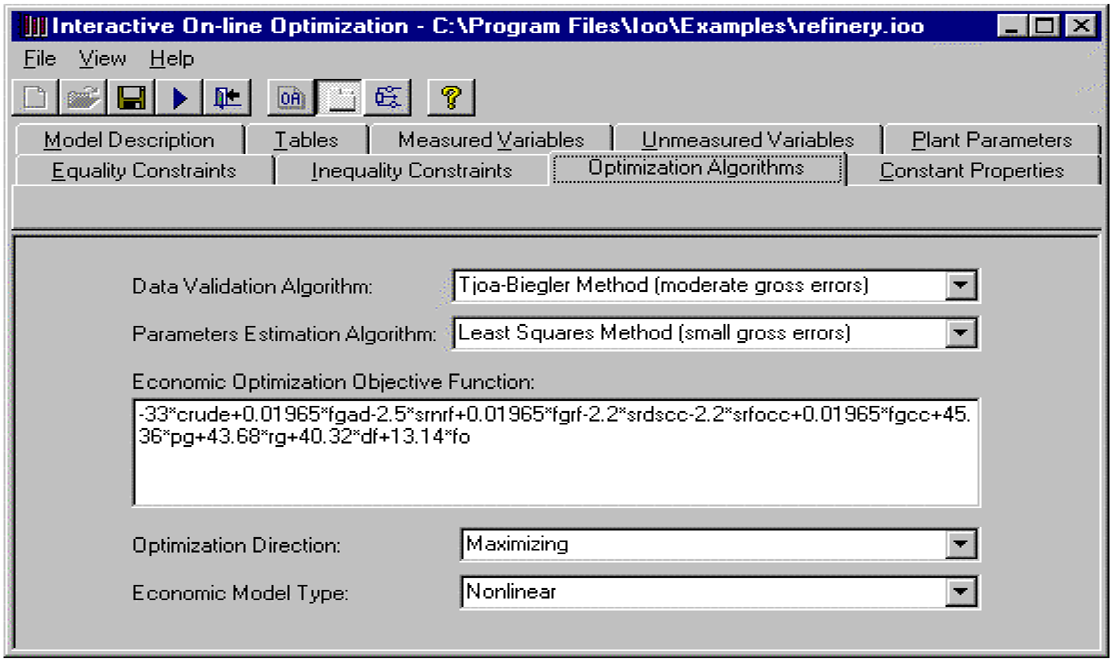


Figure 16 Specification of Algorithms and Economic Model

A-4. Summary of Process and Economic Models

Now all components required to conduct On-line Optimization have been completed. These components include the plant model, the economic model, and the algorithms for data validation and parameter estimation. In addition, the plant model includes a model description, a variable declaration, current plant sampled data from the distributed control system, a standard deviation, a parameter declaration, constraint equations, constants, bounds, initial points, scaling factors, stream numbers, and a Process Unit ID.

Clicking on the 'Options' item in 'View' menu, opens the Options window as shown in Figure 17. General GAMS Process options are set in the 'GAMS Process' tab as shown in the first window of Figure 17. The format for the GAMS output can be specified in the 'Output Format' tab as shown in second window of Figure 17. LP and NLP values for the Solver can be set in the 'Solver' tab as shown in the third window of Figure 17. The default values are CONOPT for both LP and NLP. These default values can be restored by clicking on the 'Use Defaults' button. Solver Parameters like Number of Iterations, Number of Domain Errors and Amount of Time Used can be specified in the 'Solver Parameters' tab as shown in the fourth window of Figure 17. Other advanced options can be set by clicking on the 'Advanced Options' button, which brings up the window shown in Figure 17-A.

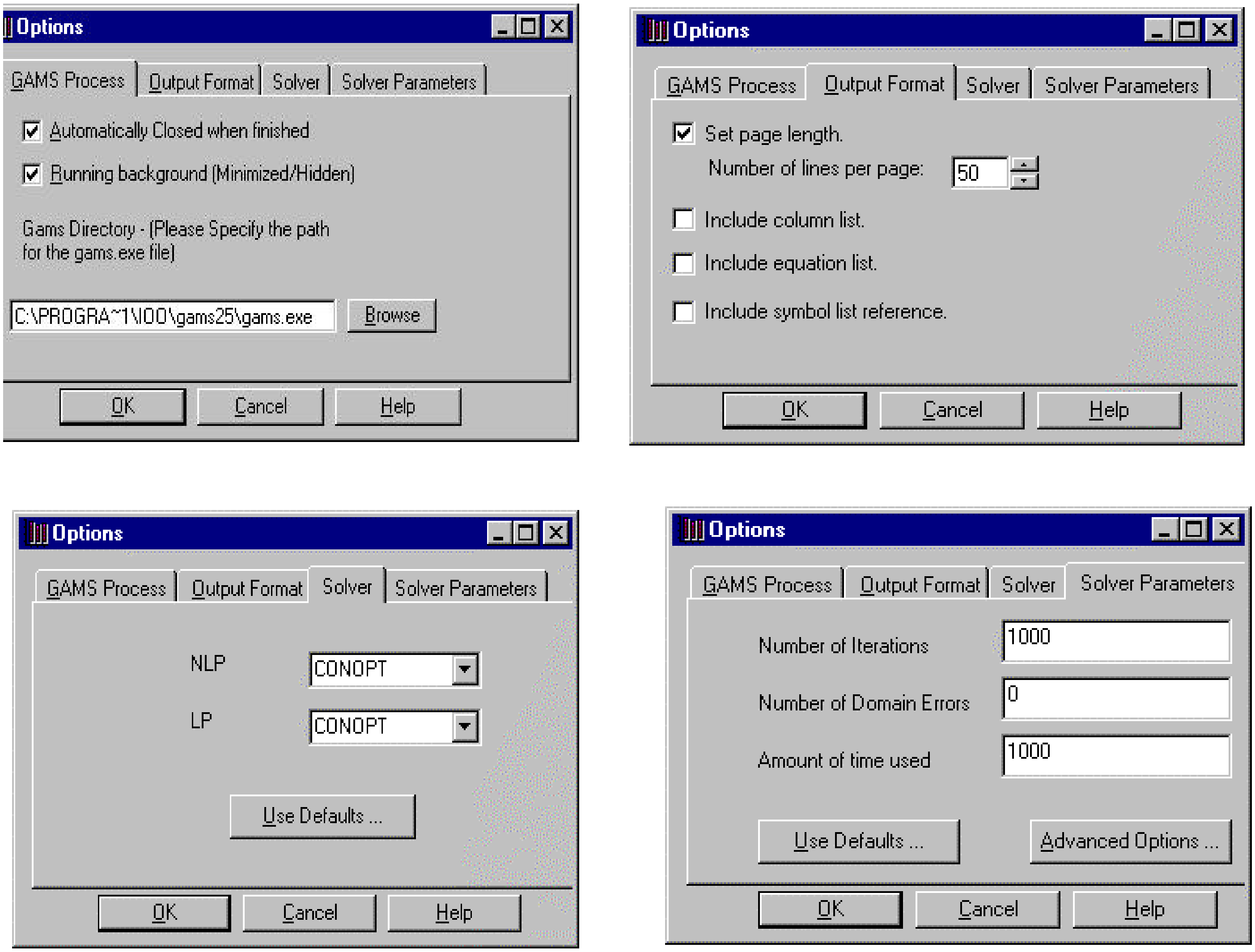


Figure 17 User Options Windows

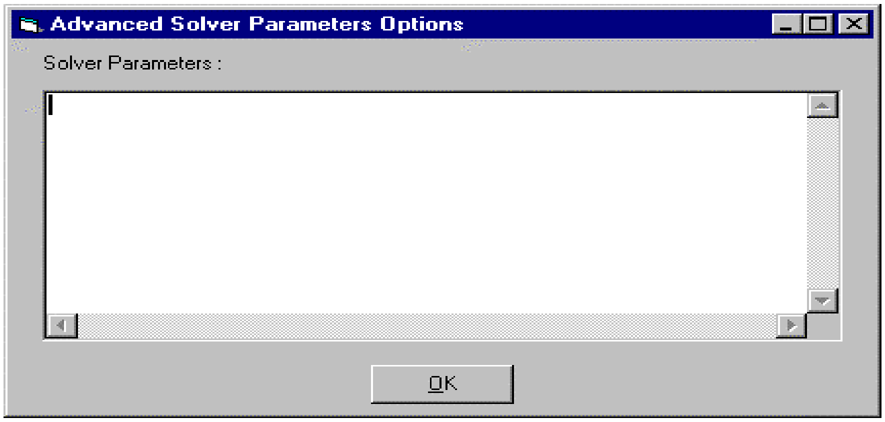


Figure 17-A. Advanced Parameters Window

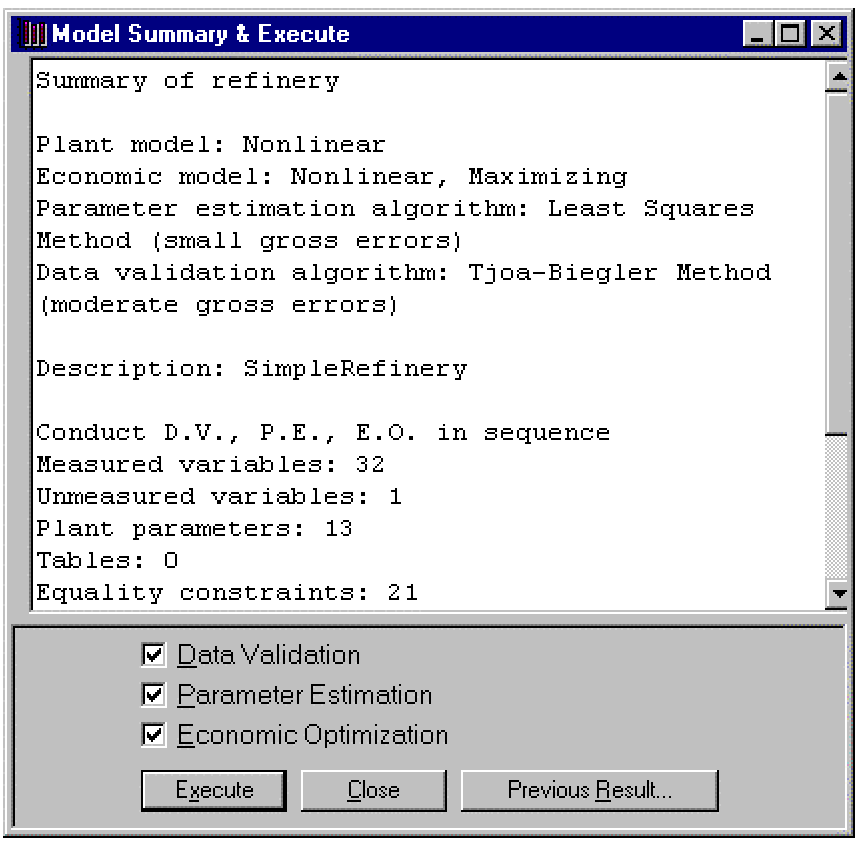


Figure 18 Model Summary and Execution Window

The information required for conducting on-line optimization is now complete. The next step is the execution that will generate three GAMS programs. These programs use all of the input process information to solve the three optimization problems. After clicking the 'Execute' icon in the toolbar of the main window, the system will open the Model Summary and Execute window as shown in Figure 18. These windows provide general information of the optimization model that was previously entered by the user. The 'Execute' button in this window must also be clicked to begin the actual execution of the GAMS program. For additional information on this procedure, refer to the Execution Section of the manual.

A-5. Drawing a Flow Sheet Diagram

Users can draw a plant flowsheet diagram by using the drawing tools provided in Flowsheet Diagram window as shown in Figure 19. Three icons are used to represent process units, streams, and the environment. These icons are located in the toolbar and include rectangles, lines, and the symbol AI/O@ respectively. The icon with the 'Select' shape in the toolbar is used to select the identity for editing. When creating a new model, the user must first enter the name and type of model before drawing the flowsheet diagram.

To develop a process flow diagram, units, streams and I/O units are drawn with the drawing tools. In this diagram, each rectangle represents a process unit. To draw a unit, click on the rectangle icon with the left mouse button, point the mouse to a position where a unit is to be drawn on the form, and drag the mouse to make the desired size of the rectangle. As soon as the user draws a rectangle, a data form is displayed for the user to enter a Unit ID, e.g. U1. Variables and plant parameters are assigned in this dialogue box. Then the user clicks the 'Refresh' button in the data form, and the variable, parameter, and associated information are displayed. If the user clicks 'OK', the unit is drawn with the given Unit ID. Double clicking on top of the unit will open the 'Edit text mode', where the Unit ID can be edited.

Each line represents a stream in the process diagram. To draw a stream, click on the line icon with the left mouse button, point the mouse on top of the unit where the stream will begin, then drag the mouse to the top of the unit where the stream will end. A stream line can only be drawn between two units or between a unit and an environment unit. As soon as the user draws a stream, a data form is displayed for the user to enter the stream number, e.g. S1. When the user clicks the 'Refresh' button in the data form, the variables and other information corresponding to this stream number is displayed in the input window. If the user clicks AOK@, the stream number is drawn with the given name.

When a stream flows from the environment to the system (a unit) or flows from a unit to the environment, the stream must be drawn between the unit and that environment unit ("I/O"). To draw a unit representing the environment, click on the AI/O@ icon with the left mouse button, point the mouse to a position where the environment unit is to be inserted, and drag the mouse to

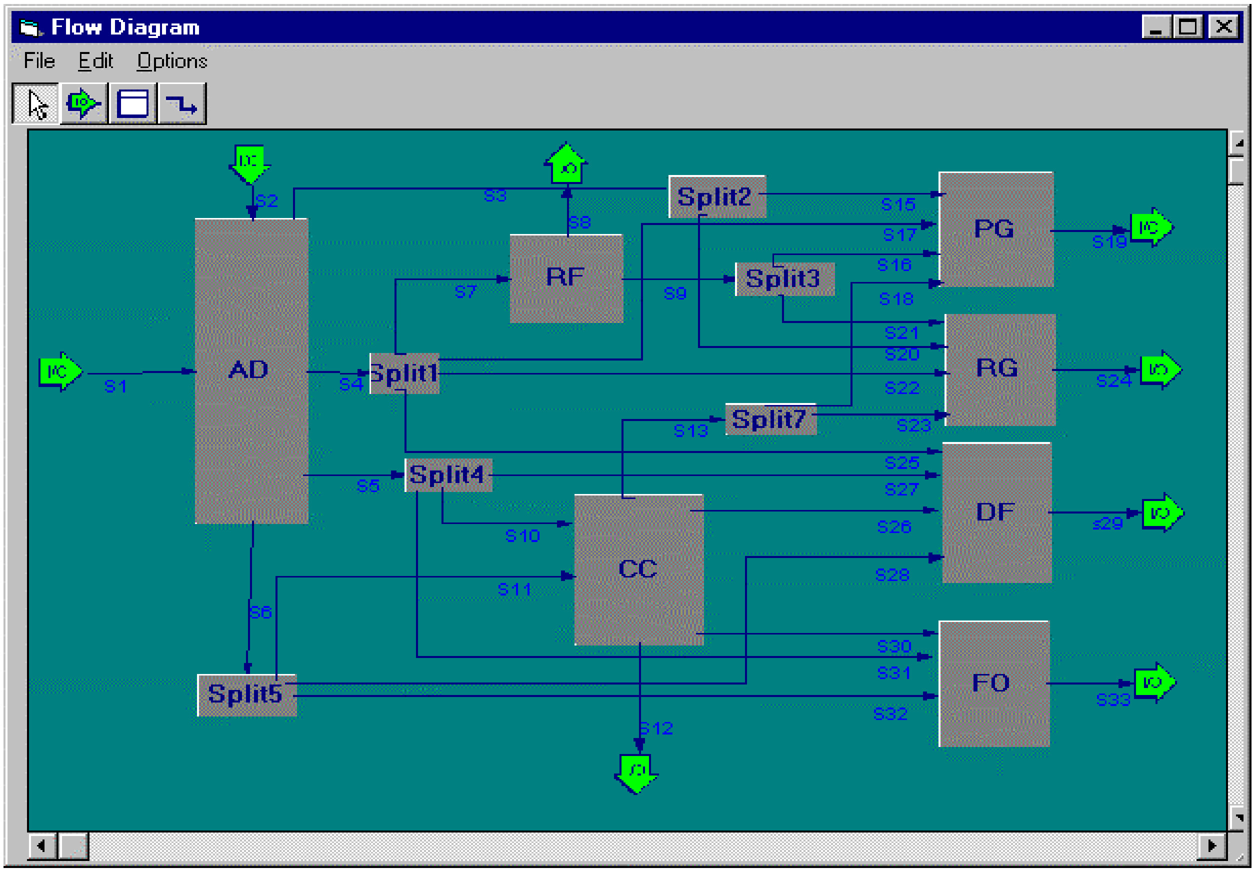


Figure 19 Flow Sheet Diagram for Simple Refinery

the mouse to the desired size. In the flowsheet diagram, multiple environment units can be drawn to connect with the beginning streams or ending streams in a process.

<u>Selecting</u>

- Click on the select icon
- Click on the stream, unit or the environment unit which needs to be selected
- The selected stream or unit is indicated by a red border

<u>Resizing a Shape</u>

- Move the mouse on top of the shape
- Select the required shape on the diagram
- When the cursor changes to the resize cursor (double sided arrow or a cross) drag using the mouse by holding down the left mouse button

<u>To move the shapes</u>

- Select the required shape
- Click on the unit and drag while holding down the left button
- Lines only move when the unit they are associated with moves

Changing the properties of shapes (applicable to stream and units)

- Clicking on the right mouse button when pointing on a shape displays the property window of that shape as shown in Figure 20.
- The user can change the text and background color of the units, the line style and text color of the streams and the environment units. The color changes are applied only to the selected stream or unit. A line can be a 3-point line, 4-point line, a 90 degree line, or a straight line

File Menu for Flowsheet Diagram Window

- The File menu contains Print, and Close options

Print - This option is used to print the flowsheet diagram.
Close - This option is used to exit from this screen.

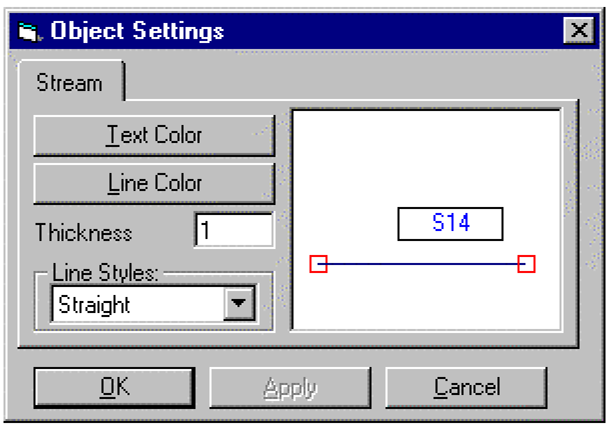


Figure 20 Property Window

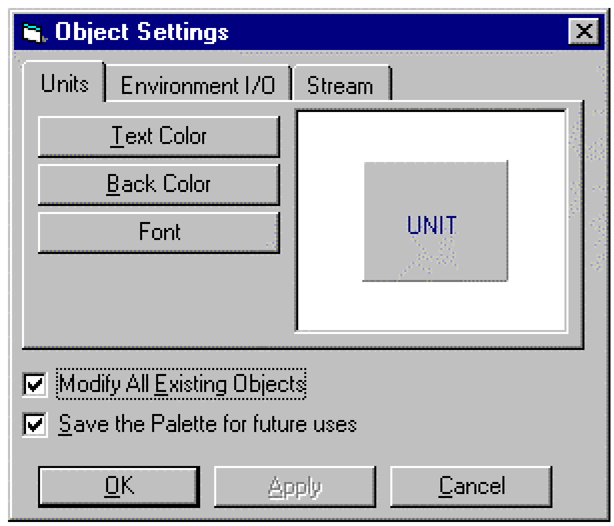


Figure21 Default Class Window

Edit Menu

- The edit menu contains Cut, Copy, Paste, Delete, Data, and Property.
- Clicking the right mouse button can also activate the edit menu.

Cut - This removes the selected unit or line from the screen and copies it onto the clipboard.
Copy -This copies the selected unit or line onto the clipboard.
Paste -This option is not enabled until a unit or line is placed in the clipboard. It copies the contents of the clipboard onto the screen.

Delete - This option deletes the selected unit or stream.

Data -This option displays the information corresponding to the selected stream or unit. When a stream is selected, the corresponding measured and unmeasured variables are displayed. When a process unit is selected, the corresponding measured and unmeasured variables and plant parameters are displayed.

Property -This option invokes the property window for the selected shape as shown in Figure 20. The changes made to the flow sheet diagram are saved under the same name of the model with an extension IDO when the database of the model is saved in the input window. To save the updated flowsheet diagram permanently, the user must save the model in the input window.

Option Menu

- The option menu in Figure 19 has four options: Lock, Zoom, Grid, and Default classes.

Lock – This option allows you to prevent from modification of the flowsheet diagram. When this option is off ,it allows you modify the flowsheet diagram.

Zoom - This option has three sub-options: Zoom in, Zoom out, and Fit to page. Zoom in reduces the size of all the streams and the units and displays the whole diagram. Zoom out increases the size of all the streams and the units and displays the whole diagram. The Fit to page option resizes all the streams and units so the diagram occupies the full page.

Default classes - The three types of classes provided are the streams, the units, and the environment. This option can be used to change the properties of all the streams or of all the units. The changes can be applied to all the existing objects belonging to that class and may be applied to all future drawings as shown in Figure 21.

Grid - This option, when clicked, displays the property window for the grid. The background color and the grid color may be changed. The user can remove the check mark from the option “Display Grid Lines” if he/she does not wish the grid lines to be displayed. Also, the “Snap to Grid” option when checked, draws the streams and units along the grid lines. If the user does not want this property, they can remove the check mark from the check box as shown in Figure 22.

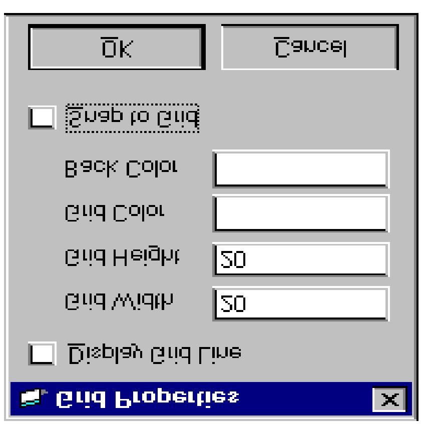


Figure 22 Grid Property

B. Execution of On-Line Optimization

A description of the development of the on-line optimization model for a process was given in the previous sections. The user can change the optimization objective by checking the three check boxes provided in the 'Model Summary and Execute' window in Figure 18. This provides the user with an option of choosing the objective which need not be the same as the one he/she has already chosen in the 'Model Description' window in Figure 8. Checking the checkboxes instructs the system to generate the GAMS program for the corresponding optimization problem and to execute this GAMS program when the 'Execute' button is clicked.

When the Execute button in the Model Summary window is clicked, the system first extracts the model information from the database and generates a GAMS program based on the input information. Then, it instructs GAMS to execute the program. The progress of the GAMS program execution is shown in a DOS mode window in Figure 23, so the user can follow the progress of the optimization algorithm as it searches for an optimal solution. This window also shows the program =s pre-process information. If an error is encountered, it is reported in this screen as well as in the GAMS output file. If the user does not wish to see the progress of the GAMS execution, they may remove the check from the checkbox 'Running Background' found in the GAMS process option of the Output Format Specification window (shown in Figure 17). This option hides the DOS mode window from view and the program will run in the background.

The DOS window is automatically closed as soon as the execution is over. During the Execution, any errors encountered will be reported and all comments and descriptions about the errors are collected in the three GAMS output files for reference. Errors are detected at various stages in the modeling process. The development of the process and economic models through the interactive On-line Optimization system is straight forward, and only typing errors are expected when entering plant information. However, this will be detected in the compilation stage, which is a proofreading stage for the modeling process.

Errors are spotted as early as possible and are reported in a way understandable to the user. Clear suggestions for how to correct the problem and a presentation of the source of the error in terms of the user's problem will also be reported. As soon as an error is detected, execution processing will stop at the next convenient opportunity and a message box with a brief error description is displayed in the Model Summary and Execute window. Therefore, a model will never be solved after an error has been detected. The only remedy is to fix the error and repeat the execution. Errors are grouped according to the three phases of GAMS modeling. They include errors due to compilation, execution, or errors due to model generation (which includes

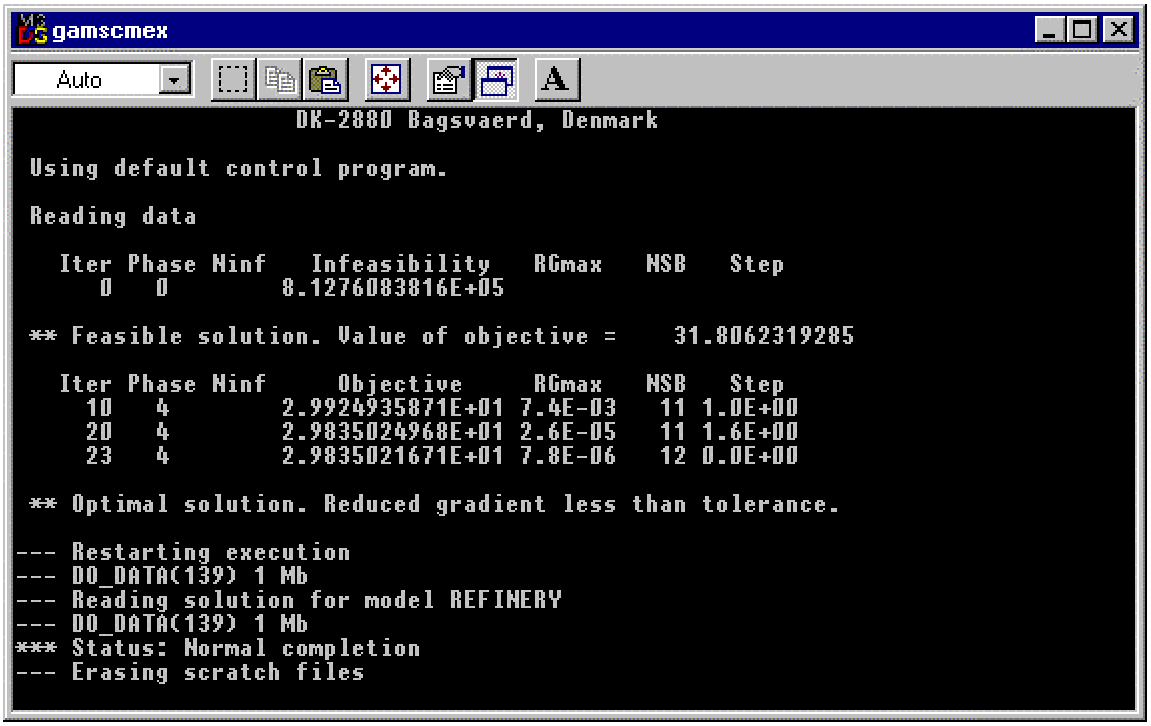


Figure 23 Progress of GAMS Execution

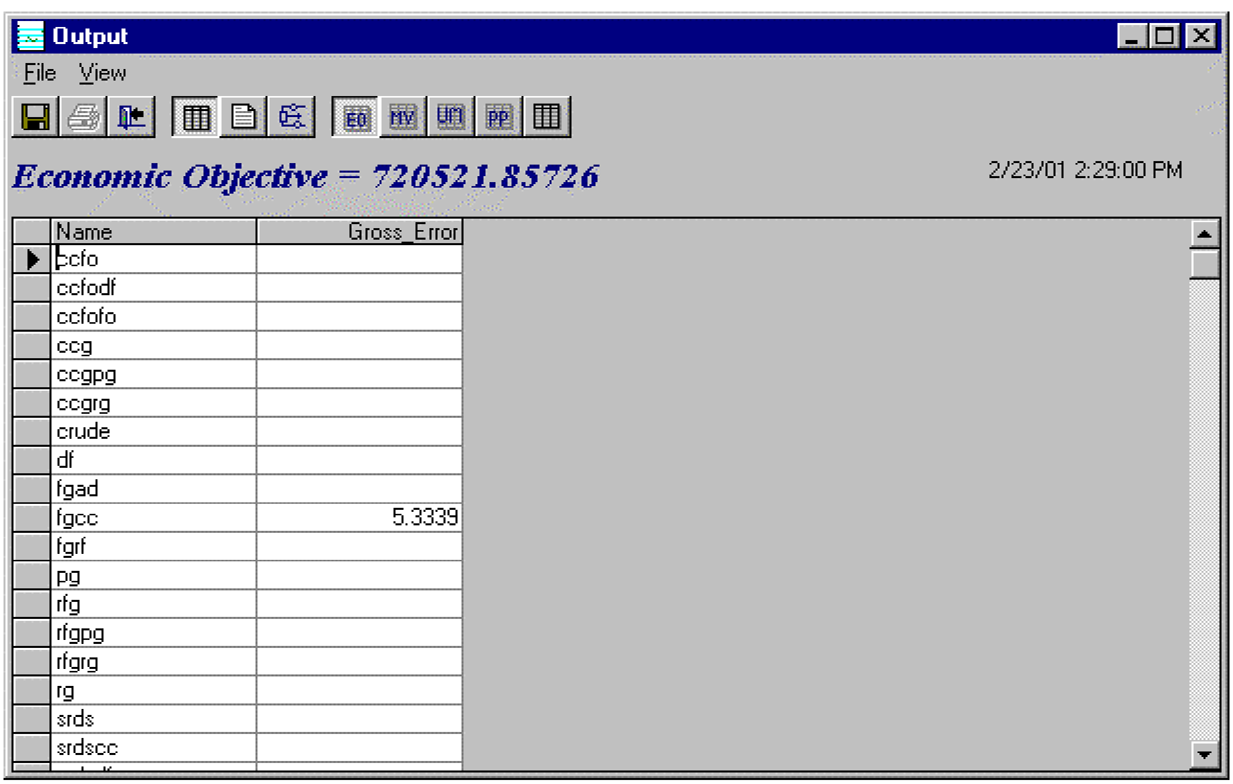


Figure 24 Window of Final Report

the solution that follows). Each of these categories will be discussed in the 'Error Reporting' section of the manual.

When the execution of the program is completed, it displays the output in the Output screen for the user to view or to save the On-line Optimization results. This will be discussed in the next section, Solution Summary.

C. Solution Summary

After the three programs have been executed, three detailed GAMS output files are generated by GAMS for the three optimization programs. These files give detailed solutions for the optimization programming problems for Data Validation, Parameter Estimation, and Economic Optimization. These files can be viewed and saved as text files for future reference. A Final Report is also generated by the interactive On-line Optimization system, which is stored in a database file containing the input information of the plant and economic models. In the Final Report, the estimated values of parameters, the reconciled values of process variables, the optimal set points, and profit from economic optimization are provided.

To view the final report, the user may choose 'Final Report' from the view menu of the Output Window or click 'Final Report' in the toolbar as shown in Figure 24. The user may also choose 'Full Output file' from the view menu or click the 'Full Output' button in the toolbar to view the three detailed GAMS output files. The 'Export' option in the 'File' menu can be used to save the Full Output files as text files and the Final Report as Excel files.

When the final report button in the toolbar is clicked, five options are available for viewing. These five options include: 'Economic Objective', 'Measured Variables', 'Plant Parameters', 'Unmeasured Variables', and the 'Stream Number'. Each option will display its respective information. Recall that the 'Final Report' may be found either in the View menu, or by clicking the 'Final Report' button in the toolbar. Any of the five options may be accessed after choosing the Final Report option. The 'Full Output' menu also contains several options. There are three options for displaying the three output files (Data Validation, Parameter Estimation and Economic Optimization). They can be accessed by the 'View' menu or 'Full Output file' option in the toolbar.

When the user clicks 'Export', a save window is opened to browse the saving location and for specifying the name of the file as shown in Figure 6. The user can export the Full Output files as text files and the Final Report as excel files. Also, the user can print the Full Output files using the 'Print' option in the 'File' menu. The Print option is disabled for the Final Report. After the user has viewed the output, he/she can click the 'Close' option in the 'File' menu. This closes the Output window, but leaves the application open. The user may, therefore, run the model several times by changing the objective in the 'Model Summary and Execute' window. The user can then make changes to the different input windows, save the model, and then execute the model again if he/she wishes to do so. The user can exit the system by clicking 'Exit'. If the 'Exit' button is clicked, it displays a message asking the user if he/she wants to save the current model. If the user clicks 'Yes', the model is saved and then the application is terminated.

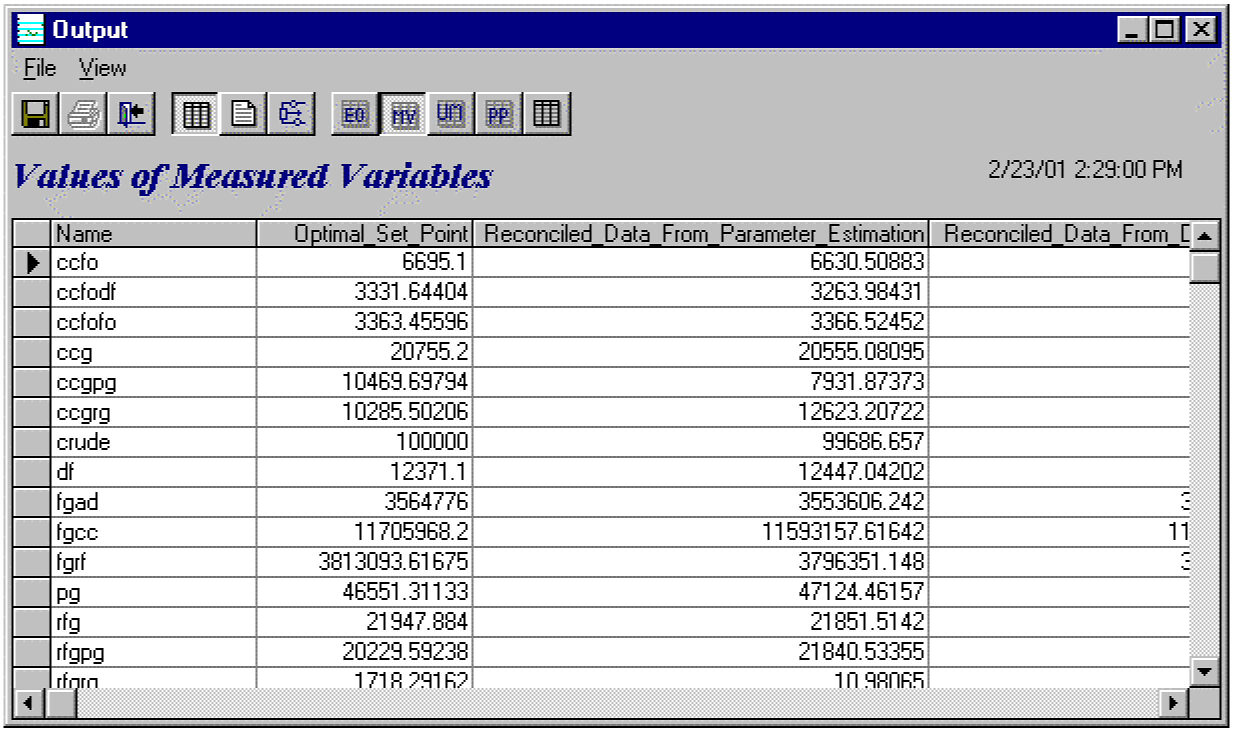
The user can also view stream and unit information through the flow sheet diagram. The flowsheet diagram can be opened by clicking on the Flowsheet option in the 'View' menu. This procedure will be discussed in section C-2.

C-1. Final Report

After the execution of the optimization programs, a final report file is generated by the interactive On-line Optimization system. The Final Report extracts important information from the solutions of three optimization problems, and it gives the main results of On-line Optimization for a process. It is convenient to use this report to view the optimal result in a concise form. The optimal result from Data Validation, Parameter Estimation, and Economic Optimization is summarized in the Final Report. It includes the gross errors detected, the estimated parameter values, the reconciled process data, the optimal operation set points, as well as optimal economic objective value.

By clicking on 'Final Report' in the 'View' menu, the user can choose to view the measured variables, unmeasured variables, economic objective, plant parameters, or the information based on the stream number. In Figure 24, the Final Report gives the optimal economic objective. By clicking on the menu option, 'Measured Variables', the system will open a spreadsheet data form, which includes the optimal set points from economic optimization, reconciled data from Parameter Estimation, reconciled data from Data Validation, and current data sampled from the distributed control system, as shown in Figure 25. By clicking on 'Plant Parameters' in the Final Report menu, the system opens a spreadsheet data form that includes the estimated values of plant parameters as shown in Figure 26. By clicking on the 'Unmeasured Variables', the system opens a spreadsheet data form which includes the information about each unmeasured variables as shown in Figure 27.

The 'Final Report' menu also provides an option of displaying all the variables, which have the same stream number. When the user clicks on the 'Stream Number' button in the toolbar or the 'Stream number' option in the 'Final Report' menu, three buttons are displayed. The buttons are enabled or disabled depending on the objective chosen in the 'Model Summary and Execute' window. For example, if 'Data Validation' is checked and the other two are not checked, then the 'Data Validation' button is enabled and the other two buttons for 'Parameter Estimation' and 'Economic Optimization' are disabled. The user cannot click on the two disabled buttons. As soon as the user clicks on any of these three buttons, an input box appears for the user to enter the stream number. Then the corresponding measured and unmeasured variables with their values and units are displayed as shown in Figure 28.



Output

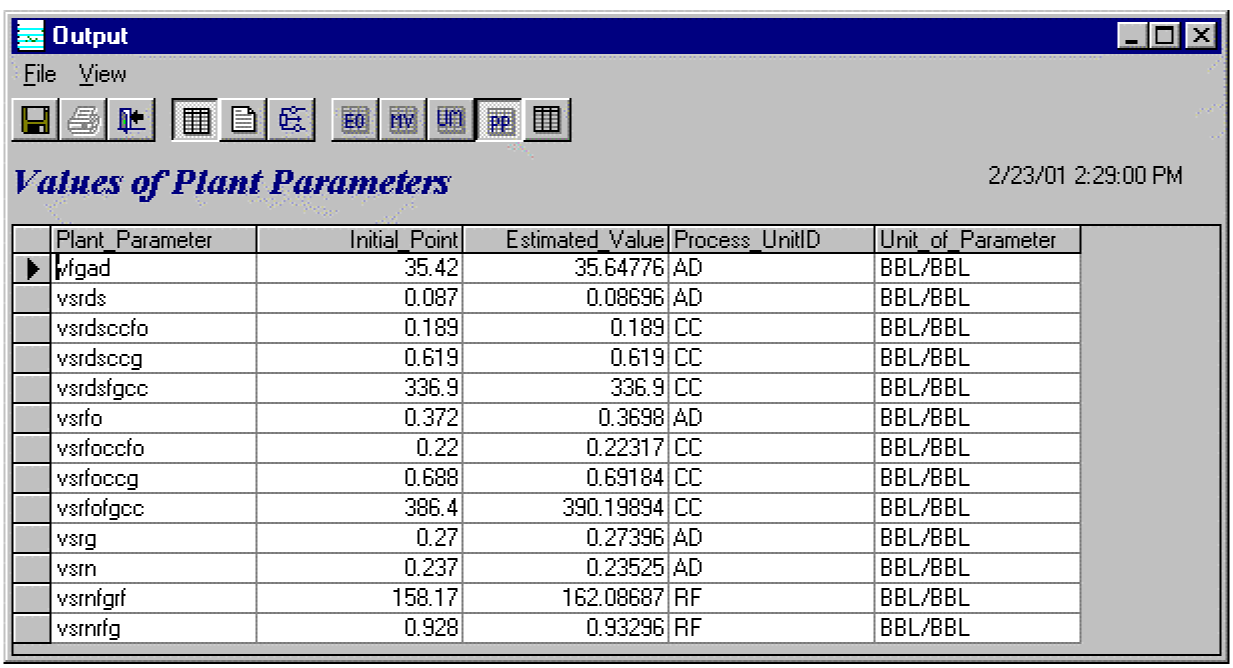
File View

Values of Measured Variables

2/23/01 2:29:00 PM

Name	Optimal_Set_Point	Reconciled_Data_From_Parameter_Estimation	Reconciled_Data_From_[
ccfo	6695.1	6630.50883	
ccfodf	3331.64404	3263.98431	
ccfofo	3363.45596	3366.52452	
ccg	20755.2	20555.08095	
ccgpg	10469.69794	7931.87373	
ccgrg	10285.50206	12623.20722	
crude	100000	99686.657	
df	12371.1	12447.04202	
fgad	3564776	3553606.242	3
fgcc	11705968.2	11593157.61642	11
fgrf	3813093.61675	3796351.148	3
pg	46551.31133	47124.46157	
rfg	21947.884	21851.5142	
rfgpg	20229.59238	21840.53355	
rfgrg	1718.29162	10.98065	

Figure 25 Optimal Set Points and Reconciled Data in Final Report for Measured Variables



Output

File View

Values of Plant Parameters

2/23/01 2:29:00 PM

Plant_Parameter	Initial_Point	Estimated_Value	Process_UnitID	Unit_of_Parameter
vfgad	35.42	35.64776	AD	BBL/BBL
vsrds	0.087	0.08696	AD	BBL/BBL
vsrdsccfo	0.189	0.189	CC	BBL/BBL
vsrdsccg	0.619	0.619	CC	BBL/BBL
vsrdsfgcc	336.9	336.9	CC	BBL/BBL
vsrfo	0.372	0.3698	AD	BBL/BBL
vsrfoccfo	0.22	0.22317	CC	BBL/BBL
vsrfoccg	0.688	0.69184	CC	BBL/BBL
vsrfofgcc	386.4	390.19894	CC	BBL/BBL
vsrg	0.27	0.27396	AD	BBL/BBL
vsrn	0.237	0.23525	AD	BBL/BBL
vsrnfgrf	158.17	162.08687	RF	BBL/BBL
vsrnrfg	0.928	0.93296	RF	BBL/BBL

Figure 26 The Estimated Values of Plant Parameters in Final Report

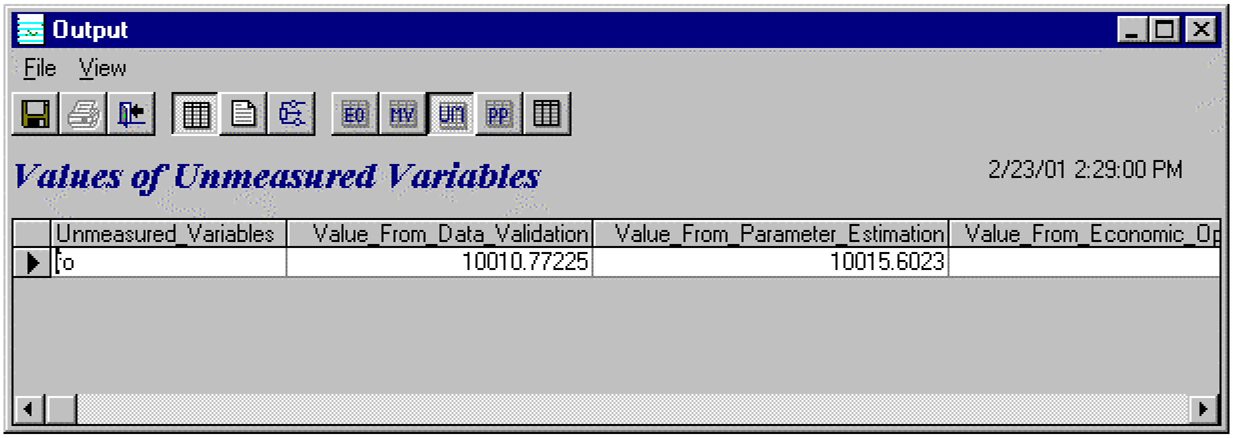


Figure 27 The Reconciled Values for Unmeasured Variables in Final Report

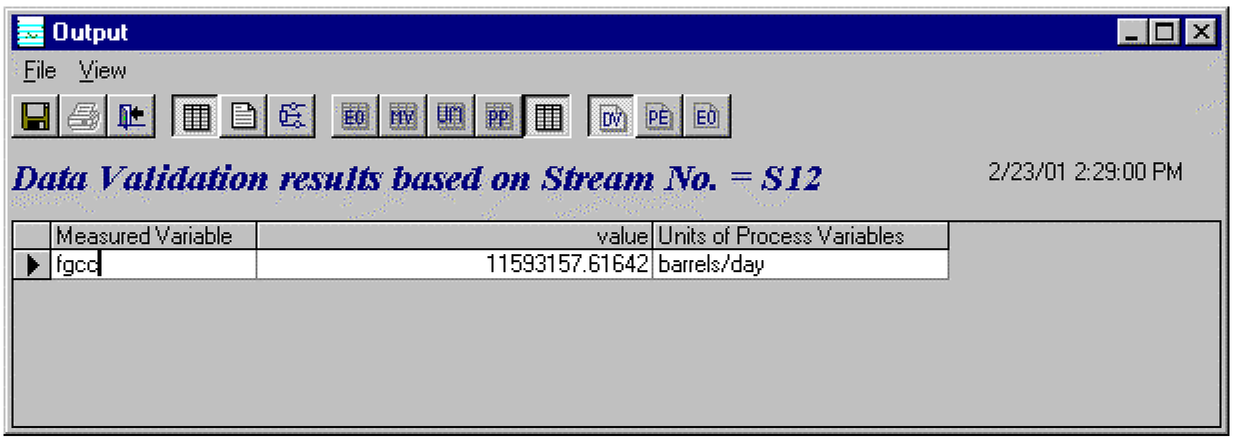


Figure 28 Output Window for Displaying Information based on Stream Number

C-2. Displaying Process Information using Flow Sheet Diagram

Flow sheet diagrams can be used to display information based on stream numbers and Process Unit ID. Flow sheet diagrams can be drawn as explained in the section 'Drawing a Flow Sheet Diagram'. It is stored as a part of the database. Information corresponding to the units or streams can be displayed by clicking the right mouse button (invokes the edit menu as a pop-up menu) and choosing the 'Data' option. The information can also be displayed by clicking on the edit menu and choosing the 'Data' option. Clicking on the 'Refresh' button retrieves the corresponding information.

When a stream is selected and the data option is chosen, the corresponding measured variables and unmeasured variables are displayed. When a unit is selected and the data option is chosen, the corresponding plant parameters, measured variables, and unmeasured variables are displayed as shown in Figure 29 for the simple refinery process.

The flowsheet diagram cannot be modified in this screen. The user needs to return to the input screen if he/she wishes to modify the diagram.

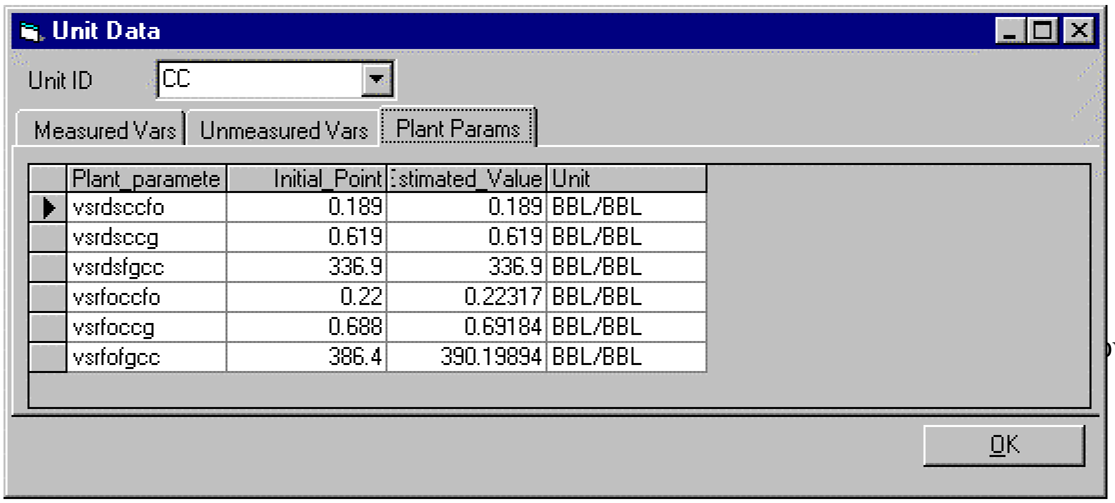


Figure 29 Unit Data Window

C-3. Full Output Files

When the ‘Full Output file’ in the ‘View’ menu is selected or when the ‘Full Output’ button in the toolbar is clicked, the system loads the three detail GAMS output files into a text box as shown in Figure 30. Figure 30 shows the content of the output file for the respective GAMS program whose corresponding button is clicked. By clicking on the ‘Full Output’ button in the toolbar or the ‘Full Output’ option in the ‘View’ menu, three buttons are displayed in the toolbar each corresponding to the three optimization problems. Clicking a button will open the corresponding output file for viewing. These files were generated by GAMS when the three optimization problems were solved.

These detailed output files are very useful in searching for error sources when some errors have occurred and/or when no optimal results are obtained. The detailed output file contains several parts including ‘Compilation Output’, ‘Execution Output’, ‘Output Produced by a Solve Statement’, and ‘Error Reporting’. The default detailed output file includes two parts: echo print of the GAMS source code and the optimal solution report. However, more information about the program can be obtained by changing the default setting of output through the Output File Format Specification window in Figure 17. The detailed output files for the simple refinery are given in Section VII.

Also, each output file can be saved through the ‘Export’ Option in the ‘File’ menu or from the ‘Export’ button in the toolbar as text files with a user specified filename. In addition, all of the output files are saved when the user saves the model in the input window. The three output files are saved with the same name as the database file but with a different extension. The Data Validation output file is saved with a ADVA@ extension, the Parameter Estimation output file is saved with a APES@ extension and the Economic Optimization output file with a AECO@ extension. These files are replaced again if that model is rerun and if the user saves the model. These three files are saved in the same directory as the AIOO@ database file. If the user wishes to keep the output file of the model permanent, he/she can export the output files as text files with different names.

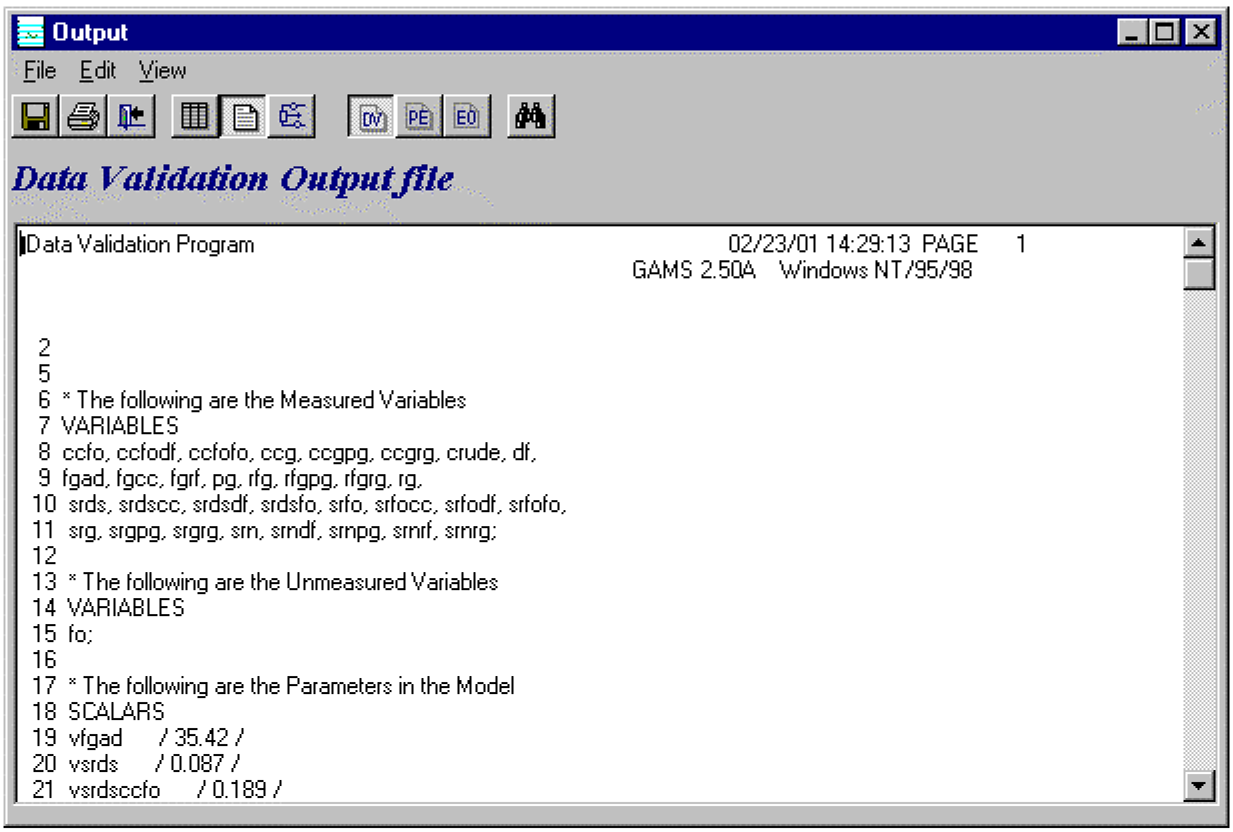


Figure 30 Full Output File of GAMS Programs

Compilation Output: (Brooke, et al., 1996)

The compilation output is produced during the initial check of the program, and it is often referred to as a compilation. It includes two or three parts: the echo print of the program, an explanation of any errors detected, and the symbol reference maps. The echo print of the program is always the first part of the output file. If errors had been detected, the explanatory messages would be found at the end of the echo print. The echo print of the GAMS program for the Economic Optimization of the simple refinery is included in the GAMS output file in Section VII, Examples.

The symbol reference maps follow the echo print, which include the symbol cross reference and the symbol listing map. These are extremely useful if one is looking into a model written by someone else, or if one is trying to make some changes in their own model after spending time away from it. The symbol cross reference lists the identifiers (symbols) in the model in alphabetical order, identifies their type, shows the line numbers where the symbols appear, and classifies each appearance. The complete list of data types is given in Table 2. The symbol list is shown in Section VII for the Economic Optimization program. Next in the listing is a list of references to the symbols, grouped by reference type and identified by the line number in the output file. The actual references can then be found by referring to the echo print of the program, which has line numbers on it. The complete list of reference types is given in Table 3. The symbol reference maps do not appear in the output files by default. However, it can be included in the output files by changing the default setting in Output File Format Specification window in Figure 17.

Table 2 A List of Data Types

Entry in symbol reference table	**GAMS data type**
SET	set
PARAM	parameter
VAR	variable
EQU	equation
MODEL	model

Table 3 A List of Reference Types

Reference	Description
DECLARED	This is where the identifier is declared as to type. This must be the first appearance of the identifier.
DEFINED	This is the line number where an initialization (a table or a data list between slashes) or symbol definition (equation) starts for the symbol.
ASSIGNED	This is when values are replaced because the identifier appears on the left of an assignment statement.
IMPL-ASN	This is an 'implicit assignment': an equation or variable will be updated as a result of being referred to implicitly in a solve statement.
CONTROL	This refers to the use of a set as the driving index in an assignment, equation, loop or other indexed operation (sum, prod, smin or smax).
REF	This is a reference: the symbol has been referenced on the right of an assignment in a display, in an equation, or in a model or solve statement.

Execution Output:

The execution output follows the compilation output and is also found in the GAMS output file. If a display statement is present in the GAMS program, then data requested by the display statement is produced in the execution output while GAMS performs data manipulations. Also, if errors are detected because of illegal data operations, a brief message indicating the cause and the line number of the offending statement, will appear in the execution output. The execution output will be shown in the GAMS output file if a display statement is present in the GAMS program (which requests the display of the value of a variable) or if an execution error is encountered.

Output Produced by a Solve Statement: (Brooke, et al., 1996)

The output triggered by a solve statement includes the equation listing, the column listing, the model statistics, solver report, the solution listing, report summary, and file summary as shown in the GAMS output file for the simple refinery in Table 28 of Section VII. All of the output produced as a result of a solve statement is labeled with a subtitle identifying the model, its type, and the line number of the solve statement. The first list in the output produced by the SOLVE statement is the Equation Listing, which is marked with that subtitle in the output file. The Equation Listing is an extremely useful debugging aid. It shows the variables that appear in each constraint, and what the individual coefficients and right-hand-side value evaluates to after the data manipulations have been made. Normally, the first three equations in every block are listed. Most of the listing is self-explanatory. The name, text, and type of constraints are shown. The four dashes are useful for mechanical searching. All terms that depend on variables are collected on the left, and all the constant terms are combined into one number on the right, with any necessary sign changes made. For example, a equation Ax + 5y - 10z +20 =e= 0" is rearranged as: Ax + 5y - 10z =e= -20". Four places of decimals are shown if necessary, but trailing zeroes following the decimal point are suppressed. E-format is used to prevent small numbers from being displayed as zero. By default, the equation listing will not appear in the output file unless specified by the user in the Output File Format Specification Window in Figure 17.

The general format in the equation listing was described above. However, the nonlinear terms in an equation are treated differently from the linear terms. If the coefficient of a variable in the Equation Listing is enclosed in parentheses, then the variable corresponding to this coefficient is nonlinear in the constraint equation, and the value of the coefficient depends on the activity levels of one or more of the variables. This coefficient is not algebraic, but it is the partial derivative of each variable evaluated at their current level values (initial points).

For an equation: $x + 2y^3 + 10$ =e= 0 with current level values x = 2 and y = 1, this equation is listed in the equation listing as: x + (6) y =e= -12, where the coefficient of y is the partial derivative of the equation with respect to y evaluated at y=1, i.e., $6y^2 = 6$. The right hand side coefficient, -12, is the sum of the constants in the equation, 10, and the constant, 2, from the linearization of the nonlinear term $2y^3$ using Taylor expansion evaluated at y = 1. x in this equation is linear, and its coefficient is shown as 1 without the parentheses.

Next, the column listing gives the individual coefficients sorted by column rather than by row. The default shows the first three entries for each variable, along with their bound and level values. The format for the coefficients is the same as in the equation listing, with the nonlinear ones enclosed in parentheses and the trailing zeroes dropped. The order in which the variables appear is the order in which they were declared.

The final information generated while a model is being prepared for solution is the statistics block to provide details on the size and nonlinearity of the model. The status for the solver (the state of the program) and the model (what the solution looks like) are characterized in solver status and model status. The model status and solver status are listed in Table 4 and Table 5, respectively.

The next section is the solver report, which is the solve summary particular to the solver program that has been used. Also, there will be diagnostic messages in plain language if anything unusual was detected, and specific performance details as well. In case of serious trouble, the GAMS listing file will contain additional messages printed by the solver, which may help identify the cause of the difficulty.

Solution listing is a row-by-row then column-by-column listing of the solutions returned to GAMS by the solver program. Each individual equation and variable is listed with four pieces of information. The four columns associated with each entry are listed in Table 6. For variables, the values in the LOWER and UPPER columns refer to the lower and upper bounds. For equations, they are obtained from the (constant) right-hand-side value and from the relational type of the equation. EPS means very small or close to zero. It is used with non-basic variables whose marginal values are very close to, or actually, zero, or in nonlinear problems with super-basic variables whose marginal values are zero or very close to it. A superbasic variable is the one between its bounds at the final point but not in the basis.

For models that do not reach an optimal solution, some constraints may be marked with the flags shown in Table 7. The final part of solution listing is the report summary marked with four asterisks. It shows the count of rows or columns that have been marked INFES, NOPT, and UNBND. The sum of infeasibilities will be shown if the reported solution is infeasible. The error count is only shown if the problem is nonlinear. The last piece of the output file is the file summary, which gives the names of the input and output disk files. If work files have been used, they will be named here as well.

Table 4 A List of Model Status in GAMS Output Files

Model status	**Meaning**
1. Optimal	This means that the solution is optimal. It only applies to linear problems or relaxed mixed integer problems (RMIP).
2. Locally Optimal	This message means that a local optimal for nonlinear problems, since all that can guarantee for general nonlinear problems is a local optimum.
3. Unbounded	That means that the solution is unbounded. It is reliable if the problem is linear, but occasionally it appears for difficult nonlinear problem that lack some strategically paced bounds to limit the variables to sensible values.
4. Infeasible	This means that he linear problem is infeasible.
5. Locally Infeasible	This message means that no feasible point could be found for the nonlinear problem from the given starting point. It does not necessarily mean that no feasible point exists.
6. Intermediate Infeasible	The current solution is not feasible, the solver program stopped, either because of a limit (iteration or resource), or some sort of difficulty.
7. Intermediate Nonoptimal	This is again an incomplete solution, but it appears to be feasible.
8. Integer Solution	An integer solution has been found to a MIP (mixed integer problem).
9. Intermediate Noninteger	This is an incomplete solution to a MIP. An integer solution has not yet been found.
10. Integer	There is no integer solution to a MIP. This message should be reliable.
11.Error Unknown, Error no Solution	There is no solution in either of these cases.

Error Reporting:

The last part in the output file is error reporting. All the comments and descriptions about errors have been collected into this section for easy reference. Errors are grouped into the three phases of GAMS modeling in the On-line Optimization system: compilation, execution and model generation (which includes the solution that follows). They will be illustrated in the next section, ÀOptimization Solver@.

Table 5 A List of Solver Status in GAMS Output Files

Solver status	**Meaning**
1. Normal Completion	This means that the solver terminated in a normal way: i.e., it was not interrupted by an iteration or resource limit or by internal difficulties. The model status describes the characteristics of the accompanying solution.
2. Iteration Interrupt	This means that the solver was interrupted because it used too many iterations. Use option iterlim to increase the iteration limit if everything seems normal.
3. Resource Interrupt	This means that the solver was interrupted because it used too much time. Use option reslim to increase the time limit if everything seems normal.
4. Terminated by Solver	This means that the solver encountered difficulty and was unable to continue. More detail will appear following the message.
5. Evaluation Error Limit	Too many evaluations of nonlinear terms at undefined values. You should use bounds to prevent forbidden operations, such as division by zero. The rows in which the errors occur are listed just before the solution.
6. Unknown Error Preprocessor(s) Error Setup Failure Error Solver Failure Error Internal Solver Error Error Post-Processor	All these messages announce some sort of unanticipated failure of GAMS, a solver, or between the two. Check the output thoroughly for hints as to what might have gone wrong.

Table 6 A List of Solution Listing Types

Heading in listing file	Description
LOWER	Lower Bound (.lo)
LEVEL	Level Value (.l)
UPPER	Upper Bound (.up)
MARGINAL	Marginal (.m)

Table 7 A List of Constraint Flags

Flag	Description
INFES	The row or column is infeasible. This mark is make for any entry whose LEVEL value is not between the UPPER and LOWER bounds.
NOPT	The row or column is non-optimal. This mark is made for any non-basic entries for which the marginal sign is incorrect, or superbasic ones for which the marginal value is too large.
UNBND	The row or column that appears to cause the problem to be unbounded.

VI. OPTIMIZATION SOLVER - GAMS (Brooke et al., 1996)

The basic components of a GAMS input model include:

- Sets
- Data (Parameters, Tables, Scalar)
- Variables
- Assignment of bounds and/or initial values
- Equations
- Model and Solve statements
- Display/Put statement

The overall content of GAMS output file is:

- Echo Print
- Reference Maps
- Equation Listings
- Status Reports
- Results

A. Format for Entering System Information

The GAMS input code generated by the interactive On-line Optimization system is based on the information provided by the user. Although the user usually does not need to consider the format of the GAMS program, there are some regulations about the format related to GAMS that must be followed to properly enter information about the plant. The input must be in correct format for an accurate GAMS input file to be generated automatically by the On-line Optimization system.

Most of the characters and words are allowable for the input information, however, the letters in the input information are case insensitive. A few characters are not allowed for the input because they are illegal or ambiguous on some machines. Generally, all unprintable and control characters are illegal. Most of the uncommon punctuation characters are not part of the language, but can be used freely. In Table 8, a full list of legal characters is given.

Table 8 A List of Full Set of Legal Characters for GAMS

A to Z	alphabet	a to z	alphabet	0 to 9	Numerals
&	ampersand	A @	double quote	#	pound sign
*	asterisk	=	equals	?	question mark
@	at	>	greater than	;	semicolon
\	back slash	<	less than	›	single quote
:	colon	-	minus	/	slash
,	comma	( )	parenthesis		space
$	dollar	[ ]	square brackets	_	underscore
.	dot	{ }	braces	!	exclamation mark
+	plus	%	percent	^	circumflex

Table 9 A List of All Reserved Words for GAMS

abort	ge	not	smin	if
acronym	gt	option	sos1	then
acronyms	inf	options	sos2	else
alias	integer	or	sum	semicont
all	le	ord	system	semiint
and	loop	parameter	table	file
assign	lt	parameters	using	files
binary	maximizing	positive	variable	putpage
card	minimizing	prod	variables	puttl
display	model	scalar	xor	free
eps	models	scalars	yes	no
eq	na	set	repeat	solve
equation	ne	sets	until	for
equations	negative	smax	while	

Table 10 A List of Non-alphanumeric Symbols for GAMS

=l=	--
=g=	++
=e=	**
=n=	

Besides characters, there are some reserved words and non-alphanumeric symbols with predefined meanings in GAMS which cannot be used as input information. The reserved words and non-alphanumeric symbols are listed in Table 9 and Table 10 respectively.

In the On-line Optimization system, numeric values are entered in a style similar to that used in other computer languages. Blanks cannot be used in a number because the system treats a blank as a separator. The common distinction between real and integer data types does not exist. If a number is entered without a decimal point, it is still stored as a real number. In addition, the system uses an extended range arithmetic that contains special symbols for infinity (INF), negative infinity (-INF), undefined (UNDF), epsilon (EPS), and not available (NA) as shown in Table 11. One cannot enter UNDF; it is only produced by an operation that does not have a proper result, such as division by zero. All other special symbols can be entered and used as if they were ordinary numbers.

GAMS uses a small range of numbers to ensure that the system will behave in the same way on a wide variety of machines. A general rule is to avoid using or creating numbers with absolute values greater than 1.0e+20. A number up to 10 significant digits can be entered on all machines, and some machines can even support more than that. However, if a number is too large, it may be treated by the system as undefined (UNDF), and all values derived from it in a model may be unusable. It is recommended to always use INF (or -INF) explicitly for arbitrarily large numbers. When an attempted arithmetic operation is illegal or has undefined results because of the value of arguments (division by zero is the normal example), an error is reported and the result is set to undefined (UNDF). Afterwards, UNDF is treated as a proper data value and does not trigger any additional error messages. Thus, the system will not solve a model if an error has been detected, but it will terminate with an error condition.

The string definition such as the variable's name in the system has to start with a letter followed by more letters or digits. It can only contain alphanumeric characters and up to 10 characters long. The comment to describe the set or element must not exceed 80 characters. Basically, there are five possible types of variables that may be used which are listed in Table 12.

The type of mathematical programming problem must be known before the problem is solved. The On-line Optimization system can only solve linear and nonlinear optimization problems. However, GAMS can solve a large number of optimization problems which are summarized in Table 13.

As the interactive On-line Optimization system writes all the required GAMS input files for the user, most of the components in the GAMS input model are automatically formulated from the information provided in the input windows. If the user can follow the explicit rules introduced above, the GAMS input file can be generated automatically. After the user enters all the plant information through the input windows, the GAMS source codes will be generated. The On-line Optimization system will then forward these source codes to the GAMS software. This initiates the execution of GAMS and also creates output files so the user can view the execution in the output window. The execution and the output were discussed in the previous sections.

Table 11 A List of Special Symbols for GAMS

Special symbol	Description
INF	Plus infinity. A very large positive number
-INF	Minus infinity. A very large negative number
NA	Not available. Used for missing data. Any operation that uses the value NA will produce the result NA
UNDF	Undefined. The result of an undefined or illegal operation. The user cannot directly set a value to UNDF
EPS	Very close to zero, but different from zero.

Table 12 A List of Types of Variables for GAMS

Keyword	Default Lower Bound	Default Upper Bound	Description
free (default)	-inf	+inf	No bounds on variables. Both bounds can be changed from the default values by the user
Positive	0	+inf	No negative values are allowed for variables. The upper bound can be changed from the default value by the user
Negative	-inf	0	No positive values are allowed for variables. The lower bound can be changed from the default value by the user.
binary	0	1	Discrete variable that can only take values of 0 or 1
Integer	0	100	Discrete variable that can only take integer values between the bounds. Bounds can be changed from the default value by the user

Table 13 A List of Types of Models for GAMS

Model Type	Description
LP	Linear programming. No nonlinear terms or discrete (binary or integer) variables.
NLP	Nonlinear programming. There are general nonlinear terms involving only Asmooth@ functions in the model, but no discrete variables.
DNLP	Nonlinear programming with discontinuous derivatives. Same as NLP, but Anon-smooth@ functions can appear as well. More difficult to solve than NLP. Not recommended to use.
RMIP	Relaxed mixed integer programming. Can contain discrete variables but the integer and binary variables can be any values between their bounds.
MIP	Mixed integer programming. Like RMIP but the discrete requirements are enforced: the discrete variables must assume integer values between their bounds.
RMINLP	Relaxed mixed integer nonlinear programming. Can contain both discrete variables and general nonlinear terms. The discrete requirements are relaxed. Same difficulty as NLP.
MINLP	Mixed integer nonlinear programming. Characteristics are the same as for RMINLP, but the discrete requirements are enforced.
MCP	Mixed Complementarily Problem
CNS	Constrained Nonlinear System

B. Equation Formulation

Aside from the rules introduced above, the equations as the main part of the input information have their own specific requirements. The mathematical definitions of equations can be written in one or multiple lines. Blanks can be inserted to improve readability, and expressions can be arbitrarily complicated. The standard arithmetic operations for the equations are listed in Table 14. The arithmetic operations listed in Table 14 are in order of precedence, which determines the order of evaluation in an equation without parentheses.

The relational operators in the equations are:

- =L= Less than: left hand side (lhs) must be less than or equal to right hand side (rhs)
- =G= Greater than: lhs must be greater than or equal to rhs
- =E= Equality: lhs must equal to rhs
- =N= No relationships enforced between lhs and rhs. This equation type is rarely used.

Table 14 A List of Standard Arithmetic Operators

Operator	Description
**	exponentiation
*, /	multiplication and division
+, -	addition and subtraction (unary and binary)

Table 15 A List of Numerical Relationship Operators

Operator	Description
lt, <	strictly less than
le, <=	less than or equal to
eq, =	equal to
ne, <>	not equal to
ge, >=	greater than or equal to
gt, >	strictly greater than

Table 16 A List of Logical Operators

Operator	Description
not	Not
and	And
or	Inclusive or
xor	Exclusive or

Table 17 The Truth Table Generated by the Logical Operators

Operands		Results			
a	b	a and b	a or b	a x or b	not a
0	0	0	0	0	1
0	non-zero	0	1	1	1
non-zero	0	0	1	1	0
non-zero	non-zero	1	1	0	0

Table 18 The Operator Procedure Order in case of Mixed Logical Conditions

Operation	Operator
Exponentiation	**
Numerical Operators	
Multiplication, Division	*, /
Unary operators - Plus, Minus	+, -
Binary operators-Addition, Subtraction	+, -
Numerical Relationship Operators	<, <=, =, <>, >=, >
Logical Operators	
Not	not
And	and
Or, xor	or, xor

Additionally, GAMS provides the numerical relationships and logical operators used to generate logical conditions for evaluating values of True or False. A result of zero is treated as a logical value of False, while a non-zero result is treated as a logical value of True. A complete list of numerical relationship operators and logical operators are found in Table 15 and Table 16.

The functions of the logical operators are expressed in Table 17. For the mixed logical conditions, the default operator precedence order used by GAMS in the absence of parenthesis is shown in Table 18 in decreasing order. For the formulation of equations, variables can appear on the left or right-hand side of an equation or on both sides. The system can automatically convert the equation to its standard form (variables on the left, no duplicate appearances) before calling the GAMS solver. For the convenience of input, the system also provides several special notations, such as summation (sum), product (prod), minimum value (smin), and maximum value (smax).

C. Functions Predefined in the System

There are two types of functions based on the type of argument: exogenous or endogenous. For exogenous arguments, the arguments are known, and examples are parameters and variable attributes. The expression is evaluated once when the model is set up. All functions except the random distribution functions, uniform and normal, are allowed. With endogenous arguments, the arguments are variables, and are, therefore, unknown. The function will be evaluated many times at intermediate points while the model is being solved. The occurrence of any function with endogenous arguments implies that the model is not linear and the use of the functions of ‘uniform’ and ‘normal’ are forbidden in an equation definition. Some built-in functions are listed in Table 19.

D. Scaling Option for Variables and Equations

To facilitate the translation between a natural model (no scaling) to a well scaled model, GAMS introduces the concept of a scale factor for variables and equations with a scaling option. This feature is incorporated in the interactive On-line Optimization system to provide a well scaled optimization problem for GAMS to solve. To use the scaling option in the interactive On-line Optimization, the user must highlight the scaling option in the variable declaration and the equations declaration windows. Then, the user must enter the values of the scale factors for the variables and equations that need to be scaled. The following describes how the scale factor is incorporated in the GAMS program and how to determine the value of a scale factor.

The scale factor on a variable V^s is used to relate the variable as seen by user (in natural model) V^u to the variable as seen by the optimization algorithm (in well scaled model) V^a as follows:

$$V^u = V^a \ \ V^s$$

This means that the scaled variable V^a will bearound 1 if the scale factor V^s is chosen to represent the order of magnitude of the user variable V^u.

If the approximate expected value for a variable in the model is known, then the magnitude of this variable value is used as the scale factor of the variable. The scale factor can be specified by users through the Measured or Unmeasured Variables window. If the approximate expected values for some of the variables in the model are not available, these values can be found in the column list of the corresponding GAMS output file. The scale factor will not change the values of variables in the solution seen by users. GAMS uses the scale factor to scale variables and transfer the model into a well scaled model for optimization algorithm.

When the optimal solution is found, GAMS will rescale the variables and transfer them back to user's notation. The effect of scaling can only be viewed in the Column and Equation lists of the GAMS output files.

The scale factor for an equation is dependent on the order of magnitude of the equation coefficients. It is slightly different from the determination of scale factor for a variable that is dependent on the magnitude of the variable. An equation usually contains several terms, and it has several coefficients that may not be in the same order. If the equation is linear, the coefficients of this equation is known. If the equation is nonlinear, then the equation is linearized first using the initial values. However, the linearized coefficients must be obtained from the equation list. Users can obtain the values of the linearized equation coefficients for nonlinear constraints from the equation list of the corresponding GAMS output file. To appropriately assign the scale factor for an equation, users need to carefully select the value of the scale factor based on the coefficients shown in equation list of the GAMS output file so that all coefficients will be in the range of 0.01 to 100 after scaling.

The column (variables) and equation lists are very important for nonlinear problems when scaling the variables and equations. It provides initial values of all variables and linearized constraint coefficients, which can be used to determine the scale factors for both variables and equations. It is suggested that the user turn off the scaling option for both variables and equations before GAMS be initiated. After the program ends, if the solution is correct and there was no difficulty in searching for an optimal solution, then the scaling option is not necessary. If the solution is not correct or some difficulty was encountered while searching for an optimal solution, then the scaling option must be incorporated in the program. In this case, users may instruct the system to include the column and equation lists in the output file. To do this, the user must change the default setting for the output files in window 12, the Output File Format Specification window. This will run the optimization program without the scaling option. Based on the values of variables in column list without scaling, users can decide the values of scale factors for variables, enter them in the Measured Variables and Unmeasured variables windows, and highlight the icon AInclude Scaling Option for variables@ to scale the variables first. After the system executes the program, a new equation list which incorporates the scale information of variables is generated and can be used for equation scaling. Based on the linearized coefficients in this new equation list, users can determine the scale factors for the equations and enter them in the Equality Constraints and Inequality Constraints windows. Also, users must highlight the icon AInclude Scaling Option for Equations@ to add the Scaling Option in the programs.

Table 19 A List of Functions Predefined in the On-line Optimization System

Function	Description	Classification	Exogenous Classification	Endogenous model type
abs	Absolute value	Non-smooth	Legal	DNLP
arctan	Arctangent	Smooth	Legal	NLP
ceil	Ceiling	Smooth	Legal	Illegal
cos	Cosine	Discontinuous	Legal	NLP
errorf	Error function	Smooth	Legal	NLP
exp	Exponential	Smooth	Legal	NLP
floor	Floor	Discontinuous	Legal	Illegal
log	Natural log	Smooth	Legal	NLP
log10	Common log	Smooth	Legal	NLP
mapval	Mapping function	Discontinuous	Legal	Illegal
max	Largest value	Non-smooth	Legal	DNLP
min	Smallest value	Non-smooth	Legal	DNLP
mod	Remainder	Discontinuous	Legal	Illegal
normal	Normal random	Illegal	Illegal	Illegal
power	Integer power	Smooth	Legal	NLP
round	Rounding	Discontinuous	Legal	Illegal
sign	Sign	Discontinuous	Legal	Illegal
sin	Sine	Smooth	Legal	NLP
sqr	Square	Smooth	Legal	NLP
sqrt	Square root	Smooth	Legal	NLP
trunc	Truncation	Discontinuous	Legal	Illegal
uniform	Uniform random	Illegal	Illegal	Illegal

E. Error Reporting

During compiling, executing, and solving the optimization problem, GAMS checks the input source code for program syntax, rearranges the information in the source code, and solves the optimization problem. At every step, GAMS records any error encountered and reports it in the GAMS output file. The following describes error reporting during solving the optimization problems.

Compilation Errors: The first type of error is a compilation error. When the GAMS compiler encounters an error in the input file, it inserts a coded error message inside the echo print on the line immediately following the scene of the offense. The message includes a $-symbol and an error number printed below the offending symbol (usually to the right). This error number is printed on a separate line starting with four asterisks (****). If more than one error occurs on a line, the $-signs may be suppressed and the error number is squeezed. GAMS programs are generated by the system, and no serious compilation errors are expected to appear. The most common error will be a spelling error, i.e., the variables defined in the equations may be mistyped and mismatched while declaring the variables. This will result in 'variable undefined error'. GAMS will not list more than 10 errors on any single line. At the end of the echo print, a list of all error numbers encountered, together with a description of the probable cause of each error, will be printed. The error messages are self-explanatory and will not be listed here. Checking the first error is recommended because it has the highest priority.

Execution Errors: The second type of error is an execution error. Execution errors are usually caused by illegal arithmetic operations such as division by zero or taking the log of a negative number. GAMS prints a message on the output file with the line number of the offending statement and continues execution. A GAMS program should never abort with an unintelligible message from the computer=s operating system if an invalid operation is attempted. GAMS has rigorously defined an extended algebra that contains all operations including illegal ones. The model library problem [CRAZY] contains all non-standard operations and should be executed to study its exceptions. GAMS arithmetic is defined over the closed interval [-INF, INF] and contains values EPS (small but not zero), NA (not available), and UNDF (the result of an illegal operation). The results of illegal operations are propagated through the entire system and can be displayed with standard display statements. The model cannot be solved if errors have been detected previously.

Solve Errors: The last type of error is a solve error. The execution of a solve statement can trigger additional errors called MATRIX errors, which report on problems encountered during transformation of the model into a format required by the solver. Problems are most often caused by illegal or inconsistent bounds, or an extended range value being used as a matrix coefficient. Some solve statement require the evaluation of nonlinear functions and the computation of derivatives. Since these calculations are not carried out by the system but by other subsystems not under its direct control, errors associated with these calculations are reported in the solution report.

If the solver returns an intermediate solution because of evaluation errors, then a solution will still be attempted. The only fatal error in the system that can be caused by a solver program is the failure to return any solution at all. If this happens, as mentioned above, all possible information is listed on the GAMS output file, but the solution will not be given.

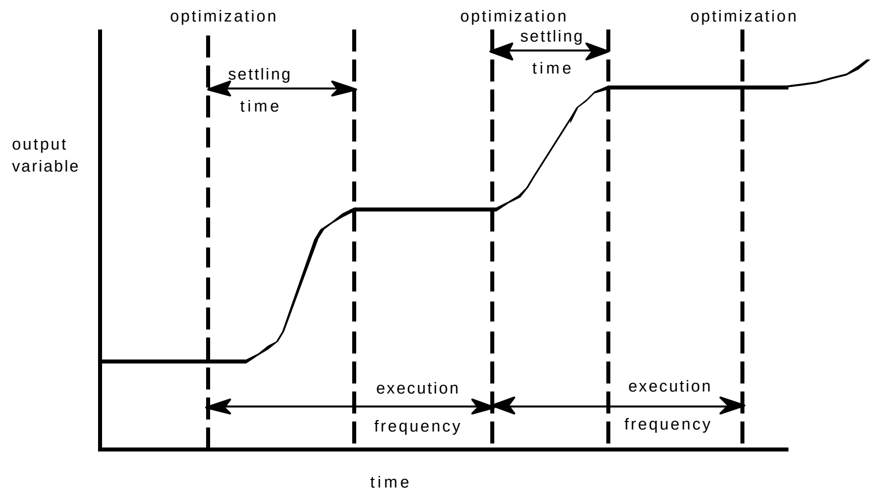
VII. STEADY-STATE DETECTION AND EXECUTION FREQUENCY

The execution frequency for optimization is the time between conducting On-line Optimization of the process, and it has to be determined for each of the units in the process. It depends on the settling time, i.e., the time required for the units in the process to move from one set of steady-state operating conditions to another. This settling time can be estimated from the time constant determined by process step testing. The time period between two On-line Optimization executions must be longer than the settling time to ensure that the units have returned to steady state operations before the optimization is conducted again. This is illustrated in Figure 31, after Darby and White (1988). The figure shows an execution frequency for optimization that was satisfactory for one process may be too rapid for another process, which has a longer settling time. In Figure 31a, the process has returned to steady-state operations and held that position until the next optimization. However, in Figure 31b, the process did not have enough time to return to steady-state operations before the optimization altered the operating conditions. The process would continue on an unsteady state path, and operator intervention would be required. The settling time for an ethylene plant is four hours according to Darby and White (1988), and this time for the sulfuric acid contact process is twelve hours according Hertwig (1997).

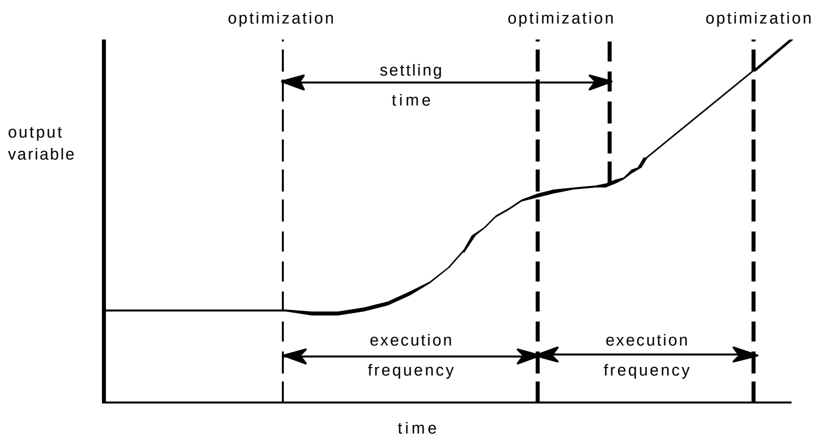
As shown in Figure 32, it is necessary to make sure the process is operating at steady state before the plant data is taken from the distributed control system for conducting On-line Optimization. Steady state plant data are required for steady state process models.

The time series horizontal screening method has been used in industry to detect steady state. In this method, the measured values for key process variables are examined over a specified time interval, and if the values are within a range, e.g. one standard deviation, then the process is operating at steady state. This requires the use of a coordinator program or operator action for identifying steady state and exchanging between the On-line Optimization program and the distributed control system. Excel spreadsheet files are widely used to move the data. Steady state detection and data exchange will be illustrated with simulated data for the simple refinery.

As shown in Figure 32 On-line Optimization executes economic optimization and generates a set of optimal set point. Then these set points are transferred to the coordinator program or the operators as an Excel spreadsheet file. These optimal set points can be sent directly to the distributed control system or they may be viewed by operators before they are sent to the DCS. Before the optimal set points are implemented, the steady state detection program is run to ensure the process is at steady state.



a. Time between optimizations is longer than settling time



b. Time between optimizations is less than settling time

Figure 31 Comparison of Time between Optimizations and Process Setting Time, after Darby and White (1988)

To incorporate the capability for steady state detection, an Excel worksheet program was prepared, *Steady-State.xls,* and it is included in the files with the On-line Optimization program. The simple refinery is used to illustrate the use of this program for time series analysis for steady state detection. The first sheet in the Excel program has 20 sets of simulated plant data that were prepared for the simple refinery. This information is shown in Figure 33 for the first 14 of these data sets. Each column represents data for the 33 measured variables that would be taken from the data historian of the distributed control system for 20 time intervals ending with the current time. The use of an Excel spreadsheet is the industry standard way of selecting data and manipulating data from a DCS.

The second Excel spreadsheet was prepared to analyze this data to determine a time interval that shows the plant operating at steady state.

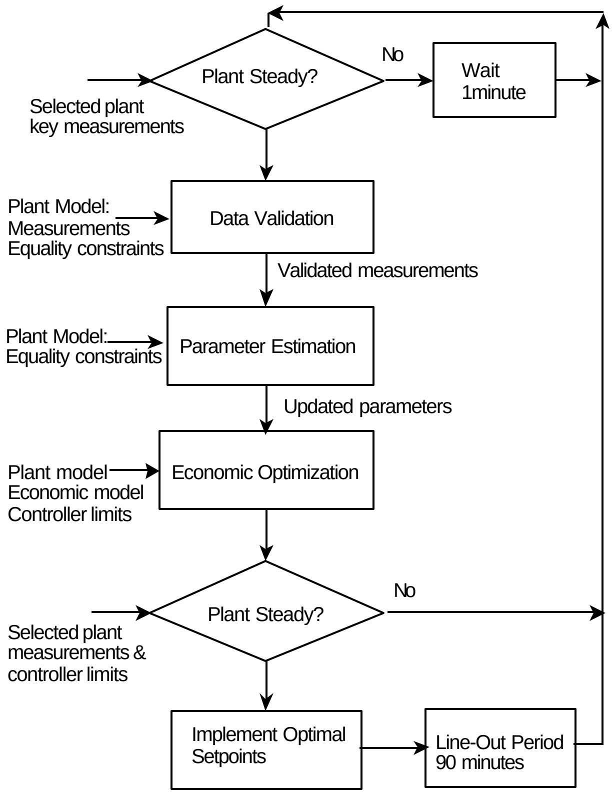


Figure 32 Implementation Procedure for On-Line Optimization, after Kelley,et al. (1996)

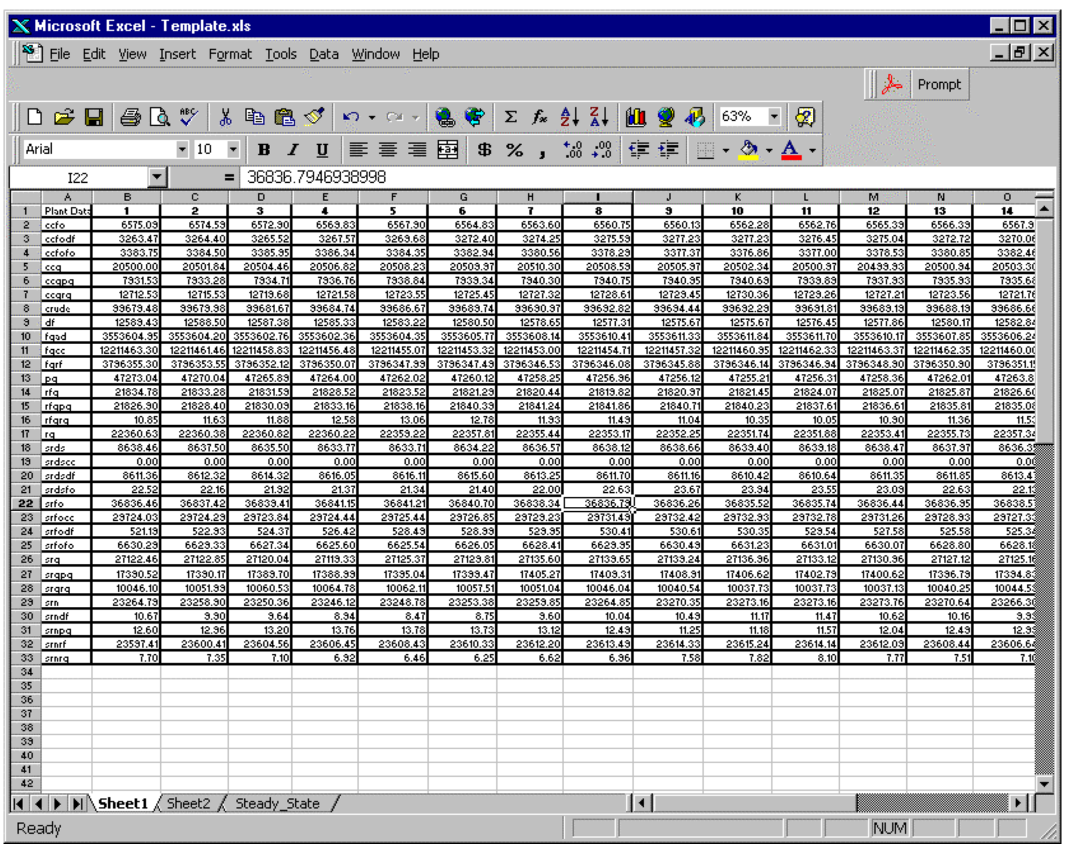
This spreadsheet is shown in Figure 34 and the graphs and buttons were developed using the Visual Basic capabilities that are part of Excel. In this figure, the time series of four of the measured variables can be viewed one at a time. The spreadsheet has the capability of displaying any four of the of the process variables, and the variables that are plotted can be changed by pulling down the menu on the lower left and selecting a variable to be displayed.

After reviewing the data in Figure 34, it was determined that the plant was at steady state during the time periods from 16 to 20, the most recent time. Consequently, the decision is to import the data from one of these periods, say the most recent-20, into the On-line Optimization program. On this diagram, when the Save Steady State Data button is clicked, the program prompts the user to designate the time interval of the data. This information is saved to the third spreadsheet, a single column of data that is not shown here as a figure.

The user is now ready to transfer this steady state data to the On-line Optimization program. Return to the Declaration Window for Measured Variables which is shown in Figure 11 and pull down the File menu (shown in Figure 35) and select Import Plant Data. This action brings up the window shown in Figure 36. This window designates the name of the Excel file that contains the steady state plant data that was previously selected with the Excel time series program. In this example, the designated name is: *Sheet1$B2:b33* . Clicking the Open button will replace the plant date currently in the program. Now that the new data is in place, the On-line Optimization program can be executed to generate the new set of optimal points for the distributed control system.

The execution of the On-line Optimization program generates the set points for the distributed control system, which are shown in Figures 25 and 27. These values can be exported from the On-line Optimization program using the same procedure as importing data. The file menu in these windows has a line Export Plant Data which, when clicked, displays a screen similar to the one in Figure 36. This screen specifies the Excel file that will transfer the data.

The On-line Optimization program requires the standard deviation of the measured variables, which is calculated by the Excel program, *Steady-State.xls* as shown in Figure 11. This program may also be used with different data sets to determine other standard deviation values. Although not shown in Figure 33, the last column in the spreadsheet is the standard deviation of the measured variables, which was calculated using the 20 measurements. This information can be transferred to the On-line Optimization program using the same procedure as was used for the measured variables. However, it is not necessary to use the current plant data to evaluate the standard deviation.



Plant Data	1	2	3	4	5	6	7
ccfo	6575.09	6574.59	6572.90	6569.83	6567.90	6564.83	6563.60
ccfodf	3263.47	3264.40	3265.52	3267.57	3269.68	3272.40	3274.25
ccfofo	3383.75	3384.50	3385.95	3386.34	3384.35	3382.94	3380.56
ccq	20500.00	20501.84	20504.46	20506.82	20508.23	20509.97	20510.30
ccqpq	7931.53	7933.28	7934.71	7936.76	7938.84	7939.34	7940.30
ccqrq	12712.53	12715.53	12719.68	12721.58	12723.55	12725.45	12727.32
crude	99679.48	99679.98	99681.67	99684.74	99686.67	99689.74	99690.97
df	12589.43	12588.50	12587.38	12585.33	12583.22	12580.50	12578.65
fqad	3553604.95	3553604.20	3553602.76	3553602.36	3553604.35	3553605.77	3553608.14
fqcc	12211463.30	12211461.46	12211458.83	12211456.48	12211455.07	12211453.32	12211453.00
fqrf	3796355.30	3796353.55	3796352.12	3796350.07	3796347.99	3796347.49	3796346.53
pq	47273.04	47270.04	47265.89	47264.00	47262.02	47260.12	47258.25
rfq	21834.78	21833.28	21831.59	21828.52	21823.52	21821.29	21820.44
rfqpq	21826.90	21828.40	21830.09	21833.16	21838.16	21840.39	21841.24
rfqrq	10.85	11.63	11.88	12.58	13.06	12.78	11.93
rq	22360.63	22360.38	22360.82	22360.22	22359.22	22357.81	22355.44
srds	8638.46	8637.50	8635.50	8633.77	8633.71	8634.22	8636.57
srdscc	0.00	0.00	0.00	0.00	0.00	0.00	0.00
srdsdf	8611.36	8612.32	8614.32	8616.05	8616.11	8615.60	8613.25
srdsfo	22.52	22.16	21.92	21.37	21.34	21.40	22.00
srfo	36836.46	36837.42	36839.41	36841.15	36841.21	36840.70	36838.34
srfocc	29724.03	29724.29	29723.84	29724.44	29725.44	29726.85	29729.23
srfodf	521.19	522.93	524.37	526.42	528.49	528.99	529.95
srfofo	6630.29	6629.33	6627.34	6625.60	6625.54	6626.05	6628.41
srq	27122.46	27122.85	27120.04	27119.33	27125.37	27129.81	27135.60
srqpq	17390.52	17390.17	17389.70	17388.99	17395.04	17399.47	17405.27
srqrq	10046.10	10051.99	10060.53	10064.78	10062.11	10057.51	10051.04
srn	23264.79	23258.90	23250.36	23246.12	23248.78	23253.38	23259.85
srndf	10.67	9.90	9.64	8.94	8.47	8.75	9.60
srnpq	12.60	12.96	13.20	13.76	13.78	13.73	13.12
srnrf	23597.41	23600.41	23604.56	23606.45	23608.43	23610.33	23612.20
srnrq	7.70	7.35	7.10	6.92	6.46	6.25	6.62

Plant Data	8	9	10	11	12	13	14
ccfo	6560.75	6560.13	6562.28	6562.76	6565.39	6566.39	6567.9
ccfodf	3275.59	3277.23	3277.23	3276.45	3275.04	3272.72	3270.06
ccfofo	3378.29	3377.37	3376.86	3377.00	3378.53	3380.85	3382.46
ccq	20508.59	20505.97	20502.34	20500.97	20499.93	20500.94	20503.30
ccqpq	7940.75	7940.95	7940.69	7939.89	7937.93	7935.93	7935.68
ccqrq	12728.61	12729.45	12730.36	12729.26	12727.21	12723.56	12721.76
crude	99692.82	99694.44	99692.29	99691.81	99689.19	99688.19	99686.66
df	12577.31	12575.67	12575.67	12576.45	12577.86	12580.17	12582.84
fqad	3553610.41	3553611.33	3553611.84	3553611.70	3553610.17	3553607.85	3553606.24
fqcc	12211454.71	12211457.32	12211460.95	12211462.33	12211463.37	12211462.35	12211460.00
fqrf	3796346.08	3796345.88	3796346.14	3796346.94	3796348.90	3796350.90	3796351.1
pq	47256.96	47256.12	47255.21	47256.31	47258.36	47262.01	47263.8
rfq	21819.82	21820.97	21821.45	21824.07	21825.07	21825.87	21826.60
rfqpq	21841.86	21840.71	21840.23	21837.61	21836.61	21835.81	21835.08
rfqrq	11.49	11.04	10.35	10.05	10.90	11.36	11.53
rq	22353.17	22352.25	22351.74	22351.88	22353.41	22355.73	22357.34
srds	8638.12	8638.66	8639.40	8639.18	8638.47	8637.97	8636.3
srdscc	0.00	0.00	0.00	0.00	0.00	0.00	0.00
srdsdf	8611.70	8611.16	8610.42	8610.64	8611.35	8611.85	8613.47
srdsfo	22.63	23.67	23.94	23.55	23.09	22.63	22.13
srfo	36836.79	36836.26	36835.52	36835.74	36836.44	36836.95	36838.57
srfocc	29731.49	29732.42	29732.93	29732.78	29731.26	29728.93	29727.33
srfodf	530.41	530.61	530.35	529.54	527.58	525.58	525.34
srfofo	6629.95	6630.49	6631.23	6631.01	6630.07	6628.80	6628.18
srq	27139.65	27139.24	27136.96	27133.12	27130.96	27127.12	27125.16
srqpq	17409.31	17408.91	17406.62	17402.79	17400.62	17396.79	17394.83
srqrq	10046.04	10040.54	10037.73	10037.73	10037.13	10040.25	10044.53
srn	23264.85	23270.35	23273.16	23273.16	23273.76	23270.64	23266.30
srndf	10.04	10.49	11.17	11.47	10.62	10.16	9.93
srnpq	12.49	11.25	11.18	11.57	12.04	12.49	12.93
srnrf	23613.49	23614.33	23615.24	23614.14	23612.09	23608.44	23606.64
srnrq	6.96	7.58	7.82	8.10	7.77	7.51	7.10

60

Figure 33 Excel Spreadsheet of Simulated Plant Data for the Simple Refinery

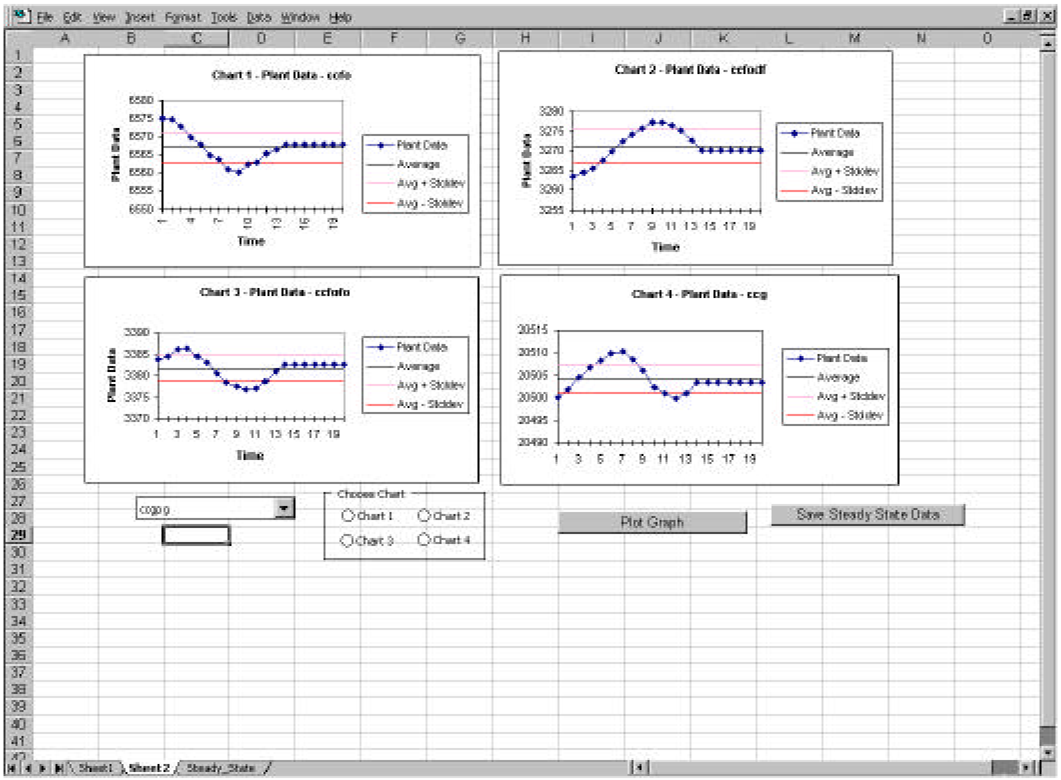


Figure 34 Excel Spreadsheet Showing the Time Series Graphs of the Data

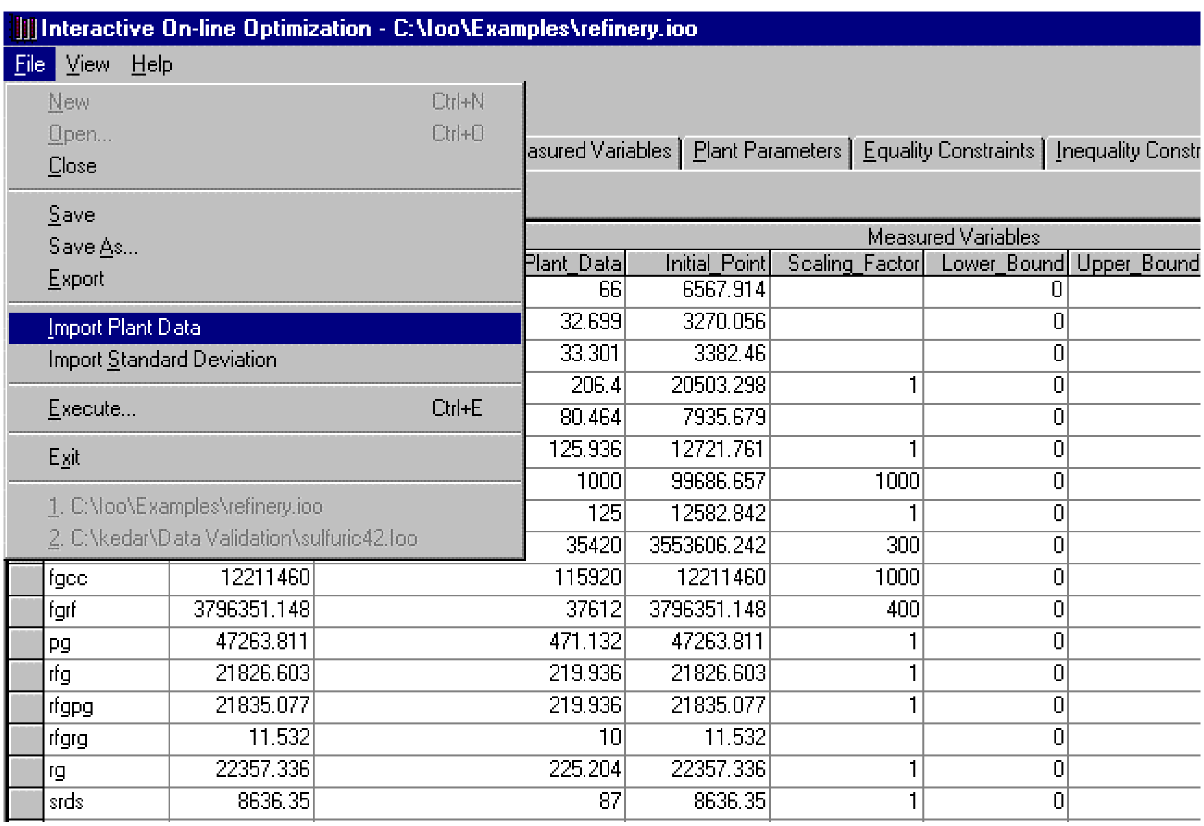


Figure 35 The Import Option in the File Menu of Interactive On-Line Optimization

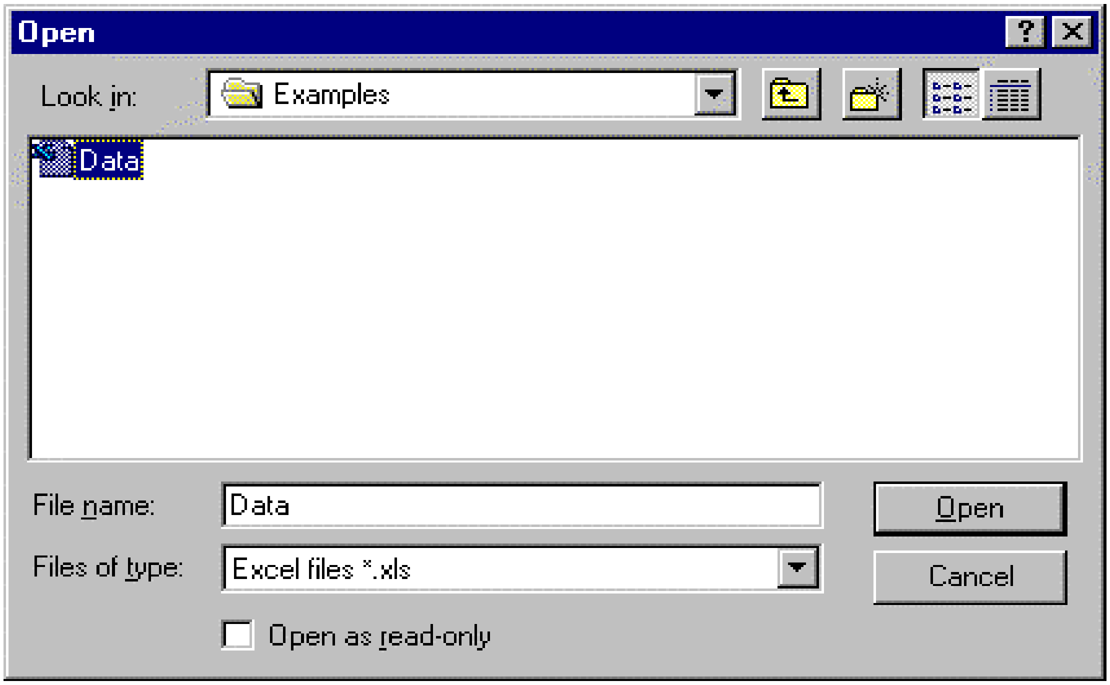


Figure 36 The Dialogue Box that opens when Import is Clicked

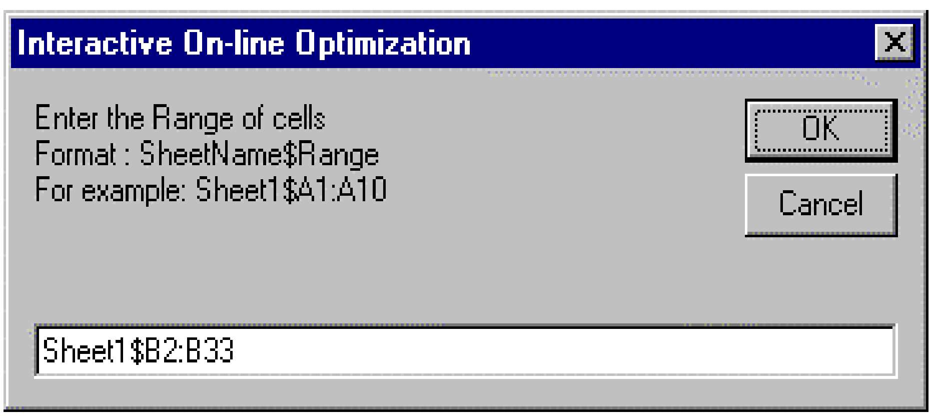


Figure 37 The Screen to Enter the Excel Sheet Name and Range

VII. EXAMPLE - SIMPLE REFINERY

The following gives a brief description of the simple refinery given by Pike (1986). This process example is used to demonstrate the procedure for entering plant simulation and economic information. It also shows how the system generates the GAMS programs for On-line Optimization, and how it presents the results to the user as discussed in previous sections.

As shown in Figure 31, the refinery process includes three units, which are a crude oil atmospheric distillation column (AD), a catalytic cracking unit (CC), and a catalytic reformer (RF). The crude oil distillation column separates crude oil into five streams: fuel gas (FGAD), straight-run gasoline (SRG), straight-run naphtha (SRN), straight-run distillate (SRDS), and straight-run fuel oil (SRFO). Part of the straight-run naphtha is processed through the catalytic reformer to improve its quality, i.e., to increase the octane number. Also, part of straight-run distillate and straight-run fuel oil are processed through the catalytic cracking unit to improve their quality so they can be blended into gasoline. The simple refinery produces four products: premium gasoline, regular gasoline, diesel fuel, fuel oil, and by-product fuel gas.

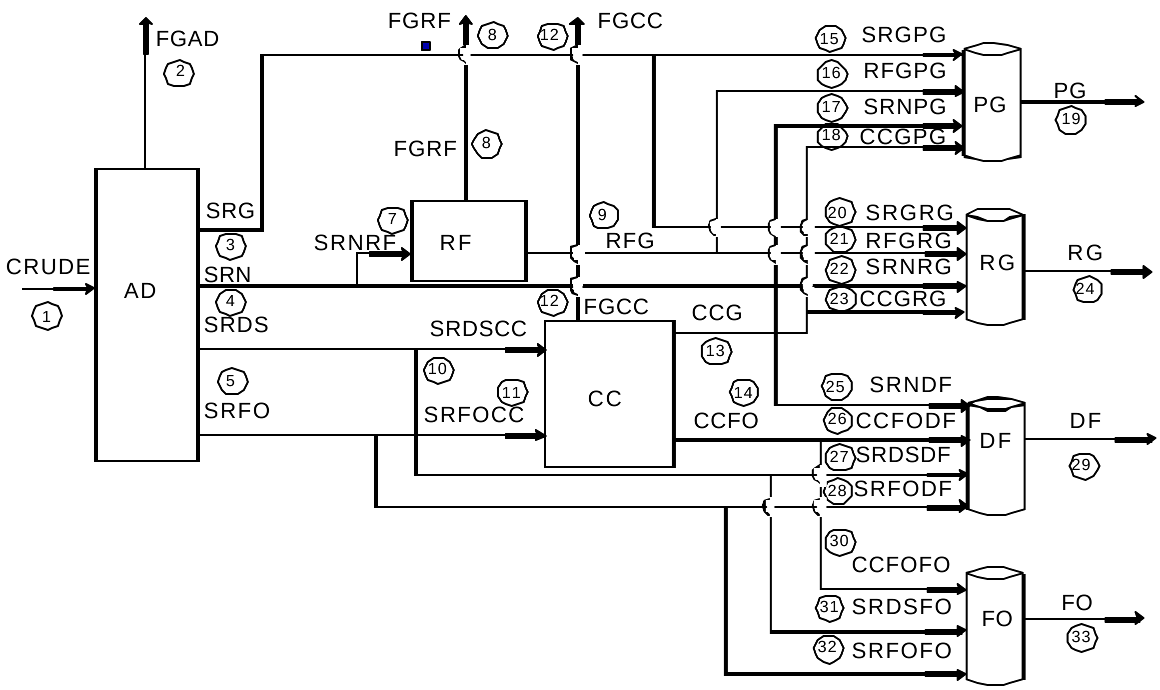


Figure 38 Process Flow Diagram for simple Refinery, after Pike(1986)

In total, the process model includes 33 process variables, 13 process parameters, 21 equality constraints, and 16 inequality constraints. The economic model is represented by a profit function that includes the cost of processing feed through each unit, the cost of raw materials and the sales from products. A description follows for applying the On-line Optimization system to the simple refinery.

In Table 20, the names and definitions of process variables are listed for the simple refinery. Among all the process variables, 32 variables are measured, and one variable is unmeasured as shown in Table 20. Also, a set of simulated plant measurements from the distributed control system and the corresponding standard deviations for these process variables are given in this table for the measured variables. In Table 21, the unit capacities, operating costs, and mass and volumetric yields are listed for the three process units in the refinery. These are typical of a medium size refinery in the Gulf coast area. Operating costs were furnished by the technical division of a major oil company which has refineries on the Gulf Coast. The mass yields were taken from those reported by Aronfsky, Dutton and Tayyaabkhan (1978) and were converted to volumetric yields by using API gravity data. The volumetric yields are considered to be process parameters which are estimated using the reconciled process data from the distributed control system. The names and definitions for these parameters are given in Table 22.

The quality specification and physical properties are given in Table 23 for the process streams, and the crude cost and the product sales prices are given in Table 24. The data in Table 23 was reported by Aronfsky, et al. (1978), and the cost and prices in Table 24 were obtained from the *Oil and Gas Journal* (Anonymous, 1982). The information given in Tables 20 to 24 is required to construct the economic model (objective function for economic optimization) and the plant model (constraint equations) for the petroleum refinery.

It is standard practice to present the process and economic models in matrix form when only linear constraints are used for the plant simulation. This matrix is shown in Table 25 for the simple refinery. In the first row the coefficients of the terms in the objective function are listed under their corresponding variables. The sales prices are shown as positive, and the costs are shown as negative in the problem, so the problem is formulated to maximize the profit. These numbers were taken from Table 24, where the crude cost ($32.00/bbl) was conveniently combined with the operating cost of the crude oil atmospheric distillation column ($1.00/bbl) to show a total cost of $33.00 per bbl of crude oil processed. Consequently, the first row of Table 25 represents the objective function given below as Equation (1):

$$\begin{aligned}&- 33.0\ CRUDE + 0.01965\ FGAD - 2.50\ SRNRF + 0.01965\ FGRF \qquad (1)\\&-2.20\ SRDSCC - 2.20\ SRFOCC + 0.01965\ FGCC + 45.36\ PG\\&+ 43.68\ RG + 40.32\ DF + 13.14\ FO\end{aligned}$$

The constraint equations begin with the second row in Table 25. In Table 25, the second row is the crude availability constraint limiting the refinery to 110,000 bbl/day. This is followed by the four quantity and quality constraints associated with each product. These are the daily production and blending requirements and two quality constraints. These have been extracted from Table 25 and are shown in Table 26 for four products. The minimum production constraint states that the refinery must produce at least 10,000 bbl/day of premium gasoline to meet the company's marketing division's requirements. The blending constraints state that the sum of the streams going to produce premium gasoline must equal the daily production of premium gasoline. The quality constraints use linear blending, and the sum of each component weighted by its quality must meet or exceed the quality of the product. This is illustrated with premium gasoline octane rating blending constraint which is written as the following using the information from the matrix:

$$78.5\ SRGPG + 104.0\ RFGPG + 65.0\ SRNPG + 93.7\ CCGPG - 93.0\ PG \geq 0 \qquad (2)$$

where the premium gasoline must have an octane number of at least 93.0. Corresponding, inequality constraints are specified in Table 26 using the same procedure for premium gasoline vapor pressure, regular gasoline octane number and vapor pressure, diesel fuel density and sulfur content and fuel oil density, and sulfur content.

The next set of information given in the constraint equation matrix, Table 25, is the description of the operation of the process unit using the volumetric yield to formulate the material balances shown in Table 21. This section of the matrix has been extracted and is shown in Table 27 for the three process units. Referring to volumetric yields for the crude oil distillation column, this data states that 35.42 times the volumetric flow rate of crude produces the flow rate of fuel gas from the distillation column.

$$35.42^{*}\ CRUDE - FGAD = 0 \qquad (3a)$$

or

$$VFGAD^{*}CRUDE-FGAD=0$$

where VFGAD is a parameter that represents the volumetric yield of fuel gas from crude which has an estimated value of 35.42. The names of the other parameters are listed in Table 22. These types of parameters are variables in parameter estimation step and are constant in data validation and economic optimization steps, where the constants are determined from parameter estimation. Corresponding yields of the other products from the crude oil distillation are determined the same way.

Table 20 Description and Plant Data for Process Variables of the Refinery

Variables	Name	Definition (Flow rates are in barrels per day)	Plant Data	Standard Deviation
Measured	CRUDE	Crude oil flow rate to atmospheric distillation column (AD)	99686.7	1000.0
	FGAD	Fuel gas flow rate from AD	3553606	35420.0
		Straight run gasoline flow rate from AD	27125.2	270.0
	SRG	Straight run naphtha flow rate from AD	23266.3	237.0
	SRN	Straight run distillate flow rate from AD	8636.35	87.0
	SRDS	Straight run fuel oil flow rate from AD	36838.6	372.0
	SRFO	Straight run naphtha feed rate to reformer	23606.6	237.0
	SRNRF	(RF)		
		Fuel gas flow rate from the reformer	3796351	37612.0
	FGRF	Reformer gasoline flow rate	21826.6	219.9
	RFG	Straight run distillate flow rate to the catalytic	0.004	10.0
	SRDSCC	cracking unit (CCU)		
		Straight run fuel oil flow rate to the CCU	29727.3	300.0
	SRFOCC	Fuel gas flow rate from the CCU	1.2212E	115920.0
	FGCC	Gasoline flow rate from CCU	+7	206.4
	CCG	Fuel oil flow rate from CCU	20503.3	66.0
	CCFO	Straight run gasoline flow rate for premium	6567.9	170.7
	SRGPG	gasoline (PG) blending	17394.8	
		Reformer gasoline flow rate for PG blending		219.9
	RFGPG	Straight run naphtha flow rate for PG	21835.1	10.0
	SRNPG	blending	12.99	80.5
	CCGPG	CCU gasoline flow rate for PG blending	7935.7	471.1
	PG	Premium gasoline flow rate	47263.8	99.3
	SRGRG	Straight run gasoline flow rate for regular	10044.6	
		gasoline (RG) blending		10.0
	RFGRG	Reformer gasoline flow rate for RG blending	11.532	10.0
	SRNRG	Straight run naphtha flow rate for RG	7.100	125.9
	CCGRG	blending	12721.8	225.2
	RG	CCU gasoline flow rate for RG blending	22357.3	10.0
	SRNDF	Regular gasoline flow rate	9.994	
		Straight run naphtha flow rate for diesel fuel		32.7
	CCFODF	(DF) blending	3270.1	87.0
	SRDSDF	CCU fuel oil flow rate for DF blending	8613.5	5.3
	SRFODF	Straight run distillate flow rate for DF	525.34	125.0
	DF	blending	12582.8	33.3
	CCFOFO	Straight run fuel oil flow rate for DF blending	3382.5	
		No. 2 diesel fuel flow rate		10.0
	SRDSFO	CCU fuel oil flow rate for fuel oil (FO)	22.13	66.7
	SRFOFO	blending	6628.2	
		Straight run distillate flow rate for FO blending		
		Straight run fuel oil flow rate for FO blending		
Unmeasured	FO	No. 6 fuel oil flow rate		

Table 21 Capacities, Operating Costs and Volumetric Yields for the Refinery Process Units

Unit	Capacity (bbl/day)	Operating Cost ($/bbl)	Input	Output	Mass Yield of Output Stream (lb/lb)	Volumetric Yield of Output Stream (bbl/bbl)
Crude oil Atmospheric Distillation Column	100,000	1.00	CRUDE	FGAD	0.029	35.42
				SRG	0.236	0.270
				SRN	0.223	0.237
				SRDS	0.087	0.087
				SRFO	0.426	0.372
Catalytic Reformer	25,000	2.50	SRNRF	FGRF	0.138	158.7
				RFG	0.862	0.928
Catalytic Cracking Unit	30,000	2.20	SRDSCC	FGCC	0.273	336.9
				CCG	0.536	0.619
				CCFO	0.191	0.189
			SRFOCC	FGCC	0.277	386.4
				CCG	0.527	0.688
				CCFO	0.196	0.220

(3b)

Table 22 Names and Definition of Parameters for the Refinery

Units of Parameters	Names of Parameters	Initial Values	Definitions of parameters Volumetric yields (BBL output/BBL input)
Crude Oil Atmospheric Distillation Column	VFGAD	35.42	BBLs of fuel gas per BBL crude
	VSRG	0.27	BBLs of straight-run gasoline per BBL crude
	VSRN	0.237	BBLs of straight-run naphtha per BBL crude
	VSRDS	0.087	BBLs of straight-run distillate per BBL crude
	VSRFO	0.372	BBLs of Straight-run fuel oil per BBL crude
Catalytic Reformer	VSRNFGRF	158.7	BBLs of reformer fuel gas per BBL of straight-run naphtha
			BBLs reformer gasoline per BBL straight-run naphtha
	VSRNRFG	0.928	
Catalytic Cracking	VSRDSFGCC	336.9	BBLs of fuel gas per BBL straight-run distillate
	VSRDSCCG	0.619	BBLs of gasoline from CC per BBL straight-run distillate
			BBLs of fuel oil per BBL of straight-run distillate
	VSRDSCCFO	0.189	BBLs of fuel gas per BBL straight-run fuel oil
	VSRFOFGCC	386.4	BBLs of gasoline from CC per BBL of straight-run fuel oil
	VSRFOCCG	0.688	BBLs of fuel oil per BBL straight-run fuel oil
	VSRFOCCFO	0.220	

For the catalytic reformer the yield of fuel gas (FGRF) and the reformer gasoline (RFG) are given by the following equations:

$$158.7 * SRNRF - FGRF = 0 \tag{4a}$$

or

$$VSRNFGRF * SRNRF - FGRF = 0 \tag{4b}$$

and

$$0.928 * SRNRF - RFG = 0 \tag{5a}$$

or

$$VSRNRFG * SRNRF - RFG = 0 \tag{5b}$$

where VSRNFGRF and VSNRFG are process parameters that represent the volumetric yields of fuel gas and gasoline respectively from straight-run naphtha. Similar equations are used for the catalytic cracking unit shown in the matrix of Table 25 and are summarized in Table 27.

Table 23 Quality Specifications and Physical Properties for Products and Intermediate Streams for the Refinery

Stream	Motor Octane Number	Vapor pressure (mmHg)	Density (lb/bbl)	Sulfur Content (lb/bbl)
Premium Gasoline	$ 93.0	# 12.7	-	-
Regular Gasoline	$ 87.0	# 12.7	-	-
Diesel Fuel	-	-	# 306.0	# 0.5
Fuel Oil	-	-	# 352.0	# 3.0
SRG	78.5	18.4	-	-
RFG	104.0	2.57	-	-
SRN	65.0	6.54	272.0	0.283
CCG	93.7	6.90	-	-
CCFO	-	-	294.4	0.353
SRDS	-	-	292.0	0.526
SRFO	-	-	295.0	0.980

Table 24 Crude Oil Cost and Product Sales Prices for the Refinery

Names	**Prices**
Gulf Cost Crude	$32.00/bbl
Premium Gasoline	$45.36/bbl
Regular Gasoline	$43.68/bbl
No.2 Diesel Fuel	$40.32/bbl
No.6 Fuel Oil	$13.14/bbl
Fuel Gas	$0.01965/bbl

The use of volumetric yields to give linear equations to describe the performance of the process units is required for linear programming. The results will be satisfactory as long as the volumetric yields precisely describe the performance of these process units. These volumetric yields are a function of the operating conditions of the unit, e.g. temperature, feed flow rate, catalyst activity etc. These volumetric yields can be estimated by treating them as parameters, which are determined from the reconciled process data, which is done at the parameter estimation step of On-line Optimization.

The last group of terms in Table 25 gives the material balance around points where streams split among process units and blend into products. The stream to be divided is given a coefficient of plus one, and the resulting streams have a coefficient of minus one. For example, the straight run naphtha from the crude oil distillation is split into four streams. One is sent to the catalytic reformer and the other three are used in blending premium gasoline, regular gasoline and diesel fuel. The equation for this split is:

$$\mathrm{SRN - SRNRF - SRNPG - SRNRG - SRNDF = 0} \tag{6}$$

There are a total of seven stream splits as shown in Table 25.

Above is a brief description about the simple refinery and the information given here is used to conduct On-line Optimization through the Interactive On-Line Optimization System as discussed in the tutorial. The GAMS output file for economic optimization of the simple refinery is given in Appendix A.

Table 25 Refinery Objective Function and Constraint Equations

		Atmospheric Distillation							Reformer	
	CRUDE	FGAD	SRG	SRN	SRDS	SRFO	SRNRF	FGRF	RFG	
Objective Function	-33.0	.01965					-2.50	.01965		
Crude Availability	1.0									
Products										
Premium Gasoline										
Min. PG Prod.										
PG Blending										
PG Octane Rating										
Regular Gasoline										
Min. RG Prod.										
RG Blending										
RG Octane Rating										
RG Vapor Press.										
Diesel Fuel										
Min. DF Prod.										
DF Blending										
DF Density Spec.										
DF Sulfur Spec.										
Fuel Oil										
Min. FO Prod.										
FO Blending										
FO Density Spec.										
FO Sulfur Spec.										
Process Units										
Atm. Distillation										
AD Capacity		1.0								
FGAD Yield		35.42	-1.0							
SRG Yield		0.270		-1.0						
SRN Yield		0.237			-1.0					
SRDS Yield		0.087				-1.0				
SRFO Yield		0.372					-1.0			
Reformer										
RF Capacity								1.0		
FGRF Yield								158.7	-1.0	
RFG Yield								0.928		-1.0
Catalytic Cracker										
CC Capacity										
FGCC Yield										
CCG Yield										
CCFO Yield										
Stream Splits										
SRG			1.0							
SRN				1.0						
SRDS							-1.0			
SRFO					1.0					
RFG						1.0				
CCG									1.0	
CCFO										

Table 25 Refinery Objective Function and Constraint Equations (continued)

	Catalytic Cracker							Premium Gasoline Blending		
	SRDSCC	SRFOCC	FGCC	CCG	CCFO	SRGPG	RFGPG	SRNPG	CCGPG	PG
Objective Function	-2.20	-2.20	-.01965							45.36
Crude Availability										
Products										
Premium Gasoline										
Min. PG Prod.										1.0
PG Blending							1.0	1.0	1.0	1.0
-1.0										
PG Octane Rating						78.5	104.0	65.0	93.7	-93.0
PG Vapor Press.						18.4	2.57	6.54	6.90	-12.7
Regular Gasoline										
Min. RG Prod.										
RG Blending										
RG Octane Rating										
RG Vapor Press.										
Diesel Fuel										
Min. DF Prod.										
DF Blending										
DF Density Spec.										
DF Sulfur Spec.										
Fuel Oil										
Min. FO Prod.										
FO Blending										
FO Density Spec.										
FO Sulfur Spec.										
Process Units										
Atm. Distillation										
AD Capacity										
FGAD Yield										
SRG Yield										
SRN Yield										
SRDS Yield										
SRFO Yield										
Reformer										
RF Capacity										
FGRF Yield										
RFG Yield										
Catalytic Cracker										
CC Capacity		1.0	1.0							
FGCC Yield		336.9	386.4	-1.0						
CCG Yield		0.619	0.688		-1.0					
CCFO Yield		0.189	0.220			-1.0				
Stream Splits										
SRG						-1.0				
SRN								-1.0		
SRDS	-1.0									
SRFO		-1.0								
RFG							-1.0			
CCG				1.0						-1.0
CCFO					1.0					

Table 25 Refinery Objective Function and Constraint Equations (continued)

Regular Gasoline Blending					Diesel Fuel Blending					Fuel Oil BLending					
SRGRG	RFGRG	SRNRG	CCGRG	RG	SRNDF	CCFODF	SRDSDF	SRFODF	DF	CCFOFO	SRDSFO	SRFOFO	FO		
				43.68					40.32				13.14	= Maximum	OBJ
														≤ 110,00	CRDAVAIL
														≥ 10,000	PGMIN
														= 0	PGBLEND
														≥ 0	PGOCTANE
														≤ 0	PGVAPP
				1.0										≥ 10,000	RGMIN
1.0	1.0	1.0	1.0	-1.0										= 0	RGBLEND
78.5	104.0	65.0	93.7	-87.0										≥ 0	RGOCTANE
18.4	2.57	6.54	6.90	-12.7										≤ 0	RGVAPP
									1.0					≥ 10,000	DFMIN
					1.0	1.0	1.0	1.0	-1.0					= 0	DFBLEND
					272.0	294.4	292.0	295.0	-306.0					≤ 0	DFDENS
					0.283	0.353	0.526	0.980	-0.50					≤ 0	DFSULFUR
													1.0	≥ 10,000	FOMIN
										1.0	1.0	1.0	-1.0	= 0	FOBLEND
										294.4	292.0	295.0	-352.0	≤ 0	FODENS
										0.353	0.526	0.980	-3.0	≤ 0	FOSULFUR
														≥ 100,000	ADCAP
														= 0	ADFGYLD
														= 0	ADSRGYLD
														= 0	ADNYLD
														= 0	ADDSYLD
														= 0	ADFOYLD
														≤ 25,000	RFCAP
														= 0	RFFGYLD
														= 0	RFRFGYLD
														≤ 30,000	CCCAP
														= 0	CCFGYLD
														= 0	CCGYLD
														= 0	CCFOYLD
-1.0														= 0	SRGSPLIT
	-1.0				-1.0									= 0	SRNSPLIT
							-1.0				-1.0			= 0	SRDSSPLIT
								-1.0				-1.0		= 0	SRFOPLIT
	-1.0													= 0	RFGSPLIT
			-1.0											= 0	CCGSPLIT
						-1.0				-1.0				= 0	CCFOSPLT

Table 26 Quantity and Quality Constraints of the Refinery Products

Premium Gasoline

	SRGPG	RFGPG	SRNPG	CCGPG	PG	RHS
Min. P.G. Production					1.0	≥10,000
PG Blending	1.0	1.0	1.0	1.0	-1.0	= 0
PG Octane Rating	78.5	104.0	65.0	93.7	-93.0	≥ 0
PG Vapor Pressure	18.4	2.57	6.54	6.90	-12.7	≤ 0

Regular Gasoline

	SRGRG	RFGRG	SRNRG	CCGRG	RG	RHS
Min R.G. Production					1.0	≥ 10,000
RG Blending	1.0	1.0	1.0	1.0	-1.0	= 0
RG Octane Rating	78.5	104.0	65.0	93.7	-87.0	≥ 0
RG Vapor Pressure	18.4	2.57	6.54	6.90	-12.7	≤ 0

Diesel Fuel

	SRNDF	CCFODF	SRDSDF	SRFODF	DF	RHS
Min D.F. Production					1.0	>10,000
DF Blending	1.0	1.0	1.0	1.0	-1.0	= 0
DF Density Spec.	272.0	294.4	292.0	295.0	-306.0	≤ 0
DF Sulfur Spec.	0.283	0.353	0.526	0.980	-0.50	≤ 0

Fuel Oil

	CCFOFO	SRDSFO	SRFOFO	FO	RHS
Min. FO Production				1.0	≥ 10,000
FO Blending	1.0	1.0	1.0	-1.0	= 0
FO Density Spec.	294.4	292.0	295.0	-352.0	≤ 0
FO Sulfur Spec.	0.353	0.526	0.980	-3.0	≤ 0

Table 27 Process Unit Material Balances Using Volumetric Yields

Crude Oil Atmospheric Distillation Column:

	CRUDE	FGAD	SRG	SRN	SRDS	SRFO	RHS
AD Capacity	1.0						≤ 100,000
FGAD Yield	35.42	-1.0					= 0
SRG Yield	0.270		-1.0				= 0
SRN Yield	0.237			-1.0			= 0
SRDS Yield	0.087				-1.0		= 0
SRFO Yield	0.372					-1.0	= 0

Catalytic Reformer:

	SRNRF	FGRF	RFG	RHS
RF Capacity	1.0			≤ 25,000
FGRF Yield	158.7	-1.0		= 0
RFG Yield	0.928		-1.0	= 0

Catalytic Cracking Unit:

	SRDSCC	SRFOCC	FGCC	CCG	CCFO	RHS
CC Capacity	1.0	1.0				≤ 30,000
FGCC Yield	336.9	386.4	-1.0			= 0
CCG Yield	0.619	0.688		-1.0		= 0
CCFO Yield	0.189	0.220			-1.0	= 0

VIII. ACKNOWLEDGEMENTS

We gratefully acknowledge the Environmental Protection Agency for support of this work. Also, the assistance of Mr. Tai Lee, Ms. Gayathri Srinivasan, Mr. Huitao Liu, Ms. Melanie Mitchum in the preparation of this manual, and Ms. Qing Chen in Visual Basic programming was invaluable in developing the program.

IX. REFERENCES:

Anonymous, 1982, *Oil and Gas Journal*, 394 (May 3, 1982).

Aronofsky, J. S., J. M. Dutton, and M. T. Tayyabkhan, (1978), *Managerial Planning with Linear Programming in Process Industry Operations*, Johns Wiley & Sons, INC., New York.

Brooke, A., D. Kendrick, and A. Meeraus, 1996, *GAMS User=s Guide*, Release 2.25, GAMS Development Corporation, Washington, D.C.

Chen, X., 1998, *Optimal Implementation of On-Line Optimization for Chemical and Refinery Processes*, Ph. D. Dissertation, Louisiana State University, Louisiana.

Chen, X., Q. Chen, Z. Zhang, R.W. Pike, T.A. Hertwig, and J.R. Hopper, 1996, AInteractive On-Line Optimization for Source Reduction in Chemical Plants,@ *The Eigth Annual Symposium for Emerging Technologies in Hazardous Waste Management, Industrial & Engineering Chemistry Division of American Chemical Society*, Birmingham, Alabama.

Darby, M. L., and D.C. White, (1988), "On-Line Optimization of Complex Process Units" *Chemical Engineering Progress*, Vol. 84, No. 8 p. 51-59.

Hertwig, T., 1997, Private Communication.

Kelly, D. N., F. C. Fatora, and S. L. Davenport, (1996), AImplementation of a Closed Loop Real Time Optimization System on a Large Scale Ethylene Plant,@ .

Pike, R.W., 1986, *Optimization for Engineering Systems*, Van Nostrand Reinhold Publishers, New York.

Appendix A
GAMS Output File of Economic Optimization for Simple Refinery

```
Economic Optimization Program                                   03/19/01 10:59:24  PAGE     1
                                                       GAMS 2.50A    Windows NT/95/98


 * The following are the Measured Variables
 VARIABLES
 ccfo, ccfodf, ccfofo, ccg, ccgpg, ccgrg, crude, df,
 fgad, fgcc, fgrf, pg, rfg, rfgpg, rfgrg, rg,
 srds, srdscc, srdsdf, srdsfo, srfo, srfocc, srfodf, srfofo,
 srg, srgpg, srgrg, srn, srndf, srnpg, srnrf, srnrg;

 VARIABLE ObjVar objective or profit function;
 * The following are the Unmeasured Variables
 VARIABLES
 fo;

 * The following are the Parameters in the Model
 SCALARS
 vfgad       / 35.64776 /
 vsrds       / 0.08696 /
 vsrdsccfo       / 0.189 /
 vsrdsccg       / 0.619 /
 vsrdsfgcc       / 336.9 /
 vsrfo       / 0.3698 /
 vsrfoccfo       / 0.22317 /
 vsrfoccg       / 0.69184 /
 vsrfofgcc       / 390.19894 /
 vsrg       / 0.27396 /
 vsrn       / 0.23525 /
 vsrnfgrf       / 162.08687 /
 vsrnrfg       / 0.93296 /
 ;

 VARIABLES
 ObjVar Objective function using ' ' algorithm;

 EQUATIONS
 * The Constraints
 EQU1, EQU2, EQU3, EQU4, EQU5, EQU6,
 EQU7, EQU8, EQU9, EQU10, EQU11, EQU12,
 EQU13, EQU14, EQU15, EQU16, EQU17, EQU18,
 EQU19, EQU20, EQU21,
 INEQU1, INEQU2, INEQU3, INEQU4, INEQU5, INEQU6,
 INEQU7, INEQU8, INEQU9, INEQU10, INEQU11, INEQU12,
 INEQU13, INEQU14, INEQU15, ObjName;

 ObjName..ObjVar=E=
 -33*crude+0.01965*fgad-2.5*srnrf+0.01965*fgrf-2.2*srdscc-2.2*srfocc+0.01965*fgcc+45.36*pg+43.68*rg+40.32*df+13.14*fo;
_Economic Optimization Program                                  03/19/01 10:59:24  PAGE     2
                                                       GAMS 2.50A    Windows NT/95/98


 EQU1..SRGPG + RFGPG + SRNPG + CCGPG - PG =E=0;
 EQU2..SRGRG + RFGRG + SRNRG + CCGRG - RG =E= 0;
 EQU3..SRNDF + CCFODF + SRDSDF + SRFODF - DF =E= 0;
 EQU4..CCFOFO + SRDSFO + SRFOFO - FO =E= 0;
 EQU5..VFGAD*CRUDE - FGAD =E= 0;
 EQU6..VSRG*CRUDE - SRG =E= 0;
 EQU7..VSRN*CRUDE - SRN =E= 0;
 EQU8..VSRDS*CRUDE - SRDS =E= 0;
 EQU9..VSRFO*CRUDE - SRFO =E= 0;
 EQU10..VSRNFGRF*SRNRF - FGRF =E= 0;
 EQU11..VSRNRFG*SRNRF - RFG =E= 0;
 EQU12..VSRDSFGCC*SRDSCC + VSRFOFGCC*SRFOCC - FGCC =E= 0;
 EQU13..VSRDSCCG*SRDSCC + VSRFOCCG*SRFOCC - CCG =E= 0;
 EQU14..VSRDSCCFO*SRDSCC + VSRFOCCFO*SRFOCC - CCFO =E= 0;
 EQU15..SRG- SRGPG - SRGRG =E= 0;
 EQU16..SRN - SRNRF - SRNPG - SRNRG - SRNDF =E= 0;
 EQU17..SRDS - SRDSCC - SRDSDF - SRDSFO =E= 0;
 EQU18..SRFO - SRFOCC - SRFODF - SRFOFO =E= 0;
 EQU19..RFG - RFGPG - RFGRG =E= 0;
 EQU20..CCG - CCGPG - CCGRG =E= 0;
 EQU21..CCFO - CCFODF - CCFOFO =E= 0;

 INEQU1..CRUDE=L=100000;
 INEQU2..CRUDE=L=110000;
 INEQU3..78.5*SRGPG + 104*RFGPG + 65*SRNPG + 93.7*CCGPG - 93*PG =G= 0;
 INEQU4..18.4*SRGPG + 2.57*RFGPG + 6.54*SRNPG + 6.9*CCGPG - 12.7*PG=L= 0;
```

```
  77  INEQU5..78.5*SRGRG + 104*RFGRG + 65*SRNRG + 93.7*CCGRG - 87*RG =G= 0;
  78  INEQU6..18.4*SRGRG + 2.57*RFGRG + 6.54*SRNRG + 6.9*CCGRG - 12.7*RG =L= 0;
  79  INEQU7..272*SRNDF + 294.4*CCFODF + 292*SRDSDF + 295*SRFODF - 306*DF =L= 0;
  80  INEQU8..0.283*SRNDF + 0.353*CCFODF + 0.526*SRDSDF + 0.98*SRFODF - 0.5*DF =L= 0;
  81  INEQU9..294.4*CCFOFO + 292*SRDSFO + 295*SRFOFO - 352*FO =L= 0;
  82  INEQU10..0.353*CCFOFO + 0.526*SRDSFO + 0.98*SRFOFO - 3*FO =L= 0;
  83  INEQU11..SRDSCC + SRFOCC =L= 30000;
  84  INEQU12..pg =g= 10000;
  85  INEQU13..rg  =g= 10000;
  86  INEQU14..df =g= 10000;
  87  INEQU15..fo =g= 10000;
  88
  89  ccfo.L=6567.914; ccfodf.L=3270.056; ccfofo.L=3382.46;
  90  ccg.L=20503.298; ccgpg.L=7935.679; ccgrg.L=12721.761;
  91  crude.L=99686.657; df.L=12582.842; fgad.L=3553606.242;
  92  fgcc.L=12211460; fgrf.L=3796351.148; pg.L=47263.811;
  93  rfg.L=21826.603; rfgpg.L=21835.077; rfgrg.L=11.532;
  94  rg.L=22357.336; srds.L=8636.35; srdscc.L=0.004;
  95  srdsdf.L=8613.47; srdsfo.L=22.133; srfo.L=36838.565;
_Economic Optimization Program                                    03/19/01 10:59:24  PAGE      3
                                                         GAMS 2.50A    Windows NT/95/98


  96  srfocc.L=29727.325; srfodf.L=525.336; srfofo.L=6628.184;
  97  srg.L=27125.161; srgpg.L=17394.826; srgrg.L=10044.588;
  98  srn.L=23266.302; srndf.L=9.994; srnpg.L=12.99;
  99  srnrf.L=23606.639; srnrg.L=7.1;
 100  ccfo.LO=0; ccfodf.LO=0; ccfofo.LO=0;
 101  ccg.LO=0; ccgpg.LO=0; ccgrg.LO=0;
 102  crude.LO=0; df.LO=0; fgad.LO=0;
 103  fgcc.LO=0; fgrf.LO=0; pg.LO=0;
 104  rfg.LO=0; rfgpg.LO=0; rfgrg.LO=0;
 105  rg.LO=0; srds.LO=0; srdscc.LO=0;
 106  srdsdf.LO=0; srdsfo.LO=0; srfo.LO=0;
 107  srfocc.LO=0; srfodf.LO=0; srfofo.LO=0;
 108  srg.LO=0; srgpg.LO=0; srgrg.LO=0;
 109  srn.LO=0; srndf.LO=0; srnpg.LO=0;
 110  srnrf.LO=0; srnrg.LO=0;
 111  srnrf.UP=25000;
 112
 113  fo.L=9978.183;
 114  fo.LO=0;
 115
 116
 117  MODEL refinery /ALL/;
 118  OPTION LIMCOL=0;
 119  OPTION LIMROW=0;
 120  OPTION ITERLIM= 1000;
 121  OPTION DOMLIM= 0;
 122  OPTION RESLIM= 1000;
 123
 124  OPTION NLP=CONOPT;
 125  SOLVE refinery Using NLP Maximizing ObjVar;
 126


COMPILATION TIME     =        0.000 SECONDS     0.7 Mb      WIN-18-097
_Economic Optimization Program                                    03/19/01 10:59:24  PAGE      4
Model Statistics    SOLVE REFINERY USING NLP FROM LINE 125              GAMS 2.50A    Windows NT/95/98



MODEL STATISTICS

BLOCKS OF EQUATIONS      37     SINGLE EQUATIONS       37
BLOCKS OF VARIABLES      34     SINGLE VARIABLES       34
NON ZERO ELEMENTS       125     NON LINEAR N-Z          0
DERIVATIVE POOL           3     CONSTANT POOL           0
CODE LENGTH               1


GENERATION TIME      =        0.000 SECONDS     1.4 Mb      WIN-18-097


EXECUTION TIME       =        0.000 SECONDS     1.4 Mb      WIN-18-097
_Economic Optimization Program                                    03/19/01 10:59:24  PAGE      5
                                                         GAMS 2.50A    Windows NT/95/98


               S O L V E      S U M M A R Y

     MODEL   REFINERY            OBJECTIVE  OBJVAR
     TYPE    NLP                 DIRECTION  MAXIMIZE
     SOLVER  CONOPT              FROM LINE  125
```

```
**** SOLVER STATUS     1 NORMAL COMPLETION
**** MODEL STATUS      1 OPTIMAL
**** OBJECTIVE VALUE          720521.8573

 RESOURCE USAGE, LIMIT          0.109      1000.000
 ITERATION COUNT, LIMIT        11          1000
 EVALUATION ERRORS              0             0


    C O N O P T   Wintel  version 2.042F-003-035
    Copyright (C) ARKI Consulting and Development A/S
                  Bagsvaerdvej 246 A
                  DK-2880 Bagsvaerd, Denmark

 Using default control program.


 ** Optimal solution. Reduced gradient less than tolerance.


 CONOPT time Total                       0.051 seconds
   of which: Function evaluations          0.000 =  0.0%
             Derivative evaluations      0.000 =  0.0%

 Work length =     0.06 Mbytes
    Estimate =    0.06 Mbytes
    Max used =    0.03 Mbytes

                    LOWER          LEVEL          UPPER         MARGINAL

---- EQU EQU1              .              .              .          -19.3200
---- EQU EQU2              .              .              .          -19.3200
---- EQU EQU3              .              .              .          -40.3200
---- EQU EQU4              .              .              .          -40.3200
---- EQU EQU5              .              .              .           -0.0196
---- EQU EQU6              .              .              .          -41.3000
---- EQU EQU7              .              .              .          -45.8776
---- EQU EQU8              .              .              .          -40.3200
---- EQU EQU9              .              .              .          -40.3200
---- EQU EQU10             .              .              .           -0.0196
_Economic Optimization Program                                              03/19/01 10:59:24  PAGE      6
                                                                   GAMS 2.50A    Windows NT/95/98


                    LOWER          LEVEL          UPPER         MARGINAL

---- EQU EQU11             .              .              .          -48.4400
---- EQU EQU12             .              .              .           -0.0196
---- EQU EQU13             .              .              .          -45.5560
---- EQU EQU14             .              .              .          -40.3200
---- EQU EQU15             .              .              .          -41.3000
---- EQU EQU16             .              .              .          -45.8776
---- EQU EQU17             .              .              .          -40.3200
---- EQU EQU18             .              .              .          -40.3200
---- EQU EQU19             .              .              .          -48.4400
---- EQU EQU20             .              .              .          -45.5560
---- EQU EQU21             .              .              .          -40.3200
---- EQU INEQU1          -INF       100000.0000    100000.0000          8.2243
---- EQU INEQU2          -INF       100000.0000    110000.0000           .
---- EQU INEQU3            .              .            +INF           -0.2800
---- EQU INEQU4          -INF      -175293.4991         .                .
---- EQU INEQU5            .              .            +INF           -0.2800
---- EQU INEQU6          -INF       -11261.5259         .               EPS
---- EQU INEQU7          -INF      -164102.6874         .                .
---- EQU INEQU8          -INF          -88.7484         .               EPS
---- EQU INEQU9          -INF      -572084.4726         .                .
---- EQU INEQU10         -INF       -22318.9353         .                .
---- EQU INEQU11         -INF        30000.0000     30000.0000          5.6631
---- EQU INEQU12     10000.0000      46551.3113        +INF              .
---- EQU INEQU13     10000.0000      23547.7727        +INF              .
---- EQU INEQU14     10000.0000      12371.1000        +INF              .
---- EQU INEQU15     10000.0000      10000.0000        +INF          -27.1800
---- EQU OBJNAME           .              .              .            1.0000

                    LOWER          LEVEL          UPPER         MARGINAL

---- VAR CCFO              .         6695.1000        +INF              .
---- VAR CCFODF            .         3331.6440        +INF              .
---- VAR CCFOFO            .         3363.4560        +INF              .
---- VAR CCG               .        20755.2000        +INF              .
---- VAR CCGPG             .        10469.6979        +INF              .
---- VAR CCGRG             .        10285.5021        +INF              .
---- VAR CRUDE             .       100000.0000        +INF              .
---- VAR DF                .        12371.1000        +INF              .
---- VAR FGAD              .      3564776.0000        +INF              .
---- VAR FGCC              .      1.1705968E+7        +INF              .
```

```
---- VAR FGRF              .        3813093.6167      +INF        .
---- VAR PG                .        46551.3113        +INF        .
---- VAR RFG               .        21947.8840        +INF        .
---- VAR RFGPG             .        20229.5924        +INF        .
```



```
                      LOWER        LEVEL        UPPER       MARGINAL

---- VAR RFGRG             .         1718.2916        +INF          .
---- VAR RG                .        23547.7727        +INF          .
---- VAR SRDS              .         8696.0000        +INF          .
---- VAR SRDSCC            .              .           +INF       -5.7434
---- VAR SRDSDF            .         8673.8670        +INF          .
---- VAR SRDSFO            .           22.1330        +INF         EPS
---- VAR SRFO              .        36980.0000        +INF          .
---- VAR SRFOCC            .        30000.0000        +INF          .
---- VAR SRFODF            .          365.5890        +INF          .
---- VAR SRFOFO            .         6614.4110        +INF          .
---- VAR SRG               .        27396.0000        +INF          .
---- VAR SRGPG             .        15852.0210        +INF          .
---- VAR SRGRG             .        11543.9790        +INF          .
---- VAR SRN               .        23525.0000        +INF          .
---- VAR SRNDF             .              .           +INF       -5.5576
---- VAR SRNPG             .              .           +INF       -8.3576
---- VAR SRNRF             .        23525.0000   25000.0000         .
---- VAR SRNRG             .              .           +INF       -8.3576
---- VAR OBJVAR          -INF      720521.8573        +INF          .
---- VAR FO                .        10000.0000        +INF          .

  CCFO
  CCFODF
  CCFOFO
  CCG
  CCGPG
  CCGRG
  CRUDE
  DF
  FGAD
  FGCC
  FGRF
  PG
  RFG
  RFGPG
  RFGRG
  RG
  SRDS
  SRDSCC
  SRDSDF
  SRDSFO
  SRFO
  SRFOCC
  SRFODF
```



```
  SRFOFO
  SRG
  SRGPG
  SRGRG
  SRN
  SRNDF
  SRNPG
  SRNRF
  SRNRG
  OBJVAR      objective or profit function
  FO


**** REPORT SUMMARY :        0     NONOPT
                             0 INFEASIBLE
                             0  UNBOUNDED
                             0     ERRORS


EXECUTION TIME      =        0.000 SECONDS    0.7 Mb      WIN-18-097


USER: Ralph W. Pike                               G990726:1450AP-WIN
      Louisiana State University, Department of Chemical EngineeriDC267

**** FILE SUMMARY

INPUT      C:\PROGRAM FILES\ADVANCED PROCESS ANALYSIS SYSTEM\GAMS25\DO_ECON
```

```
OUTPUT     C:\PROGRAM FILES\ADVANCED PROCESS ANALYSIS SYSTEM\GAMS25\DO_ECON.LST
SAVE       C:\PROGRAM FILES\ADVANCED PROCESS ANALYSIS SYSTEM\GAMS25\PUT_DATA.G0?
```