3

Computer-Aided Process Design

I. INTRODUCTION

Computer software is a tool-of-the-trade for engineers. The capabilities provided by computers for fast calculation, large storage, and logical decisions, plus the available technical and mathematical software, permit engineers to solve large problems and to do it much more rapidly than ever before. The engineers' emphasis can therefore shift from problem solving to planning, conceiving, and interpreting with the information made available. Process design is one of the engineering functions that has been impacted by computers (Peters and Timmerhaus, 1991).

Complex programs that simulate the operation and a complete process or individual units, have been developed by central design departments of the major manufacturing companies, by university departments, and by commercial software organizations. Hundreds, even thousands, of programs have been written to solve problems commonly encountered by chemical or food engineers. Many of these have become available for purchase.

An engineer may write a programming language code to solve a particular design problem. Before the explosive growth of software, this was done commonly. There is still a place for it, when appropriate software is not available or as a learning technique.

Because program writing and debugging is usually quite time-consuming, it is recommended that available software be used and that a specific design program be written only after determining that appropriate software is not available.

Alternatively, spreadsheet software has become an indispensable tool for engineers, because of the availability of personal computers, ease of use, and adaptability to many types of problems. Spreadsheet software has achieved great popularity because of its availability for microcomputers at reasonable cost, the ease of learning and using the software, and its flexible applications to many problems. Solutions for many engineering problems can be obtained more rapidly by spreadsheet than by writing a language code program, such as FORTRAN.

Furthermore, general purpose spreadsheet software can be used effectively in process design. For example, Microsoft Excel with Visual Basic for Applications is an effective tool for process design. Spreadsheets offer sufficient process model "hospitality". They are connected easily and on-line with charts and graphic objects, resulting in powerful and easy-to-use graphical interfaces. Excel also supports mathematical and statistical tools. For instance, Solver is an excellent tool for solving sets of equations and performing optimization. Databases are effectively and easily accessed. In addition, Visual Basic for Applications offers a powerful object-oriented programming language, capable for constructing commercial graphics interfaces.

It is the main scope of this book to develop complete mathematical models and integrated design procedures for most common food processes. All models and methods presented in this book are appropriate for direct computer implementation. Furthermore, all the application examples are based on programming techniques on Excel. Thus, it is the aim of this chapter to present step-by-step procedures in order to apply the proposed models into the Excel environment. The chapter refers to two main topics. Firstly, the principles for solution of a process design problem are presented and then the principles for Excel implementation are described.

The reader is encouraged to become familiar with the following topics regarding Excel software, using the relative literature:
* Modeling and Spreadsheets
* Analyzing the Solver
* Sensitivity Analysis Using Excel Tables
* Controls and Dialog Boxes to Input Data
* Graphics to Get the Results
* Databases
* Visual Basic as a Programming Language

II. PRINCIPLES OF SPREADSHEET-AIDED PROCESS DESIGN

Computer-aided design is based on computer simulators and computer simulators are based on process modeling. The basic terms modeling, simulation and design are defined in Table 3.1. Modeling is the procedure of translating the physical laws of a process to mathematical equations in order to analyze or design the process. Simulation is the appropriate software, which predicts the real performance of a process. It is based on mathematical modeling plus the appropriate graphics interface in a computer environment. Design is a procedure of sizing and rating a process in order to obtain a specific goal, such as economic production, product quality, and protection of the environment.

Modeling and simulation are useful tools in process design. Table 3.2 summarizes a step-by-step procedure for process modeling, while Table 3.3 summarizes a step-by-step procedure for process simulation. These steps are further analyzed as follows:

Table 3.1 Basic Definitions

Modeling
is the procedure to translate the physical laws of a process
to mathematical equations

Simulation
is the appropriate software which guesses the real performance of a process

Design
is a procedure to **size** and **rate** a process in order to obtain a specific goal

Size
Given the process specifications
calculate the equipment size and characteristics

Rate
Given the process specifications and the equipment size and characteristics
calculate the operating conditions

Table 3.2 Process Modeling

1. Process model formulation
2. Degrees-of-freedom analysis
3. Alternative problems formulations
4. Problem-solution algorithm
5. Cost estimation and project evaluation analysis
6. Process optimization

Table 3.3 Process Simulation Procedure in a Spreadsheet Environment

1. Model development in a spreadsheet
2. Implementation of alternative problems solution and/or optimization procedures
3. Development of graphics interface

The equations of a model describe the physical laws which are valid for the process. They are derived from material and energy balances, thermodynamic equilibria, transport phenomena, geometry, equipment characteristics, and so forth.

A degrees-of-freedom analysis is shown in Table 3.4 and in Figure 3.1. Suppose that *M* variables are incorporated into the mathematical model of *N* equations. Generally, *M* is greater than or equal to *N*, and the difference *M-N* corresponds to the degrees of freedom of the process. The degrees of freedom is characteristic of the process. In process design, some variables have given values, due the design specifications, and the remainder correspond to design variables.

The number of design variables is characteristic of the problem. Several different problems could be defined for every process (see for example Table 3.5). The values for the design variables are decided by the design engineer. The remainder *NxN* set of equations is solved using mathematical techniques. In chemical and food engineering the resulting system is rare, that is every variable appears in a few equations. In that case the system can be solved sequentially (down triangle matrix) or by using a few trial variables.

Table 3.4 Degrees-of-Freedom Analysis

Total number of variables	M	
Total number of equations	N	
Degrees of freedom	$F=M-N$	Process characteristic
Degrees of freedom	F	
Problem specifications	K	
Design variables	$D=F-K$	Problem characteristic
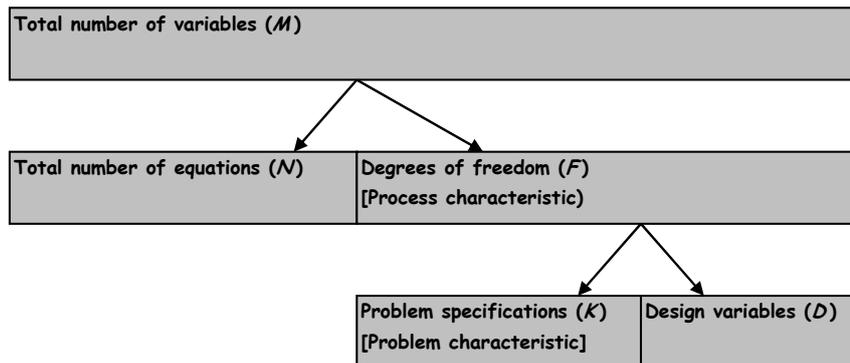
Figure 3.1 Degrees-of-freedom analysis.

Table 3.5 Some Typical Problems

Direct

Given	the characteristics of input streams
	the equipment characteristics
	the operating conditions
Calculate	the characteristics of the output streams

Design

Given	the characteristics of input streams
	the characteristics of output streams
Calculate	the equipment characteristics
	the operating conditions

Rating

Given	the characteristics of input streams
	the characteristics of output streams
	the equipment characteristics
Calculate	the operating conditions

Identification

Given	the characteristics of input streams
	the characteristics of output streams
	the operating conditions
Calculate	the equipment characteristics

The problem of selection of the design variables and the precedence order in which the equations are to be solved is usually based on the structural array. The structural array describes the structure of the process model. The columns of the structural array correspond to all the variables of the process and the rows to all the process equations. An "1" is placed wherever a variable appears in an equation and a "0" otherwise. See for example Table 3.6.

Several algorithms have been proposed in the literature for design variables selection. See for example the Lee-Christensen-Rudd algorithm presented in Table 3.7 (Rudd and Watson, 1968).

Figure 3.2 represents two solution algorithms for the system in Table 3.6. The second is based on the Lee-Christensen-Rudd algorithm.

The values of the design variables are selected so as to optimize an objective function, usually of economic origin as analyzed in Chapter 2. An information flow diagram for process optimization is shown in Figure 3.3.

Table 3.6 Example of a Simple Mathematical Model Along with the Corresponding Structural Array

Model equations

E1	$X_1 / X_2 = a$
E2	$X_1 + X_2 + X_4 = 0$
E3	$X_3 = \ln(X_2/X_4)$

Structural array

	X_1	X_2	X_3	X_4
E1	1	1		
E2	1	1		1
E3		1	1	1
Sum	2	3	1	2

Table 3.7 The Lee-Christensen-Rudd Algorithm for Design Variables Selection

1. Locate a column which contains only one "1" and delete the column and the corresponding row (equation).
2. Repeat step 1 until all equations have been eliminated
3. The remaining variables are the design variables
4. The precedence order of solving the remaining equations is the reverse of the order in which the equations were deleted
5. Should this algorithm not eliminate all equations, a recycle loop has been detected, which cannot be eliminated by the proper selection of the design variables, and certain of the equations must necessarily be solved simultaneously

(a) Solution with recycle loop   (b) Solution without recycle loop
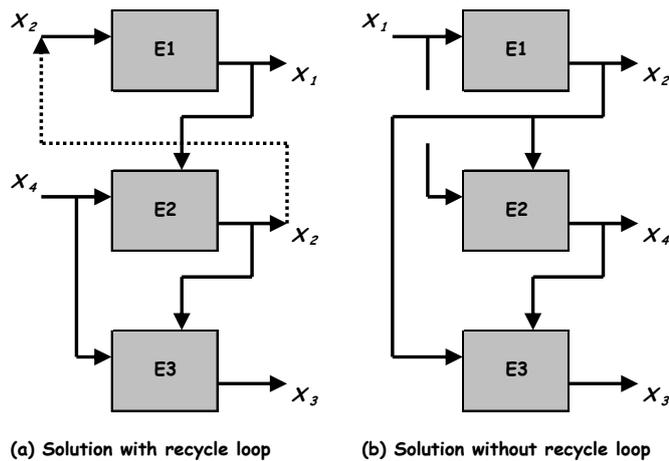
Figure 3.2 Solution algorithms for problem in Table 3.6.
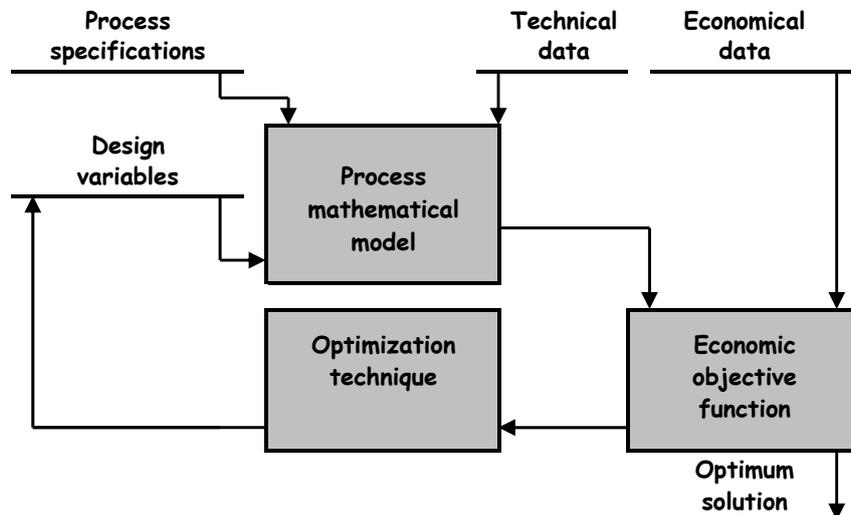
Figure 3.3 Information flow diagram for process optimization.

The above procedure is suitable for implementation in a spreadsheet environment. The resulting simulator has generally the outline presented in Figure 3.4. Four different units are distinguished, with each one developed in a different sheet (Maroulis and Kremalis, 1995).

"Process Model Worksheet" is the heart of the system calculations. It contains the process model. When no interactions are needed, the model solution uses only worksheet functions. In that case, when any change in input variables (free variables) occurs, the solution is obtained automatically on this worksheet.

Since the use of the simulator requires the solution of different problems, some different problems are formulated on the "Problem Solution Visual Basic Module". Their solution is based on the simplest problem of the process model worksheet above, and uses the Solver or the Goal Seek utilities of Excel via a Visual Basic program, to obtain a solution for the alternative problems.

All technical and required data are retrieved from the "Database worksheet", which contains all the required information in the form of "data lists". These data are extended and modified via appropriate dialog boxes.

"Graphics interface worksheet" is a user-friendly way for man-machine communication. It usually consists of three parts: (a) Problem specifications: The specifications and the required data for the problem to be solved are entered by the user or estimated from the databases. Data are inserted via dialog boxes or buttons for changing some important magnitudes. (b) Problem type selection: The type of problem to be solved is selected via buttons. (c) Results presentation: The results are obtained automatically, and are presented in the form of tables or charts. Since these charts are updated automatically, the user has at his disposal all the information needed for sizing, rating, sensitivity analysis, or comparison of alternative solutions.
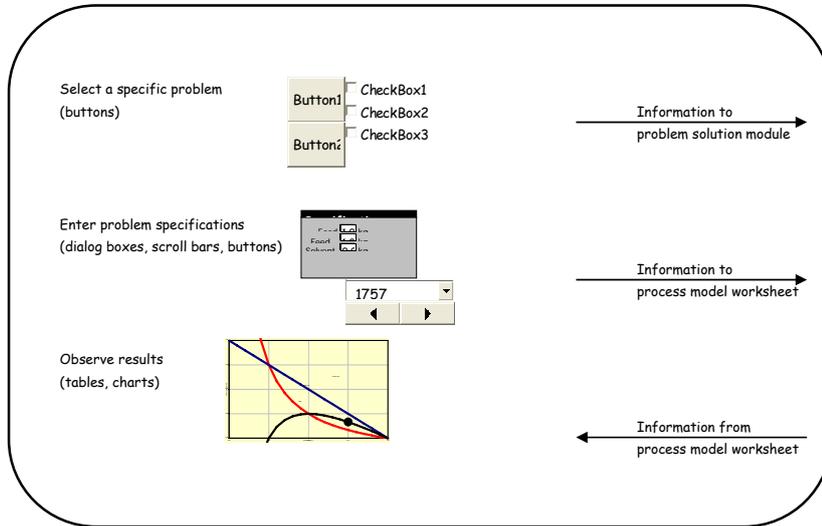
Graphics interface worksheet
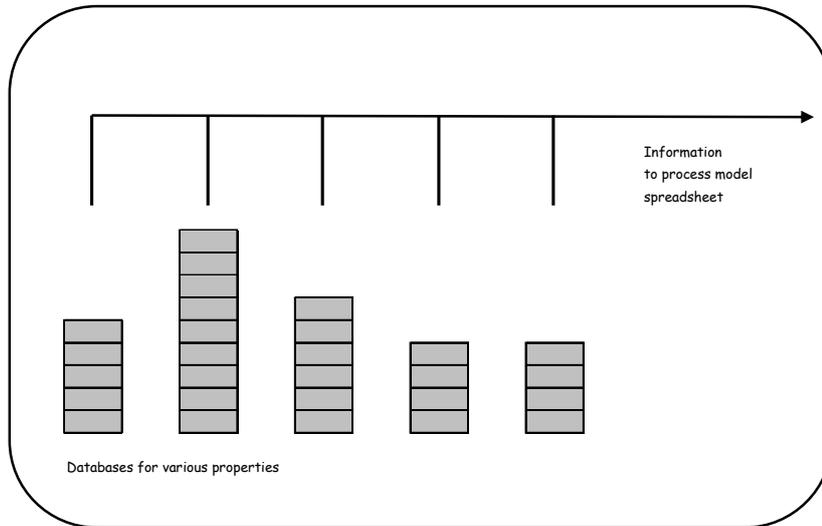
Database worksheet

Figure 3.4 Simulator architecture on a spreadsheet environment.
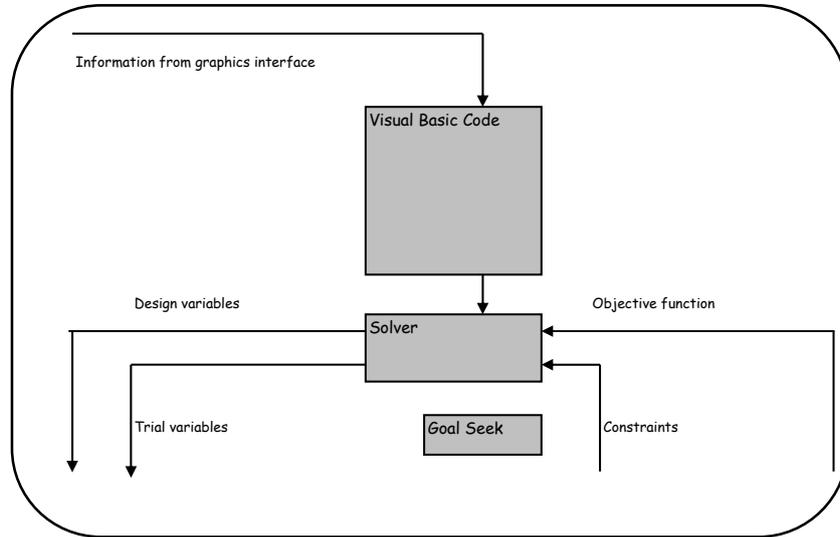
Problem solution visual basic model
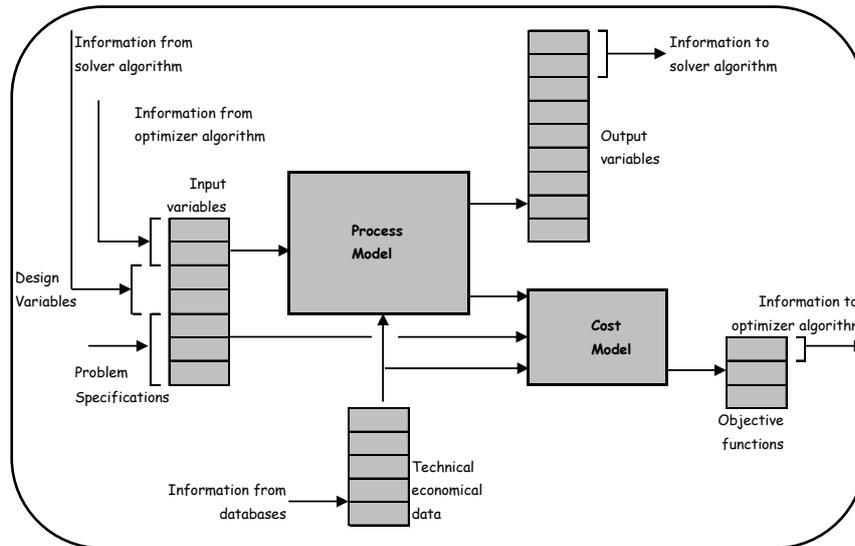
Process model worksheet

Figure 3.4 Continued.

III. APPLICATION EXAMPLE

A step-by-step Excel implementation of a simple design problem on a simplified liquid-liquid extraction system is presented in this section.

1. Process Description

It is proposed to recover a useful compound C from a water solution W by extraction with an organic solvent S (Figure 3.5). The extractor is a mixer-settler system, which uses a pump for mixing and a vessel for settling. The two phases (aqueous and organic) are considered immiscible and the compound C distributes itself between the aqueous and organic phases. The aqueous phase with a flow rate W (kgW/s) and concentration X_o (kgC/kgW) enters the system and exits at the same flow rate and a new composition X (kgC/kgW). The organic phase with a flow rate S (kgS/s) and concentration Y_o (kgC/kgS) enters the system and exits at the same flow rate and a new composition Y (kgC/kgS).

2. Process Model

A simplified mathematical model of the process presented in Figure 3.5 is summarized in Table 3.8. It includes 2 equations: The material balance for the compound C (Equation E1) and the equilibrium of the compound C between the aqueous and the organic phases (Equation E2). The profit P of the process is considered as the objective function for process optimization (Equation O1). The variables involved in the process model as well as the required technical and economical data are also presented in Table 3.8. Considering a typical design problem with three process specifications a degrees-of-freedom analysis suggests one design variable as shown in Table 3.8, and if we select the waste concentration X as design variable, a simple sequential solution algorithm is obtained as shown in Table 3.8. The optimum value for waste concentration X can be obtained by maximizing the profit P. Waste concentration must be varied in the range between zero and feed concentration X_o (process constraints in Table 3.8).
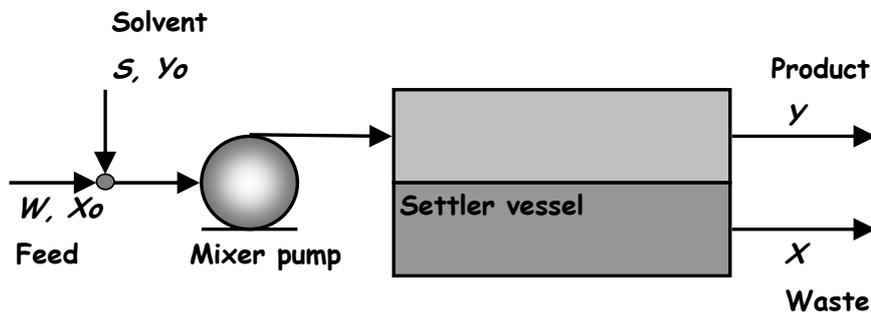
Figure 3.5 Diagram of a simplified liquid/liquid extractor.

Table 3.8 Process Model

Process Equations

$W (X_o - X) = S (Y - Y_o)$ (E1)

$Y = m X$ (E2)

Process Constraints

$X > 0$ (C1)

$X_o - X > 0$ (C2)

Objective Function

$P = Y S C_p - S C_s$ (O1)

Process Variables

W	kg/s	Feed flow rate	(V1)
S	kg/s	Solvent flow rate	(V2)
X_o	kg$C$/kg$W$	Feed concentration	(V3)
Y_o	kg$C$/kg$S$	Solvent concentration	(V4)
X	kg$C$/kg$W$	Waste concentration	(V5)
Y	kg$C$/kg$S$	Product concentration	(V6)

Technical Data

m	-	Equilibrium constant	(T1)

Cost Data

C_s	\$/kg$S$	Solvent	(R1)
C_p	\$/kg$C$	Compound	(R2)

Process Specifications

W	kg/s	Feed flow rate	(S1)
X_o	kg$C$/kg$W$	Feed concentration	(S2)
Y_o	kg$C$/kg$S$	Solvent concentration	(S3)

Degrees-of-Freedom Analysis

Process variables	6
Process equations	2
Degrees of freedom	4
Specifications	3
Design variables	1

Design Variables

X	kg$C$/kg$W$	Waste concentration	(D1)

Solution Algorithm

(E2) $\rightarrow Y$

(E1) $\rightarrow S$

(O1) $\rightarrow P$

---

3. Excel Implementation

The following steps comprise an Excel implementation procedure:
1. Workbook preparation
2. Process modeling in a spreadsheet
3. Using "Solver" for process optimization
4. Using graphs and tables for presentation of the results
5. Introducing dialog boxes and controls to modify data
6. Towards an integrated graphics interface

Step 1: Workbook Preparation

Create a new workbook and name it "Extraction.xls". Insert and name blank sheets as it is presented in Table 3.9.

Step 2: Process Modeling in a Spreadsheet

Into the spreadsheet "Process" consider seven separate ranges, as follows:

- Technical Data
- Design Variables
- Process Specifications
- Economic Data
- Process Model
- Process Constraints
- Economic Model

Each range consists of three columns and some rows, one row for every variable in the range. In each range the first column contains the variable names, the second the variable values or variable formulas, and the third the units used. Name all cells in second columns according the names in the first column. (You can use the "Ctrl Shift F3" option).

The ranges "Technical Data", "Design Variables", "Process Specifications" and "Economic Data" contain only data. Insert these data accordingly to Table 3.10. The Ranges "Process Model", "Process Constraints" and "Economic Model" contain formulas. Insert these formulas accordingly to Table 3.10.

Having inserted data and formulas, the process model implementation has been completed. The resulting spreadsheet "Process" looks like that presented in Figure 3.6. The cell ranges can be colored with different colors. The drawn arrows show the information flow in the spreadsheet.

The spreadsheet process model is ready for use. Any changes in process data, economic data, process specifications, design variables are taken into account and the results are updated immediately.

Any optimization technique, graphical or tabulated reports, any scenario analysis, or sensitivity analysis, any sophisticated graphics interface can be based on the "Process" spreadsheet. Some examples follow.

Step 3: Using "Solver" for Process Optimization

Create a Visual Basic subroutine with the name "optimum" in the "Optimize" module. The appropriate code is shown in Table 3.11.

Table 3.9 Sheets in "Extraction.xls" Workbook

Sheet Name	Purpose
Spreadsheets	
• Process	Process model
• Flowsheet	Process flowsheet
• Report	Summary report of results
• Control	Graphics interface
Visual Basic Modules	
• Optimize	Process optimization subroutines
• Controls	Subroutines for dialog boxes and controls
Dialog Box Sheets	
• Spec	Process specifications
• Tech	Technical data
• Cost	Economical data

Table 3.10 Cell Content in "Process" Spreadsheet

Cell name	Cell content	Cell name	Cell content
F	1.00	S	=F*(Xo-X)/(Y-Yo)
Xo	1.00	Y	=m*X
Yo	0.00	P	=Y*S*Cp-S*Cs
X	0.75	X	=X
m	4.00	Xo-X	=Xo-X
Cs	1.00		
Cp	1.00		
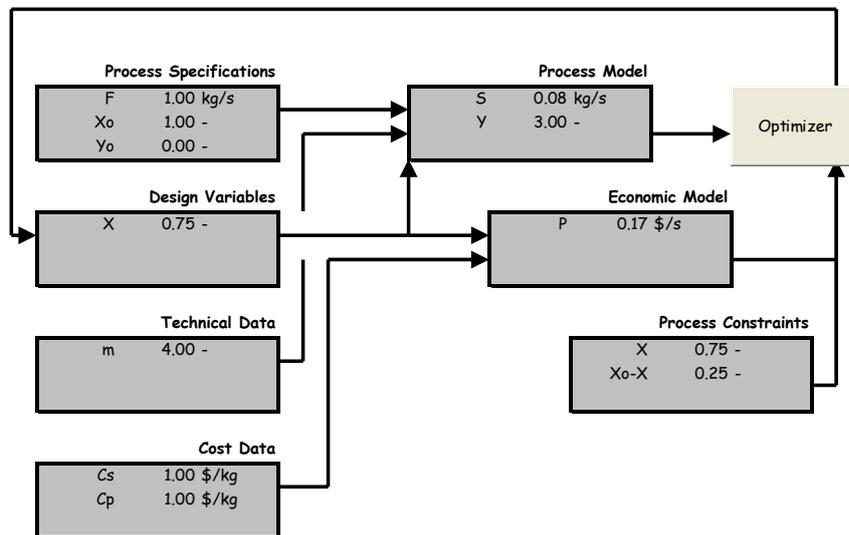
Figure 3.6 Model implementation in the "Process" spreadsheet.

Table 3.11 Visual Basic Subroutine for Process Optimization

```
  Sub optimum()
1 Sheets("Process").Activate
2 SolverReset
3 SolverOk SetCell:=Range("objective"), MaxMinVal:=1, ByChange:=Range("variables")
4 SolverAdd CellRef:=Range("constraints"), Relation:=3, FormulaText:=0#
5 SolverSolve UserFinish:=True
6 Beep
  End Sub
```

Statement 1 activates the "Process" spreadsheet. Statement 2 resets the Solver. Statement 3 selects the cell with the name "objective" to be the objective function [SetCell:=Range("objective")], requires the maximization of the objective function [MaxMinVal:=1], and selects the range "variables" to be the decision variables [ByChange:=Range("variables")]. Statement 4 suggests that all cells in the range "constraints" [CellRef:=Range("constraints")] must be greater than [Relation:=3] zero [FormulaText:=0#]. Statement 5 activates the solver to find the optimum.

The above mentioned cell names must be defined. Thus, in the sheet "Process" name:
• the cells which contain the values of the design variables as "variables"
• the cells which contain the process constraints as "constraints"
• the cell which contains the profit as "objective"

In the sheet "Process" insert a new button, name it "optimizer" and assign it to the subroutine "optimum".

Press the button "optimizer" and the optimum is reached in a few seconds as shown in Figure 3.7. A comparison between initial and optimum design can be made using Figures 3.6 and 3.7.
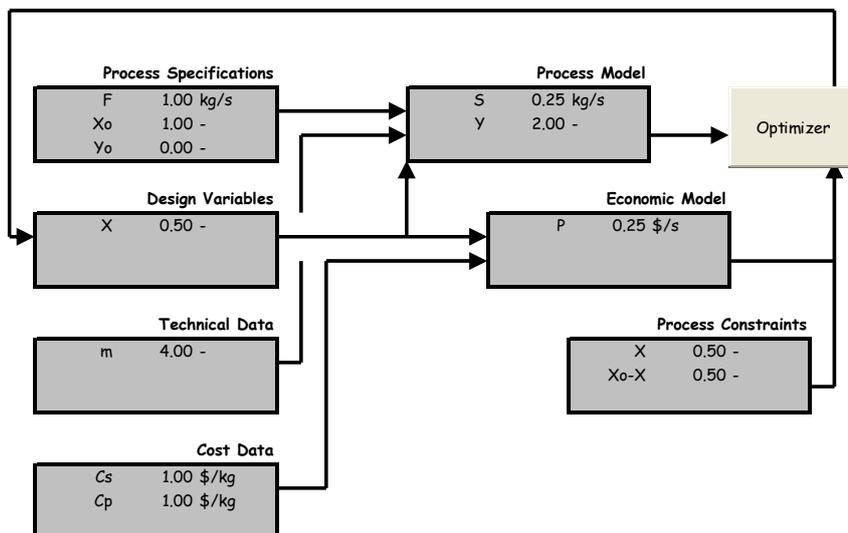
Figure 3.7 Comparison between initial and optimum design.

Step 4: Using Excel Tables and Charts for Presentation of the Results

The process design results can be further analyzed using the tools "Tables" and "Charts" supported by Excel.

For example, a process flowsheet can easily be constructed in Excel as follows: In the sheet "Flowsheet" draw a flowsheet like that presented in Figure 3.8 by using the drawing toolbar. Any information concerning process conditions can be inserted in cells near the desired point of the flowsheet. For each piece of information there need three cells, one for the variable name, one for the variable value, and one for the variable units. That is, in order to insert the feed characteristics select a cell near the icon of the feed arrow and insert the text "W=", in a neighboring cell insert the formula "=W" to get the value from the "Process" sheet, and in another cell, nearby, insert the text "kgW/s". You can add any information you like. See for example Figure 3.8. Any changes in data are updated immediately.

In order to plot the effect of the design variable on a technical and an economical variable the following steps can be used: Construct a one-dimensional Excel Table in which the "Column Input Cell" is the cell with the name "X". The second and third output columns refer to the cells "S" and "P", respectively. Next construct a "XY(Scatter)" Chart in which the first column of the Table corresponds to x-values and the second to y-values. Similarly, construct a second "XY(Scatter)" Chart in which the first column of the Table corresponds to x-values and the third to y-values. The results are presented in Figure 3.9, in which the optimum design is also shown.

Any other tabulated results or desired reports can be easily obtained as follows: Select a spreadsheet to incorporate the required information. Insert text or graphics as you like. Get the information from the "Process" sheet as described previously in the flowsheet construction procedure. See for example Figure 3.10, which is a summary report of the results obtained.
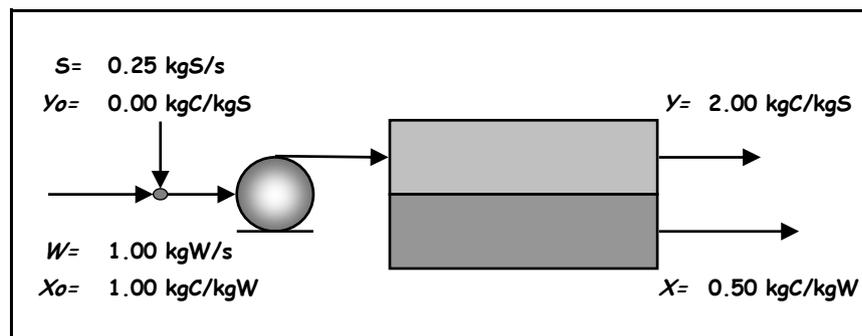
Figure 3.8 Process flowsheet implemented in the spreadsheet "Flowsheet".

X	S	P
0.50	0.25	0.25
0.00	250	-249
0.05	4.75	-3.80
0.10	2.25	-1.35
0.15	1.42	-0.57
0.20	1.00	-0.20
0.25	0.75	0.00
0.30	0.58	0.12
0.35	0.46	0.19
0.40	0.38	0.23
0.45	0.31	0.24
0.50	0.25	0.25
0.55	0.20	0.25
0.60	0.17	0.23
0.65	0.13	0.22
0.70	0.11	0.19
0.75	0.08	0.17
0.80	0.06	0.14
0.85	0.04	0.11
0.90	0.03	0.07
0.95	0.01	0.04
1.00	0.00	0.00
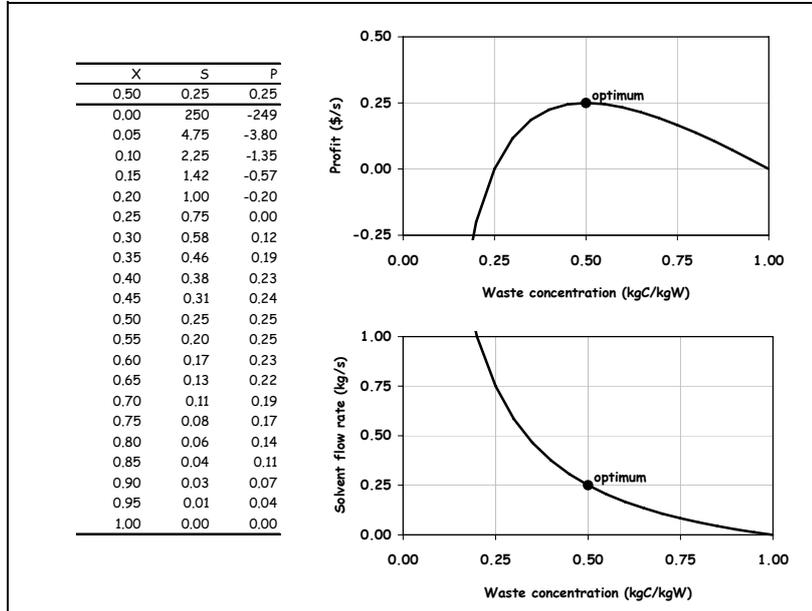
Figure 3.9 Analyze the effect of the design variable on a technical and an economic variable, using the "Tables" and "Charts" tools, supported by Excel.

Solute Recovery from a Rejected Aqueous Stream Using an Organic Solvent

Process
Liquid / liquid extraction

Technical data
Equilibrium constant   4.00 -

Cost data
Solvent cost   1.00 $/kg
Solute cost   1.00 $/kg

Specifications
Feed flow rate   1.00 kgW/s
Feed concentration   1.00 kgC/kgW
Solvent concentration   0.00 kgC/kgS

Design variable
Waste concentration   0.25 kgC/kgW

Process design results
Product concentration   1.00 kgC/kgS
Required solvent flow rate   0.75 kgS/s

Economics
Profit   0.00 $/s

Figure 3.10 A summary report of the results obtained.

Step 5: Introducing Dialog Boxes and Controls to Modify Data

A dialog box is used in order to modify the values of process specifications, which are included in the range "Process Specifications" in the spreadsheet "Process".

In the Dialog Module "db_spec" insert for every variable one "Label" (from the toolbar "forms") for its description, one "Edit Box" (from the toolbar "forms") for its value, and one "Label" for its units. Name all the Edit Boxes with the name of the corresponding variable.

In the Visual Basic Module "vb_controls" type a subroutine to use the dialog box in the sheet "d_spec", as described in Table 3.12.

In the spreadsheet "Process" insert a button, name it "specifications", and assign it to the subroutine "DialogSpecifications".

Press the button "specifications" and a dialog box appears in order to modify data for process specifications (see Figure 3.11).

A scroll bar is used for each design variable in order to modify the values of the design variables, which are included in the range "Design Variables" in the spreadsheet "Process".

A scrollbar, in order to handle the waste concentration *X,* can be inserted as follows:
• Insert the scroll bar icon from the toolbar "forms" as shown in Figure 3.11.
• Insert the minimum allowable value in a cell named "X.min".
• Insert the maximum allowable value in a cell named "X.max".
• Insert the coded value in a cell named "X.CV".
  The coded value ranges between 0 and 100 and is defined as follows:
  $X.CV = (X-X.min)/(X.max-Xmin)*100$
• Insert a scrollbar from the toolbar "forms" and assign the "Cell Link"
  (in the "Format Object" menu) to the coded value "X.CV".
• Replace the content of the cell named "X" with the following formula:
  $= X.min+X.CV*(X.max-X.min)/100$

It must be noted that the range "variables" which is handled by the solver during optimization must be redefined in order to refer to coded values, instead of the actual values. This modification guarantees the proper performance of the optimization and of scrollbars.

Table 3.12 A Subroutine to Activate the Dialog Box

```
Sub DialogSpecifications()
dbName = "d_spec"
DialogSheets(dbName).EditBoxes("W").Text = Range("W").Value
DialogSheets(dbName).EditBoxes("Xo").Text = Range("Xo").Value
DialogSheets(dbName).EditBoxes("Yo").Text = Range("Yo").Value
If DialogSheets(dbName).Show Then
Range("W").Value = DialogSheets(dbName).EditBoxes("W").Text
Range("Xo").Value = DialogSheets(dbName).EditBoxes("Xo").Text
Range("Yo").Value = DialogSheets(dbName).EditBoxes("Yo").Text
End If
Beep
End Sub
```
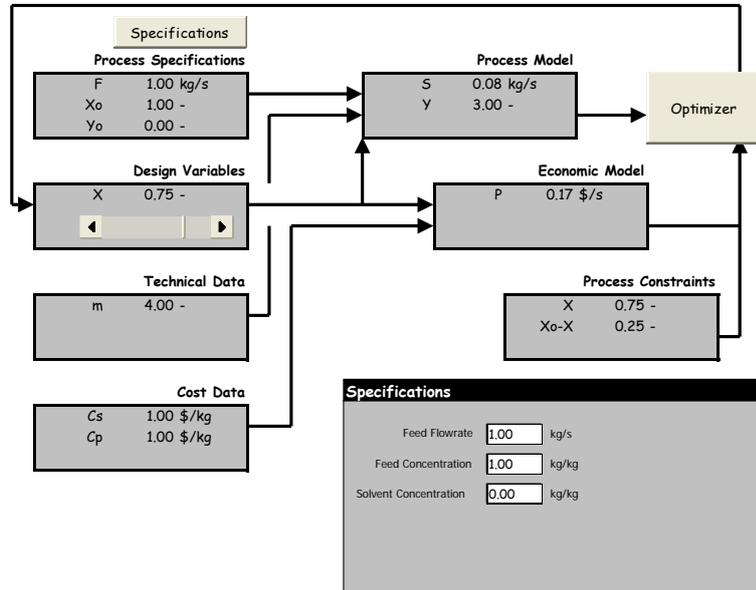
Figure 3.11 Process specifications dialog box.

Step 6: Towards an Integrated Graphics Interface

Any desired graphics interface can be developed in the spreadsheet "Control". A simple example, presented in Figure 3.12, can be constructed as follows:

- Draw a process flowsheet in sheet "Controls", as described in Step 5.
- Insert buttons to appear and disappear the crucial graphs
- Insert buttons to activate the desired dialog boxes
- Insert scroll bars to modify the desired process variables
- Insert buttons to solve different problems, e.g., process optimization.

A view of the resulting graphics interface with only one graph activated is shown in Figure 3.13. The user has now at his disposal a process simulator. He can enter data via scroll bars or dialog boxes and observe the results via buttons, which activate the desired graphs or reports.

The graphics interface could be further improved to look professional using appropriate programming code in Visual Basic.

Some integrated graphics interfaces are presented for various processes in Figures 3.14 through 3.17.
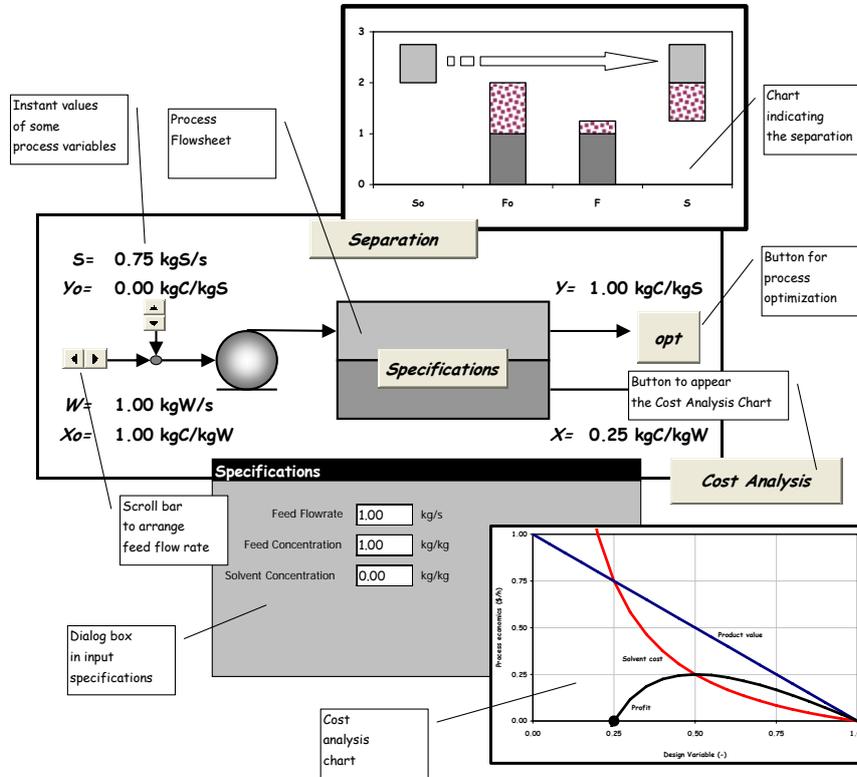
Figure 3.12 Comments on the graphics interface.
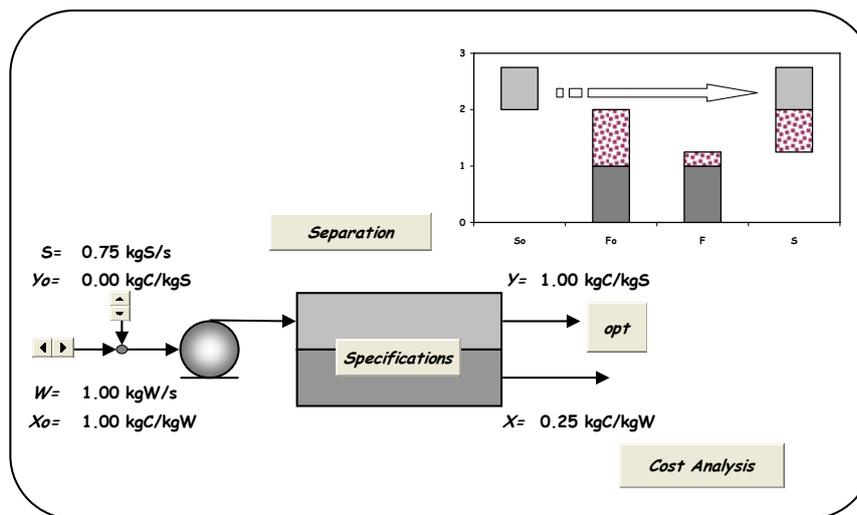
Figure 3.13 Graphics interface.
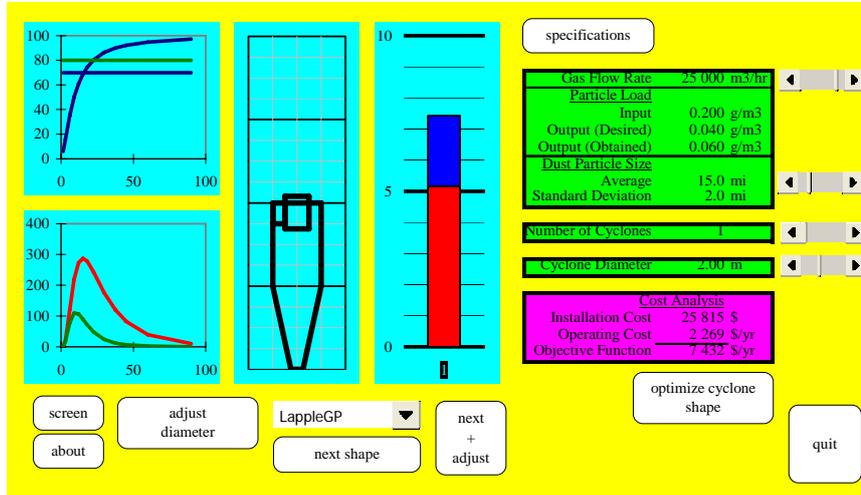
Figure 3.14 Cyclone simulator (Maroulis and Kremalis, 1995).
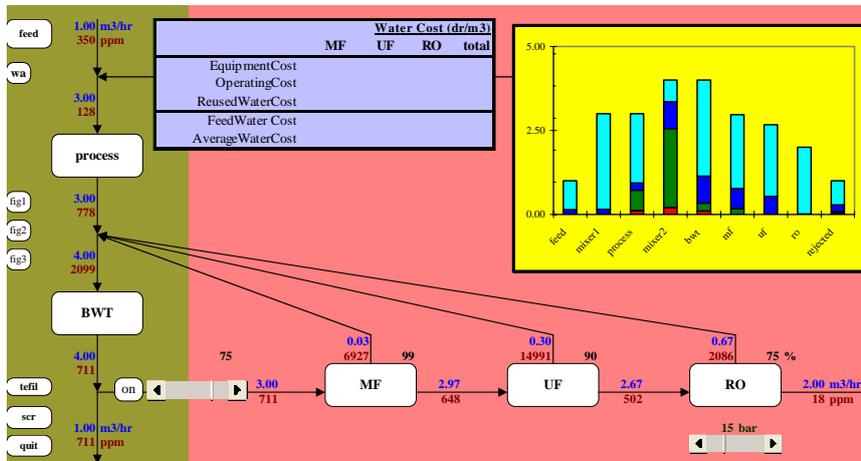
Figure 3.15 Combined UF/RO wastewater treatment system (Voros et al., 1999).
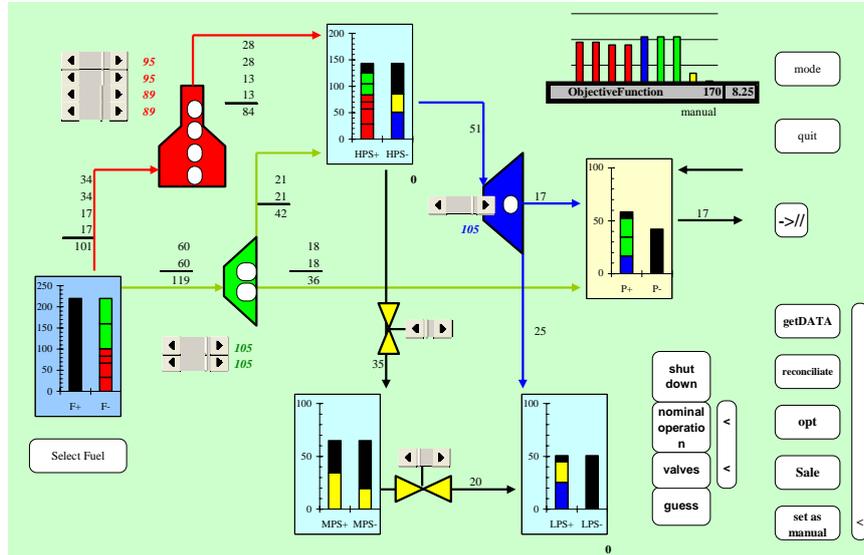
Figure 3.16 Utilities management system (Prokopakis and Maroulis, 1996).

NOMENCLATURE

C_p	\$/kg$C$	Compound cost
C_s	\$/kg$S$	Solvent cost
m	-	Equilibrium constant
P	\$/s	Profit
S	kg/s	Solvent flow rate
W	kg/s	Feed flow rate
X	kg$C$/kg$W$	Waste concentration
X_o	kg$C$/kg$W$	Feed concentration
Y	kg$C$/kg$S$	Product concentration
Y_o	kg$C$/kg$S$	Solvent concentration

REFERENCES

Maroulis ZB, Kremalis C, 1995. Development of an effective cyclone simulator under Excel. Filtration and Separation, 32(10) 969-976.

Peters MS, Timmerhaus KD, 1991. Plant Design and Economics for Chemical Engineers, 4th ed. McGraw-Hill, New York.

Prokopakis GJ, Maroulis ZB, 1996. Real-time management and optimization of industrial utilities system. Computers and Chemical Engineering, 20:623-628.

Rudd DF, Watson CC, 1968. Strategy of Process Engineering. Wiley, New York.

Voros NG, Fountoukides E, Magoulas KG, Maroulis ZB, Papadimitriou J, 1999. A combined UF/RO wastewater treatment system: Design, operation and economic assessment. Intl Desalination and Water Reuse Quarterly, 9(3), November/December 1999, 26-31.