

A NEW VISUAL PACKAGE FOR SIMULATION OF THERMAL DESALINATION PROCESSES: DEVELOPMENT & VERIFICATION

A. S. Nafey^{*}, H. E. S. Fath[‡], A. A. Mabrouk^{*}, M. A. Elzzeky[∇]

^{*} Engineering Science Dept., Faculty of Petroleum & Mining Engineering, Suez Canal University, Suez, Egypt, E-mail: asnafey31@ yahoo.com.

[‡] Mechanical Engineering Dept., Faculty of Engineering, Alexandria University.

[∇] Metallurgical and material Dept., Faculty of Petroleum & Mining Engineering, Suez Canal University, Egypt

Abstract

This paper describes a flexible visualized computer package developed for design and simulation of different types and configurations of thermal desalination processes. In this package, object oriented programming with Visual Basic (VB) is used to offer a friendly user-interface. The interface aids plants designer engineers to perform different calculations such as energy, exergy, and thermoeconomic. In addition, the package enables the designers to perform different modifications in existing plant or design new desalination configuration. The desalination plant components (units), such as heat exchangers, flash chambers, evaporators, pumps, pipes, etc. are stored as icons in a visual library. This visual library enables the user to construct different configurations by just clicking the mouse over the required units (icons). Each icon is linked with its representative mathematical model. The process mathematical model equations are represented in a sparse matrix form in order to minimize both memory storage and calculation time. The equations of each unit are grouped according to their variables type and each unit is mathematically represented by four matrices (mass, temperature, pressure and cost matrix). Robust subroutine code is developed to assemble the process mathematical model in four large matrices. The validity and reliability of the developed package is examined for different process and configurations under different operating conditions. The (5000 m³/day) MSF desalination plant (Euon Mousa, Egypt) is presented as a case study.

Keywords: desalination, object-oriented programming, sparse matrix, Process simulation.

1. Introduction

Desalination of saline (sea /brackish) water is one of the most promising technique to overcome water shortage problems in a considerable number of countries. Many desalination techniques have been proposed for utilities and industrial applications. This includes the thermal desalination processes (such as Multi Stage Flash (MSF), Multi Effect Distillation (MED) and vapor compression (VC)) and membrane desalination processes (such as reverse osmosis (RO) and Electro-Dialysis Reverse (EDR)). These processes consist of a number of interactive units. Using these units a wide range of process configurations can be obtained. To understand the behavior of these processes under different operating conditions, a general and flexible program is required.

A number of computer programs have been developed for desalination processes simulation, design and optimization. These computer programs are developed through out three stages. In the first stage, a special purpose program (one-off program) is used to solve the problem. The structure of this type of programs is simple and straightforward however, they are rigid such that each new process is considered as a new problem [5]. A large number of the published programs, for design and simulation of distillation processes are of this type, e.g. the developed programs by Hamed and Ali [2], Ithara and Stiel [3], Fath and Al-shabrawi [4]. To write and debug a program of this type for a large problem; enormous time is required and no minor flow sheet changes in normal configuration can be made. The return on investment for this type of programs is quite attractive only if the number of plants to be designed with exactly the same configuration is large; otherwise it tends to be time consuming.

In the second generation, the developed computer programs are nominated either general-purpose programs or modular programs (Flowsheeting approach) Nafey [1], [5]. These programs are developed to overcome the problems and limitations of the first generation. In these programs, the mathematical model is usually formulated in terms of a set of equations representing the unit processes. Each of these sets of equations is regarded as an independent and self-standing module. In the field of power generation plants a modular computer program was developed by Sonnenschein [6]. This program takes into account the varying in power demands and in operating conditions, as well as varying cycle configurations. A flexible computer program for thermodynamic power cycle calculations was also developed and described by Perz [7]. With this program, the designer can model different cycle schemes by selecting components from unseen library (under DOS) and connecting them in appropriate way. In the desalination field, a general Flowsheeting program was developed under DOS for design and simulation of thermal desalination processes by Nafey [1]. Calculations for different types and configurations of seawater distillation processes were presented. This type of programs needs expert users to describe the process topology and to enter the required data.

The third generation of the computer programming development is the visual modular program. These types of programs aid operators and designers to build up the process configuration and entering the required parameters easily. A visualized program was developed for power station plant by Woudstra *et al.* [8]. This program based on a strong library of thermal units. Different configurations of power plants can be considered by this program. A commercial process simulation tool ISPEpro was developed by Schausberger *et al.* [9] for studying the performance of a combined power and MSF desalination process. The user defines process flowsheets graphically by means of icons. Javier *et al.* [10] developed an oriented object program for the analysis of power and desalination plants. The software is developed in the form of Building Blocks for Water and Energy Systems (BBSWES) by using multi-platform (Java language). This software allows the user to build up a plant layout by dragging and dropping elements onto a blank. The program performs exergy and thermoeconomic analysis.

The main objective of the present work is to develop a modular computer program for Visual Synthesis of Desalination Plants (VSDP) by using visual basic language. Different calculations (energy, exergy and thermoeconomic) can be performed by the developed package. Different types of desalination processes with different configurations can be examined under different operating conditions. An executable file for each configuration can be generated by the developed program.

2. Numerical Approach

Sparse matrix: Desalination processes models encounter matrices; most of its elements are zero, so for efficient solution, the matrix sparsely should be exploited, Nafey [1]. In sparse matrix, only the value of the nonzero element is stored plus indexing information telling where each nonzero belongs in the regular array. For the irregular matrix structure, Compressed Sparse Row (CSR) is one of the most common storage formats, Sergio [11]. In this format, all the nonzero entries are stored row by row in a one dimensional real array A together with array JA containing their column indices and a pointer array IA which contains the addresses in A and JA of the beginning of each row, Sergio [11]. The Modified Sparse Row (MSR) format is a rather common variation of the CSR format which consists of keeping the main diagonal of the array A separately. Another scheme is the Coordinate storage scheme in which the nonzero entries of A are stored in any order together with their row and column indices. This scheme is used efficiently by Nafey [1] in the desalination flowsheeting processes owing to its simplicity. However, from the point of view of memory requirement, it is not as efficient as MSR format [12]. A basic tool kit for sparse matrix computation SPARSKIT is developed by Youcef [12]. This toolkit provides a collection of subroutines for modern supercomputer. A FORTRAN code for solving unsymmetrical sparse matrix was developed by Duff [13]. Watson Spars Matrix Package (WSMP) is developed recently by Gupta [14]. One of the difficulties in sparse matrix computation is the variety of types of matrices that are encountered in practical application. Flowsheeting matrices are not so well-defined either structurally or numerically. Therefore, a reliable and general subroutine for solving matrices produced by the desalination models is required. In the present work a visual basic subroutines are developed to solve the general sparse matrices of different flowsheets for desalination process.

VTBVT Technique: The sparsely and the weak nonlinearity of the model equations are among the characteristics of the thermal desalination processes mathematical models. Nafey [1] made use of the advantages of these characteristics by grouping the model equations according to the variable type. The author found that the Variable Type By Variable Type (VTBVT) technique can be easily programmed. This decomposition technique also, requires small memory, and requires only a few initial guessed values for solving different thermal desalination matrices. Using VTBVT technique with the advances in sparse matrix technology of both CSR and MSR storages, a visual package for desalination simulation is developed.

3. PROGRAM DESCRIPTION

The Integrated Development Environment (IDE) of Visual Basic (VB) is a collection of menus, tool bars, and forms window that form the programming workbench type, Reselman [15]. The form is the foundation of the building blocks of VB program and is a panel where the user (design engineer) can scheme any desalination process configuration. Toolbox is a palette of controls, where, the user can build the interface by selecting controls and placing them on the panel, see Fig (1). Using this powerful language, the user can construct any desalination process configuration at the design-time mode easily. Also, the user can change the operating conditions of the desalination plant during run-time. The desalination process configuration composes of collective icons (units and pipes) which should be seen mathematically as a big matrix

that assembled from sub matrices. The composing of different configurations means that the icon (real unit) position or type can change. Mathematically, this means that the sub matrix elements position will sequentially change. Therefore, the developed program has the following features:

- (i) It is a robust code modulates upon the unit position and type changes.
- (ii) It has a powerful graphic interface to build up different configurations easily.
- (iii) It is a reliable code because it solves the created sparse matrices for the required configuration.

3.1 Creation of the Graphic Interface

The interface is a panel where information between the user and code pass through. An intelligent interface is developed to take into consideration the unit position in the process configuration, the operating parameters conditions, the type of the required analysis, and the form of the output results. To describe the developed interface of the package, of Euon Mousa (Egypt) MSF desalination plant (5000 m³/day) is taken as a case study.

3.1.1 Process Unit Object Module

One of the advanced features of VB is the ability of creating and customizing a new ActiveX controls for use. At the heart of the ActiveX control, both the mathematical model of process unit and user object are developed. Using custom ActiveX control adds more flexibility to the package as the data are exchanged between the created control and the main form (panel) in two ways directions, Smith [16]. This interesting clue is necessary for the iterative solution of the nonlinear mathematical model. By using ActiveX control, each unit of desalination processes (Flash chamber, brine heater, evaporator, mixer, splitter, pump, pipe, setting unit, heat exchangers, etc.) is graphically and mathematically created. Due to space limitation, however, only the structure of the flash chamber unit module is illustrated in this paper. When the user click on Connect Button of the flash chamber unit of the heat recovery section, an interface window will appear as shown in Figure (2). This figure illustrates interaction dialog boxes. a, b, c, d and e. Figure (2.a) illustrates a simple sketch of the flash chamber with its junction streams according the plant configuration. Figure (2.c) window enables the user to enter or change the specified parameters such as length, diameters and numbers of the tubes of preheater / condenser, and the number of stages. The OK button is pressed to record the applied modifications if any. Figure (2.b) shows buttons by which, the user can call the required matrix (mass, temperature, pressure, exergy, cost). The mass button constituents the mathematical model of the mass balance equations for the flash chamber. The temperature Button constituents the energy balance equations. Similarly, the pressure, and cost buttons constituent their relevant mathematical model matrices. The matrix nonzero elements, are stored in CSR format in the lists arrays **IA**, **JA**, **A**, and the constant vector is stored in **CN** array as shown in Figure (2.d). These matrices are initiated when their subroutines are called by the main executive program or when their representative button of Fig. (2.b) is clicked by the user. The developed icon of the thermal unit becomes available for use anywhere with any number. Once the user drag the icon from the library and drop it on the panel, the real view of the unit will appear as shown in Fig. (1).

3.1.2 MSF process flowsheeting

The Euon Mousa MSF desalination plant of is described as an example of plant simulation. Figure (1) shows the plant process flowsheet. To construct a flowsheet of Euon Mousa MSF desalination plant, the designer needs to drag the required units from visual library. Then these icons (units) are visually arranged similar to the real plant. A menu bar is created and located at the top of a panel which associated with drop-down sub-menus. The developed menu bar constituents the following submenus:

- (1) Calculations Mode menu: this menu is associated with a drop-down sub-menu of (simulation, design, and optimization).
- (2) Connections menu: this menu is associated with buttons that enable the user to connect all units, and plant pipes.
- (3) Start Calculations menu: this menu enables the user to select the type of thermo dynamical calculations such as energy, exergy, cost, and thermoeconomic analysis.
- (4) Display Results menu: this menu allows the user to display the results in the same panel or another window in both table and chart.
- (5) Print menu: to print the oriented form.

After all units and pipes are connected with each others the scenario between the user and the computer is stopped to start the numerical calculations phase which occurs at the behind of the panel.

3.2 Numerical Calculations

To explain what occurs behind the interface, the flash chamber unit is taken as an example. The flash chamber unit is mathematically seen as 6 streams by which the unit is connected to other units; three streams for inputs and three for outputs. These streams hold and transport the properties of mass, temperature, pressure, exergy and cost from the source unit to destination unit. Once the Connect Button of each unit is clicked by either the user or automatically, three actions will occur at the behind of the panel, (1) A serial number (unit index) is initiated and stored for each unit, (2) The number of block-column which represent the stream variables of each unit is determined from the mathematical model, (3) The junctions (nodes) of the pipe connection will appear as shown in Fig. (2.a). This figure shows 6 nodes for 6 streams of the flash chamber unit. Nodes (1) and (2) represent the inlet and outlet brine water from the flash chamber. Nodes (3) and (4) represent the inlet and outlet distilled water. Nodes (6) and (5) represent the cooling sea water inlet and outlet from the unit. Based on the mathematical model of flash chamber unit, mass matrix for each stage needs 10 block-columns for stream variables as shown in matrix (1). It should be noted here that each one of the nodes (1), (2), (5) and (6) are represented by two block-columns to consider the water and salt components as shown in matrix (1). For this particular mass matrix (single stage), 10-component variables are required and only five equations are generated by the mathematical model. Evidently another five equations (block-rows) should be specified to obtain a square matrix. Array of the nonzero-element (A) contains few nonzero elements (represented by x symbol) as shown in matrix 1. To take advantage of this situation, spars matrix technique is used for efficient solution. Once the Connect Button is clicked, these 10 block-columns per stage are stacked and reserved in the main matrix.

Matrix (1): coefficient element matrix of the mass balance of the flash chamber.

		Node (1)		Node (2)		Node (3)	Node (4)	Node (5)		Node (5)			
<i>I</i>	<i>J</i>	<i>J+1</i>	<i>J+2</i>	<i>J+3</i>	<i>J+4</i>	<i>J+5</i>	<i>J+6</i>	<i>J+7</i>	<i>J+8</i>	<i>J+9</i>	<i>J+10</i>	...	<i>N</i>
		x	0	x	0	0	0	0	0	0	0		
		0	x	0	x	0	0	0	0	0	0		
		x	0	x	0	x	x	0	0	0	0		
		0	0	0	0	0	0	x	0	x	0		
		0	0	0	0	0	0	0	x	0	x		

Each stream is denoted by a node junction. Node (2) represents the brine outlet from the flash chamber unit and this node takes the $J+3$ and $J+4$ columns for water and salt components. For pipe connection, the node is defined by two independent variables, *nodetype*, *flowtype* respectively. The *nodetype* variable is denoted either sink for inlet or source for the outlet stream. The *nodetype* takes a number referred to the unit serial number (unit index). If the serial number of the flash chamber unit is 5, the *nodetype* of the outlet brine water stream will be (52) i.e. source = 52 and this number changes when the serial number of the unit changes. The *flowtype* is coded by 0 and 1 for pure water and salt water respectively. Then the node junction is assigned by its column number in the main matrix; i.e. node (52, 0) = $3+J$, and node (52, 1) = $4+J$, for water and salt components respectively.

3.3 The Calculations code

A robust code is developed in order to enable the user selecting of calculation types by just clicking the mouse without imposing himself to know how these calculations are achieved or how the numerical solution is accomplished. The main code consists of seven subroutines which are described as follows:

1. ALLOCATION: this subroutine is developed to group the model equations according to their variable type (VTBVT). Each group constitutes a matrix such as mass, temperature, pressure and cost. When ALLOCATION subroutine is asked to fill the mass matrix, for example, it will eventually, call all units for their mass matrices and stack them in main mass matrix. The main matrix elements are stored in Compact Sparse Row (CSR) format. After this subroutine is executed, In the case of 5000 m³/day MSF desalination plants, the results of Table (1) are obtained. The four matrices are not filled at the same time but sequentially in four cases; case 1 is selected to fill mass matrix. After the mass matrix is solved, case 2 is selected to fill temperature matrix to be solved. And so on for case 3 and case 4 for pressure and cost matrices respectively.

Table (1): Results of the ALLOCATION subroutine

Matrix type	Mass	Temperature	pressure	Cost
No. generated equation from the units	130	79	79	79
No. generated equation from the pipes	40	25	25	25
Total number of the equation	170	104	104	104
Total variables	170	104	104	104
No. of non zero elements	383	306	181	330
Row \otimes Column	170 \otimes 170	104 \otimes 104	104 \otimes 104	104 \otimes 104

2. **ARRANGE:** This subroutine is developed by Duff [18] to reorder the matrix so that a zero-free diagonal (maximum transversal) is obtained.
3. **ASCENARRANGE:** This subroutine is developed to sort the element in the same row in such a way that their column positions being in an increasing order.
4. **GETRIDZERO:** This subroutine is developed to get rid of zero element entry.
5. **DIAGELEMNT:** This subroutine is developed to subtract the elements of $A_{i=j}$ from the arranged matrix and fill it in the diagonal separated array (D). i.e. convert the matrix CSR format into MSR format.
6. **SIMSOLVE:** This subroutine is developed to solve any general sparse linear equations of the type $A*X = B$ based on the direct method of Gaussian elimination. It consists of seven subroutines calling each others to eliminate the lower elements of the diagonal. Because of the nonlinearity of the mathematical model, the equations are solved iteratively. Initial guess for all stream temperatures are provided to solve the mass matrix firstly. Then the temperature matrix is solved to obtain a new streams temperature, then pressure matrix is solved sequentially to give the pressure distribution for the streams. A certain tolerance between the initial and new temperatures is specified until the convergence solution is obtained. After the exergy points are calculated the cost matrix is solved.
7. **REALLOCATION:** this subroutine is developed to reallocate the value of x vector to corresponding stream variable after the oriented matrix is solved.

4. PACKAGE VERIFICATION

The package calculations are performed under the off-design specifications, of Eoun Mousa MSF desalination plant data in Table (2) and Table (3) [17]. To display the numerical output of each stream in the plant, the operating engineer should click on *Display Results* button. The output results of the package, are displayed in both tables and charts forms. Some of the output results are shown in the same panel as shown in Fig. (3). Figure (4) shows the mass of distilled water, cooling water and brine water streams for each stage. Similarly both the temperature and pressure of each stream are displayed as shown in Fig. (5) and Fig. (6). Figure (6) shows the jump in the pressure distribution of cooling water which represents the pressure head of the recirculated brine pump.

Table (2): Specifications for performance calculations of 5000 m³/day MSF desalination plant (Euon Mousa, Egypt)

Specified parameters	Brine heater	Heat recovery	Heat rejection
Heat transfer area, m ²	780	17*488	3*357
Tubes outside diameter, m	0.02199	0.01299	0.0239
Fouling factor, (kJ/m ² .hr.K) ⁻¹	0.00002	0.00002	0.00002
Recycle brine velocity, m/s	2	2	2

Table (3): Typical operating conditions of 5000 m³/day MSF desalination plant (Euon Mousa, Egypt)

Specified variables	value
Sea water flow rate, m ³ /hr	1570
Sea water temperature, °C	27
T. D. S. in sea water, ppm	48620
Heating steam temperature, °C	205
Heating steam pressure, bar	7
TBT	110
Make up flow rate, m ³ /hr	660
Cooling water splitter ratio	0.482
Reject brine splitter ratio	0.724

The results validity and accuracy of the program calculations are checked by comparing the calculated values of mass and temperature with the values of the real operating plant [17]. It is indicated from this comparison that the average error does not exceed 0.23 % and 1.18 % for both mass and temperature calculation respectively. Calculated plant performance parameters are compared with the design parameters as illustrated in Table (4). This comparison shows a satisfactory accuracy for the developed package. Due to the space limitation, the detailed analysis of the MSF process using thermoeconomic evaluation will be presented in a separate paper.

Table (4): Comparison between off-design and calculated performance parameters

	Off-Design	Calculated	% Error
Performance ratio	8	8.03	0.4
Production of distilled water, m ³ /day	5000	4995.6	-0.09

5. CONCLUSION

Visual Synthesis Desalination computer Program (VSDP) is developed for the design and simulation of different types and configurations of thermal desalination plants. Different types of calculations can be performed by this package; including material, energy, exergy analysis. The validity and reliability of the package results are satisfactory investigated under the off-design values of 5000 m³/day of the MSF desalination plant of Euon Mousa (Egypt).

REFERENCES

- [1] Nafey A. S. Design and Simulation of Seawater Thermal Desalination Plants. Ph. D., Leeds University, 1988.
- [2] Hamed O. A. and S. Aly, Simulation and Design of MSF Desalination Process, *Desalination*, 80, pp. 1-14, 1991.
- [3] Ithara S. and L. I. Stiel, The Optimal Design of Multi_Stage Flash Evaporators by Dynamic Programming, *Desalination*, 4, pp. 248-257, 1968.
- [4] Fath H. E. S., A. Al-shabrawi, Development of a Computer Package for the Process Design of MSF Desalination Unit, Fifth International Water Technology Conference, Alexandria, Egypt, pp. 373-393, 2000.
- [5] Nafey A. S., Simulation of Solar Heating Systems, Suez Canal University, Egypt, 2000.
- [6] Sonnenschein H., A Modular Optimizing Calculation Method of Power Station Energy Balance and Plant Efficiency, *Journal of Engineering for Power*, Vol. 104/255, 1982.
- [7] Perz E., A Computer Method for Thermal Power Cycle Calculation, *Transaction of the ASME*, Vol. 113, pp. 184-189, April 1991.
- [8] Woudstra N., M. Verschoor, CYCLE_TEMPO, Delft University of Technology, 1995.
- [9] Schausberger P., G. Rheina-Wolbeck, A. Friedl, M. Harasek, E. W. Perz, Enhancement of an Object-Oriented Power Plant Simulator by Seawater Desalination Topics, *Desalination*, Vol. 156, pp. 335-360, 2003.
- [10] Uche J., L. Serra, L. Alberto, A. Valero, J. Turrigano, C. Torres, Software for the Analysis of Water and Energy Systems, *Desalination*, Vol. 156, pp. 367-378, 2003.
- [11] Sergio Pissanetzky, Sparse Matrix Technology, Academic Press, Inc., 1984.
- [12] Youcef Saad, SPARSKIT: a basic tool kit for sparse matrix computations, University of Minnesota, 1994.
- [13] Duff I. S., MA28: Fortran Code for direct solution of sparse unsymmetric linear systems of equations. HSL Archive, Internet site, 2003.
- [14] Gupta A, WSMP: Watson Sparse Matrix Package, Part-II- direct solution of general sparse systems, IBM Research Report, RC 21888 (9847), 2000.
- [15] Reselman B., R. Peasley, and W. Pruchniak, Using Visual Basic 6, Copyright by Que, 1998.
- [16] Smith E., F. Weshlior, H. Markiz, Visual Basic 6, Bible, Dar El Farouk, Egypt, 1999.
- [17] Babcock - Hitachi, Operation and Maintenance Manual, Eoun Mousa Desalination Plant, Siena, Egypt, July 1998.
- [18] Duff I. S., Fortran Code for Permutations for a Zero-free Diagonal, *ACM Transaction on Mathematical Software*, 1981.

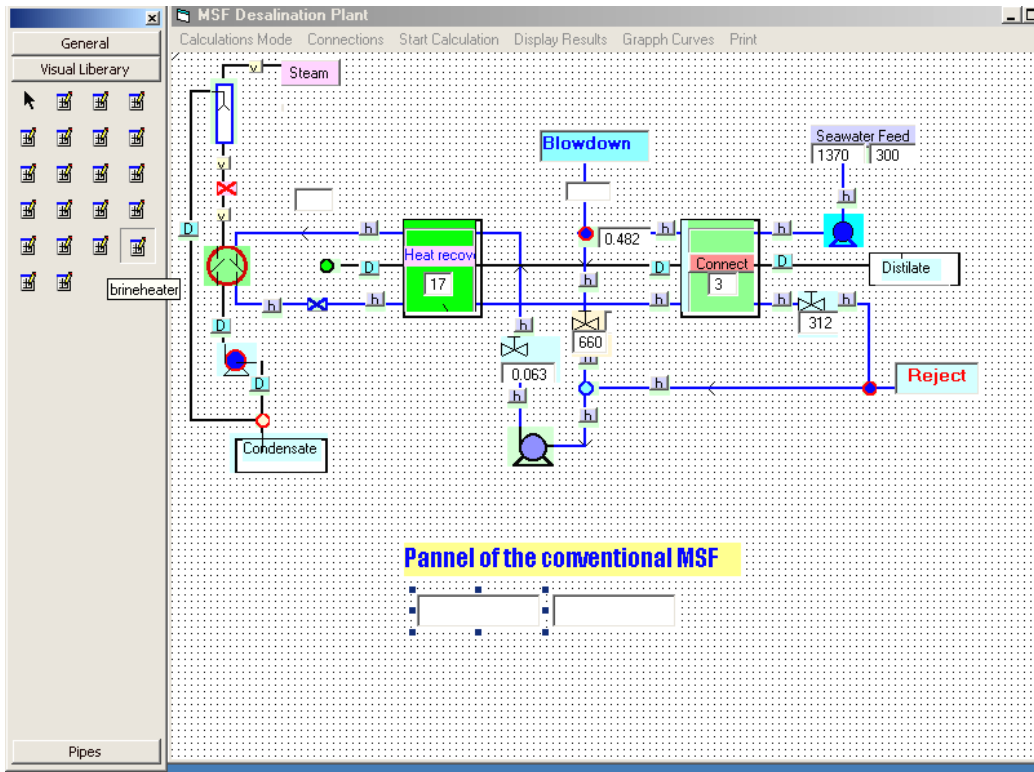


Fig. 1: Interface of MSF desalination plant (Eoun Mousa, Egypt)

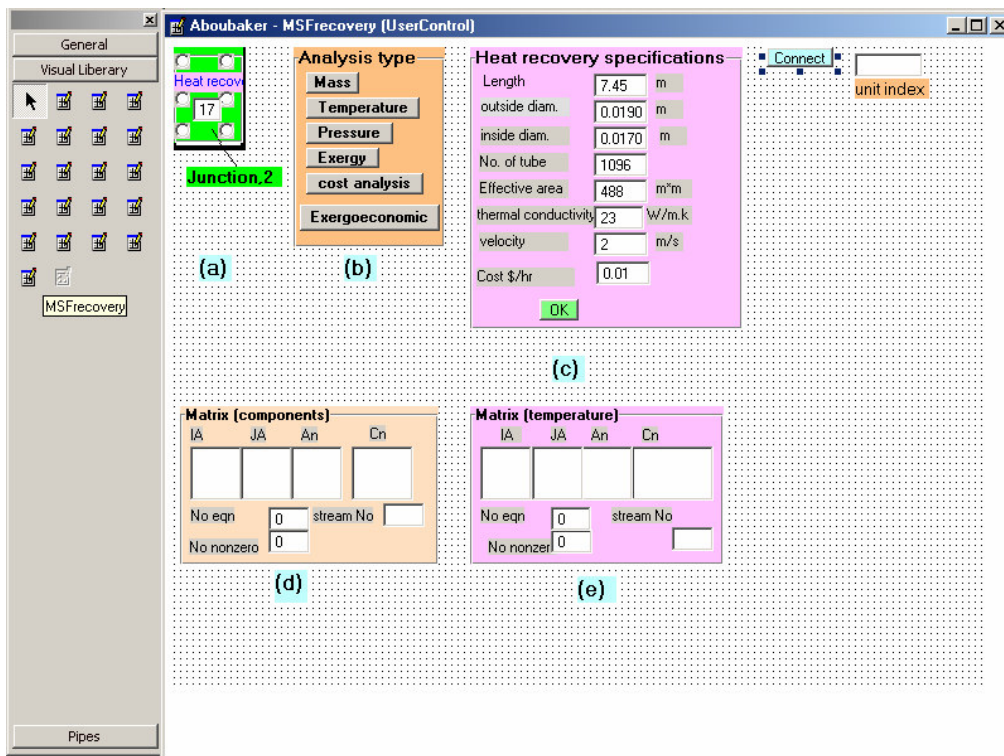


Fig. 2: Interface of flash chamber unit

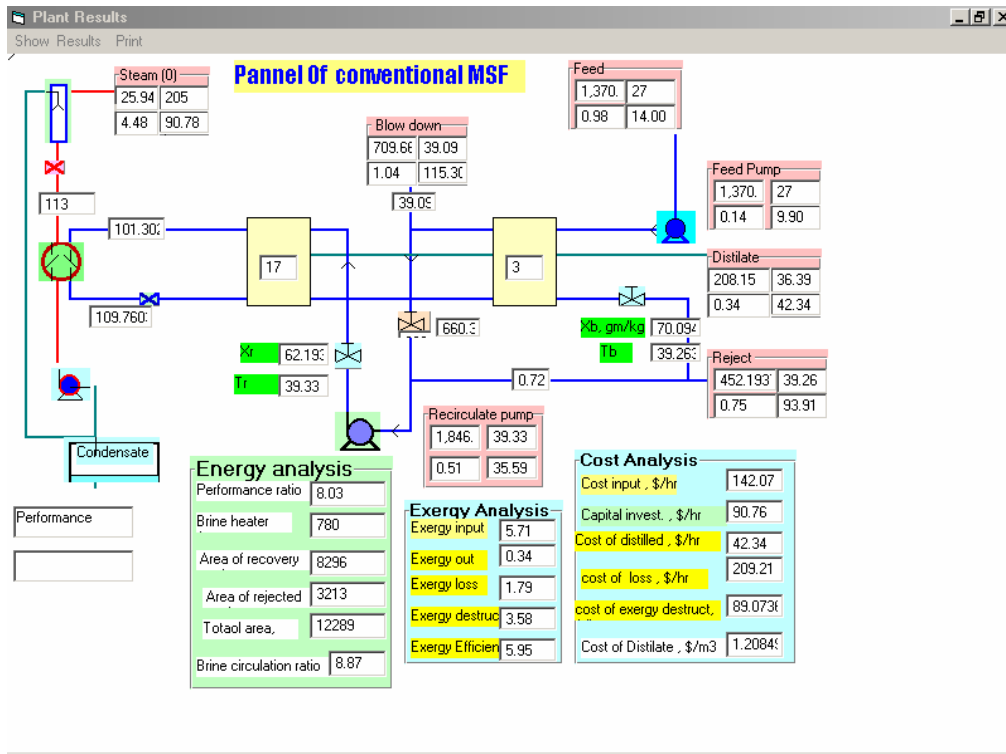


Fig. 3: Performance results of 5000 m³/d desalination plant

Display The Results

stages no.	1	2	3	4	5	6	7	8	106
Distilled (12.63	25.06	37.29	49.31	61.11	72.68	84.02	95.13	106
Cooling w _i	1,846.53	1,846.53	1,846.53	1,846.53	1,846.53	1,846.53	1,846.53	1,846.53	1,846
Brine wate	1,833.90	1,821.46	1,809.23	1,797.22	1,785.42	1,773.85	1,762.50	1,751.40	1,740

MASS BALANCE

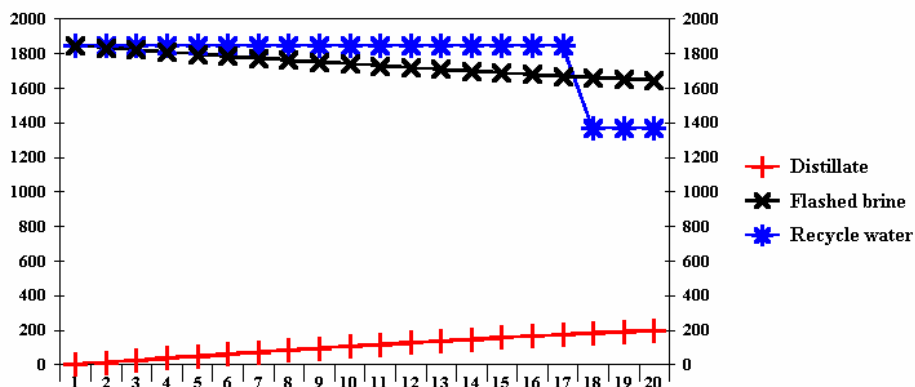


Fig. 4: Mass balance flow rate, kg/hr for 5000 m³/d MSF desalination plant

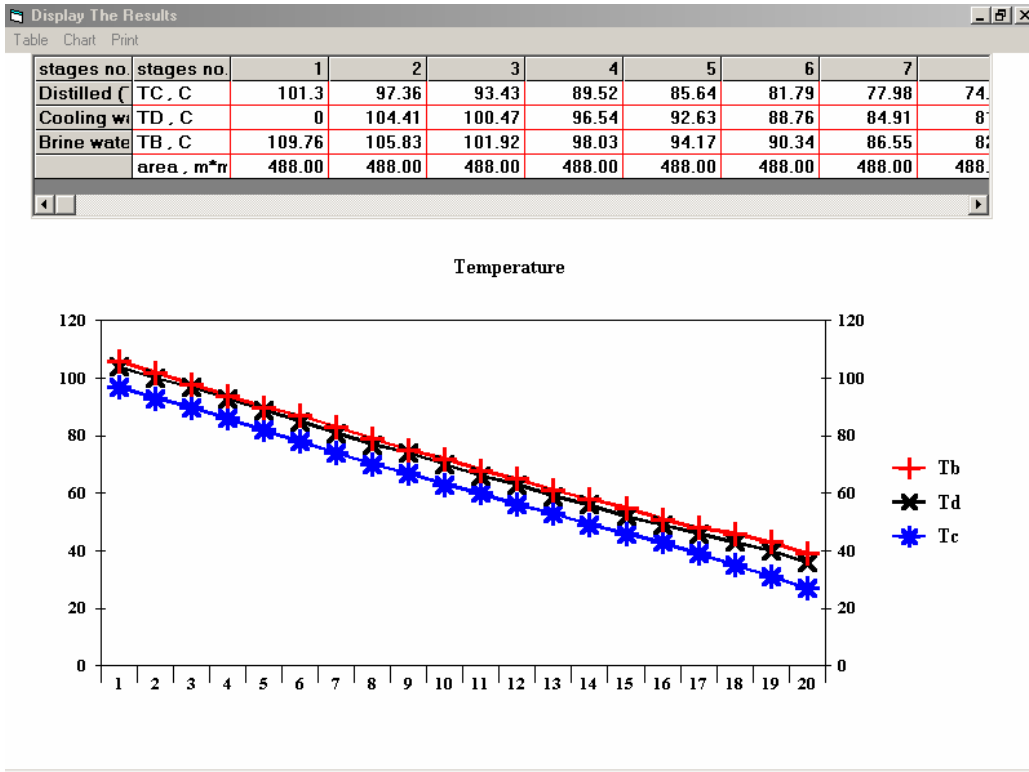


Fig. 5: Streams temperature of 5000 m³/d MSF desalination plant

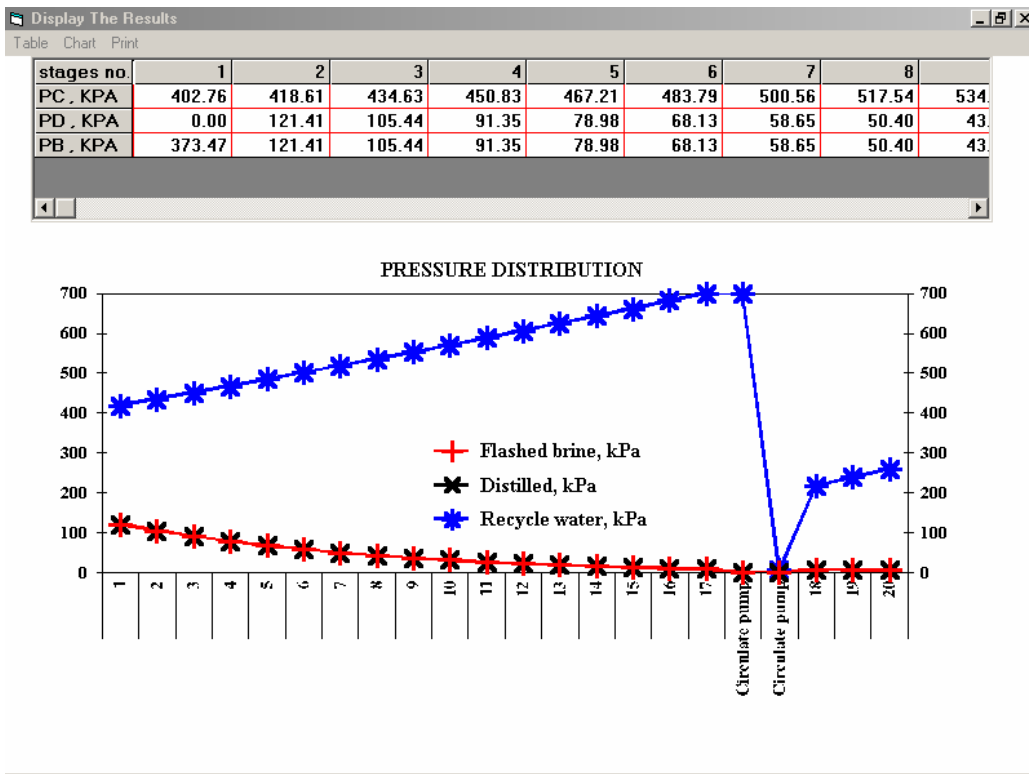


Fig. 6: streams pressure distribution of 5000 m³/d MSF desalination plant