Murdoch
UNIVERSITY

MURDOCH RESEARCH REPOSITORY

El-Harbawi, M., Acellam, D. and Yin, C-Y (2011) *Development of educational software for designing shell-and-tube heat exchangers.* International Journal of Mechanical Engineering Education, 39 (4). pp. 291-296.

Development of Educational Software for Designing Shell and Tube Heat Exchangers

Mohanad El-Harbawi, David Acellam

Department of Chemical Engineering, Universiti Teknologi PETRONAS, 31750, Tronoh, Perak, Malaysia. Email: mohanad_elharbawi@petronas.com.my

Chun-Yang Yin

School of Chemical and Mathematical Sciences, Faculty of Minerals and Energy, Murdoch University, Murdoch, WA 6150, Australia. Email: c.yin@murdoch.edu.au yinyang@streamyx.com

Abstract: A framework for developing an educational software using *Visual Basic* (VB) programming language for designing shell and tube heat exchangers (based on Kern and Bell-Delaware methods) is presented in this study. Simulation values from the software have been validated using manual calculations and compared with another related software (HEXTRAN).

Keywords: shell and tube heat exchanger; *Visual Basic*; educational software, simulation

1. Introduction

Heat exchangers are devices used for effective transfer of heat energy from one or more fluids to another across a solid surface, usually for both cooling and heating large/small scale industrial processes. Globally, they are extensively used in numerous industries, namely, petrochemical, power generation and food processing. Industrial heat exchangers, in essence, are categorized in accordance to various parameters including type of transfer process, size, flow configurations and arrangements, pass arrangements and heat-transfer mechanisms. Examples of these heat exchangers include shell and tube, compact, double pipe and plate.

In many process-based industries, shell and tube heat exchangers are used in great numbers, far more than any other type of exchanger. More than 90% of heat exchangers used in industry are of the shell and tube type [1] and they are preferred due to their robustness and capacity to handle high-pressure processes. Due to the various applications of heat exchangers in industries, many university engineering programs (especially chemical and mechanical engineering) provide courses such as *process heat transfer* to help students understand the design and analysis concepts of heat exchangers, including rating and sizing of the system component for specific applications. The established process of rating and sizing the components in a heat exchanger involves tedious and extensive routine calculations that are time-consuming and prone to human error [2]. As such, computer-aided heat exchanger design elements are typically incorporated in engineering courses in order to facilitate the undergraduates' understanding of the design while increasing the speed of design parameter simulations.

Several research groups have developed educational softwares for heat exchanger designs. Leong and co-researchers [3] developed a software for the thermal and hydraulic design of shell and tube heat exchangers with flow-induced vibration checks using *Windows*-based Delphi programming environment. Lona and co-researchers [4] developed an educational software called *Heat Exchanger Simulator* (HES) for use in chemical engineering undergraduate classes. Tan and Fok [2] designed and developed an educational computer-aided tool for designing heat exchangers that integrates thermo-hydraulics analysis with mechanical design using Java programming language. The software focuses on the shell and tube heat exchanger that aims to complement the theories behind the thermo-hydraulics design analysis with practical mechanical design details required for costing and production. Recently, Cartaxo and Fernandes [5] developed a *Windows*-based program used to introduce heat exchanger equipments to undergraduate students and provide unit operation courses with realistic exercises involving simulation of chemical processes.

This paper describes a framework for developing a standalone and user-friendly software package for designing shell and tube heat exchangers using *Visual Basic*. This software, called *Heat Exchanger Design and Analysis Simulation Software* (HEDASS), is validated via comparison of its design and analysis results with benchmark problems from established heat exchanger design text books and another simulation software.

2. Development and Description Of Software

HEDASS addresses two main heat exchanger issues, namely, rating issue in which one must determine whether a fully given specified exchanger performs a given heat-transfer duty satisfactorily or otherwise and the design issue in which one must determine the specifications for a heat exchanger that handles a given heat-transfer duty [6]. There are essentially numerous methods available for the designing and analysis of heat exchangers but the most established methods are the Bell-Delaware [7], Kern [8] and stream analysis methods. This software can be customized to apply any of these three methods. Therefore, the user is given an option to select the most appropriate method for his/her design. The development of HEDASS was based on Kern and Bell-Delaware methods, both used for the thermo-hydraulic design analysis. The mechanical design is based on TEMA standard [9]. For the heat exchanger analysis, Log Mean Temperature Difference (LMTD) method presented in textbook authored by Serth [6] is used.

To create HEDASS, we undertake a typical program development cycle that includes the following steps. Further details regarding such development cycle can be found in the following website: http://www.encyclopedia.com/doc/1G2-1552100260.html.

1. **Analysis:** defining the problem, what the program (software) should do and the appropriate inputs to produce the required outputs. Fig. 1 shows the input and output data presented in this study [6].
2. **Designing:** planning the solution to the problem, by using logical sequence of precise steps or algorithm. The design algorithm of the software gives the simple logical

sequences used in the program for the design of heat exchanger. The software design logic is illustrated in Fig. 2 [6]. The design algorithm of the software was constructed based on the simple logical sequences created by Sinnott [10] (for Open University Course T333 *Principles and Applications of Heat Transfer*) for the design of shell and tube heat exchanger.

3. **Building the graphical user interface (GUI):** selecting objects (text boxes, command tab etc), for obtaining the inputs and displaying the outputs.
4. **Coding:** writing the computer program that translates the algorithms into a programming language.
5. **Testing and Debugging:** identifying and eliminate any errors in the program. Then, testing to find the errors and debugging to correct them.
6. **Documentation:** deploying the project into distribution package with necessary manual to assist users.

HEDASS can be executed on any personal computer which has a Windows 95/98/NT/ME/2000/XP operating system with 20 MB of disk space and at least 64 MB of RAM.

3. Results and Discussion

The simulation results generated by the software depend on the data input by the user and the discussion or analysis of any result should be essentially the same. The following example shows how the software can be used to design shell and tube heat exchangers by simulating design parameters.

3.1 Example of usage

We use data provided in Example 5.1 shown in a textbook authored by Serth [6] to illustrate an example of usage and validation of HEDASS. The user is required to key in the input values for the exchanger configuration, fluids properties and conditions using the design interface (Fig. 3). By clicking the design button, the software computes the values needed for the design of the heat exchanger and generates a summary analysis report. The software is developed with an alert option, e.g. if the calculated pressure drop is greater than the allowable pressure drop, it will be highlighted in the analysis summary in red. However, if the value is within the allowable pressure drop as shown in Fig. 4 (analysis interface), the software will highlight the value in green colour indicating pressure drop compliance. Ultimately, the user is shown the sketched interface (Fig. 5) that illustrates the shell and tube fluids, their inlet and outlet temperatures, flow rates, tube length, shell and tube diameters, baffle cut as well as baffle spacing. The simulation results obtained from HEDASS are shown in Table 1. These results are compared with the text book answers and simulation results from HEXTRAN, an established software developed by SimSci-Esscor, a division of Invensys Systems, Inc. As can be seen from Table 1, the results from the HEDASS are found to be consistent with the text book answers and HEXTRAN simulation results and therefore, validate the simulation accuracy of HEDASS.

4. Conclusions

Heat Exchanger Design and Analysis Simulation Software (HEDASS) has been successfully developed using VB programming language. The utilization of VB programming language in our study is a significant plus point as this program is relatively uncomplicated and therefore affords engineering program stakeholders the prospect to develop the software themselves with relative ease. HEDASS can be used for educational purposes or perhaps as supplemental simulation software for industrial applications. The software has been validated by comparing its simulation results with benchmark problems from a well-known heat exchanger design text books and results from another simulation software. The user-friendly graphic user interface (GUI) exhibited by HEDASS renders it easy for users to input parameters and obtain output in an interactive manner.

References

[1] D. Chisholm, *Developments in Heat Exchanger Technology*, Applied Science Publishers, London, 1980.

[2] F. L. Tan and S. C. Fok, An educational computer-aided tool for heat exchanger design, *Computer Applications in Engineering Education,* **14** (2006) 77-89.

[3] K. C. Leong, K. C. Toh and Y. C. Leong, Shell and tube heat exchanger design software for educational applications, *International Journal of Engineering Education,* **14** (1998) 217-224.

[4] L. M. F. Lona, F. A. N. Fernandes, M. C. Roque and S. Rodrigues, Developing an educational software for heat exchangers and heat exchanger networks projects, *Computer & Chemical Engineering,* **24** (2000) 1247-1251.

[5] S. J. M. Cartaxo and F. A. N. Fernandes, Educational software for heat exchanger equipment, *Computer Applications in Engineering Education*, **18** (2010) 193-199.

[6] R. W. Serth, *Process Heat Transfer: Principles and Applications*, Elsevier Ltd., Oxford, United Kingdom, 2007.

[7] K. J. Bell, *Final Report of the Cooperative Research Program on Shell and Tube Heat Exchangers*, Bulletin No. 5, University of Delaware Engineering Experimental Station Newark, Delaware, 1963.

[8] D. Q. Kern, *Process Heat Transfer*, McGraw Hill, New York, 1950.

[9] TEMA, *Standards of the Tubular Exchanger Manufacturing Association,* 7th Ed, New York, 1988.

[10] R. K. Sinnott, *Chemical Engineering Design*, *Volume 6*, Butterworth-Heinemann, Oxford, United Kingdom, 2005.

Table and Figure Captions

Table 1 Simulation Results Comparison Between HEDASS, HEXTRAN and text book values.

Fig. 1 Input and output data [6].

Fig. 2 Software design logic [6].

Fig. 3 Input of data using the design interface.

Fig. 4 The analysis interface.

Fig. 5 The sketch interface.

Table 1 Simulation Results Comparison Between HEDASS, HEXTRAN and text book values.

Calculated variable	Serth [6]	HEXTRAN	HEDASS
Heat transfer area (ft^2)	454	444	454.48
Overall transfer coefficient ($Btu/h.ft^2.^oF$)	46	53.3	44.25
Fouling resistance ($h.ft^2.^oF/Btu$)	0.0056	0.00560	0.005597
Shell side transfer coefficient ($Btu/h.ft^2.^oF$)	122	191.2	104.15
Shell side Reynold's number	37158	45148	31327.87
Shell side pressure drop (psi)	2.2	2.1	1.2133
Tube side transfer coefficient ($Btu/h.ft^2.^oF$)	156	156.2	162.12
Tube side Reynold's number	10189	10189	10670
Tube side pressure drop (psi)	10.2	10.06	10.72

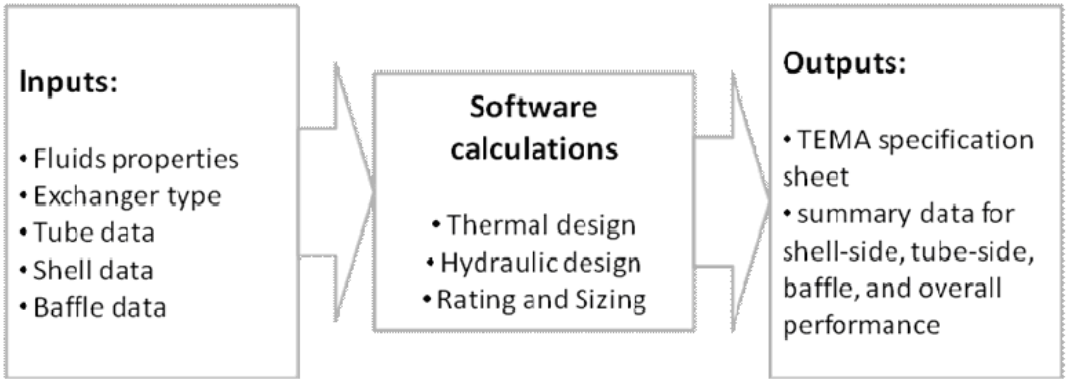

Fig. 1 Input and output data [6].

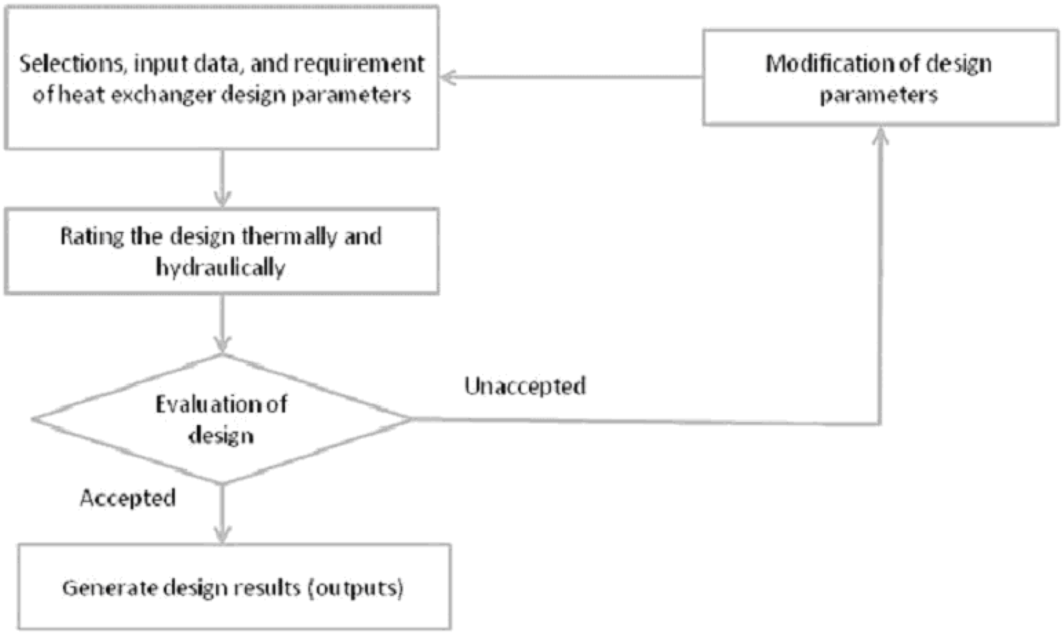

Fig. 2 Software design logic [6].

Fig. 3 Input of data using the design interface.

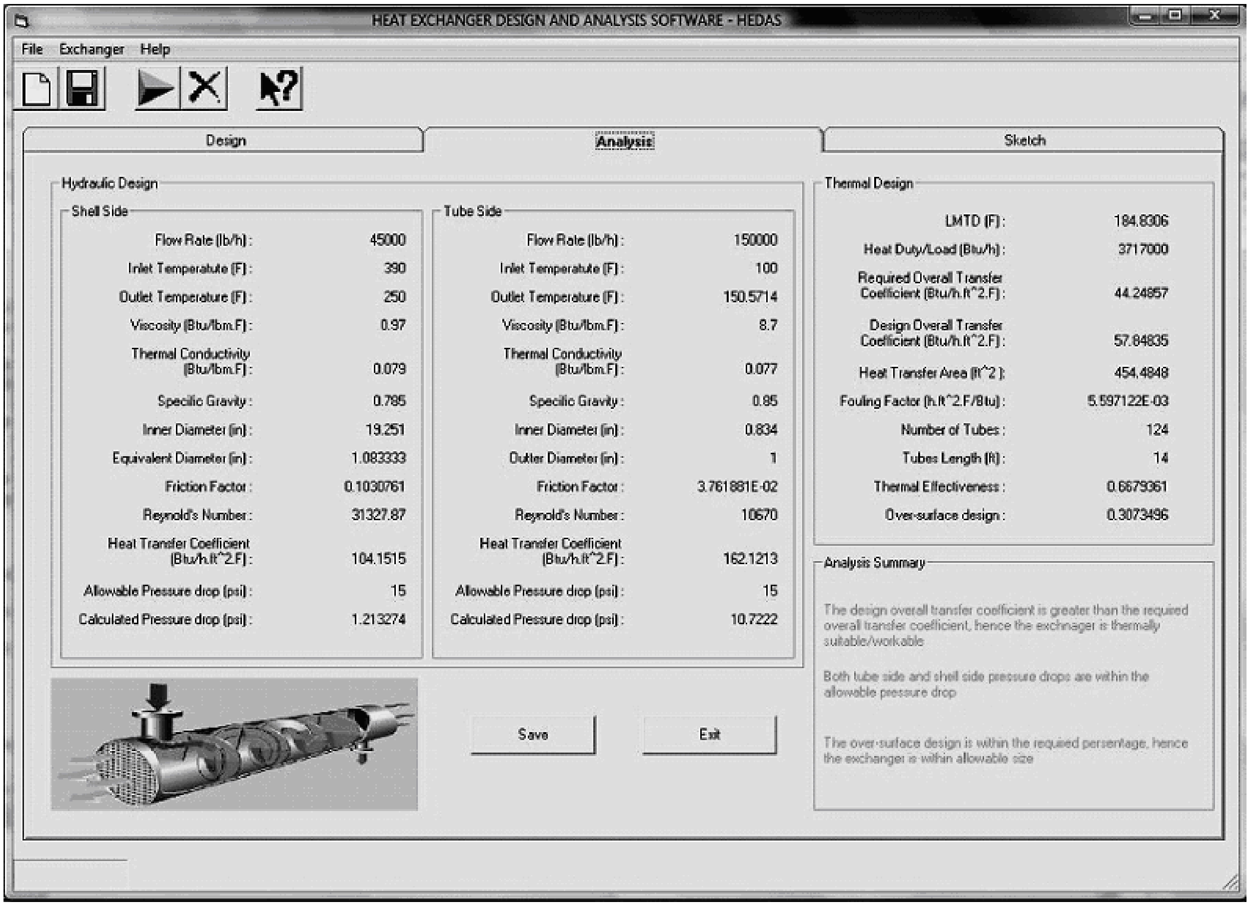

Fig. 4 The analysis interface.

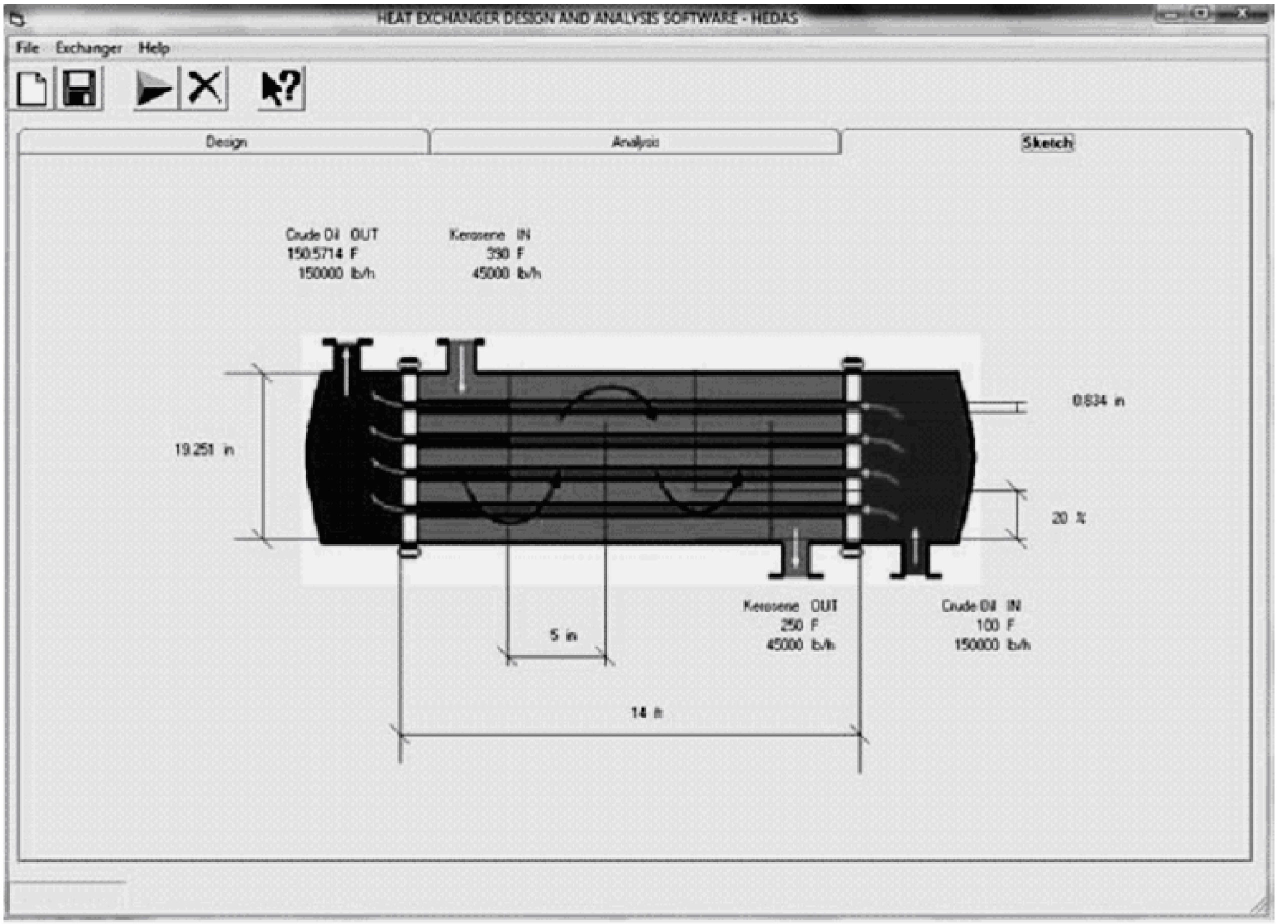

Fig. 5 The sketch interface.