International Journal of Technology and Engineering Studies
volume 5 issue 5 pp. 163-168 doi: https://dx.doi.org/10.20469/ijtes.5.10004-5

Simulating Mechanical Engineer Equations Using C++ Programming Language

Ali Q. Saeed
Northern Technical University (NTU),
Ninevah, Iraq

Muhammad T. Sultan *
Cihan University-Erbil,
Erbil, Iraq

Khaled N. Yasen
Cihan University-Erbil,
Erbil, Iraq

Abstract: This research aims to investigate the complexity of working on nested multi-variable equations of solar radiation used by mechanical engineers and how the programming languages (Matlab, C++, and even Excel spreadsheet) could facilitate their work. Mechanical engineers used several languages in executing their equations; however, varieties exist despite coding similarities. The paper first reviews all the previous studies, discussing the advantages and disadvantages of the recent works, then compares it with the proposed framework built using C++ programming language platform with the intent to facilitate any sophisticated calculations needed to generate the required result. The study concluded that mechanical engineers are highly recommended to learn at least the basics of computer programming by adding such courses to their syllabus to make them able to think logically and polish their problems solving skills.

Keywords: *Mechanical equations, C++ language, programming language, syntax and semantics, Matlab*

Received: 11 April 2019; ***Accepted:*** 10 September 2019; ***Published:*** 26 October 2019

I. INTRODUCTION

People whose specialist is other than software engineer are extremely struggle in solving complicated arithmetic equations using one of the programming languages [1]. Therefore, many of commercial user-friendly programs exist to facilitate this issue i.e., Microsoft Excel spreadsheet [2, 3], however such commercial program are manually bushy and likely to errors [4]. Which made it very difficult for mechanical engineer to handle the expected errors [5]. As a solution, nowadays cross-faculty researchers are likely to cooperate with each others to get over this kind of problem. People who have the ability to interacting with computers using one of the programming languages are called software engineer [6]. in general, they enhanced the working discipline and created new areas of practicing as [7] said "One of the main goals of software engineer is to create a software products for different problem-oriented areas and to ensure its effectiveness throughout the life cycle", many fields of knowledge a software engineer could incorporate with like (computing, mathematics and engineering, modeling, Software design, Software verification, software evolution, Software development processes, Software quality, Software project management). Further in this paper, four of mechanical engineers equations are simulated using C++ programming languages will be discussed in details.

II. LITERATURE REVIEW

Programming languages, for certain, occupies great place in engineering, mathematics, education [8], medical and other sciences. For particular, in mechanical engineer benchmark, simulating a specific problem in the real laboratories is very expensive in some cases. Therefore, computer could be used as a virtual lab to simulate such problems. Hence an engineer is required to learn how to use computer programs to fulfill his needs. According to [9] working reliability is determined by computer simulation and testing of solar still system pro-

*Correspondence concerning this article should be addressed to Muhammad T. Sultan, Cihan University-Erbil, Erbil, Iraq. E-mail: aamirkhan.es16@muetkhp.edu.pk

totype to get the accepted efficiency. Matlab is one of the most computer programs used in this field, it provides user friendly environment could easily used to express problems in mathematical form and conclude the solutions [10, 11]. Many researchers satisfied of using Matlab programming language [10] has concluded Matlab is the best choice for engineer and scientists due to the number of functions used in linear equation systems and the capability to graphically demonstrate the attributes of the problem, another [12] has used Matlab programming language to simulate the condensation and evaporation inside the still, [13] also used Matlab for theoretical analysis of solar still. Matlab provides advanced built in methods that could be used directly to alleviate the complexity of programming from scratch hence freeing great time for researchers to concentrate on their basic idea and reduced the complexity associated with learning programming languages. Matlab built-in methods have been used to solve theoretical mathematical aspects in describing an approach for teaching of scientific computing, and he enhanced the understanding of scientific computing much more than before [14, 15]. However these ready built in methods like ode45 perfectly fits on system of equations in the first order only and not for second and third order equations. Therefore modification must be done by researcher to utilize the algorithms code and find the solution, this may led to high probability of making mistakes during the transformations stages then it would be reflected on the final results, needless to mention the amount of transforming time consumed [16]. Dealing with differential0equations of the mathematical0models of engineering systems mostly encounters difficulties in term of the MATLAB programming.

Another commercial packages used by mechanical engineers researchers to virtually simulate their work are Excel [17] and MatriXx [18], these packages are relatively cheap and dont require any programming experience, they provides built it functions and formals that could be used to quickly simulate the problem and get the outputs. Using computer simulation is very important in the educating process of engineers [18]. However built-in package drawbacks lie in the modification or updating of the simulation as the source code of functions or formulas is inaccessible. According to [19] 64% of researchers use Excel spreadsheets and almost all mechanical engineers arent skilled to write a program or to develop a package. Therefore, Matlab packages would be preferred more than other programming languages.

Other programming languages like (C++, Pascal, Fortran) have been widely used by most of engineering school to teach students how to logically think and solve their mathematical problems using computer languages, to get over the problem of inaccessible source code of the commercial packages, however programming is not an easy task, simulating a program using one of these languages require good programming skills and a enough time for developing, testing and maintenance. According to [20] who have done a research to clarify the role of computer programming in mechanical engineers in Kenyan universities concluded that Matlab, MathCAD and other software application packages are very relevant to be taught to mechanical engineers however 68.75% of graduated students do not have the ability to solve engineering problems using computers.

C++, Pascal and FORTRAN are0inexpensive, their source code is accessible so modification and updating of the simulated problem is relatively possible. [21] has worked on comparing C++ and another programming language to discover their usefulness in math-intensive real word problem regarding the easiness of implementation and the usability with other languages, he concluded that C++ programming language has many elegant, good expressive features to simplify developing a software of mathematical applications.

III. SIMULATING EQUATIONS USING C++ PROGRAMMING LANGUAGE

To simulate any algorithm we need to analyze it properly, specify its requirement and understand its work then select the suitable implementation tool, this is called as logical0thinking. In our proposed paper we have used C++ programming language to simulate four different equations used by mechanical engineer to calculate:

- Convective heat transfer coefficient.
- Evaporative heat transfer coefficient.
- Distillate output.
- Calculating Unknown constant in Nusselt number expression (C and n values).

These equations work incrementally based on various inputs (constants and variables). Table 1 shows the abbreviations used in the formulas.

TABLE 1
EXAMPLE OF A FIGURE CAPTION

Abbreviations of the Constants used in the equations	Abbreviations and their values	
	Abbreviations	Values
Evaporative surface area, meter square	Aw	0.72
Time, second	T	3600
Characteristic dimension of condensing cover, meter	Lv	0.21
Acceleration due to gravity, m/s^2	G	9.81
Number of experimental observations for steady state condition	N	9

Designing a program code to execute and experiment a specific mathematic formula requires deep analyzing of the system, though we meant to breakdown the designing process into smaller parts (functions) in order to facilitate the coding process, track and catch errors easily and provides flexible implementing of the code in the future in case of modifying (adding/deleting) functions to the formula. Fig. 1 explicit the scheduling and precedence of design stage:

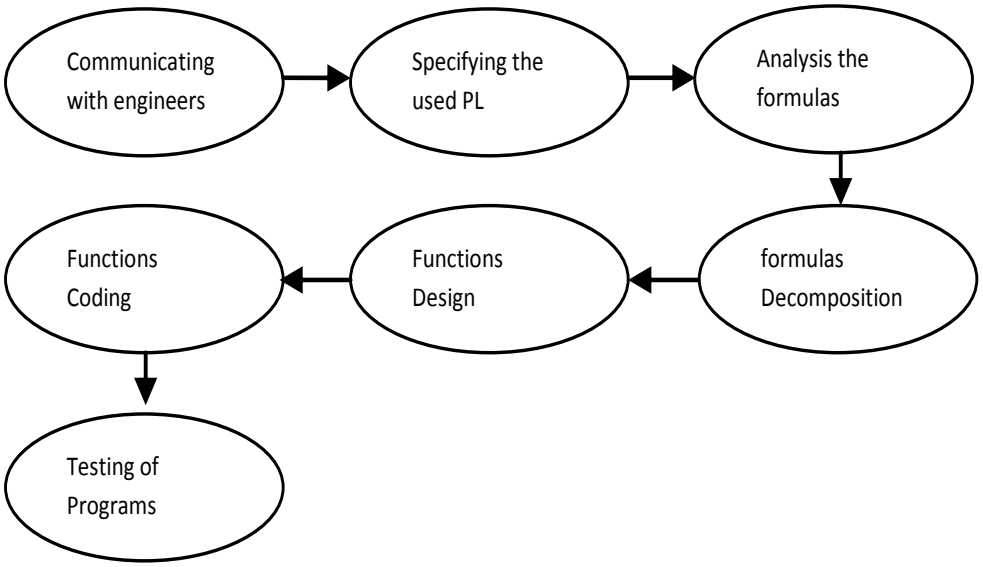

Fig. 1. Scheduling and precedence of design stage

C++ programming language has been used as a platform to build the code shown in the appendix for the following formulas:

$$x = \mathrm{In}(Gr \cdot \mathrm{Pr}) \tag{1}$$

Where Gr and Pr are the Gr ash of and r and t values respectively are considered as sub functions in the coding process. Their outputs are used to find the value of (x). Gr equations with Gr code function are shown below:

$$Gr = \frac{\beta g L_v^3 \rho^2 \Delta T}{\mu^2} \tag{2}$$

Simulating of Eq. 2 using C++ programming language is illustrated as:

$$\begin{aligned} &floatGr(floatTci, floatTw, floatTv) \\ &\{ \\ &return((Beta(Tv) * g * Lv * Lv * Lv * density(Tv) * \\ &density(Tv) * dlta_T(Tw, Tci))/(viscosity(Tv) * \\ &viscosity(Tv))); \\ &\} \end{aligned} \tag{3}$$

Notice that the terms (Beta, density, dlta_T, viscosity, viscosity) in the code line above refers to other sub functions internally invoked by Gr function these functions are coded based on the following equations:

$$\text{Expansion factor (Beta): } 1/(T_V + 273.15) \tag{4}$$

$$\text{Density } (\rho): 353.44/(T_v+273.15) \quad (5)$$

$$\delta \mathrm{T}; T_w - T_{ci} + \frac{(P_\mathrm{v}-P_{ct})(T_w+273)}{268.9\mathrm{x}10^3 - P_\mathrm{w}} \quad (6)$$

$$\text{Viscosity } (\mu): 1.718.10^{-5} + 4.620.10^{-8} T_\mathrm{r} \quad (7)$$

These functions in turn invoke other functions like (*Pci*) partial saturated vapor pressure at condensing cover temperature and (*Pw*) Partial saturated vapor pressure at water temperature which are encoded based on the following:

$$\mathrm{Pw} = \exp[25.317 - 5144/(T_w+273)] \quad (8)$$

$$\mathrm{Pci} = \exp[25.317 - 5144/(T_{ci}+273)] \quad (9)$$

The second half of the Eq. 1, shown below, also encoded as a standalone function as follow:

$$\mathrm{Pr} = \frac{\mu \cdot C_p}{\lambda} \quad (10)$$

Simulating of Eq. 9 using C++ programming language illustrated below is actually a sub function invoked by program during the calculation process:

$$floatPr(floatTv)$$
$$\{return((viscosity(Tv)*Cp(Tv))/Thermal(Tv))\} \quad (11)$$

The terms (viscosity, Cp and Thermal) written in the code above are in fact sub functions invoked internally calculated as:

$$\text{Specific heat (Cp)}: 999.2 + 0.1434.\mathrm{T_V} + 1.101.10^4 \cdot \mathrm{T_v^2} - 6.7581.10^8 \cdot \mathrm{T_V^3} \quad (12)$$

$$\text{Thermal conductivity } (\lambda): 0.0244 + 0.7673.10^{-4} T_V \quad (13)$$

Basically, The functions *Gr* and *Pr* are invoked by the main program or by any another function may need Their outputs. Arguments of *Gr* and *Pr* are passed by the main program during the invoking process. As a result, finding *x* from the above equations fulfilling the first half of the following:

$$m = \frac{N(\sum xy) - (\sum x)(\sum y)}{(N)(\sum x^2) - (\sum x)} \quad (14)$$

$$C_e = \frac{(\sum y)(\sum x^2) - (\sum x)(\sum xy)}{N(\sum x^2) - (\sum x)^2} \quad (15)$$

The second half of Eq. 12 and Eq. 13 will be calculated using Eq. 14 which is calculated by the following:

$$y = \ln\left(\frac{m_{ex}}{R}\right) \quad (16)$$

Where mew is the Distillate output (kg) expressed as:

$$m_{ew} = R.C(Gr.\mathrm{Pr})^n \quad (17)$$

C is a constant and the parameter *R* in Eq. 15 is derived from a set of equations to calculate distil output in kilogram using the following:

$$R = \frac{0.01623}{\Delta h_v} \cdot \frac{\lambda}{L_v} \cdot A_w.t.(P_w - \varphi \cdot P_{ci}) \quad (18)$$

In the built code Eq. 16 is a function which invokes Pw(water temperature) and Pci(condensing cover temperature) functions to get their values in a specific time slap under other conditions like thermal humid air and relative humidity to calculate mew and then the value of *y*.

Fulfilling all the requirements of Eq. 12 and Eq. 13 equations will facilitate in finding Eq. 17 which is as:

$$C = \exp(C_0) \quad (19)$$

Some other equations programmed as the same approach as discussed above, are:

$$h_{cw} = 0.884\left[T_w - T_{ci} + \frac{(P_w - P_{ci})(T_w+273)}{268.9\times10^3 - P_w}\right]^{1/3} \quad (20)$$

And:

$$h_{ew} = 0.01623 \cdot \frac{\lambda}{L_v} \cdot (Gr \cdot \mathrm{Pr})^n \left(\frac{P_w - \varphi \cdot P_{ci}}{T_w - T_{ci}}\right) \quad (21)$$

IV. RESULTS

Comparison is the best evaluation approach of any system, a summary of our results has obtained in comparing the run time performance and the ease of use of the language.

The mechanical engineer who has used our proposed code in his work during his master study, has generated very accurate results in least time with no effort, as mentioned in discussion, different language were used by mechanical engineers to assist executing their equations however varieties exists in spite of coding similarities. C++ is 8-10 times faster than Matlab and it is an open source code which made it more flexible than Matlab ready built-in package like ode45 in case of modifying the code is required for future implementation. In compare with other commercial package like Excel spreadsheet, Excel is very handy and it doesnt require any programming skills like C++ and Matlab however it is difficult to detect errors if occurred and it is not possible to customize functions according to our need. Fig. 2 summarizes the comparisons as the following:

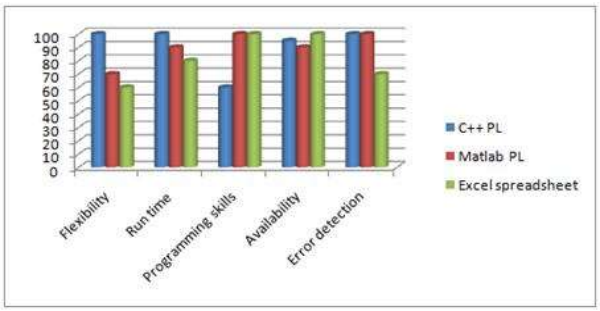

Fig. 2. Comparison among computer programs used by engineers

The differences illustrated in Fig. 2 is based on 101 studies done by mechanical engineers who have used (C++, Matlab and Excel spreadsheet) in their work as illustrated in Fig. 3. Firstly, For C++ programming language, flexibility, Run time and error detection are perfect but it is good enough in availability and it is bad in programming skills criteria because not each and every mechanical engineer knows how to program. In contrast with excel spreadsheet and Matlab ready built-in packages where programming skills is not required which is a positive indication (very good) and vice-versa for flexibility which is (very bad) for both of them as they are not an open source code whereas error detection is perfect in Matlab and bad in Excel.

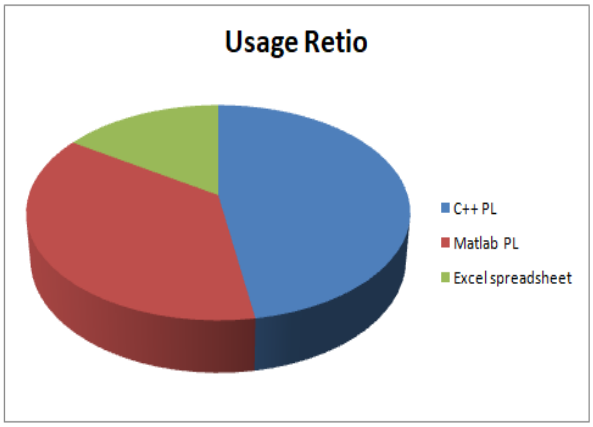

Fig. 3. Usage ratio of C++, Matlab and Excel by 101 Mechanical engineer

Lastly, Runtime is almost the same for all (C++, Matlab and Excel) where indication varies in the range of (88-100) which reflects the velocity of running a code or a function.

V. CONCLUSION AND RECOMMENDATIONS

Mechanical engineers over years have used different programming languages to fulfill their needs regardless languages advantages or disadvantages. This study shows that the core benefit of using programming languages by mechanical engineers could be summarized in the excellence of performance, high efficiency, accurate results, less effort and less time required to accomplish a single task. However each language has some limitations and this is varied from one to another. According to current paper, C++ programming language have been used to simulate solar energy equations and have successfully compromised these equations to be used in any coordinates in the world. This paper highlighted the tools, competences and knowledge which might be faced by engineers during their work. The study concluded that mechanical engineers are highly recommended to learn at least the basics of computer programming by adding such courses to their syllabus to make them able to think logically and simulate their problems themselves to get rid of using ready built-in packages or functions which are fixed for a specific kind of equations (linear equations).

C++ or Pascal programming languages are cheaper and easier to get than a licensed Matlab with its add-on tools. Moreover they dont require high computer requirement (CPU, RAM, and Motherboard) to be installed. Lastly we would like to recommend any mechanical engineers to use the code attached if they are working on solar energy to distill water as it is written to be adapted in any coordinates and working conditions.

Declaration of Conflicting Interests

We hereby declare that this work has no conflicting interests.

REFERENCES

[1] B. Milasinovicc and K. Fertalj, "Issues and challenges of adopting agile methodologies in software engineering courses," *International Journal of Technology and Engineering Studies*, vol. 4, no. 5, pp. 197–202, 2018. doi: https://dx.doi.org/10.20469/ijtes.4.10004-5

[2] T. A. Grossman. (2008) Source code protection for applications written in microsoft excel and google spreadsheet. [Online]. Available: https://bit.ly/3iaO36S

[3] T. S. Sokolova, P. I. Dorogokupets, A. M. Dymshits, B. S. Danilov, and K. D. Litasov, "Microsoft excel spreadsheets for calculation of P–V–T relations and thermodynamic properties from equations of state of MgO, diamond and nine metals as pressure markers in high-pressure and high-temperature experiments," *Computers & Geosciences*, vol. 94, pp. 162–169, 2016. doi: https://doi.org/10.1016/j.cageo.2016.06.002

[4] S. Litt. (2018) Mitigating spreadsheet risk in complex multi-dimensional models in excel. [Online]. Available: https://bit.ly/2GbAZkd

[5] P. L. Bewig. (2013) How do you know your spreadsheet is right? [Online]. Available: https://bit.ly/2Sas84Q

[6] P. L. Li, A. J. Ko, and J. Zhu, "What makes a great software engineer?" in *Proceedings of the 37th International Conference on Software Engineering,* Florence, Italy, 2015.

[7] S. Mahmudova, "Features of programming languages and algorithm for calculating the effectiveness," *Journal of Information Sciences and Computing Technologies (JISCT)*, vol. 8, no. 1, pp. 769–780, 2018.

[8] P. Byrne and G. Lyons, "The effect of student attributes on success in programming," in *Proceedings of the 6th Annual Conference on Innovation and Technology in Computer Science Education,* Canterbury, UK, 2001, pp. 49–52.

[9] J. Li, "Computer-aided design, modeling and simulation of a new solar still design," *Modelling and Simulation in Engineering*, vol. 2011, pp. 10–15, 2011.

[10] S. Sen and G. A. Shaykhian, "Matlab tutorial for scientific and engineering computations: International Federation of Nonlinear Analysts (IFNA); 2008 World Congress of Nonlinear Analysts (WCNA)," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 71, no. 12, pp. 1005–1020, 2009. doi: https://doi.org/10.1016/j.na.2009.01.069

[11] D. Higham and N. Higham, *MATLAB guide*. Bangkok, Thailand: Siam, 2005.

[12] M. I. M. Shatat, "New and renewable energy and environmental engineering," Durham University, Durham, England, Unpublished master's thesis, 2008.

[13] O. Badran, "Theoretical analysis of solar distillation using active solar still," *International Journal of Thermal & Environmental Engineering*, vol. 3, no. 2, pp. 113–120, 2011.

[14] P. R. Turner, "Teaching scientific computing through projects," in *FIE'99 Frontiers in Education. 29th Annual Frontiers in Education Conference. Designing the Future of Science and Engineering Education,* Puerto Rico, PR, 1999.

[15] M. A. Rawajbeh, I. A. Haddid, and H. Al-Zoubi, "Adoption of cloud computing in higher education sector: An overview," *International Journal of Technology and Engineering Studies*, vol. 5, no. 1, pp. 23–29, 2019. doi: https://dx.doi.org/10.20469/ijtes.5.10004-1

[16] W. K. Ahmed, "Advantages and disadvantages of using MATLAB/ode45 for solving differential equations in engineering applications," *International Journal of Engineering*, vol. 7, no. 1, pp. 25–31, 2013.

[17] J. Walkenbach, *Excel 2007 Bible*. Hoboken, NJ: John Wiley & Sons, 2007.

[18] D. Ibrahim, "Engineering simulation with MATLAB: improving teaching and learning effectiveness," *Procedia Computer Science*, vol. 3, pp. 853–858, 2011.

[19] O. Shoewu, "Problem solving using programming language," *Pacific Journal of Science and Technology*, vol. 10, no. 2, pp. 408–412, 2009.

[20] E. Ogur and W. O. Ogola, "The role of computer technology in mechanical engineering education in Kenyan Universities," *The International Journal Of Engineering And Science (IJES)*, vol. 2, no. 11, pp. 77–81, 2013.

[21] S. Frame and J. W. Coffey, "A comparison of functional and imperative programming techniques for mathematical software development," *Journal of Systemics, Cybernetics and Informatics*, vol. 12, no. 2, pp. 1–10, 2014.