Contemporary Engineering Sciences, Vol. 10, 2017, no. 23, 1103 - 1111
HIKARI Ltd, www.m-hikari.com
https://doi.org/10.12988/ces.2017.79123

A Python Based Program for the Parametric Study of Heat Transfer Processes in Transient State

Luis Molina Cantillo[1], Luis Obregón Quiñones[2] and Guillermo Valencia Ochoa[3]

[1] Chemical Engineering Program, Universidad del Atlántico, km 7 antigua vía Puerto, Colombia

[2] Sustainable Chemical and Biochemical Processes Research Group
Universidad del Atlántico, km 7 antigua vía Puerto, Colombia

[3] Mechanical Eng., Efficient Energy Management Research Group – Kaí, Universidad del Atlántico, km 7 antigua vía Puerto, Colombia

Abstract

This paper presents a program developed in Python 3.5 with GUI created with PyQt 4.11.4 for studying the transient heat transfer problems where the heat rate, final temperatures and time are calculated depending on the inputs variables. The program, called DynamicHT uses two different methods for solving the systems. The first one is the Lumped capacitance method, which assumes that the temperature gradients within the solid are negligible, and all changes of temperature are time-dependent. The second method, one-dimensional heat conduction, considers the temperature gradients within the solid and its changes with the time. In the first method, a heat balance was developed for a control volume within the solid boundaries and it was solved by methods of integration. The second method was developed for a plane wall, the equation obtained corresponds a partial differential equation, which is solved using the explicit finite difference method. The solution of two different problems are presented as examples to show the use of DynamicHT. The relation between some determinant variables is described and analyzed. The software was highly accepted by a population of 36 students that evaluate it statistically in terms of clarity, precision and relevance with the implementation of a *t* test with a significance level of 5%.

Keywords: Lumped capacitance, plane wall, difference method, Python 3.5

1 Introduction

For chemical and mechanical engineers, and scientists in general, it is important to manage concepts related to transport phenomena [1]. One of the areas that is highly important is heat transfer because of the significant variables involved when engineers need to solve a problem, optimize a process, design an equipment, etc. The process of understanding heat transfer processes starts from engineering academies, where students learn the basic concepts that are complemented with laboratory practices [2]. However, sometimes students have to face complicated problems, that involve lots of variables, causing a loss of time in mechanical calculations made by hand. This situation rises the probability of making mistakes while solving the problems making difficult the process of learning. One example of this is the transient heat conduction, where the solution involves solving partial differential equations. Generally, these equations require a mathematical rigorous treatment or a solution by a numerical method [3, 4]. For that reason, it is necessary for students to use several programs that help to solve the problems. Usually, students and scientists must create their own algorithms for solving their problems [3-5].

It has been found that using computer programs for studying improves the learning process [6-8]. Some of the programs that are widely used around the world are Aspen Hysys, Aspen Plus, Chemcad, Comsol Multiphysics, among others. All of the programs mentioned previously, have special characteristics, which make them different from the others. Despite these programs are robust, they do not emphasize in the basic phenomena, which is important for students during the learning process of heat transfer. It is not easy to see the gradients of temperature inside the wall of equipment, or how the external conditions affect the temperature in the inside boundary wall of the reactor. Furthermore, it is not easy to study how the conditions mentioned before change with time. Considering the degree of difficulty of studying transient heat transfer problems, it is necessary to complement the learning process with tools that let students to analyze different situations in a fast and simple way.

In this work a new graphical user interface (GUI) developed with PyQt 4.11.4 written in Python 3.5 is presented, where students can analyze situations where the transient heat conduction is involved, using two different methods: the lumped capacitance and one-dimensional transient conduction. The users can study the effect of time in the internal temperature of small solids, with different geometries or see the transient temperature distribution inside a plane wall. The results can be seen numerically or graphically. It is shown the importance of the software in the learning process with an evaluation made to 36 students before and after the use of the software DynamicHT by means of a t-test for independent samples.

2 Methodology

2.1 Presentation of the software

DynamicHT is a software written in Python 3.5 which allows the simulation of transient heat conduction for small solids in different geometries and plane walls, see figure 1. The user can introduce the properties of the solid that is studied or select a substance from the data base. The software is divided into two categories: DynamicHT Lumped Capacitance and DynamicHT Plane Walls. The first one allows solving problems following the lumped capacitance analysis, assuming that temperature gradients within the solids are negligible. The second one was created to simulate the temperature changes respect to the position and the time simultaneously. DynamicHT has an interface created using PyQt 4.11.4, which provides bindings for Qt 4.

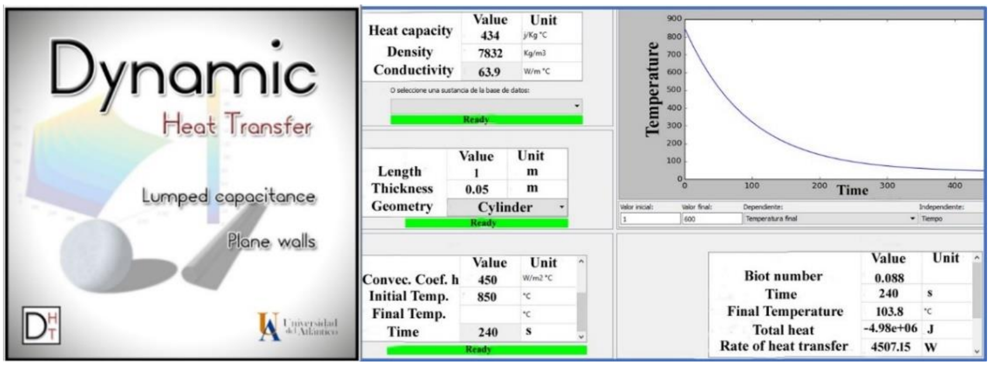

Figure 1. Interface software DynamicHT Lumped Capacitance

2.2 Fundamental equations used in lumped capacitance method

The equations used in the software follow the nomenclature shown in table 1.

Table 1. Nomenclature

Variable	**Symbol**	**Variable**	**Symbol**
Biot number	Bi	Mass	m
Characteristic length	Lc	Specific heat	C_p
Density	ρ	Superficial area	A_s
Diffusivity	α	Surface temperature	T_s
Environmental temperature	T_∞	Temperature of node m	T_m
Fourier number	τ	Thermal conductivity	k
Heat generation	e	Time	t
Heat transfer coefficient	h	Time step	Δt
Initial temperature	T_i		

For small sized solids with high thermal conductivity like a small ball of metal, the temperature gradients are negligible and the temperature can be assumed as time-dependent [9] implying that the conductivity of the material is infinite, which is physically incorrect [1]. It is correct to apply this analysis when the Biot number is lower than 0.1, see equation (1):

$$Bi = \frac{hLc}{k} \tag{1}$$

If the condition is correct, it is possible to obtain from the energy balance the equation that describers the system. The energy balance over the control volume is:

$$-hA_s(T_\infty - T_s) = mCp\frac{dT}{dt} \tag{2}$$

It is possible to solve the equation (2) by integration resulting equation (3):

$$-\frac{h}{\rho CpLc}t = \ln\frac{T - T_\infty}{T_i - T_\infty} \tag{3}$$

which let to calculate the time that takes a small piece of solid to reach a specific temperature or to determine the temperature that will get after a period of time.

2.3 Fundamental equations used in transient one-dimensional conduction

To solve problems where the temperature of the solid changes with time and position, a numerical method is applied, the explicit finite difference method. It is developed by writing the Taylor series expansion and neglecting all terms except the first two. From the energy balance, it is possible to obtain the following equation, which corresponds to the transient finite difference for an interior node:

$$kA\,\frac{T_{m-1} - T_m}{\Delta x} + kA\frac{T_{m+1} - T_m}{\Delta x} + \dot{e}_m A\Delta x = \rho A\Delta x c_p\frac{T_m^{i+1} - T_m^i}{\Delta t} \tag{4}$$

When the equation (4) is rewritten, introducing the *mesh Fourier number* and solving it for T_m^{i+1}, it yields:

$$T_m^{i+1} = \tau\left(T_{m-1}^i + T_{m+1}^i\right) + (1 - 2\tau)T_m^i + \tau\frac{\dot{e_m}\Delta x^2}{k} \tag{5}$$

For a boundary node subjected to convection (for example the node 2), the finite difference formulation can be performed similarly to the interior node, but changing the conduction condition by the Newton's law of cooling, see equation (6):

$$hA\left(T_\infty - T_2^i\right) + kA\frac{T_1^i - T_2^i}{\Delta x} + \dot{e}_2 A\frac{\Delta x}{2} = \rho A\frac{\Delta x}{2}c_p\frac{T_2^{i+1} - T_2^i}{\Delta t} \tag{6}$$

This equation can also be rewritten by introducing the Fourier number and solving for T_2^{i+1}

$$T_2^{i+1} = \left(1 - 2\tau - 2\tau\frac{h\Delta x}{k}\right)T_2^i + \tau\left(2T_1^i + 2\frac{h\Delta x}{k}T_\infty + \frac{\dot{e_2}\Delta x^2}{k}\right) \quad (7)$$

The Fourier number can be calculated by:

$$\tau = \frac{k\Delta t}{\rho c_p \Delta x^2} \quad (8)$$

The equations (5) and (7) are used by DynamicHT as general models for calculating the temperature after a period of time Δt. The explicit method is not stable unconditionally, so it is needed to use a criterion of stability that let to find the maximum value for Δt that avoids the divergence of solution, and is obtained from the *primary coefficients*, that corresponds to coefficients of all T_m^i in T_m^{i+1} expressions. The maximum Δt is such that the primary coefficient must be equal or greater than cero. When there is no convection and the temperature changes by internal conduction, the maximum value for the time step is:

$$\Delta t_{max} = \frac{\rho c_p \Delta x^2}{2k} \quad (9)$$

When convection is presented [9], the maximum value for time step is calculated by:

$$\Delta t_{max} = \frac{\rho c_p \Delta x^2}{2k(1 + h\Delta x/k\ )} \quad (10)$$

3 Results and discussion

In this section, some results obtained using DynamicHT are compared with the results registered in the literature [10].

3.1 Determination of the temperature of a thermocouple

In this case, it was studied the time that takes a thermocouple to reach 99 percent of the initial temperature difference between the thermocouple and a gas stream. The heat transfer coefficient (*h*) is 210 W/m^2 °C, the density is 8500 kg/m^3 and the thermal conductivity (*k*) is 35 W/m °C. Assuming that the initial temperature of the thermocouple is 0 °C and the gas stream temperature is 100 °C, it must be calculated the time that takes the thermocouple to reach 99 °C. The results obtained using DynamicHT and the obtained in [10] are registered in the Table 2. and the relative error was calculated.

Table 2. Results obtained with DynamicHT compared with literature results.

Parameter	Result by DynamicHT	Result by literature	Error (%)
Biot number	0.001	0.001	0.000
Time (s)	9.941	10.000	0.059

The changes in temperature of the thermocouple with time are shown in

Figure *2*. It is possible to see that temperature varies in an exponential way from 0°C to 100°C. From equation (3) it can be deduced that the thermocouple will reach exactly 100°C when time is infinite. Also in Figure 2, the thermocouple temperature was plotted for different values of *h* and it is possible to see a relation: the higher the convective coefficient, *h*, the faster the temperature reach 100 °C.

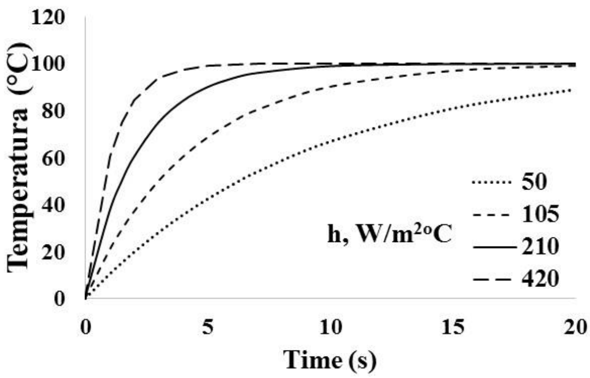

Figure 2. Temperature of thermocouple

3.2 Determination of temperature gradients within a plane wall

In this case, the temperature of a large uranium plate is studied when initially its temperature is uniform and equals to 200 °C. At *t=0* one side is brought into contact with ice and is maintained a 0 °C at all times. The other side is subjected to convection, the environmental temperature is 30 °C. The complete data of the problem is resumed in the Table 3.

Table 3. Parameters case of study

Parameter	Value
Thickness (L)	4 cm
Thermal conductivity (k)	28 W/m °C
Diffusivity (α)	12.5×10^{-6} m^2/s
Heat generated (e)	5×10^{6} W/m^3
Heat transfer coefficient (h)	45 W/m^2 °C

When the problem is solved using DynamicHT, the results are shown in the Table 4 and compared with the results obtained in [8] and the relative error is calculated and tabulated.
Figure 3 shows the temperature gradients for different times. It is possible to see how the temperature changes for each point and time.

Table 4. Results obtained with DynamicHT compared with manual solution

Time (s)	**Final temperature-DynamicHT**		**Final temperature -literature**		**% Error**	
	Node 1	Node 2	Node 1	Node 2	Node 1	Node 2
150	106.8°C	140.3°C	106.3°C	139.0°C	0.5	0.9
300	103.9°C	136.2°C	103.8°C	136.1°C	0.1	0.1
600	103.7°C	136.0°C	103.7°C	136.0°C	0.0	0.0

The temperature of the constant boundary point in *L = 0 cm* is always constant, as described by the problem. The temperature in the middle point of the wall, increases at the beginning and then decreases, until getting a stable value. The subjected convection boundary point, in *L= 4 cm* has initially a temperature of 200 °C, it rises until 214 °C and then decreases until a stationary value of 136 °C. Also, it is possible to see, that temperature changes faster at the beginning, and after 150 seconds it keeps approximately constant.

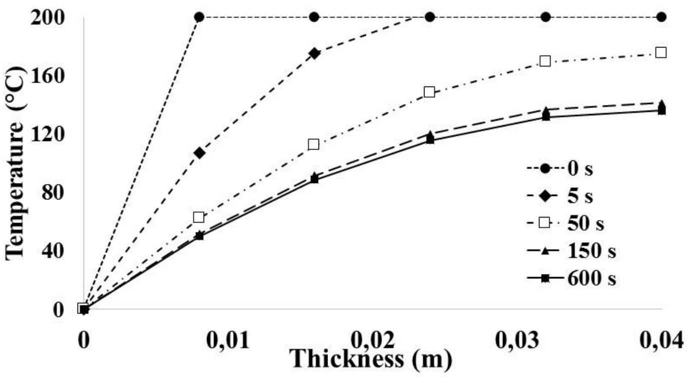

Figure 3. Temperature gradients within the plane wall

3.3 Statistical analysis in terms of the criteria clarity, precision and relevance

In order to test the importance of the software, a group of 36 students was evaluated in class solving the mentioned case studies before and after the use of the software. With the implementation of a survey, it was made a statistical analysis based on the grades given by the students according with a rubric conformed by a specific criteria and rating scale. The criteria used was clarity, precision and relevance because they are part of the intellectual standards of quality to evaluate the critical thinking. The rating scale is composed of 4 ranges as shown in table 1 with a maximum of 5 and a minimum of 0. A *t*-test was made for the criteria used to evaluate the difference in the average grade between the group of BEFORE and the group AFTER. Table 5 shows the results for the data obtained in both groups. It can be seen that all the *P*-values indicates that there is a significant difference between the grades of both groups demonstrating that the software DynamicHT has an important effect in the student learning process. This result shows the need to implement user friendly computational tools to guarantee a substantial learning and to improve the critical thinking of the students.

Table 5. Statistical results after the use of the software.

Dependent variables	N	BEFORE				AFTER				*P*-value	Tcrit	T
		M	SD	MIN	MAX	M	SD	MIN	MAX			
Clarity	36	3.57	0,23	3.30	3.80	4.58	0,29	4.15	4.90	0,0000	2.05	-9.03
Precision	36	3.43	0.15	3.10	3.80	4.51	0,33	4.00	4.80	0,0000	2.05	-10.21
Relevance	36	3.12	0.31	2.80	3.50	4.42	0,24	4.15	4.85	0,0000	2.05	-8.39

Note: M=Mean, SD= Standard Deviation, Tcrit.= T critical value

4 Conclusions

A software for studying problems of transient heat transfer, called DynamicHT, developed in Python 3.5 and PyQt 4.11.4 was presented, using two different case studies that let users to discover some of the main characteristics of the program. DynamicHT allows the solution of problems following the lumped capacitance method and one-dimensional transient heat conduction in plane walls. For the lumped capacitance method, the temperature gradients are negligible and a mathematical process beginning from an energy balance over a control volume and its solution by integration is used. DynamicHT uses the Biot number to determine whether is correct to use the method or not. It is correct when the Biot number is lower than 0.1. For one-dimensional heat conduction, gradients are considered and the finite method in explicit formulation was used to solve the partial differential equations associated with this type of problems, developing a general model which adapts itself for each situation studied. DynamicHT uses an algorithm for selecting a value for time step that claims the convergence of the solution and avoid incoherent results. A *t*-test for paired samples was used to evaluate statistically the use of the software by the students. A 95% level of confidence was used and a P-value lower than 0.05 was obtained indicating a significant effect in the use of the software resulting in a great support for teaching the concepts of transient heat transfer processes, promoting critical thinking in students, and meaningful learning.

References

[1] Theodore L. Bergman, Adrienne S. Lavine, Frank P. Incropera, David P. DeWitt, *Fundamentals of Heat and Mass Transfer*, John Wiley & Sons, Inc., Jefferson City, 2011.

[2] I. F. Meza Castro, A.E. Herrera Acuna and L. G. Obregon Quinones, Experimental Determination of New Statistical Correlations for the Calculation of the Heat Transfer Coefficient by Convection for Flat Plates, Cylinders and Tube Banks, *INGE CUC*, **13** (2017), no. 2, 9-17.
https://doi.org/10.17981/ingecuc.13.2.2017.01

[3] A. Bounouar, K. Gueraoui, M. Taibi, A. Lahlou, M. Driouich, M. Sammouda, S. Men-La-Yakhaf and M. Belcadi, Numerical and Mathematical Modeling of Unsteady Heat Transfer within a Spherical Cavity: Applications Laser in Medicine, *Contemporary Engineering Sciences*, **9** (2016), no. 24, 1183-1199.
https://doi.org/10.12988/ces.2016.67118

[4] C. N. Takayama Mori and E. C. Romao, Numerical Simulation by Finite Difference Method of 2D Convection-Diffusion in Cylindrical Coordinates, *Applied Mathematical Sciences*, **9** (2015), no. 123, 6157-6165.
https://doi.org/10.12988/ams.2015.58547

[5] M. Donisete de Campos, E. C. Romao and L.F. Mendes de Moura, Analysis of Unsteady State Heat Transfer in the Hollow Cylinder Using the Finite Volume Method with a Half Control Volume, *Applied Mathematical Sciences*, **6** (2012), no. 39, 1925-1931.

[6] R. M. Gourde and B. Akih-Kumgeh, A Matlab program for the determination of thermodynamic properties of steam, *International Journal of Mechanical Engineering Education*, **45** (2017), no. 3, 228-244.
https://doi.org/10.1177/0306419016682146

[7] L. G. Obregon Quinones, L. F. Arrieta Viana and G. E. Valencia Ochoa, Thermal Design and Rating of a Shell and Tube Heat Exchanger Using a Matlab® GUI, *Indian J. Sci. Technol.*, **10** (2017), no. 25, 1-9.
https://doi.org/10.17485/ijst/2017/v10i25/114038

[8] L. G. Obregon Quinones, L. F. Arrieta Viana and G. E. Valencia Ochoa, A Matlab-Based Program for the Design and Simulation of Wet Cooling Towers, *Chem. Eng. Trans.*, **57** (2017), 1585-1590. https://doi.org/10.3303/CET1757265

[9] Y. Cengel and A. Ghajar, *Heat and Mass Transfer*, 4 ed., McGraw-Hill, Mexico, 2011.

[10] Y. Cengel, *Heat and Mass Transfer: Fundamentals and Application*, 5 ed., McGraw-Hill, 2015.

Received: October 14, 2017; Published: November 16, 2017