Back-of-the-spreadsheet calculations: a tool for active learning and design

R J Hugo
Department of Mechanical & Manufacturing Engineering
Schulich School of Engineering
University of Calgary, Calgary, Alberta, Canada T2N 1N4

Abstract

This paper presents two different approaches that have been used to teach junior-year Mechanical Engineering students numerical analysis in a heat transfer course. Over a period of seven years, both MATLAB and Microsoft Excel were used. Examples of problems solved using both MATLAB and Excel are presented. Student reaction to both software packages is discussed. The spreadsheet-based application described in the paper is shown to be an effective tool for students to use during design-build exercises, enabling them to perform quick and low-cost "back-of-the-spreadsheet calculations."

Keywords: Heat Transfer, Microsoft Excel, Numerical Analysis, Computer-Aided Engineering

Introduction

One of the greatest change mechanisms that has influenced engineering education over the past 40 years has been the rapid development of computer hardware and software. An examination of a typical engineering program from the early 1970's would reveal a single course in computer programming, often taught using a language such as FORTRAN66. Computer programming in the early 1970's was a very time consuming process, involving the use of punched cards for program input and printouts for output. As a result of this complicated process, computer integration in engineering programs in the early 1970's was restricted mostly to introductory programming courses.

Through the evolution of computer hardware from mainframes, to minicomputers, and now microcomputers the nature by which computer applications have found their way into engineering education has changed significantly. Today's engineering educators are presented with a large number of options when it comes to deciding how to integrate computer applications into the curriculum.

Considering software alone, several levels of complexity exist. At the programming level are a variety of languages such as FORTRAN, C++, and Java that can be used to introduce programming concepts and also for the development of specific software applications. Above these languages reside analysis software packages that include MATLAB, IDL, MathCAD, Excel, and Mathematica. These applications provide data analysis routines that have been integrated into one common interface, simplifying the process of data storage, analysis, and display. At the top of the computer applications for engineering analysis hierarchy reside specialized software packages that have been developed to solve specific engineering problems, such as fluid dynamics, heat transfer or solid mechanics. These specialized software packages are often referred to as Computer-Aided Engineering (CAE) packages, and include Computational Fluid Dynamics (CFD) software such as Fluent and CFX and Finite Element Method (FEM) software including ANSYS and NASTRAN.

As a result of the wide variety of choices, many engineering programs feature an introductory programming course that teach students a programming language such as C++, followed by a numerical analysis course that teach students methods of interpolation, extrapolation, integration and differentiation using software applications such as MATLAB or Excel. Specialized CAE applications are then often left to senior-year courses, providing students with exposure to packages designed specifically to solve subject-specific problems, as illustrated in Figure 1.

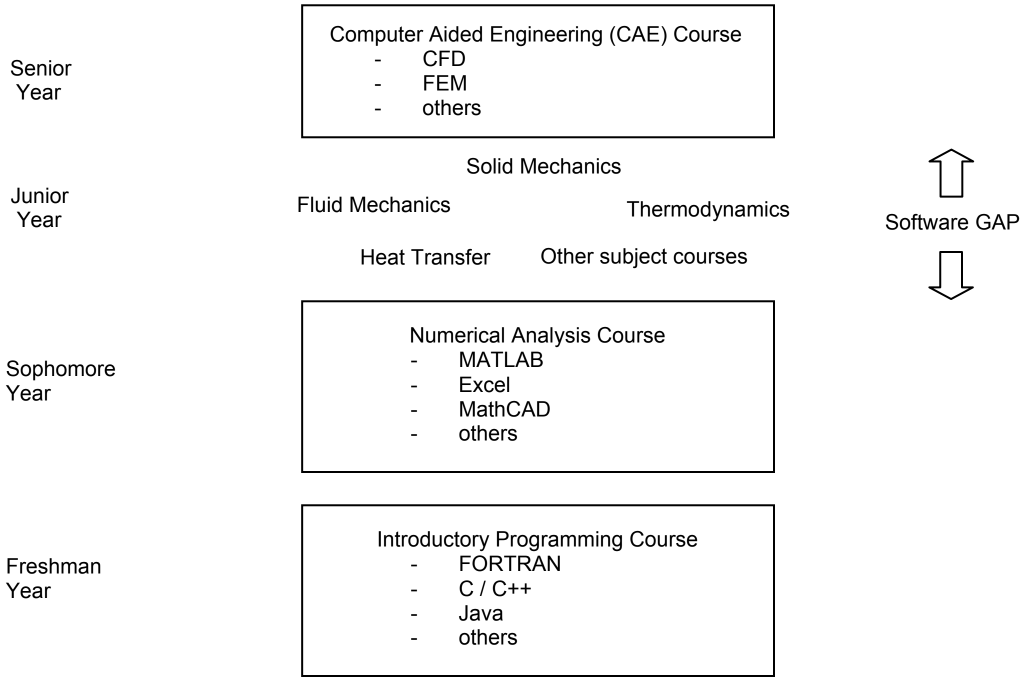


Figure 1. Course Sequence in Typical Mechanical or Aerospace Engineering Program

Although this course sequence of a programming language course, a numerical analysis course, and a CAE course is logical in its flow from the general to the specific, it has a number of drawbacks when it comes to CDIO implementation. The use of CAE applications during design-build exercises provide students with an excellent learning opportunity, linking numerical analysis to real-world systems. Unfortunately, with the course sequence shown in Figure 1 students are not exposed to CAE applications until their senior year and as a result they are not able to apply these tools to design-build exercises encountered during earlier years.

Some engineering programs have moved CAE courses to the junior year, however this can be challenging, especially if the CAE course is taught either before or simultaneously with a subject course on which the CAE model is based. A second option is to include subject-specific numerical analysis methods in junior-year engineering courses such as fluid mechanics, heat transfer, or solid mechanics. This, however, needs to be done with caution given the small number of lecture hours that can be devoted to the topic of numerical analysis in these courses. Educators involved with junior-year engineering courses can either choose to cover the fundamentals of numerical analysis applied to the particular subject, or they can introduce a specific CAE software package. Given either choice, however, the amount of coverage tends to be somewhat superficial given the limited amount of time available. This results in a situation that is labeled the "Software Gap" in Figure 1. The Software Gap refers to the large gap that exists between general numerical analysis tools learned in sophomore year to subject specific CAE applications learned in senior year.

The paper will outline the challenges encountered by the author while trying to integrate numerical analysis software into a junior-level Mechanical Engineering heat transfer course in an effort to bridge the Software Gap. The paper will discuss two different approaches that have been used to teach numerical methods for heat transfer, and it will illustrate the nature of problems encountered with both approaches. It will conclude with observations about the type of software that appears best suited to help bridge the Software Gap.

Integrating Numerical Analysis into Heat Transfer

The basis for this paper is derived from experiences gained while teaching heat transfer to junior-year Mechanical Engineering students at the University of Calgary in the Schulich School of Engineering over a period of seven years. The average number of students in a typical junior-year heat transfer course was around 110, and a typical semester would involve approximately 36 hours of lecture and 12 hours of laboratory exercise. Approximately 4 hours of lecture time (8% of total lecture time) was devoted to the topic of numerical methods applied to heat transfer, with the finite difference method forming the basis for the material. Two different approaches were used while teaching numerical methods for heat transfer: the first involved the use of MATLAB while the second involved the use of Microsoft Excel. Both methods will be discussed in the following subsections.

Numerical Methods using MATLAB

Four years of experience was gained while using MATLAB to teach numerical methods applied to heat transfer. MATLAB was used as some of the students had already been exposed to it during an earlier course in numerical analysis, and by using MATLAB in heat transfer students would further their familiarity with this software package.

During lectures students were introduced to the finite difference method using an approach based on material from the textbook by Ozisik [1], and the heat diffusion equation was recast in finite difference form. Students were shown how to solve a simple finite difference problem using MATLAB, and they were then shown how to apply the method to the solution of the heat diffusion equation. The general form of the heat diffusion equation was expressed as:

$$\frac{\partial^2 T}{\partial x^2}+\frac{\partial^2 T}{\partial y^2}+\frac{\partial^2 T}{\partial z^2}+\frac{\dot{q}}{k}=\frac{\rho C_p}{k}\frac{\partial T}{\partial t} \tag{1}$$

and this was then simplified through the assumption of steady-state, one-dimensional heat transfer with internal heat generation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\dot{q}}{k} = 0 \qquad (2)$$

Equation (2) was then used to derive the finite-difference form of the heat diffusion equation and expressed as follows:

$$T_{m-1} - 2T_m + T_{m+1} + \frac{(\Delta x)^2 \dot{q}_m}{k} = 0 \qquad (3)$$

Where $\dot{q}_m$ is the internal heat generation in units of W/m^3 at node *m*. This equation applies to all internal nodal locations, i.e., *m* = 1,2,3, …, M-2, M-1. Equations at nodal locations *m*=0 and *m*=M are found by prescribing the boundary conditions at the surface of the solid. Temperatures at nodal locations T_m were then solved for by inverting a matrix using MATLAB.

Example Assignment Problem: The assignment problems given to students while using MATLAB were always one dimensional. The one-dimensional restriction was imposed in order to reduce the complexity of the problem. An example of a typical assignment problem is:

> A straight fin, illustrated in Figure 2, has a triangular profile given by the function $y = 0.2(L - x)$ and is very long in the direction normal to the page. Both top and bottom surfaces experience convection with $T_\infty = 15^\circ$ C and $h = 100$ W/m^2 K. The base of the fin is maintained at $T_\infty = 115^\circ$ C and the fin material has a thermal conductivity of 25 W/m C. Assuming one-dimensional heat transfer and using a finite-difference method with a space increment of 10 mm, determine the fin heat rate and efficiency. Plot the computed fin temperature distribution.

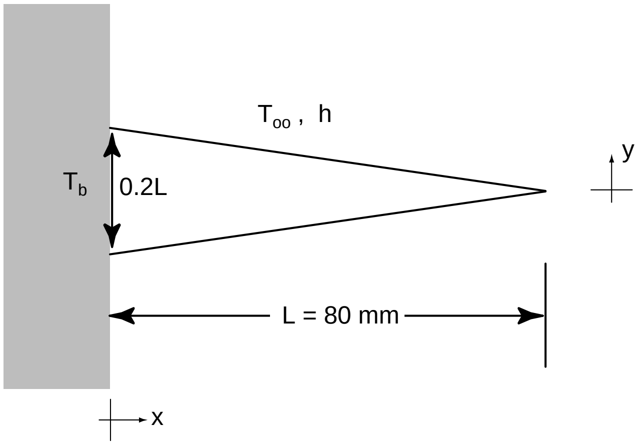


Figure 2. Tapered Fin Geometry used in Example MATLAB Assignment Problem

The above problem requires that students write out the conduction-convection equation for a fin, and then to express the resulting differential equation in finite difference form. Solving for the temperature distribution in the fin results in the temperature profile shown in Figure 3.

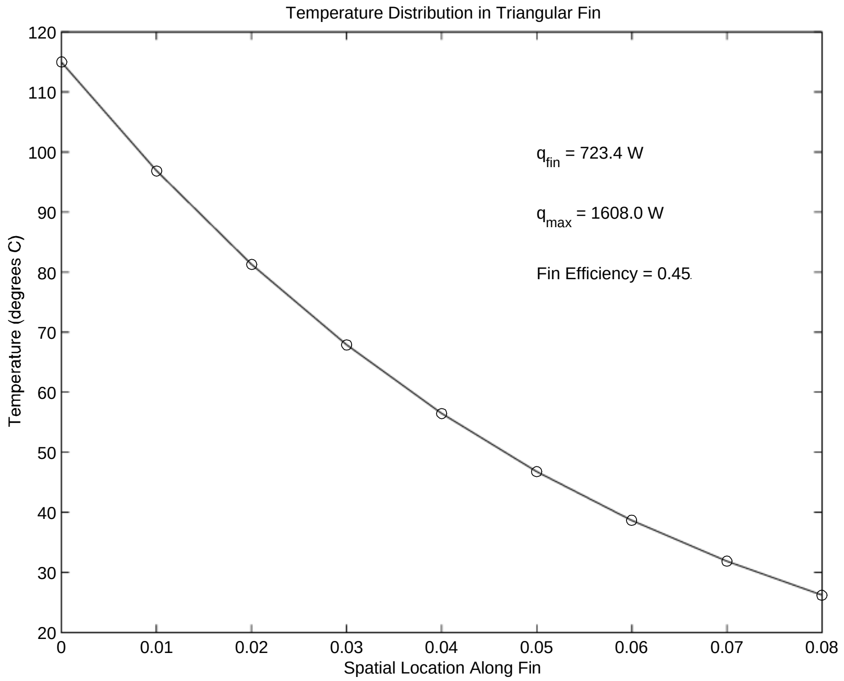


Figure 3. Temperature Distribution in Triangular Fin

Observations: Although the use of MATLAB facilitated the solution to a wide variety of one-dimensional heat transfer problems, several repeating patterns were observed with each consecutive year that numerical methods in heat transfer were taught in this manner. The original motivation in having students apply the finite difference method to the heat diffusion equation was to help them further understand the basic concepts behind the finite difference method. It was also hoped that further exposure to MATLAB would make them stronger in this application. Although they did gain further skills in using MATLAB, it was found that the exercises quickly evolved into MATLAB programming exercises as opposed to heat transfer learning exercises. Once students were able to get their programs to work, they had little interest in understanding the physical meaning behind their results due to the large amount of time spent in writing and debugging MATLAB programs.

Motivation for a Change: After using MATLAB for four years, a number of factors began to arise providing motivation for change. As mentioned earlier, there was growing concern that the students were spending more time learning how to program in MATLAB then they were learning the concepts of heat transfer. Another motivation came from informal discussions with students who had returned from internships with industry, as well as recent graduates who were working in industry. When asked what software was available to them while working in industry, a significant majority of them responded Microsoft Office. Given that Microsoft Office has 95% of the market share with over 400 million copies in use [3], this response was not overly surprising. Very few respondents would mention access to software products beyond Microsoft Office. When asked the reason why, the response was always that it was difficult to justify software expenditures of $1000 or higher for the average engineer. Another reason was that it was easier for large groups to develop custom applications using Visual Basic for Applications coupled with Excel given that all engineers already have these applications installed on their office computer, and the resulting code is simple enough for all workgroup members to read.

These responses led the author to question the validity in training students to use expensive software products when the majority of engineers working in industry only have access to Microsoft Office. Consequently, a decision was made to abandon the use of MATLAB for teaching numerical methods applied to heat transfer and instead explore the use of Microsoft Excel. The author was teaching heat transfer using a textbook by Holman [2] that included an appendix outlining the use of Excel for the solution to heat transfer problems. The use of Excel had been resisted in earlier years as the author had viewed Excel as being somewhat "unscientific," and engineering students should be exposed to more advanced software applications. Despite this original view, after teaching heat transfer using MATLAB for four years it was decided to transition to Excel.

Numerical Methods using Excel

As mentioned, the nucleus for this method originated with the textbook by Holman [2], and the material provided by Holman was expanded each year that the course was taught using student feedback. The method used by Holman involves a finite-differenced form of the heat diffusion equation. The method is two dimensional and steady state, and it is simplified through the use of a uniform grid ($\Delta x = \Delta y$). The method starts by recasting the heat diffusion equation:

$$\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} + \frac{\dot{q}}{k} = 0 \tag{4}$$

applied to an interior node

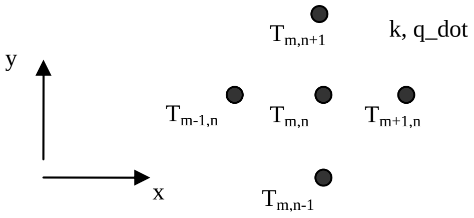


Figure 4. Interior Node Nomenclature

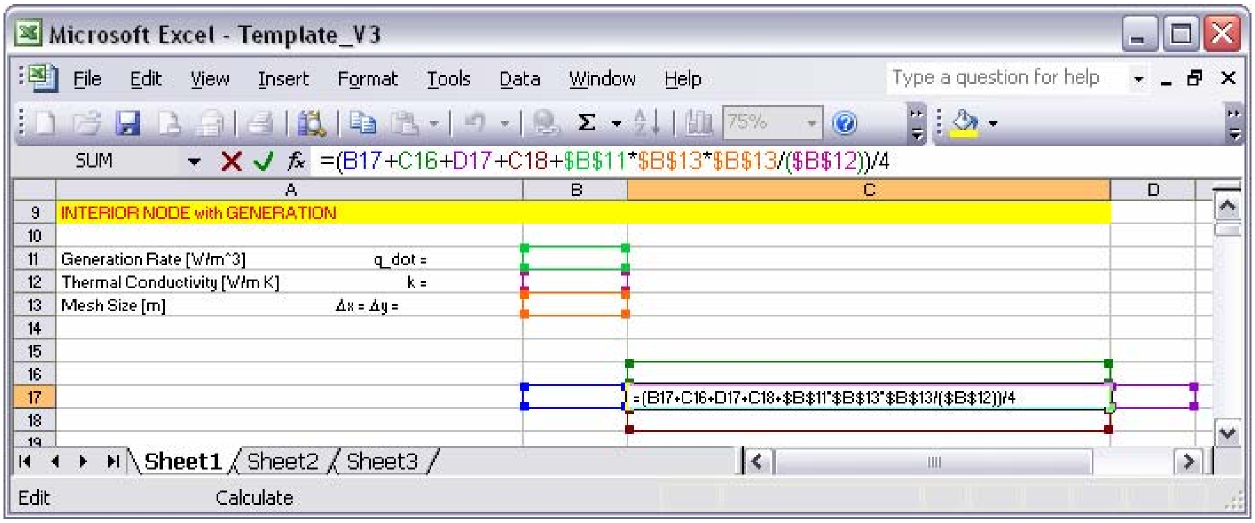
with the second-order partial differential terms written as:

$$\frac{\partial^2 T}{\partial x^2} = \frac{T_{m+1,n} - 2T_{m,n} + T_{m-1,n}}{(\Delta x)^2} \quad ; \quad \frac{\partial^2 T}{\partial y^2} = \frac{T_{m,n+1} - 2T_{m,n} + T_{m,n-1}}{(\Delta y)^2} \tag{5}$$

Introducing the second-order terms, the heat-diffusion equation can be rewritten at an interior node with uniform grid spacing ($\Delta x = \Delta y$) as:

$$T_{m,n} = \left(T_{m+1,n} + T_{m-1,n} + T_{m,n+1} + T_{m,n+1} + \frac{\dot{q}(\Delta x)^2}{k} \right) / 4 \tag{6}$$

The above equation can then be recast into Excel as shown in the screen image in Figure 5.



=(B17+C16+D17+C18+$B$11*$B$13*$B$13/($B$12))/4

A		B	C
INTERIOR NODE with GENERATION			
Generation Rate [W/m^3]	q_dot =		
Thermal Conductivity [W/m K]	k =		
Mesh Size [m]	Δx = Δy =		
			=(B17+C16+D17+C18+$B$11*$B$13*$B$13/($B$12))/4

Figure 5. Interior Node with Generation transcribed to Microsoft Excel

In Figure 5, Cell C17 represents an interior node and it uses the respective data in Cells B11, B12, and B13 for the object's internal generation rate, thermal conductivity, and mesh size. Surface boundary conditions and interior boundaries (different material interfaces) require that an energy balance be performed over a control surface that surrounds the particular node for which the boundary condition is desired. For instance, an external heat-flux boundary condition for a right outer surface or inside left surface:

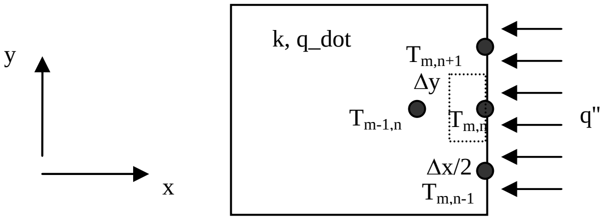


Figure 6. Surface Heat Flux Nomenclature

would be obtained by performing an energy balance on the Control Surface:

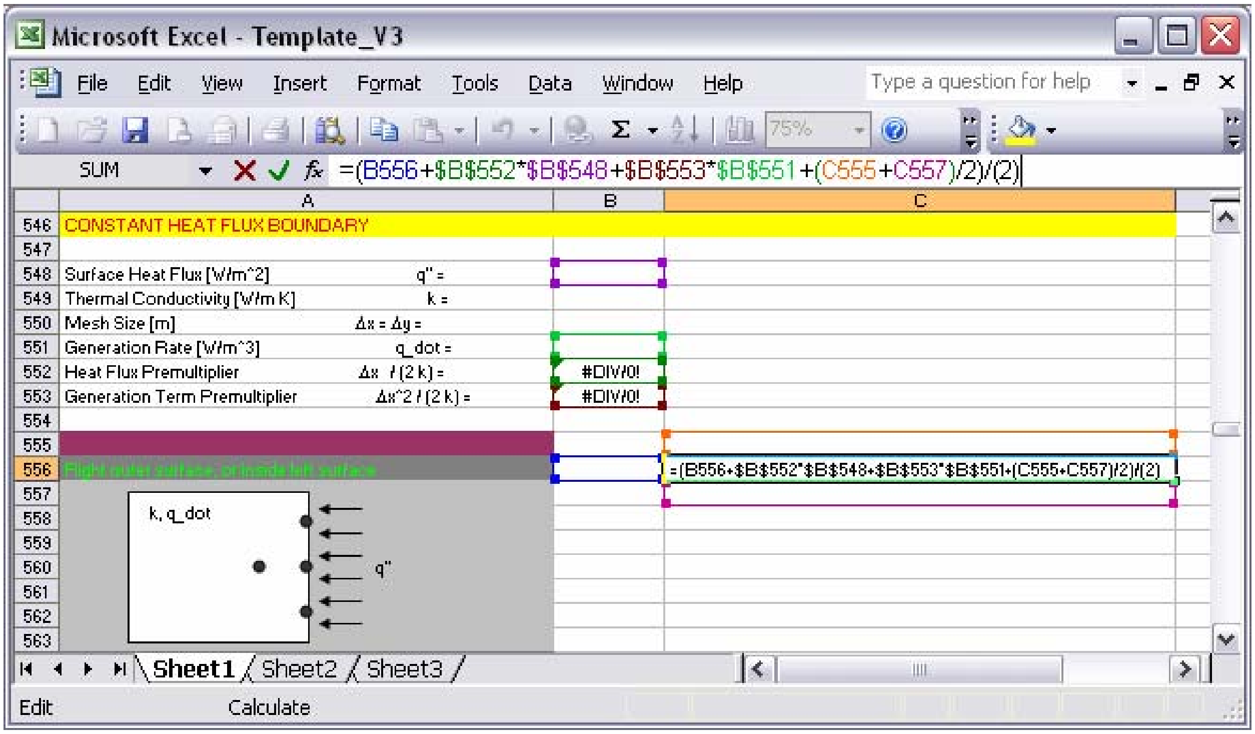
$$\begin{array}{c}\text{Heat in via}\\ \text{Surface Heat Flux}\end{array} + \begin{array}{c}\text{Heat in via}\\ \text{Conduction}\end{array} + \begin{array}{c}\text{Rate of Internal}\\ \text{Energy Generation}\end{array} = 0$$

$$q_{m,n}\Delta y + k\Delta y\frac{T_{m-1,n}-T_{m,n}}{\Delta x} + k\frac{\Delta x}{2}\frac{T_{m,n+1}-T_{m,n}}{\Delta y} + k\frac{\Delta x}{2}\frac{T_{m,n-1}-T_{m,n}}{\Delta y} + \dot{q}\frac{\Delta x}{2}\Delta y = 0 \qquad (7)$$

and if $\Delta x = \Delta y$, the temperature at a particular node location $T_{m,n}$ can be solved for:

$$T_{m,n} = \frac{\Delta x}{2k} q_{m,n} + \frac{\Delta x^2}{4k}\dot{q} + \left(\frac{2T_{m-1,n} + T_{m,n+1} + T_{m,n-1}}{4} \right) \tag{8}$$

The above equation can then be recast into Excel as shown in Figure 7.


Microsoft Excel - Template_V3
File Edit View Insert Format Tools Data Window Help
Type a question for help
75%
SUM
k, q_dot
q"
Sheet1 Sheet2 Sheet3
Edit
Calculate

=(B556+$B$552*$B$548+$B$553*$B$551+(C555+C557)/2)/(2)

	A		B	C
546	CONSTANT HEAT FLUX BOUNDARY			
547				
548	Surface Heat Flux [W/m^2]	q" =		
549	Thermal Conductivity [W/m K]	k =		
550	Mesh Size [m]	Δx = Δy =		
551	Generation Rate [W/m^3]	q_dot =		
552	Heat Flux Premultiplier	Δx / (2 k) =	#DIV/0!	
553	Generation Term Premultiplier	Δx^2 / (2 k) =	#DIV/0!	
554				
555				
556	Right outer surface, or inside left surface			=(B556+$B$552*$B$548+$B$553*$B$551+(C555+C557)/2)/(2)
557				
558				
559				
560				
561				
562				
563				

Figure 7. Surface Heat Flux Boundary Condition transcribed to Microsoft Excel

Cell C556 represents the node for a right outer surface or inside left surface with internal heat generation. Cell C556 uses data input from Cells B548, B549, B550, and B551. Cells for both internal nodes and external boundaries are copied and pasted onto a grid of cells located elsewhere in the spreadsheet that denotes the physical domain of the object being simulated, given both object size and mesh size. The Iteration feature of Excel is then enabled and solution convergence is obtained once the cell values (the temperatures within the object) converge to a certain tolerance. Early versions of the software relied on a word descriptions for each boundary condition. This was a large source of confusion for the students and thus later versions used graphical images, as shown in Figure 7, to convey the nature of each boundary condition.

Example Assignment Problem: The assignment problems given to students while using Excel were two dimensional with or without internal heat generation. Boundary conditions included convection, constant heat flux, insulated boundary, convection and constant heat flux, change of material property (composite solids), radiation, and radiation with convection. Assigned problems often involved trial-and-error solutions where students were required to simulate a number of different scenarios in order to arrive at an optimum configuration. An example of a typical assignment problem is:

A plate of thermal conductivity $k = 15$ W/m K is heated by flow of a hot fluid through channels of width $L = 20$ mm, with $T_\infty = 200^\circ$ C and $h_i = 500$ W/m^2 K, as illustrated in Figure 8. The upper surface of the plate is used to heat a process fluid at $T_{\infty,o} = 25^\circ$ C with a convection coefficient of $h_o = 250$ W/m^2 K. The lower surface of the plate is insulated. To heat the process fluid uniformly, the temperature of the plate's upper surface must be uniform to within 5° C. Determine the maximum allowable spacing, W, between the channel centerlines that will satisfy the specified temperature uniformity requirement. What is the corresponding heat rate per unit length from a single flow channel?

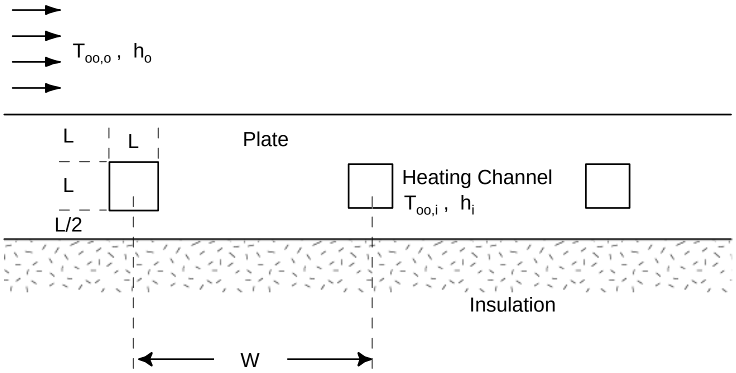


Figure 8. Geometry used in Example Excel Assignment Problem

The above problem was solved by considering one cycle of the plate / heating channel system. A mesh size of 1 mm was prescribed, and simulations were performed by varying W, the spacing between channel centerlines. The result for a 60 mm channel spacing is shown in Figure 9.

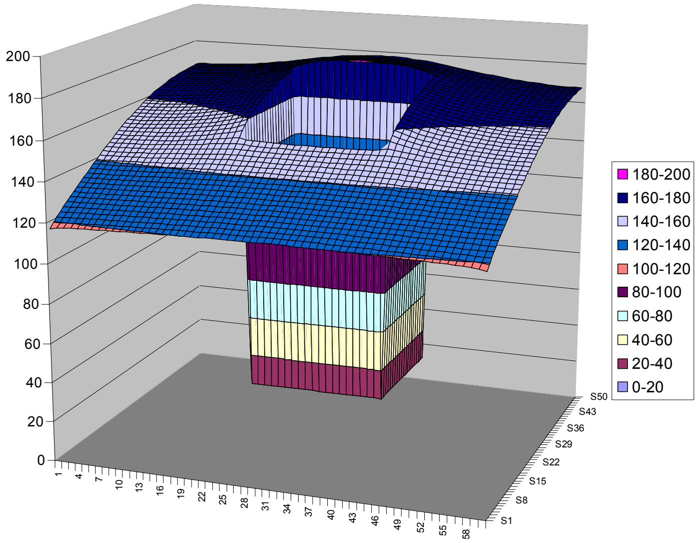


Figure 9. Temperature Distribution for one cycle of Heating Channel (W=60 mm)

Laboratory Integration: The Excel model was also used as part of one of the laboratory experiments in the heat transfer course. The experiment involved measurement of temperatures within a heated aluminum block, and then the Excel model was used in an attempt to quantify the external boundary conditions. Students were asked to develop one model using convection alone, and a second model using both convection and radiation. The problem translates into a minimization problem where the difference between the experimentally measured temperatures and the temperatures predicted by the model are minimized as a function of the surface boundary conditions.

The experimental apparatus, shown in Figure 10, consisted of an aluminum block (k = 204 W/m.K) 12 inches long, 4 inches wide and 6 inches high. The lower half of the block was insulated with fiberglass insulation, and a silicone rubber electrical resistance heater (Omegalux Model SRMU050412P) was been mounted to the bottom surface of the block. The 4 inches wide by 10.16 inches long heater produces 5 Watts / in^2. Heat from the electrical resistance heater conducts up into the block and also into the airspace below the block (the block has been raised off of the table level using metal standoffs). Thermocouple holes were drilled into the block, enabling thermocouples to be placed in the central plane of the block. Eleven thermocouples placed into the holes provide temperature measurements within the block.

The thermocouple signals were fed into homebuilt thermocouple amplifiers, and the signals from these amplifiers were then fed into a National Instruments breakout board. A computer with a National Instruments analog-to-digital card was connected to the breakout board and used to record the voltage outputs from the thermocouple amplifiers. The recorded voltages are processed and displayed using LabVIEW data acquisition and analysis software.

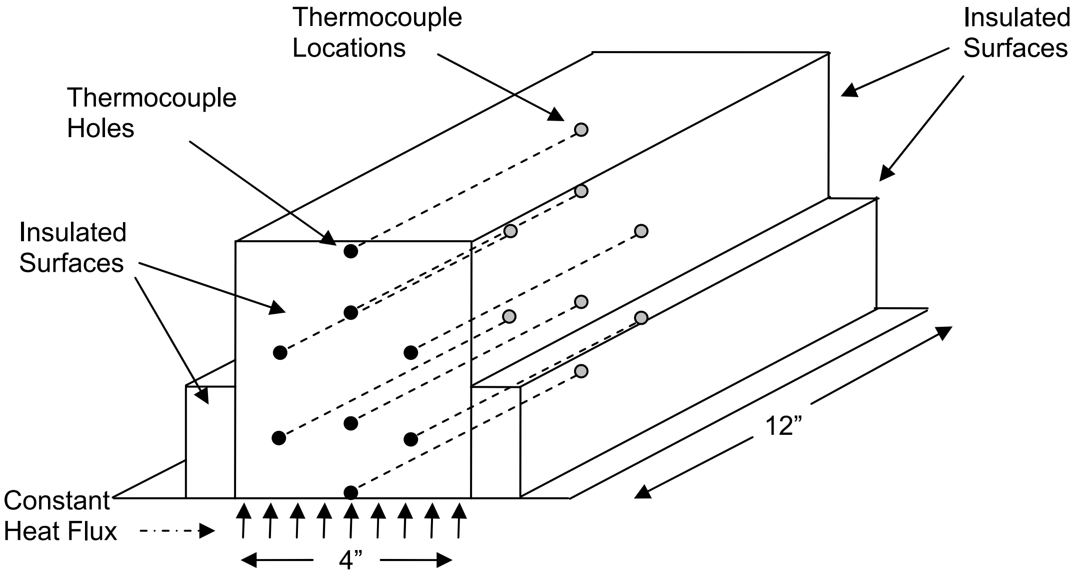
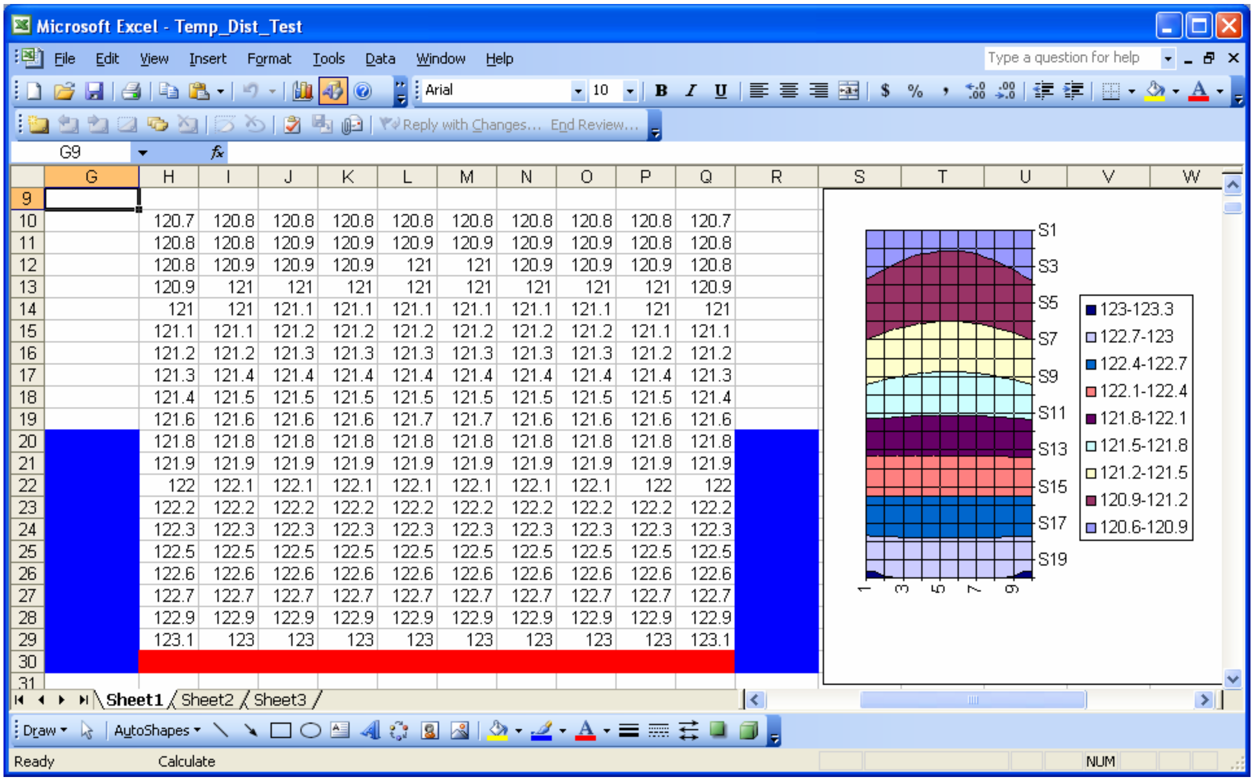


Figure 10. Schematic of Heated Block used during Experiment

An example of the Excel simulation results for the Heated Block experiment is shown in Figure 11.



	G	H	I	J	K	L	M	N	O	P	Q	R
9												
10		120.7	120.8	120.8	120.8	120.8	120.8	120.8	120.8	120.8	120.7	
11		120.8	120.8	120.9	120.9	120.9	120.9	120.9	120.9	120.8	120.8	
12		120.8	120.9	120.9	120.9	121	121	120.9	120.9	120.9	120.8	
13		120.9	121	121	121	121	121	121	121	121	120.9	
14		121	121	121.1	121.1	121.1	121.1	121.1	121.1	121	121	
15		121.1	121.1	121.2	121.2	121.2	121.2	121.2	121.2	121.1	121.1	
16		121.2	121.2	121.3	121.3	121.3	121.3	121.3	121.3	121.2	121.2	
17		121.3	121.4	121.4	121.4	121.4	121.4	121.4	121.4	121.4	121.3	
18		121.4	121.5	121.5	121.5	121.5	121.5	121.5	121.5	121.5	121.4	
19		121.6	121.6	121.6	121.6	121.7	121.7	121.6	121.6	121.6	121.6	
20		121.8	121.8	121.8	121.8	121.8	121.8	121.8	121.8	121.8	121.8	
21		121.9	121.9	121.9	121.9	121.9	121.9	121.9	121.9	121.9	121.9	
22		122	122.1	122.1	122.1	122.1	122.1	122.1	122.1	122	122	
23		122.2	122.2	122.2	122.2	122.2	122.2	122.2	122.2	122.2	122.2	
24		122.3	122.3	122.3	122.3	122.3	122.3	122.3	122.3	122.3	122.3	
25		122.5	122.5	122.5	122.5	122.5	122.5	122.5	122.5	122.5	122.5	
26		122.6	122.6	122.6	122.6	122.6	122.6	122.6	122.6	122.6	122.6	
27		122.7	122.7	122.7	122.7	122.7	122.7	122.7	122.7	122.7	122.7	
28		122.9	122.9	122.9	122.9	122.9	122.9	122.9	122.9	122.9	122.9	
29		123.1	123	123	123	123	123	123	123	123	123.1	
30												
31												

Figure 11. Heated Block Simulation Results

Observations: The use of Excel for application of numerical methods to heat transfer problems was considered to be a success for a number of reasons. Students were able to work on more complicated problems than they were able while using MATLAB given their ability to master the Excel model with little difficulty. Two lectures were used to present the concepts of the finite-difference method, followed by one lecture in which the Excel model was presented. The Excel model was presented by solving a number of analytical problems that the students had already spent time working on in previous assignments. When students observed how simple it was to configure the spreadsheet and to solve a heat transfer problem, many of them reacted with words of excitement that made it clear that they both understood and appreciated the power of the model. The integration of the numerical model into both assignment problems and laboratory experiments provided opportunities for reinforced learning.

Further signs of success started to appear in subsequent years when students would use the Excel package during their capstone design project. Although a number of them were taking a course in the Finite-Element Method, they still reverted back to the Excel model given the ease with which it could be used. Another sign of success came when an engineer from a local company that had sponsored one of the senior-year design projects requested to obtain a copy of the Excel model, citing its ease of use and ability to run using Excel as attractive features.

Conclusions

This paper has discussed the use of Microsoft Excel for teaching numerical analysis applied to a junior-level engineering course in heat transfer. The use of Microsoft Excel was found to be an effective tool given students' previous familiarity with Excel, and the relative ease in which two-dimensional steady-state heat transfer problems can be solved. A further advantage in the use of Microsoft Excel relates to the widespread availability of this application. Microsoft Excel was found to be a useful application for providing a transition between a basic numerical analysis courses and a more advanced CAE courses, helping to fill what is referred to as the "Software Gap."

Although more sophisticated CAE software applications do exist, they have certain drawbacks. One drawback is that CAE applications are often only covered during the senior year of an engineering program which makes it too late for integration into junior-year design-build exercises. More sophisticated CAE software also have restrictive software licensing requirements, forcing the software to be loaded onto designated hardware in specific computer rooms. Although this does not make it prohibitive for students to use the software, it does impede student activity as students have grown accustomed to workplace independence through their use of low-priced laptops.

By introducing students to the Microsoft Excel based application in their junior year, they are able to apply it during design-build exercises encountered both during their junior and senior years. The relatively simple nature of the application coupled with the portable computing advantage make it what can be referred to as a digital age "back-of-the-envelope tool," enabling engineering students to perform quick calculations requiring nothing more than an entry-level laptop. These capabilities fit well within the CDIO initiative, allowing students working in design groups to test out different ideas as they engage in design-build exercises.

References

[1] Ozisik M.N., Heat Transfer – A Basic Approach, McGraw Hill, 1985.

[2] Holman, J.P., Heat Transfer, 9th Edition, McGraw Hill, 2002.

[3] Hamm, S. "More to Life than the Office", Business Week, July 3, 2006.