2013-04-24

A Study into the Behavior of Reinforced-Concrete Columns under Fire Exposures using a Spreadsheet-Based Numerical Model

Richard Lawrence Emberley
Worcester Polytechnic Institute

A Study into the Behavior of Reinforced-Concrete Columns under Fire Exposures using a Spreadsheet-Based Numerical Model

by

Richard Lawrence Emberley

A Thesis
Submitted to the Faculty
of the

WORCESTER POLYTECHNIC INSTITUTE

in partial fulfillment of requirements for the

Degree of Master of Science
in
Civil Engineering

May, 2013

APPROVED:

Dr. Leonard D. Albano, Major Advisor
WPI, Civil and Environmental Engineering

Dr. Robert W. Fitzgerald, Co-Advisor
WPI, Civil and Environmental Engineering

Dr. Tahar El-Korchi, Head of Department
WPI, Civil and Environmental Engineering

Acknowledgements

I would first like to thank Dr. Leonard D. Albano for all his advising and contributions to the development of this thesis and for spending the time to further my knowledge and experience in the area of structural fire engineering. His teaching directly impacted the conception of this thesis' topic and I am extremely fortunate to have so much learned from him.

I would also like to thank Dr. Robert W. Fitzgerald for taking time out of "retirement" to listen to my presentations and provide such valuable insights and suggestions on how to narrow the scope of my project, which areas to focus on, and how best to demonstrate my work. Without his help, I would still be working on coding and would not be finished with this work.

I would like to thank Congyi Qian for his help with the original topic and with testing the model for usability. The topic started out as a class project with frustrations and limitations to what we, as project partners, saw could be done and was further developed into this thesis.

I would like to thank Dr. Jonathan Barnett. His suggestion to complete a thesis started me down this year-long road and I am extremely glad for that suggestion. It has helped increase my knowledge and has increased my excitement for future work the area of structural fire engineering.

Most importantly, I would like to thank my parents, Richard and Sarah Emberley, for everything. They have been my unwavering support over the last 24 years and especially this past year during my thesis work. They have given so much to see me succeed and believed in my abilities and potential when I could not see it in myself. This thesis would not have been completed without you. Thank you!
Everything truly does build on itself.

Abstract

Fire is a significant threat to the structural integrity of buildings. Depending on the architecture of the structure and the intensity and duration of the fire event, structural members may lose strength and stiffness eventually leading to collapse whether by flexural buckling or crushing. The focus of this research is on the behavior and fire performance of reinforced-concrete columns under fire conditions. In order to effectively study column performance with differing loading, aggregate and dimensional characteristics under varying time-temperature curves and fire exposures, a numerical model was constructed in *Microsoft Excel*. The spreadsheet model allowed for complete transparency of the calculations and provided a means to visualize the data in flexible ways. *ANSYS* and several published column furnace tests were used to benchmark the heat transfer and structural analysis portions of the model. One, three and four-sided fire exposures along with the ASTM E119 fire curve and a natural fire curve were used to study latent heating effects, increasing and decreasing eccentricities, moment magnification, and failure modes. Assessments of column structural capacity were performed in accordance with the provisions of *ACI 318*. The completed model served as an effective tool for the thesis and is available to help aid students and engineers investigate the design of reinforced concrete columns under fire conditions through integration the heat transfer analyses and the structural evaluations.

Table of Contents

Table of Figures

List of Tables

1. Introduction

Columns are the primary structural elements that transfer the loads of a building vertically to the foundation. Reinforced concrete is one of the two main material types used for columns. Numerous characteristics of a reinforced-concrete column affect the overall strength of the column: length, load eccentricity, cross-section area, end connections, and concrete and steel strength.

When concrete columns are exposed to fire, the material properties of concrete and the reinforcing steel change as a result of the temperature increases. The decreases in yield strength and modulus of elasticity reduce the overall strength of the column. Once the column strength decreases lower than the applied load, the column will fail either by crushing or by flexural buckling. In structural fire performance testing, a column is placed in a furnace and subjected to a controlled fire while being loaded with a prescribed force. The length of time from the beginning of fire exposure to failure is the fire resistance rating of a column.

Achieving code-specified fire resistance ratings is the goal in prescriptive-based design fire protection. Fire and structural code books such as the *International Building* Code list types of occupancies and the fire resistance ratings various structural members should have. For reinforced-concrete columns, minimum dimensions for concrete cover and column dimensions are listed for specific fire resistance ratings. However, in the fire protection and structural engineering industries, prescriptive-based design is slowly being replaced with performance-based design. The major difference between the two types of design is that performance-based requires the demonstration of fire safety. Numerous methods are available for assessing the fire safety of a design. In the determination of structural fire performance, three levels exist: basic hand calculations, simplified models and iterative calculations of strength, or complex computer models. Each level provides a measure of accuracy to the overall results with computer models providing the most accurate approximations.

Simplified models exist for calculating the fire performance of most structural members including concrete slabs, concrete beams, and steel beams and columns. However, due to the increased complexity of the two-dimensional heat transfer phenomenon and its application to the strength reduction of a reinforced-concrete column, simplified models are rarely discussed in educational textbooks. This omission leads to a gap in the understanding of column behavior in fire conditions for structural engineers. This was a motivating factor in the development of this thesis. The goal of the thesis is to contribute a study on the behavior of reinforced-concrete columns in fire conditions and the influencing factors on column performance as well as provide a tool for further study and education in the analysis of reinforced-concrete columns under fire exposure.

1.1 Objective

The objective of this thesis is to study the behavior of reinforced-concrete columns under fire conditions and to analyze the various characteristics of a reinforced-concrete column and their effects on the overall performance. Also, various phenomena such as slenderness effects and latent heating will be studied for determination of their significance to column fire performance. The studies will all be completed through the construction and use of a numerical calculation model using both heat transfer and reinforced-concrete strength principles.

1.2 Scope of Work

The subsequent list of activities details the scope of this thesis:

- Investigate two-dimensional heat transfer concepts as well as reinforced-concrete column strength principles and column behavior
- Investigate calculation methods for determining the fire performance of reinforced-concrete columns

- Determine the feasibility of performing unsteady, transient two-dimensional heat transfer analyses in a spreadsheet plus coupling these thermal analyses with reinforced-concrete strength equations
- Construct a numerical calculation model for investigating the performance of reinforced-concrete columns under fire conditions
- Research and perform benchmark studies for both the heat transfer and structural analysis portions of the numerical calculation model
- Conduct a study on reinforced-concrete column behavior using the benchmarked numerical calculation model exploring the characteristics of a reinforced-concrete columns and variations in fire exposure that could affect its overall fire performance
- Compile a user manual for the numerical calculation model for further educational and researching use by others

2. Background

This chapter provides an overview of key topics addressed in this thesis as well as a literature review of several papers on the fire performance of reinforced-concrete columns and finite element modeling. Background topics include: column behavior at normal and elevated temperatures, spalling, time-temperature curves, heat transfer, and structural analysis of reinforced-concrete columns.

2.1 Performance-Based Design

In the structural and fire protection engineering fields, design is slowly moving from a prescriptive-based approach to a performance-based approach as the science of structural fire engineering and the technology to analyze and design structures continues to increase. Each design method, prescriptive- or performance-based, has the goal of fire and structural safety. The main difference is how the goal of safety is determined and achieved (Society of Fire Protection Engineers, 2007).

Prescriptive-based design achieves fire safety by mandating compliance to a specific set of design criteria. The design criteria include construction characteristics, minimum and maximum dimensions, and amount of fire protection. Each design criteria is intended to provide the required strength, stiffness, fire protection and constructability features. The assumption is that if the prescribed criteria are met, then structural fire safety will be achieved (Society of Fire Protection Engineers, 2007).

The alternative to prescriptive-based design is performance-based design. In performance-based design, the stakeholders of the project define specific fire safety goals, and the objective of the design is to meet those goals through engineering analyses. The role of the engineer is not to design a building based on a set of standard dimensions and loading properties but to demonstrate through the use of engineering tools such as computer programs and accepted design methods that the proposed design meets the fire safety goals and provides adequate protection (Society of Fire Protection Engineers, 2007).

In prescriptive-based design, structural members only need to meet fire resistance ratings as specified in the governing code books, such as the *International Building Code (IBC)*. Codes such as the

IBC list minimum values for structural member dimensions in order to achieve specific fire ratings. No structural design for fire conditions is necessary beyond that point. However, for performance-based design, certain fire safety goals may dictate that the structural members continue to provide strength well past the fire being extinguished. In these cases, natural fire curves are used in the analysis of structural members since the cooling portion of the fire curve could be important to the structural stability (Dimia, Guenfoud, Gernay, & Franssen, 2011).

Numerous applications exist for assessing the safety of a design. These methods range from hand calculations to complex computer programs. One of the first and most important steps in performance-based design is to determine the level of analysis that is best for the given scenario (Fitzgerald, 2004).

2.2 Level of Performance Evaluation

Fitzgerald lists three levels in performance evaluation of a given fire design scenario. Each level is based on the degree of understanding and the depth of study and analysis. Level 1 is a coarse understanding of the problem, and analysis is done on the large-scale with answers of an order of magnitude. This allows the engineer to get a quick answer for context of the broad spectrum of the problem (Fitzgerald, 2004).

The second level is a deeper understanding of the problem. This involves a detailed knowledge of the fundamentals of the problem being studied. Answers are usually more accurate and in-depth. This level of analysis takes longer as it is more involved. Level 3 is the deepest understanding of the problem and its context. This involves the use of sensitivity analyses to determine which variations most affect the performance of the system and thus how to best protect the system (Fitzgerald, 2004).

The same evaluation levels for fire protection performance can be used in the context of structural analysis, and more specifically in the area of structural fire engineering design for reinforced-concrete column. Level 1 is a basic order of magnitude answer to the problem. This could be a simple “back of the envelope” hand calculation to gain perspective of the problem. Level 2 is the deeper understanding. This could be the use of a spreadsheet to facilitate iterative hand calculations to get an approximate fire performance for the column. Level 3 is the deepest understanding. This could be the integration of extensive material data with sophisticated finite element analysis to determine a more accurate fire performance.

2.3 Normal Column Behavior

In order to understand column behavior during fire conditions, column behavior under normal temperature conditions must first be studied. In particular, the column failure modes and reinforced-concrete column strength equations are reviewed.

2.3.1 Column Failure Modes and Length Effects

A column can collapse in one of two failure modes. The first failure mode is crushing. This occurs when the applied axial force (P) is greater than the column axial strength (P_n). The second failure mode is flexural buckling. This failure mode occurs when the applied moment on the column is greater than the moment resisting strength of the column. An example of an applied moment would be a force (P) applied at an eccentricity (e) from the plastic centroid. This moment may cause a buckling failure; however, this is not the greatest moment on the column. The greatest moment occurs as a result of secondary bending moment that develops due to flexural deformation of the column which increases the maximum moment on the column. These secondary moments are commonly called $P\Delta$ effects due to the increased moment caused by lateral deflection Δ of the column. This is illustrated in Figure 1.

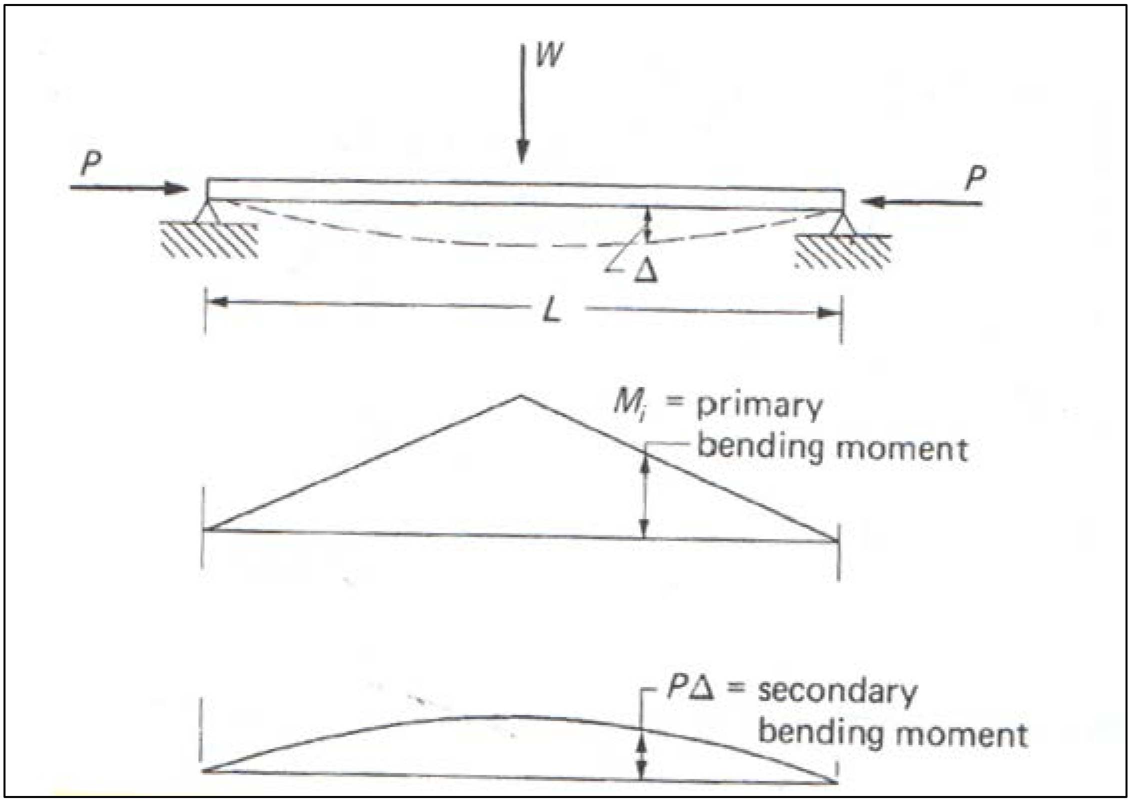

W
P
P
Δ
L
M_i = primary bending moment
$P\Delta$ = secondary bending moment

Figure 1: Secondary Bending Moment Effects (Wang, Salmon, & Pincheira, 2007)

Columns with a low slenderness ratio are prone to crushing failure where columns with a high slenderness ratio are more likely to experience a buckling failure. The slenderness ratio given by kL/r where kL is the column's effective length and r is the column's radius of gyration (Wang, Salmon, & Pincheira, 2007). In order to determine which failure mode will occur first in a column, the axial and moment resisting strengths of the column need to be determined. This is done through the application of column strength equations.

2.3.2 Reinforced-concrete Column Strength Equations

Theoretically, columns can be loaded either concentrically or eccentrically. However, due to errors in construction, material variability, and numerous other factors, columns must be designed for some minimum eccentricity (Wang, Salmon, & Pincheira, 2007). The 2005 ACI Code recommends the following minimum eccentricity to be used in the design of reinforced-concrete columns:

$$e = 0.6 + 0.03h \text{ ; } \textbf{(Equation 2.1)}$$

where e is eccentricity (inches) and h is the minimum dimension of the column (inches). An eccentric axial force produces a moment which subjects the column to bending as well as axial forces. Both force effects must be analyzed to determine whether the behavior of the section is compression-controlled or tension-controlled. The first step is to determine the eccentricity at which balanced strain condition occurs.

2.3.2.1 Balanced Strain Condition

Balanced strain condition refers to the loading scenario in which the axial load is at an eccentricity that causes the concrete to reach a compressive strain of 0.003 and the steel reinforcement to reach a strain of $\frac{f_y}{E_s}$, where f_y is the yield strength of steel and E_s is its modulus of elasticity. This means that the concrete crushes and the steel yields simultaneously, resulting in the largest axial load (P_b) and moment resistance (M_b). Knowing the balanced eccentricity (e_b) allows the failure mode to be determined for any given column and loading. If the eccentricity (e) is greater than e_b, then the steel will reach its strain condition before the concrete. This will result in a flexural failure. If, however, e is less than e_b, then the concrete will crush before the steel yields which is a crushing compression failure of the column. Column behavior can be visually described using a strength interaction diagram, as shown in Figure 2. Figure 2 is a graph of the nominal axial strength of a column versus the nominal bending strength. The point of greatest moment resistance is the balanced failure point. Any combination of load and eccentricity above that line yields a compression-controlled section, and any combination below the line is a tension-controlled section (Wang, Salmon, & Pincheira, 2007).

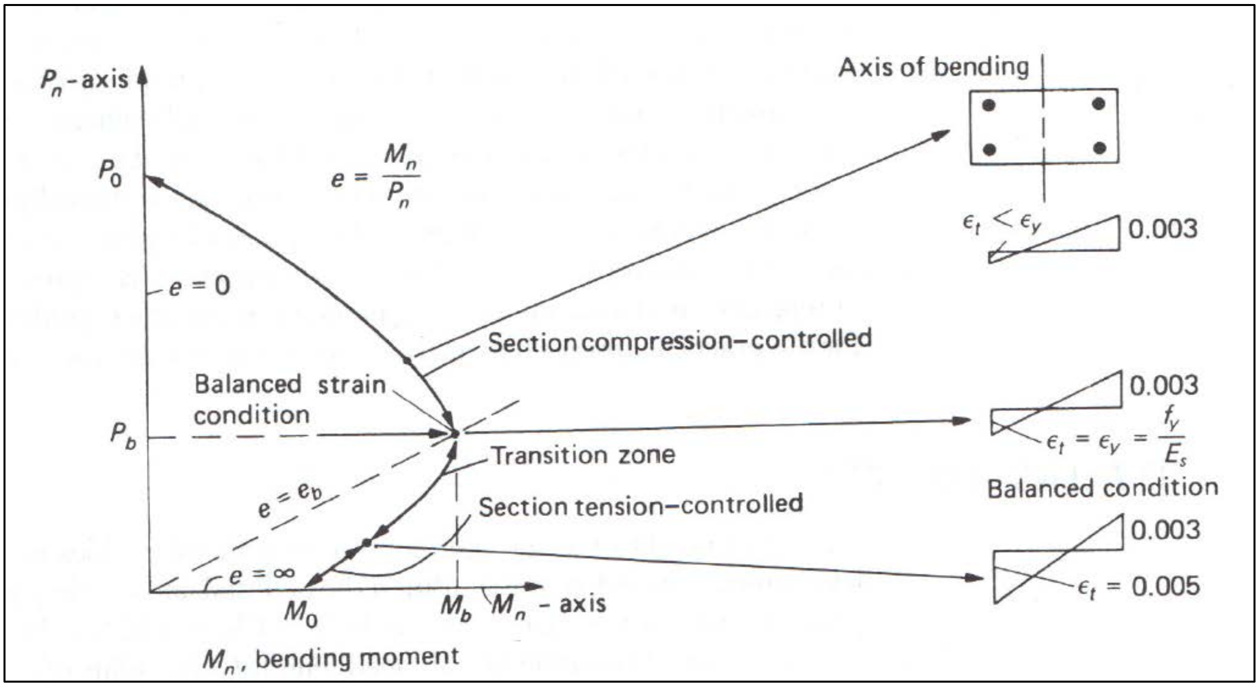


Figure 2: Strength Interaction Diagram (Wang, Salmon, & Pincheira, 2007)

Once the balanced strain condition along with its axial load (P_b) and eccentricity (e_b) are calculated, the appropriate methods for determining the nominal strength of the reinforced-concrete column can be selected using the applied eccentricity of the column.

2.3.2.2 Balanced Strain Condition Calculations

The first step in the process of determining P_b and e_b is to determine the distance from the extreme fiber of the concrete strain condition to the neutral axis (x_b). Figure 3 shows a sample cross section, a strain diagram for the balanced condition, and the corresponding internal stresses and forces. From similar triangles,

$$x_b = \frac{0.003d}{\frac{f_y}{E_s}+0.003}\text{ ; } \textbf{(Equation 2.2)}$$

The distance x_b is then used to calculate the stress contribution (C_c) of the concrete from the Whitney Stress block as shown in Equation 2.3.

$C_c = 0.85 f_c' \beta_1 x_b b$; **(Equation 2.3)**

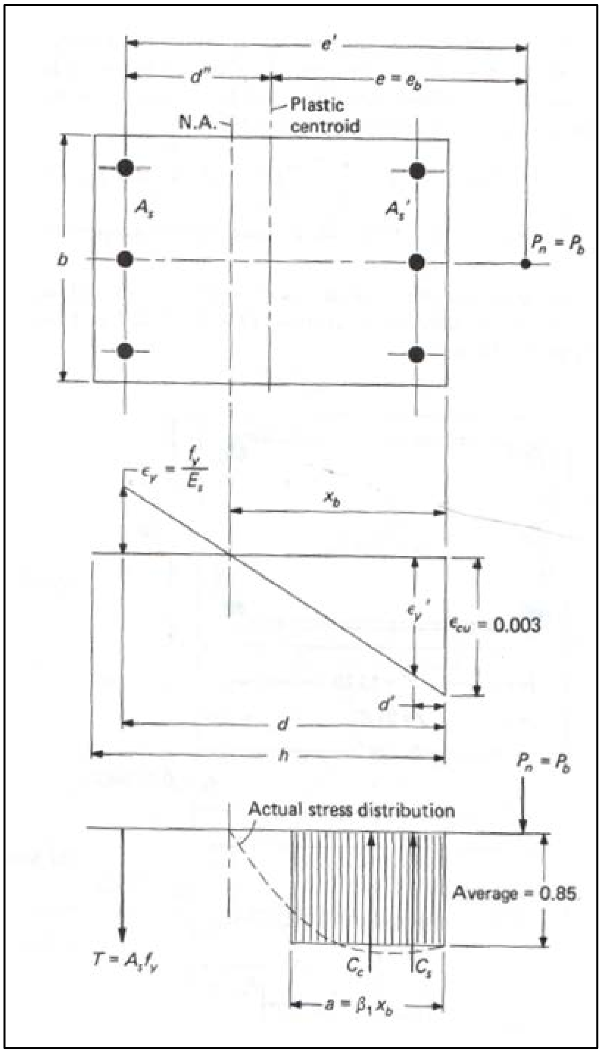


Figure 3: Column and Strain Diagram and Notations (Wang, Salmon, & Pincheira, 2007)

After the resistance force contribution from the concrete has been calculated, the forces from both the tension (T) and compression steel (C_s) can be calculated.

$$T = A_s f_y \text{ ; } \textbf{(Equation 2.4)}$$

$$C_s = A'_s\left(f_y - 0.85f'_c\right) \text{ ; } \textbf{(Equation 2.5)}$$

Summation of the forces yields the balanced strain force (P_b) and with the calculation of moments around the plastic centroid the balanced eccentricity (e_b) can be determined. The location of the plastic centroid can be determined from the resultant of the resisting forces (Park & Paulay, 1975).

$$P_b = C_c + C_s - T \text{ ; } \textbf{(Equation 2.6)}$$

$$P_b e_b = C_C\left(d - \frac{a}{2} - d''\right) + C_s(d - d' - d'') + Td'' \text{ ; } \textbf{(Equation 2.7)}$$

$$d'' = \frac{0.85f'_c bh(d-0.5h) + A'_s f_y(d-d')}{0.85f'_c bh + (A_s + A'_s)f_y} \text{ ; } \textbf{(Equation 2.8)}$$

Once the balanced eccentricity has been found, the nominal strength of the reinforced-concrete column can be calculated.

2.3.2.3 Nominal Strength of Reinforced-concrete Column

The loads of a structure are applied to a column as a force (P) at a certain eccentricity (e) from the plastic centroid of the cross-section (Figure 3). Using the applied eccentricity (e), the nominal strength of the column can be calculated. This is done by first determining whether the column is characterized by compression- or tension-controlled behavior. The following criteria are used to determine the mechanism controlling the section.

$$e < e_b \text{ – compression-controlled}$$

$$e > e_b \text{ – tension-controlled}$$

The processes for calculating both the compression-controlled and tension-controlled section are very similar: the both involve finding the distance from maximum compressive strain to the point where compressive strain transitions to tension strain. The equations for the internal forces in the concrete and the tension steel are similar to those for the balanced strain condition (Equations 2.2-2.7) except that x_b is replaced by the distance x which is the distance from the maximum compressive strain to the transition point. For a compression-controlled section, x will be greater than x_b.

$$C_c = 0.85 f_c' b \beta_1 x$$; **(Equation 2.9)**

$$T = A_s f_s$$; **(Equation 2.10)**

$$f_s = \frac{0.003 E_s (d-x)}{x}$$; **(Equation 2.11)**

C_s remains the same as Equation 2.5. C_c and T are all calculated with the unknown variable x. Then, the forces and their resulting moments are summed around the location of the applied load, a distance (e) from the plastic centroid. Once x is found, the forces Cs, C_c, and T can be determined and then with the summation of those forces, the nominal strength of the section (P_n) can be determined.

For a tension-controlled section, C_c is solved with the depth of the neutral axis, x (Equation 2.8), while T and C_s remain as defined in Equations 2.4 and 2.5 respectively. For the tension-controlled section, x is less than x_b. The distance x is solved by summing the moment around a point on the cross-section. After x is found, the forces C_s, C_c, and T can be determined along with the nominal strength of the cross-section. This solution assumes that the compression steel has yielded: the strain of the compression steel (ϵ_s') is greater than the yield strain (ϵ_y). After x has been calculated, ϵ_s' (Equation 2.12) has to be confirmed as being greater than ϵ_y to make sure the solution is indeed correct. If, however, the compression is found to have not yet yielded, the process can be repeated using the value of ϵ_s' to determine the correct force in the compression steel C_s.

$$e_s' = 0.003\frac{x-d'}{x}; \textbf{ (Equation 2.12)}$$

2.3.3 Summary

The key parameters affecting the strength of the column section are the area (b x h), location of the plastic neutral axis (d''), and the yield strength of both the compression and tension steel. The area of the column and compressive strength of the concrete changes the strength of the concrete portion of the column. The location of the plastic neutral axis changes the eccentricity on the column which could result in increasing or decreasing moments on the column. Finally, the yield strength of the steel affects the overall strength of the compression and tension steel. Changes to these parameters can lead to increasing or decreasing the overall strength of the column.

2.4 Column Behavior at Elevated Temperatures

At elevated temperatures, the values for concrete's material properties change. These property changes lead to changes in strength and heat transfer rates. Several assumptions to simplify the changes over the entire cross-section are made for use in a level 2 type of analysis.

2.4.1 Temperature Effects on Nominal Strength

The load capacity of a reinforced-concrete column will deteriorate based on the reduction of the modulus of elasticity and strength of both the concrete and the reinforcing steel. These losses are a result of the increasing cross-section temperature due to fire exposure. Figure 4 shows the normalized yield strength and modulus of elasticity of steel at elevated temperatures (Society of Fire Protection Engineers, 2008). The equations for these variables can be seen in the Appendix A.

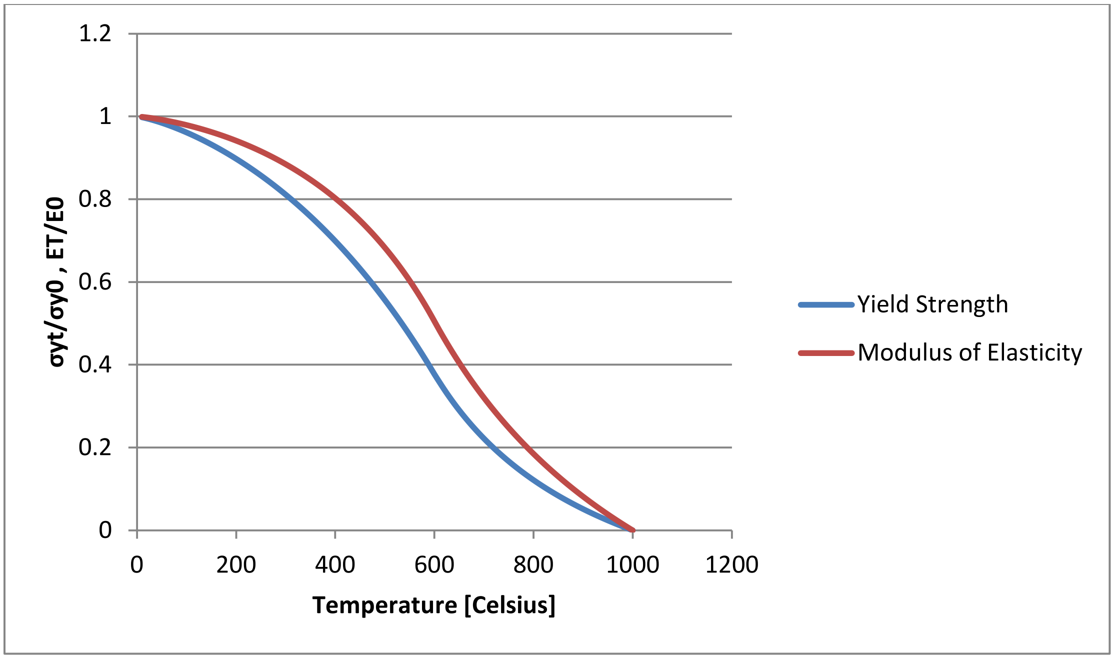


Figure 4: Normalized Steel Properties

These temperature-dependent material properties can be used with the normal temperature design calculations described above to predict column strength as a function of temperature. This provides the means to determine when the nominal strength of the column decreases to a value less than the applied force. At this point failure occurs.

2.4.1.1 Isotherm Simplification Method

In theory, a cross section with various temperature gradients throughout has areas of varying strength due to those temperature gradients. This is due to the fact that concrete and steel vary in strength at elevated temperatures. Each differential area of the cross section contributes strength to the overall strength of the column. For ease of design and analysis, several conservative methods have been recommended to deal with temperature variations across the concrete section: the 500°C isotherm method and the 400/600°C isotherm method.

The 500°C isotherm method assumes that only the concrete section below 500°C contributes strength. Any other portion of the cross section has zero strength. The reinforcing steel has strength based on its temperature at the time of calculation. Cross-section properties such as depth of neutral axis, plastic centroid location, width, depth, and radius of gyration are all calculated based on the shape and area enclosed by the 500°C isotherm (Buchanan, 2002).

The 400/600°C isotherm method uses two isotherms instead of one to approximate the strength of the column. The method assumes that concrete sections below 400°C have 90% strength, those between 400°C and 600°C have 70% strength, and sections above 600°C have no strength. The reinforcing bars are considered to have 50% strength throughout (Buchanan, 2002).

These are both approximate methods because the actual strength of concrete decreases continuously from ambient temperature. Figure 5 is a graph of the compressive strength of siliceous aggregate concrete. The compressive strength actually decreases from ambient strength around 100°C to about 20% of it ambient strength at around 700°C. The isotherm methods try to approximate the strength curve while still allowing for ease of calculations. The red line in Figure 5 is the strength assumption of the 500°C isotherm line, and the blue line is the strength assumption of the 400/600°C isotherm method. Each of these methods assumes that the cross-section remains intact throughout the entire heating and cooling process (Buchanan, 2002) (Society of Fire Protection Engineers, 2008).

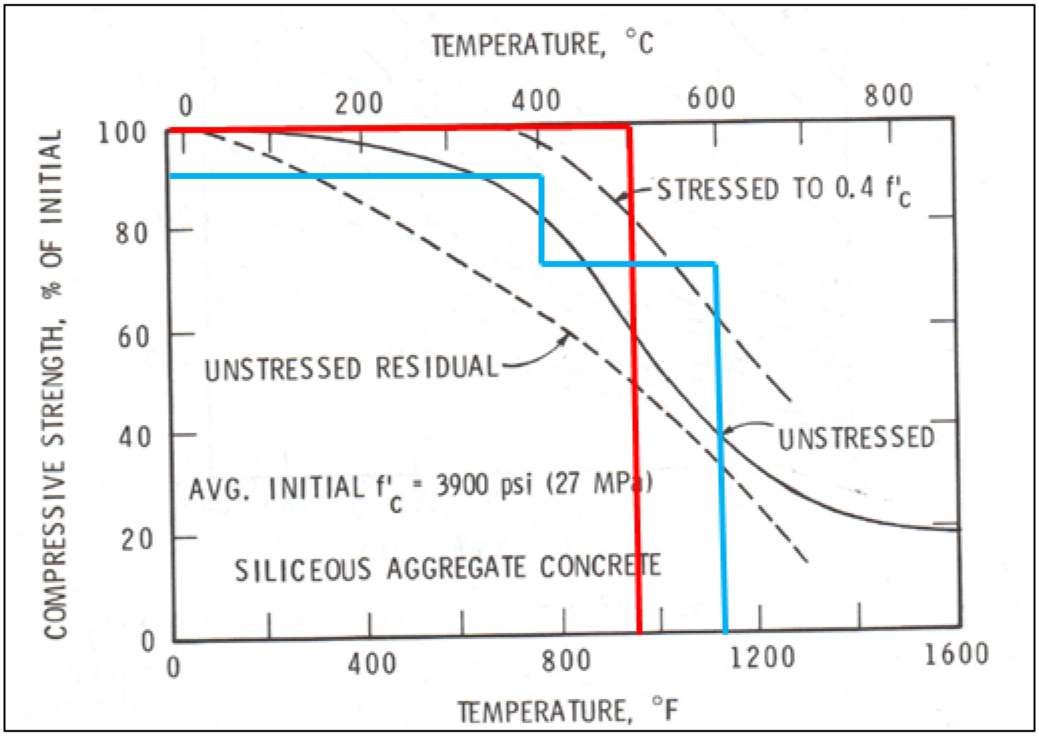


Figure 5: Compressive Strength of Concrete (American Society of Civil Engineers, 1992)

2.4.2. Spalling

Spalling is the process of the column's concrete cover breaking off during fire exposure. Water in the cement paste of the concrete vaporizes due to heating, resulting in increasing pore pressures in the concrete. When the tensile stresses in the concrete from the pore pressure reach the tensile strength of the concrete, sections of the concrete cover will break off resulting in a loss of the cross sectional area and overall cross section strength (Buchanan, 2002).

Several factors including high moisture content, rapid increases in cross section temperature, columns with high slenderness ratios, and large concrete stresses all result in an increased probability of spalling. Most normal strength concrete has been shown to be highly resistant to spalling; however, high-strength concrete is more susceptible to explosive and more frequent spalling due to the concrete properties. High-strength concrete has less pore volume leading to high pore pressures faster. High-strength concrete’s porosity is low leading to less dissipation of the water vapor (Buchanan, 2002) (Society of Fire Protection Engineers, 2008).

Spalling is a complex phenomenon that occurs with concrete. To date, a simplified model for spalling is not available due to the complexity of spalling and the lack of complete understanding of it (Buchanan, 2002). For these reasons spalling was not considered in the column behavior study; though in the design of reinforced-concrete columns for fire conditions, the engineer should investigate and account for this action.

2.4.3 Latent Heating

Latent heating is a significant phenomenon that occurs in concrete when it is subject to a natural fire curve. Due to its low conductivity and low thermal diffusivity, concrete acts like a heat sink (Buchanan, 2002). It takes much longer for a concrete member to rise in temperature than a conductive material such as steel. This also means that heat is not dissipated as quickly in concrete. In the cooling phase of a natural fire curve, even though the gas temperature is decreasing, the interior of the concrete member could still be increasing in temperature due to the still high temperatures in the cross-section. The outer portion of the column cools first while the interior is still increasing in temperature. This can be a significant problem because concrete does not fully recover its initial strength even when cooled back to ambient conditions. With latent heating, the cross could still be losing unrecoverable strength when externally everything appears safe (Dimia, Guenfoud, Gernay, & Franssen, 2011). Thus it is imperative to utilize natural fires as well as standard fire curves when analyzing the performance of concrete members in fire conditions.

2.4.4 Plastic Centroid Movement

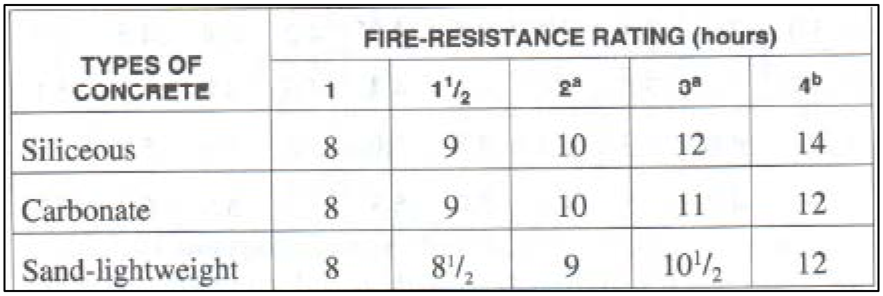
The plastic centroid is the location of centroid of the resistance forces, tension and compression steel and concrete section, in a column (Park & Paulay, 1975). The eccentricity of a column as shown in Section 2.3.2.2 is measured from the plastic centroid. Under normal conditions, the plastic centroid for a symmetrically reinforced column is located directly in the center of the column. However, in fire conditions, the plastic centroid shifts depending on the exposure. For one- and three- sided fire exposures, the column strength will deteriorate asymmetrically leading to a shift to either the right or the left of the original plastic centroid location. The shifting of the centroid leads either to an increase or decrease in the eccentricity, which either increases or decreases the applied moment on the column.

2.4.5 Slenderness Ratio

The slenderness ratio of a column is the ratio of the effective height (kL) to the radius of gyration (r). Columns with larger slenderness ratios are more susceptible to flexural buckling failure. Since the radius of gyration is based on the width and depth of a column's cross-section, the temperature gradients and assumptions made in the isotherm methods will increase the slenderness ratio throughout a fire exposure. This means that a column will be more susceptible to a flexural buckling failure later in a fire exposure than at ambient conditions.

2.5 Time-Temperature Curves

Furnace testing with standard fire curves has been used to determine the fire performance of materials and building for years (Buchanan, 2002). In structural fire performance testing, a building element, such as a column, is placed in a furnace and subjected to a controlled fire while being loaded to prescribed force. When the element fails specified criteria, a fire resistance time is given to the element. Structural members are required to have resistance times greater or equal to prescribed fire-resistance ratings in the governing building code in which the structure is being built. A frequently adopted building code is the *International Building Code* (*IBC*). An *IBC* fire-resistance rating is shown in Figure 6.



TYPES OF CONCRETE	FIRE-RESISTANCE RATING (hours)				
	1	$1^1/_2$	2[a]	3[a]	4[b]
Siliceous	8	9	10	12	14
Carbonate	8	9	10	11	12
Sand-lightweight	8	$8^1/_2$	9	$10^1/_2$	12

Figure 6: Minimum Column Size (Inches) for Specified Fire Resistance Ratings (International Code Council, 2009)

Specific descriptions of the standard fire time-temperature curves are within the purview of the testing and national agencies; however, two time-temperature curves are used universally: ASTM E119 and ISO 834 (Buchanan, 2002). Even though these are two different time-temperature curves from two different standards, both are very similar. The subtle differences can be seen in Figure 7: the ASTM E119 fire increases in temperature quicker than the ISO 834, but the ISO 834 curve has higher temperatures overall.

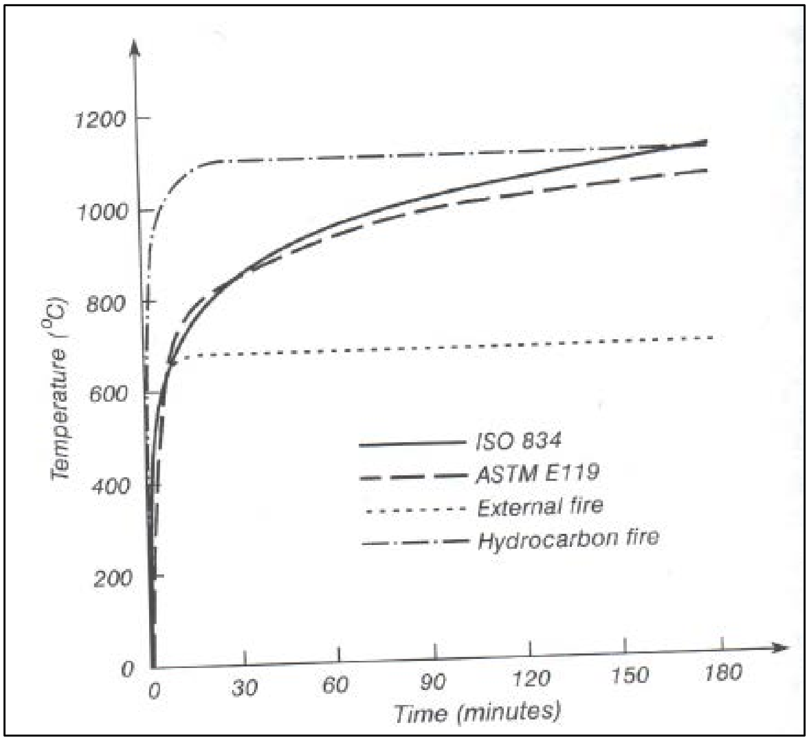


Figure 7: Standard Fire Curves (Buchanan, 2002)

Each of these standard fires can be described by an equation or an approximate equation as in the case of the ASTM E119 fire curve. Equations 2.13 and 2.14 were used in this document to study the behavior of the columns under the fire loads.

ISO 834: $T = 345\log_{10}(8t+1) + T_0$; **(Equation 2.13)**

ASTM E119: $T = 750\left[1 - e^{-3.79553\sqrt{t_h}}\right] + 170.41\sqrt{t_h} + T_0$; **(Equation 2.14)**

where T is temperature in Celsius, t_h is time in hours, t is time in minutes, and T_0 is ambient temperature in Celsius (Buchanan, 2002).The furnace is set to follow these time curves, and structural element or material is subjected to the resulting temperatures.

Although standard fires are widely used in materials testing, they are not an accurate representation of an actual fire. Actual fires have a period of growth which is followed by a period of full intensity. Then, once the fire becomes fuel-controlled, the temperatures drop as the fire decays in a process called burnout. A graphical representation of an actual fire can be seen in Figure 8.

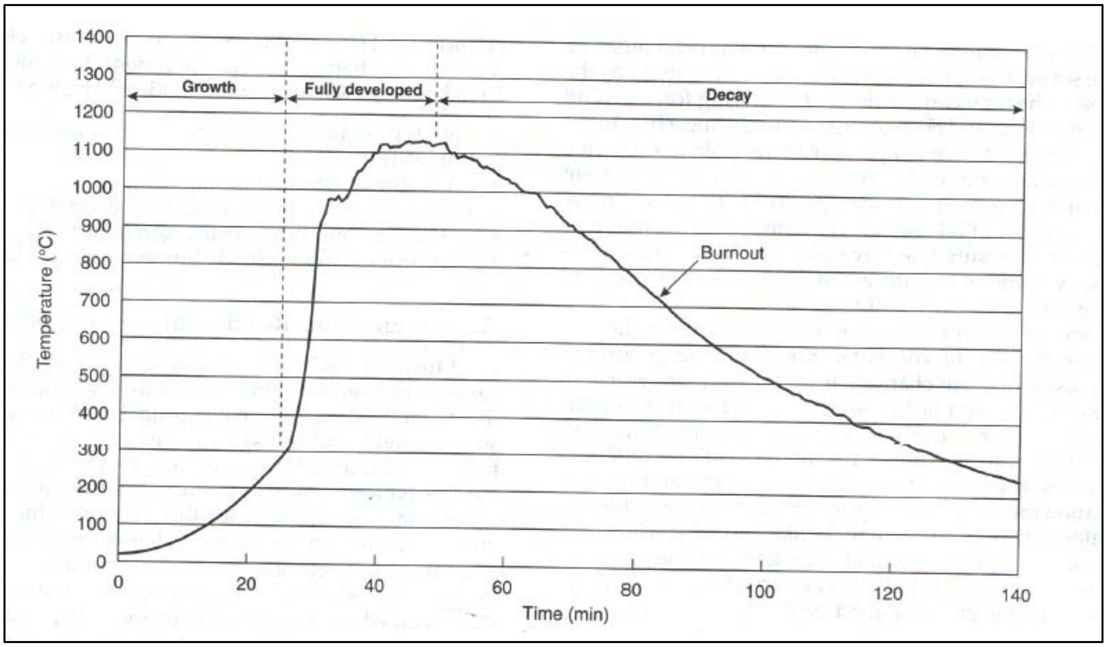


Figure 8: Natural Fire Curve (Society of Fire Protection Engineers, 2008)

One way to approximate a natural fire is through the use of the parametric fire equations from Eurocode 1 (Franssen, 2000). The parametric fire equation roughly follows the ISO 834/ASTM E119 standard fire curve until a specified heating duration has been reached. Then the fire curve undergoes decay using a series of cooling phase equations until the temperature reaches ambient conditions. The following equations (2.15 to 2.19) describe the parametric fire curve's heating and cooling phases (Franssen, 2000).

Heating:

$$T_g = 1325(1 - 0.324e^{-0.2t^*} - 0.204e^{-1.7t^*} - 0.472e^{-19t^*}) \text{ ; } \textbf{(Equation 2.15)}$$

$$t^* = \Gamma t \text{ ; } \textbf{(Equation 2.16)}$$

Cooling:

$$T_g = T_{max} - 625(t - t_d) : \text{for } t_d \leq 0.5 \text{ ; } \textbf{(Equation 2.17)}$$

$$T_g = T_{max} - 250(3 - t_d)(t - t_d) : \text{for } 0.5 \leq t_d \leq 2.0 \text{ ; } \textbf{(Equation 2.18)}$$

$$T_g = T_{max} - 250(t - t_d) : \text{for } 2.0 \leq t_d \text{ ; } \textbf{(Equation 2.19)}$$

where T_g is the gas temperature, t is time in hours, Γ is an opening and enclosure variable, and t_d is the heating duration. If Γ is taken as 1, the curve approximates the ISO 834 well (Dimia, Guenfoud, Gernay, & Franssen, 2011). With the fire curves listed, the temperature gradients within the cross-section can be found for a given time-temperature curve using the principles of heat transfer.

2.6 Heat Transfer

Heat transfer from the column surroundings through the reinforced-concrete column cross-section is the most important step in determining the reduction of nominal strength in a reinforced-concrete column.

The heat transfer analysis used in this thesis incorporates the three mechanisms of heat transfer: convection, radiation, and conduction. Convection as well as radiation are present between the hot gases surrounding the column and the column surface. Radiation also occurs between the column and other boundaries in the area, such as the walls, floor and ceiling. Conduction occurs within the column itself. Once the boundary layer of the column is heated by convection and radiation, conduction transfers the thermal energy throughout the cross section from regions of higher to lower temperatures (Kreith & Bohn, 2001).

The proceeding sections outline the general heat transfer processes used to calculate the temperature gradients throughout the cross section of the column for specified time-steps.

2.6.1 Radiation-Convection

For this thesis, the model is simplified. In a normal fire scenario, the column in question would be subjected to both radiation and convection heat transfer. Convection would be from the hot gases in the room, and radiation would be from the fire, the hot gases and the surrounding objects. True radiation heat transfer is based on the distance from the heat source to the target object and the heat flux from the heat source. This type of radiation heat transfer is extremely complex to model since numerous sources of radiation are always present plus the fact that the target objects itself gives off radiation. The result is a complex radiation network.

However, for a simplified model, only radiation from the hot gases is commonly used in the heat transfer calculations. A radiation coefficient is combined with a convection coefficient to determine the heat flux from the surrounding gases to the column. Equations 2.20 to 2.23 show the combined heat flux ($q^{"}$) equation along with each of the variables comprising each equation.

$$q^{"} = q_c^{"} + q_r^{"} \text{ ; } \textbf{(Equation 2.20)}$$

where $q_c^{"}$ is the convective heat flux and $q_r^{"}$ is the radiation heat flux.

$$q_c^{"} = \overline{h_c}(T_s - T_\infty) \text{ ; } \textbf{(Equation 2.21)}$$

where $\overline{h_c}$ is the convective heat transfer coefficient, T_s is the is the column section surface temperature, and T_∞ is the hot gas temperature. $\overline{h_c}$ is usually taken to be 25 W/m^2°C (Malhotra, 1982).

$$q_r^{"} = \overline{h_r}(T_s - T_\infty) \text{ ; } \textbf{(Equation 2.22)}$$

where $\overline{h_r}$ is the radiation heat transfer coefficient.

$$\overline{h_r} = \frac{\epsilon_1 \sigma (T_s^4 - T_\infty^4)}{T_s - T_\infty} \text{ ; } \textbf{(Equation 2.23)}$$

where $\epsilon_1$is the emissivity of the boundary areas and hot gases and σ is the Stefan Boltzmann constant (5.67*10^{-8} Wm^{-2}K^{-4}). For ϵ_1, 0.5 is an accepted value for furnace testing (Malhotra, 1982).

2.6.2 Conduction

Heat is transferred through the cross section of the column by conduction. Conduction and its governing differential equation can be easily explained by the conservation of energy and control volume principle. Heat into a control volume plus heat generated in a control volume must, in order to conserve energy, equal the heat that leaves a control volume plus the heat that is stored in a control volume. Equation 2.24 is the two-dimensional differential equation for conduction.

$$\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right)+\frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right)+Q=\rho c\frac{\partial T}{\partial t}\ ;\ \textbf{(Equation 2.24)}$$

where $\frac{\partial}{\partial x}\left(k_x\frac{\partial T}{\partial x}\right)$ is heat transfer in the x-direction, $\frac{\partial}{\partial y}\left(k_y\frac{\partial T}{\partial y}\right)$ is heat transfer in the y-direction, Q is heat generation in the control volume, and $\rho c\frac{\partial T}{\partial t}$ is the heat stored in the control volume. The variable *k* is the material's conductivity, *ρ* is the material's density, *c* is the material's specific heat, *T* is temperature, and *t* is time (Malhotra, 1982). The values for of *k*, *ρ*, and *c* are all temperature dependent material properties. For concrete, as the aggregate changes, the temperature equations for determining the material properties change as well. The equations for each aggregate type and material properties are all list in the Appendix.

Since no closed-form solution is available to solve the conduction differential equation for an entire area, a numerical method has to be used to approximate the answer. Discretization is the process of breaking up a continuous area into an area comprised of smaller subsets. The approximate solution to the differential equation can then be made for the smaller known area of subsets and, an approximate solution for the continuous area can be determined.

2.7 Literature Review

The following section is a summary of the important papers on the subject of both reinforced-concrete column fire performance and numerical modeling of heat transfer differential equations. The works listed are those that contributed significantly to the thesis.

2.7.1. Allen & Lie, 1974

This paper outlines the process of developing a numerical procedure to calculate the fire performance of reinforced-concrete columns. The authors describe material property changes due to elevated temperature well. One of most important details about this article is that the authors developed their own model and compared it against actual column furnace tests. The furnace tests used both standard and natural fire curves (Allen & Lie, 1974).

2.7.2 Lie & Irwin, 1993

The authors of this paper developed a model to analyze the fire performance of reinforced-concrete columns. One of the unique difference between this paper and the work of Allen & Lie (1974) is that this paper incorporated a heat transfer term to account for moisture in the concrete. The equations used in this paper for heat transfer calculation were used in this thesis except for the moisture content term (Lie & Irwin, 1993).

2.7.3 Yao & Tan, 2003

This paper was significant because it contributed experimental tests with which to verify the calculation model developed for this thesis. The authors studied the experimental data and compared the results to those obtained from computer program *SAFIR*. The paper conducted several case studies to cover a wide range of column sizes, connections and loadings (Yao & Tan, 2003).

2.7.4 Schumack, 2004

This article described the process of implementing finite element heat transfer equations into a spreadsheet program. This article showed how to display a two-dimensional area of nodes on a contour plot in *Microsoft Excel*. The use of the contour is extremely useful for visualization of the temperature isotherms of a cross-section (Schumack, 2004).

2.7.5 Besser, 2002

This article as with Schumack (2004) demonstrated how to implement heat transfer equations into a spreadsheet program. The author mentioned that spreadsheets could be used for transient heat transfer if the spreadsheet was constructed properly (Besser R. S., 2002). This article led to the idea of using macros to run multiple time-steps of transient heat transfer.

2.7.6 Dimia, Guenfoud, Gernay, & Franssen, 2011

This paper was a key research document in the understanding of concrete behavior during the cooling stage of a fire. The authors summarize the key principles behind concrete column collapses during the cool down portion of the fire as well as provide methods for analysis of those columns. The authors study included use of the computer program *SAFIR* to analyze numerous column characteristics and to provide evidence for the latent heating phenomenon and subsequent column collapse. The length of the duration of the fire was shown to be key in predicting when the fire would collapse. Graphs of the load and time to failure of a test column with curves of each burning duration were compiled. This type of graph could be used to determine time to failure of a column with a given load if the burning duration was known (Dimia, Guenfoud, Gernay, & Franssen, 2011).

2.7.7 Dotreppe, et al., 1996 , Woollerton & Lie, 1988

Both the articles developed their own models to predict the failure time of reinforced-concrete columns exposed to fire. They both also compared their models to column furnace test experiments. These experiments and models were used to benchmark the model in this thesis (Dotreppe, et al., 1996) (Woollerton & Lie, 1988)

2.7.8 Buchanan, 2002

Buchanan’s textbook *Structural Design for Fire Conditions* was used as the starting point for the thesis. Buchanan describes various methods for designing different structural members for fire conditions. For concrete columns, the 500°C and 400/600°C isotherm methods are both described as conservative options for column design. Generic methods of design based on experimental results of thickness of cover were list as well. Reinforced-concrete columns are not described in much detail due to the complexity of the problem and the nature of the book being a textbook (Buchanan, 2002).

2.7.9 American Society of Civil Engineers, 1992

This book like Buchanan’s textbook details the various methods of analysis and design for structural fire safety. This book was mainly used as a reference for the various thermal and mechanical properties of reinforced-concrete. The authors compiled the properties of several different kinds of aggregate including carbonate, siliceous, and expanded shale. Equations for the properties are listed for temperature ranges of 0°C and higher. These equations can be seen in the Appendix (American Society of Civil Engineers, 1992).

2.7.10 Malhotra, 1982

Malhorta's book describes the governing differential equation for conduction extremely well. Malhorta also details the combined equation for convection and radiation heat transfer while providing recommended values and appropriate approximations for the various variables (Malhotra, 1982).

2.7.11 Summary

The information gathered from these articles, papers, and books on fire performance modeling of reinforced-concrete columns, column behavior, and numerical modeling were all used to develop a calculation model for investigating the strength of concrete columns under elevated conditions, including transient heat transfer analyses. The model was then used to study the effects of fire exposure reinforced-concrete columns: column behavior at elevated temperatures, the shifting of the plastic centroid due to uneven heating, and the increase of the column slenderness ratio due to the effective loss of the section. The following chapter describes how the calculation model was constructed and what columns and exposures were analyzed for verification and study purposes.

3. Methodology

This chapter outlines the methodology used in this thesis. The methodology includes the numerical analysis of the heat transfer equations, a description of the fire exposure analyzed, and the process used to calculate length effects in the columns. Also included in this section is the process used to implement in *Microsoft Excel* the heat transfer and structural equations presented in the background. Finally, the columns and techniques used to verify the model and study reinforced-concrete column behavior under fire exposure.

3.1 Calculation Method Selection

A spreadsheet, particularly *Microsoft Excel*, was selected for a number of reasons. The first reason is that a spreadsheet allows for transparency in the calculation process. The equations used in the model are all visible to the user in either the main screen or in the macro. In a finite element program or mathematical solver such as *ANSYS* or MATLAB, the calculations are done in the program so the user does not see the nodal calculations at each time step. The transparency of the program allows for education of the user into the specific process behind reinforced-concrete column design for fire conditions.

The second reason is that with the use of macros, the spreadsheet can be designed specifically to what the user or the designer needs. For this thesis, the main need was the ability to connect both the heat transfer and structural analysis portions of the model.

The third reason is that spreadsheets are widely known and are simple to use. Most computers on the market today are equipped with a spreadsheet program and most people have the basic knowledge of spreadsheets. In order to use a program like *ANSYS*, the user needs to have significant knowledge of the program in order to efficiently gather data and transfer it to a spreadsheet for data processing. In a spreadsheet model, the data is already in a spreadsheet and readily available to organized into graphs, tables or other presentation methods. These are the main reasons why a spreadsheet was chosen for the analysis of the behavior of reinforced-concrete columns under fire exposures.

3.2 Numerical Analysis of Conduction Equation

Three methods are available for discretizing the conduction differential equation: finite difference, finite element, and control volume. This thesis will use the control volume approach as it utilizes the energy balance to conserve energy over the control volume. Because energy is conserved, the approximate answer to the differential equation can be determined with a coarser volume grid (Kreith & Bohn, 2001). This is convenient when using a spreadsheet to solve the governing equations as finer grids requires more cells and thus longer computational time to solve the problem. Using fewer cells leads to faster computations time while using less of the spreadsheet.

The first step in the discretization process is to break up the cross sectional area into a series of nodes and control volumes. A sample control volume and node arrangement can be seen in Figure 9.

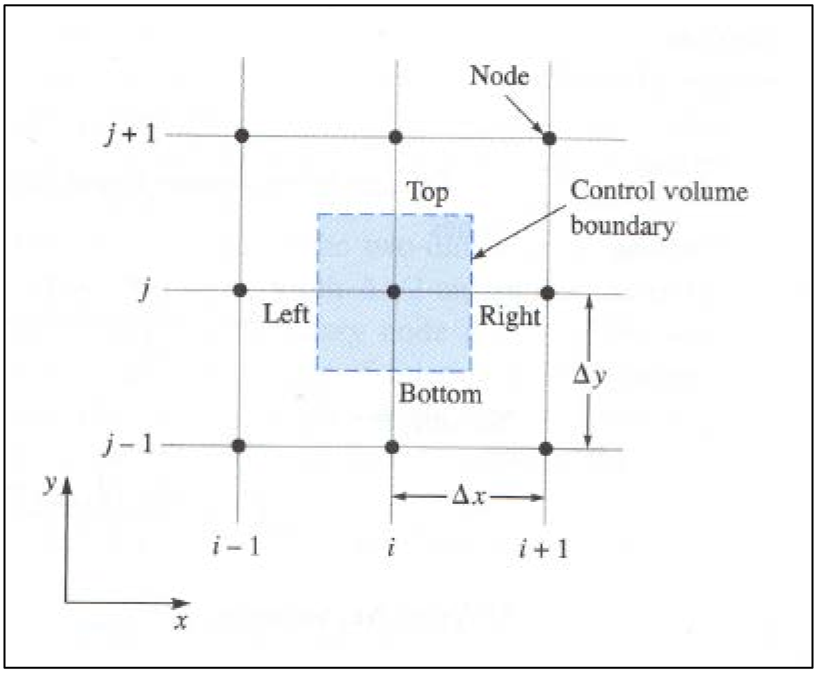


Figure 9: Interior Control Volume and Nodes (Kreith & Bohn, 2001)

Using the sample nodes and control volume in Figure 9, an energy balance can be applied to control volume and a difference equation can be formulated to express the governing differential equation. The first term of the governing differential equation is $\frac{\partial}{\partial x}\left(k_x \frac{\partial T}{\partial x}\right)$ and can be approximated by using the conservation of energy as heat into the control volume (left) minus heat out of the control volume (right) in the x direction. The difference equation for the left control volume face is shown in equations 3.1-3.3.

$$-k\frac{\partial T}{\partial x}\bigg|_{left} \text{; } \textbf{(Equation 3.1)}$$

$$k_{left} \equiv \frac{2k_{i,j}k_{i-1,j}}{k_{i,j}+k_{i-1,j}} \text{; } \textbf{(Equation 3.2)}$$

$$\frac{\partial T}{\partial x}\bigg|_{left} = \frac{T_{i,j,m}-T_{i-1,j,m}}{\Delta x} \text{; } \textbf{(Equation 3.3)}$$

The same process for formulating the difference equations can be used for the right side of the control volume as well as the term $\frac{\partial}{\partial y}\left(k_y \frac{\partial T}{\partial y}\right)$ of the governing differential equation.

The heat generation term, Q, will not be used in this thesis as the concrete will not have the capability to generation heat.

The difference equations for the x- and y-direction described above are for values in the current time step. In order to use the governing differential equation to study of transient, unsteady heat transfer, the energy storage term, $\rho c\frac{\partial T}{\partial t}$, needs an approximate difference equation. This uses the current time step (subscript m) and also the future time step (subscript m+1). Equation 3.4 is the difference equation for the storage term.

$$\rho c\, \Delta x\, \Delta y \frac{T_{i,j,m+1}-T_{i,j,m}}{\Delta t} \text{; } \textbf{(Equation 3.4)}$$

With the substitution of the difference equations into the governing differential equation, the node i,j temperature can be determined for the next time step from the temperatures and material properties for the node and its surrounding nodes. This can be done for the entire cross section for every time step throughout the entire time-temperature curves. However, due to the approximations used in the difference equations, the model is unstable if large time steps are used. Equation 3.5 is the maximum allowed time step, and smaller time steps yield the most accurate solutions.

$$\Delta t < \frac{\rho c}{2k}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)^{-1} \text{ ; (Equation 3.5)}$$

These same principles can be applied for the boundary conditions which occur at the edge of the cross section. The difference in the equations is that the control volumes are one-half and one-quarter (edge and corner control volume respectively) the area of the interior control volume. Also instead of using the conduction equation for heat leaving and entering the control volume, the edges that border the exterior have the combined heat flux $q^{"}$ applied to them. Example boundary control volumes can be seen in Figures 10 and 11.

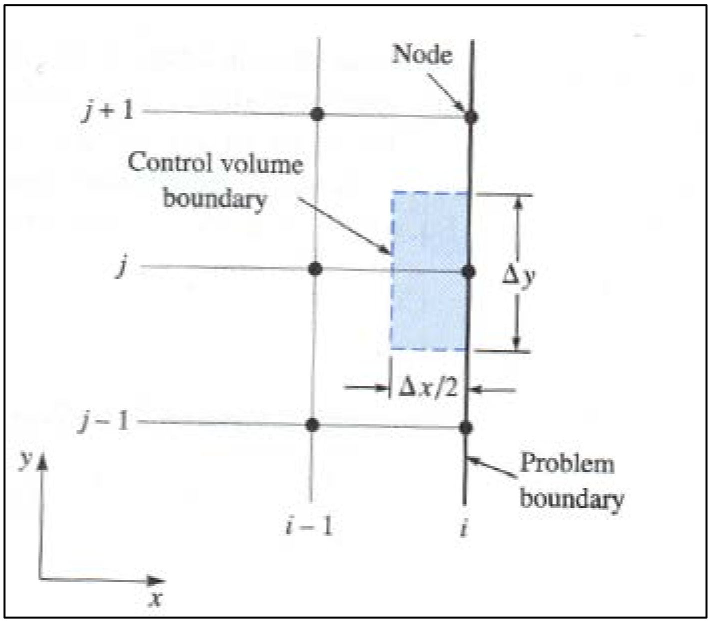


Figure 10: Edge Control Volume and Nodes (Kreith & Bohn, 2001)

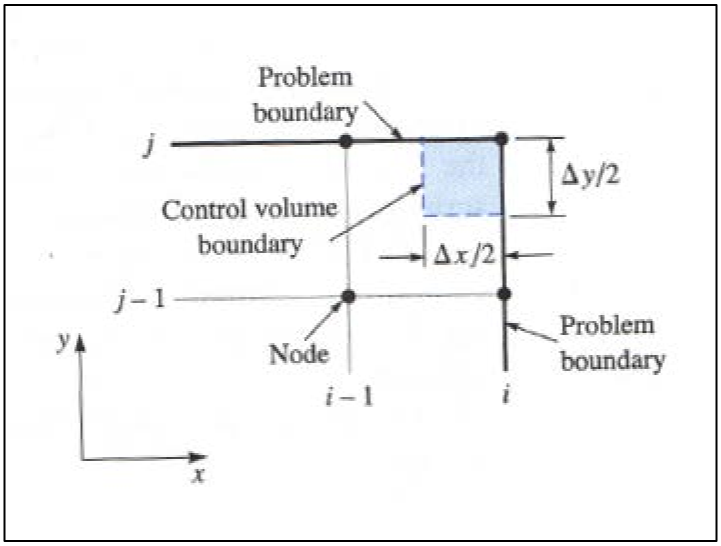


Figure 11: Corner Control Volume and Nodes (Kreith & Bohn, 2001)

These boundary control volumes combined with the interior control volumes comprise the entire cross-section of the column. The heat from the fire flows from the air to the boundary control volumes through the combined heat flux equation and then propagates through the cross section via conduction. These basic two-dimensional transient heat transfer equations are used to predict the nodal temperatures throughout a desired cross section for a given time-temperature curve.

3.3 Fire Exposures

Depending on the architecture and structural layout of a building, columns can be exposed to a fire in a number of different configurations. A wall could be built flush to column exposing only one face of the column to the room; however, a column could be located in the middle of a room thereby exposing all four sides of the column to a fire. Five configurations were chosen to be studied as they best represented all the various fire scenarios that could be presented to a column.

The first configuration is a four-sided fire exposure (Figure 12). This exposure was chosen due to the fact that the resulting temperature gradients throughout the cross section as a result of the fire are symmetrical about both the x and y axes. This results in no movement of the plastic centroid.

The second and third configurations are a one- and three-sided fire exposure with the fire side opposite any eccentricity on the column (Figure 13 and 15). This was chosen because both one- and three-sided fire exposure will result in movement of the plastic centroid along the x-axis. The fire exposure being opposite the eccentricity results in the eccentricity being reduced through time as a result of the deteriorating cross section and moving centroid.

The remaining configurations are a one- and three- sided fire exposure with both the fire and the eccentricity occurring on the same side (Figure 14 and 16). This is significant because the eccentricity will get larger with the deterioration of the cross section. The larger the eccentricity on the column, the larger the column moments will be.

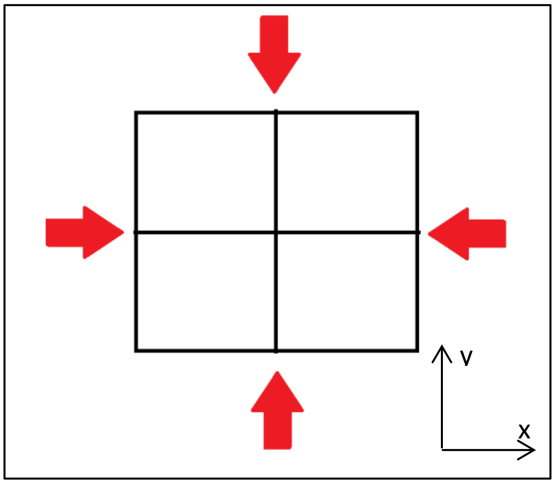


Figure 12: Four-Sided Fire Exposure

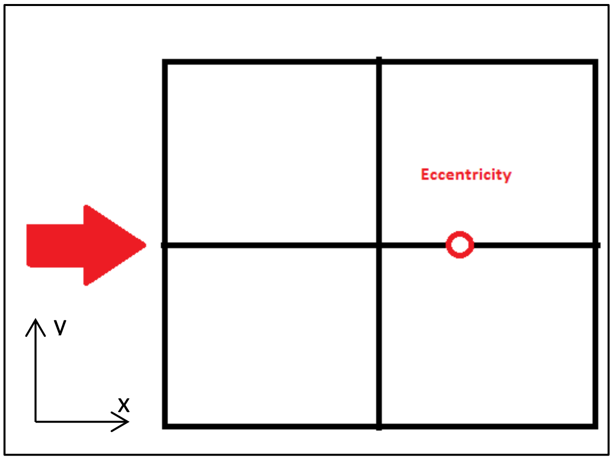


Figure 13: One-Sided Fire Exposure with Opposite Side Eccentricity

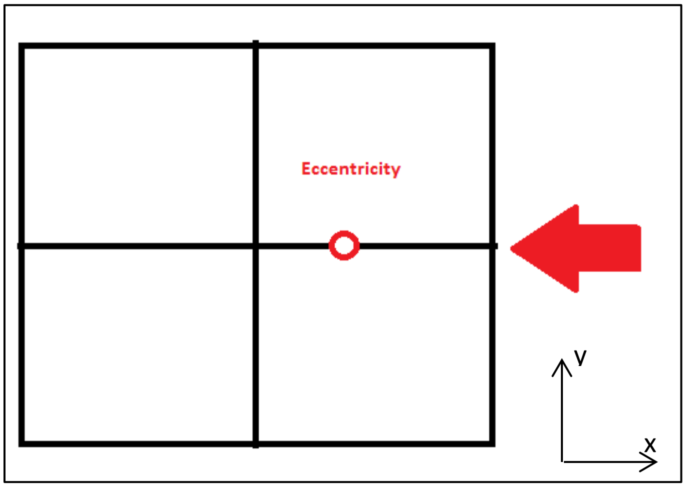


Figure 14: One-Sided Fire Exposure with Same Side Eccentricity

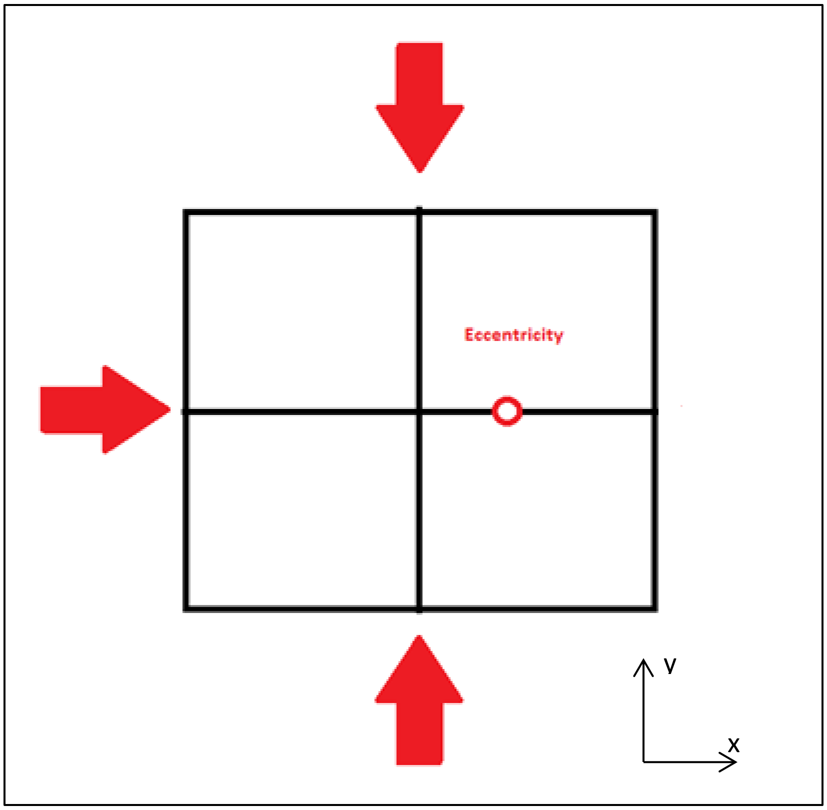


Figure 15: Three-Sided Fire Exposure with Opposite Side Eccentricity

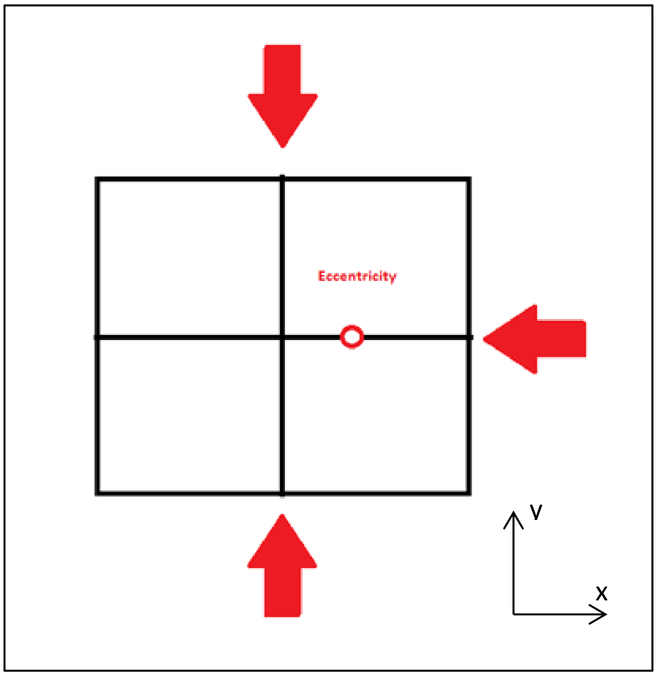


Figure 16: Three-Sided Fire Exposure with Same Side Eccentricity

3.4 Column Length Effects

The method used in this thesis to approximate the slenderness effects and determine the maximum moment on the column is the ACI moment magnifier method (Wang, Salmon, & Pincheira, 2007) (American Concrete Institute, 2004). The first step to the moment magnifier method is to determine if these slenderness effects can be ignored or not.

This is accomplished by determining if the slenderness ratio of the column is less than or equal to a limiting value based on the column end moments. Equation 3.6 is the ACI 318-05 criteria for determining whether to ignore slenderness effects or not for a column in a non-sway frame

$$\frac{kl_u}{r} \leq 34 - 12(\frac{M_1}{M_2}) \textbf{ ; (Equation 3.6)}$$

where k is the effective length factor, l_u is the unbraced column length, r is the radius of gyration, $M_1$is the smaller of the end moments, and M_2 is the larger of the end moments (American Concrete Institute, 2004). For this thesis, only identical end moments will be studied. Therefore M_1/M_2 is one.

If slenderness effects need to be addressed, the next step is to determine the magnification factor (δ_{ns}). The magnification factor is multiplied by the applied moment on the column (M_m) to determine the maximum moment (M_{max}) in the column.

$$M_{max} = \delta_{ns} M_m \textbf{ ; (Equation 3.7)}$$

$$M_m = P * e \textbf{ ; (Equation 3.8)}$$

where P is the applied load and e is P’s eccentricity to the plastic centroid.

$$\delta_{ns} = \frac{C_m}{1 - \frac{P_u}{\phi_K P_c}} \textbf{ ; (Equation 3.9)}$$

$$C_m = 0.6 + 0.4 \frac{M_1}{M_2} \geq 0.4 \textbf{ ; (Equation 3.10)}$$

The moment magnification factor is a function of C_m, factored load P_u, Euler load of the column P_c, and a stiffness reduction factor φ_K. C_m describes the type of loading on the column in terms of the moment gradient. For this thesis C_m will be one (1) as the end moments are equal. P_u is the factored load on the column. In this study, all design and factors of safety were removed from the analysis to determine the actual fire performance of the columns. Therefore, P_u is equal to P_n. Due to the removal of factors of safety, φ_K will be removed from the magnification factor equation. Thus the denominator reduced to P_u/P_c, which is the ratio of the nominal load to the Euler buckling load (P_c).

$$P_c = \frac{\pi^2 EI}{(kL_u)^2} \text{ ; \textbf{(Equation 3.11)}}$$

The flexural rigidity EI is the product of the modulus of elasticity and moment of inertia of the column. However, due to cracking of the concrete, general irregularities with the stress-strain curve of concrete and creep, two simpler equations have been given which allow for approximations of EI (Wang, Salmon, & Pincheira, 2007) (American Concrete Institute, 2004).

$$EI = \frac{0.2E_cI_g + E_sI_{se}}{1+\beta_d} \text{ ; \textbf{(Equation 3.12)}}$$

or

$$EI = \frac{0.4E_cI_g}{1+\beta_d} \text{ ; \textbf{(Equation 3.13)}}$$

where E_c is the concrete modulus of elasticity of concrete, I_g is the concrete gross moment of inertia ignoring reinforcement, E_s is the modulus of elasticity of steel, I_{se} is the reinforcement moment of inertia, and β_d is the ratio of the axial dead load to the total axial load. For this thesis, the second EI equation was used due to the conservative nature of the equation and the simplicity of implementation into the spreadsheet model.

For this thesis, β_d was assumed to be one (1). This assumption was made based on the fact that β_d is the ratio of the sustained axial load to the total axial load. In column furnace testing, the load on the column is only the sustained load.

3.5 Temperature Dependent Properties

The only temperature dependent property used in the calculation of slenderness effects is the modulus of elasticity of concrete. Figure 17 shows a graph from *Structural Fire Protection* (1992) demonstrating the reduction of the modulus of elasticity due to increased temperatures.

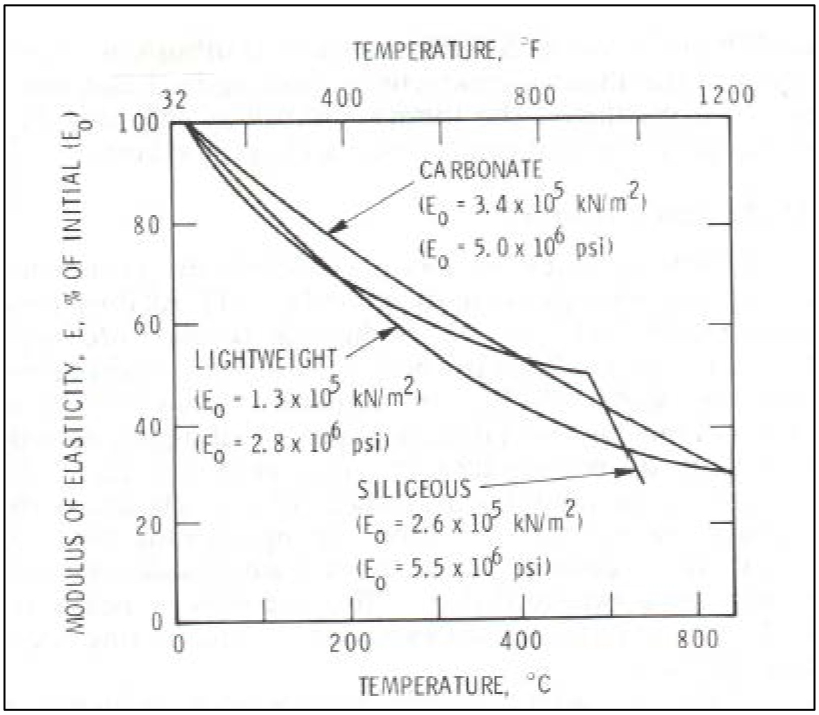


Figure 17: Concrete Modulus of Elasticity (American Society of Civil Engineers, 1992)

Since each portion of the cross-section has different temperatures, the average overall cross-section temperature was assumed in determining the modulus of elasticity value to use. This assumption was made based on the fact that the EI term is a flexural rigidity term. Even though the entire cross-section does not contribute to the overall strength of the column, the entire cross-section should contribute to the rigidity of the cross-section. Trendlines were taken from each of the graphs in Figure 17 to obtain approximate equation for each aggregate type to use in the spreadsheet model.

Each of these variables and equations are calculated at each time step to determine the maximum moment in the column. The maximum moment is compared with the moment resisting strength in the column. If the moment resisting strength is less than the maximum moment then the column fails. The next step in the thesis was to research the best method to implement the described equations through transient calculations while allowing the user the freedom to manipulate variables, column properties, and fire curve data.

3.6 Spreadsheet Implementation

A spreadsheet-based model was selected as the best tool to implement the heat transfer and structural analysis equations. The choice of a spreadsheet was based on several factors. The first factor was the usability and wide knowledge of spreadsheets. Most computers on the market today come preloaded with a spreadsheet program. Coupling this with the fact that engineers today either already have the knowledge or should have the knowledge of spreadsheets and using a spreadsheet versus a more expensive program, the best implementation method was a spreadsheet (Baughn & Rossi, 1992).

The second choice was the ease of visualizing the results with charts and graphs. Spreadsheet programs allow a variety of different visualization methods while allowing for multiple variables to be displayed on the same graph. Several spreadsheet programs also allow for contour plots which, for the case of two-dimensional heat transfer, help visualize the temperature isotherms throughout the cross-section (Schumack, 2004).

The third choice is that with a spreadsheet the calculations and processes are transparent, allowing education of the user on the process behind designing for fire conditions. With calculation methods such as MATLAB or complicated finite element computer programs, the calculation process is done behind the scenes and only the results are visible.

Microsoft Excel was chosen as the spreadsheet based program as it was readily available to the author and also to most PC users. The following sections describe the process behind coding the equations and assumptions into a re-usable model.

3.6.1 Excel Spreadsheet Overview

Two-dimensional heat transfer is relatively easy to implement in Excel for single time-steps or steady-state conditions through the use of the goal-seek function or circular referencing until the solution is achieved. However, for unsteady-transient two-dimensional heat transfer, the use of Excel Macro's is necessary to perform the repetitive calculations through the entire time frame (Besser R. S., 2002).The process listed below is the order in which the macros were created. The flowchart in Figure 18 is a map of the process to calculate fire performance and the connection between user input values and both the heat transfer and structural capacity calculations. The user executes the macros and follows the programed steps (Kharab, 1995).

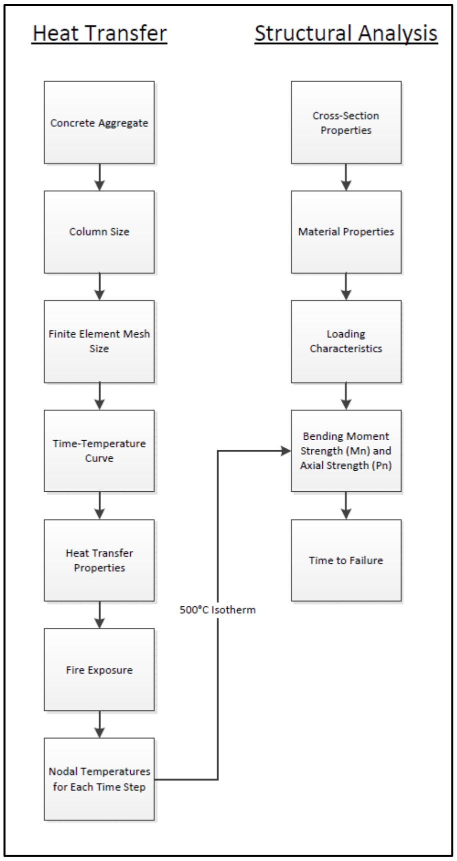
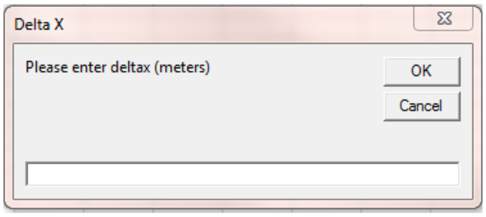


Figure 18: Spreadsheet Model Flowchart

3.6.2 User Input

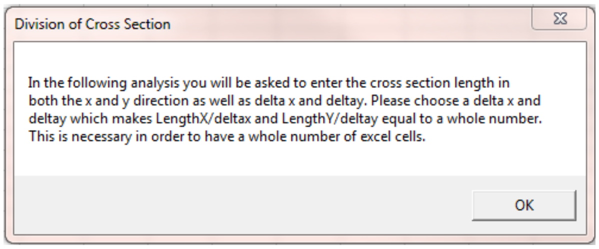
For the model, the user is prompted to make selections as to certain values such as type of aggregate, control volume size, and column characteristics. The user enters these values into “input boxes” when the macros prompts them to. A sample input box is shown in Figure 19.


Delta X

Please enter deltax (meters)

Figure 19: Input Box

Through the program, if explanation of the next step is needed, a message box describes the upcoming step to the user to allow them to either cancel the process to gather more data, or to proceed ahead to the next step. A sample message box is shown in Figure 20.



Division of Cross Section

In the following analysis you will be asked to enter the cross section length in both the x and y direction as well as delta x and deltay. Please choose a delta x and deltay which makes LengthX/deltax and LengthY/deltay equal to a whole number. This is necessary in order to have a whole number of excel cells.

Figure 20: Message Box

Before the user starts the heat transfer calculations by clicking the "heat transfer button" (Figure 21), a time-temperature curve must be entered into the appropriate columns listed in the spreadsheet. With this data the user may proceed to start the macro.

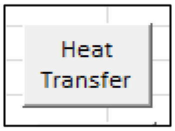


Figure 21: Heat Transfer Button

3.6.3 Time-Step Calculation

The first step of the model macro is the calculation of the maximum permissible time-step and the manipulation of the user-inputted time-temperature curve to be below the maximum time-step. The user selects the type of aggregate (carbonate, expanded shale, siliceous), and from the thermal property equations listed in the Appendix, the ambient temperature properties are determined.

The next value for the maximum time-step equation is the control volume grid size. The user inputs Δx and Δy and with the ambient temperature properties, the maximum time-step is calculated. If the user-inputted time step is more than the allowable, the macro will interpolate between the inputted values and reenter a time-temperature curve with a time-step less than the maximum.

3.6.4 Cross-Section Size and Initial Heat Transfer Time-Step

Next, the column cross-section dimensions width (b) and depth (h) are input. With b, h, Δx, and Δy the number of control volumes can be calculated. The number of control volumes in the x-direction is h divided by Δx, and the number of control volumes in the y-direction is b divided by Δy. Due to the fact that the underlying heat transfer equations describe nodal temperatures, the number of nodes needs to be calculated. This is done simple by adding one to each number of control volumes in each direction. The macro then designates a separate cell for each node in the cross-section (Eid, 1987). For example, if a column is 1 meter X 1 meter and the control volume size is 0.1 meter X 0.1 meter, then the number of control volumes in the cross-section is 10x10 or 100. However, the number of nodes needed to perform the calculations is 11x11 or 121. The number of cells needed is 121.

Once the number and shape of the cells and cross-section are determined, the macro completes the initial time-step by filling in the initial temperature values for the cross-section. From the cross-section temperature the macros calculates the thermal properties for the current time-step to be used in the next time-step calculation.

3.6.5 Subsequent Heat Transfer Time-Steps

Next, the macros conduct the heat transfer analyses for each subsequent time-step and boundary condition temperature by using a loop. The first step of the loop is to determine the number of times the loop will run. This is done by determining the number of time steps that span the duration of the time-temperature curve. From there, the temperature for each boundary node and interior node is calculated at that time-step. Once the temperature for each node at the current time step has been calculated, the thermal properties are again found for that time step to be used in the next time step. The loop will run the calculations until the end of the time-temperature curve. This process for both four-, one (right)-, and one (left) - sided fire exposure. The user at the start of the loop selects which exposure to use.

3.6.6 Structural Equation Implementation for Each Time-Step

Once the heat transfer calculations have been completed for the entire time-temperature curve, the next step is to run the structural analysis calculations. As with the heat transfer macros, the user selects the structural analysis button to start the structural analysis macro.

The first step to the structural analysis macro requests input of the column characteristics such as concrete cover, area of steel, steel yield strength and modulus of elasticity, and loading conditions. Once the variables have been defined, the centroid, area, moment of inertia, radius of gyration, and the width and depth of the column are calculated for each time step from the shape of the 500°C isotherm. The temperatures of the steel are then determined so the calculation of their yield strength and modulus of elasticity can be completed.

Next the macro calculates the balanced eccentricity and investigates the load capacity of the column. Using the balanced eccentricity, the macro determines whether the column is compression- or tension-controlled based on the applied eccentricity of the load. The macro then performs the nominal strength calculations of the column and compares the axial strength to the applied axial load to determine if crushing failure occurs. Then the macro calculates the slenderness effects and the maximum moment in the column and compares that to the moment resisting strength of the column to determine if buckling failure occurs. These calculations are performed for each time-step.

3.6.7 Post-processing

The final step in the model is the post-processing macros. The user selects the post-processing button and the macros runs. The post-processing macro collects the data from the heat transfer and structural analysis calculations and organizes them on another sheet of the spreadsheet workbook. The user is allowed to select up to nine (9) nodes for which the temperature over the whole the time frame will be collected. Besides the nodal temperature the other values compiled are the left and right steel temperatures, the area of the 500°C isotherm, the moment and axial resisting column strength, the radius of gyration, the maximum moment, the balance eccentricity and the location of the plastic centroid.

Each of these values is displayed over time which allows the user to graph any variable. The user is also able to graph the contour plot of a specific time step by selecting the entire cross-section at the desired time-step and using the surface contour chart built into Excel.

Once the model was completed, it was validated for accuracy. This was done using *ANSYS* to validate the heat transfer calculations and previous furnace tests and models from published research on reinforced-concrete columns.

3.7 Model Benchmarking

In order to determine if the spreadsheet model was working correctly and produced acceptable results, the two sections of the model, heat transfer and structural analysis, were benchmarked against two other solution sources. The heat transfer model was compared to the use of an equivalent finite element mesh within *ANSYS,* and published data on the time to failure of several columns from actual furnace tests and other computer models provided benchmarks for the structural analysis.

3.7.1 Heat Transfer Benchmarking

The heat transfer calculations were benchmarked against the use of *ANSYS*. Both one- and four-sided fire exposures were tested. In order to compare the nodal temperature between the thesis model and *ANSYS*, a test column section was selected as well as a finite element grid size. A square column comprised of 400 control volumes (441 nodes) was chosen as the example column. Table 1 shows the key properties of the column, and Figure 22 shows the nodes selected for comparison between the *ANSYS* and spreadsheet models. The four-sided fire exposure used nodes 1 through 11 due to the symmetry of the fire loading. One-sided heating used all 21 nodes.

Table 1: Four-Sided Fire Exposure Heat Transfer Benchmarking Column Properties (meters)

Fire Curve	ASTM E119
Aggregate	Siliceous
Δx	0.015
Δy	0.015
h	0.3
b	0.3

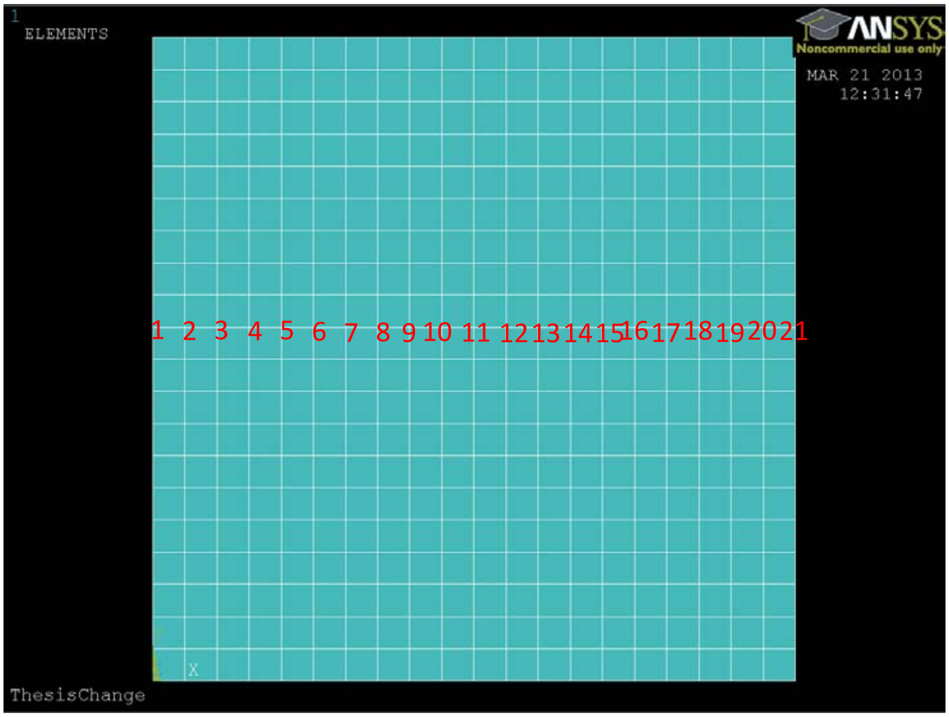


Figure 22: Nodes for Heat Transfer Comparison

The ISO 834 fire curve was used as the time-temperature curve. The curve lasted for 320 minutes, and the time step was 18.75 seconds. The emissivity of the hot gases was taken as 0.5 and the convective heat transfer coefficient was 25 W/m^2°C. For the one-sided exposure, the left face was exposed to the fire. The aggregate used was siliceous. These heat transfer and column properties were the same for both one-sided and four-sided fire exposures.

For the *ANSYS* model, several assumptions had to be made due to the limited capabilities of the program. *ANSYS* does not allow for material property equations to be input except a table of temperatures and associated material property values. This could be used except that the value is constant until another temperature and material property value is reached. The material property assumption used was that the thermal conductivity, density, and specific heat were all constant at the ambient temperature values through the analysis.

The only other assumption made using *ANSYS* was the strategy to apply thermal energy to the column. In the thesis model heat is applied to the column in a combined convective and radiation term. In *ANSYS* radiation can only be applied based on a heat source and not the surrounding gases. Therefore the assumption was that the heat applied to the column in *ANSYS* was only convective heat transfer.

3.7.2 Benchmarking Studies for Column Capacity

Eleven different columns with varying aggregate, reinforcement, and cross-section dimensions were selected from several research papers to be compared against the thesis model. Table 2 shows the selected column characteristics along with the reference sources. Each column was subjected to the ASTM E119 or similar ISO 834 curve with four-sided fire exposure. Column 2 of Table 2 is the numbering given for ease of processing results for this thesis. Table 3 shows the material properties and loading conditions of the columns. Each column model was run for 8 hours. The durations of the time steps used can be seen in Table 3.

Table 2: Benchmark Column Dimensions

Source and Article Test #	Thesis Column #	Length (mm)	Width (mm)	# of Bars	Diameter (mm)	Cover (mm)
(Dotreppe, et al., 1996); 35A	1	300	300	4	25	40
(Dotreppe, et al., 1996); 33AC	2	300	300	4	25	25
(Dotreppe, et al., 1996); 42A	3	400	400	4	25	25
(Dotreppe, et al., 1996); 22BC	4	200	300	6	12	35
(Woollerton & Lie, 1988); I2	5	305	305	4	25.5	48
(Woollerton & Lie, 1988); I12	7	305	305	4	25.5	48
(Allen & Lie, 1974); i2	8	150	150	4	20	12
(Allen & Lie, 1974); ii3	9	200	200	4	20	29
(Allen & Lie, 1974); iii11	10	300	300	6	21	30
(Lie & Irwin, 1993); nrcc 1	11	305	305	4	25	48

Table 3: Benchmark Column Material Properties

Thesis Column #	Concrete Strength (Mpa)	Steel Strength (Mpa)	Eccentricity (mm)	Force (kips)	Length (m)	Effective Length Coefficient, k	Time Step (seconds)
1	35.9	591	20	425	3.95	1	15
2	32.7	591	0	878	2.10	1	60
3	37.4	591	20	1680	3.95	1	15
4	37.6	493	0	620	2.10	1	30
5	36.9	444	0	1333	3.81	0.5	60
7	39.9	444	0	1778	3.81	0.5	30
8	38	450*	25	273	3.00	1	15
9	32	450*	25	464	3.00	1	15
10	46	450*	25	1708	3.00	1	15
11	36.1	414	0	1067	3.81	0.5	15

In several of the papers, key column characteristics were not specified so in each case a value was used that was similar to the other columns in the benchmarking study. For instance, columns 8 to10 did not have a steel strength (MPa) detailed in the article so a value of 450 MPa was assumed. Columns 1 to 4 and 7 to 9 did not have the aggregate type specified; siliceous aggregate was chosen as aggregate because it has a higher thermal conductivity than carbonate and will therefore lead to smaller resistance times.

Columns 5 and 10 were both siliceous aggregate and column 6 was the only column aggregate to be carbonate.

The authors of each paper developed their own model and compared their model to the experiments run. This thesis compares the spreadsheet model to the authors’ models and the column experiments to benchmark the spreadsheet model. After the benchmarking exercises indicate that the spreadsheet worked correctly and provided acceptable results, the example columns for the column behavior study could be run.

3.8 Column Behavior Study

Various aspects of column behavior were studied using the numerical model constructed for this thesis. The changes to load eccentricity, slenderness ratio, and end connections of a column were all studied to determine how each affected the overall performance of a reinforced-concrete column during fire conditions. The difference between neglecting or accounting for moment magnification and increasing the cover thickness of a column were also analyzed. Variation in the column properties, such as centroid location and eccentricity changes, were investigated. Finally latent heating was studied.

An example column was taken from *Reinforced-concrete Design* (Wang, Salmon, & Pincheira, 2007) to keep the overall parameters of the column the same. This allowed for uniformity of the results which enable comparison between the various tests. Table 4 shows the column characteristics kept constant throughout each test.

Table 4: Column Properties

Length (mm)	431.8
Width (mm)	431.8
Area: Right Steel (m^2)	0.00322363
Area: Left Steel (m^2)	0.00322363
Concrete Compressive Strength (MPa)	20.68
Steel Yield Strength (MPa)	275.8
Steel Modulus of Elasticity (GPa)	200
Force (kN)	2335

For the tests involving concrete cover, the cover thickness was increased from the ACI 318-05: 7.7.1 (c) minimum for the column (38.1 mm) to 63.5 mm. For eccentricity, the distance was increased from the ACI 318-05: 10.12.3.2 minimum (28.194 mm) up to 101.6 mm. For column length, it was increased from 3.048 meters to 4.2672 meters. Finally, for the end connection factor (k), both pinned (k=1.0) and fixed (k=0.5) connections were used. Numerous combinations of these variables were changed for each test run in order to obtain the data used in the results chapter. Both three and four-sided fire exposures were subjected to the column combinations. For each test the cross-section was discretized into 400 (20x20) control volumes, and the ISO 834 fire curve was applied to the column.

4. Model Benchmarking Results

The following chapter contains the results from both the heat transfer and structural benchmarking studies. The analysis of the results of this section was necessary in order to determine the acceptability of the results and the subsequent column behavior study.

4.1 Heat Transfer Benchmark Studies

The results from both the four- and one-sided fire exposures are presented in the following sections. In order to compare the temperatures of the thesis model nodes with the *ANSYS* nodes, the percent difference between the nodal temperatures was calculated, and then the percents were averaged together to the get the overall average percent difference.

4.1.1 Four-Sided Exposure

For four-sided exposure, the thesis model predicted higher temperatures for the outer nodes (1 to 3) than the *ANSYS* model. For nodes 4 to 7, the thesis model and *ANSYS* results had fairly good agreement in the temperature results. From node 8 to the center node (11), the thesis model under-predicted the temperatures. Table 5 shows the average percent difference of each of the nodes in Figure 1. The positive values indicate over-prediction by the spreadsheet model whereas the negative values indicate under prediction. Figures 23 to 25 show the time-temperature histories of specific nodes (1, 5, 11). The time-temperatures clearly show the differences between the model and *ANSYS*. Node 1 which has the largest percent difference has the largest temperature gap between the two time-temperature curves. Node 5 has minimal separation between the curves, whereas node 11 has separation but the *ANSYS* curves is above the model curve, resulting in the negative difference.

Table 5: Four-Sided Thesis and *ANSYS* Comparison

Node	Average % Difference
1	40.32
3	28.11
3	17.66
4	8.98
5	1.92
6	-3.66
7	-7.94
8	-11.05
9	-13.13
10	-14.31
11	-14.69

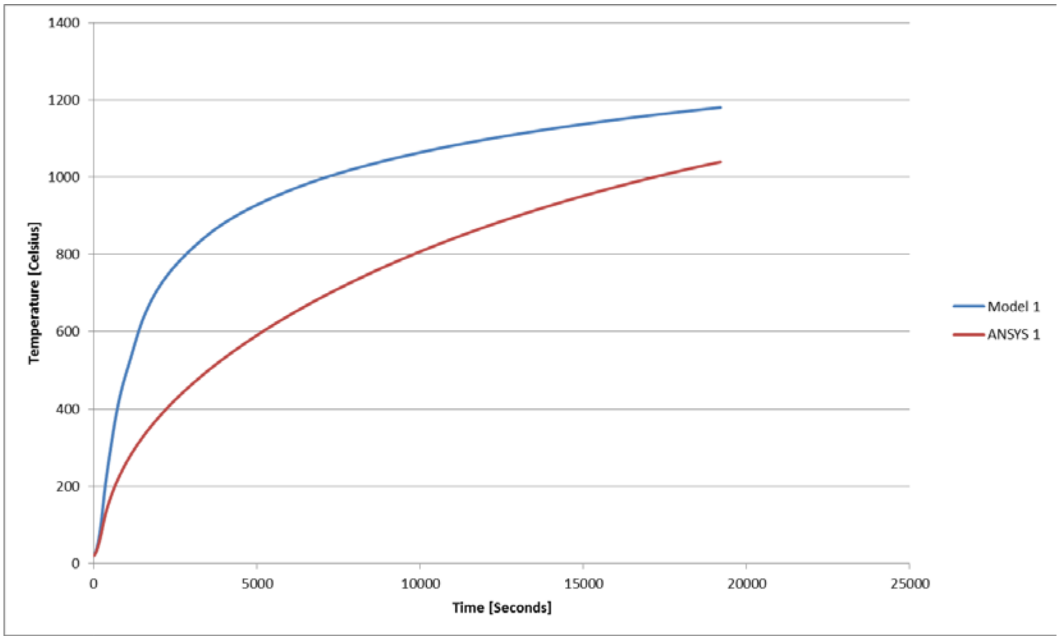


Figure 23: Node 1 Time-Temperature Comparison of Model and *ANSYS*-Four Sided Fire Exposure

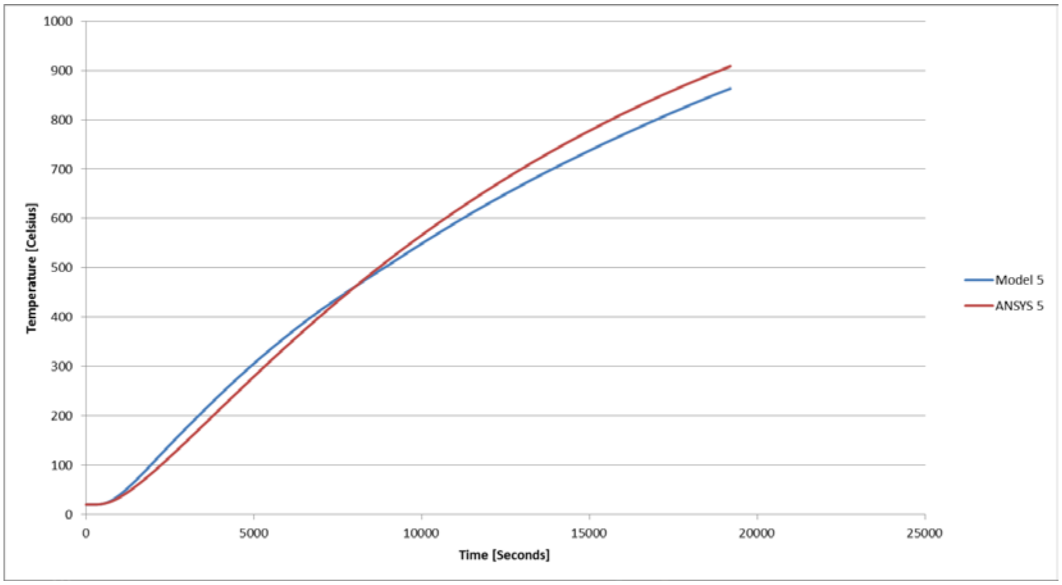


Figure 24: Node 5 Time-Temperature Comparison of Model and *ANSYS*-Four Sided Fire Exposure

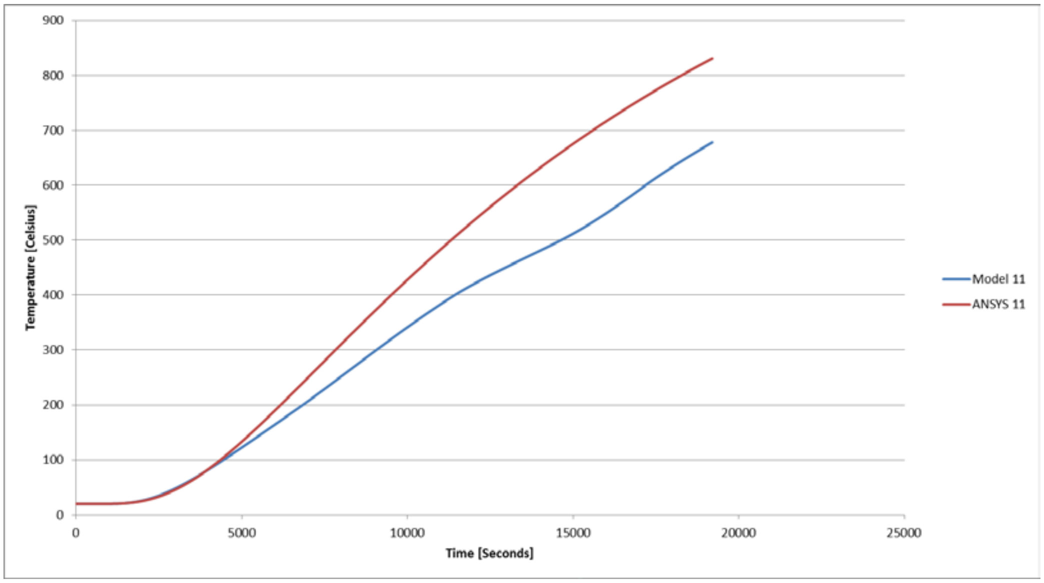


Figure 25: Node 11 Time-Temperature Comparison of Model and *ANSYS*-Four Sided Fire Exposure

4.1.2 One-Sided Exposure

For one-sided fire exposure, the thesis model generally over-predicted the temperature. Table 6 shows the average percent error for each of the nodes through the cross-section as shown in Figure 22. The interior nodes (8 to 15) had good agreement with the *ANSYS* model: the percent difference was less than 10. The exterior nodes tended to have higher percent differences with those nearest the fire exposure having greater difference than those on the far side. Figures 26 to 30 show the time-temperature histories of the several nodes throughout the cross-section. The difference between the thesis model and *ANSYS* can be clearly seen in them. In Figure 26, the difference between the temperature curves is very large; however in Figures 27 and 28, the gap between the curves diminishes to the point. In fact, for in Figures 28 and 29, the lines are almost identical. Figure 30 shows a potential error in the calculation of the *ANSYS* model. For node 18, the *ANSYS* model predicted that the temperature would decrease from ambient conditions until about 2.5 hours then increase above ambient. The thesis model predicted that the temperature would remain constant at ambient conditions.

Table 6: One-Sided Thesis and *ANSYS* Comparison

Node	Average % Error
1	55.54
2	44.34
3	34.50
4	26.39
5	20.01
6	15.10
7	11.38
8	8.54
9	6.48
10	5.16
11	4.53
12	4.60
13	5.38
14	6.93
15	9.32
16	12.68
17	17.13
18	22.79
19	29.82
20	38.44
21	49.21

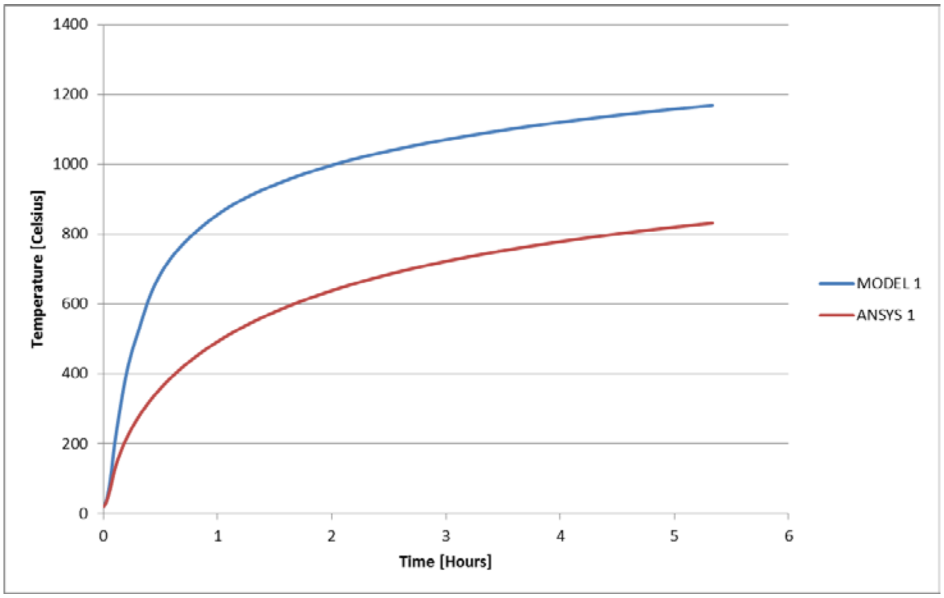


Figure 26: Node 1 Time-Temperature Comparison of Model and *ANSYS*-One-Sided Fire Exposure

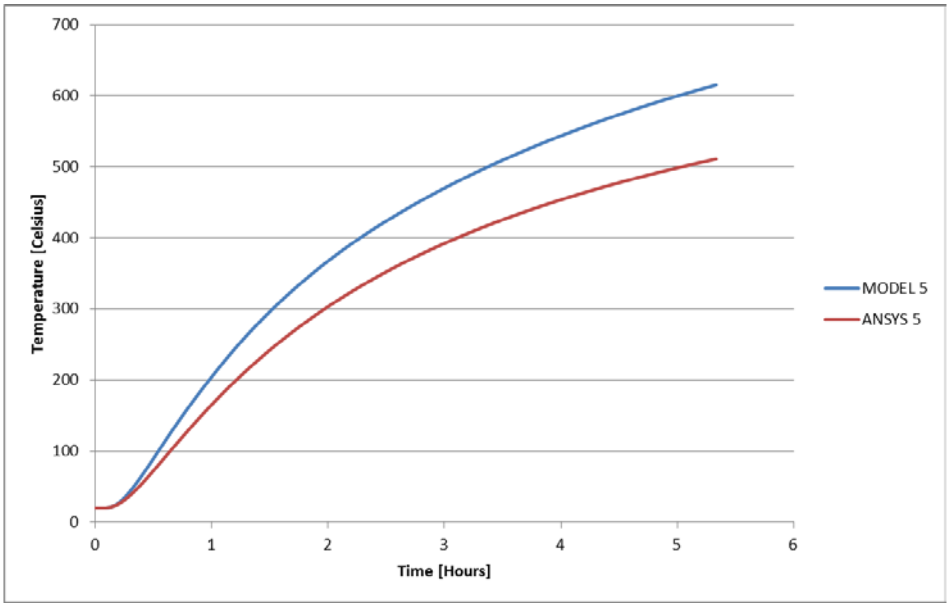


Figure 27: Node 5 Time-Temperature Comparison of Model and *ANSYS*-One-Sided Fire Exposure

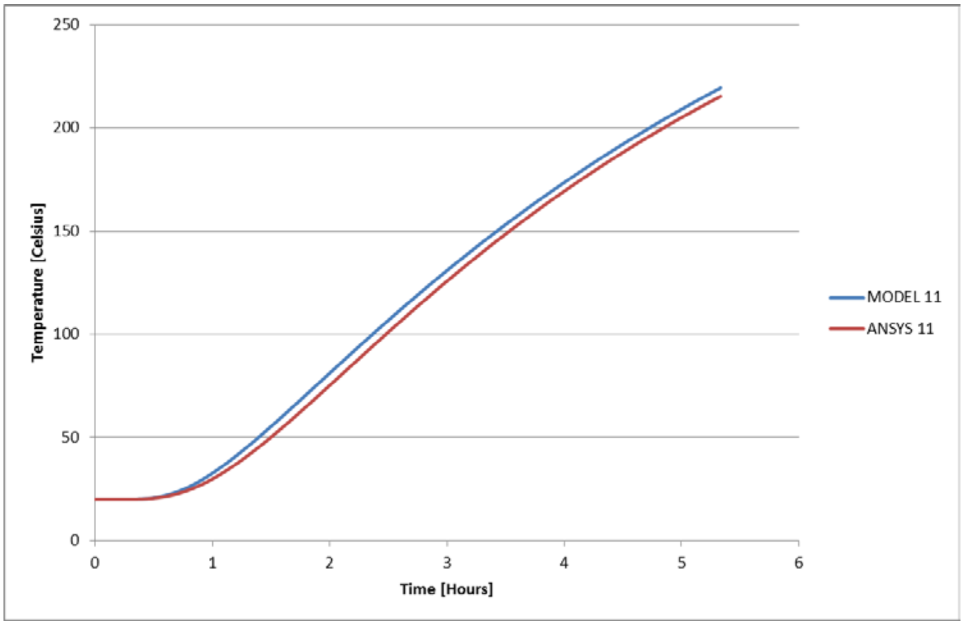


Figure 28: Node 11 Time-Temperature Comparison of Model and *ANSYS*-One Sided Fire Exposure

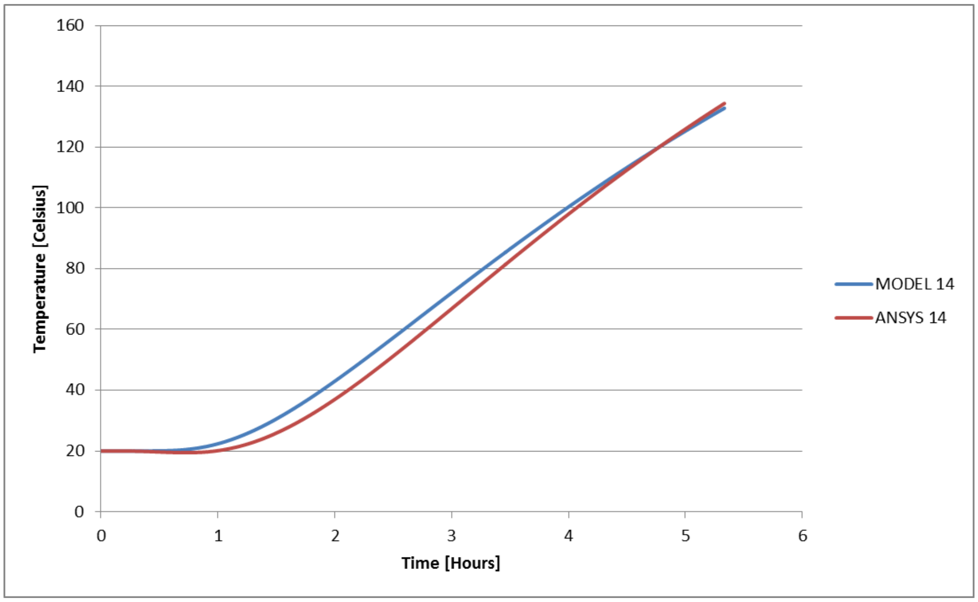


Figure 29: Node 14 Time-Temperature Comparison of Model and *ANSYS*-One-Sided Exposure

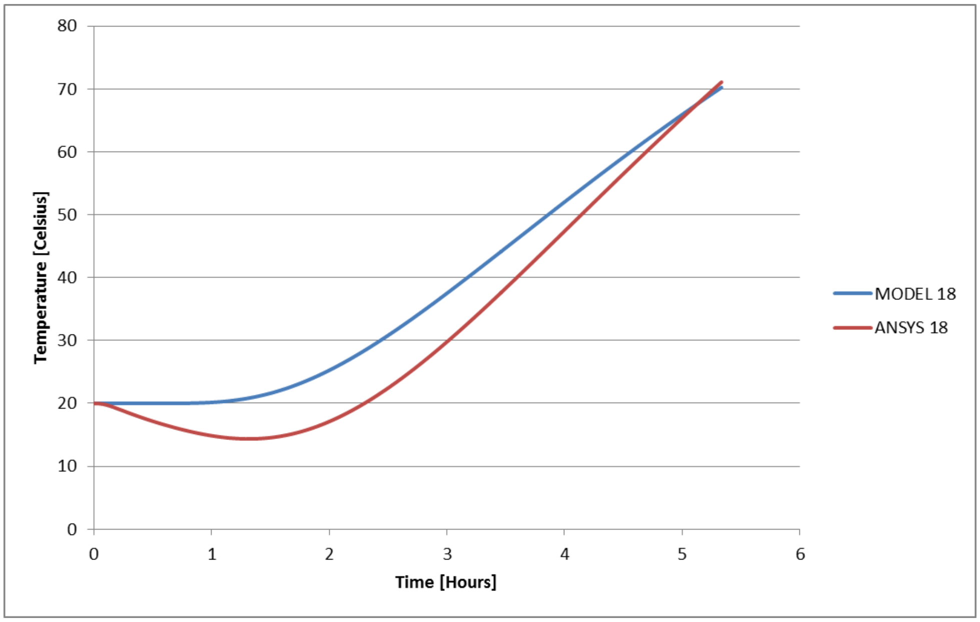


Figure 30: Node 18 Time-Temperature Comparison of Model and *ANSYS*-One-Sided Exposure

4.2 Benchmarking Studies for Column Capacity

The results of the benchmarking exercises for column capacity can be seen in Table 7. Each column example from the literature had an experimental fire resistance time (R_f actual) and a computer model fire resistance time (R_f theory). The thesis model fire resistance time was compared to both R_f actual and R_f theory. For the thesis model, each column has a crushing failure time. Columns in which the force was applied at an eccentricity will also exhibit a flexural bending failure time. The smallest value of time to failure of either the crushing failure or the flexural bending failure is the fire resistance time for the thesis model.

Table 7: Structural Benchmarking Results, Failure Time (minutes)

Column	R_f Theory	R_f Actual	Crushing	Flexural Bending
1	120	66	178	120
2	60 to 120	69	201	N/A
3	120	34	173	103
4	60 to 120	97	98	N/A
5	145	170	200	N/A
7	167	216	227	N/A
8	57	64	57.5	N/A
9	89	107	93.25	N/A
10	147	165	131.25	N/A
11	196	208	207.5	N/A

The failure times obtained from the thesis model generally agreed fairly well with the failure times presented in the literature. Failure times for columns 8, 9 and 11 were predicted within 15 minutes of both the experimental and comparison model's results. Failure times for columns 4 and 10 were within a half-hour of the benchmark data. The other columns had predicted either similar to the experimental results or the comparison model. Column 1 predicted the failure time exactly with the comparison model but was an hour off in the experimental tests. This is the same for column 3.

4.3 Summary

Based on the results from both the heat transfer and column capacity benchmarking studies, the spreadsheet calculation model is working correctly for both aspects of determining the overall performance of a reinforced-concrete column in fire conditions. The results have reasonable accuracy for application for further studies in column behavior in fire conditions.

5. Column Behavior Study Results

The following chapter describes the results for the column behavior study. The study looked into the effects of eccentricity, slenderness ratio, and end connections on a column. A comparison between accounting for slenderness effects (moment magnification) on a column and ignoring them was studied. Decreasing the cover to the *ACI* minimum was looked into. For all of these studies, the variations in column properties, such as plastic centroid location and eccentricity were analyzed over time and based on applied fire exposures.

5.1 Initial Eccentricity

Figures 31 to 34 and Tables 8 to 9 show that the fire performance of a reinforced-concrete column is reduced when the eccentricity of the applied force is increased. Both the crushing failure mode and flexural buckling failure mode are affected. Table 8 shows the failure times of carbonate aggregate columns (length = 3.048 m) with increasing eccentricity. Table 9 shows the failure times of siliceous aggregate columns (length = 3.048 m). Figure 31 shows the columns with carbonate aggregate and increasing eccentricity; whereas Figure 32 is a column comprised of siliceous aggregate. Figures 33 and 34 show the same columns but with the flexural buckling failure mode. In both the graphs the larger eccentricities yield smaller failure times. Failure occurs when either the capacity ratio P_n/P or M_n/M_{max} curves decrease below 1.0.

Table 8: Failure Times (minutes) of Carbonate Aggregate Columns with End Connection k=1.0-Eccentricity

Column	Eccentricity (mm)	Crushing	Flexural Buckling
12	0	366.25	N/A
14	28.194 (ACI min)	299.25	84.25
22	50.8	274.25	68.25
24	76.2	163	27.25

Table 9: Failure Times (minutes) of Siliceous Aggregate Columns with End Connection k=1.0-Eccentricity

Column	Eccentricity (mm)	Crushing	Flexural Buckling
35	0	263.25	N/A
36	28.194 (ACI min)	203.25	61
37	50.8	191.5	50.25
38	76.2	121.5	17.5

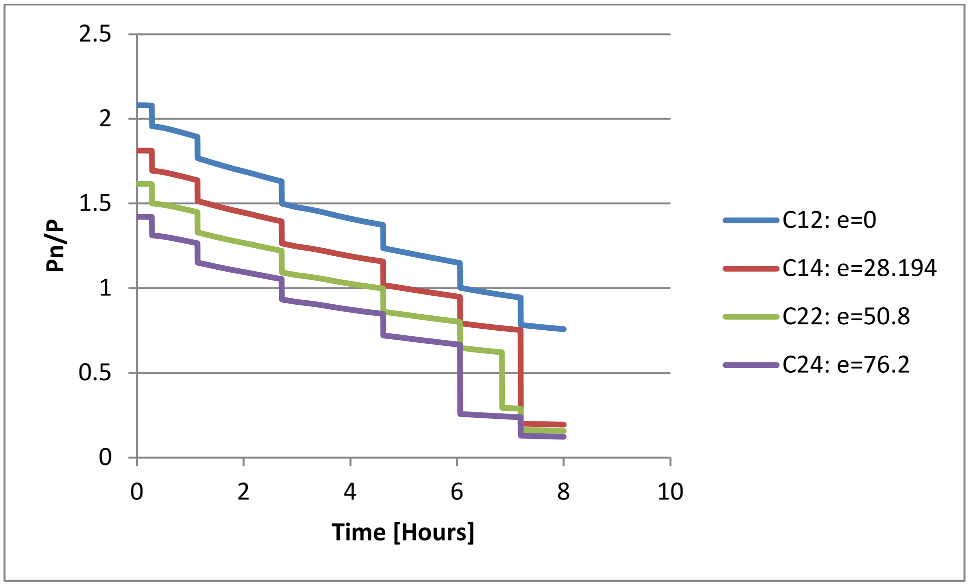


Figure 31: Carbonate Aggregate Column with Increasing Eccentricity-Crushing

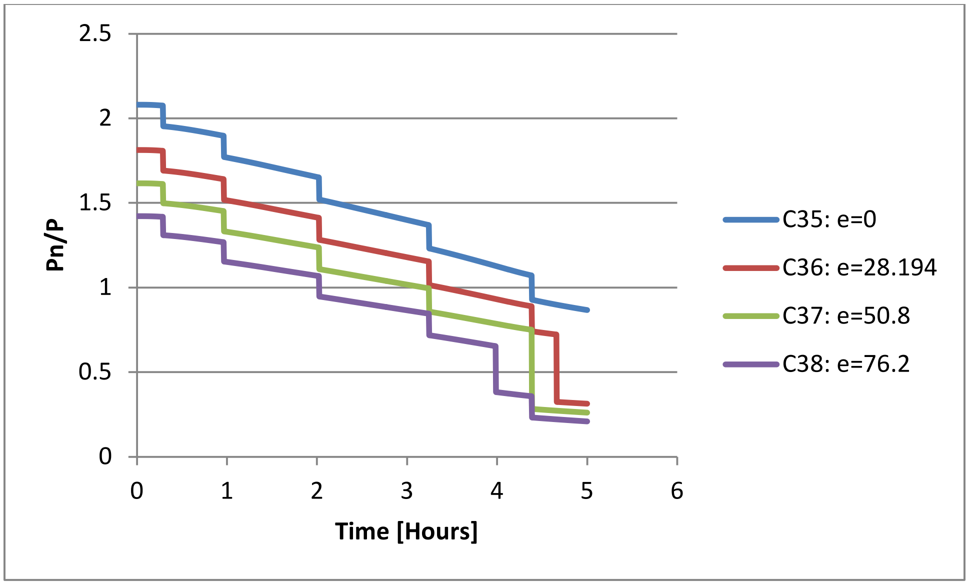


Figure 32: Siliceous Aggregate Column with Increasing Eccentricity-Crushing

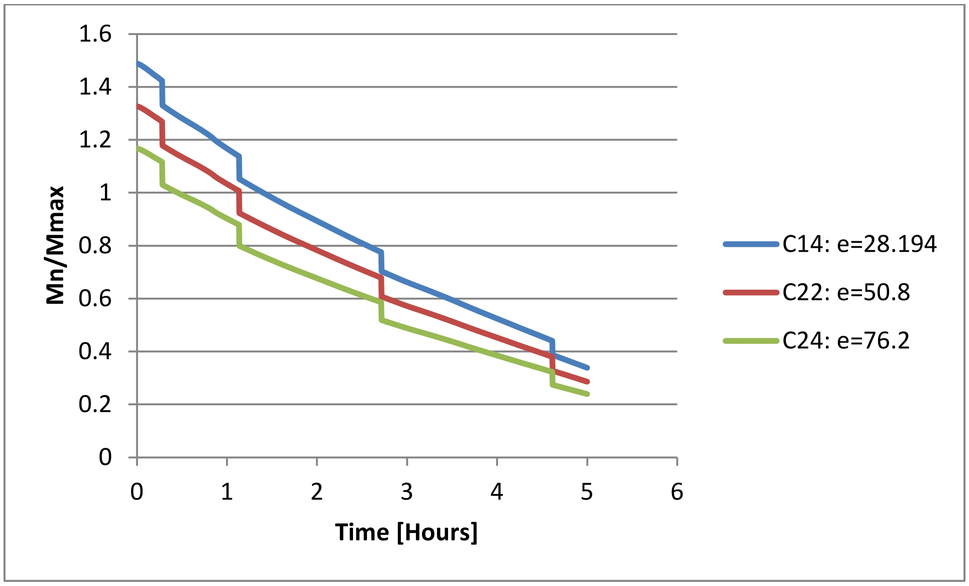


Figure 33: Carbonate Aggregate Column with Decreasing Increasing-Flexural Buckling

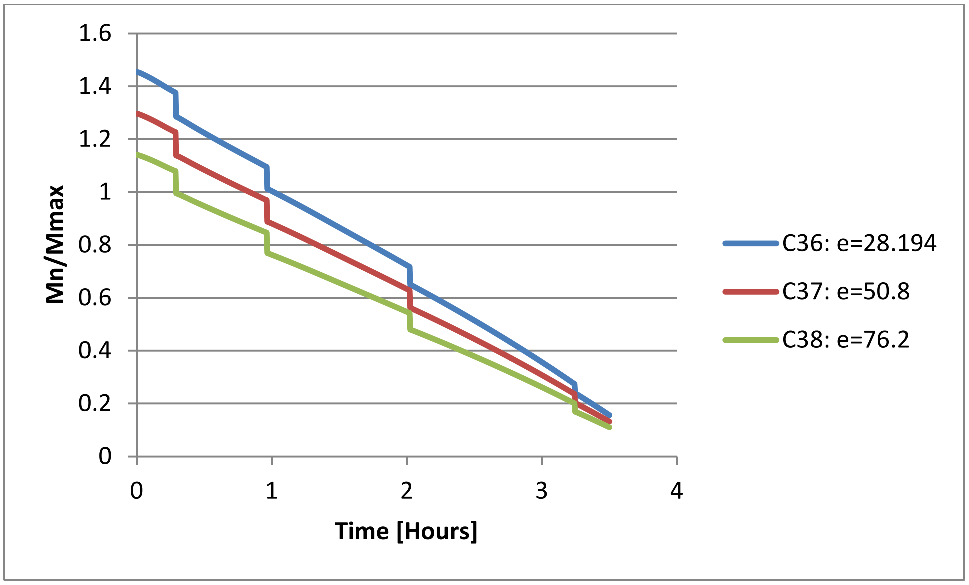


Figure 34: Siliceous Aggregate Column with Increasing Eccentricity- Flexural Buckling

For end conditions different than k=1.0, the column strength still decreases when the eccentricity of the column is increased. Tables 10 and 11 show the failure times of both carbonate and siliceous aggregate for a 3.048 meter column with the ends fixed (k =0.5).

Table 10: Failure Times (minutes) of Carbonate Aggregate Columns with End Connection k=0.5-Eccentricity

Column	Eccentricity (mm)	Crushing	Flexural Buckling
17	28.194 (ACI min)	299	258
23	50.8	274.25	163
25	76.2	163	112.75
26	101.6	75	68.25

Table 11: Failure Times (minutes) of Siliceous Aggregate Columns with End Connection k=0.5-Eccentricity

Column	Eccentricity (mm)	Crushing	Flexural Buckling
45	28.194 (ACI min)	203.25	168.75
46	50.8	191.5	121.5
47	76.2	121.5	88.75
48	101.6	66.5	58

5.2 Column Length

Increasing the length of a reinforced-concrete column decreases its fire performance. Figures 35 and 36 and Tables 12 and 13 demonstrate the results from the column study on length effects. The column used in this study was a carbonate aggregate column. Table 12 shows the failure times of the column with pin connections (k=1), and Table 12 shows the failure times of the column with fixed connections (k=0.5). As can be seen in Tables 12 and 13, the crushing mode is not affected by the increase in length of the column and the capacity is governed by flexure in all six cases. Figure 35 and 36 are the graphs corresponding to Table 12 and 13 respectively.

Table 12: Failure Times (minutes) of Carbonate Aggregate Columns with End Connection k=1.0-Length

Column	Length (m)	Crushing	Flexural Buckling
14	3.048	299.25	84.25
18	3.6576	299.25	51.5
20	4.2672	299.25	17

Table 13: Failure Times (minutes) of Carbonate Aggregate Columns with End Connection k=0.5-Length

Column	Length (m)	Crushing	Flexural Buckling
17	3.048	299	258
19	3.6576	299.25	206.75
21	4.2672	299.25	163

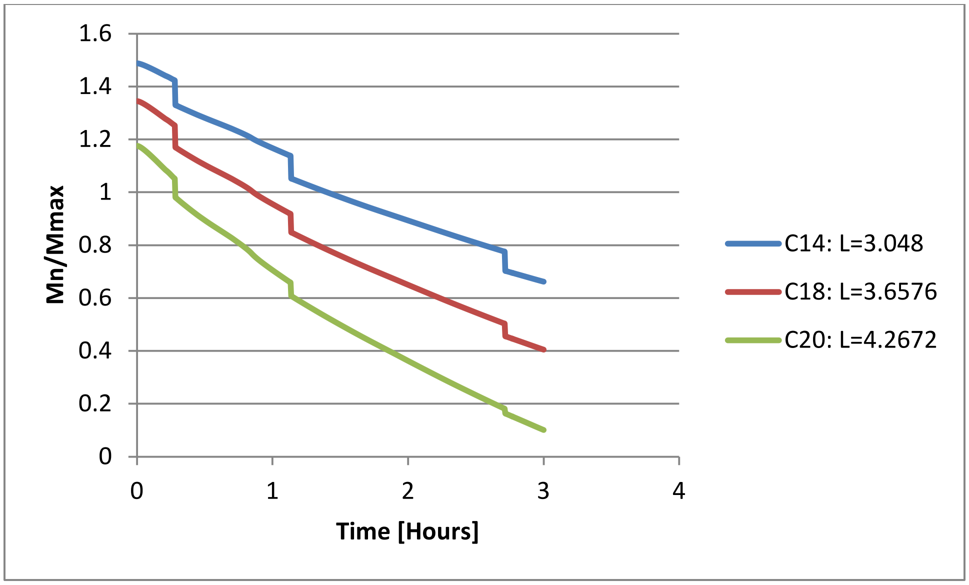


Figure 35: Carbonate Aggregate Column with Increasing Length (k=1.0)- Flexural Buckling

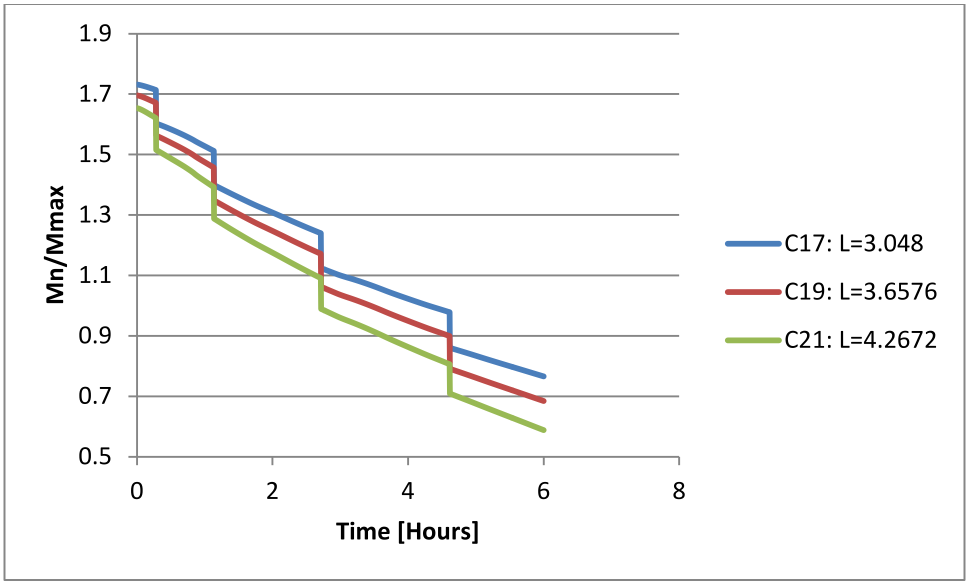


Figure 36: Carbonate Aggregate Column with Increasing Length (k=0.5)- Flexural Buckling

Both graphs can be combined to study the overall effect of the slenderness ratio (kL/r). Figure 37 and Table 14 illustrate how increasing the slenderness ratio decreases the failure time. An increase in the slenderness ratio can be achieved by either an increase in the effective length factor k (end connection coefficient), an increase in the column length, or a decrease in the radius of gyration.

Table 14: Failure Times (minutes) of Carbonate Aggregate Columns with Increasing Slenderness Ratio

Column	Slenderness Ratio	Crushing	Flexural Buckling
17	12.23	299.00	258
19	14.67	299.25	206.75
21	17.12	299.25	163
14	24.45	299.25	84.25
18	29.34	299.25	51.5
20	34.23	299.25	17

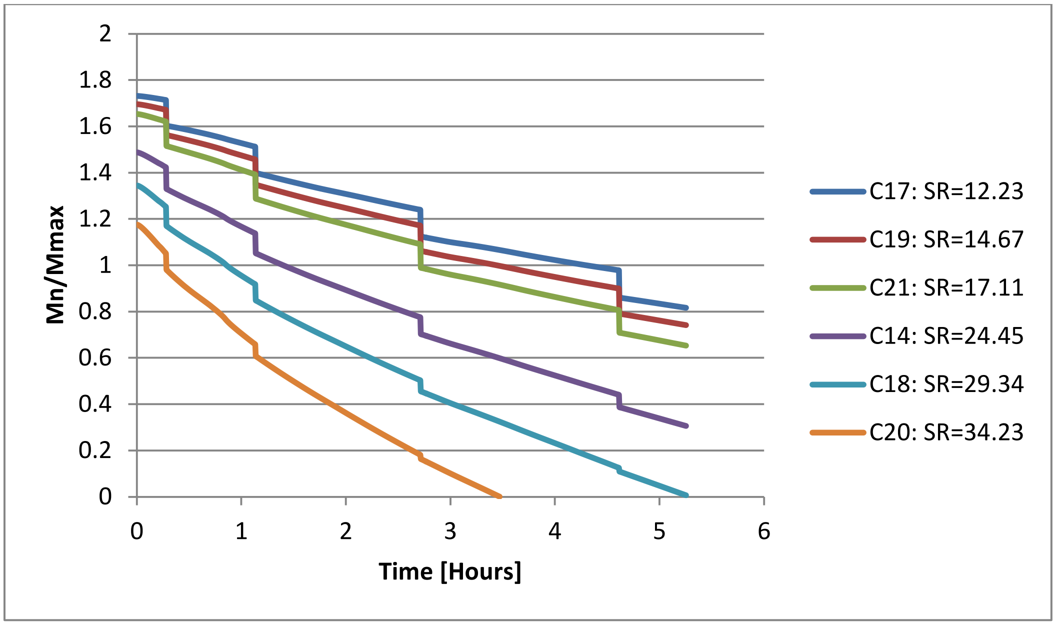


Figure 37: Carbonate Aggregate Column with Increasing Slenderness Ratio-Flexural Buckling

5.3 End Connections

Tables 15 and 16 and Figures 37 and 38 show the effects of end connections on the fire performance of a reinforced-concrete column. As the effective length factor k of the end connection increases, the failure times decrease. Table 15 show carbonate aggregate columns that are identical except for the end conditions. In every case the failure time of the k=0.5 columns is longer than that for the k=1.0 columns. Table 16 shows similar results except for siliceous aggregate columns. Figures 37 and 38 not only show that the columns with k=0.5 have larger failure times but they also have a larger moment resistance at ambient conditions (time= 0 hours).

Table 15: Failure Times (minutes) of Carbonate Aggregate Columns Differing End Connections

Column	Eccentricity (mm)	Length (m)	k Factor	Crushing	Flexural Buckling
14	28.194 (ACI min)	3.048	1	299.25	84.25
17	28.194 (ACI min)	3.048	0.5	299	258
18	28.194 (ACI min)	3.6576	1	299.25	51.5
19	28.194 (ACI min)	3.6576	0.5	299.25	206.75
20	28.194 (ACI min)	4.2672	1	299.25	17
21	28.194 (ACI min)	4.2672	0.5	299.25	163
22	50.8	3.048	1	274.25	68.25
23	50.8	3.048	0.5	274.25	163
24	76.2	3.048	1	163	27.25
25	76.2	3.048	0.5	163	112.75

Table 16: Failure Times (minutes) of Siliceous Aggregate Columns Differing End Connections

Column	Eccentricity (mm)	Length (m)	k Factor	Crushing	Flexural Buckling
36	28.194 (ACI min)	3.048	1	203.25	61
45	28.194 (ACI min)	3.048	0.5	203.25	168.75
37	50.8	3.048	1	191.5	50.25
46	50.8	3.048	0.5	191.5	121.5
38	76.2	3.048	1	121.5	17.5
47	76.2	3.048	0.5	121.5	88.75

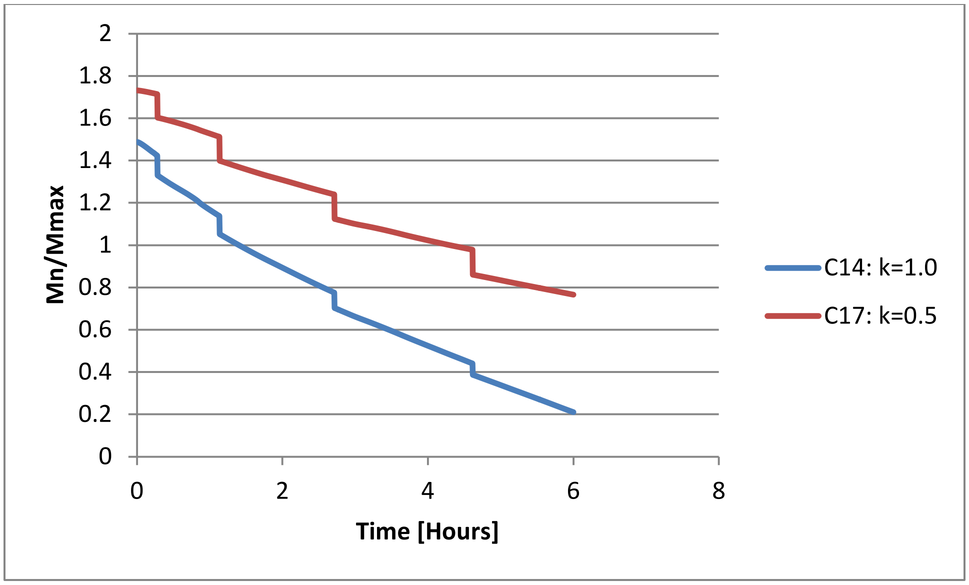


Figure 38: Carbonate Aggregate Column with Differing End Connections-Flexural Buckling

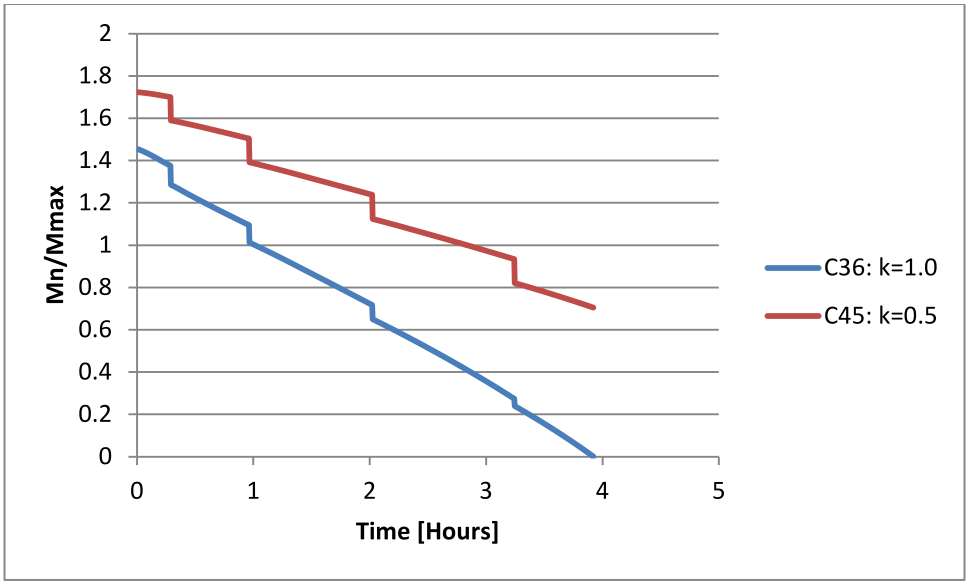


Figure 39: Siliceous Aggregate Column with Differing End Connections-Flexural Buckling

5.4 Aggregate Type

Of the two different types of aggregate analyzed in this study, carbonate aggregate yielded longer fire resistance than siliceous for both crushing and flexural buckling failure. Table 17 shows the failure times of columns with differing aggregates. All the columns in Table 17 have a length of 3.048 meters.

Figures 40 and 41 compare the axial strength of identical columns with differing aggregates. The figures clearly show the difference in failure time and axial strength degradation over time. Figure 42 shows the moment resisting strength reduction of both types of columns.

Table 17: Failure Times (minutes) of Columns with Differing Aggregates

Column	Eccentricity (mm)	Crushing	Flexural Buckling	Aggregate Type
12	0	366.25	N/A	Carbonate
35	0	263.25	N/A	Siliceous
14	28.194	299.25	84.25	Carbonate
36	28.194	203.25	61	Siliceous
22	50.8	274.25	68.25	Carbonate
37	50.8	191.5	50.25	Siliceous
24	76.2	163	27.25	Carbonate
38	76.2	121.5	17.5	Siliceous

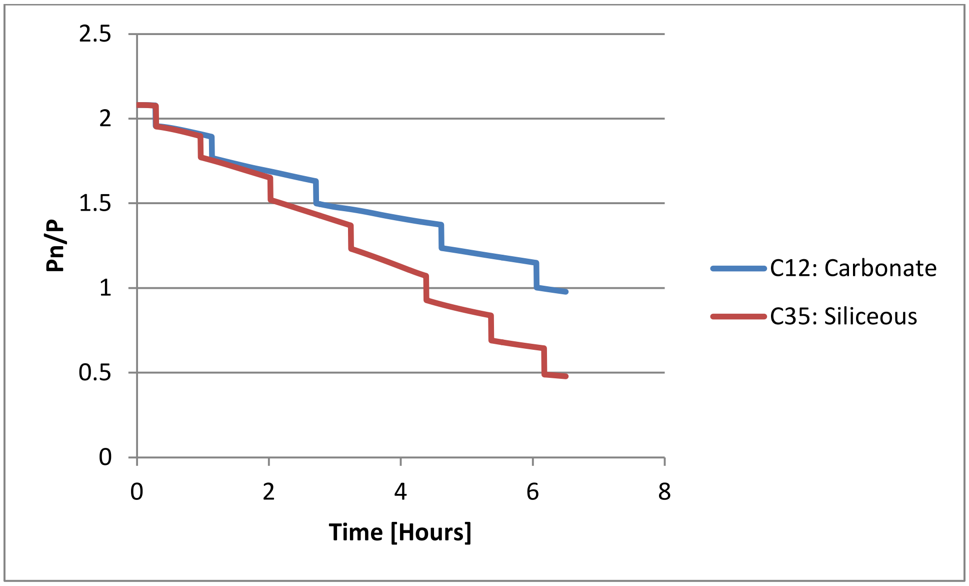


Figure 40: Different Aggregate Columns with Eccentricity=0-Crushing

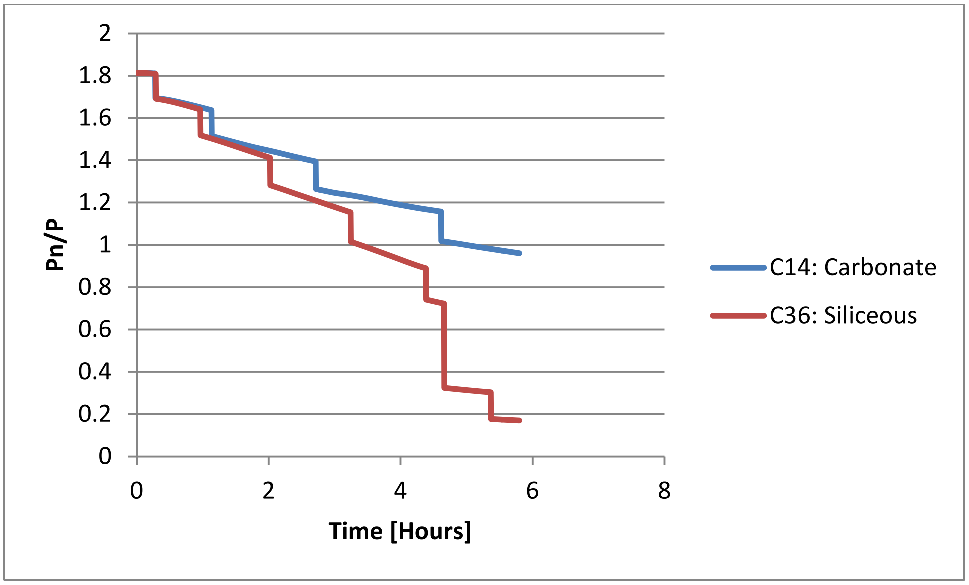


Figure 41: Different Aggregate Columns with Eccentricity=28.194-Crushing

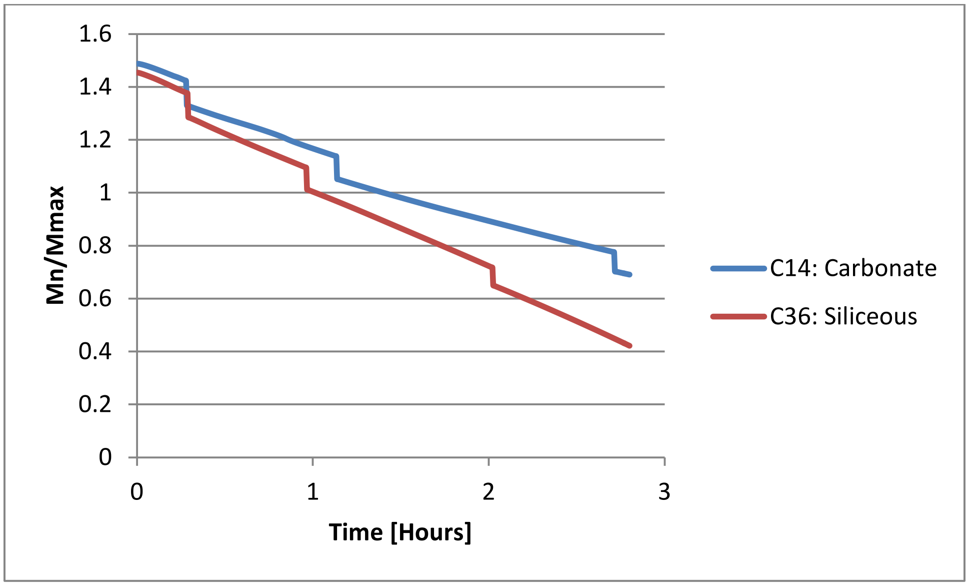


Figure 42: Different Aggregate Columns with Eccentricity=28.194-Flexural Buckling

5.5 Concrete Cover

Tables 18 and 19 show the failure times of a 3.048 meter column with the ACI minimum eccentricity for that column size (28.194 mm). The ends of the column are pin connected (k=1). Table 18 is for carbonate aggregate, and Table 19 is based on siliceous aggregate. Figures 43 to 46 show the moment and axial strength degradation over time. The tables and figures show that as the thickness of the concrete cover is decreased, the failure time for both the crushing and flexural buckling failure modes decreases.

Table 18: Failure Times (minutes) of Carbonate Aggregate Columns with Decreasing Cover

Column	Cover (mm)	Crushing	Flexural Buckling
14	63.5	299.25	84.25
29	50.8	277	76.5
32	38.1 (ACI min)	277	68.25

Table 19: Failure Times (minutes) of Siliceous Aggregate Columns with Decreasing Cover

Column	Cover (mm)	Crushing	Flexural Buckling
36	63.5	203.25	61
40	50.8	194.75	58
43	38.1 (ACI min)	163	58

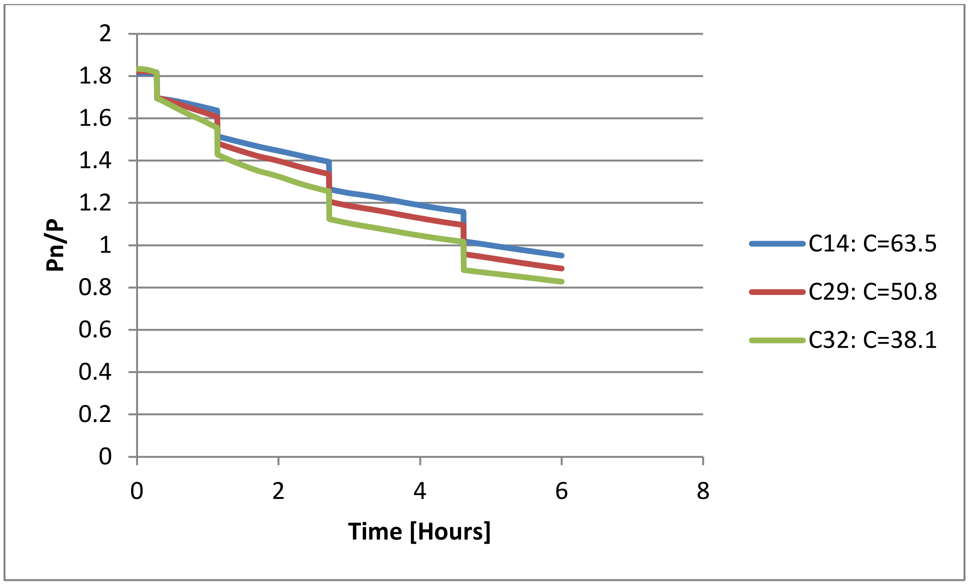


Figure 43: Carbonate Aggregate Column with Decreasing Cover-Crushing

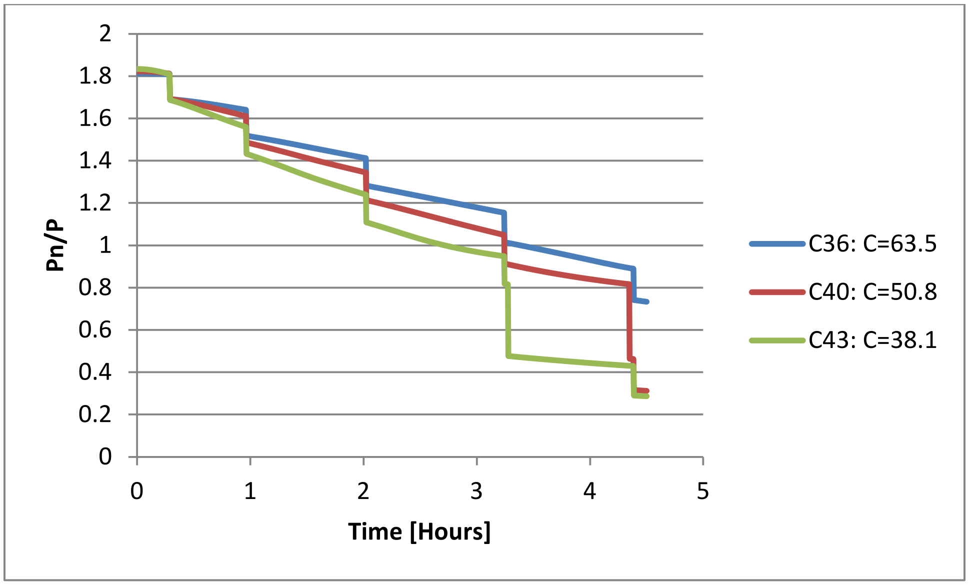


Figure 44: Siliceous Aggregate Column with Decreasing Cover-Crushing

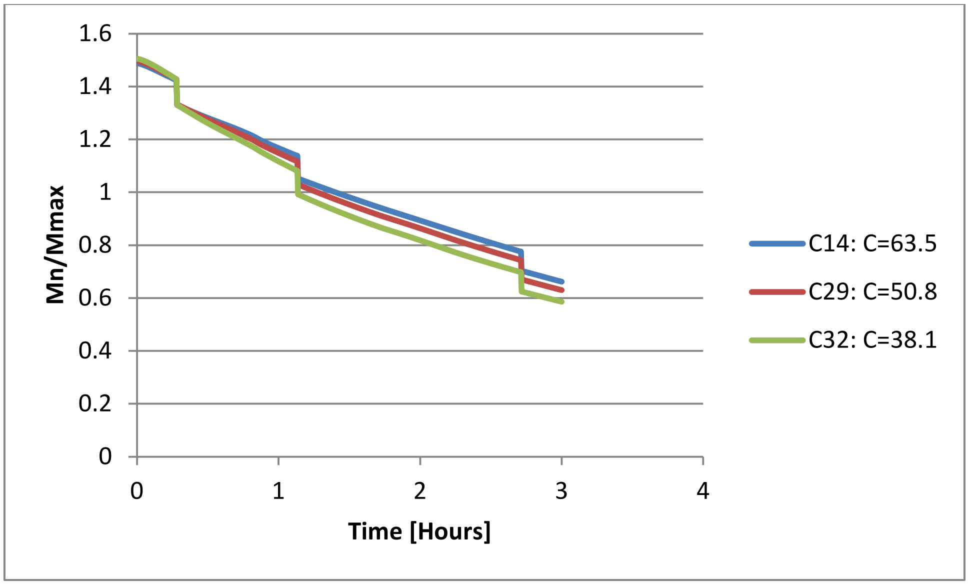


Figure 45: Carbonate Aggregate Column with Decreasing Cover-Flexural Buckling

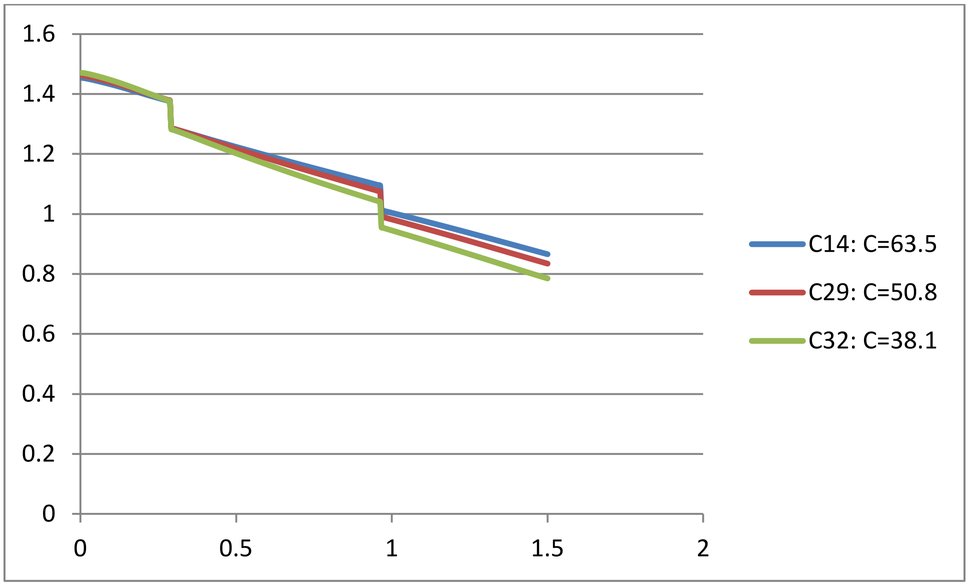


Figure 46: Siliceous Aggregate Column with Decreasing Cover-Flexural Buckling

5.6 Moment Magnification vs. Initial Applied Moment

As described in Section 3.4, the maximum moment in a column is a function of the applied moment and the secondary (PΔ) effects. The results in this section look at the difference between considering the secondary effects or ignoring them.

Figure 47 shows the generalized initial moment (M), maximum moment (M_{max}, which considers PΔ effects), and column moment strength (M_n) for a carbonate column (L=3.048) with an eccentricity of 28.194 mm. The column fails almost 5 hours earlier when moment magnification is taken into effect. Figure 48 shows a similar result except the aggregate is siliceous. Table 20 summarizes the calculated failure times.

Table 20: Failure Times (minutes) of Columns Comparing Moment Magnification vs. Applied Moment

Column	M	M_{max}	Aggregate
14	299	84.25	Carbonate
36	203	61	Siliceous

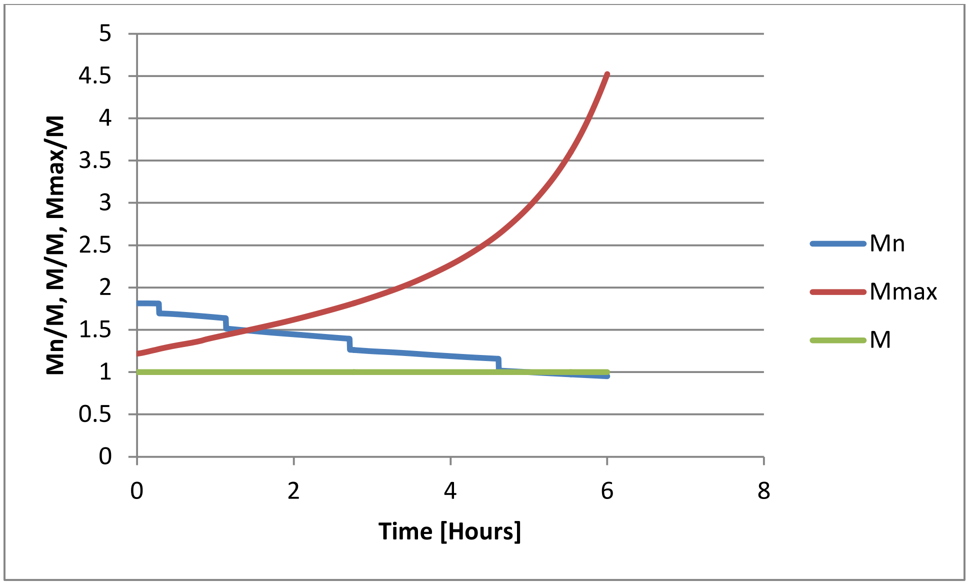


Figure 47: Comparison of Moment Magnification and Applied Moment-Carbonate

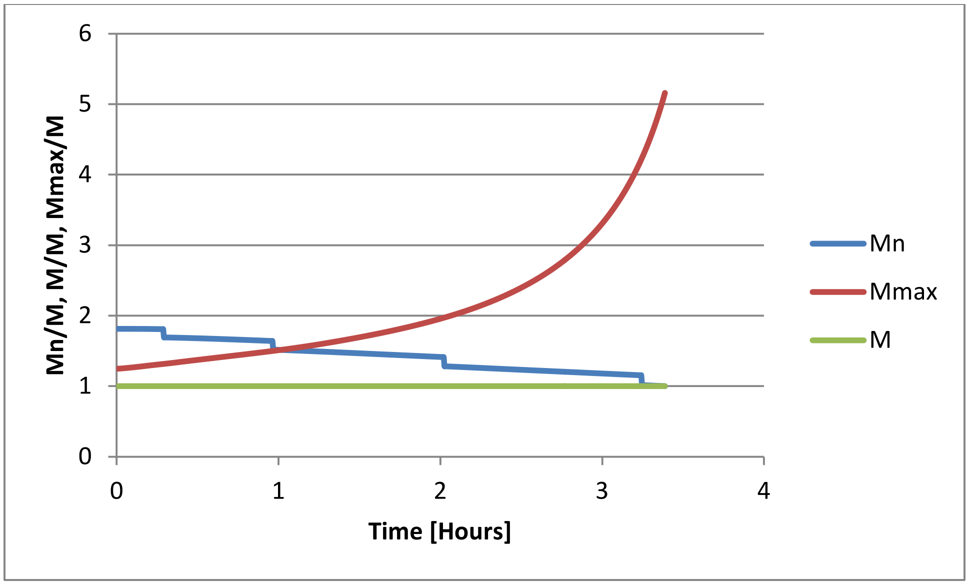


Figure 48: Comparison of Moment Magnification and Applied Moment-Siliceous

5.7 Section Properties

Various section properties such as plastic centroid location, eccentricity, and slenderness ratio change over time as a result of the deteriorating cross-section. This section illustrates the extent which these properties change over time.

5.7.1 Slenderness Ratio

During fire exposure, the slenderness ratio of the cross-section will increase due to the decreasing area enclosed by the 500°C isotherm. As the slenderness ratio increases, the failure time will decrease. Table 21 shows the flexural buckling failure time decreasing. The slenderness ratio listed in Table 21 is the initial slenderness ratio of the column before it is subjected to the fire curve. Figure 49 demonstrates the increasing slenderness ratio over the time of a fire curve and also the relation between the slenderness ratios and the ACI slenderness ratio design criteria. Figure 50 shows the decrease in the moment capacity of the column and the failure times of the column in Table 21. It can be seen in Figure 50 that the slope of the decreasing moment capacity curves for columns with a larger initial slenderness ratio are steeper than those for columns with smaller slenderness ratios.

Table 21: Failure Times (minutes) of Columns with Increasing Slenderness Ratio

Column	Slenderness Ratio	Crushing	Flexural Buckling
17	12.23	299	258
19	14.67	299.25	206.75
21	17.12	299.25	163
14	24.45	299.25	84.25
18	29.34	299.25	51.5
20	34.23	299.25	17

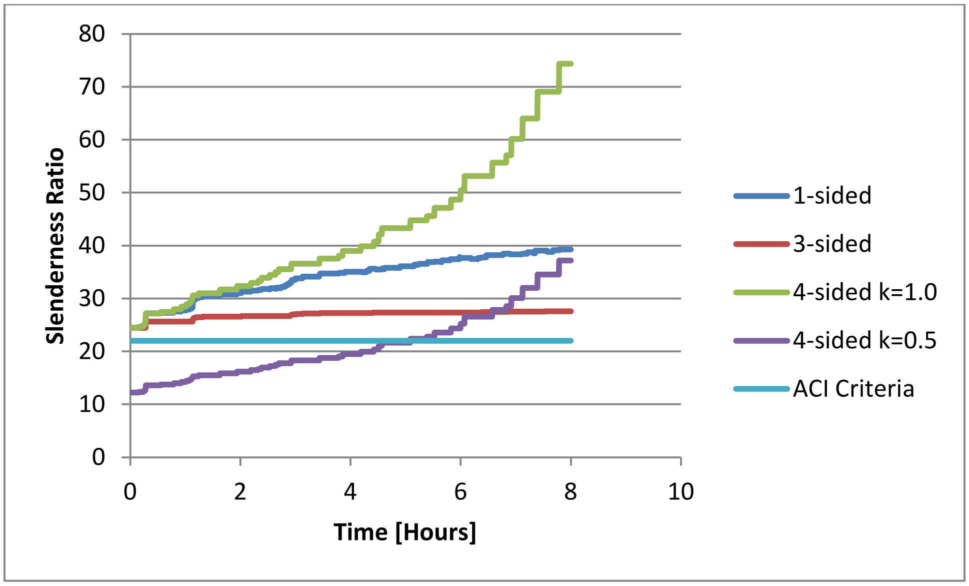


Figure 49: Comparison of Increasing Slenderness Ratios of Columns with Varying Fire Exposures

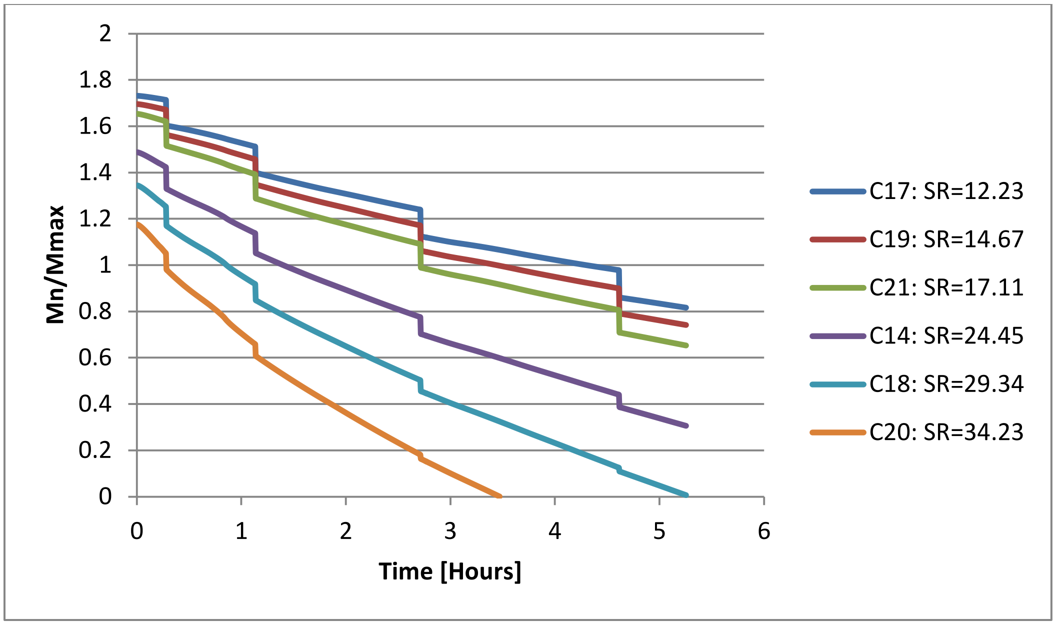


Figure 50: Decreasing Moment Capacity of Columns with Increasing Slenderness Ratio

5.7.2 Eccentricity and Plastic Centroid

The plastic centroid of a column will move depending on the shape and location of the 500°C isotherm. Because the eccentricity of load is measured from the location of the plastic centroid, as the plastic centroid moves the eccentricity will increase or decrease. A three-sided fire exposure was utilized to demonstrate the movement of the plastic centroid and the eccentricity. Figures 51 and 52 show the fire exposures used. In these two figures the top and bottom faces of the cross-section both have fire exposures, and the left and right sides change from ambient to having fire exposure depending on the analysis. Table 22 shows the difference in failure time for three-sided fire exposures with the ambient temperature sides changing. When the ambient temperature is on the right sided, the side with the eccentricity, the failure time is 309.5 minutes. However, when the ambient temperature side is on the left, opposite the eccentricity, the failure time is about 4 hours less.

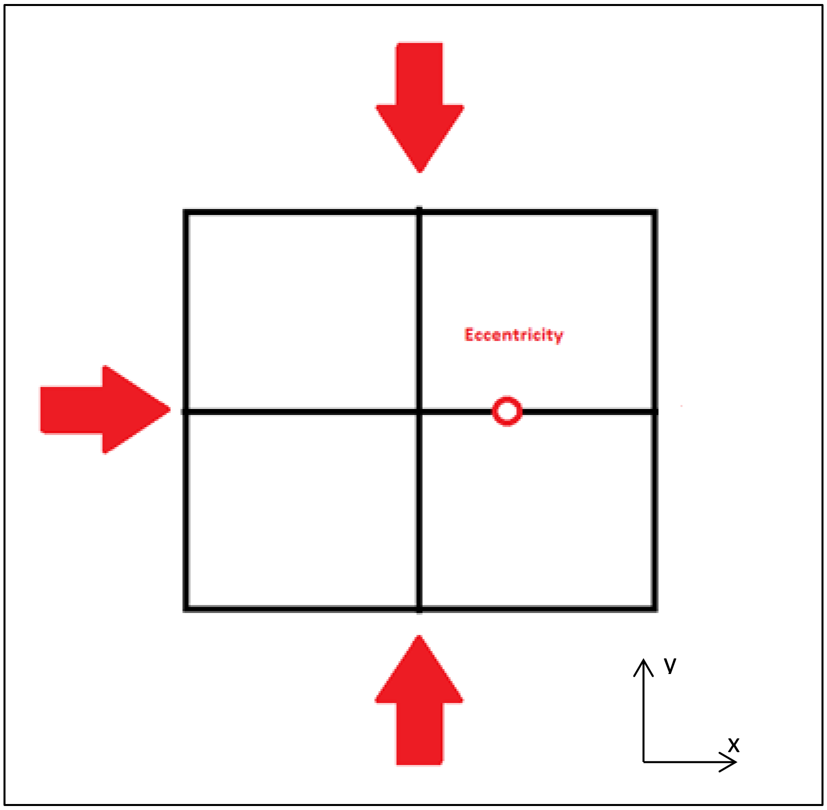


Figure 51: Three-Sided Fire Exposure with Opposite Side Eccentricity

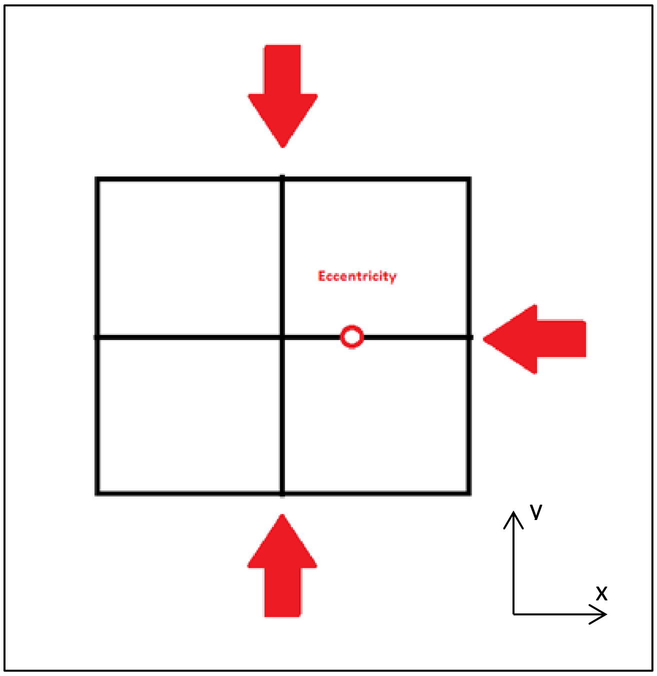


Figure 52: Three-Sided Fire Exposure with Same Side Eccentricity

Table 22: Failure Times (minutes) of Columns with 3-Sided Fire Exposure

Column	Ambient Side	Failure Time
51	Right	309.5
52	Left	68.25

Figure 53 shows the eccentricity of the columns in Table 22. The figure demonstrates the same principle. The columns start with identical eccentricities and as the fire shifts the location of the plastic centroid, the eccentricity either increases or decreases. Figures 54 and 55 show the reduction in the moment resisting strength and the maximum moment in the columns.

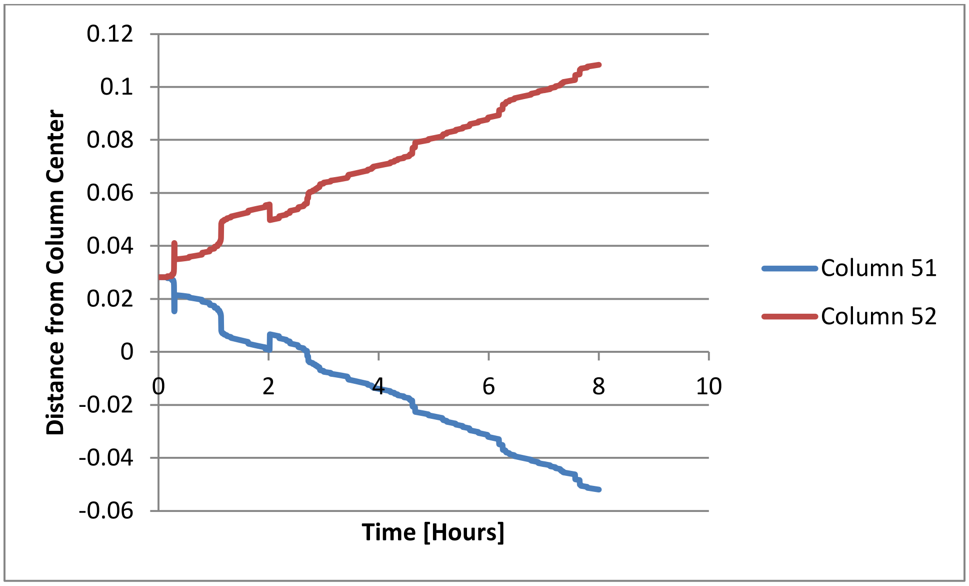


Figure 53: Eccentricity Length of Columns with 3-Sided Fire Exposures

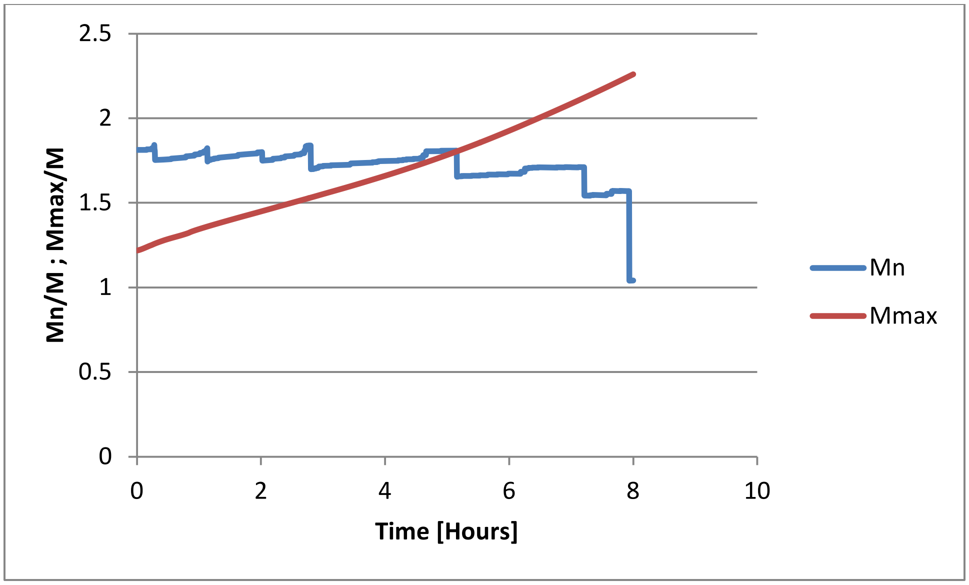


Figure 54: Moment Resistance and Maximum Moment of Column 51

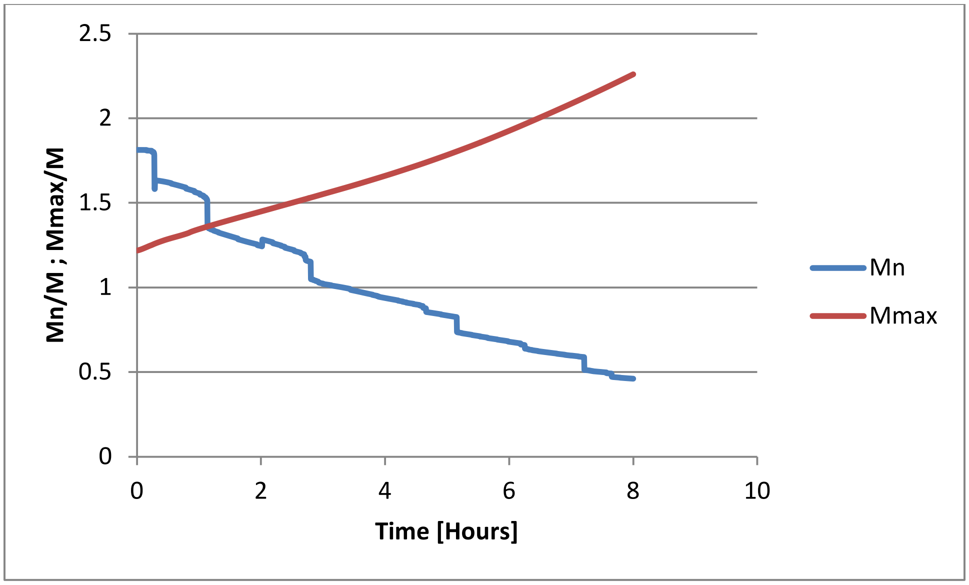


Figure 55: Moment Resistance and Maximum Moment of Column 52

Because of the movement of the plastic centroid and the increases in eccentricity and thus the maximum moment on a column, the failure time of a 3-sided fire with the ambient sided opposite the eccentricity location will fail sooner than an identical column subjected to 4-sided heating. This is due to the fact that the eccentricity is increasing. This is demonstrated in Table 23.

Table 23: Comparison of Column Failure Times (minutes) for 3- and 4-Sided Fire Exposures

Column	Fire Exposure	Failure Time
14	4-sided	84.25
52	3-sided	68.25

5.7.3 Latent Heating

Latent heating occurs when a column is subjected to a natural fire curve. Due to the low conductivity and low thermal diffusivity of concrete, the temperature in the concrete column will increase and decrease slowly. During the cooling phase of the fire, the temperature in the interior of the column may continue to rise due to the elevated temperature still inside the cross-section. This latent temperature increase is shown in Figure 56. The temperature of node 21, an exterior node, increases and decreases with the rise and fall of the gas temperature. However, the temperature of node 14, an interior node, continues to increase several hours after the gas temperature has reached its peak and decreased back to ambient conditions.

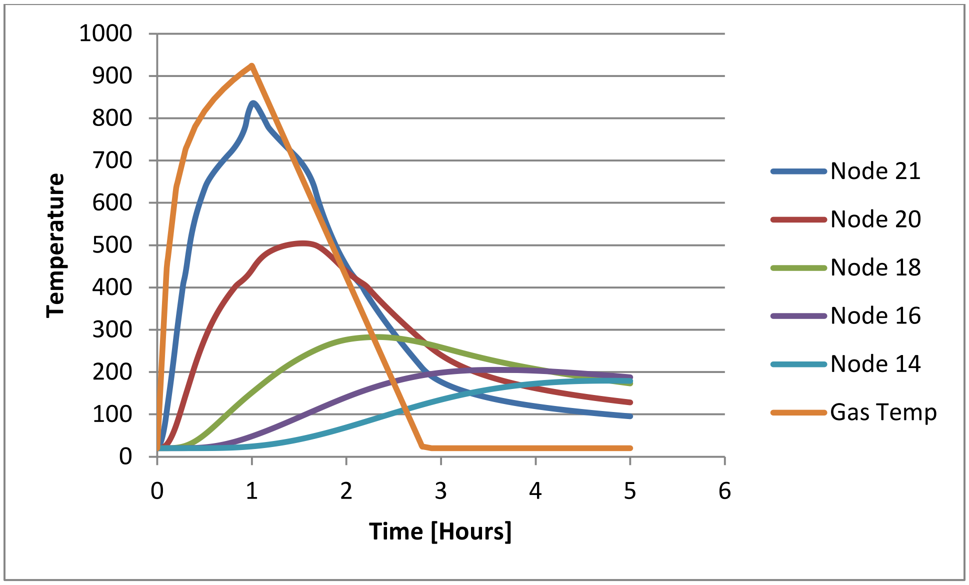


Figure 56: Temperature Profiles of Nodes Subjected to a Natural Fire-3-Sided Fire Exposure

The increases in temperature could result in column failure during the cooling phase of the fire. This is shown in Figure 57. The burning duration of the fire lasted for one hour before the gas temperature started to cool. Failure of the column occurred approximately one half-hour after the fire began to cool. This is indicated by the green vertical line in Figure 57.

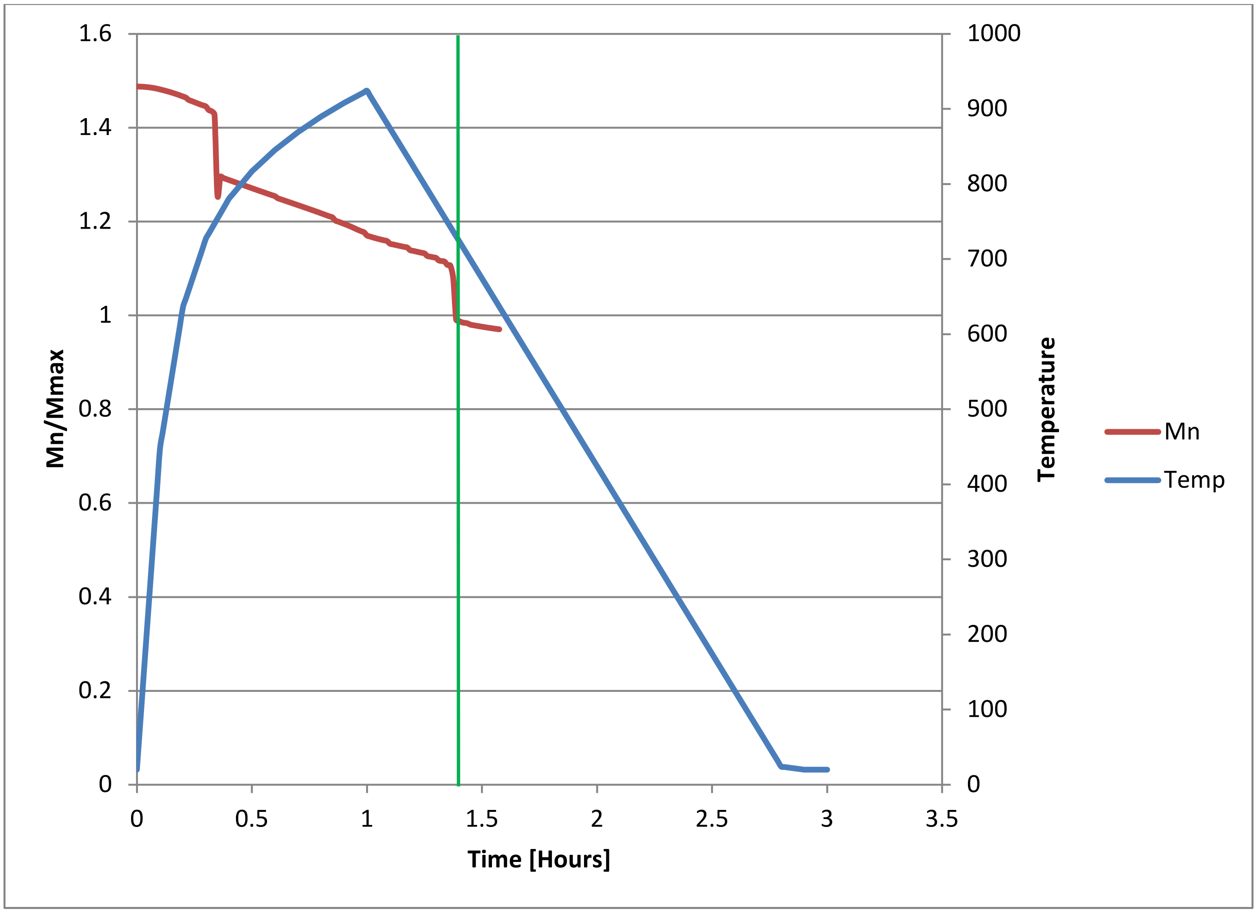


Figure 57: Moment Resistance of a Column Subjected to a Natural Fire-3-Sided Fire Exposure

6. Conclusions

The primary goal of this thesis is to study the behavior of reinforced-concrete columns under fire conditions and to analyze the various characteristics of a reinforced-concrete column that affect its overall fire performance. A secondary goal of the thesis is to provide a tool to help further the education of engineers in the analysis of reinforced-concrete columns under fire exposures.

To accomplish these goals, a numerical calculation model was constructed in *Microsoft Excel*. The heat transfer and the structural analysis portions of the model were benchmarked using *ANSYS* and published results of experimental reinforced-concrete column furnace tests, respectively. After the model was shown to be working correctly and providing acceptable results, it was used to perform a study on the behavior of a reinforced-concrete column under various fire exposures and variations of column characteristics. The overall conclusions of this thesis are discussed in the following sections as well as recommendations for further research into the topic of reinforced-concrete column behavior in fire conditions.

6.1 Column Behavior

From the column studies completed, it was shown that increasing the initial load eccentricity on a column reduces the column's fire performance for both carbonate and siliceous aggregate. The time to failure for both the crushing and flexural buckling modes was decreased significantly as the eccentricity was increased by several millimeters above the ACI specified minimum value.

Increasing the column length also decreases the performance of a reinforced-concrete column. However, unlike increasing the load eccentricity, increasing the length of the column does not decrease the crushing failure time. Only the flexural buckling failure time is decreased with increasing column length. The same is also true for increasing the slenderness ratio of the column. As the slenderness ratio increases, the flexural buckling failure time decreases as the crushing failure time remains the same.

The results also show that increasing fixity of the end connections dramatically increases the failure time. For two identical columns in which one has pinned connections and the other has fixed connections, the failure time is approximately two hours more for the fixed column than the pinned column. This result is also demonstrated in the slenderness ratio results. Increasing the amount of cover on a reinforced-concrete cover will also increase the failure time. The increase in time is several minutes.

The use of carbonate aggregate over siliceous aggregate yields longer failure times. This is due in part to the deterioration in the compressive strength of the concrete as well as due to the thermal properties of both types of aggregate. Because siliceous aggregate allows the cross-section to increase in temperature faster than carbonate aggregate, a siliceous aggregate column will lose cross-section faster and thus lose strength leading to failure faster.

According to *ACI 318* if a column's slenderness ratio is below a certain criteria slenderness effects can be ignored. However, the results show that when dealing with fire conditions, slenderness effects cannot be neglected. A column's load eccentricity depending on the fire exposure will increase along with the slenderness ratio. The difference in failure time between a column for which slenderness effects were ignored and the same column with those effects accounted for is approximately four hours. These findings indicate that the location of the plastic centroid and its effects on the overall capacity of the column need to be analyzed.

Finally, the phenomenon of latent heating can lead to collapse of a reinforced-concrete column during the cooling phase of a fire curve. This is a dangerous threat due to the timing of such a collapse. Firefighters and first-responders could be in the building continuing to rescue occupants. The results showed that failure could occur one half-hour after the fire starts to decay. This is consistent with published data on the subject of latent heating.

6.2 Numerical Calculation Model

The numerical calculation model for determining the fire performance of a reinforced-concrete column yielded good approximations to the failure times obtained from the published test data. The model does not take into account water effects, spalling of the concrete, spiral reinforcement, and it uses the 500°C isotherm method for calculating the column's strength capacity. All these factors could result in the observed differences between the actual failure time and the model's predicted failure time.

Several factors possibly lead to the differences in the heat transfer portion of the model and the results from *ANSYS*. The first difference is that the equations for temperature-dependent thermal properties used in the thesis model could not be input in *ANSYS*. The thermal properties remained constant in *ANSYS*. Also, the radiation model in *ANSYS* was not taken into account due to its complexity. The thesis model used a combined convective and radiation model based solely on the hot gases surrounding the column. *ANSYS* required a point source of radiation and a distance to the point source to calculate radiation. Therefore, radiation was neglected in *ANSYS*. All these dissimilarities could result in the differences between nodal temperatures of *ANSYS* and the thesis model.

However, even with these assumptions, the model yielded fairly accurate results against the benchmarking test performed both for the heat transfer and structural analysis portions. The model is considered an acceptable level 2 evaluation tool for calculating the fire performance of a reinforced-concrete column.

6.3 Recommendations for Future Work

The following is a list of recommendations for future work in the area of reinforced-concrete column behavior and numerical modeling.

- To better simulate the thermal effects of a fire, an incident heat flux could be used instead of calculating the thermal energy transfer to the column as a changing temperature of the hot gases at the boundary of the column. This would allow actual heat flux fire curves to be used and comparison to real fire and collapse events could be made.

- The thesis model assumes that the temperature and strength reductions calculated for two dimensions are uniform for the entire column. However, columns are three-dimensional and in a fire event the temperatures would not be uniform from floor to ceiling. Extending the model to account for three-dimensional heating and strength reduction including nonlinear effects, would yield more realistic results and provide a base to investigate the *ACI318* equations for slenderness effects and moment magnification.
- Accounting for each temperature isotherm and its contribution to the column strength would yield more accurate results than the 500°C isotherm method. Each finite element nodal temperature would have its own strength and contribution to the column strength. A detailed study into the strength equations and how to apply them to smaller elements and multiple isotherms would have to be conducted.
- Implementation of water effects, spiral reinforcement, and spalling into the model would increase the accuracy of the model.

7. Bibliography

Albano, L. (2013, March 11). Fire-Temperature Time Relationships. *CE 534: Structural Design for Fire Condtions*. Worcester.

Allen, D. E., & Lie, T. T. (1974). Fire Resistance of Reinforced Concrete Columns. *International Association For Bridge and Structural Engineering Symposium on Design and Safety of Reinforced Concrete Compression Members* (pp. 245-254). Quebec: National Research Council Canada.

American Concrete Institute. (2004). *Building Code Requirements for Structural Concrete (ACI 318-05) and Commentary (ACI 318R-05).* Farmington Hills: American Concrete Institute.

American Society of Civil Engineers. (1992). *Structural Fire Protection.* New York: American Society of Civil Engineers.

Baughn, J. W., & Rossi, M. (1992). Two-Dimensional Transient Heat Conduction Analysis Using Spreadsheets. *Heat Transfer Engineering, 13*(2), 71-79.

Besser, R. S. (2002). Spreadsheet Solutions to Two-Dimensional Heat Transfer Problems. *Chemical Engineering Education*, 160-165.

Besser, R. S. (2002). Spreadsheet Solutions To Two-Dimensional Heat Transfer Problems. *Chemical Engineering Education*(Spring), 160-165.

Buchanan, A. H. (2002). *Structural Design for Fire Safety.* Chichester: John Wiley & Sons LTD.

Dimia, M. S., Guenfoud, M., Gernay, T., & Franssen, J.-M. (2011). Collapse of Concrete Columns During and After the Cooling Phase of a Fire. *Journal of Fire Protection Engineering*, 245-263.

Dotreppe, J. C., Franssen, J. M., Bruls, A., Baus, R., Vandevelde, P., Minne, R., et al. (1996, June). Experimental Research on the Determination of the Main Parameters Affecting the Behaviour of Reinforced Concrete Columns Under Fire Conditions. *Magazine of Concrete Research, 49*(179), 117-127.

Eid, J. C. (1987). A Methodology and Tutorial for Thermal Modeling with PC Spreadsheets. *Heat Transfer Engineering, 8*(2), 95-107.

Fitzgerald, R. W. (2004). *Building Fire Performance Analysis.* West Sussex: John Wiley & Sons Ltd.

Franssen, J.-M. (2000). Improvement of the Parametric Fire of Eurocode 1 based on Experimental Test Results. *Fire Safety Science- Proceedings of the Sixth International Symposium* (pp. 927-938). Poitiers: Curtat ed.

International Code Council. (2009). *International Building Code.* Country Club Hills: International Code Council.

Kharab, A. (1995). Use of a Spreadsheet Program in a Two-Dimensional Heat Conduction Program. *Computer Methods in Applied Mechanics and Engineering*(122), 173-181.

Kreith, F., & Bohn, M. S. (2001). *Principles of Heat Transfer.* Pacific Grove: Brooks/Cole.

Lie, T. T., & Irwin, R. J. (1993). Method to Calculate the Fire Resistance of Reinforced Concrete Columns with Rectangular Cross Section. *ACI Structural Journal*, 52-60.

Malhotra, H. L. (1982). *Design of Fire Resisting Structures.* Glasgow: Surrey University Press.

Park, R., & Paulay, T. (1975). *Reinforced Concrete Structures.* New York: John Wiley & Sons.

Schumack, M. (2004). Use of a Spreadsheet Package to Demonstrate Fundamentals of Computational Fluid Dynamics and Heat Transfer. *International Journal of Engineering Education*, *20*(6), 974-983.

Society of Fire Protection Engineers. (2002). *SFPE Handbook of Fire Protection Engineering.* Quincy: National Fire Protection Association.

Society of Fire Protection Engineers. (2007). *SFPE Engineering Guide to Performance-Based Fire Protection.* Quincy: National Fire Protection Association.

Society of Fire Protection Engineers. (2008). *The SFPE Handbook of Fire Protection Engineering.* Quincy: National Fire Protection Association.

Wang, C.-K., Salmon, C. G., & Pincheira, J. A. (2007). *Reinforced Concrete Design.* Hoboken: John Wiley & Sons, Inc.

Woollerton, J. L., & Lie, T. T. (1988). *Fire Resistance of Reinforced Concrete Columns: Test Results.* National Research Council Canada.

Yao, Y., & Tan, K. H. (2003). Fire Resistance of Four-Face Heated Reinforced Concrete Columns. *Journal of Structural Engineering*, 1220-1229.

Appendix A-Material Property Equations

Aggregate Thermal Conductivity (American Society of Civil Engineers, 1992)

Siliceous Aggregate Concrete

For $0 \leq T \leq 800^{\circ}C$;	$k_c = -0.000625T + 1.5\ Wm^{-1}{}^{\circ}C^{-1}$
For $T > 800^{\circ}C$;	$k_c = 1.0\ Wm^{-1}{}^{\circ}C^{-1}$

Pure Quartz Aggregate Concrete

For $0 \leq T \leq 800^{\circ}C$;	$k_c = -0.00085T + 1.9\ Wm^{-1}{}^{\circ}C^{-1}$
For $T > 800^{\circ}C$;	$k_c = 1.22\ Wm^{-1}{}^{\circ}C^{-1}$

Carbonate Aggregate Concrete

For $0 \leq T \leq 293^{\circ}C$;	$k_c = 1.355\ Wm^{-1}{}^{\circ}C^{-1}$
For $T > 293^{\circ}C$;	$k_c = -0.001241T + 1.762\ Wm^{-1}{}^{\circ}C^{-1}$

Aggregate Thermal Capacity (American Society of Civil Engineers, 1992)

Siliceous Aggregate Concrete

For $0 \leq T \leq 200^{\circ}C$;	$\rho_c c_c = (0.005T + 1.7) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $200^{\circ}C < T \leq 400^{\circ}C$;	$\rho_c c_c = 2.7 \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $400^{\circ}C < T \leq 500^{\circ}C$;	$\rho_c c_c = (0.013T - 2.5) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $500^{\circ}C < T \leq 600^{\circ}C$;	$\rho_c c_c = (-0.013T + 10.5) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $T > 600^{\circ}C$;	$\rho_c c_c = 2.7 \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$

Carbonate Aggregate Concrete

For $0 \leq T \leq 400^{\circ}C$;	$\rho_c c_c = 2.566 \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $400^{\circ}C < T \leq 410^{\circ}C$;	$\rho_c c_c = (0.1765T - 68.034) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $410^{\circ}C < T \leq 445^{\circ}C$;	$\rho_c c_c = (-0.05043T + 25.00671) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $445^{\circ}C < T \leq 500^{\circ}C$;	$\rho_c c_c = (2.566) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $500^{\circ}C < T \leq 635^{\circ}C$;	$\rho_c c_c = (0.01603T - 5.44881) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $635^{\circ}C < T \leq 715^{\circ}C$;	$\rho_c c_c = (0.16635T - 100.90225) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $715^{\circ}C < T \leq 785^{\circ}C$;	$\rho_c c_c = (-0.22103T + 176.07343) \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$
For $T > 785^{\circ}C$;	$\rho_c c_c = 2.566 \times 10^6\ Jm^{-3}{}^{\circ}C^{-1}$

Steel (ASTM A36) Properties (Society of Fire Protection Engineers, 2008)

Thermal Conductivity (k_s), Specific Heat (c_s), and Density (ρ_s)

For $0 \leq T \leq 900^{\circ}C$;	$k_s = -0.22T + 48\ Wm^{-1}{}^{\circ}C^{-1}$
For $T > 900^{\circ}C$;	$k_s = 28.2\ Wm^{-1}{}^{\circ}C^{-1}$

For $0 \leq T \leq 650^{\circ}C$;	$c_s = 0.51T + 420\ Jkg^{-1}C^{-1}$
For $650 \leq T \leq 725^{\circ}C$;	$c_s = 8.65T + 4870\ Jkg^{-1}C^{-1}$
For $725 \leq T \leq 800^{\circ}C$;	$c_s = -10.9T + 9340\ Jkg^{-1}C^{-1}$
For $T > 800^{\circ}C$;	$c_s = 578\ Jkg^{-1}C^{-1}$

ρ_s = 7860 kgm^{-3}

Yield Strength (σ_{yT}) and Modulus of Elasticity (E_T)

For 0 ≤ T ≤ 600°C;

$$\sigma_{yT} = \left(1 + \frac{T}{900\ln\left(\frac{T}{1750}\right)}\right)\sigma_{y0}$$

$$E_T = \left(1 + \frac{T}{2000\ln\left(\frac{T}{1100}\right)}\right)E_0$$

For T > 600°C;

$$\sigma_{yT} = \frac{340 - 0.34T}{T - 240}\sigma_{y0}$$

$$E_T = \frac{690 - 0.69T}{T - 53.5}E_0$$

Appendix B-User Guide

All commands listed in the following user guide are specifically for *Microsoft Excel 2010.*

Getting Started

Before using the spreadsheet, macros need to be enabled for *Excel*. The process below details how to accomplish this.

- Open the workbook
- Select File>>Options>>Trust Center>>Trust Center Settings
- In the Trust Center Window, Select Macro Settings>>Enable all macros
- Close and reopen the workbook to allow changes to take effect

Overview of Sheets

There are four sheets in the workbook: Calculations, Results, Sample Fire Curves, and Variable Diagram. Each is used for a specific function of the spreadsheet model. Do not delete any of the sheets.

“Calculations” Sheet

Figure 1 shows the interface of the “Calculations” sheet. This is the main interface for performing the three (3) processes of the spreadsheet: heat transfer analysis, structural analysis, and post-processing. The three command buttons are shown in Figure 1. Use of these buttons is described in the Command Processes section. Columns A and B are for the user to enter any time-temperature curve desired.

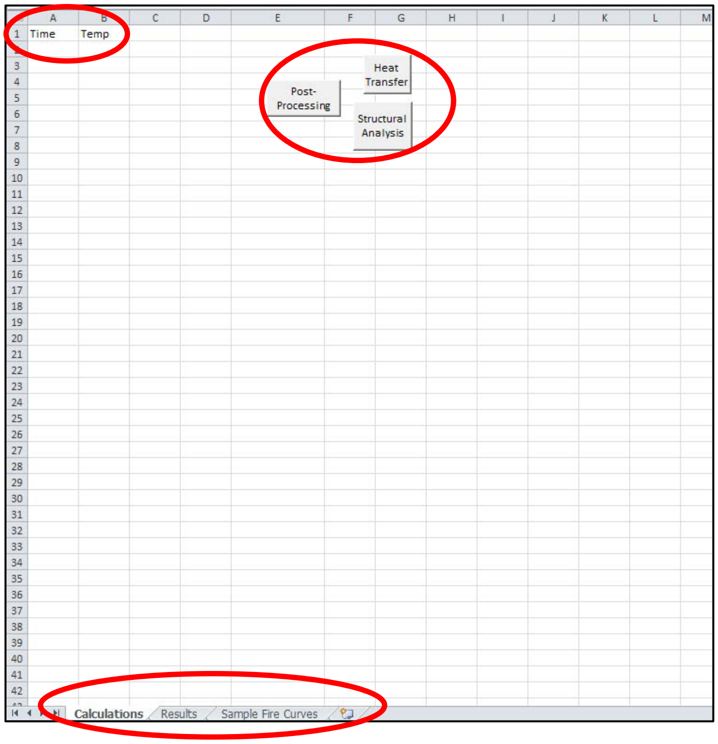


Figure 1: "Calculations" Interface

“Results” Sheet

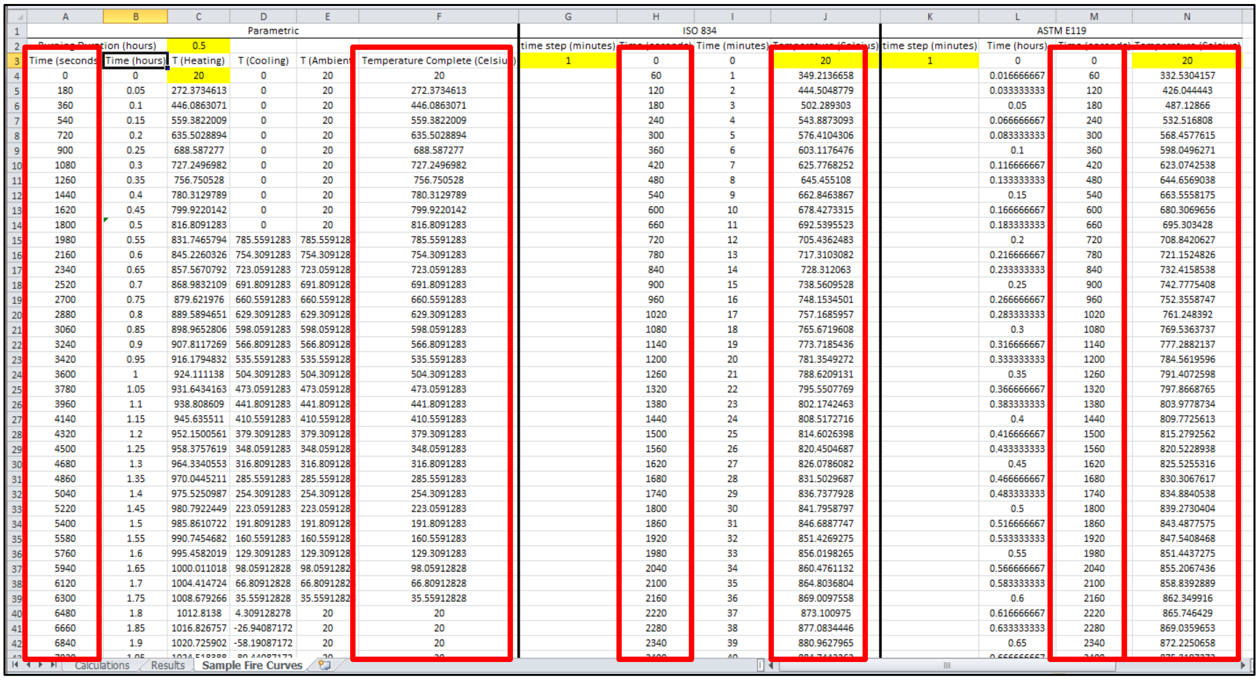
The “Results” sheet is the location where the post-process command compiles the desired results from the heat transfer and structural analyses. Column A is the time. Columns B through J are for up to 9 cross-section nodal temperature histories. Columns K to T are structural analysis variables. The specific columns and their variables are shown in Table 1.

Table 1: Variable Classification of Specific Columns in the "Results" Sheet

Variable	Column
Time	A
Nodal Temperature Histories	B-J
Left Steel Temperature	K
Right Steel Temperature	L
Concrete Area	M
Moment Strength	N
Axial Strength	O
Maximum Moment	P
Balanced Eccentricity (from plastic centroid)	Q
Plastic Centroid Location (from left face)	R
Slenderness Ratio (x-direction)	S
Slenderness Ratio (y-direction)	T

"Sample Fire Curves" Sheet

The "sample fire curves" sheet (Figure 2) provides three (3) sample fire curves for the user: ASTM E119, ISO 834, and a parametric curve (simulates natural fire curve). Each of the fire curve equations are detailed in the thesis document. The only cells subject to changes by the user are those in yellow. For the ASTM E119 and ISO 834, the user enters the desired time step and ambient temperature. For the parametric fire curve, the user enters the burning duration and the ambient temperature. The usable columns for the time-temperate columns in the "Calculations" sheet are, for the ISO 834 and ASTM E119, the "Time (seconds)" and "Temperature (Celsius)" columns. The usable columns for the parametric fire curve are the "Time (seconds)" and the "Temperature Complete (Celsius)" column. The other columns are for calculation purposes of the various columns. The user needs to copy the desired time-temperature columns and paste them into the respective columns in the "Calculations" sheet. Make sure to select "paste values" when pasting the time-temperature curves.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Parametric						ISO 834				ASTM E119			
2	Burning Duration (hours)		0.5				time step (minutes)	Time (seconds)	Time (minutes)	Temperature (Celsius)	time step (minutes)	Time (hours)	Time (seconds)	Temperature (Celsius)
3	Time (seconds	Time (hours)	T (Heating)	T (Cooling)	T (Ambient	Temperature Complete (Celsius)	1	0	0	20	1	0	0	20
4	0	0	20	0	20	20		60	1	349.2136658		0.016666667	60	332.5304157
5	180	0.05	272.3734613	0	20	272.3734613		120	2	444.5048779		0.033333333	120	426.044443
6	360	0.1	446.0863071	0	20	446.0863071		180	3	502.289303		0.05	180	487.12866
7	540	0.15	559.3822009	0	20	559.3822009		240	4	543.8873093		0.066666667	240	532.516808
8	720	0.2	635.5028894	0	20	635.5028894		300	5	576.4104306		0.083333333	300	568.4577615
9	900	0.25	688.587277	0	20	688.587277		360	6	603.1176476		0.1	360	598.0496271
10	1080	0.3	727.2496982	0	20	727.2496982		420	7	625.7768252		0.116666667	420	623.0742538
11	1260	0.35	756.750528	0	20	756.750528		480	8	645.455108		0.133333333	480	644.6569038
12	1440	0.4	780.3129789	0	20	780.3129789		540	9	662.8463867		0.15	540	663.5558175
13	1620	0.45	799.9220142	0	20	799.9220142		600	10	678.4273315		0.166666667	600	680.3069656
14	1800	0.5	816.8091283	0	20	816.8091283		660	11	692.5395523		0.183333333	660	695.303428
15	1980	0.55	831.7465794	785.5591283	785.559128	785.5591283		720	12	705.4362483		0.2	720	708.8420627
16	2160	0.6	845.2260326	754.3091283	754.309128	754.3091283		780	13	717.3103082		0.216666667	780	721.1524826
17	2340	0.65	857.5670792	723.0591283	723.059128	723.0591283		840	14	728.312063		0.233333333	840	732.4158538
18	2520	0.7	868.9832109	691.8091283	691.809128	691.8091283		900	15	738.5609528		0.25	900	742.7775408
19	2700	0.75	879.621976	660.5591283	660.559128	660.5591283		960	16	748.1534501		0.266666667	960	752.3558747
20	2880	0.8	889.5894651	629.3091283	629.309128	629.3091283		1020	17	757.1685957		0.283333333	1020	761.248392
21	3060	0.85	898.9652806	598.0591283	598.059128	598.0591283		1080	18	765.6719608		0.3	1080	769.5363737
22	3240	0.9	907.8117269	566.8091283	566.809128	566.8091283		1140	19	773.7185436		0.316666667	1140	777.2882137
23	3420	0.95	916.1794832	535.5591283	535.559128	535.5591283		1200	20	781.3549272		0.333333333	1200	784.5619596
24	3600	1	924.111138	504.3091283	504.309128	504.3091283		1260	21	788.6209131		0.35	1260	791.4072598
25	3780	1.05	931.6434163	473.0591283	473.059128	473.0591283		1320	22	795.5507769		0.366666667	1320	797.8668765
26	3960	1.1	938.808609	441.8091283	441.809128	441.8091283		1380	23	802.1742463		0.383333333	1380	803.9778734
27	4140	1.15	945.635511	410.5591283	410.559128	410.5591283		1440	24	808.5172716		0.4	1440	809.7725613
28	4320	1.2	952.1500561	379.3091283	379.309128	379.3091283		1500	25	814.6026398		0.416666667	1500	815.2792562
29	4500	1.25	958.3757619	348.0591283	348.059128	348.0591283		1560	26	820.4504687		0.433333333	1560	820.5228938
30	4680	1.3	964.3340553	316.8091283	316.809128	316.8091283		1620	27	826.0786082		0.45	1620	825.5255316
31	4860	1.35	970.0445211	285.5591283	285.559128	285.5591283		1680	28	831.5029687		0.466666667	1680	830.3067617
32	5040	1.4	975.5250987	254.3091283	254.309128	254.3091283		1740	29	836.7377928		0.483333333	1740	834.8840538
33	5220	1.45	980.7922449	223.0591283	223.059128	223.0591283		1800	30	841.7958797		0.5	1800	839.2730404
34	5400	1.5	985.8610722	191.8091283	191.809128	191.8091283		1860	31	846.6887747		0.516666667	1860	843.4877575
35	5580	1.55	990.7454682	160.5591283	160.559128	160.5591283		1920	32	851.4269275		0.533333333	1920	847.5408468
36	5760	1.6	995.4582019	129.3091283	129.309128	129.3091283		1980	33	856.0198265		0.55	1980	851.4437275
37	5940	1.65	1000.011018	98.05912828	98.0591282	98.05912828		2040	34	860.4761132		0.566666667	2040	855.2067436
38	6120	1.7	1004.414724	66.80912828	66.8091282	66.80912828		2100	35	864.8036804		0.583333333	2100	858.8392889
39	6300	1.75	1008.679266	35.55912828	35.5591282	35.55912828		2160	36	869.0097558		0.6	2160	862.349916
40	6480	1.8	1012.8138	4.309128278	20	20		2220	37	873.100975		0.616666667	2220	865.746429
41	6660	1.85	1016.826757	-26.94087172	20	20		2280	38	877.0834446		0.633333333	2280	869.0359653
42	6840	1.9	1020.725902	-58.19087172	20	20		2340	39	880.9627965		0.65	2340	872.2250658

Figure 2: "Sample Fire Curves" sheet. Time-temperature columns boxed in red

“Variable Diagram” Sheet

The variable diagram sheet is a visual representation to help define the meaning of the various distance variables in both the heat transfer and structural analysis commands. Figure 3 shows the variable diagram.

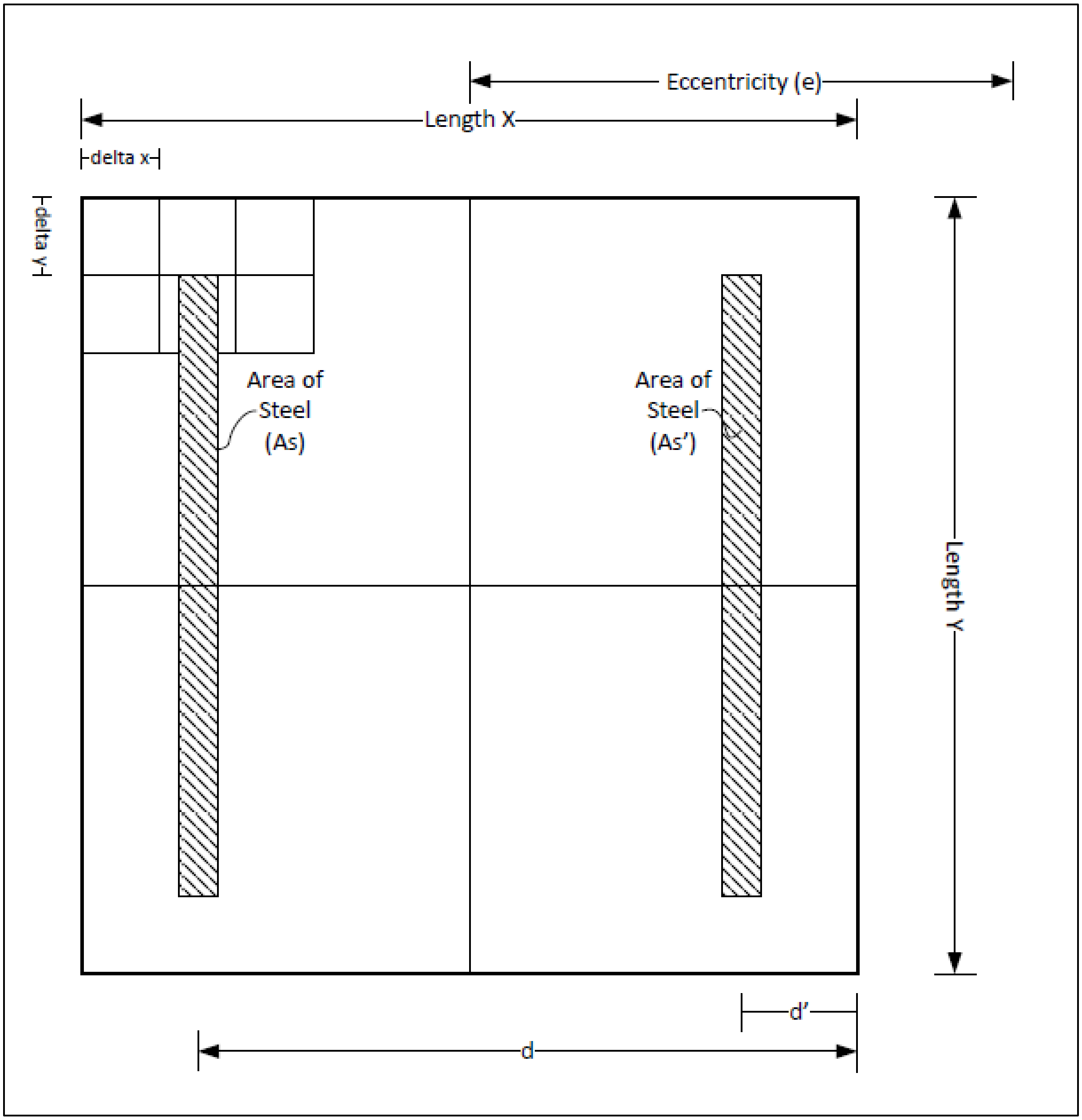


Figure 3: Variable Diagram

Command Processes

There are three command processes which run the various analyses: heat transfer, structural, post-processing. Each of these commands is run from the three tabs in the "Calculations" sheet. For each of the message boxes that appear, take care to spell the desired command exactly as it appears in the message box. Only the correct spelling will make the program run properly. Make sure the units are exact to those specified in the message box.

Heat Transfer Analysis

The following process is how to perform a heat transfer analysis using the spreadsheet.

- Enter time-temperature curve in Columns A and B in the "Calculations" sheet
    - Time curve needs to have uniform time steps throughout
- Click the "Heat Transfer" command button
- The program will ask "Have you entered a time curve with uniform values (seconds)."
    - Click yes to proceed or no to exit
- The program will ask "Have you entered a temperature curve (Celsius)."
    - Click yes to proceed or no to exit
- Next, type in desired aggregate

In the following analysis you will be asked to enter the cross-sectional lengths in both the x and y direction as well as delta x and deltay. Please choose a delta x and deltay which makes LengthX/deltax and LengthY/deltay equal to a whole number. This is necessary in order to discretize the cross section into a whole number of *Excel* cells. Refer to the variable diagram for a visual of the column cross-section variables.

- Enter delta x (meters)
    - Delta x is the length in the x-direction of the control volume
    - This is "delta x" on the variable diagram
- Enter delta y (meters)
    - Delta y is the length in the y-direction of the control volume
    - This is "delta y" on the variable diagram
- Enter convective heat transfer coefficient

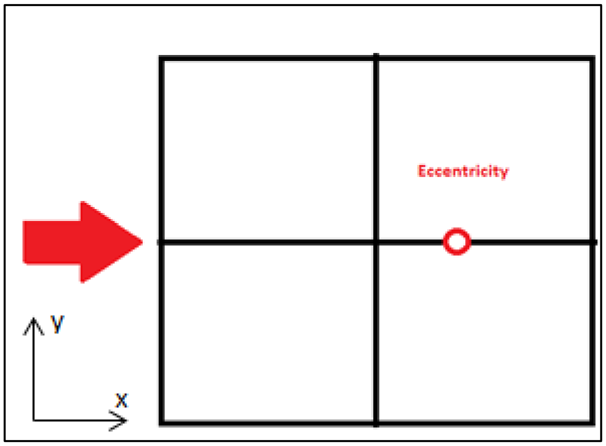
  - The recommended value is 25 W/m^2C.
- Enter the gas emissivity
  - The recommended value is 0.5 W/m^2C

At this point the spreadsheet will interpolate the time-temperature curve to in order to get a time step below the maximum allowable.

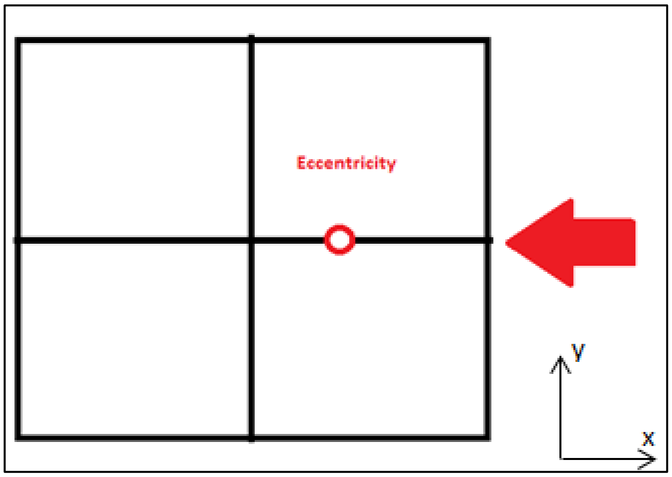
- Enter the column y-axis length (meters)
  - This is “Length Y” on the variable diagram
- Enter the column x-axis length (meters)
  - This is “Length X” on the variable diagram

At this point the spreadsheet will populate the nodes of the initial temperature step and also calculate the thermal capacity and thermal conductivity from the initial temperature of the nodes.

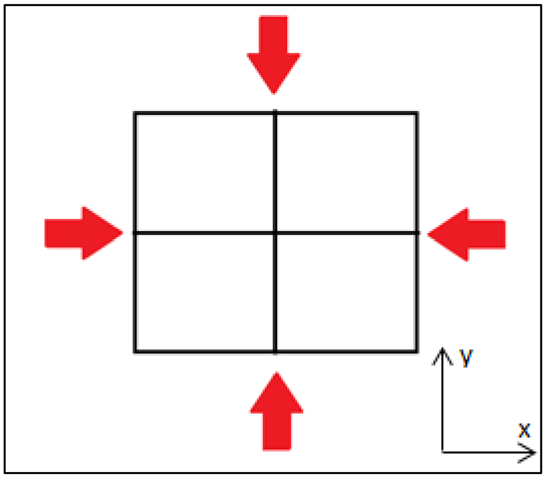
- Next, enter the desired fire exposure on the column (Fire exposure is represented with red arrow)
  - OneLeft: the fire is on the left side



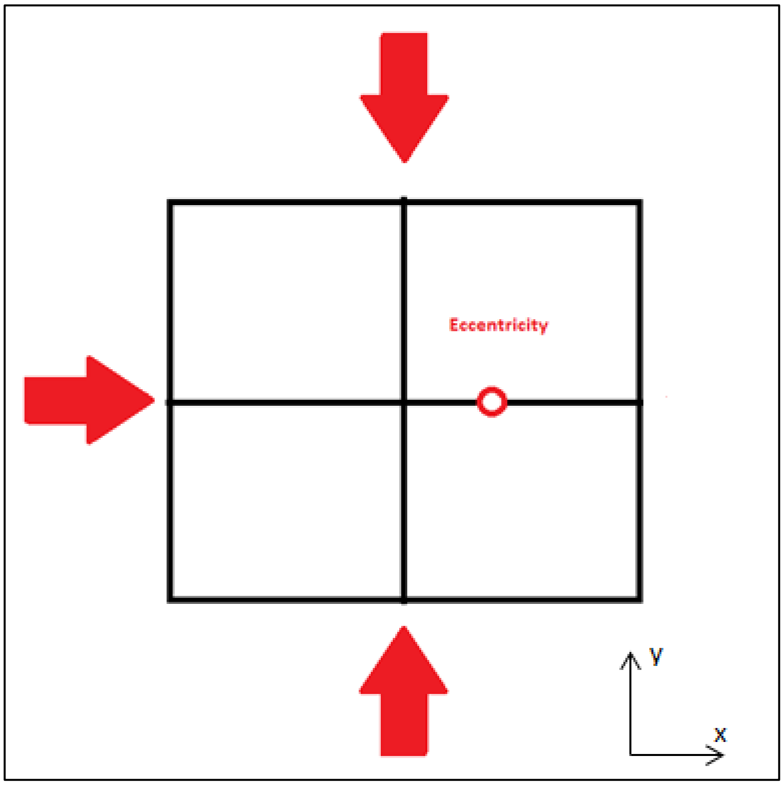
- OneRight: the fire is on the right side



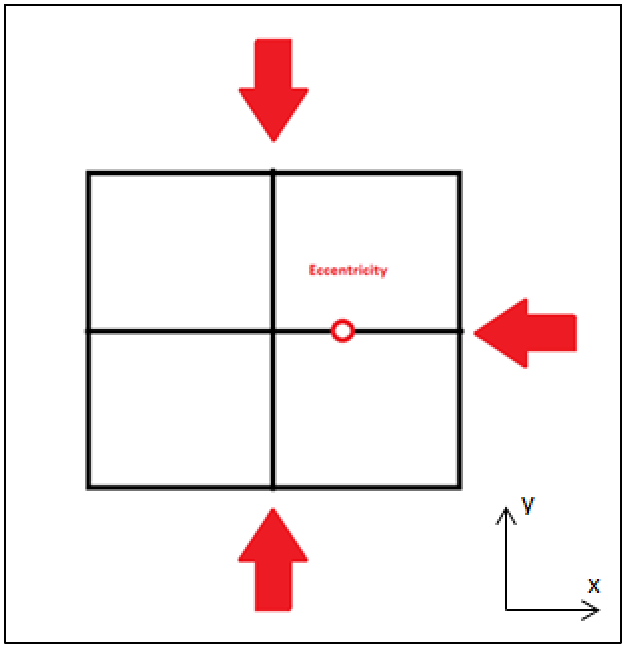
- Four: the fire is on all four sides



- ThreeRightAmb: the fire is on all the sides except the right



- ThreeLeftAmb: the fire is on all the sides except the left



- In the next box, click yes to start the heat transfer calculations or no to exit

Structural Analysis

In order to use the structural analysis portion of the spreadsheet, the heat transfer analysis has to be completed. Note that an error message may occur during the structural analysis—this is due either to the result that no 500 degree isotherms remain or too small of a cross-section remains to complete the analysis. If an error message occurs, click end in the error box to view results. The following process is how to perform a structural analysis using the spreadsheet.

- Enter the depth of concrete cover, d' (meters)
- Enter the area of the left side steel, A_s (meters^2)
- Enter the area of the right side steel, A_s (meters^2)
- Enter the yield strength of the reinforcing steel (Pa) at 20°C
- Enter the modulus of elasticity of steel (Pa) at 20°C
- Enter the compressive strength of concrete, f'_c (Pa)
- Enter the initial eccentricity of the load from the plastic centroid on the column (meters)

- Enter the equivalent effective length factor, k, for the column end connection
- Enter the unbraced length of the column (meters)
- Enter the axial dead load (Newtons)
- Enter the axial live load (Newtons)
- Click yes to start the structural analysis calculation or no to exit

At this point the calculations to determine the strength of the column at each time step will commence. The 18 columns within the spreadsheet from the edge of the thermal conductivity nodes are used to list the variables used for each time-step calculation.

Post-Processing

In order to effectively use the post-processing tabs, either the heat transfer analysis or both the heat transfer and structural analyses have to be completed. The following steps are the process to use the post-processing command.

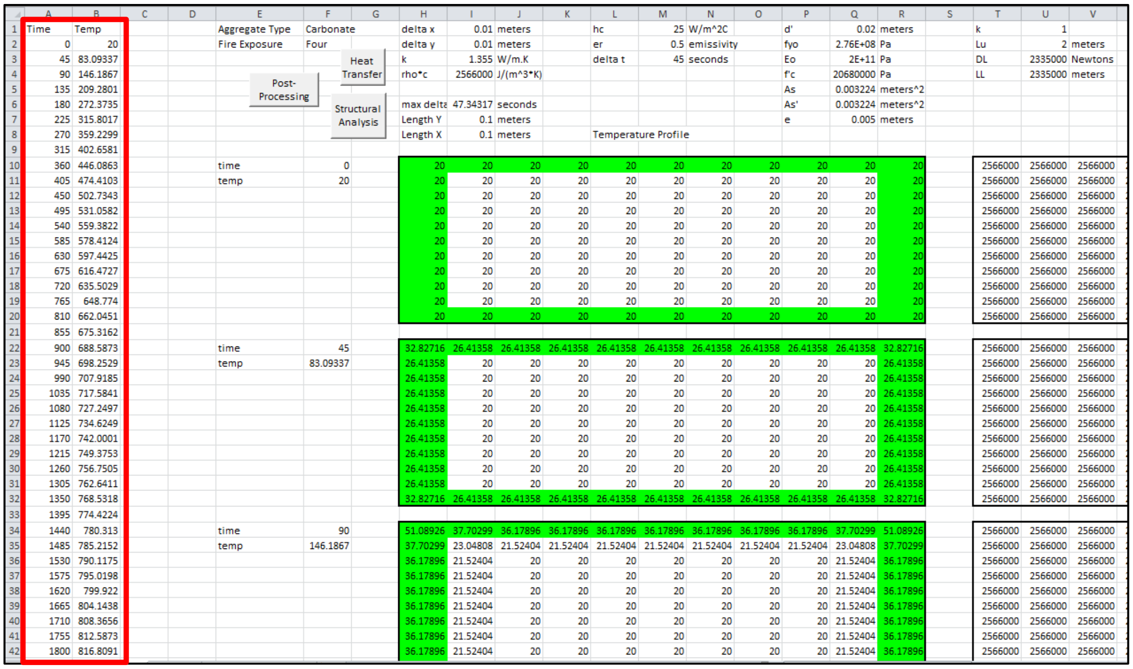
- In the first message box, click yes to select nodal temperatures on the results page or no to skip the heat transfer post-processing.
- If you selected yes, the next box will be a prompt to enter the location of the desired node. This is done by two message boxes; the first box ask for the number of cells in the x direction from the top left cell including the top left cell.
- The next prompt asks for the number of cells in the y direction from the top left cell including the top left cell
- The next message box will ask if the user would like to select another node. Click yes to select another node, or no to exit
    - The user can select up to 9 nodes

- Once the user either moves on from the heat transfer post-processing or had originally selected not to have heat transfer in the post-processing, the program will compile all the pertinent structural analysis results

The results can now be view on the “Results” sheet. The user is free to make graphs and charts based on the tabulated data; however, it is suggested to copy and paste the results into another sheet. This is because if the user decides to run another analysis or post-processing session, the “Results” sheet will be cleared. Copying the desired results to another sheet will ensure that they are not cleared.

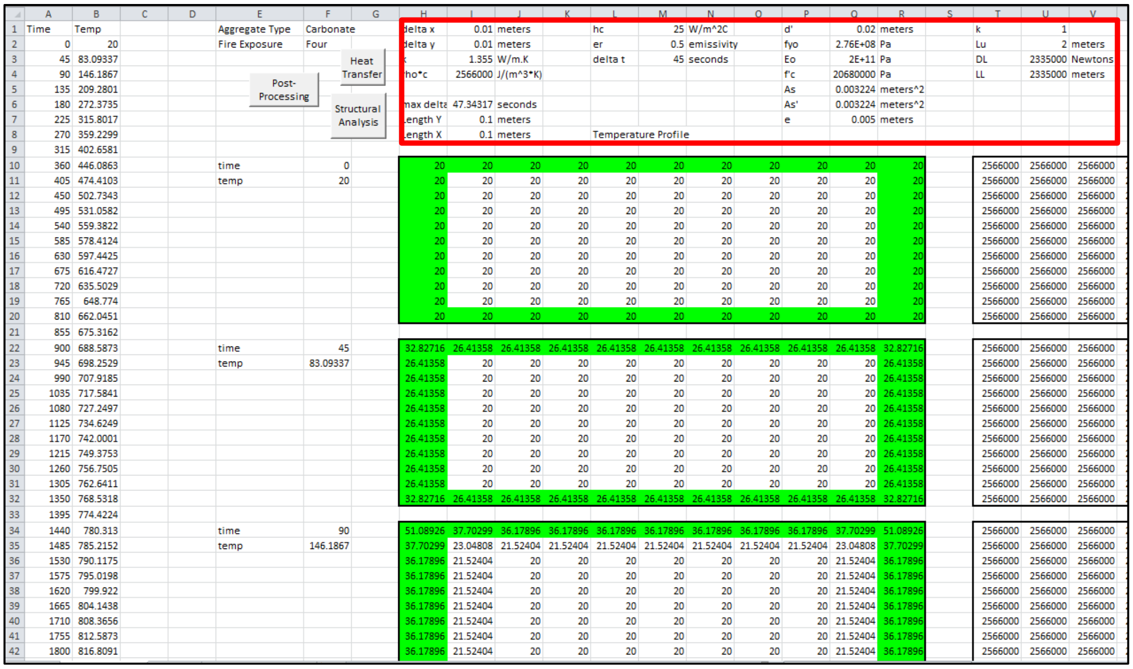
Completed Analysis Example

The following section shows several screen shots of a completed analysis. The captions of Figures 4 through 9 describe the red highlighted areas.



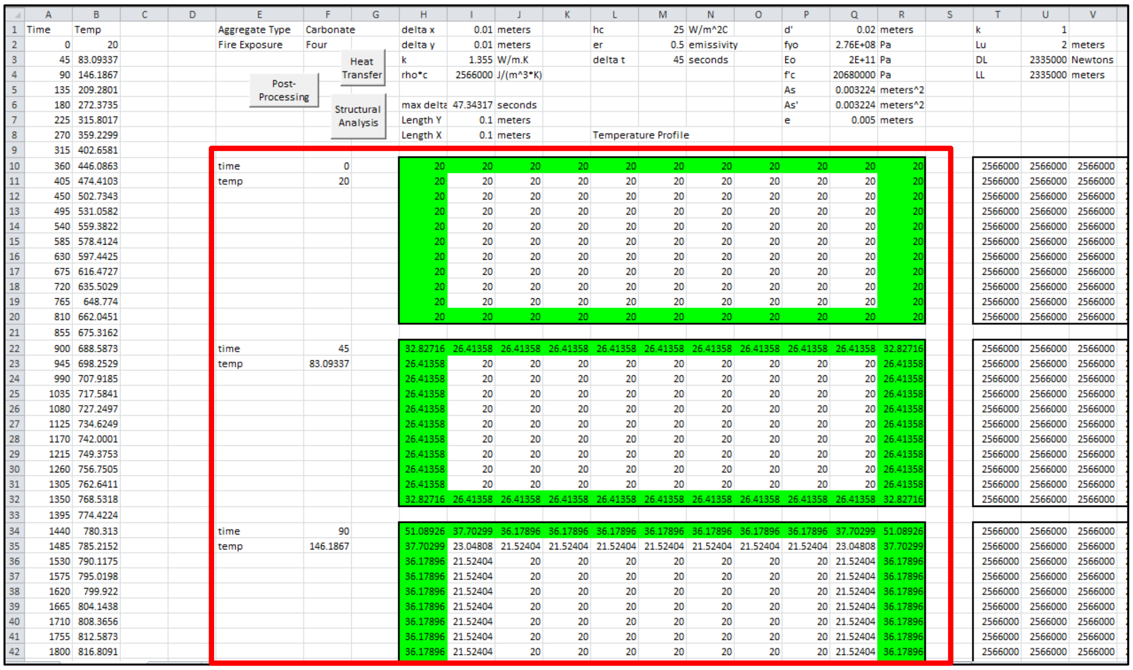
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Time	Temp			Aggregate Type	Carbonate		delta x	0.01	meters		hc	25	W/m^2C		d'	0.02	meters		k	1	
2	0	20			Fire Exposure	Four		delta y	0.01	meters		er	0.5	emissivity		fyo	2.76E+08	Pa		Lu	2	meters
3	45	83.09337					Heat Transfer	k	1.355	W/m.K		delta t	45	seconds		Eo	2E+11	Pa		DL	2335000	Newtons
4	90	146.1867						rho*c	2566000	J/(m^3*K)						f'c	20680000	Pa		LL	2335000	meters
5	135	209.2801				Post-Processing										As	0.003224	meters^2				
6	180	272.3735						max delta	47.34317	seconds						As'	0.003224	meters^2				
7	225	315.8017					Structural Analysis	Length Y	0.1	meters						e	0.005	meters				
8	270	359.2299						Length X	0.1	meters		Temperature Profile										
9	315	402.6581																				
10	360	446.0863			time	0		20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
11	405	474.4103			temp	20		20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
12	450	502.7343						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
13	495	531.0582						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
14	540	559.3822						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
15	585	578.4124						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
16	630	597.4425						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
17	675	616.4727						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
18	720	635.5029						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
19	765	648.774						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
20	810	662.0451						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
21	855	675.3162																				
22	900	688.5873			time	45		32.82716	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	32.82716		2566000	2566000	2566000
23	945	698.2529			temp	83.09337		26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
24	990	707.9185						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
25	1035	717.5841						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
26	1080	727.2497						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
27	1125	734.6249						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
28	1170	742.0001						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
29	1215	749.3753						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
30	1260	756.7505						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
31	1305	762.6411						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
32	1350	768.5318						32.82716	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	32.82716		2566000	2566000	2566000
33	1395	774.4224																				
34	1440	780.313			time	90		51.08926	37.70299	36.17896	36.17896	36.17896	36.17896	36.17896	36.17896	36.17896	37.70299	51.08926		2566000	2566000	2566000
35	1485	785.2152			temp	146.1867		37.70299	23.04808	21.52404	21.52404	21.52404	21.52404	21.52404	21.52404	21.52404	23.04808	37.70299		2566000	2566000	2566000
36	1530	790.1175						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
37	1575	795.0198						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
38	1620	799.922						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
39	1665	804.1438						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
40	1710	808.3656						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
41	1755	812.5873						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
42	1800	816.8091						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000

Figure 4: User Input Time-Temperature Curve



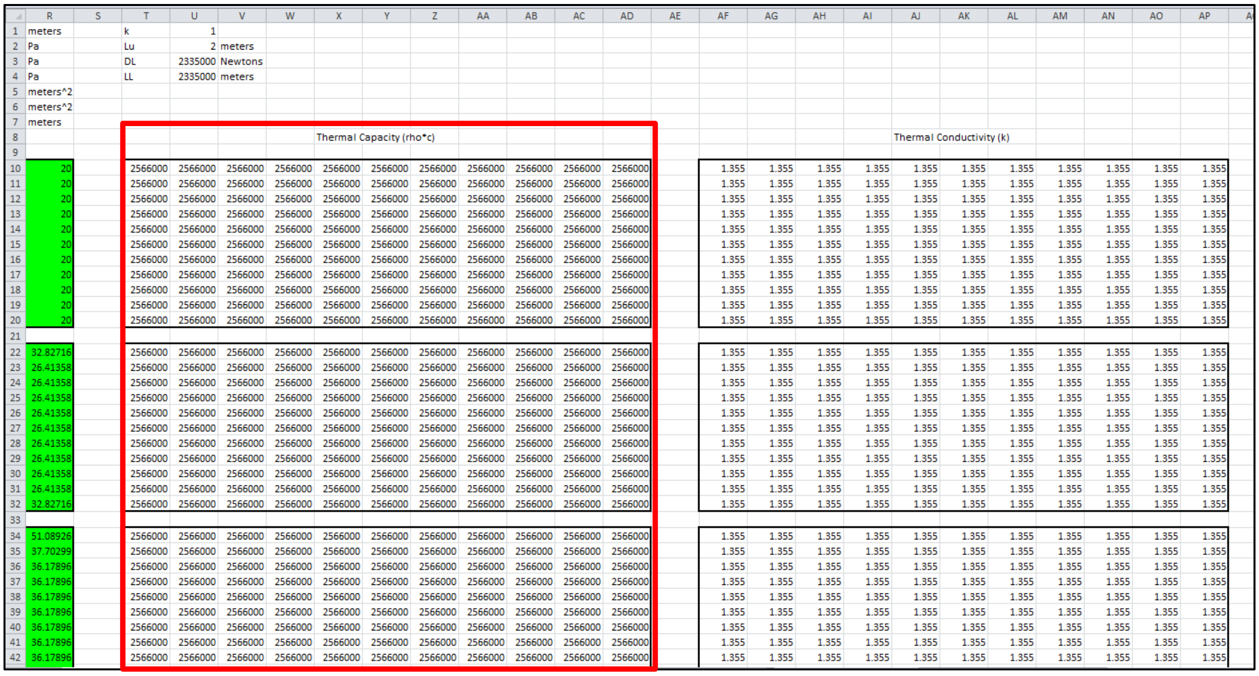
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Time	Temp			Aggregate Type	Carbonate		delta x	0.01	meters		hc	25	W/m^2C		d'	0.02	meters		k	1	
2	0	20			Fire Exposure	Four		delta y	0.01	meters		er	0.5	emissivity		fyo	2.76E+08	Pa		Lu	2	meters
3	45	83.09337						k	1.355	W/m.K		delta t	45	seconds		Eo	2E+11	Pa		DL	2335000	Newtons
4	90	146.1867						rho*c	2566000	J/(m^3*K)						f'c	20680000	Pa		LL	2335000	meters
5	135	209.2801														As	0.003224	meters^2				
6	180	272.3735						max delta	47.34317	seconds						As'	0.003224	meters^2				
7	225	315.8017						Length Y	0.1	meters						e	0.005	meters				
8	270	359.2299						Length X	0.1	meters		Temperature Profile										
9	315	402.6581																				
10	360	446.0863			time	0		20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
11	405	474.4103			temp	20		20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
12	450	502.7343						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
13	495	531.0582						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
14	540	559.3822						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
15	585	578.4124						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
16	630	597.4425						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
17	675	616.4727						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
18	720	635.5029						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
19	765	648.774						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
20	810	662.0451						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
21	855	675.3162																				
22	900	688.5873			time	45		32.82716	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	32.82716		2566000	2566000	2566000
23	945	698.2529			temp	83.09337		26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
24	990	707.9185						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
25	1035	717.5841						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
26	1080	727.2497						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
27	1125	734.6249						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
28	1170	742.0001						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
29	1215	749.3753						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
30	1260	756.7505						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
31	1305	762.6411						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
32	1350	768.5318						32.82716	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	32.82716		2566000	2566000	2566000
33	1395	774.4224																				
34	1440	780.313			time	90		51.08926	37.70299	36.17896	36.17896	36.17896	36.17896	36.17896	36.17896	36.17896	37.70299	51.08926		2566000	2566000	2566000
35	1485	785.2152			temp	146.1867		37.70299	23.04808	21.52404	21.52404	21.52404	21.52404	21.52404	21.52404	21.52404	23.04808	37.70299		2566000	2566000	2566000
36	1530	790.1175						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
37	1575	795.0198						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
38	1620	799.922						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
39	1665	804.1438						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
40	1710	808.3656						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
41	1755	812.5873						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
42	1800	816.8091						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000

Figure 5: Column and Heat Transfer Characteristics and Properties



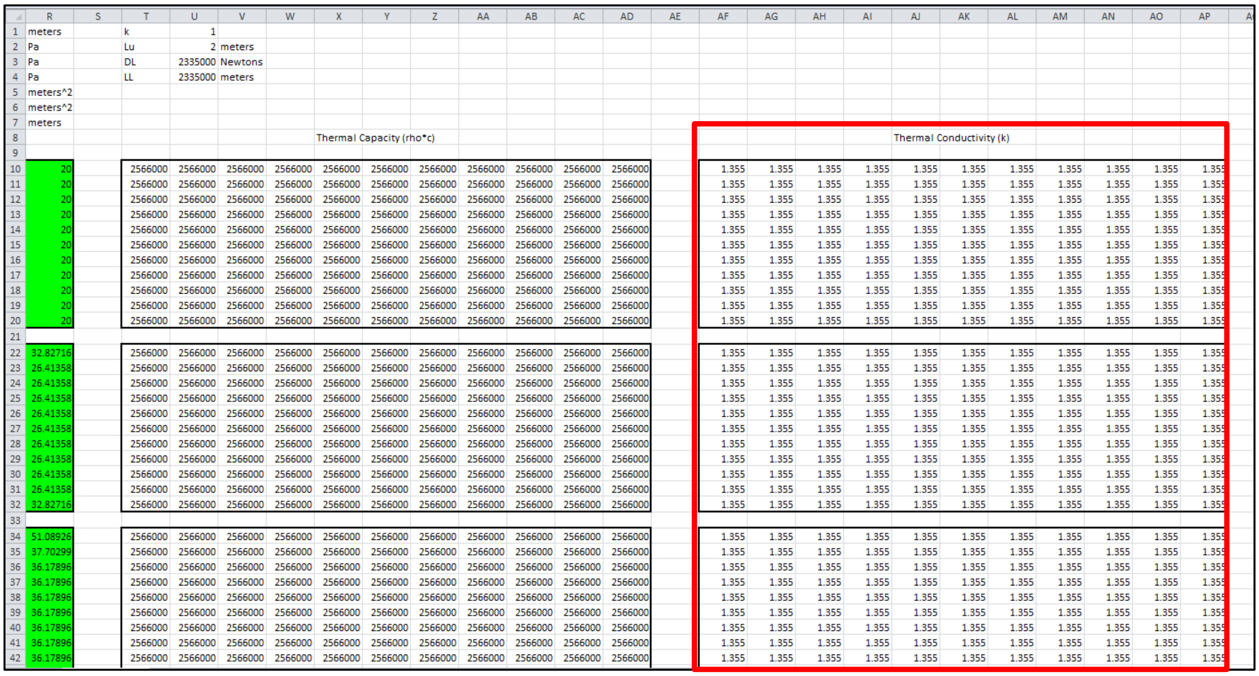
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1	Time	Temp			Aggregate Type	Carbonate		delta x	0.01	meters		hc	25	W/m^2C		d'	0.02	meters		k	1	
2	0	20			Fire Exposure	Four		delta y	0.01	meters		er	0.5	emissivity		fyo	2.76E+08	Pa		Lu	2	meters
3	45	83.09337						k	1.355	W/m.K		delta t	45	seconds		Eo	2E+11	Pa		DL	2335000	Newtons
4	90	146.1867						rho*c	2566000	J/(m^3*K)						f'c	20680000	Pa		LL	2335000	meters
5	135	209.2801														As	0.003224	meters^2				
6	180	272.3735						max delta	47.34317	seconds						As'	0.003224	meters^2				
7	225	315.8017						Length Y	0.1	meters						e	0.005	meters				
8	270	359.2299						Length X	0.1	meters		Temperature Profile										
9	315	402.6581																				
10	360	446.0863			time	0		20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
11	405	474.4103			temp	20		20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
12	450	502.7343						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
13	495	531.0582						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
14	540	559.3822						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
15	585	578.4124						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
16	630	597.4425						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
17	675	616.4727						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
18	720	635.5029						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
19	765	648.774						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
20	810	662.0451						20	20	20	20	20	20	20	20	20	20	20		2566000	2566000	2566000
21	855	675.3162																				
22	900	688.5873			time	45		32.82716	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	32.82716		2566000	2566000	2566000
23	945	698.2529			temp	83.09337		26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
24	990	707.9185						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
25	1035	717.5841						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
26	1080	727.2497						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
27	1125	734.6249						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
28	1170	742.0001						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
29	1215	749.3753						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
30	1260	756.7505						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
31	1305	762.6411						26.41358	20	20	20	20	20	20	20	20	20	26.41358		2566000	2566000	2566000
32	1350	768.5318						32.82716	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	26.41358	32.82716		2566000	2566000	2566000
33	1395	774.4224																				
34	1440	780.313			time	90		51.08926	37.70299	36.17896	36.17896	36.17896	36.17896	36.17896	36.17896	36.17896	37.70299	51.08926		2566000	2566000	2566000
35	1485	785.2152			temp	146.1867		37.70299	23.04808	21.52404	21.52404	21.52404	21.52404	21.52404	21.52404	21.52404	23.04808	37.70299		2566000	2566000	2566000
36	1530	790.1175						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
37	1575	795.0198						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
38	1620	799.922						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
39	1665	804.1438						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
40	1710	808.3656						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
41	1755	812.5873						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000
42	1800	816.8091						36.17896	21.52404	20	20	20	20	20	20	20	21.52404	36.17896		2566000	2566000	2566000

Figure 6: Temperature Profiles of the First Three Time Steps



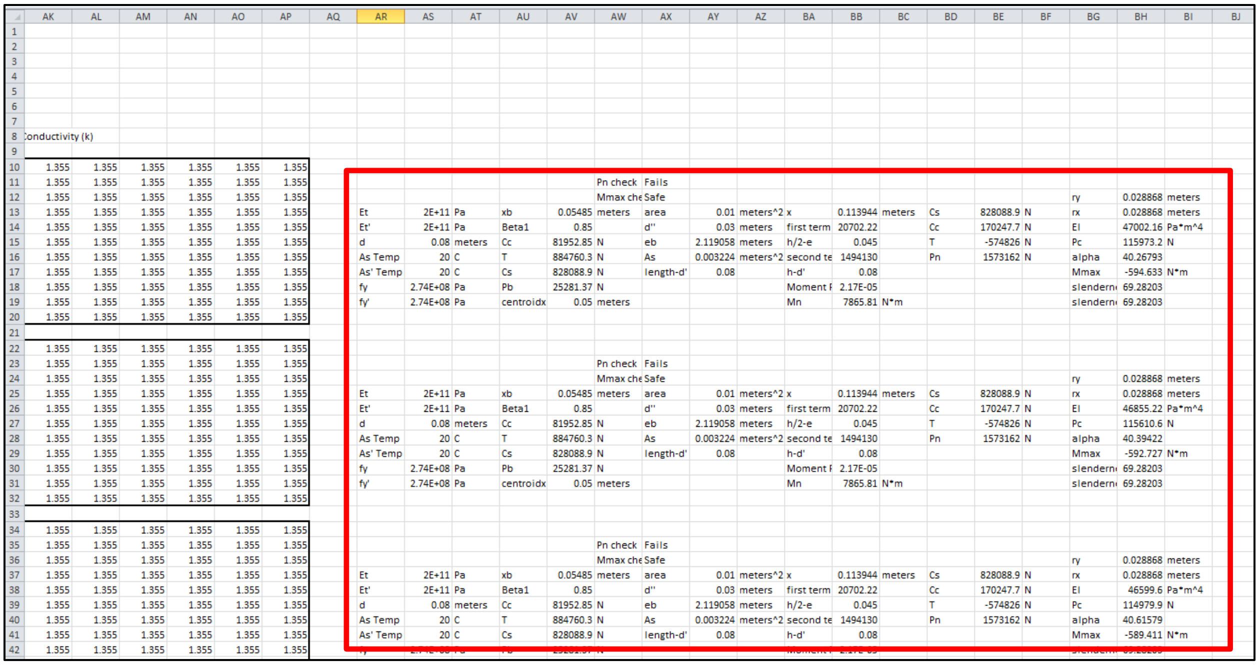
	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1	meters		k	1																					
2	Pa		Lu	2	meters																				
3	Pa		DL	2335000	Newtons																				
4	Pa		LL	2335000	meters																				
5	meters^2																								
6	meters^2																								
7	meters																								
8							Thermal Capacity (rho*c)												Thermal Conductivity (k)						
9																									
10	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
11	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
12	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
13	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
14	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
15	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
16	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
17	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
18	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
19	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
20	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
21																									
22	32.82716		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
23	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
24	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
25	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
26	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
27	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
28	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
29	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
30	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
31	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
32	32.82716		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
33																									
34	51.08926		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
35	37.70299		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
36	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
37	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
38	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
39	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
40	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
41	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
42	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355

Figure 7: Thermal Capacity for Each Time Step



	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP
1	meters		k	1																					
2	Pa		Lu	2	meters																				
3	Pa		DL	2335000	Newtons																				
4	Pa		LL	2335000	meters																				
5	meters^2																								
6	meters^2																								
7	meters																								
8							Thermal Capacity (rho*c)												Thermal Conductivity (k)						
9																									
10	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
11	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
12	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
13	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
14	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
15	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
16	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
17	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
18	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
19	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
20	20		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
21																									
22	32.82716		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
23	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
24	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
25	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
26	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
27	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
28	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
29	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
30	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
31	26.41358		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
32	32.82716		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
33																									
34	51.08926		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
35	37.70299		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
36	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
37	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
38	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
39	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
40	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
41	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355
42	36.17896		2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000	2566000		1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355	1.355

Figure 8: Thermal Conductivity for Each Time Step



	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ
1																										
2																										
3																										
4																										
5																										
6																										
7																										
8	:onductivity (k)																									
9																										
10	1.355	1.355	1.355	1.355	1.355	1.355																				
11	1.355	1.355	1.355	1.355	1.355	1.355							Pn check	Fails												
12	1.355	1.355	1.355	1.355	1.355	1.355							Mmax che	Safe									ry	0.028868	meters	
13	1.355	1.355	1.355	1.355	1.355	1.355		Et	2E+11	Pa	xb	0.05485	meters	area	0.01	meters^2	x	0.113944	meters	Cs	828088.9	N	rx	0.028868	meters	
14	1.355	1.355	1.355	1.355	1.355	1.355		Et'	2E+11	Pa	Beta1	0.85		d''	0.03	meters	first term	20702.22		Cc	170247.7	N	EI	47002.16	Pa*m^4	
15	1.355	1.355	1.355	1.355	1.355	1.355		d	0.08	meters	Cc	81952.85	N	eb	2.119058	meters	h/2-e	0.045		T	-574826	N	Pc	115973.2	N	
16	1.355	1.355	1.355	1.355	1.355	1.355		As Temp	20	C	T	884760.3	N	As	0.003224	meters^2	second te	1494130		Pn	1573162	N	alpha	40.26793		
17	1.355	1.355	1.355	1.355	1.355	1.355		As' Temp	20	C	Cs	828088.9	N	length-d'	0.08		h-d'	0.08					Mmax	-594.633	N*m	
18	1.355	1.355	1.355	1.355	1.355	1.355		fy	2.74E+08	Pa	Pb	25281.37	N				Moment F	2.17E-05					slendern	69.28203		
19	1.355	1.355	1.355	1.355	1.355	1.355		fy'	2.74E+08	Pa	centroidx	0.05	meters				Mn	7865.81	N*m				slendern	69.28203		
20	1.355	1.355	1.355	1.355	1.355	1.355																				
21																										
22	1.355	1.355	1.355	1.355	1.355	1.355																				
23	1.355	1.355	1.355	1.355	1.355	1.355							Pn check	Fails												
24	1.355	1.355	1.355	1.355	1.355	1.355							Mmax che	Safe									ry	0.028868	meters	
25	1.355	1.355	1.355	1.355	1.355	1.355		Et	2E+11	Pa	xb	0.05485	meters	area	0.01	meters^2	x	0.113944	meters	Cs	828088.9	N	rx	0.028868	meters	
26	1.355	1.355	1.355	1.355	1.355	1.355		Et'	2E+11	Pa	Beta1	0.85		d''	0.03	meters	first term	20702.22		Cc	170247.7	N	EI	46855.22	Pa*m^4	
27	1.355	1.355	1.355	1.355	1.355	1.355		d	0.08	meters	Cc	81952.85	N	eb	2.119058	meters	h/2-e	0.045		T	-574826	N	Pc	115610.6	N	
28	1.355	1.355	1.355	1.355	1.355	1.355		As Temp	20	C	T	884760.3	N	As	0.003224	meters^2	second te	1494130		Pn	1573162	N	alpha	40.39422		
29	1.355	1.355	1.355	1.355	1.355	1.355		As' Temp	20	C	Cs	828088.9	N	length-d'	0.08		h-d'	0.08					Mmax	-592.727	N*m	
30	1.355	1.355	1.355	1.355	1.355	1.355		fy	2.74E+08	Pa	Pb	25281.37	N				Moment F	2.17E-05					slendern	69.28203		
31	1.355	1.355	1.355	1.355	1.355	1.355		fy'	2.74E+08	Pa	centroidx	0.05	meters				Mn	7865.81	N*m				slendern	69.28203		
32	1.355	1.355	1.355	1.355	1.355	1.355																				
33																										
34	1.355	1.355	1.355	1.355	1.355	1.355																				
35	1.355	1.355	1.355	1.355	1.355	1.355							Pn check	Fails												
36	1.355	1.355	1.355	1.355	1.355	1.355							Mmax che	Safe									ry	0.028868	meters	
37	1.355	1.355	1.355	1.355	1.355	1.355		Et	2E+11	Pa	xb	0.05485	meters	area	0.01	meters^2	x	0.113944	meters	Cs	828088.9	N	rx	0.028868	meters	
38	1.355	1.355	1.355	1.355	1.355	1.355		Et'	2E+11	Pa	Beta1	0.85		d''	0.03	meters	first term	20702.22		Cc	170247.7	N	EI	46599.6	Pa*m^4	
39	1.355	1.355	1.355	1.355	1.355	1.355		d	0.08	meters	Cc	81952.85	N	eb	2.119058	meters	h/2-e	0.045		T	-574826	N	Pc	114979.9	N	
40	1.355	1.355	1.355	1.355	1.355	1.355		As Temp	20	C	T	884760.3	N	As	0.003224	meters^2	second te	1494130		Pn	1573162	N	alpha	40.61579		
41	1.355	1.355	1.355	1.355	1.355	1.355		As' Temp	20	C	Cs	828088.9	N	length-d'	0.08		h-d'	0.08					Mmax	-589.411	N*m	
42	1.355	1.355	1.355	1.355	1.355	1.355		fy	2.74E+08	Pa	Pb	25281.37	N				Moment F	2.17E-05					slendern	69.28203		

Figure 9: Structural Calculations

Troubleshooting

The following section contains common errors and ways to try and correct the errors.

- Error occurs during interpolation of time-temperature curve
  - Check to see if the time steps are equal throughout entire time curve
- Error occurs during the construction of the initial time-step
  - Make sure that the number of cells is a whole number
  - $\frac{LengthX}{deltax} = whole\ number$
  - $\frac{LengthY}{deltay} = whole\ number$
- Spreadsheet does not work
  - Enable macros
- Instability error in the heat transfer calculations (i.e. negative numbers or unintended decreases in temperature occur)

  - Manually shorten the time step
  - Increase finite element grid size (i.e. 20x20 → 30x30)
    - Make sure number of cells is a whole number

- Heat transfer does not run
  - Check for proper spelling on all input

Helpful Tips

- Make sure you use the correct units. The variables and the spreadsheet calculations are all based off specific units.
- Consult with the variable diagram to determine which distances the various variables represent
- In order to get used to the spreadsheet, select a textbook example for a column at ambient conditions and try to run the spreadsheet with only time=0 and temperature = 20°C
  - This will calculate the ambient strength of the column
  - Then try adding in a temperature curve
- Start with a small grid size (fewer cells) to get an approximate answer; then increase the grid size to run a sensitivity analysis on the grid size.
  - Remember, larger grid size take exponentially longer computing the answer

Appendix C-Model Code

Each of the following modules of code listed below combine in *Microsoft Excel* to perform the necessary calculations for determining the fire performance of a reinforced-concrete column. Each of the modulus has name which is listed in the module code and also in the section heading

Module 1: MaxTimeStep

```
Sub MaxTimeStep()

    'This module is used to calculate the maximum time step permissible when using the control volume
approach to two-dimensional heat transfer'

    'User input of control volume size x direction length'
    Dim deltax As String
    Cells(1, 8) = "delta x"
    Cells(1, 10) = "meters"
    deltax = InputBox("Please enter deltax (meters)", "Delta X")
    If deltax = "" Then Exit Sub
    Cells(1, 9) = deltax

    'User input of control colume size y direction length'
    Dim deltay As String
    Cells(2, 8) = "delta y"
    Cells(2, 10) = "meters"
    deltay = InputBox("Please enter deltay (meters)", "Delta Y")
    If deltay = "" Then Exit Sub
    Cells(2, 9) = deltay

    'Code which calculates the thermal conductivity based on the choice of concrete aggregate and initial
temperature'
    Dim k As String
    Cells(3, 8) = "k"
    Cells(3, 10) = "W/m.K"

            d = Cells(2, 2)

            If Cells(1, 6) = "Siliceous" Then
                If d >= 0 And d <= 800 Then k = (-0.000625 * d + 1.5)
                If d > 800 Then k = 1
            End If

            If Cells(1, 6) = "Carbonate" Then
                If d >= 0 And d <= 293 Then k = 1.355
                If d > 293 Then k = (-0.001241 * d + 1.7162)
            End If
    Cells(3, 9) = k

    'Code which calculates the thermal capacity based on the choice of concrete aggregate and initial temperature'
    Dim rhoc As String
    Cells(4, 8) = "rho*c"
    Cells(4, 10) = "J/(m^3*K)"

    d = Cells(2, 2)

            If Cells(1, 6) = "Siliceous" Then

                If d >= 0 And d <= 200 Then rhoc = (0.005 * d + 1.7) * 10 ^ 6
                If d > 200 And d <= 400 Then rhoc = (2.7 * 10 ^ 6)
                If d > 400 And d <= 500 Then rhoc = (0.013 * d - 2.5) * 10 ^ 6
                If d > 500 And d <= 600 Then rhoc = (-0.013 * d + 10.5) * 10 ^ 6
                If d > 600 Then rhoc = 2.7 * 10 ^ 6
            End If

           If Cells(1, 6) = "Carbonate" Then
                If d >= 0 And d <= 400 Then rhoc = 2.566 * 10 ^ 6
                If d > 400 And d <= 410 Then rhoc = (0.1765 * d - 68.034) * 10 ^ 6
                If d > 410 And d <= 445 Then rhoc = (-0.05043 * d + 25.00671) * 10 ^ 6
                If d > 445 And d <= 500 Then rhoc = (2.566 * 10 ^ 6)
                If d > 500 And d <= 635 Then rhoc = (0.01603 * d - 5.44881) * 10 ^ 6
                If d > 635 And d <= 715 Then rhoc = (0.16635 * d - 100.90225) * 10 ^ 6
                If d > 715 And d <= 785 Then rhoc = (-0.22103 * d + 176.07343) * 10 ^ 6
                If d > 785 Then rhoc = (2.566 * 10 ^ 6)
            End If

    Cells(4, 9) = rhoc



    'maximum permissible time step'
    Dim deltat As String
    Cells(6, 8) = "max delta t"
    Cells(6, 10) = "seconds"

    deltat = (1 / (2 * (k / (rhoc)))) * ((1 / (deltax) ^ 2) + (1 / (deltay) ^ 2)) ^ -1

    Cells(6, 9) = deltat

    'User input of convective heat transfer coefficient
    Dim hc As String
    Cells(1, 12) = "hc"
    Cells(1, 14) = "W/m^2C"
    hc = InputBox("Please enter hc (W/m^2C)- Recommended Value: 25", "Convective Heat Transfer
Coefficient")
    If hc = "" Then Exit Sub
    Cells(1, 13) = hc

    'User input of the emissivity of the concrete surface and combustion gases'
    Dim er As String
    Cells(2, 12) = "er"
    Cells(2, 14) = "emissivity"
    er = InputBox("Please enter er (W/m^2C)- Recommended Value: .5", "emissivity")
    If er = "" Then Exit Sub
    Cells(2, 13) = er

    End Sub
```

Module 2: TemperatureProfile

```
Sub TemperatureProfile()

    'Calculation of time step from the user inputted time curve'
    deltat = Cells(6, 9)
    Cells(3, 13) = (Cells(3, 1) - Cells(2, 1))
    Cells(3, 12) = "delta t"
    Cells(3, 14) = "seconds"

    'interpolation of time/temperature curve to get time step within maximum time step allowed'
    If Cells(3, 1) - Cells(2, 1) > deltat Then

    Do
        FinalRowTime = Cells(Rows.Count, 1).End(xlUp).Row
        FinalRowTemp = Cells(Rows.Count, 2).End(xlUp).Row

        FinalRowNewTime = Cells(Rows.Count, 3).End(xlUp).Row
        FinalRowNewTemp = Cells(Rows.Count, 4).End(xlUp).Row

        Cells(FinalRowNewTime, 3) = "Time"
        Cells(FinalRowNewTemp, 4) = "Temp"

        x = 2

        Cells(FinalRowNewTime + 1, 3) = Cells(x, 1)
        Cells(FinalRowNewTime + 3, 3) = Cells(x + 1, 1)
        Cells(FinalRowNewTime + 2, 3) = (Cells(x + 1, 1) + Cells(x, 1)) / 2

        For x = 3 To (FinalRowTime - 1)
        FinalRowNewTime = Cells(Rows.Count, 3).End(xlUp).Row
        FinalRowNewTemp = Cells(Rows.Count, 4).End(xlUp).Row

        Cells(FinalRowNewTime, 3) = Cells(x, 1)
        Cells(FinalRowNewTime + 2, 3) = Cells(x + 1, 1)
        Cells(FinalRowNewTime + 1, 3) = (Cells(x + 1, 1) + Cells(x, 1)) / 2
        Next x

        x = 2

        Cells(FinalRowNewTemp + 1, 4) = Cells(x, 2)
        Cells(FinalRowNewTemp + 3, 4) = Cells(x + 1, 2)
        Cells(FinalRowNewTemp + 2, 4) = (Cells(x + 1, 2) + Cells(x, 2)) / 2

        For x = 3 To (FinalRowTemp - 1)

        FinalRowNewTime = Cells(Rows.Count, 3).End(xlUp).Row
        FinalRowNewTemp = Cells(Rows.Count, 4).End(xlUp).Row

        Cells(FinalRowNewTemp, 4) = Cells(x, 2)
        Cells(FinalRowNewTemp + 2, 4) = Cells(x + 1, 2)
        Cells(FinalRowNewTemp + 1, 4) = (Cells(x + 1, 2) + Cells(x, 2)) / 2

        Next x

        Range("C:D").Select
        Selection.Cut
        Range("A:B").Select
        ActiveSheet.Paste

    Loop Until Cells(3, 1) - Cells(2, 1) < deltat
    End If

    Cells(3, 13) = (Cells(3, 1) - Cells(2, 1))
    Cells(3, 12) = "delta t"
    Cells(3, 14) = "seconds"

End Sub
```

Module 3: Experiment

```
Sub Experiment()

    'main heat transfer file'

    'clear of cells except for Columns A and B'
    Range(Columns(3), Columns(256)).Clear

    Cells(1, 1) = "Time"
    Cells(1, 2) = "Temp"

    'Asking the user to input time curve'
    Dim TimeCurve As VbMsgBoxResult
    TimeCurve = MsgBox("Have you entered a time curve with uniform values (seconds). Click Yes if
completed; No to exit and complete", vbYesNo, "Time Curve")
    If TimeCurve = vbNo Then Exit Sub

    'Asking the user to input temperature curve'
    Dim TempCurve As VbMsgBoxResult
    TempCurve = MsgBox("Have you entered a temperature curve (Celsius). Click Yes if completed; No to
exit and complete", vbYesNo, "Temperature Curve")
    If TempCurve = vbNo Then Exit Sub

    'User chooses a type of concrete aggregate'
    Dim concretetype As String
    concretetype = InputBox("Choose Aggregate: Type(Siliceous or Carbonate)", "Concrete Aggregate
Type")
    If concretetype = "" Then Exit Sub
    Cells(1, 5) = "Aggregate Type"
    Cells(1, 6) = concretetype

    'length/delta notifcation'
    Dim Divided As VbMsgBoxResult
    Divided = MsgBox("In the following analysis you will be asked to enter the cross section length in both
the x and y direction as well as deltax and deltay. Please choose a deltax and deltay which makes
LengthX/deltax and LengthY/deltay equal to a whole number. This is necessary in order to have a whole
number of excel cells.", vbOKOnly, "Division of Cross Section")

    Call MaxTimeStep
    If Cells(6, 9) = "" Then Exit Sub

    Call TemperatureProfile
    Call InitialTemperature
    Call ThermalCapacityInitial
    Call ThermalConductivityInitial

    Cells(10, 8).Select
    XCells = Cells(8, 9) / Cells(1, 9)

    ActiveCell.Offset(-2, ((XCells - 1) / 2)).Select
    ActiveCell = "Temperature Profile"
    ActiveCell.Offset(0, XCells + 2).Select
    ActiveCell = "Thermal Capacity (rho*c)"
    ActiveCell.Offset(0, XCells + 2).Select
```

```
   ActiveCell = "Thermal Conductivity (k)"



   'Asking the user the fire exposure on the column'
   Dim fireexposure As String
   fireexposure = InputBox("Fire Exposure: Type(OneLeft, OneRight, Four, ThreeRightAmb,
ThreeLeftAmb)")
   If fireexposure = "" Then Exit Sub
   Cells(2, 5) = "Fire Exposure"
   Cells(2, 6) = fireexposure

   Dim DoYouWant As VbMsgBoxResult
   DoYouWant = MsgBox("Do you want to start the heat transfer calculations. Click Yes to start; No to
exit", vbYesNo, "Heat Transfer Calculations")
   If DoYouWant = vbNo Then Exit Sub

   If Cells(2, 6) = "Four" Then
   Call TemperatureStepFourSided
   End If

   If Cells(2, 6) = "OneLeft" Then
   Call TemperatureStepOneSidedLeft
   End If

   If Cells(2, 6) = "OneRight" Then
   Call TemperatureStepOneSidedRight
   End If

   If Cells(2, 6) = "ThreeRightAmb" Then
   Call TemperatureStepThreeRightSided
   End If

   If Cells(2, 6) = "ThreeLeftAmb" Then
   Call TemperatureStepThreeLeftSided
   End If


   For Noise = 1 To 100
   Beep
   Next Noise


End Sub
```

Module 4: InitialTemperature

```
Sub InitialTemperature()

    Itemp = Cells(2, 2)
    deltax = Cells(1, 9)
    deltay = Cells(2, 9)

    'Asking the user to input the y direction length of the column'
    LengthY = InputBox("Please enter Y axis column length (meters)", "Y Length")
    If LengthY = "" Then Exit Sub
    Cells(7, 8) = "Length Y"
    Cells(7, 9) = LengthY
    Cells(7, 10) = "meters"

    'Asking the user to input the x direction length of the column'
    LengthX = InputBox("Please enter X axis column length (meters)", "X Length")
    If LengthX = "" Then Exit Sub
    Cells(8, 8) = "Length X"
    Cells(8, 9) = LengthX
    Cells(8, 10) = "meters"

    'calculates the number of cells needed to perform the necessary calculation'
    YCells = LengthY / deltay + 1
    XCells = LengthX / deltax + 1

    'complication of the initial temperature in each of the cells (nodes)'
    For x = 8 To (8 - 1 + XCells)
        For y = 10 To (10 - 1 + YCells)

        Cells(y, x) = Itemp

        Next y

    EndX = 8 - 1 + XCells
    EndY = 10 - 1 + YCells

    Next

    'draws a border around the cells in the heat transfer calcualtion'
    With Range(Cells(10, 8), Cells(EndY, EndX))
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeTop).Weight = xlMedium
    .Borders(xlEdgeLeft).Weight = xlMedium
    .Borders(xlEdgeRight).Weight = xlMedium
    End With
```

Module 5:ThermalCapacityInitial

```
Sub ThermalCapacityInitial()

   LengthY = Cells(7, 9)
   LengthX = Cells(8, 9)

   deltax = Cells(1, 9)
   deltay = Cells(2, 9)

   YCells = LengthY / deltay + 1
   XCells = LengthX / deltax + 1

   'calculates the thermal capacity for each node temperature for the initial temperature steps'
   For x2 = (8 - 1 + XCells + 2) To (8 + 2 * XCells)
      For y = 10 To (10 - 1 + YCells)

            d = Cells(y, x2 - XCells - 1)

            If Cells(1, 6) = "Siliceous" Then

               If d >= 0 And d <= 200 Then Cells(y, x2) = (0.005 * d + 1.7) * 10 ^ 6
               If d > 200 And d <= 400 Then Cells(y, x2) = (2.7 * 10 ^ 6)
               If d > 400 And d <= 500 Then Cells(y, x2) = (0.013 * d - 2.5) * 10 ^ 6
               If d > 500 And d <= 600 Then Cells(y, x2) = (-0.013 * d + 10.5) * 10 ^ 6
               If d > 600 Then Cells(y, x2) = 2.7 * 10 ^ 6
            End If

           If Cells(1, 6) = "Carbonate" Then
               If d >= 0 And d <= 400 Then Cells(y, x2) = 2.566 * 10 ^ 6
               If d > 400 And d <= 410 Then Cells(y, x2) = (0.1765 * d - 68.034) * 10 ^ 6
               If d > 410 And d <= 445 Then Cells(y, x2) = (-0.05043 * d + 25.00671) * 10 ^ 6
               If d > 445 And d <= 500 Then Cells(y, x2) = (2.566 * 10 ^ 6)
               If d > 500 And d <= 635 Then Cells(y, x2) = (0.01603 * d - 5.44881) * 10 ^ 6
               If d > 635 And d <= 715 Then Cells(y, x2) = (0.16635 * d - 100.90225) * 10 ^ 6
               If d > 715 And d <= 785 Then Cells(y, x2) = (-0.22103 * d + 176.07343) * 10 ^ 6
               If d > 785 Then Cells(y, x2) = (2.566 * 10 ^ 6)
            End If

         Next y


   EndX = (8 + 2 * XCells)
   EndY = (10 - 1 + YCells)

   Next
```

```
    'draws a border around the calculation cells'
    With Range(Cells(10, (8 - 1 + XCells + 2)), Cells(EndY, EndX))
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeTop).Weight = xlMedium
    .Borders(xlEdgeLeft).Weight = xlMedium
    .Borders(xlEdgeRight).Weight = xlMedium
    End With

End Sub
```

Module 6: ThermalConductivityInitial

```
Sub ThermalConductivityInitial()

    LengthY = Cells(7, 9)
    LengthX = Cells(8, 9)

    deltax = Cells(1, 9)
    deltay = Cells(2, 9)

    YCells = LengthY / deltay + 1
    XCells = LengthX / deltax + 1

    'calculates the thermal conductivity for the initial temperature step at each node (cell)'
    For x3 = (8 + 2 * XCells + 2) To (8 + 3 * XCells + 1)
        For y = 10 To (10 - 1 + YCells)

            d = Cells(y, x3 - 2 * XCells - 2)

            If Cells(1, 6) = "Siliceous" Then
                If d >= 0 And d <= 800 Then Cells(y, x3) = (-0.000625 * d + 1.5)
                If d > 800 Then Cells(y, x3) = 1
            End If

            If Cells(1, 6) = "Carbonate" Then
                If d >= 0 And d <= 293 Then Cells(y, x3) = 1.355
                If d > 293 Then Cells(y, x3) = (-0.001241 * d + 1.7162)
            End If

          Next y

    EndX = (8 + 3 * XCells + 1)
    EndY = (10 - 1 + YCells)

    Next
    'draws a border around the calculation cells'
    With Range(Cells(10, (8 + 2 * XCells + 2)), Cells(EndY, EndX))
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeTop).Weight = xlMedium
    .Borders(xlEdgeLeft).Weight = xlMedium
    .Borders(xlEdgeRight).Weight = xlMedium
    End With

End Sub
```

Module 7: TemperatureStepFourSided

```
Sub TemperatureStepFourSided()

'calcualtes the number of time steps needed'
Range("B1", Range("B1").End(xlDown)).Select
I = Selection.Cells.Count

Cells(10, 8).Select

ActiveCell.Offset(0, -3) = "time"
ActiveCell.Offset(0, -2) = Cells(2, 1).Value
ActiveCell.Offset(1, -3) = "temp"
ActiveCell.Offset(1, -2) = Cells(2, 2).Value

For q = 3 To I + 1

'time/temperature step data'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)

ActiveCell.Offset(YCells + 1, -3) = "time"
ActiveCell.Offset(YCells + 1, -2) = Cells(q, 1).Value
ActiveCell.Offset(YCells + 2, -3) = "temp"
ActiveCell.Offset(YCells + 2, -2) = Cells(q, 2).Value
'top left corner cell'

'coloring'
```

```
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(YCells + 1, 0).Activate

kij = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 2)).Value
kijplusi = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 3)).Value
kright = (2 * kijplusi * kij) / (kijplusi + kij)

kijminusj = ActiveCell.Offset(-(YCells + 0), (2 * XCells + 2)).Value
kbottom = (2 * kij * kijminusj) / (kijminusj + kij)

Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
Ts = ActiveCell.Offset(-(YCells + 1), 0).Value

If Ts - Tam = 0 Then

   qx = 0
   qy = 0
Else
   hr = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Ts - Tam)
   qx = Cells(1, 13) * (Ts - Tam) + hr * (Ts - Tam)
   qy = Cells(1, 13) * (Ts - Tam) + hr * (Ts - Tam)
End If

Tsj1 = ActiveCell.Offset(-(YCells + 0), 0).Value
Tsi1 = ActiveCell.Offset(-(YCells + 1), 1).Value

Pc = ActiveCell.Offset(-(YCells + 1), (XCells + 1)).Value



Tijmplus1 = ((-qx * (deltay / 2)) - (kbottom * ((Ts - Tsj1) / deltay) * (deltax / 2)) - (qy * (deltax / 2)) +
(kright * ((Tsi1 - Ts) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Ts

ActiveCell.Value = Tijmplus1

'interior cells'
ActiveCell.Offset(-(YCells + 1), 0).Select
```

```
For x = 1 To (XCells - 2)
   For y = 1 To (YCells - 2)

   ActiveCell.Offset(y, x).Select

   'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = None
   ElseIf ActiveCell > 500 Then
   ActiveCell.Interior.ColorIndex = None
   Else
   ActiveCell.Interior.ColorIndex = 4
   End If


   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kip1j = ActiveCell.Offset(0, 1).Value
   kim1j = ActiveCell.Offset(0, -1).Value
   kijp1 = ActiveCell.Offset(-1, 0).Value
   kijm1 = ActiveCell.Offset(1, 0).Value

   kleft = (2 * kij * kim1j) / (kij + kim1j)
   kright = (2 * kip1j * kij) / (kip1j + kij)
   kbottom = (2 * kij * kijm1) / (kij + kijm1)
   ktop = (2 * kij * kijp1) / (kij + kijp1)


   ActiveCell.Offset(0, 2 * -XCells - 2).Activate

   ActiveCell.Offset((YCells + 1), 0).Value = (-(kleft * ((Tij - Tim1j) / deltax) * deltay) - (kbottom * ((Tij -
Tijm1) / deltay) * deltax) + (kright * ((Tip1j - Tij) / deltax) * deltay) + (ktop * ((Tijp1 - Tij) / deltay) *
deltax)) * (deltat / (Pc * deltax * deltay)) + Tij

   ActiveCell.Offset(-y, -x).Select
   Next y
   Next x


'top right corner cell'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate

'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate
Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijm1 = ActiveCell.Offset(1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
kbottom = (2 * kij * kijm1) / (kij + kijm1)


Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate
```

```
Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
   qy = 0
Else

   hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

End If

Tijm1 = ActiveCell.Offset(1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value


Tijmplus1 = (-(kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (kbottom * ((Tij - Tijm1) / deltay) * (deltax / 2)) - qx * deltax / 2 - qy * deltay / 2) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij



ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom right corner cell'

ActiveCell.Offset(-(YCells + 1), -(XCells - 1)).Select


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
'coloring'
```

```
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate


Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)


Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate

ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
   qy = 0
Else

   hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value
```

```
Tijmplus1 = (-(qy * (deltax / 2)) - (kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (qx * (deltay / 2)) + (ktop *
((Tijp1 - Tij) / deltay) * (deltax / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij

ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom left corner cell'

ActiveCell.Offset(-(2 * YCells), -(XCells - 1)).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



ActiveCell.Offset(YCells - 1, 0).Activate
'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate


Pc = ActiveCell.Value


ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kip1j = ActiveCell.Offset(0, 1).Value
kright = (2 * kij * kip1j) / (kij + kip1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)

Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
```

```
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
   qy = 0
Else

   hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tip1j = ActiveCell.Offset(0, 1).Value


Tijmplus1 = (-(qx * (deltay / 2)) - (qy * (deltax / 2)) + (ktop * ((Tijp1 - Tij) / deltay) * (deltax / 2)) + (kright
* ((Tip1j - Tij) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij


ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1


'right vertical face'

ActiveCell.Offset(-(2 * YCells), 0).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)
```

```
    ActiveCell.Offset(0, XCells - 1).Activate
    ActiveCell.Offset(y, 0).Activate
    'coloring'

    If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(-1, 1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijp1 = ActiveCell.Offset(-1, 0).Value
    Tijm1 = ActiveCell.Offset(1, 0).Value

    If Tij - Tam = 0 Then

        qx = 0
        qy = 0
    Else

        hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
        qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
        qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kleft = (2 * kij * kim1j) / (kij + kim1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)

    Tijmplus1 = (-((kleft) * ((Tij - Tim1j) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij

    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate
```

```
   ActiveCell.Value = Tijmplus1

   ActiveCell.Offset(-(YCells + 1 + y), -(XCells - 1)).Select


   Next y

'left vertical face'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(y, 0).Activate
   'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tam = 0 Then

      qx = 0
      qy = 0
   Else

      hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
      qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
      qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

   End If
```

```
    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)

    Tijmplus1 = (((kright) * ((Tip1j - Tij) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij

    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1 + y), 0).Select



    Next y



'bottom face'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



For x = 1 To (XCells - 2)
```

```
  ActiveCell.Offset(YCells - 1, 0).Activate
  ActiveCell.Offset(0, x).Activate

  'coloring'
  If ActiveCell < 500 And ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(1, -1) = "" Then
  ActiveCell.Interior.ColorIndex = 4
  End If

  Tij = ActiveCell.Value
  Tip1j = ActiveCell.Offset(0, 1).Value
  Tim1j = ActiveCell.Offset(0, -1).Value
  Tijp1 = ActiveCell.Offset(-1, 0).Value

  If Tij - Tam = 0 Then

    qx = 0
    qy = 0
  Else

    hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
    qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
    qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

  End If

  ActiveCell.Offset(0, (XCells + 1)).Activate
  Pc = ActiveCell.Value

  ActiveCell.Offset(0, (XCells + 1)).Activate
  kij = ActiveCell.Value
  kip1j = ActiveCell.Offset(0, 1).Value
  kim1j = ActiveCell.Offset(0, -1).Value
  kijp1 = ActiveCell.Offset(-1, 0).Value

  kright = (2 * kij * kip1j) / (kij + kip1j)
  ktop = (2 * kij * kijp1) / (kij + kijp1)
  kleft = (2 * kij * kim1j) / (kij + kim1j)

  Tijmplus1 = (((ktop) * ((Tijp1 - Tij) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay / 2)) +
((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) + Tij



  ActiveCell.Offset(0, 2 * -XCells - 2).Activate

  ActiveCell.Offset((YCells + 1), 0).Activate
```

```
    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(2 * YCells), -x).Select



    Next x



'top face'



LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For x = 1 To (XCells - 2)

    ActiveCell.Offset(0, x).Activate
    'coloring'
    If ActiveCell < 500 And ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And
ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijm1 = ActiveCell.Offset(1, 0).Value

    If Tij - Tam = 0 Then

        qx = 0
        qy = 0
    Else

        hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
        qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
        qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)

    Tijmplus1 = (-(kbottom * ((Tij - Tijm1) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay /
2)) + ((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij


    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1), -x).Select



    Next x

    ActiveCell.Offset((YCells + 1), 0).Activate


    With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeTop).Weight = xlMedium
    .Borders(xlEdgeLeft).Weight = xlMedium
    .Borders(xlEdgeRight).Weight = xlMedium
    End With

'properties'
    LengthY = Cells(7, 9)
    LengthX = Cells(8, 9)
```

```
deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

ActiveCell.Offset(0, XCells + 1).Select
'thermal capacity'

For x2 = 0 To (XCells - 1)
   For y = 0 To (YCells - 1)

      ActiveCell.Offset(y, x2).Select
      d = ActiveCell.Offset(0, -(XCells + 1))


         If Cells(1, 6) = "Siliceous" Then

            If d >= 0 And d <= 200 Then ActiveCell.Value = (0.005 * d + 1.7) * 10 ^ 6
            If d > 200 And d <= 400 Then ActiveCell.Value = (2.7 * 10 ^ 6)
            If d > 400 And d <= 500 Then ActiveCell.Value = (0.013 * d - 2.5) * 10 ^ 6
            If d > 500 And d <= 600 Then ActiveCell.Value = (-0.013 * d + 10.5) * 10 ^ 6
            If d > 600 Then ActiveCell.Value = 2.7 * 10 ^ 6
         End If

        If Cells(1, 6) = "Carbonate" Then
            If d >= 0 And d <= 400 Then ActiveCell.Value = 2.566 * 10 ^ 6
            If d > 400 And d <= 410 Then ActiveCell.Value = (0.1765 * d - 68.034) * 10 ^ 6
            If d > 410 And d <= 445 Then ActiveCell.Value = (-0.05043 * d + 25.00671) * 10 ^ 6
            If d > 445 And d <= 500 Then ActiveCell.Value = (2.566 * 10 ^ 6)
            If d > 500 And d <= 635 Then ActiveCell.Value = (0.01603 * d - 5.44881) * 10 ^ 6
            If d > 635 And d <= 715 Then ActiveCell.Value = (0.16635 * d - 100.90225) * 10 ^ 6
            If d > 715 And d <= 785 Then ActiveCell.Value = (-0.22103 * d + 176.07343) * 10 ^ 6
            If d > 785 Then ActiveCell.Value = (2.566 * 10 ^ 6)
         End If

      ActiveCell.Offset(-y, -x2).Select
      Next y

Next
With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
.Borders(xlEdgeBottom).Weight = xlMedium
.Borders(xlEdgeTop).Weight = xlMedium
.Borders(xlEdgeLeft).Weight = xlMedium
.Borders(xlEdgeRight).Weight = xlMedium
End With
```

```
'thermal conductivity'

ActiveCell.Offset(0, XCells + 1).Select


For x2 = 0 To (XCells - 1)
    For y = 0 To (YCells - 1)

      ActiveCell.Offset(y, x2).Select
      d = ActiveCell.Offset(0, -2 * (XCells + 1))


        If Cells(1, 6) = "Siliceous" Then
          If d >= 0 And d <= 800 Then ActiveCell.Value = (-0.000625 * d + 1.5)
          If d > 800 Then ActiveCell.Value = 1
        End If

        If Cells(1, 6) = "Carbonate" Then
          If d >= 0 And d <= 293 Then ActiveCell.Value = 1.355
          If d > 293 Then ActiveCell.Value = (-0.001241 * d + 1.7162)
        End If

       ActiveCell.Offset(-y, -x2).Select
      Next y


Next

  With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
  .Borders(xlEdgeBottom).Weight = xlMedium
  .Borders(xlEdgeTop).Weight = xlMedium
  .Borders(xlEdgeLeft).Weight = xlMedium
  .Borders(xlEdgeRight).Weight = xlMedium
  End With

ActiveCell.Offset(0, -2 * (XCells + 1)).Select


Next q


For u = 0 To YCells

  Rows(ActiveCell.Offset(u, 0).Row).Clear
  Next u
```

```
End Sub
```

Module 8: PropertiesStep

```
Sub PropertiesStep()

    LengthY = Cells(7, 9)
    LengthX = Cells(8, 9)

    deltax = Cells(1, 9)
    deltay = Cells(2, 9)

    YCells = LengthY / deltay
    XCells = LengthX / deltax

    ActiveCell.Offset(0, XCells + 1).Select
    'thermal capacity'

    For x2 = 0 To (XCells - 1)
        For y = 0 To (YCells - 1)

            ActiveCell.Offset(y, x2).Select
            d = ActiveCell.Offset(0, -(XCells + 1))


                If Cells(1, 6) = "Siliceous" Then

                    If d >= 0 And d <= 200 Then ActiveCell.Value = (0.005 * d + 1.7) * 10 ^ 6
                    If d > 200 And d <= 400 Then ActiveCell.Value = (2.7 * 10 ^ 6)
                    If d > 400 And d <= 500 Then ActiveCell.Value = (0.013 * d - 2.5) * 10 ^ 6
                    If d > 500 And d <= 600 Then ActiveCell.Value = (-0.013 * d + 10.5) * 10 ^ 6
                    If d > 600 Then ActiveCell.Value = 2.7 * 10 ^ 6
                End If

               If Cells(1, 6) = "Carbonate" Then
                    If d >= 0 And d <= 400 Then ActiveCell.Value = 2.566 * 10 ^ 6
                    If d > 400 And d <= 410 Then ActiveCell.Value = (0.1765 * d - 68.034) * 10 ^ 6
                    If d > 410 And d <= 445 Then ActiveCell.Value = (-0.05043 * d + 25.00671) * 10 ^ 6
                    If d > 445 And d <= 500 Then ActiveCell.Value = (2.566 * 10 ^ 6)
                    If d > 500 And d <= 635 Then ActiveCell.Value = (0.01603 * d - 5.44881) * 10 ^ 6
                    If d > 635 And d <= 715 Then ActiveCell.Value = (0.16635 * d - 100.90225) * 10 ^ 6
                    If d > 715 And d <= 785 Then ActiveCell.Value = (-0.22103 * d + 176.07343) * 10 ^ 6
                    If d > 785 Then ActiveCell.Value = (2.566 * 10 ^ 6)
                End If

            ActiveCell.Offset(-y, -x2).Select
            Next y
```

```
   Next
   With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
   .Borders(xlEdgeBottom).Weight = xlMedium
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

'thermal conductivity'

ActiveCell.Offset(0, XCells + 1).Select


For x2 = 0 To (XCells - 1)
      For y = 0 To (YCells - 1)

         ActiveCell.Offset(y, x2).Select
         d = ActiveCell.Offset(0, -2 * (XCells + 1))


            If Cells(1, 6) = "Siliceous" Then
               If d >= 0 And d <= 800 Then ActiveCell.Value = (-0.000625 * d + 1.5)
               If d > 800 Then ActiveCell.Value = 1
            End If

            If Cells(1, 6) = "Carbonate" Then
               If d >= 0 And d <= 293 Then ActiveCell.Value = 1.355
               If d > 293 Then ActiveCell.Value = (-0.001241 * d + 1.7162)
            End If

           ActiveCell.Offset(-y, -x2).Select
         Next y


Next

   With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
   .Borders(xlEdgeBottom).Weight = xlMedium
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

ActiveCell.Offset(0, -2 * (XCells + 1)).Select


End Sub
```

Module 9: TemperatureStepOneSidedLeft

```
Sub TemperatureStepOneSidedLeft()

'calculates the number of time steps needed'
Range("B1", Range("B1").End(xlDown)).Select
I = Selection.Cells.Count

Cells(10, 8).Select

ActiveCell.Offset(0, -3) = "time"
ActiveCell.Offset(0, -2) = Cells(2, 1).Value
ActiveCell.Offset(1, -3) = "temp"
ActiveCell.Offset(1, -2) = Cells(2, 2).Value

For q = 3 To (I + 1)

'time/temperature step data'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)

ActiveCell.Offset(YCells + 1, -3) = "time"
ActiveCell.Offset(YCells + 1, -2) = Cells(q, 1).Value
ActiveCell.Offset(YCells + 2, -3) = "temp"
ActiveCell.Offset(YCells + 2, -2) = Cells(q, 2).Value

'top left corner cell'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(YCells + 1, 0).Activate

kij = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 2)).Value
kijplusi = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 3)).Value
kright = (2 * kijplusi * kij) / (kijplusi + kij)
```

```
kijminusj = ActiveCell.Offset(-(YCells + 0), (2 * XCells + 2)).Value
kbottom = (2 * kij * kijminusj) / (kijminusj + kij)

Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
Ts = ActiveCell.Offset(-(YCells + 1), 0).Value


If Ts - Tam = 0 Then

   qx = 0
Else

   hrx = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Ts - Tam)
   qx = Cells(1, 13) * (Ts - Tam) + hrx * (Ts - Tam)

End If


If Ts - Tamb = 0 Then

   qy = 0

Else
   hry = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Ts - Tamb)
   qy = Cells(1, 13) * (Ts - Tamb) + hry * (Ts - Tamb)
End If


Tsj1 = ActiveCell.Offset(-(YCells + 0), 0).Value
Tsi1 = ActiveCell.Offset(-(YCells + 1), 1).Value

Pc = ActiveCell.Offset(-(YCells + 1), (XCells + 1)).Value

Tijmplus1 = ((-qx * (deltay / 2)) - (kbottom * ((Ts - Tsj1) / deltay) * (deltax / 2)) - (qy * (deltax / 2)) +
(kright * ((Tsi1 - Ts) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Ts


ActiveCell.Value = Tijmplus1


'interior cells'
ActiveCell.Offset(-(YCells + 1), 0).Select
```

```
For x = 1 To (XCells - 2)
   For y = 1 To (YCells - 2)


   ActiveCell.Offset(y, x).Activate

   'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = None
   ElseIf ActiveCell > 500 Then
   ActiveCell.Interior.ColorIndex = None
   Else
   ActiveCell.Interior.ColorIndex = 4
   End If
   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kip1j = ActiveCell.Offset(0, 1).Value
   kim1j = ActiveCell.Offset(0, -1).Value
   kijp1 = ActiveCell.Offset(-1, 0).Value
   kijm1 = ActiveCell.Offset(1, 0).Value

   kleft = (2 * kij * kim1j) / (kij + kim1j)
   kright = (2 * kip1j * kij) / (kip1j + kij)
   kbottom = (2 * kij * kijm1) / (kij + kijm1)
   ktop = (2 * kij * kijp1) / (kij + kijp1)


   ActiveCell.Offset(0, 2 * -XCells - 2).Activate

   ActiveCell.Offset((YCells + 1), 0).Value = (-(kleft * ((Tij - Tim1j) / deltax) * deltay) - (kbottom * ((Tij -
Tijm1) / deltay) * deltax) + (kright * ((Tip1j - Tij) / deltax) * deltay) + (ktop * ((Tijp1 - Tij) / deltay) *
deltax)) * (deltat / (Pc * deltax * deltay)) + Tij

   ActiveCell.Offset(-y, -x).Select
   Next y
   Next x


'top right corner cell'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate

'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate
Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijm1 = ActiveCell.Offset(1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
kbottom = (2 * kij * kijm1) / (kij + kijm1)


Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tamb = 0 Then

   qx = 0
   qy = 0
Else

hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)

End If

Tijm1 = ActiveCell.Offset(1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value


Tijmplus1 = (-(kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (kbottom * ((Tij - Tijm1) / deltay) * (deltax /
2)) - qx * deltax / 2 - qy * deltay / 2) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij



ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom right corner cell'

ActiveCell.Offset(-(YCells + 1), -(XCells - 1)).Select


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
```

```
'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate

Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)


Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate

ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tamb = 0 Then

   qx = 0
   qy = 0
Else

hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
```

```
Tim1j = ActiveCell.Offset(0, -1).Value


Tijmplus1 = (-(qy * (deltax / 2)) - (kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (qx * (deltay / 2)) + (ktop *
((Tijp1 - Tij) / deltay) * (deltax / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij

ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom left corner cell'

ActiveCell.Offset(-(2 * YCells), -(XCells - 1)).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



ActiveCell.Offset(YCells - 1, 0).Activate
'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate

Pc = ActiveCell.Value


ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kip1j = ActiveCell.Offset(0, 1).Value
kright = (2 * kij * kip1j) / (kij + kip1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)

Cells(q, 2).Select
Tam = ActiveCell.Value
```

```
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then
   qx = 0
Else
   hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hrx * (Tij - Tam)
End If

If Tij - Tamb = 0 Then
   qy = 0
Else
   hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
   qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tip1j = ActiveCell.Offset(0, 1).Value


Tijmplus1 = (-(qx * (deltay / 2)) - (qy * (deltax / 2)) + (ktop * ((Tijp1 - Tij) / deltay) * (deltax / 2)) + (kright
* ((Tip1j - Tij) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij


ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1


'right vertical face'

ActiveCell.Offset(-(2 * YCells), 0).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
```

```
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(0, XCells - 1).Activate
   ActiveCell.Offset(y, 0).Activate
'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(-1, 1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tamb = 0 Then
      qx = 0
      qy = 0
   Else

      hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
      qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
   End If

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kim1j = ActiveCell.Offset(0, -1).Value
   kijp1 = ActiveCell.Offset(-1, 0).Value
   kijm1 = ActiveCell.Offset(1, 0).Value

   kleft = (2 * kij * kim1j) / (kij + kim1j)
   kbottom = (2 * kij * kijm1) / (kij + kijm1)
   ktop = (2 * kij * kijp1) / (kij + kijp1)

   Tijmplus1 = (-((kleft) * ((Tij - Tim1j) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij
```

```
   ActiveCell.Offset(0, 2 * -XCells - 2).Activate

   ActiveCell.Offset((YCells + 1), 0).Activate

   ActiveCell.Value = Tijmplus1

   ActiveCell.Offset(-(YCells + 1 + y), -(XCells - 1)).Select


   Next y

'left vertical face'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(y, 0).Activate
'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tam = 0 Then
      qx = 0
   Else
      hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
      qx = Cells(1, 13) * (Tij - Tam) + hrx * (Tij - Tam)
   End If
```

```
    If Tij - Tamb = 0 Then
        qy = 0
    Else
        hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
        qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)

    Tijmplus1 = (((kright) * ((Tip1j - Tij) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij

    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1 + y), 0).Select



    Next y



'bottom face'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)
```

```
YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



For x = 1 To (XCells - 2)

   ActiveCell.Offset(YCells - 1, 0).Activate
   ActiveCell.Offset(0, x).Activate

'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value

   If Tij - Tamb = 0 Then
      qx = 0
      qy = 0
   Else

      hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
      qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
   End If

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kip1j = ActiveCell.Offset(0, 1).Value
   kim1j = ActiveCell.Offset(0, -1).Value
   kijp1 = ActiveCell.Offset(-1, 0).Value

   kright = (2 * kij * kip1j) / (kij + kip1j)
   ktop = (2 * kij * kijp1) / (kij + kijp1)
   kleft = (2 * kij * kim1j) / (kij + kim1j)
   Tijmplus1 = (((ktop) * ((Tijp1 - Tij) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay / 2)) +
((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) + Tij



   ActiveCell.Offset(0, 2 * -XCells - 2).Activate

   ActiveCell.Offset((YCells + 1), 0).Activate

   ActiveCell.Value = Tijmplus1

   ActiveCell.Offset(-(2 * YCells), -x).Select



   Next x



'top face'



LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For x = 1 To (XCells - 2)

   ActiveCell.Offset(0, x).Activate
'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And
ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value
```

```
    If Tij - Tamb = 0 Then
        qx = 0
        qy = 0
    Else

        hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
        hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
        qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
        qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)

    Tijmplus1 = (-(kbottom * ((Tij - Tijm1) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay /
2)) + ((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij


    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1), -x).Select



    Next x

    ActiveCell.Offset((YCells + 1), 0).Activate


    With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeTop).Weight = xlMedium
```

```
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

'properties'
   LengthY = Cells(7, 9)
   LengthX = Cells(8, 9)

   deltax = Cells(1, 9)
   deltay = Cells(2, 9)

   YCells = LengthY / deltay + 1
   XCells = LengthX / deltax + 1

   ActiveCell.Offset(0, XCells + 1).Select
   'thermal capacity'

   For x2 = 0 To (XCells - 1)
      For y = 0 To (YCells - 1)

         ActiveCell.Offset(y, x2).Select
         d = ActiveCell.Offset(0, -(XCells + 1))


            If Cells(1, 6) = "Siliceous" Then

               If d >= 0 And d <= 200 Then ActiveCell.Value = (0.005 * d + 1.7) * 10 ^ 6
               If d > 200 And d <= 400 Then ActiveCell.Value = (2.7 * 10 ^ 6)
               If d > 400 And d <= 500 Then ActiveCell.Value = (0.013 * d - 2.5) * 10 ^ 6
               If d > 500 And d <= 600 Then ActiveCell.Value = (-0.013 * d + 10.5) * 10 ^ 6
               If d > 600 Then ActiveCell.Value = 2.7 * 10 ^ 6
            End If

           If Cells(1, 6) = "Carbonate" Then
               If d >= 0 And d <= 400 Then ActiveCell.Value = 2.566 * 10 ^ 6
               If d > 400 And d <= 410 Then ActiveCell.Value = (0.1765 * d - 68.034) * 10 ^ 6
               If d > 410 And d <= 445 Then ActiveCell.Value = (-0.05043 * d + 25.00671) * 10 ^ 6
               If d > 445 And d <= 500 Then ActiveCell.Value = (2.566 * 10 ^ 6)
               If d > 500 And d <= 635 Then ActiveCell.Value = (0.01603 * d - 5.44881) * 10 ^ 6
               If d > 635 And d <= 715 Then ActiveCell.Value = (0.16635 * d - 100.90225) * 10 ^ 6
               If d > 715 And d <= 785 Then ActiveCell.Value = (-0.22103 * d + 176.07343) * 10 ^ 6
               If d > 785 Then ActiveCell.Value = (2.566 * 10 ^ 6)
            End If

         ActiveCell.Offset(-y, -x2).Select
         Next y

   Next
   With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
   .Borders(xlEdgeBottom).Weight = xlMedium
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

'thermal conductivity'

ActiveCell.Offset(0, XCells + 1).Select


For x2 = 0 To (XCells - 1)
      For y = 0 To (YCells - 1)

         ActiveCell.Offset(y, x2).Select
         d = ActiveCell.Offset(0, -2 * (XCells + 1))


            If Cells(1, 6) = "Siliceous" Then
               If d >= 0 And d <= 800 Then ActiveCell.Value = (-0.000625 * d + 1.5)
               If d > 800 Then ActiveCell.Value = 1
            End If

            If Cells(1, 6) = "Carbonate" Then
               If d >= 0 And d <= 293 Then ActiveCell.Value = 1.355
               If d > 293 Then ActiveCell.Value = (-0.001241 * d + 1.7162)
            End If
          ActiveCell.Offset(-y, -x2).Select
         Next y


Next

   With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
   .Borders(xlEdgeBottom).Weight = xlMedium
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

ActiveCell.Offset(0, -2 * (XCells + 1)).Select

Next q

For u = 0 To YCells

   Rows(ActiveCell.Offset(u, 0).Row).Clear
```

```
    Next u


End Sub
```

Module 10: Structural

```
Sub Structural()


    'Asking the user to input time curve'
    Dim Struct As VbMsgBoxResult
    Struct = MsgBox("An error may occur in the following analysis. This is due either to the result that no
500 degree isotherms remain or too small of a cross section remains to complete the analysis. If this is
the case, click End no the error message box.", vbOKOnly, "Structural Error Message Notification")

    'asking user to input concrete cover thickness'
    dprime = InputBox("Please enter cover d' value (meters)", "Concerte Cover (d')")
    If dprime = "" Then Exit Sub

    Cells(1, 16) = "d'"
    Cells(1, 17) = dprime
    Cells(1, 18) = "meters"

    Asteel = InputBox("Please enter area of steel for left side (As)(meters^2)", "Left Side Area of Steel
(As)")
    If Asteel = "" Then Exit Sub
    Cells(5, 16) = "As"
    Cells(5, 17) = Asteel
    Cells(5, 18) = "meters^2"

    Asteelprime = InputBox("Please enter area of steel for right side (As')(meters^2)", "Right Side Area of
Steel (As')")
    If Asteelprime = "" Then Exit Sub
    Cells(6, 16) = "As'"
    Cells(6, 17) = Asteelprime
    Cells(6, 18) = "meters^2"

    'asking the user to input steel yield strength at 20 C (ambient)'
    fyo = InputBox("Please enter steel yield strength (Pa) at 20 C", "Initial Steel Yield Strength (fyo)")
    If fyo = "" Then Exit Sub
    Cells(2, 16) = "fyo"
    Cells(2, 17) = fyo
    Cells(2, 18) = "Pa"

    'asking the user to input steel Modulus of Elasticity at 20 C (ambient)'
```

```
   Eo = InputBox("Please enter steel Modulus of Elasticity (Pa) at 20 C", "Initial Steel Modulus of Elasticity
(Eo)")
   If Eo = "" Then Exit Sub
   Cells(3, 16) = "Eo"
   Cells(3, 17) = Eo
   Cells(3, 18) = "Pa"

   'asking the user for fprimec'
   fprimec = InputBox("Please enter concrete yield strength f'c (Pa)", "Concrete Yield Strength (f'c)")
   If fprimec = "" Then Exit Sub

   'asking user for eccentricity'
   eccentricity = InputBox("Please enter eccentricity of applied load (meters)", "Eccentricity (e)")
   If eccentricity = "" Then Exit Sub



   'ask user for end condition'
   ktara = InputBox("Please enter equivalent end condition factor k", "Equivalent end condition factor
(k)")
   If ktara = "" Then Exit Sub

   'ask user for column length'
   Lu = InputBox("Please enter column length Lu (meters)", "Column Length (Lu)")
   If Lu = "" Then Exit Sub

   'ask for dead load'
   DeadLoad = InputBox("Please enter Factored Dead Axial Load (Newtons)", "Dead Axial Load")
   If DeadLoad = "" Then Exit Sub

   'ask for total load'
   LiveLoad = InputBox("Please enter Factored Live Axial Load (Newtons)", "Live Axial Load")
   If LiveLoad = "" Then Exit Sub

   Dim DoYouWantS As VbMsgBoxResult
   DoYouWantS = MsgBox("Do you want to start the structural analysis calculations. Click Yes to start; No
to exit", vbYesNo, "Structural Analysis Calculations")
   If DoYouWantS = vbNo Then Exit Sub




   Cells(7, 16) = "e"
   Cells(7, 17) = eccentricity
   Cells(7, 18) = "meters"

   Cells(4, 16) = "f'c"
   Cells(4, 17) = fprimec
   Cells(4, 18) = "Pa"
```

```
    Cells(1, 20) = "k"
    Cells(1, 21) = ktara

    Cells(2, 20) = "Lu"
    Cells(2, 21) = Lu
    Cells(2, 22) = "meters"

    Cells(3, 20) = "DL"
    Cells(3, 21) = DeadLoad
    Cells(3, 22) = "Newtons"

    Cells(4, 20) = "LL"
    Cells(4, 21) = LiveLoad
    Cells(4, 22) = "meters"


    LengthY = Cells(7, 9)
    LengthX = Cells(8, 9)

    deltax = Cells(1, 9)
    deltay = Cells(2, 9)

    YCells = LengthY / deltay + 1
    XCells = LengthX / deltax + 1




    'start loop'

    Range("B1", Range("B1").End(xlDown)).Select
    I = Selection.Cells.Count

    Cells(10, 8).Select




For q = 3 To I + 1

    Set Nemo = ActiveCell
    Set Storm = ActiveCell.Offset(XCells - 1, YCells - 1)
    Set Winter = Range(Nemo, Storm)

    AverageT = WorksheetFunction.Average(Winter)
    AverageT = AverageT * 1.8 + 32
```

```
   x3 = 0
   k2 = 0
   ar2 = 0
   'calculation of centroid'

   For x2 = 0 To (XCells - 1)
      For y = 0 To (YCells - 1)

         ActiveCell.Offset(y, x2).Select
         If Selection.Interior.ColorIndex = 4 Then
            Set tree = ActiveCell
            Set stockholm = ActiveCell.Offset(0, -x2)
            Set Rng = Range(tree, stockholm)

            x = Application.WorksheetFunction.CountIf(Rng, ">0")
            x = x - 1

            k = 1
            k2 = k + k2


         Else
            x = 0
         End If

         x3 = x3 + x

         If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(0, 1) <> Empty And ActiveCell.Offset(1, 1) <> Empty
And ActiveCell.Offset(1, 0) <> Empty Then



            ar = 1
         Else
            ar = 0
         End If

         ar2 = ar2 + ar


         ActiveCell.Offset(-y, -x2).Select

      Next y
      Next x2

   centroidx = (x3 * deltax) / k2
```

```
  area = ar2 * (deltax * deltay)

  Jyy = 0
  SumT = 0
  totalcell = 0
   'moment of inertia'

  For y1 = 0 To (YCells - 1)
    For x5 = 0 To (XCells - 1)

      ActiveCell.Offset(y1, x5).Select


      If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(0, 1) <> Empty And ActiveCell.Offset(1, 1) <> Empty
And ActiveCell.Offset(1, 0) <> Empty Then
      'totalcell = totalcell + 1'
        Jlocal = (deltay * deltax ^ 3) / 12

        Set tree = ActiveCell
        Set stockholm = ActiveCell.Offset(0, -x5)
        Set Rng = Range(tree, stockholm)

        x = Application.WorksheetFunction.CountIf(Rng, ">0")
        x = (x * deltax) - deltax / 2

        distance = LengthX / 2 - (x)

        littlearea = deltay * deltax


        J = Jlocal + distance ^ 2 * littlearea

        Jyy = Jyy + J

      'SumT = SumT + ActiveCell.Value'



      End If



      ActiveCell.Offset(-y1, -x5).Select

    Next x5
    Next y1
```

```
      If q = 3 Then
         Joriginal = Jyy
      End If





      'AverageT = SumT / totalcell'
      'calculate the radius of gyration'

   Radiusy = (Jyy / area) ^ (1 / 2)


   Jxx = 0
   SumT = 0
   totalcell = 0
    'moment of inertia'

   For x5 = 0 To (XCells - 1)
      For y1 = 0 To (YCells - 1)

         ActiveCell.Offset(y1, x5).Select


         If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(0, 1) <> Empty And ActiveCell.Offset(1, 1) <> Empty
And ActiveCell.Offset(1, 0) <> Empty Then
         'totalcell = totalcell + 1'
            Jlocal = (deltax * deltay ^ 3) / 12

            Set tree = ActiveCell
            Set stockholm = ActiveCell.Offset(-y1, 0)
            Set Rng = Range(tree, stockholm)

            y = Application.WorksheetFunction.CountIf(Rng, ">0")
            y = (y * deltay) - deltay / 2

            distance = LengthY / 2 - (y)

            littlearea = deltax * deltay


            J = Jlocal + distance ^ 2 * littlearea

            Jxx = Jxx + J

         'SumT = SumT + ActiveCell.Value'
```

```
        End If



        ActiveCell.Offset(-y1, -x5).Select

    Next y1
    Next x5

    If q = 3 Then
        Joriginal = Jxx
    End If




    'AverageT = SumT / totalcell'
    'calculate the radius of gyration'
    Radiusx = (Jxx / area) ^ (1 / 2)





ActiveCell.Offset((YCells - 1) / 2, 0).Select

'determining how many cells are over 500 degrees C to eliminate them from calculation'
Set cell1 = ActiveCell
Set cell2 = ActiveCell.Offset(0, XCells - 1)
Set Rng = Range(cell1, cell2)

x = Application.WorksheetFunction.CountIf(Rng, "<=500")
Height = (x - 1) * deltax


'calculation of b from temperature profile'
ActiveCell.Offset(-(YCells - 1) / 2, 0).Select
ActiveCell.Offset(0, (XCells - 1) / 2).Select

Set cell3 = ActiveCell
Set cell4 = ActiveCell.Offset(YCells, 0)
Set Rng = Range(cell3, cell4)

y = Application.WorksheetFunction.CountIf(Rng, "<=500")
ActiveCell.Offset((YCells - 1) / 2, -(XCells - 1) / 2).Select
```

```
'calculation of d'

d = (x - 1) * deltax - dprime

If d < 0 Then
Exit Sub
ElseIf d > LengthX Then
Exit Sub
End If


If q = 3 Then
   doriginal = (x - 1) * deltax - dprime
End If


ActiveCell.Offset(0, 3 * (XCells + 1)).Select

ActiveCell = "d"
ActiveCell.Offset(0, 1) = d
ActiveCell.Offset(0, 2) = "meters"




'calculation of temperature at (As/d)'
 r = dprime / deltax

h = Application.WorksheetFunction.RoundUp(r, 0)
l = Application.WorksheetFunction.RoundDown(r, 0)


'ActiveCell.Offset(0, 2) = h'
'ActiveCell.Offset(0, 3) = l'

ActiveCell.Offset(0, -(3 * (XCells + 1))).Select

tempd = (((ActiveCell.Offset(0, h) - ActiveCell.Offset(0, l)) / (1)) * (r - l)) + ActiveCell.Offset(0, l)

'calculation of temperature at (As'/d')'
rr = doriginal / deltax

hh = Application.WorksheetFunction.RoundUp(rr, 0)
LL = Application.WorksheetFunction.RoundDown(rr, 0)
```

```
    tempdprime = (((ActiveCell.Offset(0, hh) - ActiveCell.Offset(0, LL)) / (1)) * (rr - LL)) +
ActiveCell.Offset(0, LL)

    ActiveCell.Offset(0, 3 * (XCells + 1)).Select
    ActiveCell.Offset(1, 0) = "As Temp"
    ActiveCell.Offset(1, 1) = tempd

    ActiveCell.Offset(2, 0) = "As' Temp"
    ActiveCell.Offset(2, 1) = tempdprime



    ActiveCell.Offset(3, 0) = "fy"
    ActiveCell.Offset(4, 0) = "fy'"
    ActiveCell.Offset(1, 2) = "C"
    ActiveCell.Offset(2, 2) = "C"
    ActiveCell.Offset(3, 2) = "Pa"
    ActiveCell.Offset(4, 2) = "Pa"



    'calculation of steel yield at As/d at elevated temperatures'
    If tempd > 0 And tempd <= 600 Then
        fy = (1 + ((tempd) / (900 * Log(tempd / 1750)))) * fyo
    ElseIf tempd > 600 Then
        fy = ((340 - 0.34 * tempd) / (tempd - 240)) * fyo
    Else
        fy = ""
    End If

    ActiveCell.Offset(3, 1) = fy

    'calculation of steel yield at As'/d' at elevated temperature'
    If tempdprime > 0 And tempdprime <= 600 Then
        fyprime = (1 + ((tempdprime) / (900 * Log(tempdprime / 1750)))) * fyo
    ElseIf tempdprime > 600 Then
        fyprime = ((340 - 0.34 * tempdprime) / (tempdprime - 240)) * fyo
    Else
        fy = ""
    End If

    ActiveCell.Offset(4, 1) = fyprime

    'calculation of Modulous of Elasticity (Et) at elevated temperature'

    ActiveCell.Offset(-2, 0) = "Et"
    ActiveCell.Offset(-1, 0) = "Et'"
    ActiveCell.Offset(-2, 2) = "Pa"
ActiveCell.Offset(-1, 2) = "Pa"

If tempd > 0 And tempd <= 600 Then
   Et = (1 + ((tempd) / (2000 * Log(tempd / 1100)))) * Eo
ElseIf tempd > 600 Then
   Et = ((690 - 0.69 * tempd) / (tempd - 53.5)) * Eo
Else
   Et = ""
End If

ActiveCell.Offset(-2, 1) = Et

'calculation of Modulous of Elasticity (Et')at elevated temperature'

If tempdprime > 0 And tempdprime <= 600 Then
   Etprime = (1 + ((tempdprime) / (2000 * Log(tempdprime / 1100)))) * Eo
ElseIf tempdprime > 600 Then
   Etprime = ((690 - 0.69 * tempdprime) / (tempdprime - 53.5)) * Eo
Else
   Etprime = ""
End If

ActiveCell.Offset(-1, 1) = Etprime

'calculation of xb'
xb = (0.003 / ((fy / Et) + 0.003)) * d
ActiveCell.Offset(-2, 4) = xb
ActiveCell.Offset(-2, 3) = "xb"
ActiveCell.Offset(-2, 5) = "meters"

If q = 3 Then
   xoriginal = xb
End If


'calculations of beta1'

If fprimec <= 30 * 10 ^ 6 Then
   Beta = 0.85
Else
   Beta = 0.85 - 0.008 * (10 ^ -6) * ((fprimec - 30 * 10 ^ 6))
End If

If Beta < 0.65 Then Beta = 0.65

ActiveCell.Offset(-1, 3) = "Beta1"
ActiveCell.Offset(-1, 4) = Beta
```

```
   'calculation of b'
   b = (y - 1) * deltay

   'Calculation of Cc'
   Ccb = 0.85 * fprimec * Beta * xb * b
   ActiveCell.Offset(0, 3) = "Cc"
   ActiveCell.Offset(0, 4) = Ccb
   ActiveCell.Offset(0, 5) = "N"

   'Calculation of T'
   Tb = Asteel * fy
   ActiveCell.Offset(1, 3) = "T"
   ActiveCell.Offset(1, 4) = Tb
   ActiveCell.Offset(1, 5) = "N"

   'Calculation of Cs'

   Csb = Asteelprime * (fy - 0.85 * fprimec)
   ActiveCell.Offset(2, 3) = "Cs"
   ActiveCell.Offset(2, 4) = Csb
   ActiveCell.Offset(2, 5) = "N"

   'Calculation of Pb'
   Pb = Ccb + Csb - Tb
   ActiveCell.Offset(3, 3) = "Pb"
   ActiveCell.Offset(3, 4) = Pb
   ActiveCell.Offset(3, 5) = "N"

   'entering of centroid of concrete'
   ActiveCell.Offset(4, 3) = "centroidx"
   ActiveCell.Offset(4, 4) = centroidx
   ActiveCell.Offset(4, 5) = "meters"

   'area'

   ActiveCell.Offset(-2, 6) = "area"
   ActiveCell.Offset(-2, 7) = area
   ActiveCell.Offset(-2, 8) = "meters^2"

   'location of plastic centroid (from left face)'

   ddprime = ((0.85 * fprimec * area * centroidx) + (Asteel * fy * dprime) + (Asteelprime * fyprime *
(LengthX - dprime))) / ((0.85 * fprimec * area) + (Asteel * fy) + (Asteelprime * fyprime)) - dprime


   ActiveCell.Offset(-1, 6) = "d"""
```

```
    ActiveCell.Offset(-1, 7) = ddprime
    ActiveCell.Offset(-1, 8) = "meters"

    'calculation of eb'

    eb = ((Ccb * (d - Beta * xb / 2 - ddprime)) + Csb * (d - dprime - ddprime) + Tb * ddprime) / Pb
    ActiveCell.Offset(0, 6) = "eb"
    ActiveCell.Offset(0, 7) = eb
    ActiveCell.Offset(0, 8) = "meters"


    'calculation of Pn'

    If q = 3 Then
        eccentricityoriginal = dprime + ddprime + eccentricity
    End If

    eccentricity = eccentricityoriginal - dprime - ddprime

    If Abs(eb) > Cells(7, 17) Then

        ActiveCell.Offset(-2, 9) = "x"
        If q = 3 Then
            ActiveCell.Offset(-2, 10) = xoriginal
        Else
            ActiveCell.Offset(-2, 10) = ActiveCell.Offset(-2 - (YCells + 1), 10).Value
        End If

        ActiveCell.Offset(-2, 11) = "meters"

        ActiveCell.Offset(-1, 9) = "first term"
        ActiveCell.Offset(-1, 10) = Asteelprime * (fyprime - 0.85 * fprimec) * ((LengthX / 2 - eccentricity) -
dprime)


        ActiveCell.Offset(0, 9) = "h/2-e"
        ActiveCell.Offset(0, 10) = LengthX / 2 - eccentricity

        ActiveCell.Offset(1, 9) = "second term"
        ActiveCell.Offset(1, 10) = 0.85 * fprimec * b * Beta

        ActiveCell.Offset(2, 9) = "h-d'"
        ActiveCell.Offset(2, 10) = Height - dprime

        ActiveCell.Offset(2, 6) = "length-d'"
        ActiveCell.Offset(2, 7) = LengthX - dprime
    ActiveCell.Offset(1, 6) = "As"
    ActiveCell.Offset(1, 7) = Asteel
    ActiveCell.Offset(1, 8) = "meters^2"

    ActiveCell.Offset(3, 9) = "Moment Pn"
    ActiveCell.Offset(3, 10).Select
    ActiveCell.FormulaR1C1 = "=R[-4]C-R[-2]C*R[-5]C*((R[-4]C[-6])/2*R[-5]C-R[-3]C)+ (R[-2]C[-3])*(((R[-5]C[-9])*(0.003)*((R[-1]C)-(R[-5]C)))/(R[-5]C))*((R[-1]C[-3])-(R[-3]C))"

    ActiveCell.Offset(-3, -10).Select

    ActiveCell.Offset(3, 10).GoalSeek Goal:=0, ChangingCell:=ActiveCell.Offset(-2, 10)

    ActiveCell.Offset(-2, 12) = "Cs"
    ActiveCell.Offset(-2, 13) = Asteelprime * (fyprime - 0.85 * fprimec)
    ActiveCell.Offset(-2, 14) = "N"

    ActiveCell.Offset(-1, 12) = "Cc"
    ActiveCell.Offset(-1, 13) = 0.85 * fprimec * b * (Beta * ActiveCell.Offset(-2, 10))
    ActiveCell.Offset(-1, 14) = "N"

    ActiveCell.Offset(0, 12) = "T"
    ActiveCell.Offset(0, 13) = Asteel * Et * 0.003 * (Height - dprime - ActiveCell.Offset(-2, 10)) / ActiveCell.Offset(-2, 10)
    ActiveCell.Offset(0, 14) = "N"

    ActiveCell.Offset(1, 12) = "Pn"
    Pn = ActiveCell.Offset(-2, 13) + ActiveCell.Offset(-1, 13) - ActiveCell.Offset(0, 13)
    ActiveCell.Offset(1, 13) = Pn
    ActiveCell.Offset(1, 14) = "N"




    If (Cells(3, 21) + Cells(4, 21)) < ActiveCell.Offset(1, 13) Then
      ActiveCell.Offset(-4, 6) = "Safe"
    Else
      ActiveCell.Offset(-4, 6) = "Fails"
    End If
    ActiveCell.Offset(-4, 5) = "Pn check"

  'Does the column fail'

    ActiveCell.Offset(4, 9) = "Mn"
    Mn = (ActiveCell.Offset(1, 13).Value) * eccentricity
    ActiveCell.Offset(4, 10) = Mn
    ActiveCell.Offset(4, 11) = "N*m"

End If


If Abs(eb) < Cells(7, 17) Then

    ActiveCell.Offset(-2, 9) = "x"
    If q = 3 Then
        ActiveCell.Offset(-2, 10) = xoriginal
    Else
        ActiveCell.Offset(-2, 10) = ActiveCell.Offset(-2 - (YCells + 1), 10).Value
    End If
    ActiveCell.Offset(-2, 11) = "meters"

    ActiveCell.Offset(-1, 9) = "T"
    ActiveCell.Offset(-1, 10) = Asteel * fy
    ActiveCell.Offset(-1, 11) = "N"

    ActiveCell.Offset(0, 9) = "Cs"
    Cs = Asteelprime * (fyprime - 0.85 * fprimec)
    ActiveCell.Offset(0, 10) = Cs
    ActiveCell.Offset(0, 11) = "N"

    ActiveCell.Offset(1, 9) = "Ccterm"
    ActiveCell.Offset(1, 10) = 0.85 * fprimec * b * Beta


    ActiveCell.Offset(2, 9) = "Cc"
    ActiveCell.Offset(2, 10).Select
    ActiveCell.FormulaR1C1 = "=R[-1]C*R[-4]C"


    ActiveCell.Offset(1, 0).Select
    ActiveCell.Offset(0, -1) = "Pn"
    ActiveCell.FormulaR1C1 = "=R[-1]C+R[-3]C-R[-4]C"
    Pn = ActiveCell.Value
    ActiveCell.Offset(0, 1) = "N"


    ActiveCell.Offset(-3, -10).Select
    ActiveCell.Offset(-2, 12) = "First Term"
    ActiveCell.Offset(-2, 13) = eccentricity + (d - dprime) / 2

    ActiveCell.Offset(-1, 12) = "Second Term"
    ActiveCell.Offset(-1, 13) = Cs * (d - dprime)

    ActiveCell.Offset(0, 12) = "Moment Pn"
```

```
        ActiveCell.Offset(0, 13).Select
        ActiveCell.FormulaR1C1 = "=-(R[3]C[-3]*R[-2]C)+R[2]C[-3]*(R[0]C[-12]-(R[-1]C[-9]*R[-2]C[-3])/2)+R[-
1]C"


        ActiveCell.Offset(0, -13).Select

        ActiveCell.Offset(0, 13).GoalSeek Goal:=0, ChangingCell:=ActiveCell.Offset(-2, 10)

        If (Cells(3, 21) + Cells(4, 21)) > ActiveCell.Offset(3, 10) Then
            ActiveCell.Offset(-4, 6) = "Fails"
        Else
            ActiveCell.Offset(-4, 6) = "Safe"
        End If
        ActiveCell.Offset(-4, 5) = "Pn check"

        'verify strain condition on the compression steel'


        ActiveCell.Offset(1, 12) = "es'"
        ActiveCell.Offset(1, 13) = 0.003 * (ActiveCell.Offset(-2, 10) - dprime) / ActiveCell.Offset(-2, 10)


        If 0.003 * (ActiveCell.Offset(-2, 10) - dprime) / ActiveCell.Offset(-2, 10) > (fyprime / Etprime) Then
            ActiveCell.Offset(1, 13).Interior.ColorIndex = 4
        Else
            ActiveCell.Offset(1, 13).Interior.ColorIndex = 3
        End If

        'Does the column fail'

        ActiveCell.Offset(4, 9) = "Mn"
        Mn = (ActiveCell.Offset(3, 10).Value) * eccentricity
        ActiveCell.Offset(4, 10) = Mn
        ActiveCell.Offset(4, 11) = "N*m"

    End If
    'radius of gyrationy'
    ActiveCell.Offset(-3, 15) = "ry"
    ActiveCell.Offset(-3, 16) = Radiusy
    ActiveCell.Offset(-3, 17) = "meters"


    'radius of gyrationx'
    ActiveCell.Offset(-2, 15) = "rx"
    ActiveCell.Offset(-2, 16) = Radiusx
    ActiveCell.Offset(-2, 17) = "meters"
    'test to see if Mmax is needed'

    'slenderness ratio'

    jarobski = ktara * Lu / Radiusx
    ActiveCell.Offset(3, 15) = "slenderness ratio x"
    ActiveCell.Offset(3, 16) = jarobski

    cotnoir = ktara * Lu / Radiusy
    ActiveCell.Offset(4, 15) = "slenderness ratio y"
    ActiveCell.Offset(4, 16) = cotnoir


    Bd = DeadLoad / (0 + DeadLoad + LiveLoad)

    MoE = 4700000 * (fprimec) ^ (1 / 2)
   'debate on whether to use temperature or not'

  '      'calculate Modulus of Elasticity'

    If Cells(1, 6) = "Carbonate" Then

        Cpercent = (1 * 10 ^ -5) * AverageT ^ 2 - 0.0751 * AverageT + 104.02
        If Cpercent > 100 Then
            Cpercent = 100
        End If

        MoE = MoE * (Cpercent / 100)
        If MoE < 0 Then
            MoE = 0
        End If


        End If



        If Cells(1, 6) = "Siliceous" Then

            Spercent = (6 * 10 ^ -6) * AverageT ^ 2 - 0.0664 * AverageT + 94.117
            If Spercent > 100 Then
                Spercent = 100
            End If

            MoE = MoE * (Spercent / 100)

            If MoE < 0 Then
                MoE = 0
            End If
   End If

   If Cells(1, 6) = "Expanded Shale" Then

      ESpercent = (5 * 10 ^ -5) * AverageT ^ 2 - (0.1259) * AverageT + 107.84
      If ESpercent > 100 Then
         ESpercent = 100
      End If

      MoE = MoE * (ESpercent / 100)

      If MoE < 0 Then
         MoE = 0
      End If

   End If

   'EI and Pc'

EI = 0.4 * MoE * Joriginal / (1 + Bd)
ActiveCell.Offset(-1, 15) = "EI"
ActiveCell.Offset(-1, 16) = EI
ActiveCell.Offset(-1, 17) = "Pa*m^4"


Pc = (WorksheetFunction.Pi) ^ 2 * EI / (ktara * Lu) ^ 2
ActiveCell.Offset(0, 15) = "Pc"
ActiveCell.Offset(0, 16) = Pc
ActiveCell.Offset(0, 17) = "N"



alpha = (0 + DeadLoad + LiveLoad) / Pc
ActiveCell.Offset(1, 15) = "alpha"
ActiveCell.Offset(1, 16) = alpha

Cm = 1
Mmax = (0 + DeadLoad + LiveLoad) * eccentricity * (Cm / (1 - alpha))

ActiveCell.Offset(2, 15) = "Mmax"
ActiveCell.Offset(2, 16) = Mmax
ActiveCell.Offset(2, 17) = "N*m"

ActiveCell.Offset(-3, 5) = "Mmax check"

If Abs(Mmax) > Abs(Mn) Then
   ActiveCell.Offset(-3, 6) = "Fails"
```

```
    End If
    If Abs(Mmax) <= Abs(Mn) Then
        ActiveCell.Offset(-3, 6) = "Safe"
    End If



    'End If'


    ActiveCell.Offset(0, -3 * (XCells + 1)).Select
    ActiveCell.Offset(-(YCells - 1) / 2, 0).Select
    ActiveCell.Offset((YCells + 1), 0).Select


Next q

For Noise = 1 To 100
Beep
Next Noise

End Sub
```

Module 11: PostProcessing

```
Sub PostProcessing()

Sheets("Results").Select
Cells.ClearContents
Sheets("Calculations").Select

'Time/Temperature Curve Values'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1


Range("B1", Range("B1").End(xlDown)).Select
I = Selection.Cells.Count
```

```
Dim AskTemp As VbMsgBoxResult
AskTemp = MsgBox("Do you want nodal temperatures on results page?", vbYesNo, "Nodal
Temperature")
If AskTemp = vbYes Then

For w = 2 To 10

   Sheets("Calculations").Cells(10, 8).Select

   Xdirection = InputBox("Please enter number of cells in x direction from top cell including top left cell
and desired cell", "Cells X Direction")
   Ydirection = InputBox("Please enter number of cells in y direction from left cell including top left cell
and desired cell", "Cells Y Direction")

   ActiveCell.Offset(Ydirection - 1, Xdirection - 1).Select

   For q = 2 To I

      Temperature = ActiveCell.Value

      Xtime = Cells(q, 1).Value
      Sheets("Results").Cells(q, 1).Value = Xtime
      Sheets("Results").Cells(q, w).Value = Temperature
      Sheets("Results").Cells(1, w) = "Temp"
      Sheets("Results").Cells(1, 1) = "Time"

      ActiveCell.Offset(YCells + 1, 0).Select

   Next q

   Dim TempNode As VbMsgBoxResult
   TempNode = MsgBox("Would you like to select another node. Click Yes if you want to enter another
node; No to exit and complete", vbYesNo, "Next Temperature Node")
   If TempNode = vbYes Then w = w
   If TempNode = vbNo Then w = 10

Next w
End If

If AskNode = vbNo Then
End If



Sheets("Calculations").Cells(10, 8).Select
ActiveCell.Offset((YCells - 1) / 2, 0).Select
ActiveCell.Offset(0, 3 * (XCells + 1)).Select
```

```
For q = 2 To l

  Xtime = Cells(q, 1).Value
  Sheets("Results").Cells(q, 1).Value = Xtime

  Astemp = ActiveCell.Offset(1, 1).Value
  Sheets("Results").Cells(1, 11) = "As Temp"
  Sheets("Results").Cells(q, 11).Value = Astemp

  Astempprime = ActiveCell.Offset(2, 1).Value
  Sheets("Results").Cells(1, 12) = "As' Temp"
  Sheets("Results").Cells(q, 12).Value = Astempprime


  area = ActiveCell.Offset(-2, 7).Value
  Sheets("Results").Cells(1, 13) = "Area"
  Sheets("Results").Cells(q, 13).Value = area


  Mn = ActiveCell.Offset(4, 10).Value
  Sheets("Results").Cells(1, 14) = "Mn"
  Sheets("Results").Cells(q, 14).Value = Mn

  If Abs(ActiveCell.Offset(0, 7).Value) > Cells(7, 17).Value Then
  Pn = ActiveCell.Offset(1, 13).Value
  End If

  If Abs(ActiveCell.Offset(0, 7).Value) < Cells(7, 17) Then
  Pn = ActiveCell.Offset(3, 10).Value
  End If

  Sheets("Results").Cells(1, 15) = "Pn"
  Sheets("Results").Cells(q, 15).Value = Pn

  Mmax = ActiveCell.Offset(2, 16).Value
  Sheets("Results").Cells(1, 16) = "Mmax"
  Sheets("Results").Cells(q, 16).Value = Mmax

  eb = ActiveCell.Offset(0, 7).Value
  Sheets("Results").Cells(1, 17) = "eb"
  Sheets("Results").Cells(q, 17).Value = eb

  ddprime = ActiveCell.Offset(-1, 7).Value
  Sheets("Results").Cells(1, 18) = "d"""
  Sheets("Results").Cells(q, 18).Value = ddprime

  slenderx = ActiveCell.Offset(3, 16).Value
```

```
   Sheets("Results").Cells(1, 19) = "sl ratio x"
   Sheets("Results").Cells(q, 19) = slenderx

   slendery = ActiveCell.Offset(4, 16).Value
   Sheets("Results").Cells(1, 20) = "sl ratio y"
   Sheets("Results").Cells(q, 20) = slendery

   ActiveCell.Offset(YCells + 1, 0).Select

   Next q

For Noise = 1 To 100
Beep
Next Noise

End Sub
```

Module 12: TemperatureStepOneSidedRight

```
Sub TemperatureStepOneSidedRight()

'calculates the number of time steps needed'
Range("B1", Range("B1").End(xlDown)).Select
I = Selection.Cells.Count

Cells(10, 8).Select

ActiveCell.Offset(0, -3) = "time"
ActiveCell.Offset(0, -2) = Cells(2, 1).Value
ActiveCell.Offset(1, -3) = "temp"
ActiveCell.Offset(1, -2) = Cells(2, 2).Value

For q = 3 To (I + 1)

'time/temperature step data'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)
```

```
ActiveCell.Offset(YCells + 1, -3) = "time"
ActiveCell.Offset(YCells + 1, -2) = Cells(q, 1).Value
ActiveCell.Offset(YCells + 2, -3) = "temp"
ActiveCell.Offset(YCells + 2, -2) = Cells(q, 2).Value

'top left corner cell'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(YCells + 1, 0).Activate

kij = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 2)).Value
kijplusi = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 3)).Value
kright = (2 * kijplusi * kij) / (kijplusi + kij)

kijminusj = ActiveCell.Offset(-(YCells + 0), (2 * XCells + 2)).Value
kbottom = (2 * kij * kijminusj) / (kijminusj + kij)

Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
Ts = ActiveCell.Offset(-(YCells + 1), 0).Value


If Ts - Tamb = 0 Then

   qx = 0
Else

   hrx = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Ts - Tamb)
   qx = Cells(1, 13) * (Ts - Tamb) + hrx * (Ts - Tamb)

End If



If Ts - Tamb = 0 Then

   qy = 0

Else
   hry = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Ts - Tamb)
   qy = Cells(1, 13) * (Ts - Tamb) + hry * (Ts - Tamb)
```

```
End If


Tsj1 = ActiveCell.Offset(-(YCells + 0), 0).Value
Tsi1 = ActiveCell.Offset(-(YCells + 1), 1).Value

Pc = ActiveCell.Offset(-(YCells + 1), (XCells + 1)).Value

Tijmplus1 = ((-qx * (deltay / 2)) - (kbottom * ((Ts - Tsj1) / deltay) * (deltax / 2)) - (qy * (deltax / 2)) +
(kright * ((Tsi1 - Ts) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Ts


ActiveCell.Value = Tijmplus1


'interior cells'
ActiveCell.Offset(-(YCells + 1), 0).Select


For x = 1 To (XCells - 2)
   For y = 1 To (YCells - 2)


   ActiveCell.Offset(y, x).Activate

   'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = None
   ElseIf ActiveCell > 500 Then
   ActiveCell.Interior.ColorIndex = None
   Else
   ActiveCell.Interior.ColorIndex = 4
   End If
   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kip1j = ActiveCell.Offset(0, 1).Value
   kim1j = ActiveCell.Offset(0, -1).Value
```

```
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kleft = (2 * kij * kim1j) / (kij + kim1j)
    kright = (2 * kip1j * kij) / (kip1j + kij)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)


    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Value = (-(kleft * ((Tij - Tim1j) / deltax) * deltay) - (kbottom * ((Tij -
Tijm1) / deltay) * deltax) + (kright * ((Tip1j - Tij) / deltax) * deltay) + (ktop * ((Tijp1 - Tij) / deltay) *
deltax)) * (deltat / (Pc * deltax * deltay)) + Tij

    ActiveCell.Offset(-y, -x).Select



    Next y
    Next x


'top right corner cell'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate

'coloring'
    If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

ActiveCell.Offset(0, (XCells + 1)).Activate
```

```
Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijm1 = ActiveCell.Offset(1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
kbottom = (2 * kij * kijm1) / (kij + kijm1)


Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
Else

   hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hrx * (Tij - Tam)

End If


If Tij - Tamb = 0 Then

   qy = 0

Else
   hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
   qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
End If



Tijm1 = ActiveCell.Offset(1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value
```

```
Tijmplus1 = (-(kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (kbottom * ((Tij - Tijm1) / deltay) * (deltax /
2)) - qx * deltax / 2 - qy * deltay / 2) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij



ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom right corner cell'

ActiveCell.Offset(-(YCells + 1), -(XCells - 1)).Select


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate

'coloring'
  If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) < 500 Then
  ActiveCell.Interior.ColorIndex = 4
  End If

ActiveCell.Offset(0, (XCells + 1)).Activate

Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)


Cells(q, 2).Select
```

```
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate

ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
Else

   hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hrx * (Tij - Tam)

End If


If Tij - Tamb = 0 Then

   qy = 0

Else
   hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
   qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value


Tijmplus1 = (-(qy * (deltax / 2)) - (kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (qx * (deltay / 2)) + (ktop *
((Tijp1 - Tij) / deltay) * (deltax / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij

ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom left corner cell'

ActiveCell.Offset(-(2 * YCells), -(XCells - 1)).Select

LengthY = Cells(7, 9)
```

```
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



ActiveCell.Offset(YCells - 1, 0).Activate
'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate

Pc = ActiveCell.Value


ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kip1j = ActiveCell.Offset(0, 1).Value
kright = (2 * kij * kip1j) / (kij + kip1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)

Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tamb = 0 Then
   qx = 0
Else
   hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
   qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
End If

If Tij - Tamb = 0 Then
   qy = 0
Else
   hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
   qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tip1j = ActiveCell.Offset(0, 1).Value


Tijmplus1 = (-(qx * (deltay / 2)) - (qy * (deltax / 2)) + (ktop * ((Tijp1 - Tij) / deltay) * (deltax / 2)) + (kright
* ((Tip1j - Tij) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij


ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1


'right vertical face'

ActiveCell.Offset(-(2 * YCells), 0).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(0, XCells - 1).Activate
   ActiveCell.Offset(y, 0).Activate
'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(-1, 1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
```

```
    End If

    Tij = ActiveCell.Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijp1 = ActiveCell.Offset(-1, 0).Value
    Tijm1 = ActiveCell.Offset(1, 0).Value

If Tij - Tam = 0 Then

    qx = 0
Else

    hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
    qx = Cells(1, 13) * (Tij - Tam) + hrx * (Tij - Tam)

End If






    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kleft = (2 * kij * kim1j) / (kij + kim1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)

    Tijmplus1 = (-((kleft) * ((Tij - Tim1j) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij

    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1 + y), -(XCells - 1)).Select
```

```
   Next y

'left vertical face'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(y, 0).Activate
'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tamb = 0 Then
      qx = 0
   Else
      hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
   End If

   If Tij - Tamb = 0 Then
      qy = 0
   Else
      hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)

   End If

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value
```

```
    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)

    Tijmplus1 = (((kright) * ((Tip1j - Tij) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij

    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1 + y), 0).Select



    Next y



'bottom face'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



For x = 1 To (XCells - 2)

    ActiveCell.Offset(YCells - 1, 0).Activate
    ActiveCell.Offset(0, x).Activate

'coloring'
    If ActiveCell < 500 And ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijp1 = ActiveCell.Offset(-1, 0).Value

    If Tij - Tamb = 0 Then
        qx = 0
        qy = 0
    Else

        hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
        hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
        qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
        qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    ktop = (2 * kij * kijp1) / (kij + kijp1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)

    Tijmplus1 = (((ktop) * ((Tijp1 - Tij) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay / 2)) +
((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) + Tij



    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1
```

```
   ActiveCell.Offset(-(2 * YCells), -x).Select



   Next x



'top face'



LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For x = 1 To (XCells - 2)

   ActiveCell.Offset(0, x).Activate
'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And
ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tamb = 0 Then
      qx = 0
      qy = 0
   Else

      hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
      qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
   End If
```

```
    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)

    Tijmplus1 = (-(kbottom * ((Tij - Tijm1) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay /
2)) + ((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij


    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1), -x).Select



    Next x

    ActiveCell.Offset((YCells + 1), 0).Activate


    With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeTop).Weight = xlMedium
    .Borders(xlEdgeLeft).Weight = xlMedium
    .Borders(xlEdgeRight).Weight = xlMedium
    End With

'properties'
    LengthY = Cells(7, 9)
    LengthX = Cells(8, 9)

    deltax = Cells(1, 9)
    deltay = Cells(2, 9)
```

```
  YCells = LengthY / deltay + 1
  XCells = LengthX / deltax + 1

  ActiveCell.Offset(0, XCells + 1).Select
  'thermal capacity'

  For x2 = 0 To (XCells - 1)
    For y = 0 To (YCells - 1)

      ActiveCell.Offset(y, x2).Select
      d = ActiveCell.Offset(0, -(XCells + 1))


        If Cells(1, 6) = "Siliceous" Then

          If d >= 0 And d <= 200 Then ActiveCell.Value = (0.005 * d + 1.7) * 10 ^ 6
          If d > 200 And d <= 400 Then ActiveCell.Value = (2.7 * 10 ^ 6)
          If d > 400 And d <= 500 Then ActiveCell.Value = (0.013 * d - 2.5) * 10 ^ 6
          If d > 500 And d <= 600 Then ActiveCell.Value = (-0.013 * d + 10.5) * 10 ^ 6
          If d > 600 Then ActiveCell.Value = 2.7 * 10 ^ 6
        End If

       If Cells(1, 6) = "Carbonate" Then
          If d >= 0 And d <= 400 Then ActiveCell.Value = 2.566 * 10 ^ 6
          If d > 400 And d <= 410 Then ActiveCell.Value = (0.1765 * d - 68.034) * 10 ^ 6
          If d > 410 And d <= 445 Then ActiveCell.Value = (-0.05043 * d + 25.00671) * 10 ^ 6
          If d > 445 And d <= 500 Then ActiveCell.Value = (2.566 * 10 ^ 6)
          If d > 500 And d <= 635 Then ActiveCell.Value = (0.01603 * d - 5.44881) * 10 ^ 6
          If d > 635 And d <= 715 Then ActiveCell.Value = (0.16635 * d - 100.90225) * 10 ^ 6
          If d > 715 And d <= 785 Then ActiveCell.Value = (-0.22103 * d + 176.07343) * 10 ^ 6
          If d > 785 Then ActiveCell.Value = (2.566 * 10 ^ 6)
        End If

      ActiveCell.Offset(-y, -x2).Select
      Next y

  Next
  With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
  .Borders(xlEdgeBottom).Weight = xlMedium
  .Borders(xlEdgeTop).Weight = xlMedium
  .Borders(xlEdgeLeft).Weight = xlMedium
  .Borders(xlEdgeRight).Weight = xlMedium
  End With

'thermal conductivity'

ActiveCell.Offset(0, XCells + 1).Select
```

```
For x2 = 0 To (XCells - 1)
    For y = 0 To (YCells - 1)

      ActiveCell.Offset(y, x2).Select
      d = ActiveCell.Offset(0, -2 * (XCells + 1))


        If Cells(1, 6) = "Siliceous" Then
          If d >= 0 And d <= 800 Then ActiveCell.Value = (-0.000625 * d + 1.5)
          If d > 800 Then ActiveCell.Value = 1
        End If

        If Cells(1, 6) = "Carbonate" Then
          If d >= 0 And d <= 293 Then ActiveCell.Value = 1.355
          If d > 293 Then ActiveCell.Value = (-0.001241 * d + 1.7162)
        End If

       ActiveCell.Offset(-y, -x2).Select
      Next y


Next

  With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
  .Borders(xlEdgeBottom).Weight = xlMedium
  .Borders(xlEdgeTop).Weight = xlMedium
  .Borders(xlEdgeLeft).Weight = xlMedium
  .Borders(xlEdgeRight).Weight = xlMedium
  End With

ActiveCell.Offset(0, -2 * (XCells + 1)).Select

Next q

For u = 0 To YCells

  Rows(ActiveCell.Offset(u, 0).Row).Clear
  Next u


End Sub
```

Module 13: TemperatureStepThreeRightSided

```
Sub TemperatureStepThreeRightSided()
```

```
'calcualtes the number of time steps needed'
Range("B1", Range("B1").End(xlDown)).Select
I = Selection.Cells.Count

Cells(10, 8).Select

ActiveCell.Offset(0, -3) = "time"
ActiveCell.Offset(0, -2) = Cells(2, 1).Value
ActiveCell.Offset(1, -3) = "temp"
ActiveCell.Offset(1, -2) = Cells(2, 2).Value

For q = 3 To I + 1

'time/temperature step data'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)

ActiveCell.Offset(YCells + 1, -3) = "time"
ActiveCell.Offset(YCells + 1, -2) = Cells(q, 1).Value
ActiveCell.Offset(YCells + 2, -3) = "temp"
ActiveCell.Offset(YCells + 2, -2) = Cells(q, 2).Value

'top left corner cell'

'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(YCells + 1, 0).Activate

kij = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 2)).Value
kijplusi = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 3)).Value
kright = (2 * kijplusi * kij) / (kijplusi + kij)

kijminusj = ActiveCell.Offset(-(YCells + 0), (2 * XCells + 2)).Value
kbottom = (2 * kij * kijminusj) / (kijminusj + kij)
```

```
Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
Ts = ActiveCell.Offset(-(YCells + 1), 0).Value

If Ts - Tam = 0 Then

   qx = 0
   qy = 0
Else
   hr = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Ts - Tam)
   qx = Cells(1, 13) * (Ts - Tam) + hr * (Ts - Tam)
   qy = Cells(1, 13) * (Ts - Tam) + hr * (Ts - Tam)
End If

Tsj1 = ActiveCell.Offset(-(YCells + 0), 0).Value
Tsi1 = ActiveCell.Offset(-(YCells + 1), 1).Value

Pc = ActiveCell.Offset(-(YCells + 1), (XCells + 1)).Value



Tijmplus1 = ((-qx * (deltay / 2)) - (kbottom * ((Ts - Tsj1) / deltay) * (deltax / 2)) - (qy * (deltax / 2)) +
(kright * ((Tsi1 - Ts) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Ts

ActiveCell.Value = Tijmplus1

'interior cells'
ActiveCell.Offset(-(YCells + 1), 0).Select


For x = 1 To (XCells - 2)
   For y = 1 To (YCells - 2)

   ActiveCell.Offset(y, x).Select

   'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = None
   ElseIf ActiveCell > 500 Then
   ActiveCell.Interior.ColorIndex = None
   Else
   ActiveCell.Interior.ColorIndex = 4
```

```
    End If


    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijp1 = ActiveCell.Offset(-1, 0).Value
    Tijm1 = ActiveCell.Offset(1, 0).Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kleft = (2 * kij * kim1j) / (kij + kim1j)
    kright = (2 * kip1j * kij) / (kip1j + kij)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)


    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Value = (-(kleft * ((Tij - Tim1j) / deltax) * deltay) - (kbottom * ((Tij -
Tijm1) / deltay) * deltax) + (kright * ((Tip1j - Tij) / deltax) * deltay) + (ktop * ((Tijp1 - Tij) / deltay) *
deltax)) * (deltat / (Pc * deltax * deltay)) + Tij

    ActiveCell.Offset(-y, -x).Select



    Next y
    Next x


'top right corner cell'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)
```

```
YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate

'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate
Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijm1 = ActiveCell.Offset(1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
kbottom = (2 * kij * kijm1) / (kij + kijm1)


Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tamb = 0 Then

   qx = 0
Else

   hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
   qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)

End If
```

```
If Tij - Tam = 0 Then

   qy = 0

Else
   hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hry * (Tij - Tam)
End If



Tijm1 = ActiveCell.Offset(1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value


Tijmplus1 = (-(kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (kbottom * ((Tij - Tijm1) / deltay) * (deltax /
2)) - qx * deltax / 2 - qy * deltay / 2) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij



ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom right corner cell'

ActiveCell.Offset(-(YCells + 1), -(XCells - 1)).Select


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate

'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = 4
```

```
    End If

ActiveCell.Offset(0, (XCells + 1)).Activate

Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)


Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate

ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tamb = 0 Then

    qx = 0
Else

    hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
    qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)

End If


If Tij - Tam = 0 Then

    qy = 0

Else
    hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
    qy = Cells(1, 13) * (Tij - Tam) + hry * (Tij - Tam)
End If
```

```
Tijp1 = ActiveCell.Offset(-1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value


Tijmplus1 = (-(qy * (deltax / 2)) - (kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (qx * (deltay / 2)) + (ktop *
((Tijp1 - Tij) / deltay) * (deltax / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij

ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom left corner cell'

ActiveCell.Offset(-(2 * YCells), -(XCells - 1)).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



ActiveCell.Offset(YCells - 1, 0).Activate
'coloring'
  If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) = "" Then
  ActiveCell.Interior.ColorIndex = 4
  End If

ActiveCell.Offset(0, (XCells + 1)).Activate


Pc = ActiveCell.Value


ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kip1j = ActiveCell.Offset(0, 1).Value
kright = (2 * kij * kip1j) / (kij + kip1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)
```

```
Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
   qy = 0
Else

   hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tip1j = ActiveCell.Offset(0, 1).Value


Tijmplus1 = (-(qx * (deltay / 2)) - (qy * (deltax / 2)) + (ktop * ((Tijp1 - Tij) / deltay) * (deltax / 2)) + (kright
* ((Tip1j - Tij) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij


ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1


'right vertical face'

ActiveCell.Offset(-(2 * YCells), 0).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1
```

```
deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(0, XCells - 1).Activate
   ActiveCell.Offset(y, 0).Activate
'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(-1, 1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tamb = 0 Then
      qx = 0
      qy = 0
   Else

      hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
      qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)
   End If

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kim1j = ActiveCell.Offset(0, -1).Value
   kijp1 = ActiveCell.Offset(-1, 0).Value
   kijm1 = ActiveCell.Offset(1, 0).Value

   kleft = (2 * kij * kim1j) / (kij + kim1j)
   kbottom = (2 * kij * kijm1) / (kij + kijm1)
   ktop = (2 * kij * kijp1) / (kij + kijp1)

   Tijmplus1 = (-((kleft) * ((Tij - Tim1j) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax / 2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) + Tij

   ActiveCell.Offset(0, 2 * -XCells - 2).Activate
```

```
    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1 + y), -(XCells - 1)).Select


    Next y

'left vertical face'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

    ActiveCell.Offset(y, 0).Activate
    'coloring'

    If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tijp1 = ActiveCell.Offset(-1, 0).Value
    Tijm1 = ActiveCell.Offset(1, 0).Value

    If Tij - Tam = 0 Then

        qx = 0
        qy = 0
    Else

        hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
        qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
        qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
```

```
  End If

  ActiveCell.Offset(0, (XCells + 1)).Activate
  Pc = ActiveCell.Value

  ActiveCell.Offset(0, (XCells + 1)).Activate
  kij = ActiveCell.Value
  kip1j = ActiveCell.Offset(0, 1).Value
  kijp1 = ActiveCell.Offset(-1, 0).Value
  kijm1 = ActiveCell.Offset(1, 0).Value

  kright = (2 * kij * kip1j) / (kij + kip1j)
  kbottom = (2 * kij * kijm1) / (kij + kijm1)
  ktop = (2 * kij * kijp1) / (kij + kijp1)

  Tijmplus1 = (((kright) * ((Tip1j - Tij) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax / 2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) + Tij

  ActiveCell.Offset(0, 2 * -XCells - 2).Activate

  ActiveCell.Offset((YCells + 1), 0).Activate

  ActiveCell.Value = Tijmplus1

  ActiveCell.Offset(-(YCells + 1 + y), 0).Select



  Next y



'bottom face'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)
```

```
For x = 1 To (XCells - 2)

    ActiveCell.Offset(YCells - 1, 0).Activate
    ActiveCell.Offset(0, x).Activate

    'coloring'
    If ActiveCell < 500 And ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijp1 = ActiveCell.Offset(-1, 0).Value

    If Tij - Tam = 0 Then

        qx = 0
        qy = 0
    Else

        hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
        qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
        qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    ktop = (2 * kij * kijp1) / (kij + kijp1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)

    Tijmplus1 = (((ktop) * ((Tijp1 - Tij) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay / 2)) +
((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) + Tij
```

```
    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(2 * YCells), -x).Select



    Next x



'top face'



LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For x = 1 To (XCells - 2)

    ActiveCell.Offset(0, x).Activate
    'coloring'
    If ActiveCell < 500 And ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And
ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijm1 = ActiveCell.Offset(1, 0).Value

    If Tij - Tam = 0 Then

        qx = 0
        qy = 0
    Else

      hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
      qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
      qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)

    Tijmplus1 = (-(kbottom * ((Tij - Tijm1) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay /
2)) + ((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij


    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1), -x).Select



    Next x

    ActiveCell.Offset((YCells + 1), 0).Activate


    With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
    .Borders(xlEdgeBottom).Weight = xlMedium
    .Borders(xlEdgeTop).Weight = xlMedium
    .Borders(xlEdgeLeft).Weight = xlMedium
    .Borders(xlEdgeRight).Weight = xlMedium
    End With
```

```
'properties'
  LengthY = Cells(7, 9)
  LengthX = Cells(8, 9)

  deltax = Cells(1, 9)
  deltay = Cells(2, 9)

  YCells = LengthY / deltay + 1
  XCells = LengthX / deltax + 1

  ActiveCell.Offset(0, XCells + 1).Select
  'thermal capacity'

  For x2 = 0 To (XCells - 1)
    For y = 0 To (YCells - 1)

      ActiveCell.Offset(y, x2).Select
      d = ActiveCell.Offset(0, -(XCells + 1))


        If Cells(1, 6) = "Siliceous" Then

          If d >= 0 And d <= 200 Then ActiveCell.Value = (0.005 * d + 1.7) * 10 ^ 6
          If d > 200 And d <= 400 Then ActiveCell.Value = (2.7 * 10 ^ 6)
          If d > 400 And d <= 500 Then ActiveCell.Value = (0.013 * d - 2.5) * 10 ^ 6
          If d > 500 And d <= 600 Then ActiveCell.Value = (-0.013 * d + 10.5) * 10 ^ 6
          If d > 600 Then ActiveCell.Value = 2.7 * 10 ^ 6
        End If

       If Cells(1, 6) = "Carbonate" Then
          If d >= 0 And d <= 400 Then ActiveCell.Value = 2.566 * 10 ^ 6
          If d > 400 And d <= 410 Then ActiveCell.Value = (0.1765 * d - 68.034) * 10 ^ 6
          If d > 410 And d <= 445 Then ActiveCell.Value = (-0.05043 * d + 25.00671) * 10 ^ 6
          If d > 445 And d <= 500 Then ActiveCell.Value = (2.566 * 10 ^ 6)
          If d > 500 And d <= 635 Then ActiveCell.Value = (0.01603 * d - 5.44881) * 10 ^ 6
          If d > 635 And d <= 715 Then ActiveCell.Value = (0.16635 * d - 100.90225) * 10 ^ 6
          If d > 715 And d <= 785 Then ActiveCell.Value = (-0.22103 * d + 176.07343) * 10 ^ 6
          If d > 785 Then ActiveCell.Value = (2.566 * 10 ^ 6)
        End If

      ActiveCell.Offset(-y, -x2).Select
      Next y

  Next
  With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
  .Borders(xlEdgeBottom).Weight = xlMedium
  .Borders(xlEdgeTop).Weight = xlMedium
  .Borders(xlEdgeLeft).Weight = xlMedium
```

```
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

'thermal conductivity'

ActiveCell.Offset(0, XCells + 1).Select


For x2 = 0 To (XCells - 1)
       For y = 0 To (YCells - 1)

          ActiveCell.Offset(y, x2).Select
          d = ActiveCell.Offset(0, -2 * (XCells + 1))


             If Cells(1, 6) = "Siliceous" Then
                If d >= 0 And d <= 800 Then ActiveCell.Value = (-0.000625 * d + 1.5)
                If d > 800 Then ActiveCell.Value = 1
             End If

             If Cells(1, 6) = "Carbonate" Then
                If d >= 0 And d <= 293 Then ActiveCell.Value = 1.355
                If d > 293 Then ActiveCell.Value = (-0.001241 * d + 1.7162)
             End If

           ActiveCell.Offset(-y, -x2).Select
          Next y


Next

   With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
   .Borders(xlEdgeBottom).Weight = xlMedium
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

ActiveCell.Offset(0, -2 * (XCells + 1)).Select


Next q


For u = 0 To YCells

   Rows(ActiveCell.Offset(u, 0).Row).Clear
   Next u
```

```
End Sub
```

Module 14: TemperatureStepThreeLeftSided

```
Sub TemperatureStepThreeLeftSided()

'calcualtes the number of time steps needed'
Range("B1", Range("B1").End(xlDown)).Select
I = Selection.Cells.Count

Cells(10, 8).Select

ActiveCell.Offset(0, -3) = "time"
ActiveCell.Offset(0, -2) = Cells(2, 1).Value
ActiveCell.Offset(1, -3) = "temp"
ActiveCell.Offset(1, -2) = Cells(2, 2).Value

For q = 3 To I + 1

'time/temperature step data'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)

ActiveCell.Offset(YCells + 1, -3) = "time"
ActiveCell.Offset(YCells + 1, -2) = Cells(q, 1).Value
ActiveCell.Offset(YCells + 2, -3) = "temp"
ActiveCell.Offset(YCells + 2, -2) = Cells(q, 2).Value
'top left corner cell'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) < 500 And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(YCells + 1, 0).Activate

kij = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 2)).Value
kijplusi = ActiveCell.Offset(-(YCells + 1), (2 * XCells + 3)).Value
```

```
kright = (2 * kijplusi * kij) / (kijplusi + kij)

kijminusj = ActiveCell.Offset(-(YCells + 0), (2 * XCells + 2)).Value
kbottom = (2 * kij * kijminusj) / (kijminusj + kij)

Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
Ts = ActiveCell.Offset(-(YCells + 1), 0).Value


If Ts - Tamb = 0 Then

   qx = 0
Else

   hrx = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Ts - Tamb)
   qx = Cells(1, 13) * (Ts - Tamb) + hrx * (Ts - Tamb)

End If


If Ts - Tam = 0 Then

   qy = 0

Else
   hry = (5.77 * Cells(2, 13) * (((Ts + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Ts - Tam)
   qy = Cells(1, 13) * (Ts - Tam) + hry * (Ts - Tam)
End If


Tsj1 = ActiveCell.Offset(-(YCells + 0), 0).Value
Tsi1 = ActiveCell.Offset(-(YCells + 1), 1).Value

Pc = ActiveCell.Offset(-(YCells + 1), (XCells + 1)).Value

Tijmplus1 = ((-qx * (deltay / 2)) - (kbottom * ((Ts - Tsj1) / deltay) * (deltax / 2)) - (qy * (deltax / 2)) +
(kright * ((Tsi1 - Ts) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Ts


ActiveCell.Value = Tijmplus1

'interior cells'
ActiveCell.Offset(-(YCells + 1), 0).Select
```

```
For x = 1 To (XCells - 2)
   For y = 1 To (YCells - 2)

   ActiveCell.Offset(y, x).Select

   'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) < 500 And ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) < 500 And ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = None
   ElseIf ActiveCell > 500 Then
   ActiveCell.Interior.ColorIndex = None
   Else
   ActiveCell.Interior.ColorIndex = 4
   End If


   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kip1j = ActiveCell.Offset(0, 1).Value
   kim1j = ActiveCell.Offset(0, -1).Value
   kijp1 = ActiveCell.Offset(-1, 0).Value
   kijm1 = ActiveCell.Offset(1, 0).Value

   kleft = (2 * kij * kim1j) / (kij + kim1j)
   kright = (2 * kip1j * kij) / (kip1j + kij)
   kbottom = (2 * kij * kijm1) / (kij + kijm1)
   ktop = (2 * kij * kijp1) / (kij + kijp1)


   ActiveCell.Offset(0, 2 * -XCells - 2).Activate

   ActiveCell.Offset((YCells + 1), 0).Value = (-(kleft * ((Tij - Tim1j) / deltax) * deltay) - (kbottom * ((Tij - Tijm1) / deltay) * deltax) + (kright * ((Tip1j - Tij) / deltax) * deltay) + (ktop * ((Tijp1 - Tij) / deltay) * deltax)) * (deltat / (Pc * deltax * deltay)) + Tij

   ActiveCell.Offset(-y, -x).Select
   Next y
   Next x


'top right corner cell'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate

'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) < 500 And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate
Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijm1 = ActiveCell.Offset(1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
kbottom = (2 * kij * kijm1) / (kij + kijm1)


Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
ActiveCell.Offset((q - 2) * (YCells + 1), 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
   qy = 0
Else

   hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

End If

Tijm1 = ActiveCell.Offset(1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value


Tijmplus1 = (-(kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (kbottom * ((Tij - Tijm1) / deltay) * (deltax /
2)) - qx * deltax / 2 - qy * deltay / 2) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij



ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom right corner cell'

ActiveCell.Offset(-(YCells + 1), -(XCells - 1)).Select


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
'coloring'
```

```
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) < 500 Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate


Pc = ActiveCell.Value

ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kim1j = ActiveCell.Offset(0, -1).Value
kleft = (2 * kij * kim1j) / (kij + kim1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)



Cells(q, 2).Select
Tam = ActiveCell.Value

Cells(10, 8).Select
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(0, XCells - 1).Activate
ActiveCell.Offset(YCells - 1, 0).Activate

ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tam = 0 Then

   qx = 0
   qy = 0
Else

   hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tim1j = ActiveCell.Offset(0, -1).Value
```

```
Tijmplus1 = (-(qy * (deltax / 2)) - (kleft * ((Tij - Tim1j) / deltax) * (deltay / 2)) - (qx * (deltay / 2)) + (ktop *
((Tijp1 - Tij) / deltay) * (deltax / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij

ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1

'bottom left corner cell'

ActiveCell.Offset(-(2 * YCells), -(XCells - 1)).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



ActiveCell.Offset(YCells - 1, 0).Activate
'coloring'
   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And ActiveCell.Offset(1, 1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, 1) < 500 And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

ActiveCell.Offset(0, (XCells + 1)).Activate

Pc = ActiveCell.Value


ActiveCell.Offset(0, XCells + 1).Activate
kij = ActiveCell.Value
kijp1 = ActiveCell.Offset(-1, 0).Value
kip1j = ActiveCell.Offset(0, 1).Value
kright = (2 * kij * kip1j) / (kij + kip1j)
ktop = (2 * kij * kijp1) / (kij + kijp1)

Cells(q, 2).Select
Tam = ActiveCell.Value
Tamb = Cells(2, 2).Value

Cells(10, 8).Select
```

```
ActiveCell.Offset((q - 3) * (YCells + 1), 0).Activate
ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Offset(YCells - 1, 0).Activate
ActiveCell.Offset(-YCells - 1, 0).Activate

Tij = ActiveCell.Value

If Tij - Tamb = 0 Then
   qx = 0
Else
   hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
   qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
End If

If Tij - Tam = 0 Then
   qy = 0
Else
   hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
   qy = Cells(1, 13) * (Tij - Tam) + hry * (Tij - Tam)

End If

Tijp1 = ActiveCell.Offset(-1, 0).Value
Tip1j = ActiveCell.Offset(0, 1).Value


Tijmplus1 = (-(qx * (deltay / 2)) - (qy * (deltax / 2)) + (ktop * ((Tijp1 - Tij) / deltay) * (deltax / 2)) + (kright
* ((Tip1j - Tij) / deltax) * (deltay / 2))) * ((4 * deltat) / (Pc * deltax * deltay)) + Tij


ActiveCell.Offset(YCells + 1, 0).Activate
ActiveCell.Value = Tijmplus1



'right vertical face'

ActiveCell.Offset(-(2 * YCells), 0).Select

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1
```

```
deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(0, XCells - 1).Activate
   ActiveCell.Offset(y, 0).Activate
   'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) = "" And ActiveCell.Offset(0, -1) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(-1, 1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tim1j = ActiveCell.Offset(0, -1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tam = 0 Then

      qx = 0
      qy = 0
   Else

      hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
      qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
      qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

   End If

   ActiveCell.Offset(0, (XCells + 1)).Activate
   Pc = ActiveCell.Value

   ActiveCell.Offset(0, (XCells + 1)).Activate
   kij = ActiveCell.Value
   kim1j = ActiveCell.Offset(0, -1).Value
   kijp1 = ActiveCell.Offset(-1, 0).Value
   kijm1 = ActiveCell.Offset(1, 0).Value

   kleft = (2 * kij * kim1j) / (kij + kim1j)
   kbottom = (2 * kij * kijm1) / (kij + kijm1)
   ktop = (2 * kij * kijp1) / (kij + kijp1)

   Tijmplus1 = (-((kleft) * ((Tij - Tim1j) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij
```

```
   ActiveCell.Offset(0, 2 * -XCells - 2).Activate

   ActiveCell.Offset((YCells + 1), 0).Activate

   ActiveCell.Value = Tijmplus1

   ActiveCell.Offset(-(YCells + 1 + y), -(XCells - 1)).Select


   Next y

'left vertical face'

LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For y = 1 To (YCells - 2)

   ActiveCell.Offset(y, 0).Activate
'coloring'

   If ActiveCell < 500 And ActiveCell.Offset(0, 1) < 500 And ActiveCell.Offset(0, -1) = "" And
ActiveCell.Offset(1, -1) = "" And ActiveCell.Offset(-1, -1) = "" Then
   ActiveCell.Interior.ColorIndex = 4
   End If

   Tij = ActiveCell.Value
   Tip1j = ActiveCell.Offset(0, 1).Value
   Tijp1 = ActiveCell.Offset(-1, 0).Value
   Tijm1 = ActiveCell.Offset(1, 0).Value

   If Tij - Tamb = 0 Then
      qx = 0
   Else
      hrx = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
      qx = Cells(1, 13) * (Tij - Tamb) + hrx * (Tij - Tamb)
   End If

   If Tij - Tamb = 0 Then
```

```
        qy = 0
    Else
        hry = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tamb + 273) / 100) ^ 4)) / (Tij - Tamb)
        qy = Cells(1, 13) * (Tij - Tamb) + hry * (Tij - Tamb)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    ktop = (2 * kij * kijp1) / (kij + kijp1)

    Tijmplus1 = (((kright) * ((Tip1j - Tij) / deltax) * (deltay)) - ((kbottom) * ((Tij - Tijm1) / deltay) * (deltax /
2)) + ((ktop) * ((Tijp1 - Tij) / deltay) * (deltax / 2)) - (qx * deltay)) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij

    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1 + y), 0).Select



    Next y



'bottom face'


LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
```

```
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)



For x = 1 To (XCells - 2)

    ActiveCell.Offset(YCells - 1, 0).Activate
    ActiveCell.Offset(0, x).Activate

    'coloring'
    If ActiveCell < 500 And ActiveCell.Offset(1, 0) = "" And ActiveCell.Offset(-1, 0) < 500 And
ActiveCell.Offset(1, 1) = "" And ActiveCell.Offset(1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijp1 = ActiveCell.Offset(-1, 0).Value

    If Tij - Tam = 0 Then

        qx = 0
        qy = 0
    Else

        hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
        qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
        qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijp1 = ActiveCell.Offset(-1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    ktop = (2 * kij * kijp1) / (kij + kijp1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)
    Tijmplus1 = (((ktop) * ((Tijp1 - Tij) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay / 2)) + ((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) + Tij



    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(2 * YCells), -x).Select



    Next x



'top face'



LengthY = Cells(7, 9)
LengthX = Cells(8, 9)

deltax = Cells(1, 9)
deltay = Cells(2, 9)

YCells = LengthY / deltay + 1
XCells = LengthX / deltax + 1

deltat = Cells(3, 13)


For x = 1 To (XCells - 2)

    ActiveCell.Offset(0, x).Activate
    'coloring'
    If ActiveCell < 500 And ActiveCell.Offset(1, 0) < 500 And ActiveCell.Offset(-1, 0) = "" And ActiveCell.Offset(-1, 1) = "" And ActiveCell.Offset(-1, -1) = "" Then
    ActiveCell.Interior.ColorIndex = 4
    End If

    Tij = ActiveCell.Value
    Tip1j = ActiveCell.Offset(0, 1).Value
    Tim1j = ActiveCell.Offset(0, -1).Value
    Tijm1 = ActiveCell.Offset(1, 0).Value
    If Tij - Tam = 0 Then

        qx = 0
        qy = 0
    Else

        hr = (5.77 * Cells(2, 13) * (((Tij + 273) / 100) ^ 4 - ((Tam + 273) / 100) ^ 4)) / (Tij - Tam)
        qx = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)
        qy = Cells(1, 13) * (Tij - Tam) + hr * (Tij - Tam)

    End If

    ActiveCell.Offset(0, (XCells + 1)).Activate
    Pc = ActiveCell.Value

    ActiveCell.Offset(0, (XCells + 1)).Activate
    kij = ActiveCell.Value
    kip1j = ActiveCell.Offset(0, 1).Value
    kim1j = ActiveCell.Offset(0, -1).Value
    kijm1 = ActiveCell.Offset(1, 0).Value

    kright = (2 * kij * kip1j) / (kij + kip1j)
    kbottom = (2 * kij * kijm1) / (kij + kijm1)
    kleft = (2 * kij * kim1j) / (kij + kim1j)

    Tijmplus1 = (-(kbottom * ((Tij - Tijm1) / deltay) * (deltax)) - ((kleft) * ((Tij - Tim1j) / deltax) * (deltay /
2)) + ((kright) * ((Tip1j - Tij) / deltax) * (deltay / 2)) - qy * deltax) * ((2 * deltat) / (Pc * deltax * deltay)) +
Tij


    ActiveCell.Offset(0, 2 * -XCells - 2).Activate

    ActiveCell.Offset((YCells + 1), 0).Activate

    ActiveCell.Value = Tijmplus1

    ActiveCell.Offset(-(YCells + 1), -x).Select



    Next x

    ActiveCell.Offset((YCells + 1), 0).Activate


    With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
    .Borders(xlEdgeBottom).Weight = xlMedium
```

```
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

'properties'
   LengthY = Cells(7, 9)
   LengthX = Cells(8, 9)

   deltax = Cells(1, 9)
   deltay = Cells(2, 9)

   YCells = LengthY / deltay + 1
   XCells = LengthX / deltax + 1

   ActiveCell.Offset(0, XCells + 1).Select
   'thermal capacity'

   For x2 = 0 To (XCells - 1)
      For y = 0 To (YCells - 1)

         ActiveCell.Offset(y, x2).Select
         d = ActiveCell.Offset(0, -(XCells + 1))


            If Cells(1, 6) = "Siliceous" Then

               If d >= 0 And d <= 200 Then ActiveCell.Value = (0.005 * d + 1.7) * 10 ^ 6
               If d > 200 And d <= 400 Then ActiveCell.Value = (2.7 * 10 ^ 6)
               If d > 400 And d <= 500 Then ActiveCell.Value = (0.013 * d - 2.5) * 10 ^ 6
               If d > 500 And d <= 600 Then ActiveCell.Value = (-0.013 * d + 10.5) * 10 ^ 6
               If d > 600 Then ActiveCell.Value = 2.7 * 10 ^ 6
            End If

           If Cells(1, 6) = "Carbonate" Then
               If d >= 0 And d <= 400 Then ActiveCell.Value = 2.566 * 10 ^ 6
               If d > 400 And d <= 410 Then ActiveCell.Value = (0.1765 * d - 68.034) * 10 ^ 6
               If d > 410 And d <= 445 Then ActiveCell.Value = (-0.05043 * d + 25.00671) * 10 ^ 6
               If d > 445 And d <= 500 Then ActiveCell.Value = (2.566 * 10 ^ 6)
               If d > 500 And d <= 635 Then ActiveCell.Value = (0.01603 * d - 5.44881) * 10 ^ 6
               If d > 635 And d <= 715 Then ActiveCell.Value = (0.16635 * d - 100.90225) * 10 ^ 6
               If d > 715 And d <= 785 Then ActiveCell.Value = (-0.22103 * d + 176.07343) * 10 ^ 6
               If d > 785 Then ActiveCell.Value = (2.566 * 10 ^ 6)
            End If

         ActiveCell.Offset(-y, -x2).Select
         Next y
   Next
   With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
   .Borders(xlEdgeBottom).Weight = xlMedium
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

'thermal conductivity'

ActiveCell.Offset(0, XCells + 1).Select


For x2 = 0 To (XCells - 1)
      For y = 0 To (YCells - 1)

         ActiveCell.Offset(y, x2).Select
         d = ActiveCell.Offset(0, -2 * (XCells + 1))


            If Cells(1, 6) = "Siliceous" Then
               If d >= 0 And d <= 800 Then ActiveCell.Value = (-0.000625 * d + 1.5)
               If d > 800 Then ActiveCell.Value = 1
            End If

            If Cells(1, 6) = "Carbonate" Then
               If d >= 0 And d <= 293 Then ActiveCell.Value = 1.355
               If d > 293 Then ActiveCell.Value = (-0.001241 * d + 1.7162)
            End If

          ActiveCell.Offset(-y, -x2).Select
         Next y


Next

   With Range(ActiveCell, ActiveCell.Offset(YCells - 1, XCells - 1))
   .Borders(xlEdgeBottom).Weight = xlMedium
   .Borders(xlEdgeTop).Weight = xlMedium
   .Borders(xlEdgeLeft).Weight = xlMedium
   .Borders(xlEdgeRight).Weight = xlMedium
   End With

ActiveCell.Offset(0, -2 * (XCells + 1)).Select


Next q
```

```
For u = 0 To YCells

    Rows(ActiveCell.Offset(u, 0).Row).Clear
    Next u

End Sub
```