NTNU
Norwegian University of
Science and Technology

Time Constant Considerations for a Simplified Charcoal Fired Furnace

Ingelin Garmann

Chemical Engineering and Biotechnology
Submission date: June 2017
Supervisor: Tore Haug-Warberg, IKP

Norwegian University of Science and Technology
Department of Chemical Engineering

Preface

This report represents my master thesis written for the Department of Chemical Engineering at NTNU as part of the study program Industrial Chemistry and Biotechnology. The master thesis was written during the spring semester of 2017. The assignment was supervised by Professor Tore Haug-Warberg from the Process Systems Engineering group.

The thesis involves partial differential equations, numerical approximation to first and second order derivatives, matrix algebra, eigenvalues and eigenvectors, Laplace transformation, chemical reaction and time constant considerations. The formulations in the report are directed to those with an understanding of the above mentioned fields, such as fellow engineering students. That being said, most calculations and derivations are thoroughly explained, and the results can be interpreted without a deep understanding of the mathematical derivations.

Trondheim, 12. juni 2017

Ingelin Garmann

Ingelin Garmann

Acknowledgment

The greatest appreciation should be directed to my supervisor Professor Tore Haug-Warberg for giving me the opportunity to work on such a challenging and complex problem during my last year at NTNU. The description of the physical scenario being investigated in this thesis can be understood by any layman, but the formulation and deeper explanation, from a scientific point of view, demands deeper understanding into a wide range of areas. "Forgotten" knowledge from the basic subjects I have been taught during my years at NTNU has been dug back up and utilized in the process of understanding and formulate a solution to the problem at hand. The most valuable lesson acquired from my cooperation with Professor Tore Haug-Warberg is how to procede when faced with a complex and difficult problem. His method of simplifying the complex generic into the solvable specific is a strategy I once hope to master.

I. G

Summary

A mathematical model for a spherical control volume containing a combustion chamber buried in moist sand was derived based on the thermodynamic laws for heat transfer. The combustion chamber holds an iron core and a carbon-containing furnace enclosed by a steel wall. Temperature models were obtained both for coal combustion in the furnace, and for pure conduction of heat in a multilayered material. The heat generated by the furnace would cause the moisture in the surrounding sand to evaporate. A moving boundary condition was implemented account for the increasing layer of dry sand surrounding the furnace. The requirement for the increasement of the boundary was complete evaporation of the moisture content in the surrounding spherical shell. The differential equations describing heat conduction were discretized by central finite differences and rearranged to yield a first order linear differential equation system with respect to time. The non-perfect heat transfer over interfaces were accounted for by the thermal contact conductance coefficient. The complete equation system for a fixed number of temperature nodes, were formulated on the general linear time invariant form, and solved by the application of eigenvalues and eigenvector. The temperature evolution in the burning furnace was derived from the energy balance, and also formulated into a matrix differential system. The numerical models were implemented in MATLAB. The temperature profiles for combustion of coal in a furnace was obtained for different amounts of reactants, for the control volume holding an initial temperature of 373.15 K. The obtained temperature evolution at the different nodes coincide with what is expected of a heat generating process. The solution process was repeated for conductive heat transfer through the materials, with the temperatures of the final heating stage as initial condition. This is only presented for one amount of limiting reactant, due to the small effect on the conductive process. The evaporation times and transient temperature evolutions were considered and discussed in an attempt to determine a characteristic time constant for the process of heating and cooling of the system.

Contents

Chapter 1

Introduction

1.1 Background

The background and motivation for the thesis stems from the ancient art of iron smelting. The following paragraph, written by Professor Tore Haug-Warberg, describes the background for the assignment:

The archeology of ancient iron smelters in Norway does for the most part focus on a classification of the smelting sites according to the morphology of the exterior of the furnace. This is adequate information for understanding the smelting activity in a social context, but it sheds no light on how the furnaces would have been operated. In an attempt to understand the ancient smelters we could try to understand their operational characteristics from a natural science perspective. However, it quickly turns out that the old technologies are far from being primitive. They were in fact surprisingly well tuned and even in the light of our modern understanding of physics, chemistry and metallurgy there are several techniques connected to the manufacture of steels which today is lost art. The problem is to find where to start the scrutiny for an increased understanding of ancient technology. Theoretically, challenging questions are quickly piling up, ranging from how aggressive slags react with clays, to the metallurgy of sintered iron and the use of slag-metal composites for weaponry. However, maybe the simplest question to be answered is what the time constant of the furnace was (and still is). The furnaces usually had very crude shapes and would be hard to model to any degree of satisfaction, but since we are asking only for the (characteristic) time constant of the furnace (as a function of the size) we can do a very crude model and still get valuable insight. In this context it is of interest to calculate the time constant of a hypothetically spherical furnace buried in (damp) sand or soil to see how it varies with the size of the furnace, in order to achieve a better understanding of the operational characteristics of the furnace, i.e. the time spent on one complete cycle from the heating up of the furnace until extracting the iron bloom.

1.2 Objectives

The objectives of the thesis are:

1. Discretize the heat conduction equation by central finite difference equations in the radial direction
2. Describe the equations as a linear time-invariant equation system on the form $\dot{x} = Ax - b$ with A being a matrix containing constant coefficients
3. Solve the matrix system in time by the use of eigenvalues and eigenvectors
4. Implement a numerical solution to the matrix system in MATLAB, including the moving boundary condition
5. Formulate and implement a solution that includes combustion of coal in the furnace
6. Calculate a characteristic time for the system being heated and for the system cooling down

1.3 Limitations

The solutions obtained in this report are only valid for this particular discretization of the control volume. Not much attention has been put into finding correct material properties, and all the properties are assumed to be constant over the temperature range in the problem. The combustion of coal in the furnace is a severly simplified model. The thermal properties are assumed independent of temperature, and the only reaction taking place in the furnace is combustion of coal to carbon monoxide. Due to the variable substitution relating the spherical and the slab conduction equation, the temperature at r=0 is undefined. Thus, the temperature in the center is only an extrapolated value. The numeric results obtained in the report are unvalidated by experiments or analytical results. Therefore, the most valuable results from this report are the method and derivation of the solutions.

1.4 Approach

As part of the description and background information provided by Professor Tore Haug-Warberg, a general matrix system discretization of the one dimensional heat conduction equation for a slab was included. This became the foundation for the solution methodology of the task. A variable substitution relating a one dimensional slab to a one dimensional sphere was applied to make the method valid for this particular problem. Complexity, in form of multiple layers and combustion in the furnace, was added gradually.

A great many hours were spent on implementing code in MATLAB, to solve the temperature evolution of the discretized nodes of the problem. The code behind the solution is attached in appendix B. The solution consists of a *main*-script where the discretization of temperature and radial position of the different nodes are defined, where figures are generated, functions are called and output tables are written to files. The main-script is constructed so that it can solve for both the case of a burning furnace and for the case of pure conduction of heat, but also for cases where the furnace is switched on and off, determined by the user. The constant material properties for all the different materials are defined in the function *matProperties*. To create the matrix system for solution of the differential equations for the burning furnace, the function *reactionFurnace* calculates the coefficients for the temperature evolution in the iron core and in the furnace, then the *constructFurnace*-function generates the A_1-, A_2-, B_1-matrices and the b_1-vector according to the derivation in 7.2. Equivalently, the *constructConduction*-function generates the A_1-, A_2-, B_1-matrices for the case of a non-burning furnace. In *constructMatrixA*, the matrix system is rearranged to the $\dot{x} = Ax - b$, and the eigenvalues and eigenvectors of A are computed. The solutions for the incremental shell size is found by looping over the moist radial nodes. The heat flux into the moist sand and the required evaporation time is calculated by use of symbolic variables in the *determineVaptime*-function. Thereafter, the temperature evolution of the inner nodes and the ghost nodes are solved by use of eigenvalues and eigenvectors in the *nodeSolve*- and the *ghostSolve*-functions. The matrix system is increased by one temperature node in the function called *matrixIncrease*. The functions *rtransf* and *backtransf* are used to switch between the temperature definition in the spherical equations and the slab equations.

Chapter 2

Description of the control volume

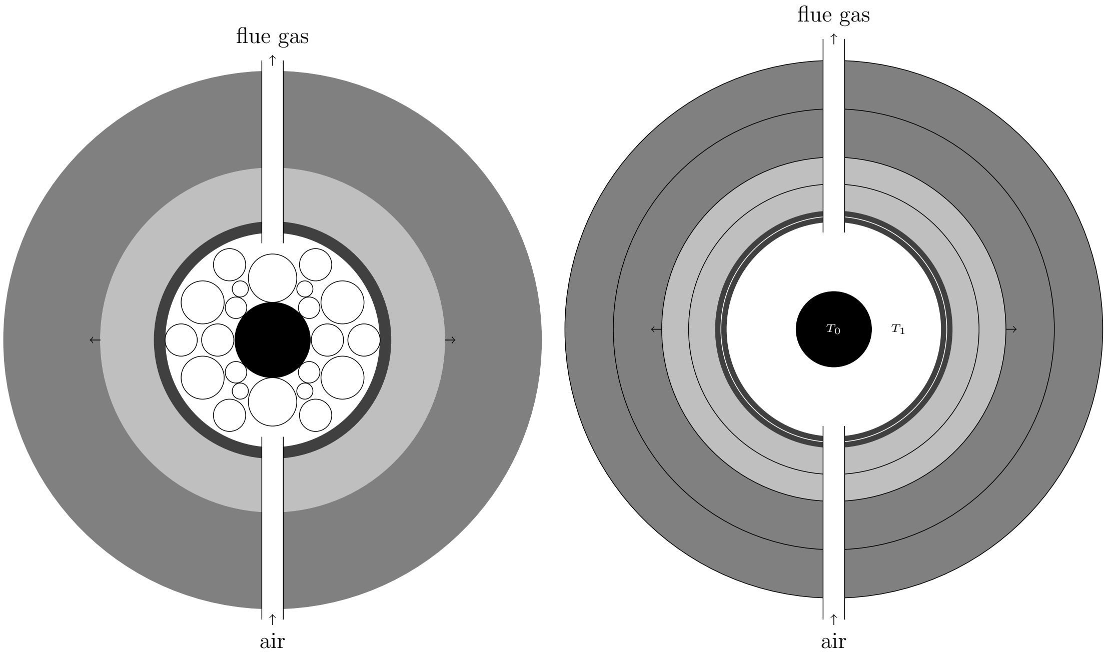

Figure 2.1: An illustration of the cross section of the sphere, both without and with temperature discretization. There is an iron core, inside the coal-containing furnace separated from the surroundings by a metal wall. The light gray area contains a layer of dry sand which will expand as the moisture in the outer sand layer evaporates due to heat transfer.

The control volume is limited to a sphere with outer radius equal to 1 m. The different layers of the sphere, consisting of different materials, also have their individual thickness. The interfaces are listed in table 2.1 and are considered constant on the inside of the dry sand. The combustion chamber, including the iron core, the furnace and the steel wall, is assumed to resemble the size of a soccer ball, of approximately 15 cm. The chamber is submerged in a bed of sand holding 373.15 K, and containing 1vol% of water. Combustion of coal, modelled as carbon, takes place in the furnace and produces heat that raises the temperature in the control volume. In figure 2.1 the cross section of the sphere is shown. To the left, the layers of different materials are illustrated by different colors. The black core consists of iron, the next layer is the furnace filled with carbon and enclosed by a steel wall illustrated in dark gray. The initial thin layer of dry sand is given in light gray. The arrows indicates that this layer will expand as heat is transferred through the materials causing the water in the moist sand, illustrated by gray color, to evaporate. To the right, the different layers are divided so that there are 3 nodes to compute the temperature in each material. This is the minimum number of nodes able to account for the second order nature of the derivative. For combustion in the furnace, the iron core and the furnace temperature is assumed to be uniform. For conduction, these layers will also be divided into temperature nodes. The equations describing the different materials, as well as the interface and the moving boundary condition are derived in the following sections.

Material	Outer radius [m]	Thickness [cm]
Iron	0.06	6
Carbon furnace	0.12	6
Steel wall	0.15	2
Dry sand	0.25	10

Table 2.1: The interface radial positions of the control volume. The outer radius for the dry sand layer is only valid for the initial situation, as it will expand to 1 m.

Chapter 3

Thermodynamic Laws

When there exists two regions holding different temperatures, heat will flow from the warmer region to the colder. That is the essence of the second law of thermodynamics. Heat transferring phenomena are divided into three categories: conduction, convection and radiation. Radiation is neglected in the spherical model. The first law of thermodynamics states that the energy of an isolated system is constant. It can be transformed from one form to another, but not created nor destroyed. That defines the principle of conservation of energy. By considering these basic thermodynamic concepts, the phenomena taking place in the spherical geometry can be described and investigated.

3.1 Conservation of energy

The 1^{st} law of thermodynamics states that the change of energy, $\frac{dE}{dt}$ of a system equals the sum of heat supplied and work done to the system.

$$\frac{dE}{dt} + \nabla \cdot (E\mathbf{u}) = \frac{\partial Q}{\partial t} - \nabla \cdot q + \rho \mathbf{f} \cdot \mathbf{u} + \nabla \cdot (\mathbf{\Pi_{ij}} \cdot \mathbf{u}) \tag{3.1}$$

where E is the total energy of the control volume including internal energy, U, kinetic energy, K, potential energy, P, etc. The transient change of energy and the change of energy due to advection, $\nabla \cdot (E\mathbf{u})$ equals the heat produced in the control volume, $\frac{\partial Q}{\partial t}$, the energy change due to conduction, $\nabla \cdot q$, work done on the fluid by external forces, $\rho\mathbf{f} \cdot \mathbf{u}$, and work done by surface forces, $\nabla \cdot (\mathbf{\Pi_{ij}} \cdot \mathbf{u})$.

3.2 Heat Transfer

For a solid material it is appropriate to assume no advection, since advection is transport due to bulk fluid motion. By also neglecting any work done on the system, the energy equation simplifies to:

$$\frac{\partial E}{\partial t} = \frac{\partial Q}{\partial t} - \nabla \cdot q \tag{3.2}$$

3.2.1 Specific heat capacities

For the spherical control volume, the energy change of interest is the internal energy, U. For internal energy described by temperature and volume, $U = U(T, V)$, the following relation exists:

$$dU = \left(\frac{\partial U}{\partial T}\right)_V dT + \left(\frac{\partial U}{\partial V}\right)_T dV \tag{3.3}$$

For a constant control volume $dV = 0$ and the second term vanishes. Specific heat capacity at constant volume is defined by $C_V = \left(\frac{\partial U}{\partial T}\right)_V$. Thus, the transient change of internal energy, for constant specific heat, becomes:

$$\frac{dU}{dt} = C_V \frac{dT}{dt} = mc_V \frac{dT}{dt} \tag{3.4}$$

The heat content of a thermodynamic system can more accurately be described in terms of enthalpy, H. The enthalpy accounts both for internal energy required to create the system, and also the energy required to make room for the system:

$$H = U + pV \tag{3.5}$$

where the product of pressure, p and volume, V, relates enthalpy to internal energy.

For enthalpy described by temperature and pressure, $H = H(T, p)$, the following is true:

$$dH = \left(\frac{\partial H}{\partial T}\right)_p dT + \left(\frac{\partial H}{\partial p}\right)_T dp \tag{3.6}$$

For constant pressure $dp = 0$ processes, the second term disappears, and by the definition of specific heat capacity at constant pressure $C_p = \left(\frac{\partial H}{\partial T}\right)_p$, the transient change in enthalpy for constant heat capacity is:

$$\frac{dH}{dt} = C_p \frac{dT}{dt} = mc_p \frac{dT}{dt} \tag{3.7}$$

For a solid, the distinction between specific heat at constant volume and at constant pressure are often neglected.

3.2.2 Heat flux by conduction

The dominant form of heat transfer occurring in a solid is through conduction, in which heat passes through the material itself. The conductive heat flux, q_{cond}, is given by Fourier's law:

$$q_{cond} = -k\nabla T \tag{3.8}$$

where k is thermal conductivity. In one dimension, the law becomes:

$$q_{cond} = -k\frac{\partial T}{\partial x} \tag{3.9}$$

Inserting this expression for heat flux into the energy balance in equation 3.2, pure conduction in one dimension is described by:

$$\rho c_p \frac{\partial T}{\partial t} = -\frac{\partial}{\partial x} \cdot (-k\frac{\partial T}{\partial x}) = k\frac{\partial^2 T}{\partial x^2} \tag{3.10}$$

for constant thermal conductivity, k.

3.2.3 Heat flux over an interface

In solid materials convection is absent, because it requires relative motion of parts of the heat bearing media. But when two solid materials are in contact, there exists a resistance to heat transfer modelled in the same way as convection. The thermal contact conductance coefficient, h_c, accounts for the non-perfect thermal contact at an interface. The heat flux at an interface is given by:

$$q_{cont} = h_c A \Delta T \tag{3.11}$$

Over an interface, the following equation, as stated by M.G. Cooper and Yovanovich [1968], describes the heat flux with thermal contact resistance:

$$-k\frac{\partial T}{\partial x} = h_c \Delta T \tag{3.12}$$

This equation is explained in more detail in the next chapter 4.3.

Chapter 4

Numerical discretization

4.1 Finite differences

To obtain an accuracy of second order in both first and second order space derivatives, the central finite difference is used. The central difference is obtained from manipulation of the forward and the backward differences.

Forward difference in space:

$$T(x_i + \Delta x) = T(x_i) + \Delta x \left.\frac{\partial T}{\partial x}\right|_{x=x_i} + \frac{\Delta x^2}{2} \left.\frac{\partial^2 T}{\partial x^2}\right|_{x=x_i} + O(\Delta x^3) \tag{4.1}$$

Backward difference in space:

$$T(x_i - \Delta x) = T(x_i) - \Delta x \left.\frac{\partial T}{\partial x}\right|_{x=x_i} + \frac{\Delta x^2}{2} \left.\frac{\partial^2 T}{\partial x^2}\right|_{x=x_i} + O(\Delta x^3) \tag{4.2}$$

The central difference for the first order derivative is found by subtracting the backward difference from the forward difference:

$$T(x_i + \Delta x) - T(x_i - \Delta x) = 2\Delta x \left.\frac{\partial T}{\partial x}\right|_{x=x_i} + O(\Delta x^3) \tag{4.3}$$

Rearranging, the following difference equation is obtained for the first order derivative at $x = x_i$:

$$\left.\frac{\partial T}{\partial x}\right|_{x=x_i} = \frac{T_{i+1} - T_{i-1}}{2\Delta x} + O(\Delta x^2) \tag{4.4}$$

The central difference for the second order derivative is found by adding the forward and the backward difference:

$$T(x_i + \Delta x) + T(x_i - \Delta x) = 2T(x_i) + \Delta x^2 \left.\frac{\partial^2 T}{\partial x^2}\right|_{x=x_i} + O(\Delta x^4) \tag{4.5}$$

At $x = x_i$, the following difference equation approximates the second order derivative:

$$\left.\frac{\partial^2 T}{\partial x^2}\right|_{x=x_i} = \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} + O(\Delta x^2) \tag{4.6}$$

4.2 Discretization of the heat conduction equation

The equation describing the temperature evolution due to pure conduction was derived in the previous chapter, equation 3.10, for a three dimensional control volume. For a one dimensional slab, the conduction equation is given by:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{4.7}$$

where T is the temperature in the slab and α is thermal diffusivity given by $\alpha = \frac{k}{\rho c_p}$

The term on the right hand side is a second order differential in space, and can be discretized by the second order difference equation given in equation 4.6:

$$\dot{T}_i = \alpha \frac{T_{i-1} - 2T_i + T_{i+1}}{\Delta x^2} \tag{4.8}$$

The one dimensional conduction equation for a sphere is given by:

$$\frac{\partial V}{\partial t} = \alpha \left(\frac{2}{r}\frac{\partial V}{\partial r} + \frac{\partial^2 V}{\partial r^2} \right) \tag{4.9}$$

where V is the temperature in the sphere.

The temperature in the equation for the slab and the sphere is related by the radius, and thus the radial-dependent spherical equation can be transformed into the slab equation by a variable substitution:

$$T(r,t) = rV(r,t) \tag{4.10}$$

Differentiating equation 4.10 gives:

$$V = \frac{T}{r} \tag{4.11}$$

$$\frac{\partial V}{\partial t} = \frac{1}{r}\frac{\partial T}{\partial t} \tag{4.12}$$

$$\frac{\partial V}{\partial r} = -\frac{T}{r^2} + \frac{1}{r}\frac{\partial T}{\partial r} \tag{4.13}$$

$$\frac{\partial^2 V}{\partial r^2} = \frac{2T}{r^3} - \frac{2}{r^2}\frac{\partial T}{\partial r} + \frac{1}{r}\frac{\partial^2 T}{\partial r^2} \tag{4.14}$$

Inserting equation 4.13 and equation 4.14 into the conduction equation for the sphere gives:

$$\frac{\partial V}{\partial t} = \alpha \left[\frac{2}{r}\frac{\partial V}{\partial r} + \frac{\partial^2 V}{\partial r^2} \right] \tag{4.15}$$

$$\frac{1}{r}\frac{\partial T}{\partial t} = \alpha \left[\frac{2}{r}\left(-\frac{T}{r^2} + \frac{1}{r}\frac{\partial T}{\partial r} \right) + \frac{2T}{r^3} - \frac{2}{r^2}\frac{\partial T}{\partial r} + \frac{1}{r}\frac{\partial^2 T}{\partial r^2} \right] \tag{4.16}$$

$$\frac{1}{r}\frac{\partial T}{\partial t} = \alpha \left[-\frac{2T}{r^3} + \frac{2}{r^2}\frac{\partial T}{\partial r} + \frac{2T}{r^3} - \frac{2}{r^2}\frac{\partial T}{\partial r} + \frac{1}{r}\frac{\partial^2 T}{\partial r^2} \right] \tag{4.17}$$

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial r^2} \tag{4.18}$$

which yields the conduction equation for the slab, and the discretization in equation 4.8 is valid when the temperature in the sphere is given by equation 4.10. The temperature V in the slab is obtained by back-substitution according to equation 4.12.

4.3 Discretization of interface temperatures between two layers

On the interface where two materials are in contact, the conductive heat flux leaving one material and entering the next material is reduced because of a resistance to heat transfer. The interface, and the temperature nodes around the interface, are illustrated in figure 4.1.

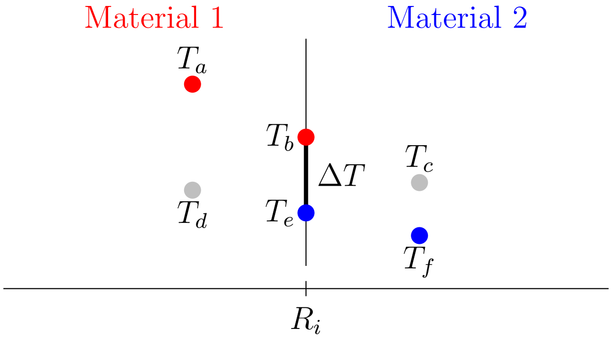

Figure 4.1: Illustration of heat transfer across an interface located at R_i separating material 1 with inner temperature nodes T_a and T_b and ghost node T_c, and material 2 with ghost node T_d and inner temperature nodes T_e and T_f.

The equation for conductive heat flux is given by equation 3.9, and the contact resistance on the interface is given by equation 3.11. The derivative in equation 3.9 was discretized by a central difference, and both equations were applied the temperature variable substitution given by equation 4.10. In accordance with the labelling in figure 4.1, the equations for the temperature in the two materials over the interface becomes:

$$-k\left(\frac{\frac{T_c}{r_c}-\frac{T_a}{r_a}}{2\Delta r}\right)=h\left(\frac{T_b}{R_i}-\frac{T_e}{R_i}\right) \tag{4.19}$$

$$h\left(\frac{T_b}{R_i}-\frac{T_e}{R_i}\right)=-k\left(\frac{\frac{T_f}{r_f}-\frac{T_d}{r_d}}{2\Delta r}\right) \tag{4.20}$$

with a little rearrangement the equations can be written as:

$$\frac{r_c}{r_a}T_a-\frac{2hr_c\Delta r}{kR_i}T_b-T_c+\frac{2hr_c\Delta r}{kR_i}T_e=0 \tag{4.21}$$

$$\frac{2hr_d\Delta r}{kR_i}T_b-T_d-\frac{2hr_d\Delta r}{kR_i}T_e+\frac{r_d}{r_f}T_f=0 \tag{4.22}$$

For the equations concerning the interfaces listed in the section on multilayer heat transfer in the sphere, all the interface conditions are modelled according to equation 4.21 and 4.22.

Chapter 5

Combustion of coal

The combustion process of burning coal in air is greatly simplified in this model. The initial ignition of the coal is neglected, and it is assumed that coal is fed continously in counterflow to the exiting flue gas. It is assumed that the amount of coal is constant and holds the correct burning temperature at all times. Air is fed to the furnace at outside temperature. The only reaction taking place is assumed to be combustion of coal in air to produce carbon monoxide. The stoichiometric balanced reaction is:

$$2C(s) + O_2(g) + 3.728N_2(g) \rightarrow 2CO(g) + 3.728N_2(g) \tag{5.1}$$

Air is containing 20.946% O_2 and 78.084% N_2, resulting in a stoichiometry of 3.728 moles of N_2 per moles of O_2. The flue gas contains carbon monoxide and nitrogen at furnace temperature. The combustion is assumed to take place under constant pressure, and the specific heat capacities of the different substances are assumed to be constant with respect to temperature. Due to carbon being present and in abundance at all times, oxygen gas is the limiting reactant for the reaction. The following table is an excerpt of a selection of specific heats for the reactants and products in the combustion, taken from the Yanaf Thermochemical Tables:

Table 5.1: Specific heat capacities at constant pressure taken from The Yanaf Thermochemical Tables.

T	$c_{p,C}$ (s)	c_{p,O_2} (g)	c_{p,N_2} (g)	$c_{p,CO}$ (g)
K	$\text{Jmol}^{-1}\text{K}^{-1}$			
298.15	8.517	29.376	29.123	29.142
500	14.623	31.091	29.583	29.794
1000	21.610	34.870	32.696	33.138
1200	22.766	35.667	33.723	34.175
1500	23.904	36.544	34.844	35.217

5.1 Adiabatic flame temperature

The adiabatic flame temperature is the maximum reachable temperature for the products when no heat is lost to the environment. The calculation is made for complete combustion. For the reaction in equation 7.7 the following steps describe the process:

1. Heating of reactants, O_2 and N_2, from $T_{air} = 288$ K to T_{ref}
2. Cooling of carbon (coal) from T to T_{ref}
3. Reaction at T_{ref}
4. Heating of the products, CO and N_2, from T_{ref} to T

If choosing the reference temperature, $T_{ref} = 298.15$ K, the standard enthalpies of formation for elements in their standard state are defined to be zero. Due to the reactants entering the furnace at a temperature lower than the one in which reaction is taking place, the reactants require heating. By using the specific heat capacity for 298.15 K, the required heat becomes:

$$\begin{aligned} q_1 &= c_{p,O_2}(T_{air} - T_{ref}) + 3.728 c_{p,N_2}(T_{air} - T_{ref}) + 2c_{p,C}(T - T_{ref}) \\ q_1 &= -1379.5 \text{ J} \end{aligned} \tag{5.2}$$

Carbon is available inside the furnace and holds a temperature higher than the reference, and will need to be cooled. The specific heat is chosen at 1000 K:

$$\begin{aligned} q_2 &= 2c_{p,C}(T - T_{ref}) \\ q_2 &= 43.22(T - T_{ref}) \text{ JK}^{-1} \end{aligned} \tag{5.3}$$

For reaction at reference conditions, the enthalpy change is the difference between standard enthalpies of formation for the products and the reactants. Since all the components, except carbon monoxide, are present in their elemental state the reaction enthalpy becomes the enthalpy of formation for CO:

$$-q_3 = 2h^{\ominus}_{f,CO} \tag{5.4}$$

$$q_3 = 221054 \text{ J} \tag{5.5}$$

The sum of q_1, q_2 and q_3 is the energy available to raise the temperature of the products. By assuming that the specific heat values at 1000 K holds for the temperature range, the calculation becomes:

$$\left(2c_{p,CO} + 3.728 c_{p,N_2}\right)(T - T_{ref}) = q_1 + q_2 + q_3 \tag{5.6}$$

$$\begin{aligned} T &= T_{ref} + \frac{q_1 + q_3}{2c_{p,CO} + 3.728 c_{p,N_2} - 2c_{p,C}} \\ T &= 298.15 + 1514.6 = 1812 \text{ K} \end{aligned} \tag{5.7}$$

For complete, adiabatic combustion of carbon at the adiabatic temperature and air at 288 K, the maximum reachable temperature is 1812 K. In practical applications, heat will be lost to the surroundings and combustion will rarely go to completion. For the special furnace in this particular problem, there will also be carbon present after "full" combustion. This material would also require heat to reach the adiabatic flame temperature, and thus the maximum temperature calculated above will never be reached by the spherical furnace. The adiabatic temperature is therefore only theoretical, but at least it gives a measure of what temperatures the furnace is this problem should not be obtaining.

5.2 Amount of oxygen in inlet feed

The amount of oxygen gas in the inlet air stream will be the limiting reactant of the combustion reaction. The amount of oxygen fed to the furnace is determined so that the reaction produces 10 000 W when operated stationary. The energy balance for steady state heat production becomes:

$$\dot{Q} = \dot{H}_{out} - \dot{H}_{in} \tag{5.8}$$

The calculation is made for an operational furnace temperature of 1200 K, outside air temperature of 288 K and a reference state of 298.15 K. The enthalpy of the feed is:

$$\dot{H}_{in} = \dot{n}_{O_2} c_{p,O_2}(T_{in} - T_{ref}) + 3.728\dot{n}_{O_2} c_{p,N_2}(T_{in} - T_{ref}) \tag{5.9}$$

where $\dot{n}_{O_2}$ is the flow of oxygen gas in moles per seconds.

The enthalpy of the flue gas is:

$$\dot{H}_{out} = 2\dot{n}_{O_2}\left[h^{\ominus}_{f,CO} + c_{p,CO}(T_{out} - T_{ref})\right] + 3.728\dot{n}_{O_2} c_{p,N_2}(T_{out} - T_{ref}) \tag{5.10}$$

Inserting the enthalpies into equation 5.8 and solving for $\dot{n}_{O_2}$ gives:

$$\dot{n}_{O_2} = \frac{Q}{2h^{\ominus}_{f,CO} + (2c_{p,CO} + 3.728c_{p,N_2})(T_{out} - T_{ref}) - (c_{p,O_2} + 3.728c_{p,N_2})(T_{in} - T_{ref})}$$

By assuming the specific heat capacites given at 1000 K holds for the temperature range, the solution to the equation is:

$$\dot{n}_{O_2} = 0.1898 \text{ mol s}^{-1} \tag{5.11}$$

$$\dot{m}_{O_2} = \dot{n}_{O_2} M_{m,O_2} = 0.006 \text{ kg s}^{-1} \tag{5.12}$$

$$\dot{m}_{air} = \frac{\dot{m}_{O_2}}{0.20946} = 0.029 \text{ kg s}^{-1} \tag{5.13}$$

An inlet air stream of 0.2 moles per second, or equivalently 0.03 kg per second, or 1.8 kg per minute, gives the desired heat production of 10 kW.

Chapter 6

Matrix equation system

To find the temperature profile in the spherical geometry, supervisor Tore Haug-Warberg proposed a method involving representation by a matrix equation system and solution of the time derivates by use of eigenvalues and eigenvectors. The proposition included a numerical discretization stencil of inner nodes as well as ghost nodes to yield the desired interface and boundary conditions. The discretization stencil for a two-layer one-dimensional slab is illustrated in figure 6.1. The corresponding matrix system is given in figure 6.16. A more detailed derivation of the equations, as well as the eigenvalue/eigenvector solution method, follows in the next section.

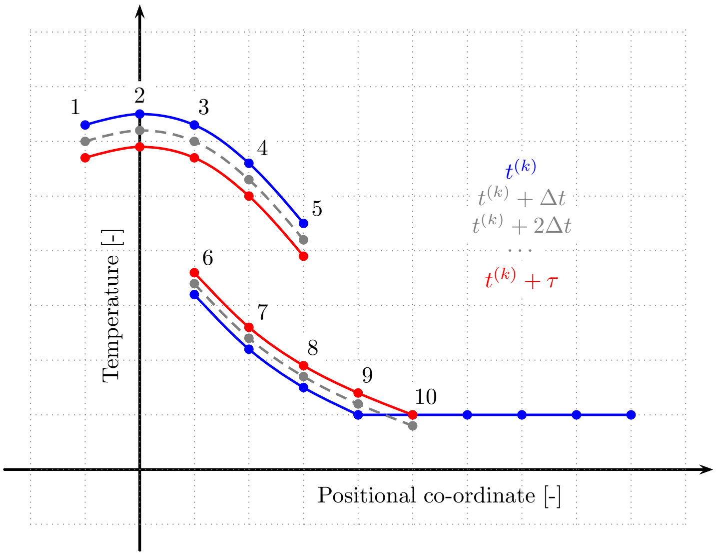

Figure 6.1: Stencil for the temperature discretization of a slab of two materials holding different temperatures.

6.1 Heat equation for a slab

For simplicity, the slab in this illustration of the method consists of one material, but with different initial temperatures, $T_0(x)$, depending on position. Heat transfer will occur by conduction in the inner nodes of the materials, given by equation 6.1.

Heat conduction equation in x-direction:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2} \tag{6.1}$$

Initial condition:

$$T(x,0) = T_0(x), \quad t = 0 \tag{6.2}$$

The heat flux over the boundaries and over the interface are given by Fourier's and Newton's Law. The left end of the slab, at $x = 0$, is insulated and thus there is zero heat flux crossing this boundary. At the interface, at $x = a$, the conductive and convective heat flux must be equal. At the right boundary, at $x = L$, the heat flux to the surroundings is denoted by q.

The equations for the boundary and interface conditions are:

$$-k\frac{\partial T}{\partial x} = 0, \quad x = 0 \tag{6.3}$$

$$-k\frac{\partial T}{\partial x} = h\Delta T, \quad x = \mathrm{a}^- \tag{6.4}$$

$$-k\frac{\partial T}{\partial x} = h\Delta T, \quad x = \mathrm{a}^+ \tag{6.5}$$

$$-k\frac{\partial T}{\partial x} = q, \quad x = L \tag{6.6}$$

Discretization by central finite differences, given by equation 4.4 and 4.6, gives for inner nodes:

$$\dot{T_i} = T_{i-1} - 2T_i + T_{i+1} \tag{6.7}$$

where α and Δx has been set equal to one.

And for boundary and interface nodes:

$$-k\frac{T_3 - T_1}{2\Delta x} = 0 \tag{6.8}$$

$$-k\frac{T_5 - T_3}{2\Delta x} - h(T_4 - T_7) = 0 \tag{6.9}$$

$$-k\frac{T_8 - T_6}{2\Delta x} - h(T_4 - T_7) = 0 \tag{6.10}$$

$$-k\frac{T_{10} - T_8}{2\Delta x} = q \tag{6.11}$$

Rearranged and for $\Delta x = 1$:

$$-T_1 + T_3 = 0 \tag{6.12}$$

$$T_3 - mT_4 - T_5 + mT_7 = 0 \tag{6.13}$$

$$mT_4 - T_6 - mT_7 + T_8 = 0 \tag{6.14}$$

$$T_8 - T_{10} = n_q \tag{6.15}$$

where $m = \frac{2h}{k}$ and $n_q = \frac{2q}{k}$.

In the next section, the equations for all the nodes will be structured in a matrix form.

6.2 Transformation to matrix system

For the special case of the slab described above, the matrix representation of the differential equations becomes:

$$\begin{bmatrix} -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -m & -1 & 0 & m & 0 & 0 & 0 \\ 0 & 0 & 0 & m & 0 & -1 & -m & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_4 \\ T_5 \\ T_6 \\ T_7 \\ T_8 \\ T_9 \\ T_{10} \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{T}_2 \\ \dot{T}_3 \\ \dot{T}_4 \\ 0 \\ 0 \\ \dot{T}_7 \\ \dot{T}_8 \\ \dot{T}_9 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ n_q \end{bmatrix} \tag{6.16}$$

This system is on the form $Ax = \dot{x} + b$.

The matrix system can be separated into one system for inner nodes and another system for boundary nodes:

$$\begin{bmatrix} -2 & 1 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 1 & -2 \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_7 \\ T_8 \\ T_9 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_5 \\ T_6 \\ T_{10} \end{bmatrix} = \begin{bmatrix} \dot{T}_2 \\ \dot{T}_3 \\ \dot{T}_4 \\ \dot{T}_7 \\ \dot{T}_8 \\ \dot{T}_9 \end{bmatrix} \tag{6.17}$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -m & m & 0 & 0 \\ 0 & 0 & m & -m & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} T_2 \\ T_3 \\ T_4 \\ T_7 \\ T_8 \\ T_9 \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} T_1 \\ T_5 \\ T_6 \\ T_{10} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ n_q \end{bmatrix} \tag{6.18}$$

6.3 Solution by use of eigenvalues and eigenvectors

The system can be written as:

$$A_1 x_1 + A_2 x_2 = \dot{x_1} \tag{6.19}$$

$$B_1 x_1 + B_2 x_2 = b_2 \tag{6.20}$$

where $B_2 = -I$. Solving for x_2 in equation 6.20 gives:

$$x_2 = B_1 x_1 - b_2 \tag{6.21}$$

Inserting x_2 into equation 6.19 gives:

$$A_1 x_1 + A_2(B_1 x_1 - b_2) = \dot{x_1} \tag{6.22}$$

$$(A_1 + A_2 B_1)x_1 - A_2 b_2 = \dot{x_1} \tag{6.23}$$

which, once again, is on the simple form:

$$\dot{x_1} = A x_1 - b \tag{6.24}$$

with:

$$A = (A_1 + A_2 B_1) \tag{6.25}$$

$$b = A_2 b_2 \tag{6.26}$$

A system on this form can be solved using the eigenvalue matrix, Λ, and eigenvector matrix, S, of the A-matrix. The relation between the system matrix, A, and its eigenvalues and eigenvectors is:

$$A = S \Lambda S^{-1} \tag{6.27}$$

Inserting equation 6.27 for A in equation 6.24, the system is transformed into eigenvalue-form:

$$\dot{x_1} = S \Lambda S^{-1} x_1 - b \tag{6.28}$$

$$S^{-1}\dot{x_1} = \Lambda S^{-1} x_1 - S^{-1} b \tag{6.29}$$

where new variables $y = S^{-1}x_1$ and $c = S^{-1}b$ are introduced giving:

$$\dot{y} = \Lambda y - c \tag{6.30}$$

Due to Λ being a diagonal matrix with the system eigenvalues on the diagonal, equation 6.30 constitutes a set of scalar equations:

$$\dot{y_i} = \lambda_i y_i - c_i \tag{6.31}$$

which have an analytical solution in time. For $\lambda_i \neq 0$ the solution is given by:

$$y_i(t) = y_i^0 \exp(\lambda_i t) + \frac{c_i}{\lambda_i}(1 - \exp(\lambda_i t)) \tag{6.32}$$

where $y^0 = S^{-1}x^0$ is the initial state. For $\lambda_i = 0$ the differential equation simplifies $\dot{y_i} = -c_i$, and the solution is:

$$y_i = y_i^0 - c_i t \tag{6.33}$$

The solutions for the original variables x_1 and x_2 are found by back-transformation:

$$x_1 = S y \tag{6.34}$$

$$x_2 = B_1 x_1 - b_2 \tag{6.35}$$

The idea is to alter the method described in this chapter, so that it can be used to solve the case of heat conduction in a spherical geometry consisting of layers with different material properties. Also, the heat flux out of the outer spherical layer into the surroundings is unknown, but shall equal the amount of energy required to evaporate the moisture content in a layer of given thickness.

6.4 The moving boundary condition

The heat flux out from the dry sphere must equal the required evaporation energy necessary to evaporate the water content of the next spherical shell. The required evaporation energy is given by:

$$Q_{vap} = L_v m_w = L_v \rho_w V_w = L_v \rho_w \phi_w \frac{4}{3}\pi((r_N + \Delta r)^3 - r_N^3) \tag{6.36}$$

where L_v is latent heat of evaporation of water, m_w the mass of water which is calculated from density, ρ_w and volume, V_w. ϕ_w is the fraction of the spherical shell consisting of water.

The energy balance at the boundary becomes:

$$qA_N = \frac{Q_{vap}}{t_{vap}} \tag{6.37}$$

where A is the area perpendicular to the flux, Q_{vap} the evaporation energy and t_{vap} the amount of time necessary to reach the energy requirement. The heat flux at the outer boundary, at $r = R$, is given by Fouriers law:

$$q = -k\frac{\partial T}{\partial r} \tag{6.38}$$

Substituting $q = \frac{Q_{vap}}{At_{vap}}$ into the discretized equation for the outer boundary results in an expression in the unknown variable t_{vap}:

$$T_{N-1}(t_{vap}) - T_{N+1}(t_{vap}) = Y\frac{Q_{vap}}{A_N t_{vap}} \tag{6.39}$$

where Y is a coefficient taking into account the material properties and the discretization method. And $T_{N-1}(t_{vap})$ and $T_{N+1}(t_{vap})$ are the temperature to the left and right of the outer radius, r_N, of the dry shell at time t_{vap}. The temperature in the shell where water is evaporating must equal the boiling temperature, T_b. Thus, the required time, t_{vap}, giving the correct heat transfer by conduction through the core and the dry sphere, can be calculated from:

$$T_{N-1}(t_{vap}) - Y\frac{Q_{vap}}{A_N t_{vap}} = T_b \tag{6.40}$$

where $T_{N-1}(t)$ is given by the solution to the equation system.

Chapter 7

Numerical discretization of model equations

In order to model the heat transfer in the mulitilayer spherical geometry, a numerical method for discretization of the partial differential equation describing heat conduction had to be chosen. A proposition from supervisor Tore Haug-Warberg, was to discretize the second order temperature differential in terms of radius, by a second order central difference scheme. By doing so, a constant coefficient matrix differential equations system could be formulated. By use of eigenvalues and eigenvectors of the matrix system, the equations could be solved analytically in time. The overall method thus yields a semi-analytical solution.

The heat transfer in this problem can have to modes. Mode one is when there is happening a combustion reaction in the furnace so that heat is being generated and transferred to both the iron core and the dry sand layer. Mode two is when the furnace is not burning and heat is transferred by conduction from hot to cold areas in the sphere. Due to the different modes, two stencils and two matrix systems have been developed to solve the problem. They only differ in how the temperature in the core and the furnace is modelled. Figure 7.1 illustrates the temperature at different nodes in the sphere for the case of combustion taking place in the furnace. For the heating case, there are 8 inner nodes and 4 ghost nodes in the initial matrix. Figure 7.2 is the equivalent model for conduction in all layers. To preserve the second order derivative in the heat conduction equation, the discretization contain three inner nodes for all constant layers. The initial matrix therefore includes 12 inner temperature nodes and 8 ghost nodes.

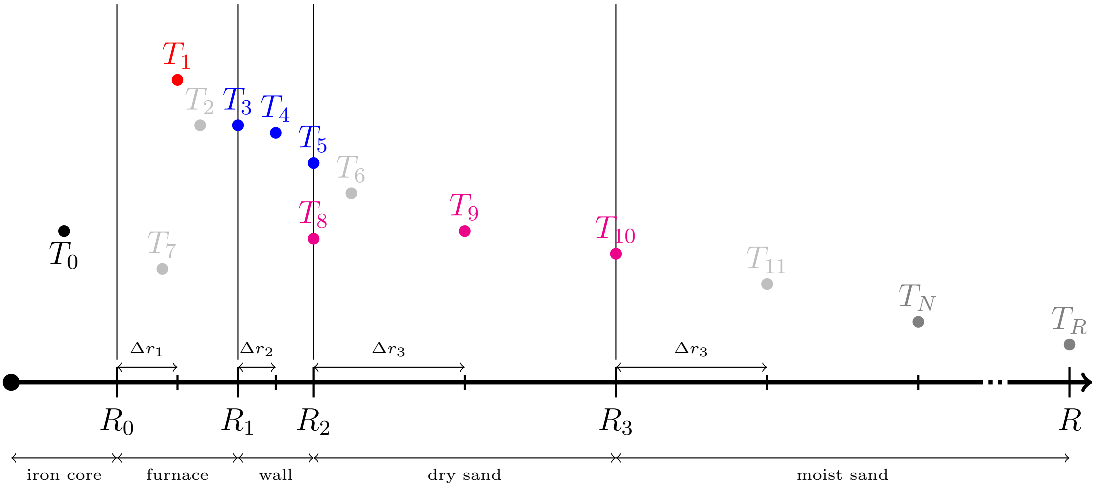

Figure 7.1: The discretization of temperature nodes for the case of a burning furnace. The temperatures in the iron core, T_0, and in the furnace, T_1, are assumed to be uniform for the spherical layer. Heat is transferred by conduction from the furnace to the wall and the dry sand. The red temperature nodes, T_2, T_6, T_7 and T_11, are ghost nodes and the blue inner nodes are modelled by the discretized heat conduction equation.

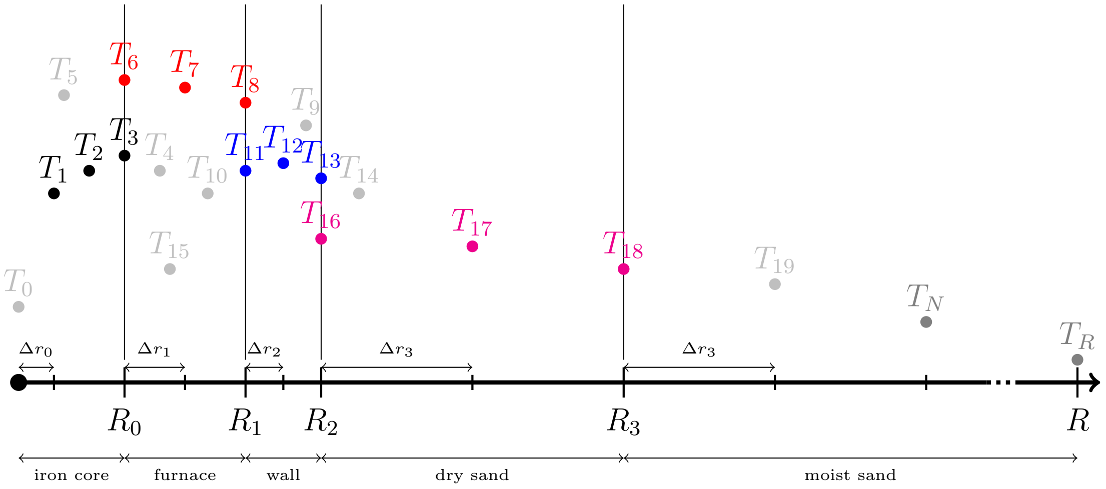

Figure 7.2: The stencil for the numerical discretization of the heat transfer in the sphere. The red temperature nodes are ghost nodes, the black nodes in the core and the furnace are modelled by the energy balance for the combustion reaction and the blue inner temperature nodes are governed by conduction.

7.1 Equations modelling the temperature of the different layers

7.1.1 Iron core

When there is heating in the furnace, the iron core is assumed to have a uniform temperature, T_0, governed by the heat transfer from the hot coal-gas mixture in the surrounding layer.

The isothermal energy balance for the core becomes:

$$\frac{dU_0}{dt} = \rho_0 V_0 \frac{\partial u_0}{\partial T_0}\frac{\partial T_0}{\partial t} = Q = h_0 A_0 (T_1 - T_0) \tag{7.1}$$

with variable transformation of T:

$$\frac{\rho_0 V_0 c_{p,Fe}}{R_0}\frac{\partial T_0}{\partial t} = h_0 A_0 \left(\frac{T_1}{R_1} - \frac{T_0}{R_0}\right) \tag{7.2}$$

Rearranged for the time dependence of the core temperature:

$$\dot{T_0} = -\frac{h_0 A_0}{\rho_0 V_0 c_{p,Fe}} T_0 + \frac{h_0 A_0 R_0}{\rho_0 V_0 c_{p,Fe} R_1} T_1 \tag{7.3}$$

The temperature evolution in the iron core when there is no heat supply from the furnace, is governed by heat conduction:

$$\dot{T_i} = \frac{\alpha_{Fe}}{\Delta r_0^2}\left[T_{i-1} - 2T_i + T_{i+1}\right] = a_0 \left[T_{i-1} - 2T_i + T_{i+1}\right], \quad \text{i=1,2,3} \tag{7.4}$$

The temperature in the center, $r = 0$, is set to zero at all times due to the variable substitution, $T = rV$:

$$T_0 = 0 \tag{7.5}$$

The actual temperature in the center is extrapolated from the neighboring node.

The interface condition between core and furnace is given by equation which rearranges to

$$\frac{(R_0 + \Delta r_0)}{(R_0 - \Delta r_0)} T_1 - \frac{2h_0 \Delta r_0 (R_0 + \Delta r_0)}{k_{Fe} R_0} T_2 - T_3 + \frac{2h_0 \Delta r_0 (R_0 + \Delta r_0)}{k_{Fe} R_0} T_5 = 0 \tag{7.6}$$

7.1.2 Coal-fired furnace

It is assumed that the only reaction taking place in the furnace is:

$$2C(s) + O_2(g) + 3.728 N_2(g) \rightarrow 2CO(g) + 3.728 N_2(g) \tag{7.7}$$

where air is assumed to contain 20.946% O_2 and 78.084% N_2.

The energy balance for the second layer as a control volume becomes:

$$\frac{dU_1}{dt} = \dot{H}_{in} - \dot{H}_{out} + \dot{Q} \tag{7.8}$$

$$N_C \frac{\partial u_C}{\partial T_1}\frac{\partial T_1}{\partial t} = \dot{H}_{O_2}(T_{air}) + \dot{H}_{N_2}(T_{air}) - \dot{H}_{CO}(T_1) - \dot{H}_{N_2}(T_1) - h_0 A_0 (T_1 - T_0) - h_1 A_1 (T_1 - T_3) \tag{7.9}$$

$$\frac{\rho_C V_C}{M_{m,C}} c_{v,C} \frac{\partial T_1}{\partial t} = \dot{H}_{O_2}(T_{air}) + \dot{H}_{N_2}(T_{air}) - \dot{H}_{CO}(T_1) - \dot{H}_{N_2}(T_1) - h_0 A_0 (T_1 - T_0) - h_1 A_1 (T_1 - T_3) \tag{7.10}$$

where

$$\dot{H}_{O_2}(T_{air}) = \dot{n}_{O_2}\left[\overline{h}^{\ominus}_{f,O_2} + \overline{c}_{p,O_2}(T_{air} - T^{\ominus})\right] = \dot{n}_{O_2}\overline{c}_{p,O_2}(T_{air} - T^{\ominus}) \tag{7.11}$$

$$\dot{H}_{N_2}(T_{air}) = 3.728\dot{n}_{O_2}\left[\overline{h}^{\ominus}_{f,N_2} + \overline{c}_{p,N_2}(T_{air} - T^{\ominus})\right] = 3.728\dot{n}_{O_2}\left[\overline{c}_{p,N_2}(T_{air} - T^{\ominus})\right] \tag{7.12}$$

$$\dot{H}_{CO}(T_1) = 2\dot{n}_{O_2}\left[\overline{h}^{\ominus}_{f,CO} + \overline{c}_{p,CO}(T_1 - T^{\ominus})\right] \tag{7.13}$$

$$\dot{H}_{N_2}(T_1) = 3.728\dot{n}_{O_2}\left[\overline{h}^{\ominus}_{f,N_2} + \overline{c}_{p,N_2}(T_1 - T^{\ominus})\right] = 3.728\dot{n}_{O_2}\left[\overline{c}_{p,N_2}(T_1 - T^{\ominus})\right] \tag{7.14}$$

The standard enthalpy of formation, $\overline{H}^{\ominus}_f$, and the heat capacity, $\overline{c}_p$, are on molar basis. By definition, elements in their standard state (graphite, oxygen gas and nitrogen gas) have a standard enthalpy of formation defined as zero. The energy balance for the furnace becomes:

$$\begin{aligned}\frac{\rho_C V_C}{M_{m,C}} c_{v,C}\frac{\partial T_1}{\partial t} &= \dot{n}_{O_2}\overline{c}_{p,O_2}(T_{air} - T^{\ominus}) + 3.728\dot{n}_{O_2}\overline{c}_{p,N_2}(T_{air} - T^{\ominus}) \\ &- 2\dot{n}_{O_2}\left[\overline{h}^{\ominus}_{f,CO} + \overline{c}_{p,CO}(T_1 - T^{\ominus})\right] - 3.728\dot{n}_{O_2}\overline{c}_{p,N_2}(T_1 - T^{\ominus}) \\ &- h_0 A_0(T_1 - T_0) - h_1 A_1(T_1 - T_3)\end{aligned} \tag{7.15}$$

The dominant mass in the furnace will be the solid coal particles, assumed to be pure carbon (graphite). Thus, the specific heat capacity at constant volume for the control volume, can be approximated by the specific heat capacity of carbon at constant pressure.

The equation transformed to temperature in sphere:

$$\begin{aligned}\dot{T}_1 = \frac{M_{m,C}}{\rho_C V_C c_{p,C}}\Bigg[&\frac{h_0 A_0 R_1}{R_0}T_0 + \left[-\dot{n}_{O_2}\left(2\overline{c}_{p,CO} + 3.7282\overline{c}_{p,N_2}\right) - h_0 A_0 - h_1 A_1\right]T_1 + h_1 A_1 T_3 \\ &+\dot{n}_{O_2}\left(\overline{c}_{p,O_2}\left(T_{air} - T^{\ominus}\right) + 3.7282\overline{c}_{p,N_2}T_{air} - 2\overline{h}^{\ominus}_{f,CO} + 2\overline{c}_{p,CO}T^{\ominus}\right)R_1\Bigg]\end{aligned} \tag{7.16}$$

When there is no combustion reaction taking place, the temperature in the furnace will also be governed by heat conduction. Ghost nodes are introduced to satisfy the interface conditions. The interface condition between core and furnace is given by:

$$\frac{2h_0\Delta r_1(R_0 - \Delta r_1)}{k_C R_0}T_2 - T_4 - \frac{2h_0\Delta r_1(R_0 - \Delta r_1)}{k_C R_0}T_5 + \frac{(R_0 - \Delta r_1)}{(R_0 + \Delta r_1)}T_6 = 0 \tag{7.17}$$

The temperature evolution in the non-burning furnace is governed by heat conduction:

$$\dot{T}_i = \frac{\alpha_C}{\Delta r_1^2}\left[T_{i-1} - 2T_i + T_{i+1}\right] = a_1\left[T_{i-1} - 2T_i + T_{i+1}\right], \quad \text{i=5,6,7} \tag{7.18}$$

The interface condition between furnace and wall is given by:

$$\frac{(R_1 + \Delta r_1)}{(R_1 - \Delta r_1)}T_6 - \frac{2h_1\Delta r_1(R_1 + \Delta r_1)}{k_C R_1}T_7 - T_8 + \frac{2h_1\Delta r_1(R_1 + \Delta r_1)}{k_C R_1}T_{10} = 0 \tag{7.19}$$

7.1.3 Furnace wall

The wall encapsulating the furnace will transfer heat through conduction due to the temperature gradient. The formulation and discretization of the diffusion equation is explained in more detail in the next chapter. The inner nodes of the wall are described by:

$$\frac{\partial T_i}{\partial t} = \frac{\alpha_w}{\Delta r_2^2}\left[T_{i-1} - 2T_i + T_{i+1}\right] = c_w\left[T_{i-1} - 2T_i + T_{i+1}\right], \quad \text{i=3,4,5} \tag{7.20}$$

The interface condition between furnace and wall is given by:

$$\frac{2h_1\Delta r_2(R_1 - \Delta r_2)}{k_w R_1}T_1 - T_2 - \frac{2h_1\Delta r_2(R_1 - \Delta r_2)}{k_w R_1}T_3 + \frac{(R_1 - \Delta r_2)}{(R_1 + \Delta r_2)}T_4 = 0 \tag{7.21}$$

The interface condition between furnace and wall:

$$\frac{2h_1\Delta r_2(R_1 - \Delta r_2)}{k_w R_1}T_7 - T_9 - \frac{2h_1\Delta r_2(R_1 - \Delta r_2)}{k_w R_1}T_{10} + \frac{(R_1 - \Delta r_2)}{(R_1 + \Delta r_2)}T_{11} = 0 \tag{7.22}$$

Inner nodes of wall:

$$\dot{T_i} = \frac{\alpha_w}{\Delta r_2^2}\left[T_{i-1} - 2T_i + T_{i+1}\right] = a_2\left[T_{i-1} - 2T_i + T_{i+1}\right], \quad \text{i=10,11,12} \tag{7.23}$$

Interface condition between wall and dry sand:

$$\frac{(R_2 + \Delta r_2)}{(R_2 - \Delta r_2)}T_{11} - \frac{2h_2\Delta r_2(R_2 + \Delta r_2)}{k_w R_2}T_{12} - T_{13} + \frac{2h_2\Delta r_2(R_2 + \Delta r_2)}{k_C R_2}T_{15} = 0 \tag{7.24}$$

7.1.4 Dry sand

The heat supplied by the furnace is used in evaporating moisture from the outer sand layer. In the starting geometry there exists a thin layer of dry sand that conducts heat to the next moist layer. The equation describing the inner nodes of the sand layer is equivalent to the conduction in the furnace wall:

$$\frac{\partial T_j}{\partial t} = \frac{\alpha_s}{\Delta r_3^2}\left[T_{j-1} - 2T_j + T_{j+1}\right] = c_s\left[T_{j-1} - 2T_j + T_{j+1}\right], \quad \text{j=8,9,10,..,N} \tag{7.25}$$

The interface condition between the wall and the dry sand becomes:

$$\frac{(R_2 + \Delta r_2)}{(R_2 - \Delta r_2)}T_4 - \frac{2h_2\Delta r_2(R_2 + \Delta r_2)}{k_w R_2}T_5 - T_6 + \frac{2h_2\Delta r_2(R_2 + \Delta r_2)}{k_w R_2}T_8 = 0 \tag{7.26}$$

$$\frac{2h_2\Delta r_3(R_2 - \Delta r_3)}{k_s R_2}T_5 - T_7 - \frac{2h_2\Delta r_3(R_2 - \Delta r_3)}{k_s R_2}T_8 + \frac{(R_2 - \Delta r_3)}{(R_2 + \Delta r_3)}T_9 = 0 \tag{7.27}$$

The right boundary condition for dry sand is an equation to yield the necessary evaporation energy for the next layer to become dry:

$$-k_s A_4 \frac{\frac{T_{11}}{r_{11}} - \frac{T_9}{r_9}}{2\Delta r_3} = q \tag{7.28}$$

$$\frac{(R_3+\Delta r_3)}{(R_3-\Delta r_3)}T_9 - T_{11} = \frac{\Delta r_3(R_3+\Delta r_3)}{2k_s\pi R_3^2}q \tag{7.29}$$

The heat supplied by the furnace is used in evaporating moisture from the outer sand layer. In the starting geometry there exists a thin layer of dry sand that conducts heat to the next moist layer.

The interface condition between the wall and the dry sand becomes:

$$\frac{2h_2\Delta r_3(R_2-\Delta r_3)}{k_sR_2}T_{12} - T_{14} - \frac{2h_2\Delta r_3(R_2-\Delta r_3)}{k_sR_2}T_{15} + \frac{(R_2-\Delta r_3)}{(R_2+\Delta r_3)}T_{16} = 0 \tag{7.30}$$

The equation describing the inner nodes of the sand layer is equivalent to the conduction in the furnace wall:

$$\dot{T_i} = \frac{\alpha_s}{\Delta r_3^2}\left[T_{i-1} - 2T_i + T_{i+1}\right] = a_3\left[T_{i-1} - 2T_i + T_{i+1}\right], \quad \text{i=15,16,17,..,N} \tag{7.31}$$

The right boundary condition for dry sand is an equation to yield the necessary evaporation energy for the next layer to become dry:

$$-k_s\frac{\frac{T_{N+1}}{r_{N+1}} - \frac{T_{N-1}}{r_{N-1}}}{2\Delta r} = q \tag{7.32}$$

$$\frac{(R+\Delta r)}{(R-\Delta r)}T_{N-1} - T_{N+1} = \frac{2\Delta r(R+\Delta r)}{k_s}q \tag{7.33}$$

$$q = \frac{Q_{vap}}{A_Rt} = \frac{\chi\rho_wV_wH_{vap,w}}{A_Rt_{vap}} \tag{7.34}$$

From equation 7.32 - 7.34 the required evaporation time, t_{vap}, can be solved by substituting eq. 7.34 in the equation system and solve for t_{vap} when the temperature in the next shell is equal to the evaporation temperature of water $T_{N+1} = T_{b,w}$.

7.2 Matrix equation system for burning furnace

Equation *core*, *furnace* and *conduction in wall - sand* can be represented in matrix form. The temperature nodes are seperated into vectors containg the inner nodes and the ghost nodes. The resulting coupled matrix system becomes:

$$\begin{bmatrix} -a_{00} & a_{01} & 0 & 0 & 0 & 0 & 0 & 0 \\ a_{10} & a_{11} & a_{13} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -2a_{22} & a_{22} & 0 & 0 & 0 & 0 \\ 0 & 0 & a_{22} & -2a_{22} & a_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{22} & -2a_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -2a_{33} & a_{33} & 0 \\ 0 & 0 & 0 & 0 & 0 & a_{33} & -2a_{33} & a_{33} \\ 0 & 0 & 0 & 0 & 0 & 0 & a_{33} & -2a_{33} \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_3 \\ T_4 \\ T_5 \\ T_8 \\ T_9 \\ T_{10} \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ a_{22} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & a_{22} & 0 & 0 \\ 0 & 0 & a_{33} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_{33} \end{bmatrix} \begin{bmatrix} T_2 \\ T_6 \\ T_7 \\ T_{11} \end{bmatrix} = \begin{bmatrix} \dot{T_0} \\ \dot{T_1} \\ \dot{T_3} \\ \dot{T_4} \\ \dot{T_5} \\ \dot{T_8} \\ \dot{T_9} \\ \dot{T_{10}} \end{bmatrix} - \begin{bmatrix} 0 \\ C_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & b_{21} & -b_{21} & b_{24} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & b_{64} & -b_{65} & b_{65} & 0 & 0 \\ 0 & 0 & 0 & 0 & b_{75} & -b_{75} & b_{79} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & b_{119} & 0 \end{bmatrix} \begin{bmatrix} T_0 \\ T_1 \\ T_3 \\ T_4 \\ T_5 \\ T_8 \\ T_9 \\ T_{10} \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} T_2 \\ T_6 \\ T_7 \\ T_{11} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ qy \end{bmatrix}$$

where

$$\begin{aligned} a_{00} &= \frac{h_0 A_0}{\rho_0 V_0 c_{p,Fe}} & a_{01} &= \frac{h_0 A_0 R_0}{\rho_0 V_0 c_{p,Fe} R_1} \\ a_{10} &= \frac{h_0 A_0 R_1}{\rho_1 V_1 c_{p,C} R_0} & a_{13} &= \frac{h_1 A_1}{\rho_1 V_1 c_{p,C}} \\ a_{11} &= \frac{-2\overline{c}_{p,CO} - 3.7282\overline{c}_{p,N_2} - h_0 A_0 - h_1 A_1}{\rho_1 V_1 c_{p,C}} & & \\ b_{21} &= \frac{2(R_1 - \Delta r_2)\Delta r_2 h_1}{k_w} & b_{24} &= \frac{R_1 - \Delta r_2}{R_1 + \Delta r_2} \\ c_{64} &= \frac{R_2 + \Delta r_2}{R_1 + \Delta r_2} & c_{65} &= \frac{2(R_2 + \Delta r_2)\Delta r_2 h_2}{k_w} \\ c_{75} &= \frac{2(R_2 - \Delta r_3)\Delta r_3 h_2}{k_s} & c_{79} &= \frac{R_2 - \Delta r_3}{R_2 + \Delta r_3} \\ c_{119} &= \frac{R_3 + \Delta r_3}{R_3 - \Delta r_3} & y &= \frac{(R_3 + \Delta r_3)\Delta r_3}{2 k_s \pi R_3^2} \end{aligned}$$

$$C_1 = \frac{R_1}{\rho_1 V_1 c_{p,C}} \left(\overline{c}_{p,O_2} \left(T_{air} - T^{\ominus} \right) + 3.7282\overline{c}_{p,N_2} T_{air} - 2\overline{h}^{\ominus}_{f,CO} + 2\overline{c}_{p,CO} T^{\ominus} \right)$$

The matrix system can be written as:

$$\begin{aligned} A_1 x_1 + A_2 x_2 &= \dot{x_1} - b_1 \\ B_1 x_1 + B_2 x_2 &= b_2 \end{aligned}$$

where $B_2 = -I$, so that $x_2 = B_1 x_1 - b_2$, and inserted in equation 7.35:

$$A_1 x_1 + A_2(B_1 x_1 - b_2) = \dot{x_1} - b_1 \tag{7.35}$$

$$(A_1 + A_2 B_1)x_1 - (A_2 b_2 - b_1) = \dot{x_1} \tag{7.36}$$

With matrix A defined by $A = A_1 + A_2 B_1$ and b defined by $b = A_2 b_2 - b_1$, the system can be solved by the eigenvalue-eigenvector method described in section 6.3.

7.3 Matrix representation for pure conduction

When there is no heat generation by the furnace, the heat will diffuse from hot to cold regions by conduction. The resulting matrix system closely resembles the simplified example in section , but the existance of multiple layers require more ghost nodes.

$$\begin{bmatrix} -2a_0 & a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ a_0 & -2a_0 & a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_0 & -2a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -2a_1 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & -2a_1 & a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_1 & -2a_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -2a_2 & a_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_2 & -2a_2 & a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_2 & -2a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2a_3 & a_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_3 & -2a_3 & a_3 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_3 & -2a_3 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_6 \\ T_7 \\ T_8 \\ T_{11} \\ T_{12} \\ T_{13} \\ T_{16} \\ T_{17} \\ T_{18} \end{bmatrix} + \begin{bmatrix} a_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & a_0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & a_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & a_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & a_2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & a_2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & a_3 \end{bmatrix} \begin{bmatrix} T_0 \\ T_4 \\ T_5 \\ T_9 \\ T_{10} \\ T_{14} \\ T_{15} \\ T_{19} \end{bmatrix} = \begin{bmatrix} \dot{T}_1 \\ \dot{T}_2 \\ \dot{T}_3 \\ \dot{T}_6 \\ \dot{T}_7 \\ \dot{T}_8 \\ \dot{T}_{11} \\ \dot{T}_{12} \\ \dot{T}_{13} \\ \dot{T}_{16} \\ \dot{T}_{17} \\ \dot{T}_{18} \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & c_{31} & -c_{32} & c_{32} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & c_{42} & -c_{42} & c_{46} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{86} & -c_{87} & c_{87} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & c_{97} & -c_{97} & c_{911} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{1311} & -c_{1312} & c_{1312} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{1412} & -c_{1412} & c_{1416} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & c_{1716} & 0 \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ T_6 \\ T_7 \\ T_8 \\ T_{11} \\ T_{12} \\ T_{13} \\ T_{16} \\ T_{17} \\ T_{18} \end{bmatrix} + \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} T_0 \\ T_4 \\ T_5 \\ T_9 \\ T_{10} \\ T_{14} \\ T_{15} \\ T_{19} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ yq \end{bmatrix}$$

where

$$\begin{aligned} a_0 &= \frac{\alpha_0}{\Delta r_0^2} & a_1 &= \frac{\alpha_1}{\Delta r_1^2} \\ a_2 &= \frac{\alpha_2}{\Delta r_2^2} & a_3 &= \frac{\alpha_3}{\Delta r_3^2} \\ c_{31} &= \frac{R_0+\Delta r_0}{R_0-\Delta r_0} & c_{32} &= \frac{2h_0\Delta r_0(R_0+\Delta r_0)}{k_{Fe}R_0} \\ c_{42} &= \frac{2h_0\Delta r_1(R_1-\Delta r_1)}{k_C R_0} & c_{46} &= \frac{R_0-\Delta r_1}{R_0+\Delta r_1} \\ c_{86} &= \frac{R_1+\Delta r_1}{R_1-\Delta r_1} & c_{87} &= \frac{2h_1\Delta r_1(R_1+\Delta r_1)}{k_C R_1} \\ c_{97} &= \frac{2h_1\Delta r_2(R_1-\Delta r_2)}{k_w R_1} & c_{911} &= \frac{R_1-\Delta r_2}{R_1+\Delta r_2} \\ c_{1311} &= \frac{R_2+\Delta r_2}{R_2-\Delta r_2} & c_{1312} &= \frac{2h_2\Delta r_2(R_2+\Delta r_2)}{k_w R_2} \\ c_{1412} &= \frac{2h_2\Delta r_3(R_2-\Delta r_3)}{k_s R_2} & c_{1416} &= \frac{R_2-\Delta r_3}{R_2+\Delta r_3} \\ c_{1716} &= \frac{R_3+\Delta r_3}{R_3-\Delta r_3} \end{aligned}$$

The form of the system is equivalent to the form in section 6.2 with:

$$A = (A_1 + A_2 B_1) \tag{7.37}$$

$$b = A_2 b_2 \tag{7.38}$$

The solution is found directly from the method described in section 6.3.

Chapter 8

Time constant

8.1 Time constant for convection of heat

The concept of a characteristic time, τ, for first order differential systems is often used in describing electrical circuits and radioactive decay. The general differential equation for quantity, X, and the general solution, $X(t)$, for such systems are given by:

$$\begin{aligned} \frac{dX}{dt} &= -\frac{1}{\tau}X + \frac{1}{\tau}X_a && (8.1) \\ X(t) &= X_0 e^{-\frac{t}{\tau}} + X_a\left(1 - e^{-\frac{t}{\tau}}\right) && (8.2) \end{aligned}$$

where X_0 is the initial condition and X_a a constant force term. By investigtion of the solution, it is evident that a large time constant results in a slow reacting system and vice versa.

For thermal systems, the concept of a thermal time constant can be used to describe uniform heating or cooling by convection. It is then assumed that an object with density, ρ, volume, V, heat capacity, c_p and a uniform temperature, $T(t)$, has a transient temperature evolution given by:

$$\rho V c_p \frac{dT}{dt} = \pm hA(T(t) - T_a) \tag{8.3}$$

The right hand side of the equation models convective heat flux through surface A given heat transfer coefficient h and ambient temperature T_a. The term is defined as positive for heating and negative for cooling.

Rearranged to the general form in equation 8.1, the τ-expression becomes:

$$\tau = \frac{\rho V c_p}{hA} \tag{8.4}$$

which corresponds with the expectation that a large specific heat capacity will lead to a slow transient response in temperature, and that a large heat transfer coefficient and a large surface area will give faster change in temperature of the body.

The solution to equation 8.3 for cooling, can be written:

$$\frac{T(t) - T_a}{T_0 - T_a} = e^{-\frac{t}{\tau}} \tag{8.5}$$

The left hand side is a fraction of the temperature difference between the body and the ambient at time t, $T(t) - T_a$, divided by the difference between the body and the ambient at the initial time. For a hot body in cold surroundings, the initial difference will be largest. The present time temperature difference divided by the initial difference yields a percentage whose magnitude depends on the value of τ. For elapsed time equal to one thermal time constant, $t = \tau$, this percentage is 36.79%. In other word, after one time constant, the energy of the body has decreased by 63.21% of the initial energy.

8.2 Effective time constant for diffusion of heat

A time constant of the same definition as above does not exist for a second order partial differential equation such as the diffusion equation. A method for finding an effective time constant for diffusive systems has been proposed by [Collins, 1980]. It involves applying the Laplace transformation to the diffusion equation. The proposition is that an effective time constant can be expressed as a limit:

$$\tau = \lim_{s \to 0} \left(\frac{T_e}{s^2} + \frac{dT(s)}{ds} \right) \left[\lim_{s \to 0} \left(\frac{T_e}{s} - T(s) \right) \right]^{-1} \tag{8.6}$$

where the stationary temperature is calculated from "final value" theorem:

$$T_e = \lim_{s \to 0} sT(s) \tag{8.7}$$

For small s, the Laplace solution, $T(s)$, may be expressed as a series expansion in powers of s on the form

$$T(s) = \frac{T_e}{s} + B + Cs + \text{Higher order terms in } s \tag{8.8}$$

If so, the effective time constant takes the simple form of

$$\tau_{\text{eff}} = -\frac{C}{B} \tag{8.9}$$

By transforming the heat conduction equation into the form given in equation 8.8, and applying the appropriate boundary and initial conditions, an effective time constant corresponding to the definitions in [Collins, 1980] was found. The derivation is shown in appendix **??**. The Laplace transform of the heat conduction equation is

$$T(r, s) = \frac{R(T_R - T_0)}{rs} \frac{\sinh(ar)}{\sinh(aR)} + \frac{T_0}{s}, \quad a = \sqrt{\frac{s}{\alpha}} \tag{8.10}$$

The stationary solution is given by the limit as time approaches ∞, which means that s approches 0:

$$T_e = \lim_{s \to 0} sT(r, s) = T_R \tag{8.11}$$

For small s, the expression for $T(r,s)$ can be expanded as a serie in powers of s. Series expansion of the hyperbolic functions gives the following power series for $T(r,s)$

$$T(r,s) = \frac{T_R}{s} + \frac{(T_R - T_0)(r^2 - R^2)}{6\alpha} + \frac{(T_R - T_0)(7R^4 - 10r^2R^2 + 3r^4)}{360\alpha^2}s \tag{8.12}$$

$$+ \frac{r^2R^2(7R^2 - 3r^2)}{6 \cdot 360\alpha^3}s^2 \tag{8.13}$$

Thus,

$$B = \frac{(T_R - T_0)(r^2 - R^2)}{6\alpha}$$

$$C = \frac{(T_R - T_0)(7R^4 - 10r^2R^2 + 3r^4)}{360\alpha^2}$$

yielding the expression for an effective time constant τ

$$\tau_{\text{eff}} = \frac{(7R^2 - 3r^2)}{60\alpha} \tag{8.14}$$

The effective time constant τ for a sphere of dimension $R = 0.12$ m, and thermal diffusivity, $\alpha = 1.24 \cdot 10^{-7}\ m^2/s$, reflecting the properties of the carbon containing spherical furnace used in this model, are listed in table 8.1

r [m]	t [min]
0.06	200.57
0.07	191.88
0.08	181.85
0.09	170.49
0.10	157.79
0.11	143.75
0.12	128.37

Table 8.1: The effective time constant at different radial positions calculated from equation 8.14.

The effective time constant is highly dependent on the geometry. The positional variation increases with increased size of the sphere.

8.3 Time constant based on total amount of diffused substance

From [Crank, 1975] a solution to the non-steady diffusion problem in a sphere of radius a was found by applying the substitution $u = rC$:

$$\frac{\partial u}{\partial t} = D\frac{\partial^2 u}{\partial r^2} \tag{8.15}$$

with initial and boundary conditions:

$$u = 0 \quad r = 0, \quad t < 0, \tag{8.16}$$

$$u = aC_0 \quad r = a, \quad t < 0, \tag{8.17}$$

$$u = rC_1 \quad 0 \geq r \leq a, \quad t = 0, \tag{8.18}$$

where C_0 is a constant consentration at the surface and C_1 is a uniform consentration at the initial time.

The solution is given by:

$$\frac{C - C_1}{C_0 - C_1} = 1 + \frac{2a}{\pi r}\sum_{n=1}^{\infty}\frac{(-1)^n}{n}\sin\frac{n\pi r}{a}\exp\left(-Dn^2\pi^2 t/a^2\right) \tag{8.19}$$

By integation of the above expression over the entire sphere, the total amount of diffusing substance leaving the sphere was found. The fraction of diffused amount as a function of time over the diffused amount after infinite time is given by [Crank, 1975]:

$$\frac{M_t}{M_\infty} = 1 - \frac{6}{\pi^2}\sum_{n=1}^{\infty}\frac{1}{n^2}\exp\left(-Dn^2\pi^2 t/a^2\right) \tag{8.20}$$

In figure 8.1, the fraction of diffused amount is given as a function of the dimensionless number $\left(\frac{Dt}{a^2}\right)^{\frac{1}{2}}$. The red dots indicate fractions corresponding to multiples of the traditional first order time constants, see table . Thus, a single time constant for the entire sphere can be found from $\left(\frac{Dt}{a^2}\right)^{\frac{1}{2}}$ by solving for t. The corresponding time constant values are given in table 8.2 for the different choices of $\frac{M_t}{M_\infty}$.

8.4 Comparison of the two time constants

The expression for the effective time constant based on the derivations in [Collins, 1980], is given by

$$\tau_{\text{eff}} = \frac{7R^2 - 3r^2}{60D} \tag{8.21}$$

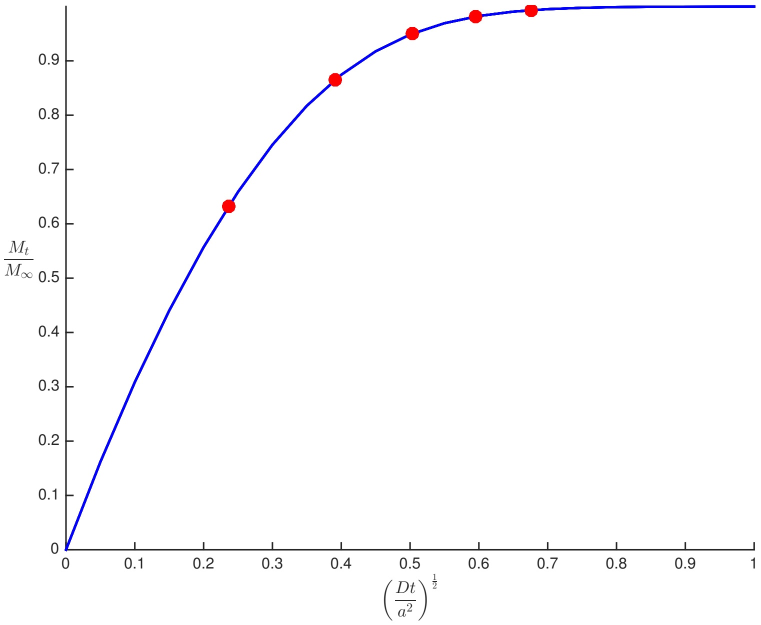

Figure 8.1: Diffused percentage from a sphere as a function of the dimensionless number $\frac{-Dn^2\pi^2 t}{a^2}$. The equation for the plot is given in 8.20. The red dots indicate 63.2%, 86.5%, 95%, 98% and 99%, and gives the times listed in table 8.2.

$\frac{M_t}{M_\infty}$ [%]	t [min]
63.21	107.39
86.47	293.63
95.02	488.22
98.17	683.29
99.33	878.38

Table 8.2: Time constants, t, for different percentages of diffused amount calculated by equation 8.20 from [Crank, 1975]. The percentages correspond to τ times 1-5, respectively.

The time constant calculated from the expression of total amount of diffused substance found in [Crank, 1975] is expressed by

$$\tau_{M_t} = \frac{R^2 x^2}{D} \tag{8.22}$$

where x is dependent on the diffusion fraction $\frac{M_t}{M_\infty}$ and can be solved from eq. or graphically from figure 8.1.

The correspondance between the two time constants can be expressed through the fractional relationship

$$\tau_{\text{rel}} = \frac{\tau_{\text{eff}}}{\tau_{M_t}} = \frac{7R^2 - 3r^2}{60D} \frac{D}{R^2 x^2} = \frac{1}{20x^2}\left(\frac{7}{3} - \frac{r^2}{R^2}\right) \tag{8.23}$$

From this relative time constant, it is seen that it depends on the choice of diffusion fraction $\frac{M_t}{M_\infty}$ through x, and on the position in the sphere through r. In the center of the sphere, the relation becomes $\tau_{\text{eff}} = \frac{7}{60x^2}\tau_{M_t}$, whereas at the surface the relation is $\tau_{\text{eff}} = \frac{1}{15x^2}\tau_{M_t}$.

In figure 8.2, effective time constant τ_{eff} and τ_{M_t} for 75% of total diffusion, are plotted as a function of radial position in a sphere of radius $R = 0.15$ m.

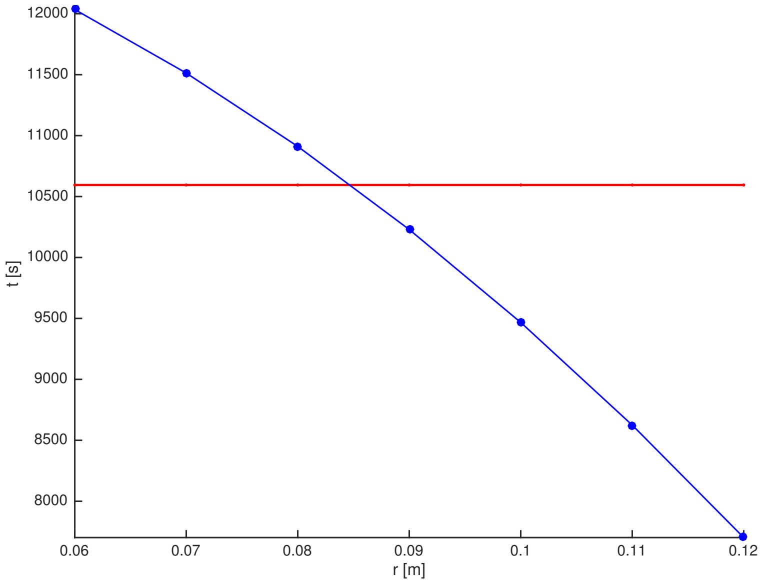

Figure 8.2: Comparison of radius-dependent time constant and the energy-based time constant for 75% diffusion.

The energy-based time constant can be used to determine the specified extent of the process corresponding to the effective time constant, and the relative expression can be used to account for the positional variation neglected by the energy-based method.

Chapter 9

Temperature profiles

To solve the matrix system, initial temperatures for all the inner nodes must be given. For the case with heating by combustion of the sphere, the initial temperatures were set uniformly to the boiling temperature of water T_0=373.15 K. For the case of pure conduction, the initial temperatures were set to the respective final temperatures in the nodes after heating. A full simulation of heating and conduction is presented for an inlet feed containing 0.2 $\frac{\text{mol}}{\text{s}}$ of O_2. To investigate the impact of the O_2 content in the feed, temperature profiles for heating were calculated for both a factor 5 increase of the O_2 and a factor 5 decrease of O_2 in inlet feed.

Because of the way the moving boundary condition is defined, the temperature profiles are not illustrated in the traditional manner of temperature plotted as a function of time. Rather, the profiles are shown at specific times, corresponding to the movement of the boundary, with the temperature plotted at radial nodes. The radial coordinate is therefore increasing for each temperature profile due to the increasing layer of dry sand. The interfaces of the layers are marked on the horizontal axis. To avoid too much overlapping, the 15 profiles required to increase the dry radius from 0.25 m to 1 m, are separated into different figures where necessary.

9.1 Combustion: 0.2 $\frac{\text{mol}}{\text{s}}$ of O_2

The temperature profiles with a burning furnace in the sphere are given in figure 9.1 and figure 9.2. Initially, all the inner nodes held a temperature of 373.15 K. As the furnace produces heat, the temperature rises quickly in the core, the furnace and the wall. The response in the dry sand is slower due to the high heat capacity and low conductivity. Additional information to the temperature profiles are listed in table 9.1. After 31 minutes, the moving boundary expands for the first time. At this point, the temperature has already increased to 1182 K, which is 75% of the maximum temperature reached by the furnace. The temperature in the furnace has passed 1500 K before 3 hours of combustion has been completed. As is shown in figure 9.1, the temperature in the furnace has reached 99% of its final value when the dry sand layer has increased 5 times. This happens 7 hours and 20 minutes after initialization. The blue graph in figure 9.1, which is the profile after the first increment of the boundary, clearly shows that the different thermal properties of the materials greatly effect the heat transferring capabilities of the material. The furnace, quite obviously, holds the warmest temperature, but the iron core with its high thermal conductivity is almost at the same temperature. The steel wall needs longer to reach the furnace temperature, and the sand is significantly cooler than the other materials, also after long time. The graphs in figure 9.2 all show the same trends. The iron core, the furnace and the steel wall have reached a maximum temperature that are close to uniform over the nodes of the three materials. For each movement of the boundary, the dry sand gains another node holding 373.15 K and over many hours, the nodes in the sand increase its temperature. It takes 61 hours for the furnace to produce enough heat to evaporate the water contant of the moist layer when the initial temperature of the entire geometry was set to 373.15 K.

Table 9.1: Explanation to the colored graphs of figure 9.1 and 9.2. R$_o$ is the outer radius of dry sand, t_{vap} is evaporation time for each shell and t_{tot} is total time after initialization. The temperature in the furance is T_f, after each shell incrementation and the percentage of maximum furnace temperature, $\frac{T_f}{T_{f,max}}$ is given in the last column.

		#	R_o	t_{vap}	t_{tot}	T_f	$\frac{T_f}{T_{f,max}}$
			[m]	[min]	[min]	[K]	[%]
Figure 9.1:	—●—	1	0.30	31.08	31.08	1182.62	74.59
	—●—	2	0.35	57.01	88.09	1466.33	92.48
	—●—	3	0.40	86.96	175.05	1541.69	97.23
	—●—	4	0.45	117.65	292.70	1561.83	98.50
	—●—	5	0.50	148.20	440.90	1570.28	99.04
Figure 9.2:	—●—	6	0.55	178.58	619.48	1574.82	99.32
	—●—	7	0.60	209.05	828.53	1577.64	99.50
	—●—	8	0.65	239.89	1068.42	1579.59	99.62
	—●—	9	0.70	271.29	1339.71	1581.04	99.71
	—●—	10	0.75	303.33	1643.04	1582.18	99.79
	—●—	11	0.80	336.06	1979.11	1583.11	99.84
	—●—	12	0.85	369.48	2348.59	1583.88	99.89
	—●—	13	0.90	403.58	2752.17	1584.53	99.93
	—●—	14	0.95	438.34	3190.51	1585.09	99.97
	—●—	15	1.00	473.74	3664.26	1585.57	100.00

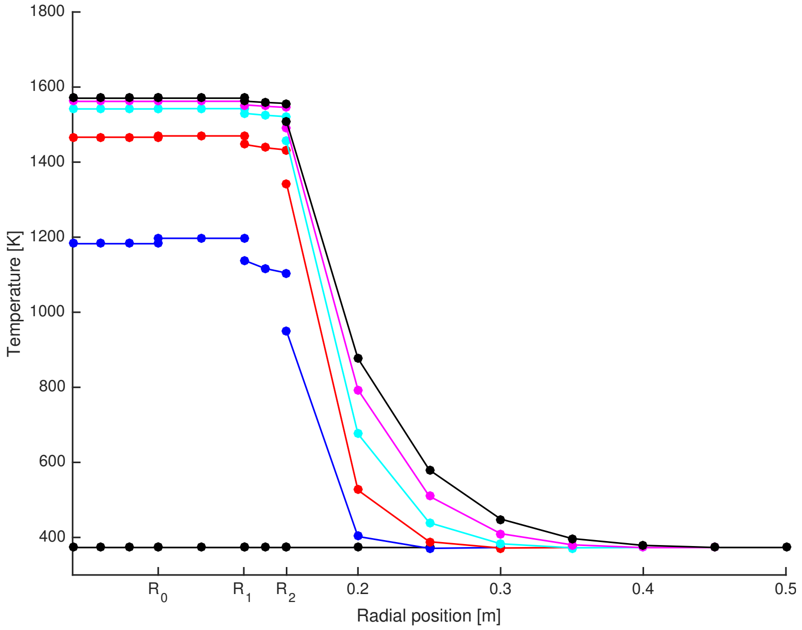

Figure 9.1: Temperature profiles for the first 5 increments of the spherical shell.

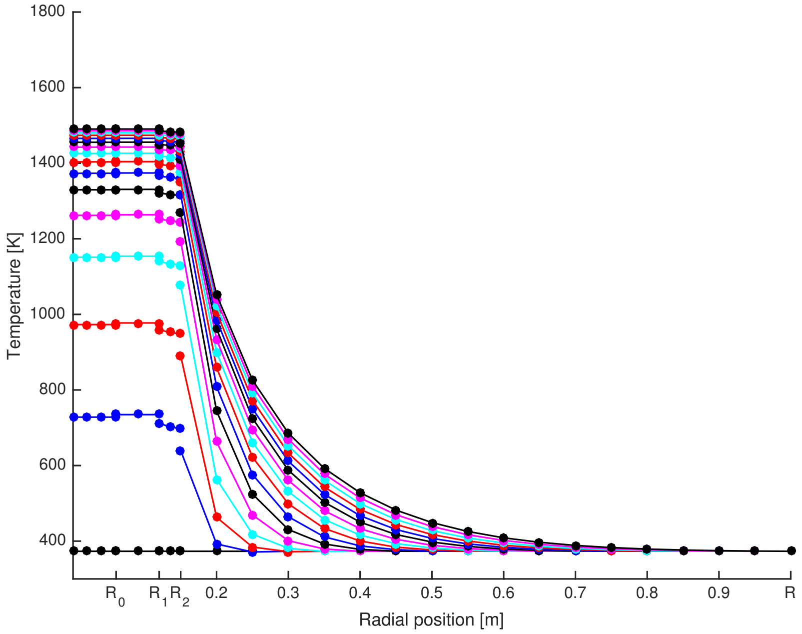

Figure 9.2: Temperature profiles for the final 10 increments of the spherical shell.

9.2 Conduction

Initial temperatures were set to the final temperatures calculated for the heating of the furnace by the process described above. In figure 9.3 the 3 first graphs are overlapping. This is caused by the large temperature difference in the three sand nodes which gives a large gradient out from the sand layer, and thus a large heat flux and a shorter evaporation time. It only takes 23 minutes in total for the first 3 increments of the boundary, and the temperature in the nodes of the core, the furnace and the wall have only decreased by a few degrees. From the fourth increment and onward, the temperature in the wall and in the furnace is affected by the heat loss to the cool sand layer. All 3 nodes of the wall are decreasing uniformily as time passes, whereas the nodes in the furnace significantly differs in temperature. The iron core is a good conductor of heat and transfers heat to the furnace. The carbon in the furnace does not conduct heat as well as metals, so the large temperature differences in the furnace arises. The inclusion of an iron core slows down the rapidity of the cool down. In figure 9.4 the trends continue. All nodes decrease in temperature, and the iron core and the steel wall hold a uniform temperature throughout the layer. A temperature gradient arises in the furnace and in the sand, which is expected as additional nodes, holding 373.15 K, are added to the dry sand layer. The heat transfer to the sand is decreasing with time, and profiles flattens for each shell increment. After 12 increments, and 32 hours, the furnace holds a mean temperature that is half of the initial temperature. If calculated from the time constant expression derived for heat loss by convection, equation 8.5, this equals the fraction for one time constant. The gradient in the furnace suggests that a radial dependent characteristic time constant is necessary. The full simulation time is 56 hours, and the temperature in the furnace, calculated as a mean, has decreased by 60%.

Table 9.2: Explanation to the colored graphs of figure 9.3 and 9.4. R$_o$ is the outer radius of dry sand, t_{vap} is evaporation time for each shell and t_{tot} is total time after initialization. The temperature in the furance is T_f, after each shell incrementation and the percentage of maximum furnace temperature, $\frac{T_f}{T_{f,max}}$ is given in the last column.

	#	R_o [m]	t_{vap} [min]	t_{tot} [min]	T_f [K]			$\frac{\overline{T}_f}{T_{f,0}}$ [%]	$\frac{\overline{T}_f - T_{amb}}{T_{f,0} - T_{amb}}$ [%]
—●—	1	0.30	1.20	1.20	1585.57	1585.51	1578.87	0.14	99.81
—●—	2	0.35	1.74	2.94	1585.57	1585.36	1575.69	0.21	99.72
—●—	3	0.40	20.22	23.16	1585.42	1580.78	1549.46	0.86	98.87
—●—	4	0.45	55.08	78.24	1582.73	1554.90	1489.52	2.72	96.44
—●—	5	0.50	94.82	173.06	1567.80	1495.05	1399.15	6.20	91.90
—●—	6	0.55	134.35	307.41	1527.10	1407.24	1289.80	11.20	85.36
—●—	7	0.60	172.87	480.28	1453.82	1302.90	1176.19	17.32	77.35
—●—	8	0.65	211.08	691.37	1351.81	1192.13	1068.66	24.05	68.54
—●—	9	0.70	249.94	941.30	1232.11	1082.18	971.16	30.93	59.55
—●—	10	0.75	290.22	1231.53	1107.33	977.91	884.10	37.58	50.86
—●—	11	0.80	332.51	1564.04	987.73	882.47	806.85	43.72	42.82
—●—	12	0.85	377.26	1941.30	879.74	797.65	738.75	49.21	35.65
—●—	13	0.90	424.90	2366.20	786.27	724.06	679.30	53.97	29.42
—●—	14	0.95	475.83	2842.03	707.71	661.45	628.01	58.01	24.13
—●—	15	1.00	530.54	3372.57	642.98	608.99	584.28	61.40	19.71

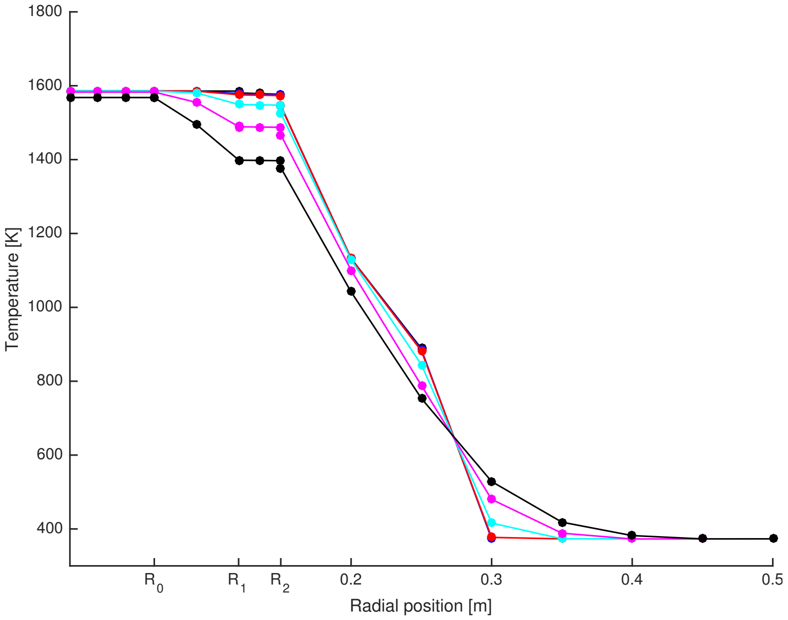

Figure 9.3: Temperature profiles for the first 5 increments of the spherical shell.

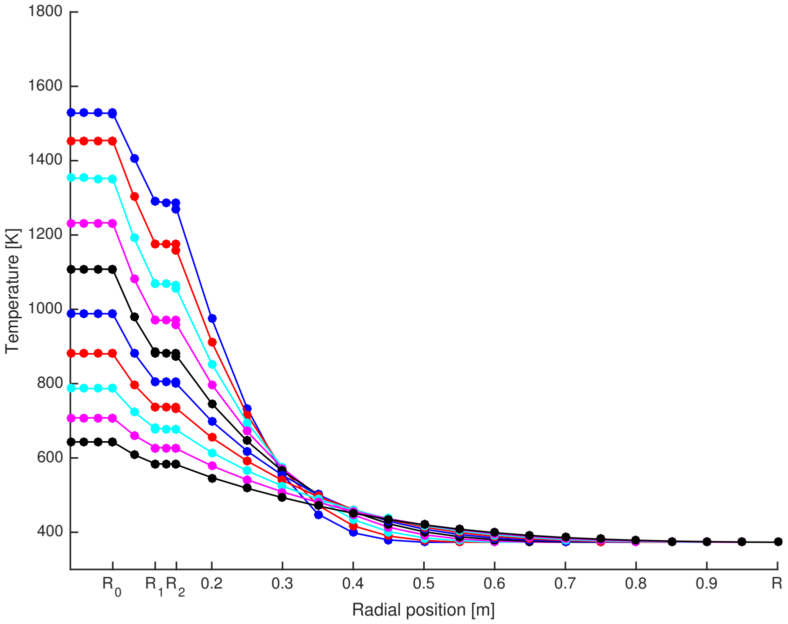

Figure 9.4: Temperature profiles for the final 10 increments of the spherical shell.

9.3 Combustion: 1 $\frac{\text{mol}}{\text{s}}$ of O_2

To investigate the effect of increasing the amount of limiting reactant, the model was computed for 1 $\frac{\text{mol}}{\text{s}}$ of O_2 fed to the furnace. Initial temperatures in all layers were set to 373.15 K. The temperature profiles are shown in figure 9.5 and figure 9.6. The only significant difference is that the maximum temperature is reached much faster, as is expected for a greater heat production term. The evaporation times, on the other hand, does not differ by more than 5-6 minutes. The maximum temperature is 1603 K, not even 20 degrees more than for the feed containing 0.2 $\frac{\text{mol}}{\text{s}}$ of O_2.

Table 9.3: Explanation to the colored graphs of figure 9.3 and 9.4. R_o is the outer radius of dry sand, t_{vap} is evaporation time for each shell and t_{tot} is total time after initialization. The temperature in the furance is T_f, after each shell incrementation and the percentage of maximum furnace temperature, $\frac{T_f}{T_{f,max}}$ is given in the last column.

		#	R_o [m]	t_{vap} [min]	t_{tot} [min]	T_f [K]	$\frac{T_f}{T_{f,max}}$ [%]
Figure 9.5:	—●—	1	0.30	23.74	23.74	1538.15	95.91
	—●—	2	0.35	49.39	73.12	1595.70	99.50
	—●—	3	0.40	80.56	153.69	1599.11	99.71
	—●—	4	0.45	112.30	265.98	1600.45	99.79
	—●—	5	0.50	143.43	409.41	1601.32	99.85
Figure 9.6:	—●—	6	0.55	174.07	583.48	1601.92	99.88
	—●—	7	0.60	204.61	788.09	1602.35	99.91
	—●—	8	0.65	235.42	1023.51	1602.68	99.93
	—●—	9	0.70	266.72	1290.23	1602.94	99.95
	—●—	10	0.75	298.64	1588.87	1603.15	99.96
	—●—	11	0.80	331.23	1920.10	1603.32	99.97
	—●—	12	0.85	364.50	2284.60	1603.47	99.98
	—●—	13	0.90	398.44	2683.04	1603.59	99.99
	—●—	14	0.95	433.04	3116.09	1603.70	99.99
	—●—	15	1.00	468.28	3584.37	1603.79	100.00

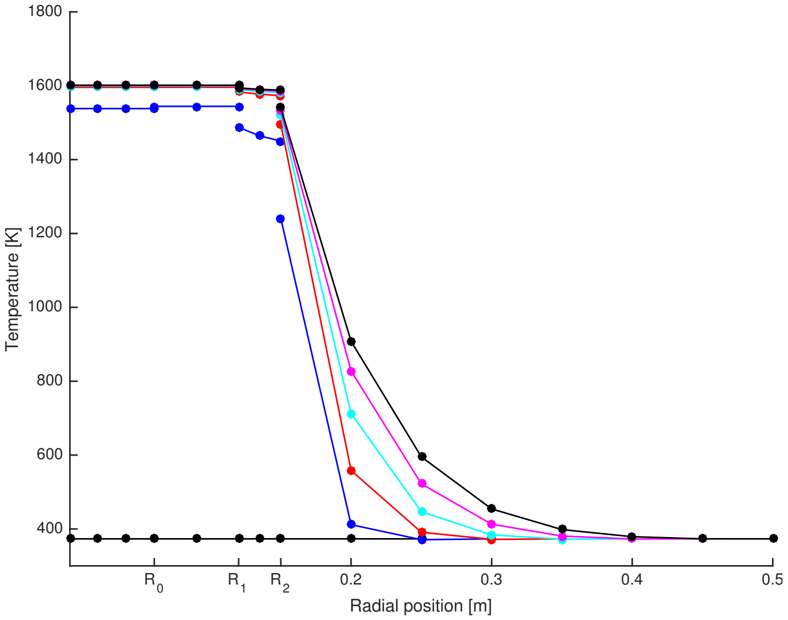

Figure 9.5: Temperature profiles for the first 5 increments of the spherical shell.

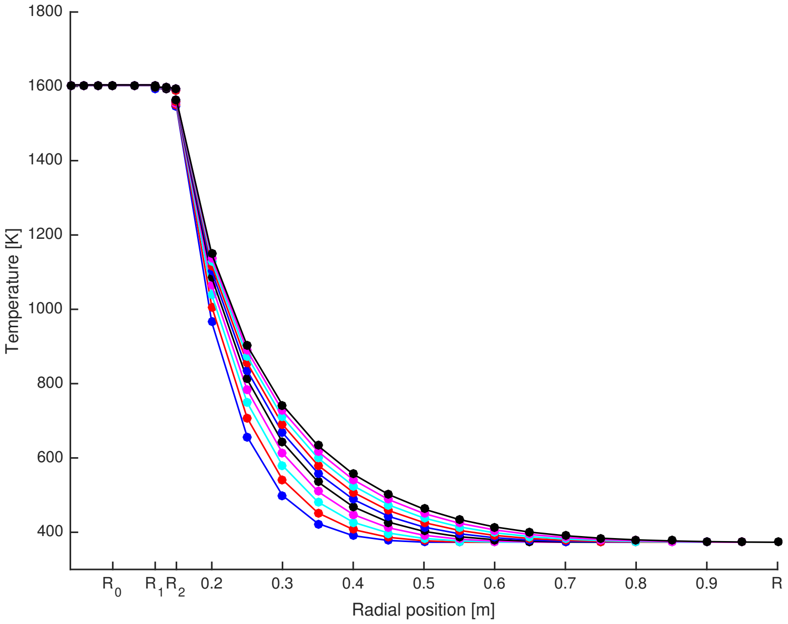

Figure 9.6: Temperature profiles for the final 10 increments of the spherical shell.

9.4 Combustion: 0.04 $\frac{\text{mol}}{\text{s}}$ of O_2

Increasing the limiting reactant by a factor of 5 did not result in large differences, thus it is interesting to investigate the impact of decreasing the oxygen by a factor of 5. Initial temperatures were set to 373.15, as for the other cases. The temperature profiles are shown in figure 9.7. All nodes are increasing in temperature during the entire simulation time. In other words, the combustion reaction does not produce enough heat for the furnace to reach a maximum temperature. The total simulation time has increased to 66 hours, and the final furnace temperature is 1490 K, almost a 100 K less than for 0.2 $\frac{\text{mol}}{\text{s}}$ of O_2.

Table 9.4: Explanation to the colored graphs of figure 9.7. R_o is the outer radius of dry sand, t_{vap} is evaporation time for each shell, t_{tot} is total time after initialization and the temperature in the furance is T_f after each shell incrementation.

	#	R_o [m]	t_{vap} [min]	t_{tot} [min]	T_f [K]
—●—	1	0.30	47.31	47.31	728.11
—●—	2	0.35	78.82	126.13	973.09
—●—	3	0.40	109.17	235.30	1151.33
—●—	4	0.45	139.09	374.39	1262.01
—●—	5	0.50	169.06	543.45	1329.67
—●—	6	0.55	199.25	742.70	1373.37
—●—	7	0.60	229.87	972.56	1403.61
—●—	8	0.65	261.09	1233.65	1425.70
—●—	9	0.70	293.02	1526.67	1442.44
—●—	10	0.75	325.70	1852.37	1455.41
—●—	11	0.80	359.13	2211.50	1465.65
—●—	12	0.85	393.28	2604.78	1473.84
—●—	13	0.90	428.14	3032.93	1480.46
—●—	14	0.95	463.68	3496.61	1485.88
—●—	15	1.00	499.87	3996.48	1490.35

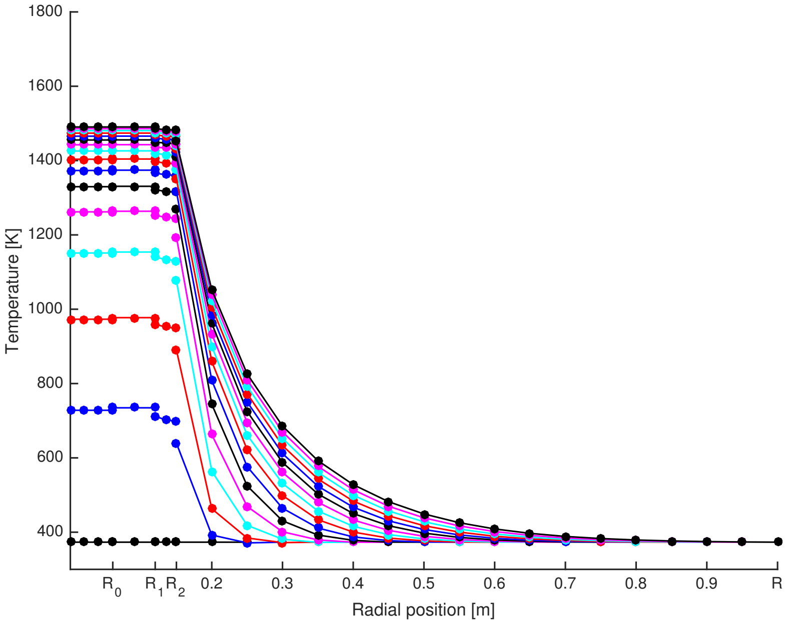

Figure 9.7: Temperature profiles for all increments of the spherical shell.

Chapter 10

Conclusion

10.1 Temperature profiles

The proposed matrix method for solution, suggested by Professor Tore Haug-Warberg, results in temperature profiles showing trends that are consistent with the expectations for heating and cooling. The variable substitution relating a slab and sphere gives very simple expressions for the computation of temperature evolution at inner nodes, but the interface equations remain functions of radial positions. The definition of the conditon at the moving boundary, with both the heat flux and the required time as unknowns, results in the need for symbolic solution of the entire matrix system. And then in turn a new solution corresponding to the obtained time and heat flux. Due to the solution being calculated from eigenvalues and eigenvectors, the implemented method in MATLAB is not very fast. The solution is also completely determined by how the discretization is done. In the current solution, only 3 nodes are defined in all the stationary materials. To more correctly model the conduction, more nodes in all layers should be considered. The method used in this report gives a semi-analytical expression for the derivative with respect to time for the transient response in the given temperature node. A temperature profile as a function of time, for each node, could be generated and perhaps used to compute a first order time constant for the temperature evolution in the furnace. The total simulation times for complete vaporization of the moisture is much longer than expected, although experiments have not been carried out for comparison. The long simulation time for the heating is probably due to the low and uniform initial condition of 373.15 K, which means that a temperature gradient in the sand must evolve before water will evaporate. The conduction after full heating is much faster because of the existing temperature gradient in the sand. Without validation of the results in this report, the temperature profiles can only confirm that the matrix method solved by eigenvalues and eigenvectors give results that have reasonable physical meaning.

10.2 Time constant considerations for this particular problem

The characteristic time can be defined as the time it takes the furnace to reach a certain percentage of its maximum temperature for the heating process. As the heating model contains a heat source, involves diffusion through multiple layers, is radially dependent, and has a moving boundary, non of the methods in chapter 8 for time constant calculations, can be applied directly. The inspiring idea for the problem was the smelting of iron by ancient techniques. From this perspective, the characteristic time should be the time range necessary to achieve an appropriate temperature for operation. The definition could involve the ratio of current temperature and the maximum obtainable temperature for the given combustion reaction, as was calculated and tabulated in the results. A perhaps more thermodynamically valid quantity for calculation of a characteristic time is energy. The ratio could be current energy in the system divided by total amount of produced energy, or maximum achievable energy for the system. Alternatively, it could be defined as a percentage of the current temperature to the adiabatically obtainable flame temperature. If so, the maximum temperature would be constant and the percentages comparable for differnent amounts of reactants.

The characteristic time for heat loss by diffusion is another interesting calculation. After the heat production ends and heat is lost to the surrounding sand, the temperature in the furnace will decrease. A time constant could be defined in a similar manner to the one for heat loss by convection. In such a case the time constant corresponds to a ratio of temperature differences. As was described by the derivation of an effective time constant for the diffusion of heat, the temperature will not decrease uniformly in the furnace. The nature of the equations are highly dependent on radius and the furnace in particular develops a temperature gradient that supports the calculation of a radially dependent time constant. Unfortunately, the inclusion of multiple layers and unknown boundary conditions make an analogous calculation to the one in section 8.2, extremely cumbersome, if not impossible.

10.3 Recommendations for further work

A fullfilling determination of a characteristic time, neither for heating by combustion of coal in a spherical furnace, nor for conduction of heat after concluded combustion, was obtained in the report. Given the temperature profiles obtained from solving the matrix equation systems, an energy-based characteristic time could be investigated for the process of heating and cooling of the control volume. The discretization could be redone for smaller radial increments, and larger number of nodes in each material. More complex description of the combustion of coal could be formulated, at least the inclusion of temperature dependent heat capacities. Perhaps another numerical method, avoiding the variable substitution of temperature between slab and sphere should be considered. At least for comparative reasons, and a way to strengthen the results obtained in this report. Experimental procedures for investigation of the temperature evolution in a sphere should be formulated and tested in order to validate any of the results given in this report, as they are now completely unvalidated.

Bibliography

R. Collins. The choice of an effective time constant for diffusive processes in finite systems. *Journal of Physics D: Applied Physics*, 1980.

J. Crank. *The Mathematics of Diffusion.* Oxford University Press, 2. edition, 1975. p. 90-95.

B.B. Mikic M.G. Cooper and M.M. Yovanovich. Thermal contact conductance. *Journal of Heat and Mass Transfer*, 1968.

Nomenclature

$\dot{m}$	Mass flow	kgs^{-1}
$\dot{n}$	Molar flow	$mols^{-1}$
Λ	Diagonal eigenvalue matrix	
A	Surface area	m^2
C	Concentration	$molm^{-3}$
c_p	Specific heat capacity at constant pressure	$Jkg^{-1}K{-}1$
c_v	Specific heat capacity at constant volume	$Jkg^{-1}K{-}1$
D	Diffusion coefficient	m^2s^{-1}
E	Total energy	J
H	Enthalpy	
$h_f^{\ominus}$	Formation enthalpy	$Jmol^{-1}$
h_c	Thermal conductance contact coefficient	$Wm^{-2}K^{-1}$
k	Thermal conductivity	$Wm^{-1}K^{-1}$
L_v	Latent heat of evaporation	Jkg^{-1}
m	Mass	kg
M_m	Molar mass	$kgmol^{-1}$
M_t	Diffused amount at time t	mol
M_∞	Diffused amount after infinite time	mol
p	Pressure	$kgm^{-1}s^{-2}$
Q	Heat source	J
q	Heat flux	$Jm^{-2}s^{-1}$
R	Radius	m
r	Radius	m
S	Eigenvector matrix	

T	Temperature	K
t	Time	s
T_b	Boiling temperature	K
T_{ref}	Reference temperature	K
t_{vap}	Evaporation time	s
U	Internal energy	J
V	Volume	m^3
f	External forces vector	
u	Velocity vector	ms^{-1}

Greek symbols

α	Thermal diffusivity	m^2s^{-1}
ϕ	Porosity	
Π	Stress tensor	$kgm^{-1}s^{-2}$
ρ	Density	kgm^{-3}
τ	Time constant	s

Other symbols

∇	Vector differential operator

Subscripts

0	Iron core
1	Furnace
2	Steel wall
3	Dry sand
C	Carbon
f	Furnace
Fe	Iron
i	Node subscript
N_2	Nitrogen gas
O_2	Oxygen gas
s	Dry sand
w	Wall

Appendix A

Derivation of the effective time constant

The following Laplace formulas were used in the transformation of the heat equation

$$\mathcal{L}[c_1 f(t) + c_2 g(t)] = c_1 F(s) + c_2 G(s) \tag{A.1}$$

$$\mathcal{L}[a] = \frac{a}{s} \tag{A.2}$$

$$\mathcal{L}\left[\frac{\partial f(r,t)}{\partial t}\right] = sF(r,s) - f(r,0) \tag{A.3}$$

$$\mathcal{L}\left[\frac{\partial^2 f(r,t)}{\partial r^2}\right] = \frac{d^2 F(r,s)}{dr^2} \tag{A.4}$$

The heat equation in spherical coordinates is

$$\frac{\partial T(r,t)}{\partial t} = \alpha \left[\frac{\partial^2 T(r,t)}{\partial r^2} + \frac{2}{r}\frac{\partial T(r,t)}{\partial r}\right] \tag{A.5}$$

By applying a variable substitution $U(r,t) = rT(r,t)$ (in Laplace coordinates $U(r,s) = rT(r,s)$) the heat equation takes the form of the slab equation

$$\frac{\partial U(r,t)}{\partial t} = \alpha \frac{\partial^2 U(r,t)}{\partial r^2} \tag{A.6}$$

The following conditions yield for the PDE

$$T(r,0) = T_0 \quad U(r,0) = rT_0 \tag{A.7}$$

$$T(0,t) = 0 \quad U(0,t) = 0 \tag{A.8}$$

$$T(R,t) = T_R \quad U(R,t) = RT_R \tag{A.9}$$

Laplace transformation yields

$$su(r,s) - U(r,0) = \alpha \frac{d^2 u(r,s)}{dr^2} \tag{A.10}$$

which is rearranged into the general form of a non-homogeneous second order linear ordinary differential equation with constant coefficients

$$\frac{d^2u(r,s)}{dr^2} - \frac{s}{\alpha}u(r,s) = -\frac{rT_0}{\alpha} \tag{A.11}$$

with boundary conditions

$$u(0,s) = 0 \tag{A.12}$$

$$u(R,s) = \frac{RT_R}{s} \tag{A.13}$$

Solution to the non-homogeneous second order linear ordinary differential equation is found by linear combination of the homogeneous solution and the particular solution.

The homogenous solution is found from the characteristic equation

$$x^2 - \frac{s}{\alpha} = 0 \tag{A.14}$$

$$x = \pm\sqrt{\frac{s}{\alpha}} \tag{A.15}$$

$$u_H(r) = c_1 \mathrm{e}^{\left(r\sqrt{\frac{s}{\alpha}}\right)} + c_2 \mathrm{e}^{\left(-r\sqrt{\frac{s}{\alpha}}\right)} \tag{A.16}$$

Based on the non-homogeneous term, a guess for a possible particular solution is a first order polynomial in r

$$u_P(r) = A_1 r + A_0 \tag{A.17}$$

$$u'_P(r) = A_1 \tag{A.18}$$

$$u''_P(r) = 0 \tag{A.19}$$

Inserted in the ODE

$$-\frac{A_1 s}{\alpha}r - \frac{A_0 s}{\alpha} = -\frac{rT_0}{\alpha} \tag{A.20}$$

Thus, $A_0 = 0$ and $A_1 = \frac{T_0}{s}$, and the general solution becomes

$$u(r,s) = u_H + u_P = c_1 \mathrm{e}^{\left(r\sqrt{\frac{s}{\alpha}}\right)} + c_2 \mathrm{e}^{\left(-r\sqrt{\frac{s}{\alpha}}\right)} + \frac{T_0}{s}r \tag{A.21}$$

The coefficients, c_1 and c_2, are solved for by applying the boundary conditions

$$u(0,s) = c_1 + c_2 = 0 \tag{A.22}$$

$$u(R,s) = c_1 \mathrm{e}^{aR} + c_2 \mathrm{e}^{-aR} + \frac{T_0 R}{s} = \frac{T_R R}{s} \tag{A.23}$$

$$c_2\left(\mathrm{e}^{-aR} - \mathrm{e}^{aR}\right) = \frac{R(T_R - T_0)}{s} \tag{A.24}$$

$$c_2 = \frac{R(T_R - T_0)}{s\left(\mathrm{e}^{-aR} - \mathrm{e}^{aR}\right)} \tag{A.25}$$

$$c_1 = -\frac{R(T_R - T_0)}{s\left(\mathrm{e}^{-aR} - \mathrm{e}^{aR}\right)} \tag{A.26}$$

Inserting the coefficient into the general solution gives the particular solution

$$u(r,s) = \frac{R(T_R - T_0)}{s}\frac{(\mathrm{e}^{ar} - \mathrm{e}^{-ar})}{(\mathrm{e}^{aR} - \mathrm{e}^{-aR})} + \frac{T_0}{s}r \tag{A.27}$$

Definition of hyperbolic sine

$$\sinh(x) = \frac{\mathrm{e}^{x} - \mathrm{e}^{-x}}{2} \tag{A.28}$$

The solution to the ODE fullfilling the boundary conditions

$$u(r,s) = \frac{R(T_R - T_0)}{s}\frac{\sinh(ar)}{\sinh(aR)} + \frac{T_0}{s}r, \quad a = \sqrt{\frac{s}{\alpha}} \tag{A.29}$$

Back-substitution to the $T(r,s)$-variable

$$T(r,s) = \frac{R(T_R - T_0)}{rs}\frac{\sinh(ar)}{\sinh(aR)} + \frac{T_0}{s}, \quad a = \sqrt{\frac{s}{\alpha}} \tag{A.30}$$

A more thorough derivation of the time constant:

The stationary solution

$$T_e = \lim_{s\to 0} sT(r,s) = \lim_{s\to 0} \frac{R(T_R - T_0)}{r}\frac{\sinh(r\sqrt{\frac{s}{\alpha}})}{\sinh(R\sqrt{\frac{s}{\alpha}})} + T_0 \tag{A.31}$$

$$= \lim_{s\to 0} \frac{R(T_R - T_0)}{r}\frac{r\cosh(r\sqrt{\frac{s}{\alpha}})}{R\cosh(R\sqrt{\frac{s}{\alpha}})} + T_0 = T_R \tag{A.32}$$

Note that L'Hôpital's rule

$$\lim_{s\to 0}\frac{f(s)}{g(s)} = \lim_{s\to 0}\frac{f'(s)}{g'(s)} \tag{A.33}$$

was applied to the $\frac{0}{0}$-expression resulting from letting $s \to 0$ in $\sinh(r\sqrt{\frac{s}{\alpha}})$ and $\sinh(R\sqrt{\frac{s}{\alpha}})$.

For small s, the expression for $T(r,s)$ can be expanded as a serie in powers of s by applying the following relations

$$\frac{\sinh(x)}{\sinh(y)} = \sinh(x)\mathrm{csch}(y) \tag{A.34}$$

$$\sinh(x) = x + \frac{1}{6}x^3 + \frac{1}{120}x^5 + \mathrm{H.O.T} \tag{A.35}$$

$$\mathrm{csch}(y) = \frac{1}{y} - \frac{1}{6}y + \frac{7}{360}y^3 + \mathrm{H.O.T} \tag{A.36}$$

Series expansion of the hyperbolic term yields

$$
\begin{aligned}
&\sinh\left(r\sqrt{\frac{s}{\alpha}}\right)\operatorname{csch}\left(R\sqrt{\frac{s}{\alpha}}\right) = \\
&\left[\left(r\sqrt{\frac{s}{\alpha}}\right) + \frac{1}{6}\left(r\sqrt{\frac{s}{\alpha}}\right)^3 + \frac{1}{120}\left(r\sqrt{\frac{s}{\alpha}}\right)^5\right]\cdot \\
&\left[\left(R\sqrt{\frac{s}{\alpha}}\right)^{-1} - \frac{1}{6}\left(R\sqrt{\frac{s}{\alpha}}\right) + \frac{7}{360}\left(R\sqrt{\frac{s}{\alpha}}\right)^3\right] \\
&= \frac{r}{R} - \frac{rR}{6}\frac{s}{\alpha} + \frac{7rR^3}{360}\frac{s^2}{\alpha^2} + \frac{r^3}{6R}\frac{s}{\alpha} - \frac{r^3R}{36}\frac{s^2}{\alpha^2} + \frac{7r^3R^3}{6\cdot 360}\frac{s^3}{\alpha^3} \\
&+ \frac{r^5}{120R}\frac{s^2}{\alpha^2} - \frac{r^5R}{6\cdot 120}\frac{s^3}{\alpha^3} + O(s^4) \\
&= \frac{r}{R} + \left(\frac{r^3}{6R} - \frac{rR}{6}\right)\frac{s}{\alpha} + \left(\frac{7rR^3}{360} - \frac{r^3R}{36} + \frac{r^5}{120R}\right)\frac{s^2}{\alpha^2} + \left(\frac{7r^3R^3}{6\cdot 360} - \frac{r^5R}{6\cdot 120}\right)\frac{s^3}{\alpha^3} \\
&= \frac{r}{R} + \frac{r(r^2-R^2)}{6\alpha R}s + \frac{r(7R^4 - 10r^2R^2 + 3r^4)}{360\alpha^2 R}s^2 + \frac{r^3R(7R^2-3r^2)}{6\cdot 360\alpha^3}s^3
\end{aligned}
$$

Inserted in the full expression

$$
\begin{aligned}
T(r,s) = \frac{T_0}{s} + \frac{R(T_R - T_0)}{rs}\cdot& \\
\left[\frac{r}{R} + \frac{r(r^2-R^2)}{6\alpha R}s + \frac{r(7R^4 - 10r^2R^2 + 3r^4)}{360\alpha^2 R}s^2 + \frac{r^3R(7R^2-3r^2)}{6\cdot 360\alpha^3}s^3\right]&
\end{aligned}
$$

$$
T(r,s) = \frac{T_R}{s} + \frac{(T_R - T_0)(r^2 - R^2)}{6\alpha} + \frac{(T_R - T_0)(7R^4 - 10r^2R^2 + 3r^4)}{360\alpha^2}s \tag{A.37}
$$

$$
+ \frac{r^2R^2(7R^2-3r^2)}{6\cdot 360\alpha^3}s^2 = \frac{T_e}{s} + B + Cs + O(s^2) \tag{A.38}
$$

Thus,

$$
\begin{aligned}
B &= \frac{(T_R - T_0)(r^2 - R^2)}{6\alpha} \\
C &= \frac{(T_R - T_0)(7R^4 - 10r^2R^2 + 3r^4)}{360\alpha^2}
\end{aligned}
$$

yielding the expression for an effective time constant

$$
\begin{aligned}
\tau_{\text{eff}} &= -\frac{C}{B} = -\frac{(T_R - T_0)(7R^4 - 10r^2R^2 + 3r^4)}{360\alpha^2}\frac{6\alpha}{(T_R - T_0)(r^2 - R^2)} \\
&= \frac{(7R^2 - 3r^2)}{60\alpha}
\end{aligned}
$$

Appendix B

MATLAB code

B.1 main.m

```
clear all
close all
clc
%% MAIN SCRIPT FOR MULTILAYERED SPHERE
%% CASE SPECIFICATION
HEAT=0;                            % [1 or 0] for burning or conduction in furnace
nO2=0.2;                            % [mol/s] amount of oxygen in inlet air stream
%% Temperatures for air fed to the furnace
Tref=1200;                                                                    % [K]
Tair=288;                                                                     % [K]
T_amb=373.15;                                                                 % [K]
%% Radial discretization
dr0=0.020;
dr1=0.030;
dr2=0.015;
dr3=0.050;
%Interfaces:
R0=3*dr0;
R1=R0+2*dr1;
R2=R1+2*dr2;
R3=R2+2*dr3;
%Outer radius:
R=1;
%Radial vectors:
r0=dr0:dr0:R0;
r1=R0:dr1:R1;
r2=R1:dr2:R2;
r3=R2:dr3:R3;
r=R3:dr3:R;
%% Material properties and initial matrix system
[ k0,k1,k2,k3,rho0,rho1,rho2,rho3,MmC,cpFe,cpC,cpsteel,cpsand,HfCO,cpmC,...
    cpmO2,cpmN2,cpmCO,alpha0,alpha1,alpha2,alpha3,h0,h1,h2]...
    = materialProperties();
%% CONSTRUCTING COMBUSTION SYSTEM (HEAT=1)
r_heat=[R0 R1 r2 r3]';
[c00,c01,c10,c11,c13,C1] = reactionFurnace(R0,R1,...
```

```
    Tref,Tair,h0,h1,cpFe,cpmC,cpmCO,cpmO2,cpmN2,HfCO,rho0,rho1,MmC,nO2);
[ A1_H, A2_H, B1_H, b1_H ] = constructFurnace(dr2,dr3,alpha2,alpha3,...
    c00,c01,c10,c11,c13,C1,R1,R2,R3,k2,k3,h1,h2);
rk_H=r_heat;
%% CONSTRUCTING PURE CONDUCTION SYSTEM (HEAT=0)
r_cool=[r0 r1 r2 r3]';
[A1_C, A2_C, B1_C] = constructConduction(dr0,dr1,dr2,dr3,alpha0,alpha1,...
alpha2,alpha3,R0,R1,R2,R3,k0,k1,k2,k3,h0,h1,h2);
rk=r_cool;
%% User-input: burning furnace(1) or pure conduction(0)
% prompt=('Press 1 for heating or 0 for pure conduction: ');
% HEAT=input(prompt);
%% Initial temperature vectors
if HEAT==1
    U0 = [T_amb T_amb T_amb T_amb T_amb T_amb T_amb T_amb]';
    T0 = r_transf(U0,r_heat);                       % Variable substitution
    if nO2==0.2
        Tmax = 1585.57;
    elseif nO2==1
        Tmax = 1603.79;
    end
elseif HEAT==0
    if nO2==0.2
        U0 = [1585.56995 1585.56995 1585.56995 1585.57181 1585.57181...
        1585.57181 1580.68845 1578.55243 1576.84450 1545.80057 1132.27014...
        888.64388]';
    end
    T0 = r_transf(U0,r_cool);                       % Variable substitution
end
%% Evaporation data for water
Ub=373.15;                                        % Boiling temperature [K]
Hvap=2257000;                                 % Evaporation enthalpy [J/kg]
d=1000;                                                   % Density [kg/m3]
w=0.001;                                             % Water content [Vol%]
%% Allocating vectors
VapTime=zeros(1,length(r)-1);
VapOutputTable0=zeros(7,length(r)-1);
VapOutputTable1=zeros(4,length(r)-1);
VapOutputTable2=zeros(3,length(r)-1);
%% PLOT THE INITIAL CONDITION
figure(1)
hold on
if HEAT==1
    plot([0; rk(1:3)],[U0(1); U0(1); U0(1); U0(1)],'k.-',...
    rk(4:6),[U0(2); U0(2); U0(2)],'k.-',rk(7:9),U0(3:5),'k.-',...
    rk(10:end),U0(6:end),'k.-','MarkerSize',20)
elseif HEAT==0
    plot([0; rk(1:3)],[U0(1); U0(1:3)],'k.-',...
    rk(4:6),U0(4:6),'k.-',rk(7:9),U0(7:9),'k.-',...
    rk(10:end),U0(10:end),'k.-','MarkerSize',20)
end
axis([0 rk(end) 300 1800])
xlabel('Radial position [m]')
ylabel('Temperature [K]')
%% TEMPERATURE PROFILES SOLVED FOR EACH INCREMENT OF RADIUS
for k=1:(length(r)-1)
    if HEAT==1
        A1=A1_H;
```

```
    A2=A2_H;
    B1=B1_H;
    b1=b1_H;
    if length(A1)~=length(T0)
        T0=[T0(1);T0(4);T0(7:end)];
    end
elseif HEAT==0
    A1=A1_C;
    A2=A2_C;
    B1=B1_C;
    b1 = [0; 0; 0; 0; 0; 0; 0; 0];
    if length(A1)~=length(T0)
        T0=[T0(1);T0(1);T0(1);T0(2);T0(2);T0(2);T0(3:end)];
    end
end
R_outer=r(k);
R_next=r(k+1);
Tb=r_transf(Ub,R_next);                                 %Tb transformation
Qevap=Hvap*d*w*Vol(R_outer,R_next);                    %Evaporation energy
[ A,S,D ] = constructMatrixA( A1,A2,B1 ); %Eigenvalues and eigenvectors
% Solving for required evaporation time and heat flux
[time,qflux]=determineVaptime(S,D,Qevap,Tb,T0,A2,b1,k3,R_outer,dr3,HEAT);
time=double(time);
VapTime(k)=time;
qflux=double(qflux);
if time==Inf
sprintf('Infinite time required to increase from %g to %g',r(k),r(k+1))
    break
end
% Temperature solution at time
x1 = nodeSolve(S,D,T0,A2,b1,k3,R_outer,dr3,qflux,time,HEAT);
x2 = ghostSolve( B1,x1,qflux,dr3,R_outer,k3,HEAT );
sprintf('Radius of dry sphere has increased to: %g',R_next)
%Temperature and radial position
rk = [rk; R_next];
rk_H=[rk_H; R_next];
Tk = [x1; x2(end)];
%Backtransformation to spherical temperatures
if HEAT==1
    Uk = r_backtransf(Tk,rk_H);
    Uk = [Uk(1);Uk(1);Uk(1);Uk(1);Uk(2);Uk(2);Uk(2);Uk(3:end)];
    if nO2==0.2 || nO2==1
    % OUTPUT TABLES:
    VapOutputTable1(:,k)=[time/60, sum(VapTime(1:k))/60, Uk(2),...
        Uk(2)/(Tmax)*100];
    elseif nO2==0.04
        VapOutputTable2(:,k)=[time/60, sum(VapTime(1:k))/60, Uk(2)];
    end
elseif HEAT==0
    Uk = r_backtransf(Tk,rk);
    Uk = [Uk(1); Uk];
    T_mf=sum(Uk(5:7))/length(Uk(5:7));        % Average temp in furnace
    % OUTPUT TABLES:
    if nO2==0.2 || nO2==1
    VapOutputTable0(:,k)=[time/60, sum(VapTime(1:k))/60,...
        Uk((length(r0)+2):(length(r0)+length(r1)+1))',...
        100-((T_mf)/(U0(6)))*100,((T_mf-T_amb)/(U0(6)-T_amb))*100];
    end
```

```
    end
    %% PLOT OF SOLUTION
    plotStyle = ...
        {'b.-','r.-','c.-','m.-','k.-','b.-','r.-','c.-','m.-','k.-'};
    if nO2~=0.04 && k<=5
    figure(1)
    hold on
    plot([0; rk(1:3)],Uk(1:4),plotStyle{k},rk(4:6),Uk(5:7),plotStyle{k},...
    rk(7:9),Uk(8:10),plotStyle{k},rk(10:end),Uk(11:end),plotStyle{k},...
    'MarkerSize',20)
    axis([0 rk(end) 300 1800])
    xlabel('Radial position [m]')
    ylabel('Temperature [K]')
    elseif nO2~=0.04 && (k>5)
    figure(2)
    hold on
    plot([0; rk(1:3)],Uk(1:4),plotStyle{k-5},rk(4:6),Uk(5:7),...
        plotStyle{k-5},rk(7:9),Uk(8:10),plotStyle{k-5},rk(10:end),...
        Uk(11:end),plotStyle{k-5},'MarkerSize',20)
    axis([0 rk(end) 300 1800])
    xlabel('Radial position [m]')
    ylabel('Temperature [K]')
    elseif nO2==0.04
    plotStyle = {'b.-','r.-','c.-','m.-','k.-','b.-','r.-',...
        'c.-','m.-','k.-','b.-','r.-','c.-','m.-','k.-'};
    figure(1)
    hold on
    plot([0; rk(1:3)],Uk(1:4),plotStyle{k},rk(4:6),Uk(5:7),plotStyle{k},...
    rk(7:9),Uk(8:10),plotStyle{k},rk(10:end),Uk(11:end),plotStyle{k},...
    'MarkerSize',20)
    axis([0 rk(end) 300 1800])
    xlabel('Radial position [m]')
    ylabel('Temperature [K]')
    end
    if k ~= (length(r)-1)
    %% New initial state and increment of the matrix system:
        T0 = r_transf(Uk(2:end),rk);
        [A1_H,A2_H,B1_H,b1_H]=matIncr(A1_H,A2_H,B1_H,b1,alpha3,dr3,R_next);
        [A1_C,A2_C,B1_C]=matIncr(A1_C,A2_C,B1_C,b1,alpha3,dr3,R_next);
    %% USER-INPUT:
%     prompt=('Press 1 for heating or 0 for pure conduction: ');
%     HEAT=input(prompt);
    end
end
%% WRITING OUTPUT
%% HEATING
%% nO2=0.2mol/s:
if HEAT==1 && nO2==0.2
%% FIGURES
figure(1)
xt={'R_0' ; 'R_1' ; 'R_2' ; R2+dr3:2*dr3:R-dr3 ; 'R'} ;
set(gca,'xtick',[R0 R1 R2 R2+dr3:2*dr3:R]);
set(gca,'xticklabel',xt);
print('..\..\REPORT\FIGURES\TemperatureProfileHeating1','-depsc')
figure(2)
xt={'R_0' ; 'R_1' ; 'R_2' ; R2+dr3:2*dr3:R-dr3 ; 'R'} ;
set(gca,'xtick',[R0 R1 R2 R2+dr3:2*dr3:R]);
set(gca,'xticklabel',xt);
```

```
print('..\..\REPORT\FIGURES\TemperatureProfileHeating2','-depsc')
%% VAPOUTPUT
fileID = fopen('VapTable1.txt','w');
fprintf(fileID,'%6s & %6s & %6s & %6s \\\\','$t_{vap}$','$t_{tot}$',...
'$T_{f}$','$\frac{T_f}{T_{f,max}}$');
fprintf(fileID,'%6s & %6s & %6s & %6s\\\\','{[min]}','[min]','[K]','[\%]');
fprintf(fileID,'%12.2f & %12.2f & %12.2f & %12.2f \\\\',VapOutputTable1);
fclose(fileID);
disp('Output written for HEAT=1, nO2=0.2')
%% nO2=1mol/s:
elseif HEAT==1 && nO2==1
%% FIGURES
figure(1)
xt={'R_0' ; 'R_1' ; 'R_2' ; R2+dr3:2*dr3:R-dr3 ; 'R'} ;
set(gca,'xtick',[R0 R1 R2 R2+dr3:2*dr3:R]);
set(gca,'xticklabel',xt);
print('..\..\REPORT\FIGURES\TemperatureProfileHeating1for1mols','-depsc')
figure(2)
xt={'R_0' ; 'R_1' ; 'R_2' ; R2+dr3:2*dr3:R-dr3 ; 'R'} ;
set(gca,'xtick',[R0 R1 R2 R2+dr3:2*dr3:R]);
set(gca,'xticklabel',xt);
print('..\..\REPORT\FIGURES\TemperatureProfileHeating2for1mols','-depsc')
%WRITE OUTPUT TO LATEX READABLE FILE
fileID = fopen('VapTable1for1mols.txt','w');
fprintf(fileID,'%6s & %6s & %6s & %6s \\\\','$t_{vap}$','$t_{tot}$',...
'$T_{f}$','$\frac{T_f}{T_{f,max}}$');
fprintf(fileID,'%6s & %6s & %6s & %6s\\\\','{[min]}','[min]','[K]','[\%]');
fprintf(fileID,'%12.2f & %12.2f & %12.2f & %12.2f \\\\',VapOutputTable1);
fclose(fileID);
disp('Output written for HEAT=1, nO2=1')
%
elseif HEAT==1 && nO2==0.04
%% FIGURES
figure(1)
xt={'R_0' ; 'R_1' ; 'R_2' ; R2+dr3:2*dr3:R-dr3 ; 'R'} ;
set(gca,'xtick',[R0 R1 R2 R2+dr3:2*dr3:R]);
set(gca,'xticklabel',xt);
print('..\..\REPORT\FIGURES\TemperatureProfileHeating2','-depsc')
%WRITE OUTPUT TO LATEX READABLE FILE
fileID = fopen('VapTable2.txt','w');
fprintf(fileID,'%6s & %6s & %6s \\\\','$t_{vap}$','$t_{tot}$',...
'$T_{f}$');
fprintf(fileID,'%6s & %6s & %6s \\\\','{[min]}','[min]','[K]');
fprintf(fileID,'%12.2f & %12.2f & %12.2f \\\\',VapOutputTable2);
fclose(fileID);
disp('Output written for HEAT=1, nO2=0.01')
%
%% COOLING
elseif HEAT==0 && nO2==0.2
%% nO2=0.2mol/s:
%% FIGURES
figure(1)
xt={'R_0' ; 'R_1' ; 'R_2' ; R2+dr3:2*dr3:R-dr3 ; 'R'} ;
set(gca,'xtick',[R0 R1 R2 R2+dr3:2*dr3:R]);
set(gca,'xticklabel',xt);
print('..\..\REPORT\FIGURES\TemperatureProfileCooling1','-depsc')
figure(2)
xt={'R_0' ; 'R_1' ; 'R_2' ; R2+dr3:2*dr3:R-dr3 ; 'R'} ;
```

```
set(gca,'xtick',[R0 R1 R2 R2+dr3:2*dr3:R]);
set(gca,'xticklabel',xt);
print('..\..\REPORT\FIGURES\TemperatureProfileCooling2','-depsc')
% %WRITE OUTPUT TO LATEX READABLE FILE
fileID = fopen('VapTable00.txt','w');
fprintf(fileID,'%6s & %6s & %6s & %6s & %6s & %6s & %6s\\\\','$t_{vap}$',...
'$t_{tot}$',' ','$T_{f}$',' ','$\frac{\overline{T}_{f}}{T_{f,0}}$',...
'$\frac{\overline{T}_{f}-T_{amb}}{T_{f,0}-T_{amb}}$');
fprintf(fileID,'%6s & %6s & %6s & %6s & %6s & %6s & %6s \\\\','{[min]}',...
'{[min]}',' ','[K]',' ','[\%]','[\%]');
fprintf(fileID,'%12.2f&%12.2f&%12.2f&%12.2f&%12.2f&%12.2f&%12.2f\\\\',...
VapOutputTable0);
fclose(fileID);
disp('Output written for HEAT=0, nO2=0.2')
%
end
```

B.2 materialProperties.m

```
function [ k0,k1,k2,k3,rho0,rho1,rho2,rho3,MmC,cpFe,cpC,cpsteel,cpsand,...
    HfCO,cpmC,cpmO2,cpmN2,cpmCO,alpha0,alpha1,alpha2,alpha3,h0,h1,h2]...
    = materialProperties()
    % Constant material properties:
    %% Thermal conductivity
    k0=70;                                                    % Iron [W/mK]
    k1=0.2;                                      % Carbon(graphite) [W/mK]
    k2=30;                                  % Stainless steel (wall) [W/mK]
    k3=0.25;                                              % Dry sand [W/mK]
    %% Density
    rho0=7860;                                               % Iron [kg/m3]
    rho1= 2260;                                  % Carbon(graphite) [kg/m3]
    rho2=7820;                                              % Steel [kg/m3]
    rho3=1400;                                           % Dry sand [kg/m3]
    %% Molar mass
    MmC = 0.012011;                                       % Carbon [kg/mol]
    %% Specific heat capacity (mass basis)
    cpFe=450;                                                % Iron [J/kgK]
    cpC=710;                                     % Carbon(graphite) [J/kgK]
    cpsteel=490;                                            % Steel [J/kgK]
    cpsand=800;                                          % Dry soil [J/kgK]
    %% Formation enthalpy (from Yanaf):
    % At 1200 K
    HfCO = -113217;                               % Carbon monoxide [J/mol]
    % At 1500 K
    %HfCO = -115229;                              % Carbon monoxide [J/mol]
    %% Specific heat capacity (molar basis)
    % At 1200 K (from Yanaf):
    cpmC = 22.766;                                        % Carbon [J/molK]
    cpmO2 = 35.667;                                   % Oxygen gas [J/molK]
    cpmN2 = 33.723;                                 % Nitrogen gas [J/molK]
    cpmCO = 34.175;                              % Carbon monoxide [J/molK]
    % At 1500 K (from Yanaf):
%     cpmC = 23.904;                                      % Carbon [J/molK]
%     cpmO2 = 36.544;                                 % Oxygen gas [J/molK]
%     cpmN2 = 34.844;                               % Nitrogen gas [J/molK]
```

```
%     cpmCO = 35.217;                            % Carbon monoxide [J/molK]
    %% Thermal diffusivity
    alpha0=k0/(rho0*cpFe);                                    % Iron [m2/s]
    alpha1=k1/(rho1*cpC);                                   % Carbon [m2/s]
    alpha2=k2/(rho2*cpsteel);                           % Steel wall [m2/s]
    alpha3=k3/(rho3*cpsand);                              % Dry sand [m2/s]
    %% Interface contact conductance coefficient
    h0=1000;                                         % Iron-Furnace [W/m2K]
    h1=1000;                                         % Furnace-Wall [W/m2K]
    h2=100;                                         % Wall-Dry sand [W/m2K]
end
```

B.3 reactionFurnace.m

```
function [c00,c01,c10,c11,c13,C1] = reactionFurnace(R0,R1,...
    Tref,Tair,h0,h1,cpFe,cpmC,cpmCO,cpmO2,cpmN2,HfCO,rho0,rho1,MmC,mO2)
    % Calculation of coefficients for the temperature evolution of the iron
    % core and the furnace, when combustion takes place
    A0=4*pi*R0*R0;                         % Surface area of iron core [m2]
    V0=(4*pi*R0*R0*R0)/3;                        % Volume of iron core [m3]
    A1=4*pi*R1*R1;                           % Surface area of furnace [m2]
    V1=(4*pi*(R1^3-R0^3))/3;                       % Volume of furnace [m3]
    VC=0.9*V1;                           % Volume of carbon in furnace [m3]
    mN2 = 3.728*mO2;                             % Molar flow of N2 [mol/s]
    mCO = 2*mO2;                                 % Molar flow of CO [mol/s]
    %% Coefficients:
    c00=(h0*A0)/(rho0*V0*cpFe);
    c01=(h0*A0*R0)/(rho0*V0*cpFe*R1);
    c10=(MmC*h0*A0*R1)/(rho1*VC*cpmC*R0);
    c11=(MmC*(-mCO*cpmCO-mN2*cpmN2-h0*A0-h1*A1))/(rho1*VC*cpmC);
    c13=(MmC*h1*A1)/(rho1*VC*cpmC);
    C1=(((mO2*cpmO2*(Tair-Tref)+mN2*cpmN2*(Tair)-...
        (mCO*HfCO)+mCO*cpmCO*Tref))*MmC*R1)/(rho1*VC*cpmC);
end
```

B.4 constructFurnace.m

```
function [ A1, A2, B1, b1 ] = constructFurnace(dr2,dr3,alpha2,alpha3,...
                              c00,c01,c10,c11,c13,C1,R1,R2,R3,kw,ks,h1,h2)
% The initial set of matrices A1, A2, B1 and vector b1 for the case of a
% burning furnace
%% COEFFICIENTS
    c2=alpha2/(dr2*dr2);
    c3=alpha3/(dr3*dr3);
    %
    c21=(2*h1*dr2*(R1-dr2))/(kw*R1);
    c24=(R1-dr2)/(R1+dr2);
    c64=(R2+dr2)/(R2-dr2);
    c65=(2*h2*dr2*(R2+dr2))/(kw*R2);
    c75=(2*h2*dr3*(R2-dr3))/(ks*R2);
    c79=(R2-dr3)/(R2+dr3);
    c119=(R3+dr3)/(R3-dr3);
    %
```

```
    %% MATRICES
    A1 = [-c00  c01   0     0     0     0     0     0;
          c10  c11  c13   0     0     0     0     0;
          0    0    -2*c2 c2    0     0     0     0;
          0    0    c2    -2*c2 c2    0     0     0;
          0    0    0     c2    -2*c2 0    0     0;
          0    0    0     0      0    -2*c3 c3    0;
          0    0    0     0      0    c3    -2*c3 c3;
          0    0    0     0      0    0     c3   -2*c3];
    %
    A2 = [0  0  0  0;
          0  0  0  0;
          c2 0  0  0;
          0  0  0  0;
          0  c2 0  0;
          0  0  c3 0;
          0  0  0  0;
          0  0  0  c3];
    %
    B1 = [0 c21 -c21 c24 0    0    0    0;
          0 0   0    c64 -c65 c65  0    0;
          0 0   0    0   c75  -c75 c79  0;
          0 0   0    0   0    0    c119 0];
    %
    b1 = [0; C1; 0; 0; 0; 0; 0; 0];
end
```

B.5 constructConduction.m

```
function [A1,A2,B1] = constructConduction(dr0,dr1,dr2,dr3,...
              alpha0,alpha1,alpha2,alpha3,R0,R1,R2,R3,k0,k1,k2,k3,h0,h1,h2)
% Initial set of matrices A1, A2 and B1 for pure conduction
    %% COEFFICIENTS
    a0=alpha0/dr0^2;
    a1=alpha1/dr1^2;
    a2=alpha2/dr2^2;
    a3=alpha3/dr3^2;
    %
    c31=(R0+dr0)/(R0-dr0);
    c32=(2*h0*dr0*(R0+dr0))/(k0*R0);
    c42=(2*h0*dr1*(R1-dr1))/(k1*R0);
    c46=(R0-dr1)/(R0+dr1);
    %
    c86=(R1+dr1)/(R1-dr1);
    c87=(2*h1*dr1*(R1+dr1))/(k1*R1);
    c97=(2*h1*dr2*(R1-dr2))/(k2*R1);
    c911=(R1-dr2)/(R1+dr2);
    %
    c1311=(R2+dr2)/(R2-dr2);
    c1312=(2*h2*dr2*(R2+dr2))/(k2*R2);
    c1412=(2*h2*dr3*(R2-dr3))/(k3*R2);
    c1416=(R2-dr3)/(R2+dr3);
    c1716=(R3+dr3)/(R3-dr3);
    %
    %% MATRICES
```

```
    A1=[-2*a0 a0 0 0 0 0 0 0 0 0 0 0;
        a0 -2*a0 a0 0 0 0 0 0 0 0 0 0;
        0 a0 -2*a0 0 0 0 0 0 0 0 0 0;
        0 0 0 -2*a1 a1 0 0 0 0 0 0 0;
        0 0 0 a1 -2*a1 a1 0 0 0 0 0 0;
        0 0 0 0 a1 -2*a1 0 0 0 0 0 0;
        0 0 0 0 0 0 -2*a2 a2 0 0 0 0;
        0 0 0 0 0 0 a2 -2*a2 a2 0 0 0;
        0 0 0 0 0 0 0 a2 -2*a2 0 0 0;
        0 0 0 0 0 0 0 0 0 -2*a3 a3 0;
        0 0 0 0 0 0 0 0 0 a3 -2*a3 a3;
        0 0 0 0 0 0 0 0 0 0 a3 -2*a3];
    %
    A2=[a0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 0;
        0 a0 0 0 0 0 0 0;
        0 0 a1 0 0 0 0 0;
        0 0 0 0 0 0 0 0;
        0 0 0 a1 0 0 0 0;
        0 0 0 0 a2 0 0 0;
        0 0 0 0 0 0 0 0;
        0 0 0 0 0 a2 0 0;
        0 0 0 0 0 0 a3 0;
        0 0 0 0 0 0 0 0;
        0 0 0 0 0 0 0 a3];
    %
    B1=[0 0 0 0 0 0 0 0 0 0 0 0;
        0 c31 -c32 c32 0 0 0 0 0 0 0 0;
        0 0 c42 -c42 c46 0 0 0 0 0 0 0;
        0 0 0 0 c86 -c87 c87 0 0 0 0 0;
        0 0 0 0 0 c97 -c97 c911 0 0 0 0;
        0 0 0 0 0 0 0 c1311 -c1312 c1312 0 0;
        0 0 0 0 0 0 0 0 c1412 -c1412 c1416 0;
        0 0 0 0 0 0 0 0 0 0 c1716 0];
end
```

B.6 constructMatrixA.m

```
function [ A,S,D ] = constructMatrixA( A1,A2,B1 )
% Combining A1, A2 and B1 to put the system in xdot=Ax-b form. S are the
% corresponding eigenvectors of A and D is the diagonal matrix containg the
% eigenvalues of A
    A = A1+A2*B1;
    [S,D]=eig(A);
end
```

B.7 determineVaptime.m

```
function [ time,qflux ] = determineVaptime( S,D,Q,Tb,X0,A2,b1,k,R,dr,HEAT)
    % Solving for the unknown evaporation time and heat flux required to
    % meet the evaporation energy demand of the next shell
    syms t
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
    A=4*pi*R^2;
    Y = (2*dr*(R+dr))/(k);
    if HEAT==1
        b2 = [0; 0; 0; (Y*Q)/(A*t)];
        b = A2*b2-b1;
    elseif HEAT==0
        b2 = [0; 0; 0; 0; 0; 0; 0; (Y*Q)/(A*t)];
        b = A2*b2;
    end
    %%%%%%%%%%%%%%%%%%
    c=S\b;
    Y0=S\X0;
    %%%%%%%%%%%%%%%%%%
    %% Finding q and tau
    y=sym(zeros(size(Y0)));
    for j=1:length(Y0)
        if abs(D(j,j))<(1.0e-10)
            y(j) = sym(Y0(j) - c(j)*t);
        else
           y(j) = sym(exp(D(j,j)*t)*Y0(j) + (c(j)/D(j,j)) * (1-exp(D(j,j)*t)));
        end
    end
    T=S*y;
    time = vpasolve(((R+dr)/(R-dr))*T(end-1)==Tb+(Y*Q)/(A*t),t,1000);
    qflux=Q/(A*time);
end
```

B.8 nodeSolve.m

```
function [ x1 ] = nodeSolve( S,D,X0,A2,b1,k,R,dr,q,t,HEAT)
% Solution of temperature at inner nodes given flux q and at time t
    Y=2*dr*(R+dr)/(k);
    if HEAT==1
        b2 = [0; 0; 0; Y*q];
        b = A2*b2-b1;
    elseif HEAT==0
        b2 = [0; 0; 0; 0; 0; 0; 0; (Y*q)];
        b=A2*b2;
    end
    c=S\b;
    %% SOLVING USING EIGENVALUES AND EIGENVECTORS
    % Transforming to diagonal form
    y0 = S\X0;
    y = zeros(length(y0),1);
    for j=1:length(y0)
        if abs(D(j,j))<(1.0e-10)
            y(j) = y0(j) - c(j)*t;
        else
           y(j) = exp(D(j,j)*t)*y0(j) + (c(j)/D(j,j)) * (1-exp(D(j,j)*t));
        end
    end
    % Back-transformation
    x1 = S*y;
end
```

B.9 ghostSolve.m

```
function [ x2 ] = ghostSolve( B1,x1,qflux,dr,R,k,HEAT )
% Solution of temperature at ghost nodes given qflux and for x1 being the
% solution at inner nodes
    Y=2*dr*(R+dr)/k;
    if HEAT==1
        b2 = [0; 0; 0; Y*qflux];
    elseif HEAT==0
        b2 = [0; 0; 0; 0; 0; 0; 0; (Y*qflux)];
    end
    x2 = B1*x1-b2;
end
```

B.10 matrixIncrease.m

```
function [ A1,A2,B1,b1 ] = matIncr( A1,A2,B1,b1,alpha,dr,R )
%Increasing the matrices for increased shell size
    cs=alpha/(dr*dr);
    addcol = zeros(length(A1),1);
    addcol(end)=cs;
    addrow=addcol';
%
    A1 = [A1 addcol;
          addrow -2*cs];
%
    A2 = [A2(1:size(A2,1)-1,:); zeros(1,size(A2,2)); A2(length(A2),:)];
%
    c=(R+dr)/(R-dr);
    B1 = [B1(1:size(B1,1)-1,:) zeros(size(B1,1)-1,1);
          zeros(1,size(B1,2)-1) c 0];
%
    b1=[b1; 0];
end
```

B.11 rtransf.m

```
function [T] = r_transf( U,r )
% Transformation of temperature U in sphere to T in slab
    T=U.*r;
end
```

B.12 backtransf.m

```
function [U] = r_backtransf( T,r )
% Back-transformation of temperature T in slab to U in sphere
    U=T./r;
end
```

B.13 vol.m

```
function [V]=Vol(r1,r2)
% Volume of a sherical shell between radius r1 and r2
    V=4/3*pi*(r2^3-r1^3);
end
```

B.14 effectiveTime.m

```
clear all
close all
clc
%% Time constant calculation based on diffused amount of substance
D=1.246e-07;                                          % Thermal diffusivity
R=0.12;                                           % Outer radius of furnace
%% Time constant calculated from Crank
% Fractions of diffused amount corresponding to 1-5*time constant:
M=[1-exp(-1) 1-exp(-2) 1-exp(-3) 1-exp(-4) 1-exp(-5)];
M1=0.75;
TimeTab=zeros(2,length(M));
for k=1:length(M)
    %Times for different values of M
    Mfunction = @(x)CrankExpression(x,M(k));
    x0=fzero(Mfunction,0.2);
    t=x0^2*R^2/D;
    TimeTab(:,k)=[M(k)*100 t/60];
end
M1=@(x)CrankExpression(x,M1);
x1=fzero(M1,0.2);
t1=x1^2*R^2/D;
%WRITE OUTPUT TO LATEX READABLE FILE
fileID = fopen('CrankTimeConstant.txt','w');
fprintf(fileID,'%6s&%6s\\\\','$\frac{M_t}{M_{\infty}}$ [\%]','$t$ [min]');
fprintf(fileID,'%12.2f & %12.2f \\\\',TimeTab);
fclose(fileID);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Effective time constant expression from Collins for diffusion
% Calculated from the expression obtained by the methods described by
% Collins
dr=0.01;
r=0.06:dr:R;
EffTimeTab=zeros(2,length(r));
tau=zeros(size(r));
for i=1:length(r)
    tau(i)=-(3*r(i)*r(i)-7*R^2)/(60*D);
    EffTimeTab(:,i)=[r(i) tau(i)/60];
end
%WRITE OUTPUT TO LATEX READABLE FILE
fileID = fopen('EffectiveTimeConstant.txt','w');
fprintf(fileID,'%6s & %6s \\\\','$r$ [m]', '$t$ [min]');
fprintf(fileID,'%12.2f & %12.2f \\\\',EffTimeTab);
fclose(fileID);
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Relating the radial dependent and independent time constants
```

```
tvec=ones(size(r))*t1;
rel=[tau(1)/tvec(end) tau(end)/tvec(end)];
%% Figure of effective and energy-based time constant
figure(1)
hold on
plot(r,tvec,'r.-','LineWidth',1.5)
plot(r,tau,'b.-','MarkerSize',20)
xlabel('r [m]')
ylabel('t [s]')
xlim([min(r) max(r)])
ylim([min([tau tvec]) max([tau tvec])])
print('..\..\REPORT\FIGURES\EffectiveTimeConstantComparison','-depsc')


function fun = CrankExpression(x,M)
% The sum in the expression for diffused amount according to Crank
Sum=0;
for n=1:10000
    Sum = Sum + (1/n^2*exp(-x^2*n^2*pi^2));
end
fun = (pi^2/6)*(1-M) - Sum;
end
```