

Desenvolvimento de uma Rotina Computacional para Modelagem Termoquímica de Combustíveis Sólidos

André Miranda Dantas e Pedro Paulo Miranda Dantas

PROJETO DE GRADUAÇÃO APRESENTADO AO
DEPARTAMENTO DE ENGENHARIA MECÂNICA DA
UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO PARA
OBTENÇÃO DO TÍTULO DE ENGENHEIRO MECÂNICO

Orientador: Prof. Dr. Márcio Ferreira Martins

Vitória (ES), Julho de 2011

**Desenvolvimento de uma Rotina Computacional para Modelagem
Termoquímica de Combustíveis Sólidos
apresentado ao Departamento de Engenharia Mecânica**

Este projeto de graduação trata-se da versão original
dos alunos André Miranda Dantas e Pedro Paulo Miranda Dantas.

**Desenvolvimento de uma Rotina Computacional para Modelagem
Termoquímica de Combustíveis Sólidos a ser apresentado ao
Departamento de Engenharia Mecânica da Universidade Federal
do Espírito Santo**

Trabalho de conclusão de curso
realizado por André Miranda Dantas
e Pedro Paulo Miranda Dantas em 29/06/2011.

Comissão Julgadora:

- Prof. Dr. Carlos Friedrich Löffler Neto
- Mestrando. Davi Faria Buson

Agradecimentos

Agradeço aos céus que agora alguma coisa me restitua à memória dos últimos anos. O que, outrora, me parecia aspereza da sorte, aprecio hoje como sabedoria da Providência. Enquanto a necessidade me oprimia e ameaçava aniquilar-me, crescia a vontade de lutar. E, finalmente, foi vitoriosa a vontade. Agradeço àqueles tempos o ter-me tornado forte e poder sê-lo ainda. E ainda mais agradeço o ter-me livrado do tédio da vida fácil e ter-me tirado do conforto despreocupado do ócio, para dar-me sofrimento e lançar-me na luta da vida, que aprendi a conhecer e pelo qual continuarei a lutar.

Resumo

A compreensão de processos de combustão é muitas vezes conveniente para o entendimento de muitos fenômenos presentes na engenharia. Entre estes fenômenos citamos o processo de pirólise do xisto betuminoso.

Partindo de equações físicas e termoquímicas que podem ser encontradas na literatura desenvolvemos com sucesso um modelo físico-matemático definido por equações EDP's Não-Lineares resolvidos pelos métodos iterativos de Newton e do Gradiente Biconjugado.

Com isso podemos prever o comportamento de propriedades físicas nestes processo, como a temperatura, umidade e a massa específica -frequentemente omitidas por simplificações adotadas na literatura- todas ao longo da geometria de uma partícula de combustível sólido quando submetida aos processos de secagem e pirólise.

Palavras-chave: Processos de Pirólise, secagem, EDP's, Métodos Iterativos, Gradiente Biconjugado.

Abstract

The comprehension of combustion processes is convenient to understand many phenomena of engineering. Among these phenomena we make mention of the oil shale's pyrolysis process.

With development of physical and thermochemical equations that can be found in the literature, we have successfully reached a physical and mathematical model defined by Nonlinear PDE's solved by the Iterative Methods of Newton and The Biconjugate Gradient Method.

So we can predict the behavior of physical properties in this processes, such as temperature, humidity and density - often not considered by simplifications adopted in the literature - all along the geometry of a solid fuel particle when subjected to drying and pyrolysis.

Keywords: Pyrolysis, Drying, Iterative Method of Newton, PDE's, Biconjugate Gradient Method.

Sumário

Lista de Abreviaturas	ix
Lista de Símbolos	xi
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
2 Modelo Matemático e Algoritmo de Integração	5
2.1 Considerações Físicas	5
2.2 Algoritmo Numérico	10
3 Resultados e Discussões	15
3.1 Estudo de Malha	15
3.2 Validação Numérica	16
3.3 Estudo Paramétrico	17
3.4 Influência das Propriedades Física	17
3.5 Outros Casos	18
4 Conclusões	25
A Anexos	27
A.1 equacoes.m	27
A.2 jac.m	30
A.3 massaN.m	32
Referências Bibliográficas	35

Lista de Abreviaturas

EDPNL	Equações Diferenciais Parciais Não-Lineares
GLP	Gás liquefeito de petróleo
MIN	Método Iterativo de Newton
MGBC	Método do Gradiente Biconjugado
TGA	Analisador Termogravimétrico

Lista de Símbolos

Letras Latinas

A_i	Fator pré-exponencial de Arrhenius [s^{-1}]
C_p	Calor específico do xisto betuminoso [$J.Kg^{-1}K^{-1}$]
E_i	Energia de ativação [$J.mol^{-1}$]
h	Coefficiente de convecção [$J.m^{-2}.s.K$]
m	Massa [Kg]
q	Transferência de calor [$W.m^{-2}$]
Q	Termo fonte [$W.m^{-2}.K^{-1}$]
R_g	Constante universal dos gases [$J.mol^{-1}.K^{-1}$]
r_i	Taxa de evaporação ou taxa de devolatilização
r	Raio da partícula [m]
t	Tempo [s]
T	Temperatura [K]
W	Umidade
Bi	Número de Biot
Py	Módulo de Thiele

Letras Gregas

ΔH_i	Entalpia [$J.Kg^{-1}$]
ϵ	Emissividade
σ	Constante de Boltzmann [$W.m^{-2}.k^{-4}$]
λ	Condutividade térmica [$(W.m^{-1}.K^{-1})$]
ρ	Massa específica [$Kg.m^{-3}$]

Subscritos

a	aproximação
e	evaporação
o	inicial
p	pirólise
s	solução

Lista de Figuras

1.1	Aspecto do xisto betuminoso - rocha escura e consistente	1
1.2	Carcterística de uma Retorta	2
2.1	Fluxograma - Pré-Processamento	10
2.2	Fluxograma - Processamento	12
2.3	Fluxograma - Processamento e Análise	13
3.1	Teste de Malha - 5 pontos, 10 pontos e 15 pontos	16
3.2	Teste de Malha - intervalo ampliado $400^{\circ}C \leq T \leq 500^{\circ}C$	17
3.3	Comparação dados Experimentais e Simulação Numérica - Gradiente de $10^{\circ}C.min^{-1}$	18
3.4	Comparação dados Experimentais e Simulação Numérica - Gradiente de $3^{\circ}C.min^{-1}$	19
3.5	Massa Normalizada para um gradiente de $3^{\circ}C.min^{-1}$	20
3.6	Massa Normalizada para um gradiente de $10^{\circ}C.min^{-1}$	20
3.7	Comparação da variação da Massa Normalizada entre gradientes de $3^{\circ}C.min^{-1}$ e $10^{\circ}C.min^{-1}$	21
3.8	Distribuição da Masssa Específica ao longo da Partícula $r = 1,63mm$	21
3.9	Distribuição da Temperatura ao longo da Partícula $r = 1,63mm$	22
3.10	Distribuição da Massa Específica ao longo da Partícula $r = 15mm$	22
3.11	Distribuição da Temperatura ao longo da Partícula $r = 15mm$	23
3.12	Distribuição da Massa Específica ao longo da Partícula - Temperatura Máxima $500^{\circ}C$ $r = 30mm$	23
3.13	Distribuição da Temperatura ao longo da Partícula - Temperatura Máxima $500^{\circ}C$ $r = 30mm$	24
3.14	Distribuição da Massa Normaliza ao longo da Partícula - Temperatura Máxima $500^{\circ}C$ $r = 30mm$	24

Lista de Tabelas

2.1	Funções para o Xisto Betuminoso.	6
3.1	Parâmtros e Constantes Físicas utilizados para a Simulação.	15

Capítulo 1

Introdução

A compreensão de processos de combustão é muitas vezes conveniente para o entendimento de muitos fenômenos presentes na engenharia. Entre estes fenômenos podemos citar o processo de retortagem de sólidos betuminosos com a obtenção de produtos de pirólise. O xisto betuminoso [1.1](#) é



Figura 1.1: *Aspecto do xisto betuminoso - rocha escura e consistente*

uma rocha sedimentar que contém querogênio, um complexo de matéria orgânica. Após o processo de extração, o minério bruto é submetido a tratamentos com o objetivo de reduzir e selecionar a sua granulometria, a "britagem". Em seguida, esses fragmentos são transportados por correias até certos tipos especiais de fornos - as retortas - sendo submetidos a um processo de aquecimento - a pirólise (decomposição da matéria orgânica). Neste processo o xisto betuminoso, com uma granulometria que varia entre 6 a 70 milímetros é aquecido por uma corrente gasosa a uma temperatura de aproximadamente 500 °C. Os produtos gasosos resultantes da pirólise devem ser resfriados, o que resulta na condensação dos vapores de óleo, que, sob a forma de gotículas, são transportados para fora da retorta [1.2](#). Os gases de xisto passam por outro processo de limpeza para a obtenção do óleo leve. O restante é encaminhado para a unidade de tratamento de gases, onde são produzidos gás combustível e gás liquefeito (GLP) e onde é feita a recuperação do enxofre. Por isso, têm-se realizado nos últimos anos muitos estudos que objetivam uma descrição detalhada de modelos matemáticos que procuram descrever o processo de pirólise como um todo: tais como o comportamento das estruturas de reação de secagem e pirólise [Martins *et al.* \(2010\)](#), [Saastamoinen e Richard \(1996\)](#) e [Benkoussas *et al.* \(2007\)](#), contração e expansão do volume da partícula [Blasi \(1996\)](#) e [Ouedraogo *et al.* \(1998\)](#), regime de devolatilização [Pyle e Zaror \(1984\)](#), [Saastamoinen e Richard \(1996\)](#) e [Peters e Bruch \(2003\)](#), influência que os aspectos geométricos da partícula exercem na combustão, [Peters e Bruch \(2003\)](#), entre outros. Cada um dos quais apresentam as suas vantagens, desvantagens e simplificações.

Dois fatores são de suma importância para a descrição de muitos destes modelos matemáticos: as taxas de transferência de calor durante o processo e os gradientes de temperatura nas partículas que são possíveis de ocorrer. Muitos trabalhos simplesmente desconsideram os gradientes.

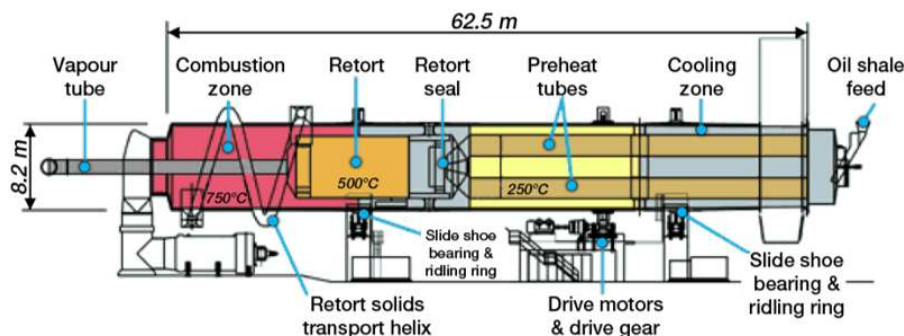


Figura 1.2: *Característica de uma Retorta*

Outra característica importante desses modelos é que eles envolvem sistemas de Equações Diferenciais Parciais Não-Lineares (EDPNL) definidos por várias equações simultâneas e suas respectivas condições de contorno, o que torna inviável, ou mesmo, praticamente impossível de serem resolvidos analiticamente. Deste modo a recorrência a métodos numéricos é fundamental e muito útil para a resolução destes problemas específicos, simplificando assim os esforços e custos para determinar-se a solução, com a vantagem desta estar tão próxima o quanto se queira da solução exata.

O Método Iterativo de Newton (MIN) para a linearização de sistemas EDPNL aliado à técnica de diferenciação numérica para a discretização das derivadas tanto de primeira ordem como as de ordens mais elevadas tem sido aplicados em problemas similares [Li \(2005\)](#). Para a solução do sistema resultante o Método do Gradiente Biconjugado (MGBC) é citado como versátil [GOLUB \(1966\)](#), obtendo através de poucas iterações uma solução numérica com uma estimativa muito próxima da solução exata. No entanto, dificuldades na implementação dos MIN e MGBC em rotinas computacionais que possam ser generalizadas e posteriormente incorporadas à softwares Open-Source, trazem grandes desafios como, por exemplo, a não garantia de minimização de resíduos

Todo o tratamento numérico e implementação de rotinas computacionais desenvolvidos aqui podem ser usadas para prever o comportamento de propriedades físicas como a temperatura, umidade e a massa específica, todas ao longo da geometria de uma partícula de combustível sólido quando submetida aos processos de secagem e pirólise.

Portanto, o objetivo deste trabalho é a descrição do comportamento detalhado da temperatura, da umidade e da massa específica de uma partícula de xisto betuminoso quando submetido ao processo de pirólise. Para isto alguns objetivos específicos são realizados:

- Desenvolvimento de um modelo matemático por equações físicas e termoquímicas que podem ser encontradas na literatura.
- Aplicação dos Métodos Numéricos de Newton e do Gradiente Bi-conjugado aliados com as técnicas de Diferenciação Simbólica e Discretização de Derivadas para a implementação de um algoritmo que resolva o sistema de EDPNL.
- Validação do modelo físico-matemático através de comparação com outros estudos para conjecturarmos o comportamento de uma partícula de xisto betuminoso submetida à pirólise.

O trabalho está dividido em 4 capítulos. No capítulo 2 será apresentado detalhadamente todas as considerações físicas adotadas, assim como todo o desenvolvimento matemático para a resolução do problema. Explicaremos ainda, sucintamente, o funcionamento do algoritmo aqui desenvolvido. No capítulo 3 é primeiramente apresentado os resultados, como a validação do modelo, e posteriormente no capítulo 4 apresentaremos as conclusões com as possíveis melhoras que podem ser feitas. As rotinas computacionais estão em anexo no final deste trabalho.

Capítulo 2

Modelo Matemático e Algoritmo de Integração

Muitos fatores influenciam no modo que a secagem e a devolatização ocorrerão num processo de pirólise. Um destes fatores é a taxa de transferência de calor para a partícula, juntamente como as taxas transferência internas de calor na partícula e a própria geração de calor pela reação química [Marco A. B. Zanonía \(2010\)](#). Dependendo da relação dessas taxas de transferência de calor podem ser gerados gradientes de temperatura ao longo da partícula, fato que muitos trabalhos desconsideram, o que não será feito aqui, o gradiente de temperatura também será avaliado.

Outro fator importante diz respeito à geometria. A utilização de modelos tridimensionais para modelar problemas de combustão não é uma prática viável, dado que eles exigem cálculos que excedem a capacidade dos computadores atuais, [Peters e Bruch \(2003\)](#). Pelo contrário, outros autores [Peters e Bruch \(2003\)](#) também defendem que o modelo unidimensional pode fornecer resultados que se comportam muito semelhante a dados experimentais.

2.1 Considerações Físicas

Assim este modelo considera, num regime transiente e unidimensional, uma partícula esférica de combustível sólido, inicialmente a uma temperatura T_0 , sendo exposta então a pirólise numa temperatura T_∞ . Neste modelo será considerada a validade das seguintes hipóteses [Marco A. B. Zanonía \(2010\)](#):

- Os princípios do volume médio local são utilizados para o desenvolvimento das equações de conservação;
- Equilíbrio térmico local;
- Os mecanismos de transporte na equação de conservação de massa são desconsideráveis, as mudanças no tempo são somente devido aos termos de reação;
- A biomassa é considerada um material isotrópico poroso, onde a porosidade (φ) depende da umidade;
- A radiação no interior da matriz porosa, os efeitos de dissipação viscosa e trabalho realizado por mudanças no campo de pressão são negligenciáveis;
- O calor efetivos específico e condutividade térmica da biomassa depende dos campos de temperatura e umidade;
- Os efeitos do transporte de massa por difusão da umidade e vapor d'água são negligenciados na equação de energia;

Termos	Xisto Betuminoso	Unidade
r_e	$(\rho_w - \rho_s)A_e \exp(-\frac{E_g}{R_g T})$	
r_p	$(\rho_s - \rho_{sc})A_p \exp(-\frac{E_p}{R_g T})$	
λ	0,21	$[(W.m^{-1}.K^{-1})]$
Q	$-\varphi \rho_e \Delta H_e + (1 - \varphi)r_p \Delta H_p$	$[W.m^{-2}.K^{-1}]$
C_p	587	$[J.Kg^{-1}.K^{-1}]$
ρ_s	2214	Kgm^{-3}
ρ_w	$\rho_s(1 + W)$	Kgm^{-3}
φ	$1 - (0,67 + W)S_g$	
$F(T)$	$h(T_\infty - T) + \varepsilon \sigma(T_\infty^4 - T^4)$	$[J.s^{-1}]$

Tabela 2.1: Funções para o Xisto Betuminoso.

- Supõe-se que a transferência de calor para a partícula de madeira é a partir do gás adjacente e do forno, sendo consequência de uma combinação de convecção e radiação;
- A temperatura da fase gasosa é assumida constante e igual à temperatura da parede do forno, T_∞ .

Tomando-se um volume de controle diferencial, e aplicando-se a equação de conservação de energia e massa, o autor [Marco A. B. Zanonía \(2010\)](#) obteve:

$$\rho_w C_{p_w}(T) \frac{\partial T}{\partial t} = \frac{1}{r_2} \frac{\partial [r^2 \lambda(T) \frac{\partial T}{\partial r}]}{\partial r} + Q \quad (2.1)$$

Derivando-se o termo entre colchetes, e assumindo $\frac{\partial \lambda}{\partial r}$ desprezível:

$$\rho_w C_{p_w}(T) \frac{\partial T}{\partial t} = \frac{1}{r^2} \lambda(T) (2r \frac{\partial T}{\partial r} + r^2 \frac{\partial^2 T}{\partial r^2}) + Q \quad (2.2)$$

Analogamente as equações cinéticas e químicas são fornecidas por [Marco A. B. Zanonía \(2010\)](#) através de (2.3) e (2.4):

$$\frac{\partial W}{\partial t} = -\frac{r_e}{\rho_{s0}} \quad (2.3)$$

$$\frac{\partial \rho_s}{\partial t} = -r_p \quad (2.4)$$

Sendo os termos r_e , r_p , λ , Q , C_p , ρ_w, ρ_s, φ apresentados na tabela 2.1 [Marco A. B. Zanonía \(2010\)](#):

Considerado anteriormente que a transferência de calor para a partícula de xisto betuminoso é a partir do gás adjacente e do forno, feita por uma combinação de convecção e radiação, têm-se que na superfície da partícula de xisto betuminoso ($r=R$) [Marco A. B. Zanonía \(2010\)](#):

$$r = R \longrightarrow k \frac{\partial T}{\partial r} = -F(T) \quad (2.5)$$

Onde $F(T)$ é o termo que quantifica a transferência de calor por convecção e radiação, dado na tabela 2.1. Temos também que no tempo inicial ($t_0=0$) [Marco A. B. Zanonía \(2010\)](#):

$$t = t_0 \left\{ \begin{array}{l} T = T_0 \\ W = W_0 \\ \rho_s = \rho_{h_2o} S_g = \rho_{s0} \end{array} \right. \quad (2.6)$$

Assim temos um sistema definido por três equações diferenciais parciais (2.2), (2.3) e (2.4), com suas respectivas condições de contorno. Deste modo, aplicar-se-á para a resolução do sistema

descrito a discretização através de técnicas de diferenças finitas [Li \(2005\)](#), que apresentam alguns importantes benefícios para determinar derivadas de primeira ordem como também ordens mais elevadas, baseando-se na aproximação de derivadas por diferenças finitas obtidas da série de Taylor para a função derivada. Obtêm-se então um sistema definido por três equações simultâneas.

$$\rho_w C p_w(T) \frac{T_i - T_{i-1,0}}{\partial t} = \frac{1}{r^2} \lambda(T) \left(2r \frac{T_{i+1} - T_{i-1}}{\partial r} + r^2 \frac{T_{i-1} - 2T_i + T_{i+1}}{\partial r^2} \right) + Q \quad (2.7)$$

$$\frac{W_i - W_{i-1,0}}{\partial t} = -\frac{r_e}{\rho_{s0}} \quad (2.8)$$

$$\frac{\rho_{s_i} - \rho_{i-1,0}}{\partial t} = -r_p \quad (2.9)$$

Reorganizando-se as equações (2.7), (2.8) e (2.9), temos:

$$\rho_w C p_w(T) \frac{T_i - T_{i-1,0}}{\partial t} - \frac{1}{r^2} \lambda(T) \left(2r \frac{T_{i+1} - T_{i-1}}{\partial r} + r^2 \frac{T_{i-1} - 2T_i + T_{i+1}}{\partial r^2} \right) - Q = 0 \quad (2.10)$$

$$\frac{W_i - W_{i-1,0}}{\partial t} + \frac{r_e}{\rho_{s0}} = 0 \quad (2.11)$$

$$\frac{\rho_{s_i} - \rho_{i-1,0}}{\partial t} + r_p = 0 \quad (2.12)$$

Os termos Q , r_e , e r_p são funções exponenciais, então as equações (2.10), (2.11) e (2.12) são não lineares, e o sistema pode ser então reescrito de tal forma:

$$\begin{cases} f_1(T, W, \rho) = 0 \\ f_2(T, W, \rho) = 0 \\ f_3(T, W, \rho) = 0 \end{cases} \quad (2.13)$$

Será utilizado o método de Newton para Equações não Lineares na resolução deste problema. Assim, procedemos da forma como se segue. Tendo o sistema definido pelas equações:

$$\begin{cases} f_1(x_1, x_2, \dots, x_n) = 0 \\ f_2(x_1, x_2, \dots, x_n) = 0 \\ f_3(x_1, x_2, \dots, x_n) = 0 \end{cases} \quad (2.14)$$

Seja $(x^{a1}, x^{a2}, \dots, x^{am})$ uma aproximação para a solução $(x^{s1}, x^{s2}, \dots, x^{sm})$ do sistema acima. Admitindo que $f_1, f_2, \dots, f_m$ sejam suficientemente diferenciáveis, expandimos $f_1(x_1, x_2, \dots, x_m), f_2(x_1, x_2, \dots, x_m), \dots, f_m(x_1, x_2, \dots, x_m)$, usando série de Taylor para funções de n variáveis em torno de $(x_1, x_2, \dots, x_m)$ assim:

$$\begin{cases} f_1(x_1, x_2, \dots, x_m) = f_1(x^{a1}, x^{a2}, \dots, x^{am}) + f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am})(x^{s1} - x^{a1}) + \dots \\ f_2(x_1, x_2, \dots, x_m) = f_2(x^{a1}, x^{a2}, \dots, x^{am}) + f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am})(x^{s1} - x^{a1}) + \dots \\ \vdots \\ f_m(x_1, x_2, \dots, x_m) = f_m(x^{a1}, x^{a2}, \dots, x^{am}) + f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am})(x^{s1} - x^{a1}) + \dots \end{cases} \quad (2.15)$$

Admitindo que $(x^{a1}, x^{a2}, \dots, x^{am})$ esteja suficientemente próximo da solução $(x^{s1}, x^{s2}, \dots, x^{sm})$ a ponto de poderem ser abandonados os termos de mais alta ordem, podemos determinar uma nova aproximação para a raiz $(x^{s1}, x^{s2}, \dots, x^{sm})$ fazendo $f_1(x_1, x_2, \dots, x_m) = f_2(x_1, x_2, \dots, x_m) = \dots =$

$f_m(x_1, x_2, \dots, x_m) = 0$. Obtemos, então, o sistema:

$$\begin{cases} -f_1(x^{a1}, x^{a2}, \dots, x^{am}) = f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am})(x_1 - x^{a1}) + \dots + f_{1x_m}(x^{a1}, x^{a2}, \dots, x^{am})(x_m - x^{am}) \\ -f_2(x^{a1}, x^{a2}, \dots, x^{am}) = f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am})(x_1 - x^{a1}) + \dots + f_{2x_m}(x^{a1}, x^{a2}, \dots, x^{am})(x_m - x^{am}) \\ \vdots \\ -f_m(x^{a1}, x^{a2}, \dots, x^{am}) = f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am})(x_1 - x^{a1}) + \dots + f_{mx_m}(x^{a1}, x^{a2}, \dots, x^{am})(x_m - x^{am}) \end{cases} \quad (2.16)$$

Explicitando o sistema acima para que a matriz Jacobiano esteja mais destacável:

$$\begin{bmatrix} f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{1x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{1x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{2x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{2x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{mx_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{mx_m}(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix} \begin{bmatrix} x_1 - x^{a1} \\ x_2 - x^{a2} \\ \vdots \\ x_m - x^{am} \end{bmatrix} = \begin{bmatrix} -f_1(x^{a1}, x^{a2}, \dots, x^{am}) \\ -f_2(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots \\ -f_m(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix} \quad (2.17)$$

Equivalente a:

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_m} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \dots & \frac{\partial f_m}{\partial x_m} \end{bmatrix} \begin{bmatrix} x_1 - x^{a1} \\ x_2 - x^{a2} \\ \vdots \\ x_m - x^{am} \end{bmatrix} = \begin{bmatrix} -f_1(x^{a1}, x^{a2}, \dots, x^{am}) \\ -f_2(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots \\ -f_m(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix} \quad (2.18)$$

Um possível modo de resolver o sistema acima seria a utilização da regra de Cramer [FRANCO \(2007\)](#) obtendo:

$$x^{s1} - x^{a1} = \frac{\begin{bmatrix} -f_1(x^{a1}, x^{a2}, \dots, x^{am}) & f_{1x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{1x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ -f_2(x^{a1}, x^{a2}, \dots, x^{am}) & f_{2x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{2x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots & \vdots & \ddots & \vdots \\ -f_m(x^{a1}, x^{a2}, \dots, x^{am}) & f_{mx_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{mx_m}(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix}}{\begin{bmatrix} f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{1x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{1x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{2x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{2x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{mx_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{mx_m}(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix}} \quad (2.19)$$

$$x^{s2} - x^{a2} = \frac{\begin{bmatrix} f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & -f_1(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{1x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & -f_2(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{2x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am}) & -f_m(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{mx_m}(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix}}{\begin{bmatrix} f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{1x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{1x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{2x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{2x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{mx_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{mx_m}(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix}} \quad (2.20)$$

$$x^{sm} - x^{am} = \frac{\begin{bmatrix} f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{1x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & -f_1(x^{a1}, x^{a2}, \dots, x^{am}) \\ f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{2x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & -f_2(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{mx_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & -f_m(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix}}{\begin{bmatrix} f_{1x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{1x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{1x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ f_{2x_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{2x_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{2x_m}(x^{a1}, x^{a2}, \dots, x^{am}) \\ \vdots & \vdots & \ddots & \vdots \\ f_{mx_1}(x^{a1}, x^{a2}, \dots, x^{am}) & f_{mx_2}(x^{a1}, x^{a2}, \dots, x^{am}) & \dots & f_{mx_m}(x^{a1}, x^{a2}, \dots, x^{am}) \end{bmatrix}} \quad (2.21)$$

As matrizes do numerador e denominador são mal condicionadas, pois apresentam muitos elementos nulos - este fato pode ser visto através do seguinte modo: ao discretizar as equações nos "p" pontos de "controle", será obtido um vetor de "3.p linhas" (ver equações (2.10), (2.11) e (2.12) e veja também que 3.p = m). A seguir será necessária então a obtenção da matriz Jacobiano. Observe agora as equações (2.10), (2.11) e (2.12), elas possuem para o mesmo ponto "i" (i=1,2,...,p) somente 3 incógnitas diferentes na mesma variável, e inclusive quando "i" variar até "i+2", por exemplo, essa mesma incógnita pode aparecer somente até mais duas vezes no vetor de "3.p linhas", então quando for aplicado a operação de obtenção do jacobiano, ou seja, a derivação em uma variável qualquer, a linha resultante da matriz jacobiano terá somente até cinco elementos não nulos, e os demais "3.p - 5" elementos desta linha serão nulos, portanto a matriz jacobiano terá muitos elementos nulos.

Considerando ainda que o custo computacional para a obtenção de determinantes de matrizes de ordem alta é elevado, será utilizado então outro método para a solução do sistema anterior, o método iterativo do Gradiente Biconjugado.

O grande valor prático do ponto de vista para o tempo de cálculo vem do fato que este método partindo de um pré-condicionador permite obter através de poucas iterações uma estimativa muito próxima da solução exata, na prática por um número de iterações muito menor que o número de incógnitas. Outra vantagem é sua aplicação independentemente da simetria da matriz dos coeficientes e da necessidade de ser positiva definida, como ocorre no método de gradiente conjugado, sendo assim o método de Gradiente Bconjugado uma generalização do método de Gradiente Conjugado **GOLUB (1966)**. Sendo A^* a matriz adjunta de A e Considerando o sistema linear:

$$[A] [x] = [b] \quad (2.22)$$

Definimos um vetor x_0 referente a um chute inicial e um pré-condicionador M , freqüentemente $M^{-1} = 1$, procedemos da seguinte forma **GOLUB (1966)**:

1-Escolhemos um chute inicial x_0 e um pré-condicionador M

$$2-r_0 \leftarrow b - Ax_0$$

$$3-r_0^* \leftarrow b - x_0^* A$$

$$4-p_0 \leftarrow M^{-1} r_0$$

$$5-p_0^* \leftarrow r_0^* M^{-1}$$

para $K=0,1,2,\dots,n$

$$6-\alpha_k \leftarrow \frac{r_k^* M^{-1} r_k}{p_k^* A p_k}$$

$$7-x_{k+1} \leftarrow x_k + \alpha_k p_k$$

$$8-x_{k+1}^* \leftarrow x_k^* + \overline{\alpha_k} p_k^*$$

$$9-r_{k+1} \leftarrow r_k + \alpha_k A p_k$$

$$10-r_{k+1}^* \leftarrow r_k^* + \overline{\alpha_k} p_k^* A$$

$$11-\beta_k \leftarrow \frac{r_{k+1}^* M^{-1} r_{k+1}}{r_k^* A r_k}$$

$$12-p_{k+1} \leftarrow M^{-1} p_{k+1} + \beta_k r_k$$

$$13-p_{k+1}^* \leftarrow p_{k+1}^* M^{-1} + \overline{\beta_k} r_k^*$$

Assim é possível resolver o sistema inicial que estava definido por três equações diferenciais parciais não lineares, com suas condições de contorno, e então podemos prever o comportamento das três propriedades físicas que são as variáveis do nosso modelo matemático: a temperatura, a umidade e a variação da massa específica. Conforme dito inicialmente, outros autores desconsideram o efeito do gradiente de temperatura, fato que não será feito neste trabalho. As três variáveis serão avaliadas com a variação do tempo e ao longo da geometria esférica da partícula.

Muito dos trabalhos usados como referência para o desenvolvimento deste modelo matemático apresentam seus resultados também na forma de variação da massa normalizada da partícula submetida à pirólise. Portanto, com o objetivo de comparação com outros trabalhos já existentes, será feito uma integração numérica para a obtenção da massa normalizada ao longo do tempo, obtida simplesmente pela integração da massa específica ao longo do volume:

$$m_n = \frac{\int \int \int \rho \partial V}{\int \int \int \rho_0 \partial V} \quad (2.23)$$

2.2 Algoritmo Numérico

A seguir através de fluxogramas será explicado como o algoritmo incorpora cada um dos passos descritos anteriormente.

O pré-processamento deve ser entendido como uma fase inicial da modelagem das leis físico-matemáticas que descrevem o problema proposto.

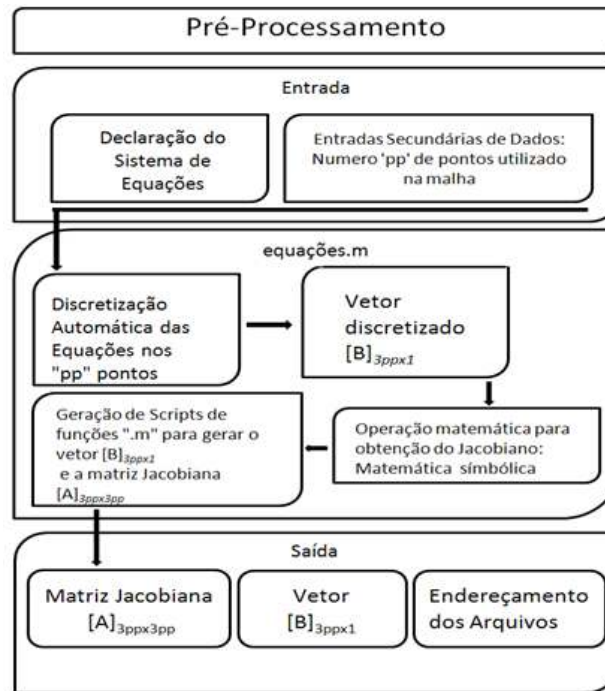


Figura 2.1: Fluxograma - Pré-Processamento

A declaração do sistema consiste em definir as equações de nosso modelo, ou seja, as equações (2.2), (2.3) e (2.4) devem ser digitadas com o formato padrão de (2.10), (2.11) e (2.12), atenta-se para que as equações sejam digitadas nos pontos referentes $r=0$, $r = R$, assim como nos pontos interiores a este intervalo. É de responsabilidade do usuário a correta discretização das equações, incluindo também a aplicação das condições de contorno adequadas. A facilidade que o arquivo

"equacoes.m" fornece, é o fato que devem ser digitadas somente os três formatos padrão de equações, referente aos pontos onde $r=0$, $r=R$ e o terceiro formato para os pontos internos aos dois anteriores. Onde "r" refere-se ao raio da partícula. Assim economiza-se tempo, porque não é necessário digitar todas as equações referentes a cada um dos pontos da "malha" o que diminui a possibilidade de erros de digitação, o algoritmo produz então o vetor discretizado $[B]_{3pp \times 1}$ (ver entre as linhas 102 a 127 do algoritmo "equacoes.m", em anexo).

Ao mesmo tempo devemos também entrar com o número de pontos "pp", que define o número de pontos que são utilizados na malha numérica. Assim, ao escolher, por exemplo, "pp"=50, teremos um total de cento e cinquenta incógnitas e cento e cinquenta equações - cinquenta incógnitas para a variável T_i , cinquenta incógnitas para a variável W_i e cinquenta incógnitas para a variável ρ_i , assim como cinquenta equações para (2.10), cinquenta equações para (2.11) e cinquenta equações para (2.12).

Realizado os dois primeiros passos citados acima, o arquivo "equacoes.m" gera para a entrada fornecida três vetores coluna, com os três tipos de incógnitas (T_i , W_i , ρ_i , $i = 1, 2, \dots, pp$). Notou-se durante a construção do algoritmo que vetores linha são mais fáceis de serem manipulados, então utilizamos a transformação de vetores coluna para vetores linha, por um processo que é chamado de concatenação vertical. Também, é gerado no arquivo "equacoes.m" os vetores coluna para o "chute" inicial, que do mesmo modo é submetido ao processo de concatenação - note que a definição do método de Newton necessita de uma aproximação inicial, que é o que chamamos de "chute" inicial (ver entre as linhas 33 a 83 do algoritmo "equacoes.m", em anexo).

Com os três vetores de incógnitas, mencionados acima, produz-se a matriz Jacobiano através da derivada simbólica aplicada ao vetor $[B]_{3pp \times 1}$, ver linha 132 do algoritmo "equacoes.m", em anexo. Em teoria, por estar se aplicando a derivada analítica a matriz Jacobiano está livre de erros.

Feito as operações descritas acima, o vetor $[B]_{3pp \times 1}$ e a matriz Jacobiano são escritos em um novo arquivo "m", que serão utilizados posteriormente na etapa de processamento, ver linha 136 a 169 do algoritmo "equacoes.m", em anexo.

Como será dito adiante, as soluções do sistema serão armazenadas em cada instante de tempo, para isso é necessário que os "chutes" iniciais sejam sempre alterados e também que esta solução não se perca com a "marcha no tempo". Um jeito de realizar este procedimento seria a utilização de uma função, o "armazenador.m" garantido que este procedimento de escolha de um "chute" melhor ao longo do tempo seja executado. Assim termina-se a fase de pré-processamento - para comodidade do usuário, a localização dos arquivos gerados é um dado de saída deste algoritmo, sendo a chamada posterior destas funções automática.

A seguir será descrito a etapa de processamento. Nesta etapa o objetivo é a resolução do sistema formado pelas equações (2.10), (2.11) e (2.12). A partir do endereçamento realizado na etapa de pré-processamento obtêm-se duas entradas, o vetor $[B]_{3pp \times 1}$ e a matriz Jacobiano, utilizadas para o processamento, sendo ainda necessária a declaração dos valores numéricos dos parâmetros físicos deste modelo, mencionados na Lista de Simbolos. Isso deve ser feito acessando-se o arquivo "jac.m" e alterando-se os valores das constantes presentes entre a linha 73 e 93. As condições iniciais são alteradas entre as linhas 38 e 43 do algoritmo "jac.m", em anexo - Recomenda-se que o chute inicial seja as próprias condições iniciais.

Entre as linhas 48 e 71 do algoritmo "jac.m", em anexo, são feitas a concatenação dos vetores referente a (T_i , W_i , ρ_i). Essa concatenação não deve ser confundida com a concatenação citada na etapa de pré-processamento, desta vez a concatenação irá armazenar todos os instantâneos da solução, enquanto o "armazenador.m" da etapa anterior armazena a solução somente enquanto estiver no loop da "marcha no tempo".

O algoritmo então realiza os cálculos através do método de Gradiente Biconjugado para estimar a solução do sistema linear. Para cada instante de tempo é armazenado sua respectiva solução, até que seja atingido o último instante de tempo. Deve-se deixar claro que o "armazenador.m" tem a função de alterar os novos "chutes iniciais" de cada iteração, de modo que estes agora são, por exemplo, para o segundo instante de tempo, a solução do primeiro instante de tempo - ver entre as linhas 101 e 118 do algoritmo "jac.m", em anexo. Pode-se também perceber a importância

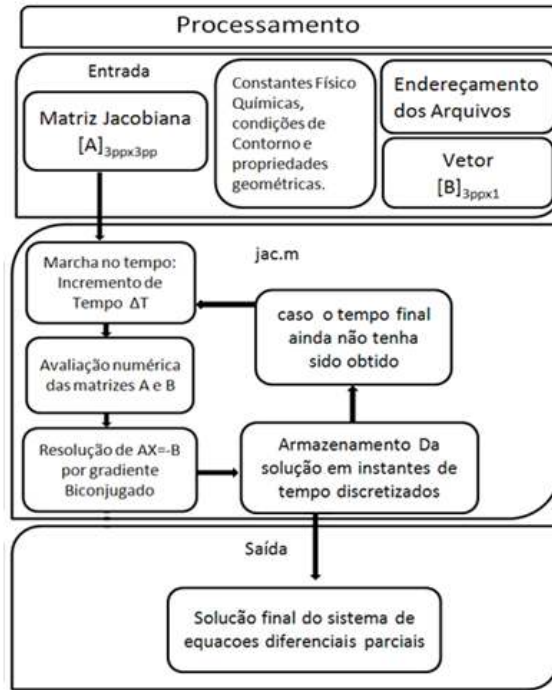


Figura 2.2: Fluxograma - Processamento

das matrizes Jacobiano e $[B]_{3pp \times 1}$, já que a cada iteração elas são novamente avaliadas através da chamada das funções geradas no pré-processamento, devido a alteração da nova aproximação inicial pelo "armazenador.m".

Finalmente descreveremos a última etapa do algoritmo o Pós-processamento e a Análise. Tendo sido a etapa de processamento concluída, têm-se as soluções para cada instante de tempo, ao longo do raio da partícula, armazenadas por uma matriz em que as linhas indicam as "3xpp" variáveis e as colunas a variação destas com o tempo. Estas variáveis armazenadas serão usadas aqui para a determinação da variação da massa normalizada da partícula submetida à pirólise com o variar do tempo. Para isso desenvolvemos a equação (2.23):

$$m_n = \frac{\int \rho(1+W)4\pi r^2 dV}{\int \rho_0(1+W_o)4\pi r^2 dV} \quad (2.24)$$

Esta integração é computada pelo método de integração numérica do trapézio. Gerando então a massa normalizada.

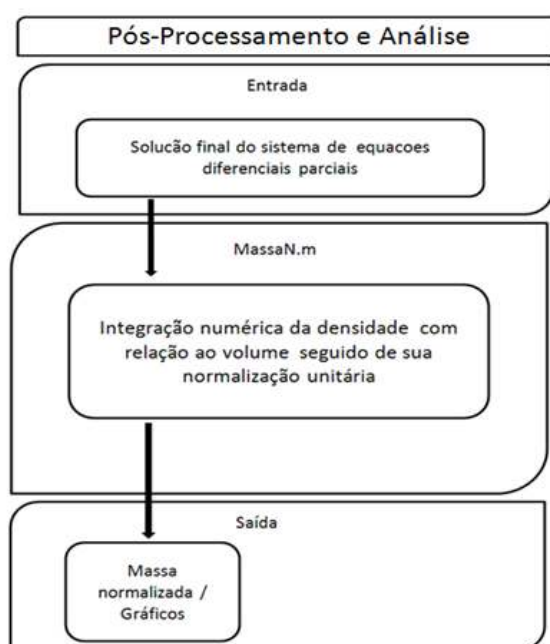


Figura 2.3: *Fluxograma - Processamento e Análise*

Termos	Valor	Unidade
A_p	$16.42e^{+02}$	s^{-1}
C_p	587	$J.Kg^{-1}K^{-1}$
E_e	$19.95e^{+03}$	$J.mol^{-1}$
E_p	$79.94e^{+03}$	$J.mol^{-1}$
h	5	$J.m^{-2}.s.K$
r	0.001627	m
R_g	8.31	$J.mol^{-1}.K^{-1}$
S_g	2214	Kgm^{-3}
T_0	300K	K
dt	0.05	s
W_0	0.01	*
ΔH_e	$2700e^{+03}$	$J.kg^{-1}$
ΔH_p	$4500e^{+03}$	$J.kg^{-1}$
ε	0.9	*
λ	0.21	$W.m^{-1}.K^{-1}$
ρ_c	1918	Kgm^{-3}
ρ_{s0}	2214	Kgm^{-3}
ρ_{H_2O}	1000	Kgm^{-3}
σ	$5.67e^{-08}$	$W.m^{-2}.k^{-4}$

Tabela 3.1: *Parâmtros e Constantes Físicas utilizados para a Simulação.*

Capítulo 3

Resultados e Discussões

3.1 Estudo de Malha

Para comprovar a validade do modelo deve-se antes de tudo confirmar a precisão da malha de nosso algoritmo numérico. Para isso presume-se que não haja uma variação significativa ao alterar-se o número de pontos da malha. Ou seja, independentemente do número de pontos que se escolha para a malha, o resultado final, teoricamente, não deveria variar significativamente para o número de pontos adotados.

A condição necessária para fazermos considerações com relação a resultados de diferentes números de pontos da malha é obviamente que todos os parâmetros físicos, como a condutividade térmica do xisto betuminoso - por exemplo - sejam os mesmos para cada caso.

Essa comparação de resultados finais foi feita selecionando a massa normalizada prevista para o caso em que o número de pontos " pp " = 5, para o caso em que " pp " = 10, e finalmente para o caso em que " pp " = 15. A seguir, com os parâmetros listados na tabela 3.1 temos a base de nosso estudo inicial.

Simulando para os valores acima foi obtido os seguintes dados:

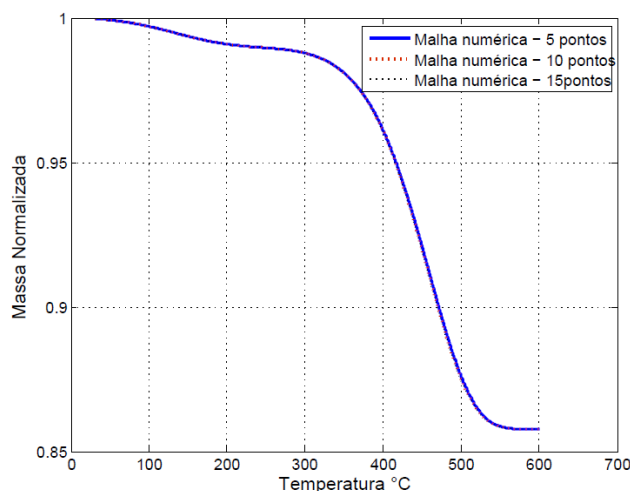


Figura 3.1: *Teste de Malha - 5 pontos, 10 pontos e 15 pontos*

Pelo gráfico 3.1 confirma-se que as curvas estão realmente muito próximas umas das outras. Ou seja, não há uma diferença significativa entre os resultados obtidos, para as mesmas condições, com a variação do número de pontos da malha.

Pelo gráfico 3.2, ampliação do primeiro 3.1, referente ao intervalo pertencente à temperatura $400^{\circ}\text{C} \leq T \leq 500^{\circ}\text{C}$ torna mais notável ainda a aproximação dos resultados relativos ao número de pontos da malha de dez e quinze pontos. Outra consideração importante que pode ser feita daí é o erro relativo entre as malhas de dez e quinze pontos com a malha de cinco pontos - da ordem de milésimos. Portanto a partir daqui, todos os resultados serão obtidos a partir de uma malha de cinco pontos, uma vez que malhas com números de ponto muito alto exigem um custo computacional maior e demandam maior tempo de processamento.

3.2 Validação Numérica

Com o objetivo de validar o modelo físico-matemático deste trabalho simulou-se a pirólise em duas ocasiões com a configuração dos parâmetros físicos que correspondessem com as mesmas condições iniciais de dois experimentos reais diferentes. Os autores [Marco A. B. Zanonia i \(2010\)](#) dispunham de um dispositivo, Analisador Termogravimétrico (TGA), que podia determinar a variação da massa da partícula pirolisada em função do aumento gradual de temperatura.

Naquele experimento procedeu-se com o auxílio de um TG - DSC 111 (Setaram) que dispunha de uma balança com sensibilidade de detecção da ordem de microgramas. As amostras em um dos experimentos possuíam cerca de 40mg , e foram submetidas a uma taxa de aquecimento de $10^{\circ}\text{C}.\text{min}^{-1}$ até atingirem 800°C . No outro a mesma quantia foi submetida a uma taxa de aquecimento de $3^{\circ}\text{C}.\text{min}^{-1}$ até atingir 800°C .

Os gráficos 3.3 e 3.4 mostram a comparação entre a simulação do algoritmo e os resultados obtidos pelo TGA.

Examinando os gráficos 3.3 e 3.4 nota-se que realmente existe uma aproximação muito boa do modelo aqui desenvolvido com os dados experimentais. Para o caso de um gradiente de $10^{\circ}\text{C}.\text{min}^{-1}$ é evidente esta aproximação, os pontos experimentais são coincidentes com os pontos obtidos por simulação, já no caso de um gradiente de $3^{\circ}\text{C}.\text{min}^{-1}$ os pontos apresentam um ligeiro afastamento, porém possuem ainda uma aproximação boa com os dados experimentais.

Quando a pirólise está chegando ao fim, tem-se início o processo de oxidação, que este modelo não considera, motivo este da diferença entre os dados experimentais e a simulação nos últimos

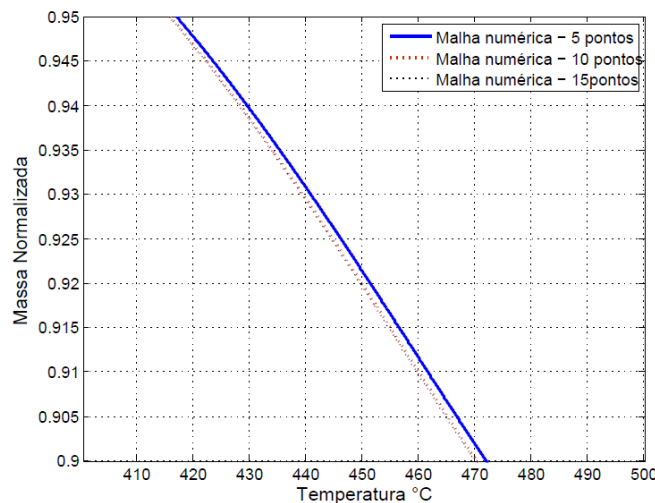


Figura 3.2: *Teste de Malha - intervalo ampliado $400^{\circ}C \leq T \leq 500^{\circ}C$*

instantes da pirólise tanto para o gradiente de $3^{\circ}C.min^{-1}$ quanto para o gradiente de $10^{\circ}C.min^{-1}$.

3.3 Estudo Paramétrico

Tendo a garantia de aproximação de resultados do modelo físico-matemático com dados experimentais analisaremos a influência que o tamanho da partícula e o gradiente de temperatura que ela está submetida exercem sobre a reação de pirólise para o xisto betuminoso.

Serão feitas comparações para dois casos: o efeito da variação do raio da partícula para um gradiente de $3^{\circ}C.min^{-1}$ e o efeito da variação do raio da partícula para um gradiente de $10^{\circ}C.min^{-1}$. Nas duas análises são consideradas as mesmas condições iniciais, as condições dadas na tabela 3.1, exceto, é claro, a dimensão do raio.

As duas figuras 3.5 e 3.6 apresentam a variação da massa normalizada em função da temperatura, para valores de raio 1, 63mm, 5mm, 10mm e 15mm. Em ambos os casos pode-se observar que a reação de pirólise conclui-se primeiro para valores de raio menores, acontecimento este que é esperável, já que raios maiores implicam uma menor taxa de aquecimento na partícula e exigem mais tempo para evaporar e devolatilizar uma massa maior. Podemos assim mais uma vez apresentar confiança em relação a validade do modelo.

Na figura 3.7 está a comparação entre a variação da massa normalizada em função da temperatura para os casos em que os gradientes de temperatura valem $3^{\circ}C.min^{-1}$ e $10^{\circ}C.min^{-1}$.

Para o caso 3.7, onde é mantido o raio fixo, a comparação dos gradientes de temperatura do TGA aponta que quanto menor for este gradiente, menor será a temperatura em que o processo de pirólise se concluirá. Deve ser observado que este gráfico não mostra o tempo da reação de pirólise diretamente, mas este tempo pode ser estimado considerando a temperatura na qual a massa normalizada estabiliza-se e dividindo-a pelo gradiente de temperatura do TGA.

3.4 Influência das Propriedades Física

Até então nosso estudo estava focado no efeito que a variação do raio da partícula, assim como o gradiente de temperatura provocava na massa normalizada daquela, agora será visto como a temperatura e a massa específica se comportam ao longo da partícula para um determinado instante no tempo.

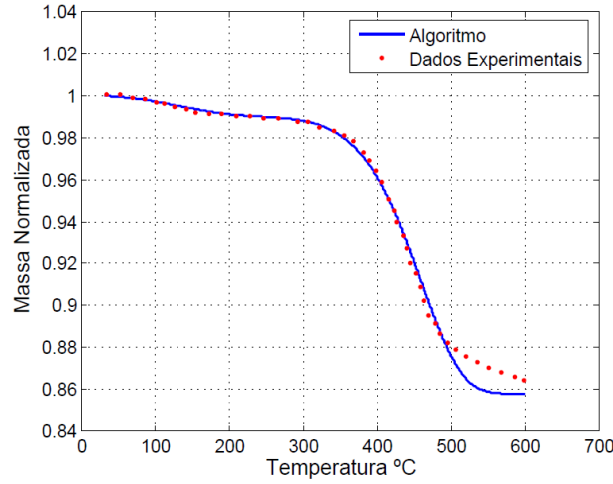


Figura 3.3: Comparação dados Experimentais e Simulação Numérica - Gradiente de $10^{\circ}\text{C}.\text{min}^{-1}$

A reação de pirólise é controlada por dois parâmetros adimensionais importantes, o módulo de Thiele (Py) e o número de Biot (Bi):

$$Py = \frac{\lambda}{A_p \cdot e^{\frac{E_p}{R_g \cdot T}} \rho c_p r^2} \quad (3.1)$$

$$Bi \propto \frac{h}{\lambda} r \quad (3.2)$$

Caso $Py \gg 1$, a reação é cineticamente limitada, podendo a massa normalizada ser integrada diretamente na temperatura, equivalente a desconsiderar este gradiente de temperatura [Peters e Bruch \(2003\)](#). Para partículas muito pequenas - ou seja, $Bi \ll 1$ - a transferência de calor dependerá quase que exclusivamente da condutividade térmica do material, o que gera um gradiente de temperatura ao longo da partícula quase que uniforme. Estas duas observações podem ser comprovadas para o caso em que o raio da partícula de xisto betuminoso valia $1,63\text{mm}$, conforme vemos nos gráficos [3.8](#) e [3.9](#).

Se $Py \ll 1$, o processo é limitado pela transferência de calor, ou seja devem ser levados em consideração os efeitos do gradiente de temperatura [Peters e Bruch \(2003\)](#). Com relação ao tamanho da partícula, grandes raios - ou seja, $Bi \gg 1$ - a transferência interna de calor ao longo da partícula é muito menor que a transferência externa de calor para a partícula, gerando assim um gradiente de temperaturas. Isto também pôde ser observado para o caso em raio da partícula de xisto betuminoso valia 15mm . Conforme vemos nos gráficos [3.10](#) e [3.11](#):

3.5 Outros Casos

Situações de pirólise do xisto betuminoso ocorrem freqüentemente na indústria. Há anos têm-se aproveitado o potencial deste recurso mineral para a produção de energia, entre outras atividades, e o Brasil é um dos países que lidera a produção de óleo a partir do xisto.

Finalmente, encerraremos os nossos resultados fazendo conjecturas sobre o comportamento de uma partícula de xisto betuminoso com um raio de 30mm submetida à pirólise limitada a 500°C , situação típica empregada nestas indústrias.

Pela figura [3.14](#) além da variação da massa específica do xisto betuminoso podemos constatar que o tempo da reação é relativamente alto em relação aos outros casos estudados até aqui, cerca de 6000 segundos, isto porque além da limitação da temperatura até 500°C há a influência do

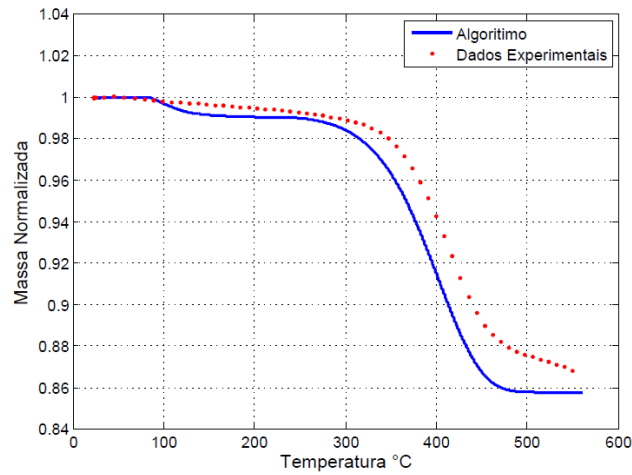


Figura 3.4: Comparação dados Experimentais e Simulação Numérica - Gradiente de $3^{\circ}\text{C}.\text{min}^{-1}$

tamanho do raio, conforme vimos anteriormente, partículas de raios maiores gastam mais tempo para completar a reação de pirólise.

Já na figura 3.13 podemos observar o comportamento da temperatura ao longo do raio da partícula durante a reação. Nos primeiros instantes é evidente que existe um gradiente de temperatura, isto porque, como já vimos, o raio da partícula é grande, porém no decorrer da reação há uma tendência de diminuição deste gradiente de temperatura, a razão disso: a temperatura está limitada a 500°C , esta temperatura será atingida muito antes do término da reação, havendo então tempo o bastante para que ocorra o equilíbrio térmico ao longo da partícula de xisto betuminoso.

Finalmente com a figura 3.12 podemos notar o gradiente de massa específica da partícula, fato novamente esperado uma vez que o raio desta partícula é significativo. Para os instante de tempo inicial, esse gradiente é pouco perceptível, uma vez que a perda de massa por evaporação de água é muito pequena, a água contida na partícula representa menos de 1% da massa total do xisto betuminoso, porém, com o decorrer da reação, a perda de massa se torna significativa e o gradiente se torna claramente visível, e ao seu término o gradiente novamente tende a se anular, uma vez que a massa específica irá se estabilizar.

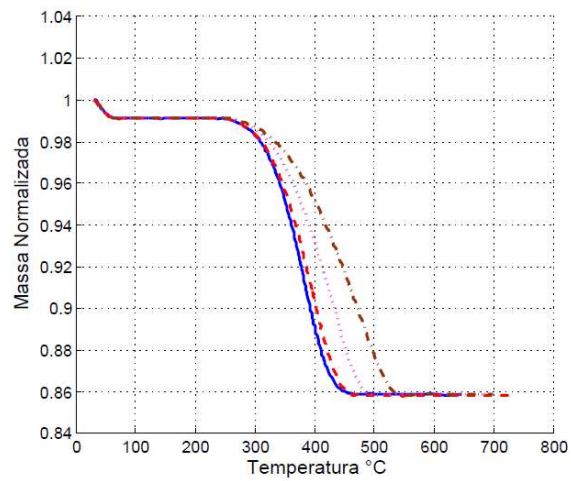


Figura 3.5: *Massa Normalizada para um gradiente de $3^{\circ}\text{C}.\text{min}^{-1}$*

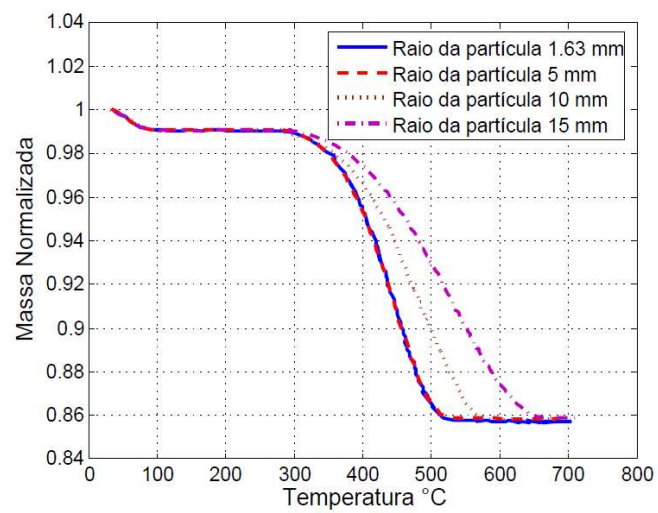


Figura 3.6: *Massa Normalizada para um gradiente de $10^{\circ}\text{C}.\text{min}^{-1}$*

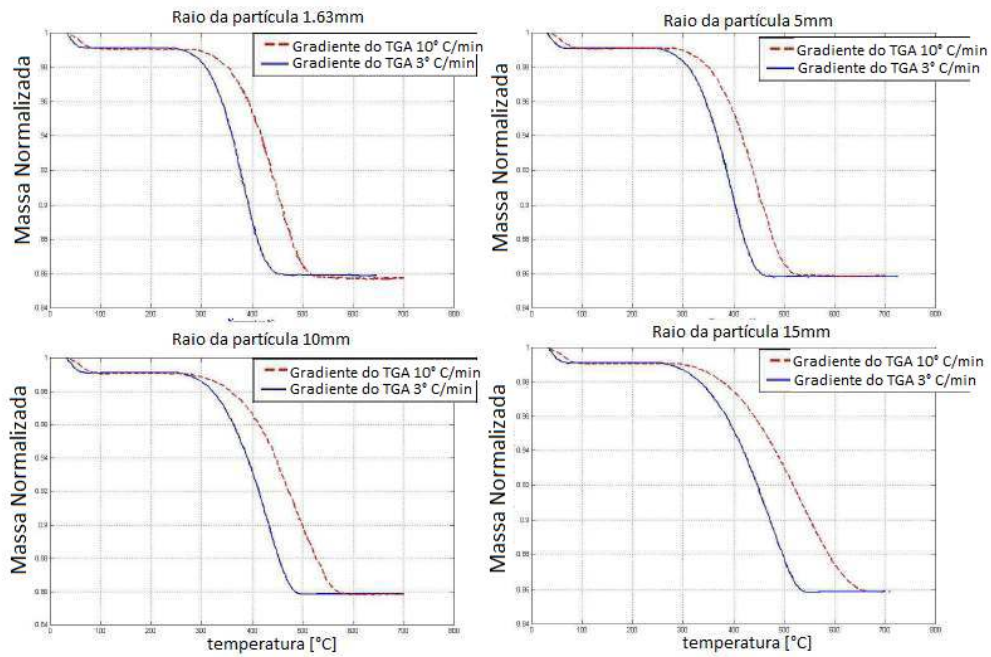


Figura 3.7: Comparação da variação da Massa Normalizada entre gradientes de $3^{\circ}\text{C}.\text{min}^{-1}$ e $10^{\circ}\text{C}.\text{min}^{-1}$

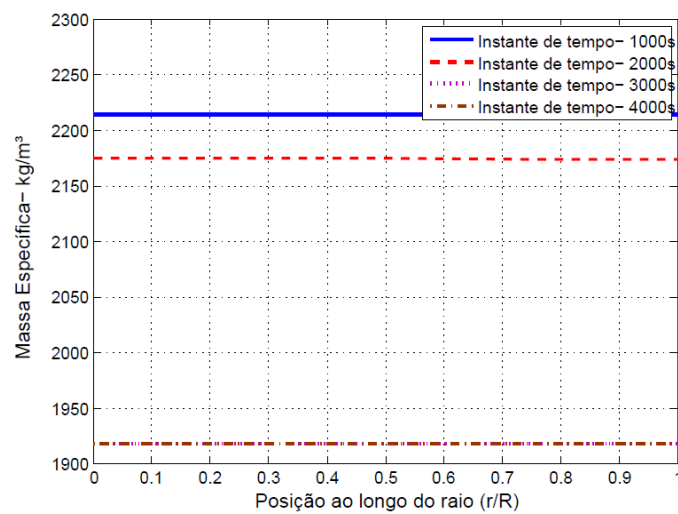


Figura 3.8: Distribuição da Massa Específica ao longo da Partícula $r = 1,63\text{mm}$

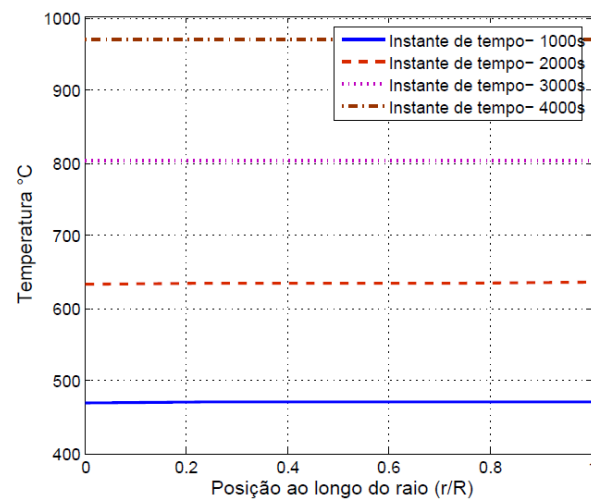


Figura 3.9: Distribuição da Temperatura ao longo da Partícula $r = 1,63\text{mm}$

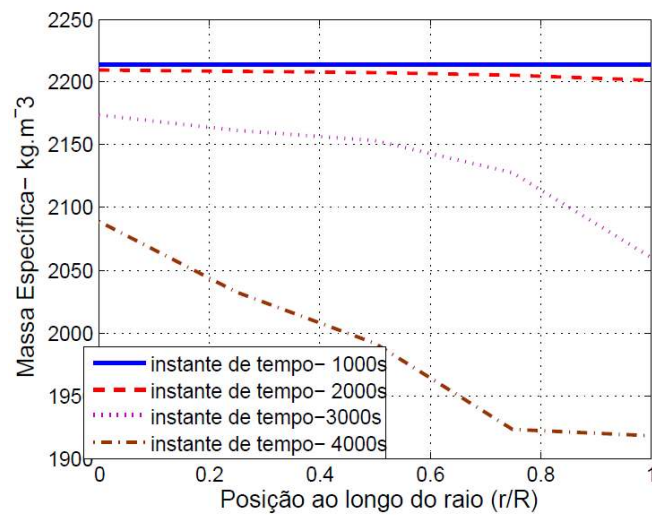


Figura 3.10: Distribuição da Massa Específica ao longo da Partícula $r = 15\text{mm}$

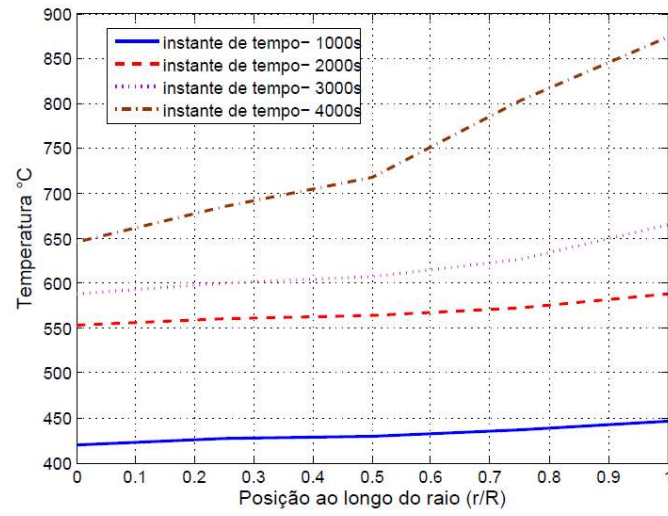


Figura 3.11: Distribuição da Temperatura ao longo da Partícula $r = 15\text{mm}$

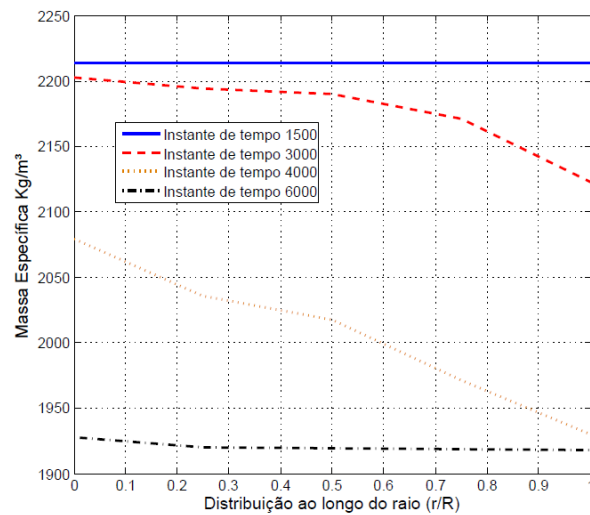


Figura 3.12: Distribuição da Massa Específica ao longo da Partícula - Temperatura Máxima 500°C $r = 30\text{mm}$

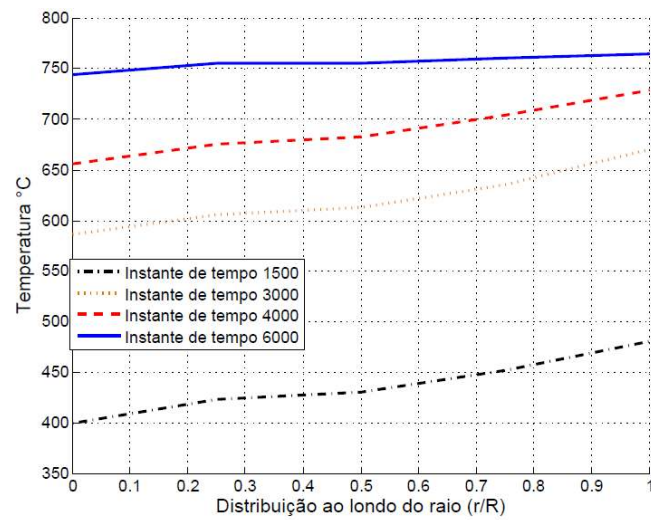


Figura 3.13: Distribuição da Temperatura ao longo da Partícula - Temperatura Máxima 500°C $r = 30\text{mm}$

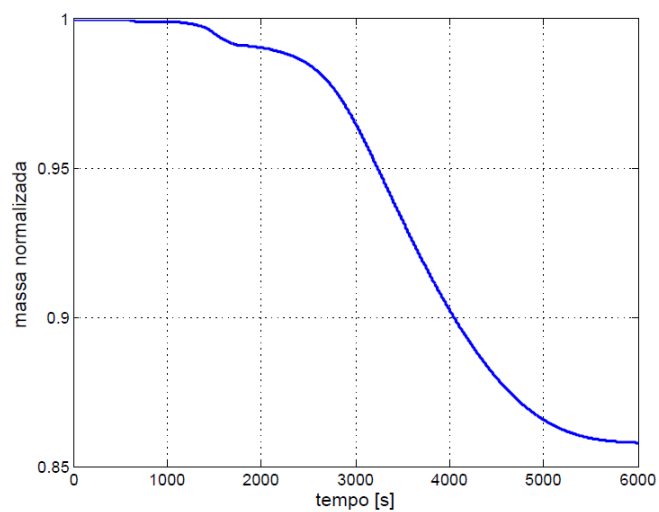


Figura 3.14: Distribuição da Massa Normalizada ao longo da Partícula - Temperatura Máxima 500°C $r = 30\text{mm}$

Capítulo 4

Conclusões

Um modelo matemático - com a equação de conservação da massa e com equações termoquímicas - foi desenvolvido com sucesso. Com o auxílio do Método Iterativo de Newton e Gradiente Biconjugado foi implementado um algoritmo para avaliação de parâmetros termo-físicos envolvidos no processo de pirólise.

A princípio verificou-se a precisão dos resultados através de testes de malha, variando-se o número de pontos nodais, obtendo erros absolutos da ordem de 10^{-3} . O erro relativo - entre as malhas de 10 e 15 pontos - foi da ordem de 10^{-4} .

Pela comparação dos resultados teóricos com os dados experimentais da análise de Termogravimetria, para os casos em que as velocidades de aquecimento da partícula variam de $3^{\circ}C.min^{-1}$ à $10^{\circ}C.min^{-1}$ constatou-se que o modelo teórico coincidia com os dados experimentais durante um amplo intervalo da reação de pirólise, apresentando uma ligeira alteração durante os instantes finais, isto porque nosso modelo não considera as reações de decarbonetação, cujo início se dá no fim da pirólise.

Os estudos paramétricos foram realizados utilizando dois parâmetros, velocidade de aquecimento e raio da partícula. Conforme se devia esperar as reações de pirólise são controladas pelo número de Biot (Bi) e o módulo de Thiele (Py). Se $Py \ll 1$, o processo é limitado pela transferência de calor, ou seja devem ser levados em consideração os efeitos do gradiente de temperatura, e caso $Py \gg 1$, a reação é cineticamente limitada, podendo a massa normalizada ser integrada diretamente na temperatura, equivalente a desconsiderar este gradiente de temperatura. Do mesmo modo vimos que quando $Bi \ll 1$ a transferência de calor é limitada basicamente pela condutividade térmica do material, e para $Bi \gg 1$ haverá um gradiente de temperatura considerável ao longo da partícula.

Apesar do relativo sucesso de nosso modelo físico-matemático existe ainda um fator limitante conforme já foi dito: Os efeitos das reações químicas de oxidação não foram levados em consideração, o que gerou uma falta de precisão ao término da reação de pirólise, podendo ser melhorado com a adição de novas equações químicas na rotina computacional.

Outra inconveniência observada foi o tempo gasto para a obtenção da solução, isto porque apesar das técnicas computacionais de diferenciação simbólica não apresentarem erros de truncamento, elas possuem custos computacional alto, sendo necessário um longo tempo de execução para a resolução de problemas de grande porte.

Este problema poderia ser contornado ao utilizarmos técnicas alternativas, como a Diferenciação Automática, que consegue aliar resultados precisos a uma rápida execução. Estudos ? mostram que estas ferramentas são eficientes para a otimização no tempo de execução dos processos de derivação, fato que o autor verificou com uma redução de 50% no tempo de processamento de determinadas equações.

Apêndice A

Anexos

A.1 equacoes.m

```
1 function [x y C D pp] = equacoes(pp)
2 %
3 %
4 %
5 %[x y C D pp] = equacoes(pp);
6 %
7 %
8 % equacoes(pp) Gera um sistema de equacoes numa malha com 'pp' pontos.
9 % Essas equacoes sao reescritas num arquivo m e utilizadas depois para
10 % resolucao do problema de pirolise. Em equacoes(pp) e gerado tambem a
11 % matriz jacobiana, para implementacao de metodo de gauss-newton para
12 % resolucao de sistemas nao-lineares.
13 % E necessario garantir que nao exista os arquivos .m com mesmo nome dos
14 % arquivos gerados na pasta de trabalho, caso contrario equacoes(pp) nao ira
15 % gerar corretamente novos arquivos
16 %
17 %
18 %Descricao dos argumentos de saida:
19 %
20 %
21 %[x y C D pp] = equacoes(pp);
22 %
23 %
24 % X - matriz Jacobiana do sitema de equacoes do problema de pirolise
25 % Y - equacoes fisico-quimicas discretizadas nos 'pp' pontos
26 % C - argumento de string especificando o atalho do arq m gerado, para
27 % posterior calculo de jacobiano em jac.m
28 % D - argumento de string especificando o atalho do arq m gerado, para
29 % posterior armazenamento de solucao em jac.m
30 % PP - numeros de pontos utilizados na malha esferica
31
32
33 %variaveis simbolicas pra escrita de equacoes
34 syms t w ps t0 wo pso Pso1 r Rg dr dp Dhe Dhp Rg dt Ep Ee Ae Ap pc pso1 w1 w2 w3
    w4 w5 T1 T2 T3 T4 T5 ps1 ps2 ps3 ps4 ps5 xx h tinf sig eps i1 i2 i3 i4 i5
    wo1 wo2 wo3 wo4 wo5 pso1 pso2 pso3 pso4 pso5
35
36 %icognitas do problema inicial
37 t=[T1 T2 T3 T4 T5];
38 w=[w1 w2 w3 w4 w5];
39 ps=[ps1 ps2 ps3 ps4 ps5];
40
41 %chute inicial
42 t0=[to1 to2 to3 to4 to5];
43 wo=[wo1 wo2 wo3 wo4 wo5];
```

```

44 pso=[pso1 pso2 pso3 pso4 pso5];
45
46
47 %dev=[T1 T2 T3 T4 T5 w1 w2 w3 w4 w5 ps1 ps2 ps3 ps4 ps5];
48
49
50
51
52 n=pp*3;
53
54                                     %geracao de vetor, com as icognitas do problema, para
55                                     %uso posterior na obtemcao do jacobiano
56 dev(1:pp)=t(1:pp);
57 dev(pp+1:2*pp)=w(1:pp);
58 dev(2*pp+1:3*pp)=ps(1:pp);
59 dev';
60                                     %vetor analogo ao vetor acima, so que desta vez, sao
61                                     %utilizados os chutes iniciais
62
63 dev2(1:pp)=t0(1:pp);
64 dev2(pp+1:2*pp)=wo(1:pp);
65 dev2(2*pp+1:3*pp)=pso(1:pp);
66
67
68
69 argm='';                                     %geracao de agumentos de strings, para chamada
        posterior
70 for i=1:n
71 argm=strcat(argm, char(dev(i)), ', ');
72 end
73
74                                     %geracao de agumentos de strings,
75                                     %para chamada posterior
76 argm2='';
77 for i=1:n-1
78 argm2=strcat(argm2, char(dev2(i)), ', ');
79 end
80 argm2=strcat(argm2, char(dev2(n)), '');
81
82
83                                     %geracao de agumentos de
84                                     %strings, para chamada posterior
85
86
87 C=strcat(' [A B]=matrizjac_vetB(Pso1,r,Rg,dr,Dhe,Dhp,dt,Ep,Ee,Ae,Ap,pc,h,tinf,
        sig,eps,',argm,argm2,',','');
88 B=strcat(' function [f g]=matrizjac_vetB(Pso1,r,Rg,dr,Dhe,Dhp,dt,Ep,Ee,Ae,Ap,pc,
        h,tinf,sig,eps,',argm,argm2,',','');
89
90
91
92                                     %formato padrao da equacoes
93
94 %ps(i)*(1+w(i))*((0.1031 + 0.003867*t(i) + 4.19*w(i))/(1 + w(i)) + (0.02355*t(i)
        ^(-1.32)*w(i)^(-5.191)))*1000*(t(i)-to(i)) - dt*(((0.2 + 0.404*w(i))*ps(i)
        /1000 + 0.238 )*(0.002*t(i)+0.40368)/(((i-1)*r/(pp-1))^2)*(2*((i-1)*r/(pp-1))
        *((t(i+1)-t(i-1))/dr) +((i-1)*r/(pp-1))^2*(t(i-1)-2*t(i)+t(i+1))/(dr)^2)
        -((1-(0.667+w(i))*ps(i)/1000)*(ps(i)*(1+w(i))-ps(i))*Ae*exp(-Ee/(Rg*t(i)))*
        Dhe+(1-(1-(0.667+w(i))*ps(i)/1000))*(ps(i)-pc)*Ap*exp(-Ep/(Rg*t(i)))*Dhp))
95
96 %(w(i)-wo(i))/dt=-(ps(i)*(1+w(i)) - ps(i))*Ae*exp(-Ee/(Rg*t(i)))/pso1
97
98 %(ps(i)-pso(i))/dt=-(ps(i)-pc)*Ap*exp(-Ep/(Rg*t(i)))
99
100

```

```

101
102         for i=2:pp-1;           %geracao de equacoes para pontos internos  $\tilde{A}$  malha
103
104 f(i,1)=ps(i)*(1+w(i))*((0.1031 + 0.003867*t(i) + 4.19*w(i))/(1 + w(i)) +
      (0.02355*t(i)^(-1.32)*w(i)^(-5.191)))*1000*(t(i)-t0(i)) - dt*((0.2 + 0.404*w
      (i))*ps(i)/1000 + 0.238)*(0.002*t(i)+0.40368)/(((i-1)*r/(pp-1))^2*(2*((i-1)
      *r/(pp-1))*((t(i+1)-t(i-1))/dr) + ((i-1)*r/(pp-1))^2*(t(i-1)-2*t(i)+t(i+1))/(
      dr)^2)-((1-(0.667+w(i))*ps(i)/1000)*(ps(i)*w(i))*Ae*exp(-Ee/(Rg*t(i)))*Dhe
      +(1-(1-(0.667+w(i))*ps(i)/1000))*(ps(i)-pc)*Ap*exp(-Ep/(Rg*t(i)))*Dhp));
105
106 f(i+pp,1)=(w(i)-wo(i))/dt +(ps(i)*(w(i)))*Ae*exp(-Ee/(Rg*t(i)))/Pso1;
107
108 f(i+2*pp,1)=(ps(i)-pso(i))/dt +(ps(i)-pc)*Ap*exp(-Ep/(Rg*t(i)));
109
110         end
111
112
113         for i=1                 %geracao de equacoes para pontos extremo a malha, em r=0
114 f(1,1)=ps1*(1+w1)*((0.1031 + 0.003867*T1 + 4.19*w1)/(1 + w1) + (0.02355*T1
      ^(-1.32)*w1^(-5.191)))*1000*(T1-t0(i)) - dt*((0.2 + 0.404*w1)*ps1/1000 +
      0.238)*(0.002*T1 + 0.40368)*((T2-2*T1+T2)/(dr)^2) - ((1-(0.667+w1)*ps1
      /1000)*(ps1*w1)*Ae*exp(-Ee/(Rg*T1))*Dhe + (1-(1-(0.667+w1)*ps1/1000))*(ps1-
      pc)*Ap*exp(-Ep/(Rg*T1))*Dhp));
115
116 f(i+pp,1)=(w(i)-wo(i))/dt +(ps(i)*(w(i)))*Ae*exp(-Ee/(Rg*t(i)))/Pso1;
117
118 f(i+2*pp,1)=(ps(i)-pso(i))/dt +(ps(i)-pc)*Ap*exp(-Ep/(Rg*t(i)));
119         end
120
121
122         for i=pp                 %geracao de equacoes para pontos extremo a malha, em r=R
123 f(i,1)=ps(i)*(1+w(i))*((0.1031 + 0.003867*t(i) + 4.19*w(i))/(1 + w(i)) +
      (0.02355*t(i)^(-1.32)*w(i)^(-5.191)))*1000*(t(i)-t0(i)) -dt*((0.2 +
      0.404*w(i))*ps(i)/1000 + 0.238)*(0.002*t(i) + 0.40368)/(r^2)*(2*r*-1*((h*(
      tinf-t(i))+ sig*eps*(tinf^4-t(i)^4))/((0.2 + 0.404*w(i))*ps(i)/1000 + 0.238
      )*(0.002*t(i) + 0.40368))) + r^2*(t(i-1)-2*t(i)+tinf)/((dr)^2) )
      -((1-(0.667+w(i))*ps(i)/1000)*(ps(i)*w(i))*Ae*exp(-Ee/(Rg*t(i)))*Dhe +
      (1-(1-(0.667+w(i))*ps(i)/1000))*(ps(i)-pc)*Ap*exp(-Ep/(Rg*t(i)))*Dhp));
124
125 f(i+pp,1)=(w(i)-wo(i))/dt +(ps(i)*(w(i)))*Ae*exp(-Ee/(Rg*t(i)))/Pso1;
126
127 f(i+2*pp,1)=(ps(i)-pso(i))/dt +(ps(i)-pc)*Ap*exp(-Ep/(Rg*t(i)));
128         end
129
130
131 y=f;
132 x=jacobian(f,dev);           %jacobiano simbolico do sistema de equacao
133
134
135
136 diary 'matrizjac_vetB.m'      %geracao de arquivo m para uso posterior, no programa
      jac.m
137 disp(B)
138
139 for i=1:n
140
141     A='['; %formatacao da matriz jacobiano, (impressao sequencial de vetor de
      linhas), para ser escrito no arquivo m
142
143     A=strcat('f(',num2str(i),',:)=',strcat(A,char(x(i,1))));
144
145
146     for j=2:n-1
147         A=strvcat(A,strcat(' ',char(x(i,j))));
148     end

```

```

149
150     A=strvcat(A, strcat(' ', char(x(i,n)), ' ');
151     disp(A);
152 end
153
154
155
156
157     A=' ';%formatacao do vetor de equacoes, (impressao sequencial de vetor de
        linhas), para ser escrito no arquivo m
158
159
160     A=strcat('g=', strcat(A, char(y(1))));
161
162     for i=2:n-1
163     A=strvcat(A, strcat(' ', char(y(i))));
164     end
165
166     A=strvcat(A, strcat(' ', char(y(n)), ' ');
167     disp(A);
168
169
170 diary off %termino de geracao arquivo
171
172
173 diary 'vetB.m' %geracao de arquivo m para uso posterior, no programa jac.m
174 disp(B)
175 disp(A);
176 diary off
177
178 diary 'armazenador.m' %Geracao de arquivo m para armazenar as solucoes de jac.
        m
179
180 D=strcat('function [', argm, argm2, ']=armazenador(a, ', argm, argm2, ') ');
181 disp(D)
182 D=strcat(' [', argm, argm2, ']=armazenador(a, ', argm, argm2, ') ');
183 for i=1:pp
184
185     disp(strcat('to ', num2str(i), '=to ', num2str(i), '+a(', num2str(i), ');'))
186     disp(strcat('wo ', num2str(i), '=wo ', num2str(i), '+a(', num2str(i+pp), ');'))
187     disp(strcat('pso ', num2str(i), '=pso ', num2str(i), '+a(', num2str(i+2*pp), ');'))
188
189     disp(strcat('T ', num2str(i), '=T ', num2str(i), '+a(', num2str(i), ');'))
190     disp(strcat('w ', num2str(i), '=w ', num2str(i), '+a(', num2str(i+pp), ');'))
191     disp(strcat('ps ', num2str(i), '=ps ', num2str(i), '+a(', num2str(i+2*pp), ');'))
192
193 end
194 diary off %termino de geracao arquivo

```

A.2 jac.m

```

1 function y=jac(C,D,pp);
2 %
3 %
4 %y=jac(C,D,pp);
5 %
6 %
7 %
8 %Utilize inicialmente a funcao equacoes.m para gerar o sistema de equacoes,
9 %a Matriz de jacobiano, e os argumentos de string necessarios para chamada
10 %interna de funcoes.
11 %Argumentos de entrada de jac, e sua descricao :

```

```

12 %
13 %y=jac(C,D,pp);
14 %
15 %
16 %y— e a solucao do sistema;
17 %
18 %pp— pontos utilizados na malha esferica, deve em concordancia com o
19 %com o numero de pontos utilizados em equacoes.m
20 %
21 %D— argumento de strings gerado em equacoes.m, para chamada de funcao de
    armazenamento
22 %de solucao
23 %
24 %C— argumento de strings gerado em equacoes.m para chamada de funcao para
25 %obtencao da matriz jacobiano e vetor "B", em A*X=B
26 %
27 %
28 %Para alteracao de condicoes iniciais, e constantes fisicas, modificar
29 %diretamente no arquivo codigo.
30
31 %syms t w ps t0 wo pso Pso1 r Rg dr dp Dhe Dhp Rg dt Ep Ee Ae Ap pc pso1 w1 w2
    w3 w4 w5 T1 T2 T3 T4 T5 ps1 ps2 ps3 ps4 ps5 xx h tinf sig eps i1 i2 i3 i4 i5
    wo1 wo2 wo3 wo4 wo5 pso1 pso2 pso3 pso4 pso5
32
33 %dev=[T1 T2 T3 T4 T5 w1 w2 w3 w4 w5 ps1 ps2 ps3 ps4 ps5];
34
35
36
37
38 for i=1:pp                                %aqui e gerada a condicao inicial
39     eval(strcat('T',num2str(i),'=300;'))
40     eval(strcat('w',num2str(i),'=0.07;'))
41     eval(strcat('ps',num2str(i),'=450;'))
42     eval(strcat('to',num2str(i),'=300;'))    %aqui e declarado o chute inicial
        de solucao
43     eval(strcat('wo',num2str(i),'=0.07;'))
44     eval(strcat('pso',num2str(i),'=450;'))
45 end
46
47
48 arg=''; %Concatenacao de strings para gerar o vetor Temperatura que armazenara
    as solucoes, em cada instante de tempo
49 arg=strcat(arg,'T',num2str(1),' ');
50 for i=2:pp-1
51     arg=strcat(arg,strcat('T',num2str(i)),' ');
52 end
53 arg=strcat(arg,strcat('T',num2str(pp)),' ');
54
55
56 arg2=''; %Concatenacao de strings para gerar o vetor Umidade que armazenara as
    solucoes, em cada instante de tempo
57 arg2=strcat(arg2,'w',num2str(1),' ');
58 for i=2:pp-1
59     arg2=strcat(arg2,strcat('w',num2str(i)),' ');
60 end
61 arg2=strcat(arg2,strcat('w',num2str(pp)),' ');
62
63
64 arg3=''; %Concatenacao de strings para gerar o vetor Densidade que armazenara
    as solucoes, em cada instante de tempo
65 arg3=strcat(arg3,'ps',num2str(1),' ');
66 for i=2:pp-1
67     arg3=strcat(arg3,strcat('ps',num2str(i)),' ');
68 end
69 arg3=strcat(arg3,strcat('ps',num2str(pp)),' ');

```

```

70
71 arg=strcat(arg,arg2,arg3); %Concatenacao dos 3 Vetores acima para armazenar a
    solucao geral do sistema
72
73 Wo=0.07;          %Declaracao de paremetros, constantes fisico-quimicas,
    propriedades termicas, etc
74 pc=95;
75 sg=450;
76 To=300;
77 Dhp=4500e3;
78 Dhe=2700e3;
79 Ee=120e3;
80 Ep=124e3;
81 Ae=5.6e6;
82 Ap=8.35e9;
83 dt=0.01;
84 r=0.0085;
85 Rg=8.31;
86 Psol=sg;
87 h=5;
88 tinf=670;
89 eps=0.9;
90 sig=5.67e-8;
91 dr=r/(pp-1);
92 dp=1;
93 rp=1;
94
95
96 xx(:,1)= eval(arg)';          %avaliacao e armazenamento do vetor da solucao
97
98
99
100
101 for k=2:10000 %marcha no tempo
102
103
104     tinf=445+285*0.01/130*k;
105
106     eval(C)          %chamada da funcao para obtencao da matriz jacobiano, e da
        matriz "B", para resolucao de A*X=B
107
108
109     a=bicg(A,-B,1e-6,450); %resolucao do sistema inicial por gradiente bi-
        conjugado
110     k
111
112
113     eval(D) %chamada de funcao, p/obtencao de novos T, W, Ps, To, Wo, Pso, para
        posterior armazenamento de solucao
114
115
116
117
118     xx(:,k)= eval(arg)'; %avaliacao e armazenamento do vetor da solucao
119
120 end
121 y=xx;

```

A.3 massaN.m

```

1 function a=massaN(x,r,dt);
2 % integrador numerico da solucao do problema de pirolise:

```

```

3 % a massa normalizada e a saida desse algoritmo
4 %
5 %Especificacoes:
6 %x - o vetor solucao do problema de pirolise
7 %r - o raio adotado no problema de pirolise
8 %dt - intervalo de tempo considerado na solucao do problema de pirolise
9 %
10 [l,c]=size(x);
11
12
13 b(1:l/3,1)=linspace(0,r,l/3);
14 b=b.*b*4*pi;
15 raio=linspace(0,r,l/3);
16
17 for i=1:c;
18
19 xx=x(2*(l/3)+1:l,i) ;
20
21 yy=x((l/3)+1:2*(l/3),i).*x(2*(l/3)+1:l,i);
22
23
24 a(i,1)=trapz(raio,xx.*b);
25 a(i,2)=trapz(raio,yy.*b);
26 a(i,3)=(a(i,1)+a(i,2))/(a(1,1)+a(1,2));
27
28
29 end
30
31
32 t=linspace(24,+24+c*dt*3/60,c);
33 plot(t,a(:,3))
34 grid on
35 xlabel('tempo [s]')
36 ylabel('massa normalizada')

```


Referências Bibliográficas

- Benkoussas et al.(2007)** B. Benkoussas, J.-L. Consalvi, B. Porterie, N. Sardoy e J.-C. Loraud. Modelling thermal degradation of woody fuel particles. *International Journal of Thermal Sciences*, 46(4):319 – 327. ISSN 1290-0729. doi: DOI:10.1016/j.ijthermalsci.2006.06.016. URL <http://www.sciencedirect.com/science/article/pii/S1290072906001657>. Citado na pág. 1
- Blasi(1996)** Colomba Di Blasi. Heat, momentum and mass transport through a shrinking biomass particle exposed to thermal radiation. *Chemical Engineering Science*, 51(7):1121 – 1132. ISSN 0009-2509. doi: DOI:10.1016/S0009-2509(96)80011-X. URL <http://www.sciencedirect.com/science/article/pii/S000925099680011X>. Citado na pág. 1
- FRANCO(2007)** Neide Maria Bertoldi FRANCO. *Calculo numerico*. Pearson Prentice Hall. Citado na pág. 8
- GOLUB(1966)** Charles F GOLUB, Gene H.; VAN LOAN. *Matrix computations*. Johns Hopkins University, 3rd ed edição. Citado na pág. 2, 9
- Li(2005)** Jianping Li. General explicit difference formulas for numerical differentiation. *Journal of Computational and Applied Mathematics*, 183(1):29 – 52. ISSN 0377-0427. doi: DOI:10.1016/j.cam.2004.12.026. URL <http://www.sciencedirect.com/science/article/pii/S0377042704006454>. Citado na pág. 2, 7
- Marco A. B. Zanonía(2010)** Marcio F. Martins Marco A. B. Zanonía. Modeling the thermochemical conversion of single wood particle. Em *23rd International Conference on Efficiency, Cost, Optimization, Simulation and Environmental Impact of Energy Systems*. Citado na pág. 5, 6
- Marco A. B. Zanonía i(2010)** Marcio F. Martins Marco A. B. Zanonía i. Applying inverse problem to determine the reactions parameters for oil shale combustion. Em *13th Brazilian Congress of Thermal Sciences and Engineering*. Citado na pág. 16
- Martins et al.(2010)** M.F. Martins, S. Salvador, J.-F. Thovert e G. Debenest. Co-current combustion of oil shale - part 2: Structure of the combustion front. *Fuel*, 89(1):133 – 143. ISSN 0016-2361. doi: DOI:10.1016/j.fuel.2009.06.040. URL <http://www.sciencedirect.com/science/article/pii/S0016236109003433>. Citado na pág. 1
- Ouedraogo et al.(1998)** Abdoulaye Ouedraogo, J.C. Mulligan e John G. Cleland. A quasi-steady shrinking core analysis of wood combustion. *Combustion and Flame*, 114(1-2):1 – 12. ISSN 0010-2180. doi: DOI:10.1016/S0010-2180(97)00291-5. URL <http://www.sciencedirect.com/science/article/pii/S0010218097002915>. Citado na pág. 1
- Peters e Bruch(2003)** Bernhard Peters e Christian Bruch. Drying and pyrolysis of wood particles: experiments and simulation. *Journal of Analytical and Applied Pyrolysis*, 70(2):233 – 250. ISSN 0165-2370. doi: DOI:10.1016/S0165-2370(02)00134-1. URL <http://www.sciencedirect.com/science/article/pii/S0165237002001341>. Citado na pág. 1, 5, 18
- Pyle e Zaror(1984)** D.L. Pyle e C.A. Zaror. Heat transfer and kinetics in the low temperature pyrolysis of solids. *Chemical Engineering Science*, 39(1):147 – 158. ISSN 0009-2509.

doi: DOI:10.1016/0009-2509(84)80140-2. URL <http://www.sciencedirect.com/science/article/pii/0009250984801402>. Citado na pág. 1

Saastamoinen e Richard(1996) Jaakko Saastamoinen e Jean-Robert Richard. Simultaneous drying and pyrolysis of solid fuel particles. *Combustion and Flame*, 106(3):288 – 300. ISSN 0010-2180. doi: DOI:10.1016/0010-2180(96)00001-6. URL <http://www.sciencedirect.com/science/article/pii/0010218096000016>. Citado na pág. 1