

**UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA QUÍMICA**

**ÁREA DE CONCENTRAÇÃO  
SISTEMAS DE PROCESSOS QUÍMICOS E INFORMÁTICA**

**MÉTODOS MATEMÁTICOS EM  
MODELAGEM E SIMULAÇÃO DO CRAQUEAMENTO  
TÉRMICO DO 1,2-DICLOROETANO**

Autor (a): Helianildes Silva Ferreira.

Orientador (a): Prof. Dr. João Alexandre Ferreira da Rocha Pereira.

Co-orientador (a): Profa. Dra. Maria de Fátima dos Santos Lopes.

Tese de Doutorado apresentada à Faculdade de Engenharia Química como parte dos requisitos exigidos para a obtenção do título de Doutor em Engenharia Química.

**Campinas – São Paulo**

**Julho de 2003**

**UNICAMP  
BIBLIOTECA CENTRAL  
SEÇÃO CIRCULANTE**

UNIDADE	58
Nº CHAMADA	UNICAMP F413m
V	EX
TOMBO, BC/	56877
PROC.	16/12/04
C	<input type="checkbox"/>
D	<input checked="" type="checkbox"/>
PREÇO	11,00
DATA	20/01/2004
Nº CPD	

CM00194015-3

bb id 309334

FICHA CATALOGRÁFICA ELABORADA PELA  
BIBLIOTECA DA ÁREA DE ENGENHARIA - BAE - UNICAMP

F413m Ferreira, Helianildes Silva  
Métodos matemáticos em modelagem e simulação do craqueamento térmico do 1, 2-Dicloroetano / Helianildes Silva Ferreira. --Campinas, SP: [s.n.], 2003.

Orientadores: João Alexandre Ferreira da Rocha Pereira e Maria de Fátima dos Santos Lopes  
Tese (doutorado) - Universidade Estadual de Campinas, Faculdade de Engenharia Química.

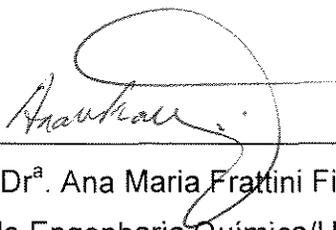
1. Craqueamento. 2. Cloreto de vinila. 3. Modelagem de dados. 4. Análise multivariada. 5. Redes neurais (Computação). 6. Simulação (Computadores). 7. Modelos matemáticos. I. Pereira, João Alexandre Ferreira da Rocha. II. Lopes, Maria de Fátima dos Santos. III. Universidade Estadual de Campinas. Faculdade de Engenharia Química. IV. Título.

Tese de Doutorado defendida por Helianildes Silva Ferreira e aprovada em  
25 de julho de 2003 pela Banca Examinadora constituída por:



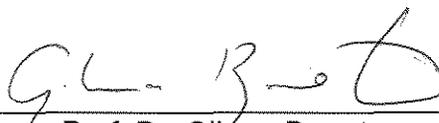
---

Prof. Dr. João Alexandre Ferreira da Rocha Pereira  
Presidente e Orientador  
Faculdade de Engenharia Química/UNICAMP



---

Prof.ª Dr.ª Ana Maria Frattini Fileti  
Faculdade de Engenharia Química/UNICAMP



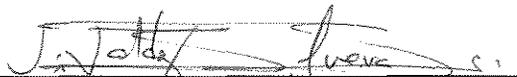
---

Prof. Dr. Gilmar Barreto  
Faculdade de Engenharia Elétrica e de Computação/UNICAMP



---

Prof. Dr. José Vicente Hallak d'Angelo  
Faculdade de Engenharia Química/UNICAMP



---

Prof. Dr. Vivaldo Silveira Júnior  
Faculdade de Engenharia de Alimentos /UNICAMP

Campinas, 25 de Julho de 2003.

Este exemplar corresponde à versão final da Tese de Doutorado em Engenharia Química.



---

João Alexandre Ferreira da Rocha Pereira  
Orientador

## **Dedicatória**

A Deus e Jesus Cristo, razão da minha existência, que me ensinaram que um sonho pode ser transformado em realidade pela fé.

Ao meu marido Sergio Ferreira, pelo carinho, compreensão e amor de todos os dias...

Aos meus filhos, Andréa e Rodrigo Ferreira pela felicidade e encontro de todos os dias...

Aos meus pais pelo amor e dedicação de sempre...

Aos meus irmãos que sempre foram a motivação para o meu crescimento profissional.

## **Agradecimentos**

Em especial, ao Professor Dr. João Alexandre F. da Rocha Pereira por ter me orientado neste trabalho e pela sua compreensão e amizade.

A Professora Maria de Fátima dos Santos Lopes que um dia me ofereceu a mão para que eu me desenvolvesse na pesquisa acadêmica. Pela sua amizade e estímulo em todos os momentos em que estivemos juntas.

Aos colegas e amigos de trabalho, que sempre me estimularam e me deram apoio, dentre eles em especial a Carlos Roberto de Souza, Alex Antônio Conceição Santiago, Luís Mário, Arlinda Vianna e a Gustavo Leal Sales Filho com quem convivi alguns anos de minha vida profissional.

A todos os colegas do SENAI-CETIND que me incentivaram e apoiaram no sentido de realizar este trabalho, especialmente a Lídia pelo auxílio na bibliografia necessária.

Aos professores, funcionários e alunos do Mestrado em Engenharia Química da UFBA e da UNICAMP que compartilharam comigo a realização deste trabalho. Em especial aos Professores Samuel Luporini, Frede Oliveira e a colega Valéria Azevedo.

Ao meu marido Sérgio Ferreira pela compreensão e amor de todos os dias e aos meus filhos Andréa e Rodrigo Ferreira pelo amor e carinho de sempre.

Aos meus pais, Manoel e Damiana e aos meus irmãos Helenice, Josué, Diva, Divanildes, Ailton, Cristina e Jailton.

Á todos aqueles que de maneira direta ou indireta contribuíram para a realização deste trabalho.

*O SENHOR, com sabedoria fundou a terra; com entendimento preparou os céus.  
Pelo seu conhecimento se fenderam os abismos, e as nuvens destilam o orvalho.  
Filho meu, não se apartem estas coisas dos teus olhos: guarda a verdadeira  
sabedoria e o bom siso;  
Porque serão vida para a tua alma, e adorno ao teu pescoço.  
Então andarás confiante pelo teu caminho, e o teu pé não tropeçará.*

**Provérbios 3:19-23.**

---

## RESUMO

A simulação de processos químicos é de fundamental importância para a melhoria e otimização das unidades industriais existentes. A modelagem do processo envolve a descrição fenomenológica do mesmo e a resolução das equações obtidas através de métodos matemáticos adequados. Esta é a simulação dita convencional. Outra forma de abordagem do problema pode ser através da utilização de ferramentas computacionais que sejam capazes de "aprender" o comportamento do processo através dos seus dados históricos ou simulados, e desta forma, conseguir prever o comportamento do processo em condições futuras. Uma vez modelado, o processo pode ser simulado e os resultados obtidos nesta simulação são validados através de comparação com os resultados reais obtidos nas unidades industriais ou através de modelos de simulação, sendo que neste trabalho foram utilizados modelos de simulação.

Tem sido crescente a necessidade de otimizar e controlar os processos industriais com os objetivos de redução de custos, melhoria na qualidade do produto e segurança operacional. Para isto faz-se necessário simular o comportamento dinâmico do processo e avaliar a resposta do mesmo quanto a mudanças em suas variáveis de entrada.

No presente trabalho foi simulado o comportamento dinâmico do processo que ocorre na serpentina de uma fornalha para o craqueamento térmico do 1,2-dicloroetano através da simulação convencional e não-convencional. Na simulação convencional utilizamos os métodos das Diferenças Finitas, Colocação Ortogonal e Runge-Kutta para resolver o sistema de equações diferenciais parciais. A simulação convencional deste processo pode ser útil para fins de avaliação operacional, otimização de processo e construção de modelos para simulação e controle de processos.

Na simulação não convencional do processo foi utilizada a metodologia de Redes Neurais Artificiais (RNA). As RNA's são particularmente úteis em aplicações de controle de processos onde são necessárias maiores velocidades de processamento para que os objetivos do controle sejam alcançados.

---

Para acelerar o treinamento das redes neurais artificiais (RNA), foi empregada uma técnica estatística multivariada denominada Análise de Componentes Principais (PCA). Através da utilização desta técnica é possível melhorar o desempenho da simulação e selecionar as variáveis ou combinação de variáveis mais importantes no processo. Também foi utilizada a Regressão por Componentes Principais (PCR) para modelar e simular a conversão da reação.

De maneira geral o processo foi representado por um modelo dinâmico que considera os princípios da conservação da massa e energia na serpentina onde ocorre a reação. As equações obtidas foram resolvidas numericamente pelos métodos das diferenças finitas e colocação ortogonal, sendo que o método de Runge-Kutta foi utilizado para as equações diferenciais ordinárias no tempo. Dentre os métodos utilizados para resolver numericamente as equações diferenciais parciais, o que apresentou melhores resultados foi o método das diferenças finitas. Os resultados obtidos foram compatíveis com os esperados teoricamente. O método da colocação ortogonal não ofereceu bons resultados.

Quanto à aplicação da simulação não convencional através da metodologia de redes neurais artificiais, observou-se que o aumento da taxa de aprendizagem diminui o coeficiente de correlação entre as saídas preditas e desejadas, aumenta o erro quadrático médio durante a evolução do treinamento podendo provocar um sobre-ajuste dos dados. Foi observado que o pré-processamento dos dados de entrada influenciam significativamente no treinamento da rede. Foi verificado também que o aumento da taxa de aprendizagem diminui o tempo de processamento da rede provocando uma desestabilização na trajetória do espaço dos erros.

De maneira geral o aumento do número de neurônios na rede favoreceu a uma diminuição do erro, um aumento do tempo de processamento da rede e uma redução do número de ciclos de treinamento. Foram utilizados alguns algoritmos de treinamento da rede, dentre eles o do Gradiente Descendente e o de Levenberg-Marquadt, sendo que o algoritmo de Levenberg-Marquadt oferece menores erros com maior tempo de processamento. Este método também conduziu a melhores coeficientes de correlação entre os valores preditos pela rede

---

e os valores desejados, entretanto ele pode conduzir a sistemas com rigidez, difíceis de serem resolvidos numericamente.

Foram também utilizadas redes recorrentes para acelerar o treinamento das redes neurais artificiais sendo que houve uma redução significativa do número de ciclos de treinamento para a predição da conversão.

A análise dos componentes principais (PCA) foi utilizada como pré-processamento dos dados de entrada de forma a reduzir a dependência das variáveis de entrada, dentro de um grau de significância previamente estipulado. Através da aplicação desta metodologia, verificou-se que das sete variáveis de entrada do craqueamento, apenas três das componentes principais são suficientes para descrever o processo representando 99,9% da informação da matriz de entrada.

De maneira geral, a aplicação da análise dos componentes principais (PCA) favoreceu a uma redução do número de ciclos de treinamento da rede e descreveu pelas componentes principais as variáveis ou combinação de variáveis mais significativas do processo em estudo.

Foi construído e utilizado um modelo de regressão por componentes principais (PCR) para a predição da conversão do craqueamento térmico do 1,2-dicloroetano. Observou-se uma maior rapidez na construção e utilização do modelo quando comparado às redes neurais artificiais (RNA) e que os resultados têm uma aplicabilidade restrita a uma faixa da variável predita sendo aconselhável dividir os conjuntos de treinamento.

---

## ABSTRACT

The simulation of chemical processes is of fundamental importance for the improvement and optimization of the existing industrial units. The modeling of the process involves the phenomenological description of it and the resolution of the equations through the suitable mathematical methods. This is the so-called conventional simulation. Another way of development is the use of computational tools that "learn" the behavior of the process through historical or simulated data and based on this can foresee the behavior of the process in future conditions. Once modeled, the process can be simulated and the results obtained in this simulation generally are compared to the real results obtained in industrial units or through simulated models, being that in this work simulation models had been used. It has been increasing the need to optimize and control industrial processes with the objectives of costs reduction, improve in product quality and operational safety. For this, it's necessary to simulate the dynamic behavior of the process and evaluate the answer of it in front of variations in your input variables.

In the present work it was simulated the dynamic behavior of the process that occurs in the coil of one furnace to the thermal cracking of 1,2-dichloroethane through the conventional and non-conventional simulation. In the conventional simulation it was used Finite Differences, Orthogonal Collocation and Runge-Kutta methods to solve the systems of partial differential equations. The conventional simulation of this process could be useful for ends of operational evaluation, process optimization and construction of models for simulation and control of processes.

In the non-conventional simulation of the process it was used the methodology of artificial neural nets (ANN). The ANN's are particularly useful in applications of processes control where are necessary higher processing velocities so that the control objectives are reached.

To speed up the training of the artificial neural nets (ANN), it was used one technique multivaried statistics called Principal Componente Analysis (PCA). Through the use of this technique is possible to improve the performance of the simulation and select the variables or combination of variables more important in

---

the process. It was also used the Principal Component Regression (PCR) to model and simulate the conversion of the reaction.

In general way the process was represented by a dynamic model that considers the principles of mass and energy conservation in the coil where the reaction occurs. The obtained equations were numerically solved by the finite differences method and orthogonal collocation, being that the Runge-Kutta method was used to the ordinary differential equations in the time. Amongst the methods used to solve numerically the partial differential equations, what presented better results was the difference finite method. The obtained results had been compatible with the waited ones theoretically. The orthogonal collocation method didn't give good results.

According to the application of the non-conventional simulation through the methodology of the artificial neural nets, it was observed that the increasing of the learning rate decreases the correlation coefficient between the predicted and the desired output, increases the mean quadratic error during the evolution of the training being able to provoke a on-adjustment of the data. It was observed that the daily pay-processing of the input data influences significantly in the training of the net. It was also verified that the increasing in the learning rate decreases the processing time of the net provoking a run down in the trajectory of the space of the errors.

In general way the increase in the number of neurons of the net favored to a reduction of the error, an increase of the processing time of the net and reduction in the number of training cycles. One used some algorithms of training of the net, amongst them of the Descending Gradient and the Levenberg-Marquadt being that the Levenberg-Marquadt give lower errors with higher processing time. This method also lead to better correlation coefficients between the predicted and desired values of the net, however it can lead to systems with rigidity ("stiffness") that are difficult to solve numerically.

It was also used recurrent nets to speed up the training of the artificial neural nets being that it had one significant reduction of the numbers of cycles of training for the prediction of the conversion.

---

The Principal Component Analysis (PCA) was used as daily pay-processing of the input data of form to reduce significantly of the input variables, inside a degree of significance previously stipulated. Through the application of this methodology it was verified that from the seven input variables of the cracking only three of the principal components are enough to describe the process representing 99,9% of the information of the entrance matrix.

In general way, the application of the principal component analysis (PCA) favored to a reduction in the number of cycles of training of the net and described by the principal component analysis the variables or combination of variables more significant of the process in study

It was built and used one regression model by principal components for the prediction of the conversion in the thermal cracking of 1,2-dichloroethane. It was observed a higher rapidity in building and using the model when compared to the artificial neural nets (ANN) and that the results has a restricted applicability to a band of the predicted variable being advisable to divide the sets of training.

## SUMÁRIO

RESUMO	xiii
ABSTRACT	xvii
NOMENCLATURA	xxv
INTRODUÇÃO	1
1. REVISÃO BIBLIOGRÁFICA	7
1.1 - PROCESSO DE PRODUÇÃO DO MONOCLORETO DE VINILA	7
1.2 - MODELOS MATEMÁTICOS	8
1.3 - REDES NEURAS ARTIFICIAIS	14
1.4 - REDES NEURAS EM PROCESSOS QUÍMICOS	19
1.5 - ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)	23
1.6 - PRÉ-PROCESSAMENTO DAS REDES NEURAS ARTIFICIAIS	24
1.7 - CONCLUSÕES	25
2. MODELAGEM DINÂMICA DO CRAQUEAMENTO TÉRMICO	27
2.1 - DEDUÇÃO DA EQUAÇÃO DE CONSERVAÇÃO DA MASSA PARA OS COMPONENTES	27
2.1.1 - Balanço de massa dos componentes	31
2.2 - DEDUÇÃO DA EQUAÇÃO DE CONSERVAÇÃO DA ENERGIA	31
2.3 - DEDUÇÃO DA EQUAÇÃO DA QUANTIDADE DE MOVIMENTO	35
2.4 - ADIMENSIONALIZAÇÃO DAS EQUAÇÕES	37
2.4.1 - Balanço de Massa	37
2.4.2 - Balanço de Energia	37

2.4.3 - Balanço da Quantidade de Movimento	38
3 - MÉTODOS MATEMÁTICOS	41
3.1 - INTRODUÇÃO	41
3.2 - DIFERENÇAS FINITAS	41
3.3 - COLOCAÇÃO ORTOGONAL	44
3.4 - MÉTODOS DE RUNGE-KUTTA	47
3.4.1 - Métodos de Runge-Kutta de 2 <sup>a</sup> . Ordem	48
3.4.2 - Métodos de Runge-Kutta de Ordens mais Elevadas	50
4 - RESULTADOS E DISCUSSÕES	53
4.1 - APLICAÇÃO DO MÉTODO DAS DIFERENÇAS FINITAS ÀS EQUAÇÕES DE BALANÇO	53
4.1.1 - Resultados da Aplicação - Método das Diferenças Finitas	55
4.1.2 - Análise da Influência do número de pontos na malha	61
4.2 - APLICAÇÃO DO MÉTODO DA COLOCAÇÃO ORTOGONAL ÀS EQUAÇÕES DE BALANÇO	63
4.2.1 - Balanço de Massa	64
4.2.2 - Balanço de Energia	66
4.2.3 - Resultados da Aplicação - Método da Colocação Ortogonal	67
4.3 - REDES NEURAIIS ARTIFICIAIS - ESTADO ESTACIONÁRIO	72
4.3.1 - Funções de Ativação da Rede	74
4.3.2.- Predição da Conversão da Reação	75
4.3.2.1 - Análise após Regressão	76
4.3.2.2 - Efeito das Funções de Ativação	77

4.3.2.3 - Influencia da Taxa de Aprendizado	77
4.3.2.3.1 - Comparação do Efeito da Taxa de Aprendizado	78
4.3.2.4 - Influencia do Número de Neurônios na Camada Oculta	79
4.3.2.5 - Análise do Efeito do Algoritmo de Treinamento da Rede	82
4.3.2.6 - Influencia do Algoritmo de Treinamento da Rede - Gradiente Descendente e Levenberg-Marquadt	84
4.3.2.7 - Aceleração do Treinamento da Rede pela utilização dos Pesos e Bias de Treinamentos Anteriores	86
4.3.3 - Predição da Seletividade	88
4.3.3.1 - Efeito do Pré-processamento da Rede	92
4.3.4 - Predição do Tempo de Residência	95
4.3.5 - Aplicação da PCA como Pré-Tratamento nas Redes Neurais Artificiais	97
4.4 - ANÁLISE DOS COMPONENTES PRINCIPAIS (PCA)	101
4.4.1 - Aplicação da Análise dos Componentes Principais à Matriz X	104
4.5 - REGRESSÃO EM COMPONENTES PRINCIPAIS (PCR)	111
4.5.1 - Construção do Modelo	111
4.5.2 - Utilização do Modelo	112
5 - CONCLUSÕES E SUGESTÕES PARA TRABALHOS FUTUROS	115
5.1 - CONCLUSÕES	115
5.2 - SUGESTÕES PARA TRABALHOS FUTUROS	118
REFERÊNCIAS BIBLIOGRÁFICAS	121

ANEXOS	129
ANEXO 1 – METODOLOGIA DE REDES NEURAIS ARTIFICIAIS – ALGORITMO DE TREINAMENTO BACKPROPAGATION	129
ANEXO 2 – FLUXOGRAMAS DOS PROGRAMAS UTILIZADOS	135
ANEXO 3 – PROGRAMAS EM FORTRAN	143
3.1 - MÉTODO DAS DIFERENÇAS FINITAS	143
3.2 - PROGRAMA PARA O CÁLCULO DAS RAÍZES DA COLOCAÇÃO ORTOGONAL E DOS COEFICIENTES $A(I,J)$ - DMATA.FOR	159
3.3 – MÉTODO DA COLOCAÇÃO ORTOGONAL	163
ANEXO 4 – DADOS PROCESSADOS NA RNA	180

## NOMENCLATURA

### LETRAS LATINAS

- A(I), B(I) - parâmetros para o cálculo da capacidade calorífica do componente i;
- AK(I), BK(I), CK(I)- parâmetros para o cálculo da condutividade térmica do componente i;
- ALFA - difusividade térmica da mistura gasosa ( $m^2/h$ );
- AV(I), BV(I), CV(I) - parâmetros para o cálculo da viscosidade do componente i;
- BL - comprimento do Joelho (m);
- C<sub>EDC</sub> - fração mássica do 1,2-dicloroetano (%p/p);
- C<sub>MVC</sub> - fração mássica do monocloreto de vinila (%p/p);
- C<sub>HCl</sub> - fração mássica do ácido clorídrico (%p/p);
- C(I) - concentração molar do componente i ( $kgmol/m^3$ );
- CKT - condutividade térmica do coque ( $kcal/m^2hK$ );
- CONV - conversão da reação (%);
- CP(I) - capacidade calorífica do componente i ( $kcal/kgmolK$ );
- CPM - capacidade calorífica da mistura gasosa ( $kcal/kgmolK$ );
- CT(I) - constante cinética da taxa da reação química( $h^{-1}$ );
- DCP(I) - calor sensível de aquecimento e resfriamento dos reagentes e produtos da reação ( $kcal/kgmolK$ );
- DE - diâmetro externo do tubo (m);
- DELTAH0(I) - calor de reação padrão da reação i à 25°C ( $kcal/kgmol$ );
- DELTAQ(I) - termo de calor necessário à reação química i;
- DI - diâmetro interno do tubo (m);
- EDC - 1,2- dicloroetano;

EA(I) - energia de ativação da reação i (kcal/kgmol);

EC - espessura da camada de coque (m);

EP - espessura da parede do tubo (m);

F(I) - funções que representam as equações diferenciais de conservação do sistema;

FF - fator de fricção de Fanning;

FM(I) - fração molar do componente i (%);

FQR – Fluxo térmico (kcal/m<sup>2</sup>h);

FR -parâmetro de fricção;

GAMA1 – G - velocidade mássica (kg/m<sup>2</sup>h);

GAMA2 - kgmol/kg m;

H - passo de integração (m);

H0 - coeficiente de película externo (kcal/hm<sup>2</sup>K);

H1 - coeficiente de película interno (kcal/hm<sup>2</sup>K);

M<sub>j</sub> = massa molar do componente j;

MVC - monocloreto de vinila;

NI - nº de equações diferenciais;

N(I) - número de moles do componente i (kgmol);

NC - nº de componentes;

NE - número total de elementos (tubos+joelhos);

NJ - número de joelhos na serpentina;

NP - nº de pontos para o intervalo de impressão;

NR - nº de reações na rede;

NT - número total de tubos da serpentina;

Nu – UNUD - número de Nusselt;

PM(I) - peso molecular do componente i (kg/kgmol);

PMM - peso molecular médio da mistura gasosa (kg/kgmol);

Pr - número de Prandtl;

PT - pressão total da corrente na entrada (kgf/cm<sup>2</sup> abs);

QT - fluxo de calor através da serpentina (kcal/h);

R(I) - taxa da reação química i (kgmol/m<sup>3</sup>h);

RB - raio do joelho (°);

RBM - raio do joelho (m);

Re - número de Reynolds

RG - constante universal dos gases (kcal/kgmolK);

RI - raio interno do tubo considerando a espessura do tubo e a espessura do coque (m);

ROM - massa específica da mistura gasosa (kg/m<sup>3</sup>);

RT - raio interno do tubo considerando a espessura do mesmo (m);

S(I) - fator pré-exponencial da reação i (h<sup>-1</sup>);

Sel - seletividade da reação (%);

TC - concentração molar da mistura (kgmol/m<sup>3</sup>);

TE - espaçamento centro a centro entre os tubos (m);

TK(I) - condutividade térmica do componente i (kcal/m<sup>2</sup>hK);

TKM - condutividade térmica da mistura gasosa (kcal/m<sup>2</sup>hK);

TL - comprimento do tubo (m);

TN - número total de moles (kgmol);

TP - temperatura da mistura gasosa (K);

TR - tempo de residência da mistura gasosa na fornalha (seg);

TS - temperatura interna da parede do tubo (°C);

TW - temperatura interna da parede do tubo (K);

UI - coeficiente global de troca térmica (kcal/hm<sup>2</sup>K);

V(l) - vetor posição na serpentina (m);  
R0 - raio externo do tubo (m);  
VISA(l) - viscosidade absoluta do componente i (kg/mh);  
VISAM - viscosidade absoluta da mistura gasosa (kg/mh);  
VISCAM - viscosidade cinemática da mistura gasosa(m<sup>2</sup>/h);  
VM - vazão mássica (kg/h);  
VV - velocidade da mistura gasosa (m/h);  
w<sub>j</sub> = fração mássica do componente j;  
W10 - fração mássica do EDC na entrada do reator (% p/p);  
Y(1) - fração mássica do EDC (% p/p);  
Y(2) - fração mássica do MVC (% p/p);  
Y(3) - fração mássica do HCl (% p/p);  
Y(4) - temperatura da mistura gasosa (K);  
Y(5) - 1/Mm - inverso do peso molecular da mistura gasosa (kgmol/kg);  
Y(6) - pressão total do sistema (kgf/cm<sup>2</sup> abs);  
Y(7) - fração mássica do C<sub>2</sub>H<sub>2</sub> (% p/p);  
Y(l) - valores das variáveis do sistema;  
ZF - comprimento total da serpentina (m);

## LETRAS GREGAS

$\rho$  - massa específica (kg/m<sup>3</sup>);

$\nu$  - viscosidade cinemática;

$\mu$  - viscosidade absoluta;

$\alpha$  - difusividade térmica;

## Subscritos

i -

j -

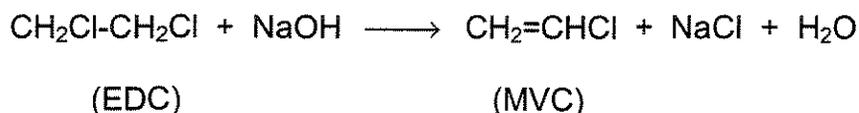
## INTRODUÇÃO

### *Processos de Produção do Monocloreto de Vinila*

A indústria do monocloreto de vinila (MVC) está intimamente relacionada com a do polímero policloreto de vinila (PVC), uma vez que aquele é matéria prima para a obtenção deste. O crescimento da indústria de PVC traz, portanto, grandes perspectivas na otimização e evolução dos processos de produção do monômero monocloreto de vinila (MVC).

O craqueamento do 1,2-dicloroetano (EDC) para a obtenção do monocloreto de vinila (MVC) pode ser realizado em fase líquida ou gasosa.

- O processo em fase líquida não tem utilidade industrial devido ao fato de que o 1,2-dicloroetano (EDC) quando tratado com soda produz um sal clorado, o que representa uma retirada de átomos de cloro no processo, ou seja:



Além disso, o efluente do processo a ser descartado gera problemas ambientais, o que exigiria gastos no seu tratamento.

- O processo em fase gasosa pode ser catalítico ou não-catalítico sendo que o último é o mais utilizado industrialmente.

Dentre as empresas que utilizam o processo não catalítico estão a B.F.Goodrich, Dow Chemical, ICI, Hoechst, Monsanto, Ethyl, Mitsui Toatsu e Stauffer, enquanto que o processo catalítico é utilizado por um pequeno número de empresas (ex.: Wacker).

O processo que envolve apenas o craqueamento térmico (não catalítico) é mais econômico pela própria concepção de otimização do tempo de campanha da unidade quando não se usa catalisador.

Na figura 1 tem-se um fluxograma de processo simplificado para a produção do monocloreto de vinila através do craqueamento térmico do 1,2-dicloroetano.

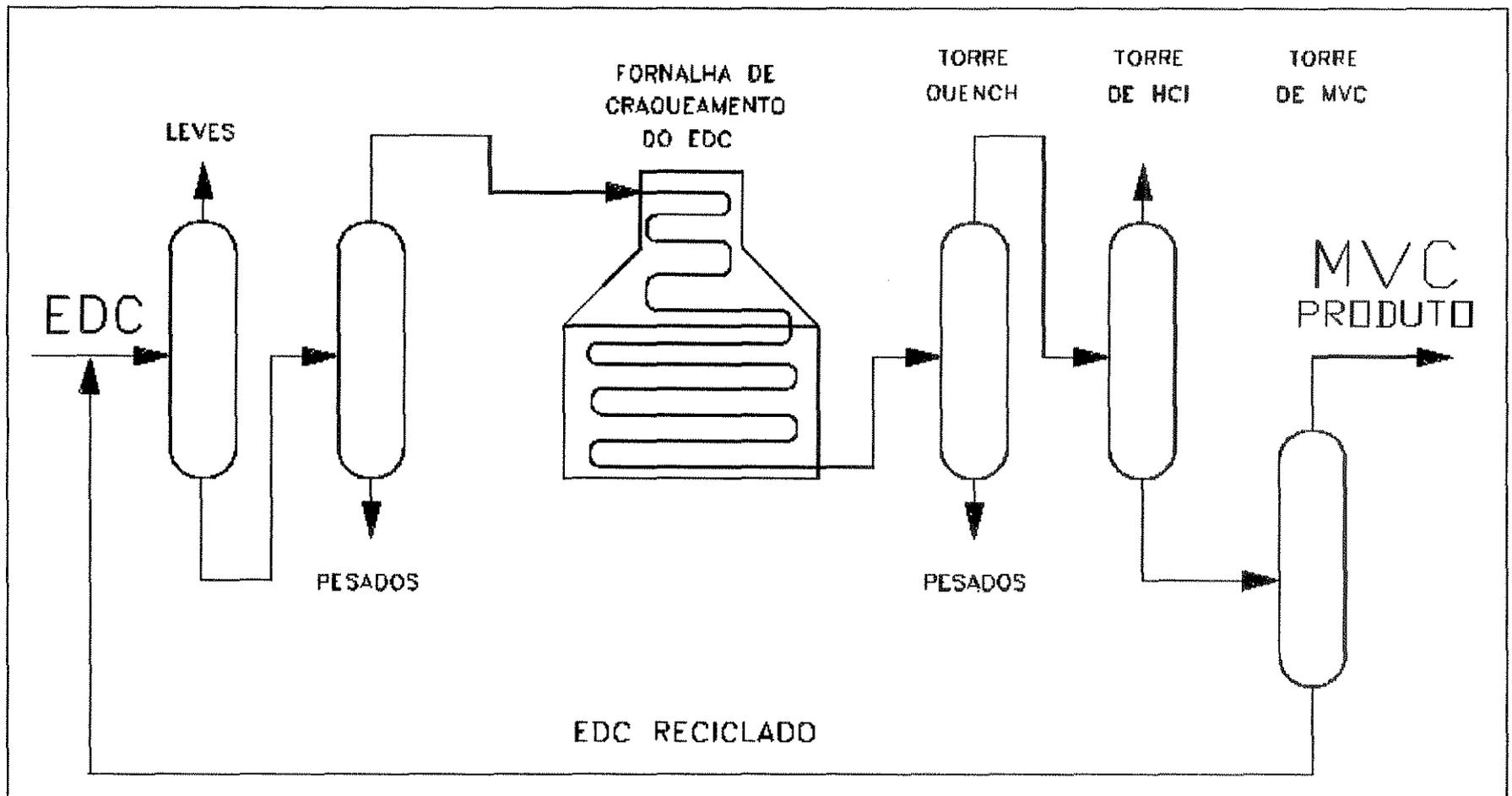
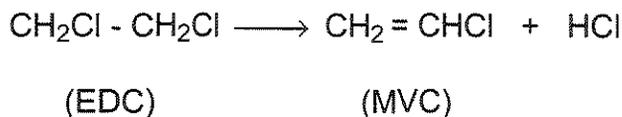
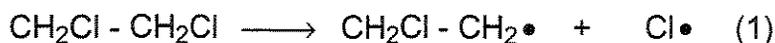


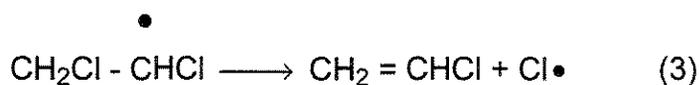
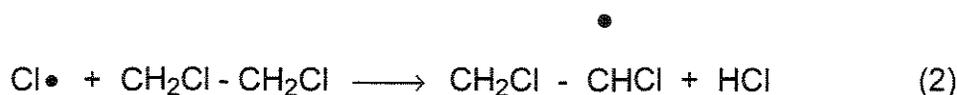
Figura 1 - FLUXOGRAMA SIMPLIFICADO DO PROCESSO DE PRODUÇÃO DO MVC ATRAVÉS DO CRAQUEAMENTO DO EDC

*Decomposição Térmica do 1,2-Dicloroetano***Reação principal:****Mecanismo:**

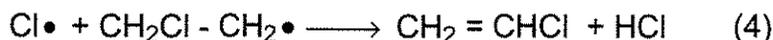
Iniciação:



Propagação:



Término:



O processo do craqueamento térmico ocorre em fornalhas onde o calor necessário à reação é fornecido pela queima de gás combustível nas paredes do forno. As fornalhas geralmente operam com uma pureza da carga na faixa de 96-99% em peso de 1,2-dicloroetano, temperaturas de 480-530°C e pressões de 6-35 atmosferas.

As condições operacionais são determinadas pela tecnologia do licenciador e pela configuração da planta. Por exemplo, pressões mais altas de craqueamento favorecem a uma redução no tamanho da fornalha devido a uma melhoria na troca térmica e melhora a separação entre o ácido clorídrico (HCl) e o monocloreto de vinila (MVC).

A faixa de conversão do 1,2-dicloroetano (EDC) é de 50-65% em peso, com o objetivo de otimizar o tempo de campanha da unidade e reduzir coeficientes técnicos. A presença de impurezas na alimentação pode ter uma forte influência na conversão do 1,2-dicloroetano (EDC), uma vez que as impurezas podem atuar como promotores ou inibidores de craqueamento. Além disso, estas substâncias podem influenciar no fenômeno de deposição de coque, que pode aumentar os custos de produção e limitar a continuidade operacional da unidade.

As fornalhas de craqueamento do 1,2-dicloroetano (EDC) consistem em uma série de tubos, a altas pressões, posicionadas horizontalmente em um forno com queima de gás combustível nas laterais do mesmo. Elas contêm uma seção de aquecimento por convecção dos gases de combustão, uma seção de aquecimento por radiação entre as paredes do forno, os gases de combustão e a serpentina de processo, e uma seção denominada 'shield' que separa a zona de convecção da zona de radiação. Na figura 2 temos um desenho do tipo de fornalha utilizada no presente trabalho.

O 1,2-dicloroetano líquido é alimentado na fornalha a cerca de 25-240°C e pressão de 20 a 35 kgf/cm<sup>2</sup> (manométrica). O 1,2-dicloroetano é aquecido e vaporizado na seção de convecção. O 1,2-dicloroetano gasoso passa então para a zona de radiação da fornalha sendo superaquecido até que a reação de craqueamento comece a ocorrer. A maior parte da reação ocorre na segunda seção da zona de radiação. O gás resultante deixa a fornalha a cerca de 450-650°C segundo citado por Albright (1967).

Vários trabalhos foram desenvolvidos sobre o estudo da reação de craqueamento térmico do 1,2-dicloroetano a pressões abaixo da atmosférica. A nível industrial, Krekeler (1954) observou que, operando as fornalhas de craqueamento à pressão atmosférica, ocorre uma intensa deposição de sólidos ou subprodutos de alto ponto de ebulição os quais "coqueiam"<sup>1</sup> os tubos da serpentina e limitam o processo à utilização de períodos de operação relativamente pequenos. Os depósitos de coque formados tinham que ser removidos da serpentina, o que reduzia o tempo de campanha da unidade.

---

<sup>1</sup> Coqueiam - recobrir as paredes da serpentina com subprodutos da reação de craqueamento que são denominados como coque, produtos carbonáceos ou compostos poliméricos saturados ou insaturados.

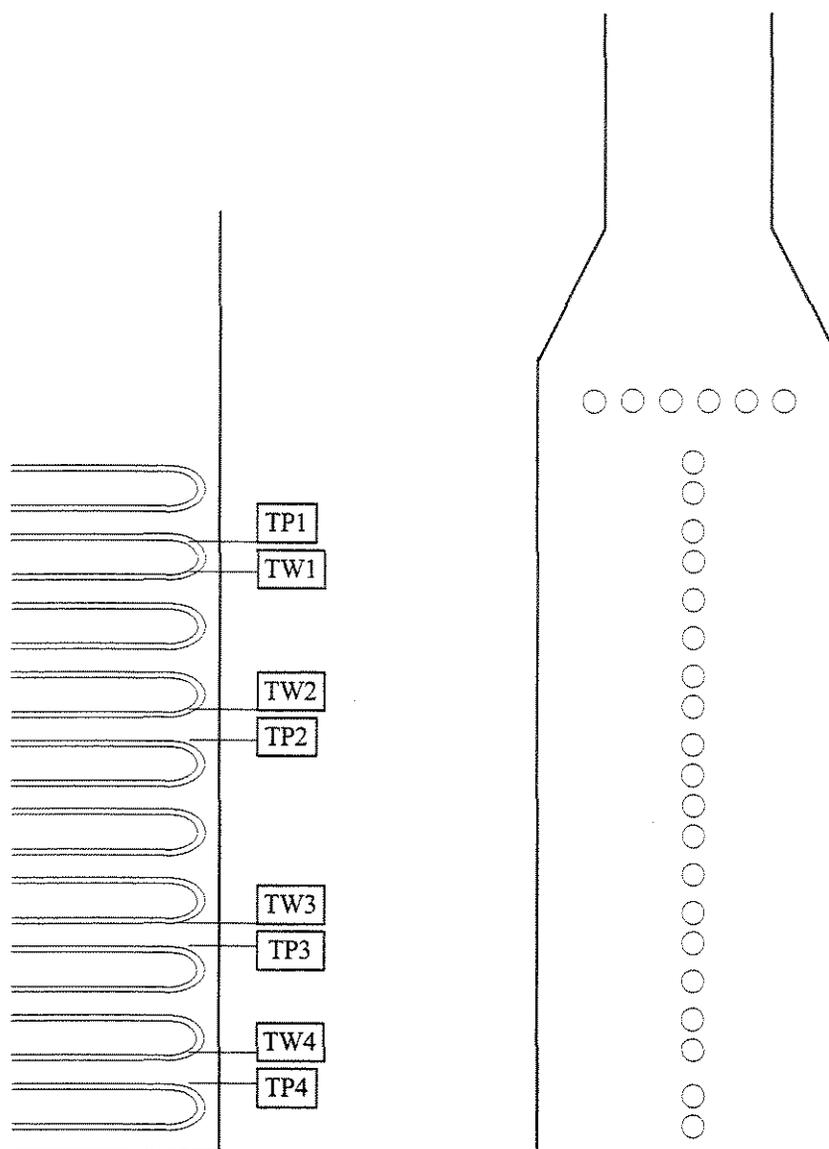


Figura 2 - Desenho de uma Fornalha

Krekeler (1955) propôs a utilização de altas pressões para a redução da formação de coque nas serpentinas de craqueamento. Ele descobriu que quando se opera à pressão manométrica de 15 atmosferas ocorre uma redução na formação de depósitos carbonáceos e que esta redução é quase completa a 20 atmosferas. Sugeriu-se então a utilização de pressões manométricas entre 20 e 35 atmosferas.

Segundo a Ullmann's Encyclopedia (1985), atualmente o monocloreto de vinila é produzido em pressões manométricas de até 25-30 atmosferas e temperaturas até 500-550 °C com o objetivo de melhorar a transferência de

calor, reduzir o tamanho de equipamentos e facilitar a separação do ácido clorídrico do monocloreto de vinila por destilação fracionada. O grau de conversão nos reatores de craqueamento é normalmente mantido em 50-60%, tempos de residência de 2-30 segundos, com seletividade a monocloreto de vinila de 96 a 99% em peso.

A utilização de maiores temperaturas de craqueamento fornece uma maior conversão do 1,2-dicloroetano, mas também resulta em menor seletividade a monocloreto de vinila, uma vez que alguns dos subprodutos gerados durante o craqueamento agem como inibidores na seqüência de radicais livres do mecanismo da reação. Assim sendo, o craqueamento nestas condições nos fornece aumentos na conversão do 1,2-dicloroetano progressivamente menores e problemas progressivamente maiores com o coqueamento da serpentina e purificação dos produtos obtidos.

No presente trabalho elaboramos um modelo dinâmico representativo do processo que ocorre em um tipo de serpentina de craqueamento.

## CAPÍTULO 1. REVISÃO BIBLIOGRÁFICA

### 1.1 - PROCESSO DE PRODUÇÃO DO MONOCLORETO DE VINILA

Krekeler (1955) patenteou um processo para produção do monocloreto de vinila pelo craqueamento térmico do 1,2-dicloroetano a altas pressões. A utilização de pressões de 20 a 35 atmosferas provocava uma redução na formação de depósitos carbonáceos e, portanto uma otimização no tempo de campanha da unidade industrial. A faixa de temperatura utilizada foi de 450 a 650°C. Neste processo, onde se utiliza alta pressão, ocorre uma diminuição do craqueamento de hidrocarbonetos clorados a subprodutos e carbono, em relação à quantidade formada quando se trabalha à pressão atmosférica. Krekeler explicou que isto ocorre porque o calor de reação é melhor distribuído em uma mistura gasosa de alta densidade do que em uma de baixa densidade. Conseqüentemente menos carbono é produzido a altas pressões do que a baixas. Além disso, a aplicação de altas pressões fornece uma importante vantagem técnica que é a facilidade de separação do ácido clorídrico por destilação fracionada. Foram utilizadas serpentinas de aço inox de seis milímetros de diâmetro interno para a realização dos experimentos.

A B. F. Goodrich (1963) patenteou um processo para produção do monocloreto de vinila utilizando maiores diâmetros de serpentina. Pensava-se que somente utilizando diâmetro interno de 6 milímetros conseguiria êxito no processo de produção do monocloreto de vinila. Esta dimensão, entretanto, mostrou-se inadequada para a produção do monocloreto de vinila em escala comercial. Achava-se que se utilizando serpentinas de maiores diâmetros, a transferência de calor da parede para o centro do tubo deveria ser de tal forma que uma maior temperatura seria necessária para promover a dehidrocloração do material localizado no centro do tubo. Uma maior temperatura poderia causar um superaquecimento do material que passa pelas paredes da serpentina resultando na decomposição a produtos indesejados. A B.F. Goodrich (1963) descobriu que serpentinas de diâmetro interno de 2,42 a 8,5 cm poderiam ser utilizadas sem as desvantagens previamente citadas. A temperatura de reação utilizada variava de 482 a 538°C.

A BASF (1974) patenteou um processo para preparação do monocloreto de vinila em presença de tetracloreto de carbono, como promotor de craqueamento. A concentração de tetracloreto de carbono utilizada foi de 0,01 a 0,3% em peso. A temperatura de craqueamento foi de 400 a 600°C, o tempo de residência de 0,1 a 20 segundos e a reação foi realizada em presença de gases inertes, tais como nitrogênio. Não foi citada a faixa de pressão estudada.

Rechmeier (1974) patenteou um processo para produção do monocloreto de vinila com redução da formação de coque e de produtos secundários tais como butadieno, cloreto de metila e monovinilacetileno. No estágio de vaporização 30 - 70% em peso do 1,2- dicloroetano era vaporizado em temperaturas de 210 - 250°C e sob pressões de 20-35 atmosferas. A fase vapor resultante era craqueada termicamente na faixa de temperatura de 450-650°C e pressão de 20-35 atmosferas, enquanto que a fase líquida era filtrada e reciclada.

Rae e Fry (1977) patentearam um processo para preparação do monocloreto de vinila em presença de 200-5000 ppm em peso de 1,1,2-tricloroetano para minimização da formação de depósitos carbonáceos e plugeamento dos tubos, onde ocorre a reação. O craqueamento do 1,2-dicloroetano foi realizado a pressões de 20-24 atmosferas e perfil de temperatura na fornalha entre 180 e 600°C.

## 1.2 - MODELOS MATEMÁTICOS

Higgins (1969) desenvolveu um modelo matemático do sistema de craqueamento térmico do 1,2-dicloroetano em fornalha industrial considerando escoamento empistonado e perfil de temperatura uniforme ao longo do raio da serpentina. Ele utilizou um esquema de reações contendo a reação principal de decomposição do 1,2-dicloroetano em monocloreto de vinila e duas reações secundárias. O método de Euler foi utilizado para realizar a integração do sistema de equações passo a passo. Os resultados foram coerentes com os dados da planta industrial.

Dente e Ranzi (1979) desenvolveram um estudo da importância e aplicabilidade da simulação utilizando reações via radicais livres no craqueamento

térmico de hidrocarbonetos a olefinas e aromáticos. Foi dado enfoque à modelagem específica de sistemas com cinética complexa (SPYRO) e à solução matemática das equações com rigidez (“stiffness”) de balanço material envolvidas. Os resultados dos modelos molecular e mecanístico<sup>1</sup> (SPYRO) da pirólise do etano e propano foram comparados e os resultados do modelo mecanístico foram considerados mais satisfatórios em relação aos dados industriais. Apesar disto, a resolução das equações utilizando o modelo mecanístico é muito mais complicada.

Sundaram e Froment (1979) incluíram a cinética de deposição de coque no modelo do craqueamento térmico do propano. O trabalho contém uma parte experimental e uma parte teórica de elaboração do modelo com deposição de coque. A deposição de coque altera tanto o diâmetro interno do tubo quanto a transferência de calor. Para levar isto em consideração são necessários os valores locais de deposição de coque, que são dependentes do tempo. Assim sendo, eles utilizaram as equações de taxa de deposição de coque, acopladas às equações de balanço de massa, energia e quantidade de movimento do sistema, com a equação de deposição de coque sendo considerada dependente do tempo. O diâmetro do tubo era atualizado periodicamente, devido à espessura de coque depositado e através da resolução do modelo passo a passo foi possível obter o perfil da espessura do coque ao longo do reator em qualquer instante  $t$ .

Em outro trabalho, Sundaram e Froment (1980) desenvolveram um modelo bidimensional para a simulação de reatores tubulares para o craqueamento térmico do etano. O artigo compara os resultados do modelo uni e bidimensional e conclui que o modelo bidimensional representou melhor os resultados industriais.

Heynderickx et al (1992) fizeram a modelagem simultânea do forno e da serpentina de craqueamento. Eles assumiram um modelo unidimensional de escoamento empistonado com o número de Reynolds na faixa de 250.000. Consideraram também que a turbulência elimina qualquer diferença de temperatura ao longo do raio do tubo.

---

<sup>1</sup> utilizando mecanismo da decomposição do 1,2-dicloroetano através de radicais livres.

Parameswaran et al (1988) modelaram o craqueamento da nafta em serpentinas de diâmetro constante e variável e compararam os resultados da simulação dos dois modelos. Para serpentinas de diâmetro constante, foi adotado o seguinte procedimento: Para um dado perfil de temperatura de parede, as equações diferenciais do balanço de massa, energia e momentum foram resolvidas utilizando o método de Runge-Kutta de 4ª ordem para obtenção dos perfis axiais de conversão, composição dos produtos, pressão no reator, temperatura interna de parede e temperatura do gás de processo. Uma vez que a pressão era especificada na saída do reator, as equações diferenciais foram resolvidas por tentativa, utilizando-se diferentes valores de pressão de entrada.

Já que a deposição de coque afeta o desempenho do reator, foi incorporado ao modelo a taxa de deposição de coque na serpentina. À medida que o coque deposita-se nas paredes da serpentina há uma redução na área transversal do escoamento (redução do diâmetro) e do coeficiente global de troca térmica. Assim sendo o diâmetro do tubo era atualizado periodicamente. Uma vez que o coque deposita-se em camadas de coque já existentes, a taxa de deposição de coque era avaliada na temperatura da superfície do coque. Os autores assumiram estado pseudo-estacionário, uma vez que a taxa de formação de coque era muito menor que a taxa das outras reações. Assim sendo, não utilizaram termos de acúmulo nas equações de balanço de massa, energia e momentum, e somente o diâmetro interno da serpentina com deposição de coque ( $dt_c$ ), era atualizado periodicamente. Para o perfil de temperatura de processo da mistura gasosa nas condições de tubo limpo, as equações do balanço de massa, energia e momentum no sistema foram resolvidas, juntamente com as equações dos parâmetros utilizados nas equações de balanço.

Parameswaran et al (1988) realizaram a simulação utilizando serpentina com diâmetro variável. Eles concluíram que a performance dos fornos de nafta melhorou pela utilização de diâmetro variável. Comparando-se com fornalhas de diâmetro de serpentina constante, concluíram que nas fornalhas com diâmetro de serpentina variável, há um aumento significativo do tempo de campanha da unidade. Além disso, essas fornalhas podem trabalhar com uma maior capacidade de produção. Com a variação do diâmetro da serpentina, os autores conseguiram

uma redução na queda de pressão e na temperatura máxima de parede da serpentina do forno, sem deposição de coque.

Froment (1992) estudou a cinética e o projeto de reatores para o craqueamento térmico no processo de produção de olefinas. Considerou a necessidade de utilização de modelos mecanísticos e simulou o processo de craqueamento térmico da nafta. Também modelou a distribuição de calor e temperatura no forno, usando a formulação de transferência de calor por radiação.

Sundaram et al (1981) obtiveram um modelo cinético para a deposição de coque no craqueamento térmico do etano, à partir de experimentos em laboratório. Utilizaram este modelo em conjunto com as equações de conservação para a simulação do craqueamento do etano a nível industrial. As taxas de deposição de coque são inicialmente altas, mas rapidamente decrescem para atingir um valor assintótico. Os resultados do modelo para tempo de operação, espessura da camada de coque e evolução das temperaturas de parede e de processo, estavam de acordo com os dados industriais.

Barendregt et al (1981) desenvolveram um programa com base mecanística para a otimização de fornalhas de craqueamento. Através deste programa (SPYRO) foi possível prever o comportamento dinâmico do sistema em estudo. Foi incorporado um modelo para a deposição do coque. Esse modelo pode ser aplicado a cargas de etano e nafta. Avaliou-se a sensibilidade do modelo a diferentes parâmetros, tais como, pressão, diâmetro, configuração da serpentina, diluição com vapor e outros.

Plehiens et al (1990) simularam o craqueamento térmico do etano levando em consideração a deposição de coque e um modelo para a transferência de calor por radiação no forno. A taxa de deposição de coque considerava a existência de mais de um precursor de coque de forma que a utilização conjunta das equações da conservação, deposição de coque e transferência de calor, forneceu um modelo que representou satisfatoriamente o processo industrial em estudo.

Ranzi et al (1980) estudaram o mecanismo de deposição sob diversos aspectos. Eles observaram que o coque algumas vezes é atribuído a um processo de transferência de massa, com resistência limitante de filme. Isto implicava em

taxas de deposição de coque proporcionais ao coeficiente de transferência de massa e esta hipótese era confirmada pelo decréscimo das taxas de deposição com o aumento da temperatura. Esta teoria era, entretanto incoerente com o fato de que a taxa de formação de precursores de coque era muito maior que a taxa de deposição de coque, o que levou os autores a acreditarem que o fenômeno de deposição de coque não era puramente físico.

Albright (1967) fez uma revisão bibliográfica sobre a deposição de coque no craqueamento de hidrocarbonetos. No craqueamento a maioria das reações ocorre na fase gasosa e via radicais livres. A formação de material carbonáceo é um processo secundário de grande importância devido à redução da eficiência energética do processo. A formação de material carbonáceo é um processo complexo que envolve diferentes tipos de resíduos e carbonos. O coque pode conter substâncias aromáticas de alto ponto de ebulição e carbonos ordenados ou desordenados originados na fase gasosa ou na superfície. A formação de carbono na superfície pode ser promovida pela natureza do material na superfície (carbono catalítico).

Albright (1967) esclarece ainda que o carbono pode apresentar um grande número de diferentes estruturas desde a amorfa até a altamente cristalina. O carbono formado na fase gasosa é constituído de partículas esféricas formadas por pequenos cristalitos, os quais tendem a se ligar em estruturas de cadeia. Este carbono tem uma estrutura aparentemente diferente da molécula que o originou. O carbono gerado na superfície metálica tem uma estrutura diferente, cristalitos maiores, menores espaçamentos entre camadas e densidades maiores. Eles são obtidos como filmes com depósitos cristalíticos bem ordenados em substratos sólidos. Na pirólise heterogênea, o processo de superfície parece ser o fator controlador na determinação da cristalinidade do carbono. Em alguns substratos metálicos, depósitos altamente cristalinos são formados à baixas temperaturas. Existem superfícies metálicas catalíticas e inertes. Níquel, cobalto e ferro são substratos altamente efetivos para a formação catalítica de carbono.

Ranzi et al (1990,1993) modelaram e simularam a pirólise e cloração de hidrocarbonetos de cadeia pequena. O modelo cinético adotado envolvia mais de trezentas (300) reações elementares e 46 espécies, moleculares e radicais. O

modelo cinético foi aplicado ao craqueamento do 1,2-dicloroetano, sendo que se incluiu neste modelo a taxa de deposição de coque, uma vez que a deposição de coque na serpentina determina o tempo de operação das fornalhas. O modelo proposto é composto pelas equações de balanço de massa e energia na serpentina, balanço de energia na fornalha e balanço de quantidade de movimento no sistema. A integração das equações foi realizada através do uso do pacote LSODE que é bastante apropriado para resolução de sistema de equações diferenciais com dificuldade de convergência.

Na incorporação do modelo de deposição de coque, Ranzi et al (1990, 1993) fizeram as seguintes considerações:

O modelo de deposição de coque foi concebido pela analogia entre o fenômeno de deposição de coque na pirólise do 1,2-dicloroetano e o processo de craqueamento de olefinas. O modelo assume que a deposição inicial de coque ocorre devido ao processo de polimerização catalítica nos sítios ativos da superfície metálica. Posteriormente a estrutura do polímero sofre dehidrocloração e dehidrogenação, resultando na reticulação e “cross-linking” do mesmo. Isto aumenta a resistência à difusão de monômeros olefinicos e evita que eles atinjam os sítios ativos na superfície metálica. Simultaneamente o polímero se transforma em coque por desidrogenação adicional. Considerou as taxas de deposição de coque através de adições de radicais ( $r_{rad}$ ) e adições moleculares ( $r_{mol}$ ).

Ranzi et al (1985) fizeram um estudo sobre a morfologia e influência do coque no craqueamento térmico do etano e da nafta. Observou-se que o coque depositado no craqueamento da nafta é mole e amorfo, enquanto que o coque resultante do craqueamento do etano é duro e mais cristalino. Os autores observaram que a condutividade térmica do coque depositado pelo craqueamento de naftas e gasóleos fica na faixa de 3-5 kcal / m h °C, enquanto que o coque depositado pelo craqueamento do etano e gases liquefeitos de petróleo têm condutividade térmica na faixa de 10-12 kcal / m h °C. Essa diferença na condutividade térmica é refletida no perfil de temperatura de parede da serpentina. As temperaturas de parede de fornalhas de nafta crescem rapidamente devido à crescente resistência da camada de coque que está sendo depositada. Já nas fornalhas de etano, os aumentos na temperatura de parede são menores, para

uma mesma espessura de coque, devido à menor resistência oferecida pelo coque mais cristalino.

### 1.3 - REDES NEURAIS ARTIFICIAIS

“Redes Neurais Artificiais é o nome dado a um conjunto de métodos matemáticos e algoritmos computacionais especialmente projetados para simular o processamento de informações e aquisição de conhecimento do cérebro humano”.

As redes neurais são um conjunto de neurônios dispostos em camadas segundo uma topologia, sendo que os neurônios da camada de entrada são submetidos aos inputs ou sinais de entrada, os sinais são processados segundo métodos matemáticos e a camada de saída gera um ou mais sinais de saída que são comparados com os valores desejados até que haja um erro mínimo entre estes conjuntos de dados.

Na figura 4 temos a representação esquemática de uma rede neural com três camadas, uma camada de entrada, uma camada intermediária e uma camada de saída.

Usualmente as camadas são classificadas em três grupos:

- Camada de Entrada: onde os padrões são apresentados à rede;
- Camadas Intermediárias ou Escondidas: onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características;
- Camada de Saída: onde o resultado final é concluído e apresentado.

Na figura 4 é apresentada a estrutura de uma rede MLP (Multi Layer Perceptron) com 3(três) camadas. A camada de entrada recebe os sinais, processa-os e os envia para a camada intermediária onde os sinais são processados segundo a função de ativação escolhida. Finalmente os resultados obtidos nesta camada são enviados para a camada de saída, onde são novamente processados segundo a função de ativação escolhida gerando os resultados preditos pela rede.

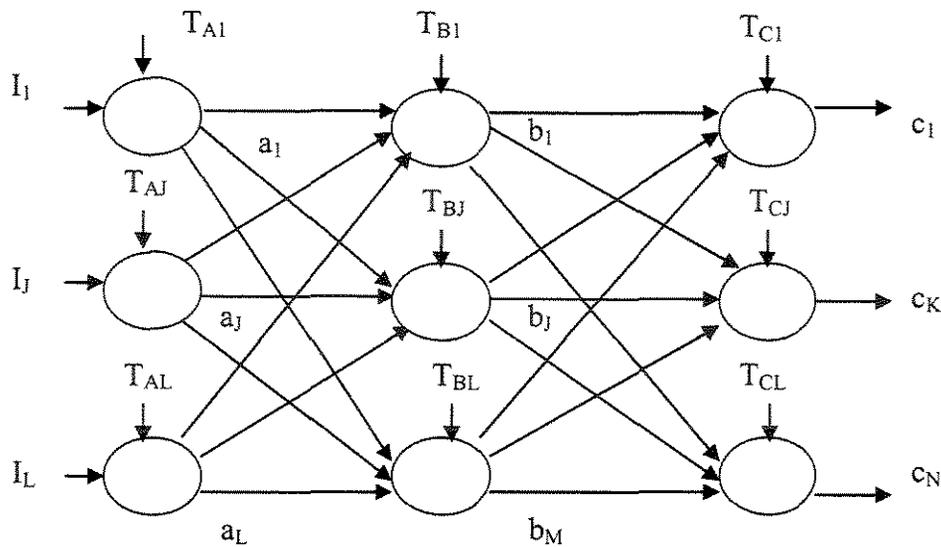


Figura 4 – Organização de uma Rede Neural em Camadas.

camada A (entrada)	camada B (interna)	camada C (Saída)
$I = O$ vetor de entradas	$M =$ nós (neurônios)	$N =$ nós (neurônios)
$L =$ nós (neurônios)	$b_j (j = 1 \text{ a } m) =$ saída,	$c_k (k = 1 \text{ a } n) =$ saída,
$a_i (i = 1 \text{ a } L) =$ saída,	$m =$ pode ser ajustado	$n =$ um para cada saída
um neurônio para cada entrada.	$V_{ij} =$ peso de conexão	$w_{ij} =$ peso de conexão

De uma maneira geral a concepção de uma rede neural artificial envolve as seguintes etapas:

1. Definição das variáveis mais importantes do processo em estudo;
2. Definição da quantidade de entradas e saídas da rede;
3. Definição do número de camadas da rede, do número de neurônios em cada camada, das funções de ativação de cada camada, etc...;
4. Escolha do conjunto de dados de treinamento, teste e validação;
5. Normalização do conjunto de dados de entrada da rede;
6. Treinamento da Rede;
7. Teste da Rede;
8. Validação da rede construída;

Uma rede neural é especificada, principalmente pela sua topologia, pelas características dos nós e pelas regras de treinamento.

O treinamento de uma rede neural consiste basicamente nas seguintes etapas.

1. Recebimento dos sinais de entrada nos neurônios da camada de entrada;
2. Ponderação dos sinais de entrada com os pesos da primeira camada;
3. Aplicação da função de ativação aos resultados calculados anteriormente;
4. Cálculo do erro entre o valor de saída e o valor desejado;
5. Ajuste dos pesos segundo o método de ajuste utilizado;

A seguir, serão citados alguns processos treinamento ou aprendizado da rede.

### **Processos de Aprendizado**

A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

Denomina-se algoritmo de aprendizado a um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais, estes algoritmos diferem entre si principalmente pelo modo como os pesos são modificados.

Outro fator importante é a maneira pela qual uma rede neural se relaciona com o ambiente. Nesse contexto existem os seguintes paradigmas de aprendizado:

- 1) Aprendizado Supervisionado, quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada;

- 2) Aprendizado Não Supervisionado (auto-organização), quando não existe uma agente externo indicando a resposta desejada para os padrões de entrada;
- 3) Reforço, quando um crítico externo avalia a resposta fornecida pela rede.

Denomina-se ciclo uma apresentação de todos os  $N$  pares (entrada e saída) do conjunto de treinamento no processo de aprendizado. A correção dos pesos num ciclo pode ser executada de dois modos:

1) Modo Padrão: A correção dos pesos acontece a cada apresentação à rede de um exemplo do conjunto de treinamento. Cada correção de pesos baseia-se somente no erro do exemplo apresentado naquela iteração. Assim, em cada ciclo ocorrem  $N$  correções.

2) Modo Batch: Apenas uma correção é feita por ciclo. Todos os exemplos do conjunto de treinamento são apresentados à rede, seu erro médio é calculado e a partir deste erro fazem-se as correções dos pesos.

Minsky e Papert (1969) analisaram matematicamente o “Perceptron” e demonstraram que redes de uma camada não são capazes de solucionar problemas que não sejam linearmente separáveis. Como não acreditavam na possibilidade de se construir um método de treinamento para redes com mais de uma camada, eles concluíram que as redes neurais seriam sempre suscetíveis a essa limitação. Contudo, o desenvolvimento do algoritmo de treinamento “backpropagation”, por Rumelhart, Hinton e Williams em 1986, precedido por propostas semelhantes ocorridas nos anos 70 e 80, mostrou que é possível treinar eficientemente redes com camadas intermediárias, resultando no modelo de Redes Neurais Artificiais mais utilizado atualmente, as redes Perceptron Multi-Camadas (MLP), treinadas com o algoritmo por retroalimentação.

Nessas redes, cada camada tem uma função específica. A camada de saída recebe os estímulos da camada intermediária e constrói o padrão que será a resposta. As camadas intermediárias funcionam como extratoras de características, seus pesos são uma codificação de características apresentadas

nos padrões de entrada e permitem que a rede crie sua própria representação, mais rica e complexa do problema.

Se existirem as conexões certas entre as unidades de entrada e um conjunto suficientemente grande de unidades intermediárias, pode-se sempre encontrar a representação que irá produzir o mapeamento correto da entrada para a saída através das unidades intermediárias. Como provou Cybenko (1988), a partir de extensões do Teorema de Kolmogoroff, são necessárias no máximo duas camadas intermediárias, com um número suficiente de unidades por camada, para se produzir quaisquer mapeamentos. Também foi provado que apenas uma camada intermediária é suficiente para aproximar qualquer função contínua.

### **Treinamento pelo Método da Retroalimentação ou “Backpropagation”**

Durante o treinamento com o algoritmo por retroalimentação, a rede opera em uma seqüência de dois passos. Primeiro, um padrão é apresentado à camada de entrada da rede. A atividade resultante flui através da rede, camada por camada, até que a resposta seja produzida pela camada de saída. No segundo passo, a saída obtida é comparada à saída desejada para esse padrão particular. Se esta não estiver correta, o erro é calculado. O erro é propagado a partir da camada de saída até a camada de entrada, e os pesos das conexões das unidades das camadas internas vão sendo modificados conforme o erro é retropropagado.

As redes que utilizam o algoritmo por retroalimentação trabalham com uma variação da regra delta, apropriada para redes multicamadas: a regra delta generalizada. A regra delta padrão essencialmente implementa um gradiente descendente no quadrado da soma do erro para funções de ativação lineares. Redes sem camadas intermediárias podem resolver problemas onde a superfície de erro tem a forma de um parabolóide com apenas um mínimo. Entretanto, a superfície do erro pode não ser tão simples, e suas derivadas mais difíceis de serem calculadas. Nestes casos devem ser utilizadas redes com camadas intermediárias. Ainda assim, as redes ficam sujeitas aos problemas de mínimos locais.

A metodologia de treinamento de uma rede neural artificial com 3(três) camadas utilizando o algoritmo “backpropagation” é apresentada no anexo 1.

O treinamento das redes MLP(Multi-Layer Perceptron) com “backpropagation” pode demandar muitos passos no conjunto de treinamento, resultando um tempo de treinamento consideravelmente longo. Se for encontrado um mínimo local, o erro para o conjunto de treinamento pára de diminuir e estaciona em um valor maior que o aceitável. Uma maneira de aumentar a taxa de aprendizado sem levar à oscilação é modificar a regra delta generalizada para incluir o termo momentum, uma constante que determina o efeito das mudanças passadas dos pesos na direção atual do movimento no espaço de pesos.

Desta forma, o termo momentum leva em consideração o efeito de mudanças anteriores de pesos na direção do movimento atual no espaço de pesos. O termo momentum torna-se útil em espaços de erro que contenham longas gargantas, com curvas acentuadas ou vales com descidas suaves.

Dentre os métodos de correção dos pesos, os mais utilizados são o de retropropagação de erros (backpropagation) e o de Levenberg-Marquadt.

#### **1.4 -REDES NEURAIIS EM PROCESSOS QUÍMICOS**

Baghat (1990) modelou através das redes neurais artificiais dois reatores CSTRs (Constant Stirred Tank Reactor) para prever a concentração transiente no fluxo de saída a partir de um pulso na concentração de entrada. A rede foi treinada para, a partir do perfil de concentração transiente de saída, prever as correspondentes vazões de entrada. Os dados utilizados para treinamento do sistema em estudo foram obtidos a partir dos resultados obtidos de um modelo fenomenológico do sistema.

Baghat (1990) desenvolveu também um modelo baseado em redes neurais para identificar o grau de mistura de um sistema reacional, ou seja, o número de reatores CSTR necessários para atingir o grau de mistura desejado. A rede foi treinada com os resultados de um modelo fenomenológico dos reatores em série.

Citra (1992,1993) aplicou as redes neurais artificiais no estudo da cinética de um reator de hidrogenação. Utilizou o planejamento experimental para determinar a cinética e a transferência de massa que ocorre durante a reação de hidrogenação. Através destes experimentos ele pode descrever bem algumas regiões do CSTR. As outras regiões foram modeladas de forma satisfatória por uma rede neural artificial. Foram avaliados os efeitos da massa do catalisador na cinética da reação a uma taxa constante. Os modelos de redes neurais foram considerados satisfatórios em todas as regiões do CSTR.

Citra (1993) também aplicou um modelo de redes neurais no processo de fabricação de um catalisador em solução com o objetivo de encontrar as condições ótimas de fabricação do catalisador mantendo a qualidade da resina e minimizando a quantidade de precipitado formado durante o processo. O modelo de redes neurais foi comparado a um modelo estatístico desenvolvido e ele concluiu que ambos os modelos foram satisfatórios.

Yanamoto (1993) e Nascimento e Yanamoto (1993) utilizaram as redes neurais artificiais na simulação estacionária do processo de síntese de amônia em reator catalítico de leito fixo. Os dados de treinamento e teste da rede foram obtidos através de um modelo fenomenológico do processo. Observaram que um aumento do número de neurônios na camada escondida pode conduzir a um sobreajuste (overfitting), representando bem os dados de treinamento, porém com uma previsão não muito boa.

As redes neurais artificiais tem sido amplamente empregadas em processos de polimerização envolvendo radicais livres, onde se têm dificuldades para modelar o processo fenomenologicamente, não só pela reação química, como também pela transferência de calor, massa e estimativa de parâmetros. Nascimento e Chan (1994) e Chan e Nascimento (1994) realizaram a modelagem da reação de polimerização de olefinas através das redes neurais artificiais. Os dados de treinamento e testes foram obtidos experimentalmente de um reator tubular industrial. Eles concluíram que a modelagem através das redes neurais artificiais para a predição do perfil de temperatura do reator e das propriedades finais do polímero foram melhores que os resultados obtidos através do modelo fenomenológico.

Lieb et al (1996) utilizaram as redes neurais artificiais para a predição da oxidação parcial do propileno a acroleína e vários subprodutos, em reator de leito fluidizado. As variáveis de entrada da rede foram selecionadas como sendo as variáveis mais significativas para o processo. As previsões do modelo foram satisfatórias e os dados foram obtidos de um modelo teórico.

Stevanovic (1994) aplicou as redes neurais artificiais para a análise e otimização da produção em processos de separação a partir de dados industriais. Foram utilizadas redes feedforward com algoritmo de treinamento por retroalimentação. Foi observado que um aumento do número de neurônios na camada escondida melhoravam a capacidade de predição da rede. As redes neurais também foram utilizadas como ferramenta para estimativa de parâmetros do processo.

Marlet et al (1994) utilizaram as redes neurais como alternativa na análise estatística de dados históricos das plantas químicas. Eles compararam a modelagem estatística utilizada na análise de dados de processo de polimerização de acrinonitrila com a modelagem via redes neurais, sendo que os resultados obtidos com as redes foram melhores.

A modelagem fenomenológica dinâmica de reatores de leito fixo geralmente é simplificada e mesmo a utilização de redes neurais artificiais teria uma complexidade muito grande e gastaria muito tempo para o seu treinamento. Devido a isto, Zhou et al (1996) desenvolveram um modelo híbrido para a oxidação do benzeno catalisado pelo óxido de vanádio realizada em reator de leito fixo. Foi utilizada a análise de componentes principais (PCA) para redução da dimensionalidade do sistema, selecionando as variáveis mais significativas no processo em estudo pelo a representação das mesmas em um novo sistema de coordenadas com máxima variância.

Redes Neurais e Lógica Fuzzy são métodos que tem sido utilizados em aspectos de propagação de falhas e distúrbios no comportamento global da planta. Wang et al (1996) desenvolveram um algoritmo, considerando as vantagens e desvantagens de cada método, dando suporte às decisões operacionais tais como o diagnóstico de falhas. As principais limitações são as

dificuldades de extrapolação, o tempo de processamento e inexistência de uma ferramenta para seleção de variáveis e aprendizado na inversão da simulação de modelos com diferentes dados de entrada.

Dirion et al (1996), devido a dificuldades do controle convencional de reatores batelada e semibatelada, propuseram desenvolver um controlador baseado em redes neurais (controlador neural), utilizando-o para o controle de temperatura em tempo real de um reator piloto semibatelada. Utilizou-se para isto redes neurais com mapeamento dinâmico, onde as saídas da rede eram retroalimentadas às entradas, dando o comportamento dinâmico do processo. Os autores concluíram que os resultados foram satisfatórios e que as redes neurais foram capazes de agir como controladores de processos.

Molga et al (1999) mostraram que as redes neurais artificiais poderiam ser utilizadas em sistemas difíceis de serem modelados fenomenologicamente como a reação de hidrólise do anidrido propiônico catalisado com ácido sulfúrico. Foi proposta então a aplicação das redes neurais para prever todas as propriedades desconhecidas do sistema citado para a sua cinética de reação. Foi proposto um modelo híbrido contendo equações de balanço de massa e energia, enquanto que a expressão desconhecida da cinética foi descrita por uma rede neural artificial. Os perfis de temperatura e concentração preditos pelo modelo híbrido foram comparados com os perfis obtidos experimentalmente com uma boa precisão nos resultados.

Qi et al (1999) desenvolveram um modelo híbrido para um reator de leito fixo de oxidação de benzeno a anidrido maleico. Foi utilizada uma rede feedforward com três camadas operando em regime estacionário e dinâmico. Foi utilizado um modelo unidimensional para o reator com o gradiente radial de temperatura sendo fornecido ao modelo fenomenológico pela rede neural, podendo-se considerar o modelo híbrido como bidimensional e mais adequado para a descrição do sistema. Concluíram que o modelo híbrido proposto tanto para o estado estacionário quanto para o dinâmico, pode prever com precisão o comportamento do reator.

Gao et al (2000) propuseram uma estratégia para controlar processos não lineares desconhecidos através da combinação de um modelo de redes neurais com uma simples lei de controle não linear de múltiplas entradas e única saída (MISO). Foi utilizada uma rede neural recorrente diagonal (RNRD) com algoritmo de treinamento por retroalimentação, sendo que a RNRD é uma rede neural recorrente com mapeamento dinâmico. Neste tipo de rede as saídas dos neurônios da camada externas são retroalimentadas à entrada dos neurônios da camada de entrada. Utilizou-se a função de ativação sigmóide na camada escondida e a função linear na camada de saída.

### **1.5 - ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)**

Em ambientes industriais, os dados de processos fornecem a base para o monitoramento da qualidade, avaliação e controle dos mesmos. O armazenamento de grande quantidade de dados tem se tornado uma operação de rotina como uma consequência dos avanços na tecnologia dos computadores. Uma etapa importante é entender o comportamento do processo e extrair os aspectos mais significantes que podem ser obtidos dos dados, isto é avaliar a real significância dos dados que estão sendo acompanhados no processo. A técnica mais comumente utilizada é a análise de componentes principais (PCA) que geralmente é utilizada em processos lineares ou linearizáveis.

A Análise de Componentes Principais (PCA) tem sido aplicada amplamente para o monitoramento do desempenho de plantas em alguns processos industriais. A Análise dos Componentes Principais (PCA) é uma técnica linear e é então não estritamente aplicável a problemas de processos que exibem comportamento não linear significativa.

A Análise de Componentes Principais é uma técnica matemática baseada em transformações lineares de um conjunto de dados multivariados. Esta técnica permite encontrar as direções segundo as quais a maioria das informações nos dados pode ser comprimida e expressa com um certo grau de significância.

As variáveis menos significativas são não correlacionadas e são ordenadas por grau descendente de importância, isto é, as primeiras têm as

maiores variâncias. As transformações lineares deixam o conteúdo original da informação intacto, ao lado que o ordenamento para relevância estatística oferece um método conveniente para a filtragem e redução da dimensionalidade de novos dados, com a mínima perda de informação dos dados originais.

Uma maneira usual de encontrar os Componentes Principais de um conjunto amostral é pelo cálculo dos autovetores da matriz de correlação. Utiliza-se o novo conjunto de eixos para descrever os dados e a projeção dos dados nos autovetores são as Componentes Principais. Os correspondentes autovalores são as variâncias do conjunto de dados de média zero, de tal forma que o conjunto de dados é posicionado nos eixos de acordo com sua importância estatística decrescente.

As Componentes Principais também tem sido utilizadas na detecção e identificação de erros grosseiros em processo químicos. Tong (1996) apresentou uma abordagem para construir testes seqüenciais de Componentes Principais para detecção e identificação de erros grosseiros na reconciliação de dados pela combinação da Análise por Componentes Principais com a Análise Seqüencial.

## **1.6 - PRÉ- PROCESSAMENTO DAS REDES NEURAS ARTIFICIAIS**

O treinamento, teste e validação das Redes Neurais Artificiais podem ser conseguidos pela análise das componentes principais, dentro de um grau de significância das variáveis que estão sendo utilizadas. A utilização da Análise das Componentes Principais reduz a complexidade do sistema e faz com que o número de cálculos matemáticos necessários ao treinamento, teste e validação das Redes Neurais Artificiais sejam reduzidos.

Rossela e Mário (2001) utilizaram a Análise de Componentes Principais no pré-processamento dos dados de reconhecimento e classificação de sinais através de Redes Neurais Artificiais. Observaram que a utilização da Análise de Componentes Principais acelerou o desempenho das Redes.

As técnicas de monitoramento estatístico de processos são geralmente baseadas em técnicas estatísticas lineares, entretanto, em alguns processos onde existem correlações intrinsecamente não lineares entre as variáveis de processo

as técnicas estatísticas lineares não são muito eficientes na redução da dimensão dos dados de processo. Quando a análise de componentes principais é utilizada nestes processos, um grande número de componentes principal é necessário para explicar suficientemente a variância dos dados. Em alguns tipos de processos, a análise por componentes principais não linear é mais adequada para descrever as correlações entre as variáveis do processo em estudo.

Zhang et al (1997) utilizaram a análise não linear de Componentes Principais no monitoramento de um reator de polimerização e observaram que esta técnica pode fornecer uma compressão mais efetiva dos dados que a técnica de linear de componentes principais. Necessitou-se de menor número de componentes principais na análise não linear para explicar a mesma percentagem de variância quando em comparação à análise linear de componentes principais. Concluiu-se que a análise não linear de componentes principais foi mais adequada para o monitoramento dos dados de polimerização.

Lee et al (2001) utilizaram técnicas estatísticas multivariadas para aumentar em aproximadamente 10% a produção do monocloreto de vinila, sem alterar a qualidade do produto. O objetivo principal do trabalho era descobrir a relação entre a produtividade e a pureza do monocloreto de vinila. Levou-se em consideração algumas variáveis operacionais que influenciam significativamente na qualidade do produto e tempo de operação da unidade, tais como: pureza da carga de 1,2-dicloroetano, perfil de temperatura na fornalha, etc.. As técnicas estatísticas multivariadas foram utilizadas para esclarecer a relação entre variáveis de processo e variáveis de qualidade. Pela otimização dos modelos de correlação analisados, bons resultados foram atingidos. A produção de monocloreto de vinila foi aumentada em mais de 10% mantendo-se os mesmos limites de pureza.

## 1.7 - CONCLUSÕES

Através da revisão bibliográfica foi possível observar que na literatura são descritas várias iniciativas para simular fenomenologicamente o craqueamento térmico do 1,2-dicloroetano, desde métodos simples a métodos mais robustos que levam em consideração a taxa mecanística da reação de decomposição térmica.

Verificou-se, entretanto que o estudo e a modelagem dinâmica do craqueamento térmico do 1,2-dicloroetano, serve para esclarecer melhor as reações do processo em relação as suas variáveis operacionais e visualizar como as mesmas se comportam com a evolução do tempo, estabelecendo parâmetros de comparação com os valores teoricamente esperados.

Observou-se também que a simulação não convencional de processos químicos através das redes neurais artificiais (RNA) foi encontrada em várias referências na literatura, sendo que apenas uma referência foi encontrada na simulação estacionária do craqueamento térmico do 1,2-dicloroetano.

Foi encontrada apenas uma referência de aplicação da Análise dos Componentes Principais (PCA) ao craqueamento térmico do 1,2-dicloroetano e várias referências na utilização desta metodologia na aceleração do treinamento das Redes Neurais Artificiais (RNA's).

## CAPÍTULO 2. MODELAGEM DINÂMICA DO CRAQUEAMENTO TÉRMICO

### MODELAGEM DO REATOR

#### 2.1- Dedução da Equação da Conservação da Massa para os Componentes

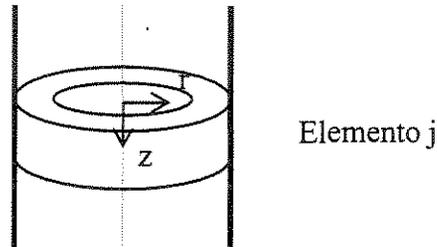


Figura 6 – Elemento Diferencial de Volume de um Reator Tubular

Dentro do elemento,  $j$  pode ser produzido por reação química a uma taxa de  $R_j^*$ .

As contribuições do Balanço de Massa são:

- Taxa de Acúmulo de Massa da espécie  $j$ :

$$(2\pi r \Delta r \Delta z) \cdot \frac{\partial \rho_j}{\partial t}$$

- Taxa de Massa da espécie  $j$  que entra em  $Z$  (por transporte convectivo e difusivo):

$$(2\pi r \Delta r) \rho_j v_{j,z} \Big|_z$$

- Taxa de Massa da espécie  $j$  que sai em  $Z+\Delta Z$  (por transporte convectivo e difusivo):

$$(2\pi r \Delta r) \rho_j v_{j,z} \Big|_{z+\Delta z}$$

- Taxa de Massa da espécie  $j$  que entra em  $r$  (por transporte convectivo e difusivo):

$$(2\pi r \Delta z) \rho_j v_{j,r} \Big|_r$$

- Taxa de Massa da espécie  $j$  que sai em  $r+\Delta r$  (por transporte convectivo e difusivo):

$$(2\pi r \Delta r) \rho_j v_{j,r} \Big|_{r+\Delta r}$$

- Taxa de Produção da espécie j por reação química:

$$(2\pi r \Delta r \Delta z) R_j^*$$

Sendo  $R_j^*$  baseado no volume do fluido

Substituindo na equação de balanço:

Taxa de Acúmulo de Massa	=	Taxa de Massa que entra por transporte convectivo e difusivo	-	Taxa de Massa que sai por transporte convectivo e difusivo	+	Taxa de Massa Gerada por Reação Química
-----------------------------	---	--	---	--	---	---

$$(2\pi r \Delta r \Delta z) \cdot \frac{\partial \rho_j}{\partial t} = (2\pi r \Delta r) \cdot \rho_j v_{j,z} - (2\pi r \Delta r) \cdot \rho_j v_{j,z} \Big|_{z-\Delta z} + (2\pi r \Delta r) \cdot \rho_j v_{j,r} \Big|_{r-\Delta r} - (2\pi r \Delta z) \cdot \rho_j v_{j,r} \Big|_{r+\Delta r} + (2\pi r \Delta r \Delta z) \cdot R_j^*$$

Dividindo a equação por  $(2\pi r \Delta r \Delta z)$  e fazendo o limite quando  $\Delta z \rightarrow 0$  e  $\Delta r \rightarrow 0$ :

$$r \cdot \frac{\partial \rho_j}{\partial t} = - \frac{\partial}{\partial z} (r \cdot \rho_j v_{j,z}) - \frac{\partial}{\partial r} (r \cdot \rho_j v_{j,r}) + R_j^* r$$

$$r \cdot \frac{\partial \rho_j}{\partial t} = - r \cdot \frac{\partial}{\partial z} (\rho_j v_{j,z}) - \frac{\partial}{\partial r} (r \cdot \rho_j v_{j,r}) + R_j^* r$$

Dividindo por r

$$\frac{\partial \rho_j}{\partial t} = - \frac{\partial}{\partial z} (\rho_j v_{j,z}) - \frac{1}{r} \cdot \frac{\partial}{\partial r} (r \cdot \rho_j v_{j,r}) + R_j^*$$

Como:

$$C_j = \frac{\rho_j}{M_j} \quad \text{e} \quad M_j = \frac{M}{\text{mol}}$$

Onde:

$C_j$  - Concentração molar de j (kgmol de j / m<sup>3</sup>)

$M_j$  - Peso molecular de j (kg/kgmol);

Dividindo por  $M_j$  tem-se,

$$\frac{\partial C_j}{\partial t} = -\frac{\partial}{\partial z}(C_j v_{j,z}) - \frac{1}{r} \cdot \frac{\partial}{\partial r}(r \cdot C_j v_{j,r}) + R_j'$$

Sendo  $J_j$  o fluxo molar difusivo [mol/(cm<sup>3</sup>s)] dado por:  $J_j = C_j \cdot (v_j - v)$

Onde:

$(v_j - v)$  - velocidade de difusão de J com relação a v

$J_j$  - Fluxo molar relativo à velocidade média v (LT<sup>-1</sup>)

$$J_j = C_j \cdot (v_j - v)$$

$$J_j = C_j \cdot v_j - C_j \cdot v$$

$$C_j \cdot v_j = J_j + C_j \cdot v$$

Resulta em:

$$\frac{\partial C_j}{\partial t} = -\frac{\partial}{\partial z}(J_{j,z} + C_j \cdot v_z) - \frac{1}{r} \cdot \frac{\partial}{\partial r} r(J_{j,r} + C_j \cdot v_r) + R_j'$$

O coeficiente de dispersão é definido como:

$$J_j = -D_j \cdot \nabla C_j, \text{ daí:}$$

$$J_{j,z} = -D_{j,z} \cdot \frac{\partial C_j}{\partial z} \quad \text{e} \quad J_{j,r} = -D_{j,r} \cdot \frac{\partial C_j}{\partial r}$$

Reorganizando a equação para em seguida introduzir coeficiente de dispersão:

$$\frac{\partial C_j}{\partial t} = -\frac{\partial J_{j,z}}{\partial z} - \frac{\partial}{\partial z}(C_j \cdot v_z) - \frac{1}{r} \left[ \frac{\partial}{\partial r}(r J_{j,r}) + \frac{\partial}{\partial r}(r \cdot C_j \cdot v_r) \right] + R_j'$$

Considerando  $v_r = 0$  e os componentes de dispersão constantes

$$\frac{\partial C_j}{\partial t} = -\frac{\partial J_{j,z}}{\partial z} - \frac{\partial}{\partial z}(C_j \cdot v_z) - \frac{1}{r} \frac{\partial}{\partial r}(r J_{j,r}) + R_j'$$

$$\frac{\partial C_j}{\partial t} = D_{j,z} \frac{\partial^2 C_j}{\partial z^2} - \frac{\partial}{\partial z}(C_j \cdot v_z) + \frac{1}{r} D_{j,r} \frac{\partial}{\partial r} \left( r \frac{\partial C_j}{\partial r} \right) + R_j'$$

$$\frac{\partial C_j}{\partial t} = D_{j,z} \frac{\partial^2 C_j}{\partial z^2} - \frac{\partial}{\partial z}(C_j \cdot v_z) + \frac{1}{r} D_{j,r} \left[ r \frac{\partial^2 C_j}{\partial r^2} + \frac{\partial C_j}{\partial r} \right] + R_j'$$

$$\frac{\partial C_j}{\partial t} = D_{jz} \frac{\partial^2 C_j}{\partial z^2} - \frac{\partial}{\partial z} (C_j \cdot v_z) + D_{jr} \left[ \frac{\partial^2 C_j}{\partial r^2} + \frac{1}{r} \cdot \frac{\partial C_j}{\partial r} \right] + R_j'$$

Considerando fluxo empistonado, não se tem dispersão axial de massa, nem dispersão radial de massa:

$$\frac{\partial C_j}{\partial t} = - \frac{\partial}{\partial z} (C_j \cdot v_z) + R_j'$$

$$\frac{\partial C_j}{\partial t} + v_z \frac{\partial C_j}{\partial z} + C_j \frac{\partial v_z}{\partial z} = R_j'$$

Mas,

$$C_j = \frac{\rho_j}{M_j} \quad \text{e} \quad v_z = \frac{G}{\rho}$$

$$\frac{1}{M_j} \frac{\partial \rho_j}{\partial t} + \frac{G}{\rho M_j} \frac{\partial \rho_j}{\partial z} - G \frac{\partial \rho}{M_j \rho^2} \frac{\partial \rho}{\partial z} = R_j'$$

Substituindo  $\rho_j = w_j \cdot \rho$  na equação acima ( $w_j$  = fração mássica)

$$\frac{\rho}{M_j} \frac{\partial w_j}{\partial t} + \frac{w_j}{M_j} \frac{\partial \rho}{\partial t} - \frac{G}{M_j} \frac{\partial w_j}{\partial z} = R_j'$$

A massa específica é função apenas da temperatura

$$\frac{\rho}{M_j} \frac{\partial w_j}{\partial t} + \frac{G}{M_j} \frac{\partial w_j}{\partial z} = R_j'$$

Sendo  $R_j' = \sum_{i=1}^{NR} \nu_{ji} r_i$

Onde:

$\nu_{j,i}$  = coeficiente estequiométrico do componente J na reação i;

$r_i$  = taxa da reação i;

NR = número de reações químicas no sistema de reações.

$$\left( \frac{\rho}{M_j} \right) \left( \frac{\partial w_j}{\partial t} \right) + \left( \frac{G}{M_j} \right) \left( \frac{\partial w_j}{\partial z} \right) = \sum_{i=1}^{NR} \nu_{ji} r_i$$

### 2.1.1 - Balanço de Massa dos Componentes

O balanço de massa para cada componente que participa das reações químicas é descrito pelas seguintes equações:



#### 1. 1,2-Dicloroetano (EDC)

$$\rho \frac{\partial w_1}{\partial t} + G \frac{\partial w_1}{\partial z} = (-R(1))M_1$$

#### 2. Monocloreto de Vinila (MVC)

$$\rho \frac{\partial w_2}{\partial t} + G \frac{\partial w_2}{\partial z} = (R(1) - R(2))M_2$$

#### 3. Ácido Clorídrico (HCl)

$$\rho \frac{\partial w_3}{\partial t} + G \frac{\partial w_3}{\partial z} = (R(1) + R(2))M_3$$

#### 4. Acetileno (C<sub>2</sub>H<sub>2</sub>)

$$\rho \frac{\partial w_4}{\partial t} + G \frac{\partial w_4}{\partial z} = (R(2))M_4$$

### 2.2 - Dedução da Equação de Conservação da Energia

A Equação da Conservação de Energia pode ser obtida de maneira análoga à equação de conservação de massa, ou seja, através de um balanço de energia.

- Taxa de Acumulação de Energia Interna e Cinética:

$$(2\pi r \Delta r \Delta z) \cdot \frac{\partial (\rho C_p T)}{\partial t}$$

- Taxa de Energia Interna e Cinética que entra em Z por convecção:

$$(2\pi r \Delta r) \cdot \rho \cdot v \cdot C_p \cdot T|_z$$

- Taxa de Energia Interna e Cinética que sai em  $Z+\Delta Z$  por convecção:

$$(2.\pi.r.\Delta r). \rho.v.C_p.T|_{z+\Delta z}$$

- Taxa Líquida de Energia adicionada por condução nas direções r e Z:

$$2.\pi.r.\Delta z. q_r|_r - 2.\pi.r.\Delta z. q_r|_{r+\Delta r}$$

$$2.\pi.r.\Delta r. q_z|_z - 2.\pi.r.\Delta r. q_z|_{z+\Delta z}$$

- Taxa Líquida de Trabalho cedida pelo sistema às vizinhanças:

$$0 \text{ (zero)}$$

- Taxa Líquida de Calor produzido por Reação Química:

$$2.\pi.r.\Delta r.\Delta z.(-\Delta H).R^*$$

Substituindo na equação de balanço e fazendo as seguintes considerações:

$$q = -K \cdot \frac{dT}{dX} \quad (\text{Lei de Fourier})$$

$$\sum_j (-\Delta H_j).R_j^* = (-\Delta H).R^*$$

$$v_r = 0 \quad \text{e} \quad v = v_z$$

$$\frac{\partial(\rho C_p T)}{\partial t} = K_z \cdot \frac{\partial^2 T}{\partial z^2} - \frac{\partial(v_z \rho C_p T)}{\partial z} + K_r \left( \frac{\partial^2 T}{\partial r^2} + \frac{1}{r} \cdot \frac{\partial T}{\partial r} \right) + \sum_{i=1}^{NR} (-\Delta H_i).R_i^*$$

onde:

$\rho$  = massa específica da mistura;

$C_p$  = capacidade calorífica da mistura;

$T$  = temperatura absoluta da mistura;

$t$  = tempo;

$K_z$  = condutividade térmica da mistura em Z;

$K_r$  = condutividade térmica da mistura em r;

$r$  = raio da serpentina;

$\Delta H_i$  = calor da reação i;

$R_i$  = taxa da reação i.

Considerando Fluxo empistonado, não teremos transferência de calor por dispersão axial de calor, nem dispersão radial de calor:

$$K_z \frac{\partial^2 T}{\partial z^2} = 0 \quad \text{e} \quad K_r \frac{\partial^2 T}{\partial r^2} = 0$$

$$\frac{\partial(\rho C_p T)}{\partial t} + \frac{\partial(v_z \rho C_p T)}{\partial z} = K_r \left( \frac{1}{r} \cdot \frac{\partial T}{\partial r} \right) + \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i^*$$

Considerando  $C_p$  e  $\rho$  dependentes apenas da temperatura e fazendo  $\rho \cdot v_z = G$ , temos:

$$\rho C_p \frac{\partial T}{\partial t} + C_p \frac{\partial(G \cdot T)}{\partial z} = K_r \left( \frac{1}{r} \cdot \frac{\partial T}{\partial r} \right) + \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i^*$$

Aplicando a seguinte condição de contorno na parede:

$$K_r \cdot \frac{\partial T}{\partial r} \Big|_{r=R} = h_w (T_w - T)$$

Onde:

$T_w$  = temperatura da parede interna do tubo da serpentina

$h_w$  = coeficiente de transferência de calor entre a parede externa da serpentina e o fluido de processo

Colocando em função do coeficiente global de troca térmica, U:

$$\frac{1}{RU} = \frac{1}{r_o h_{wo}} + \frac{1}{r_i h_{wi}} + \frac{1}{(r_o - r_i) K_t} + \frac{\ln(r_o/r_i)}{K_{coque}}$$

Onde:

U = coeficiente global de troca térmica;

$r_o$  = raio externo do tubo;

$h_{w0}$  = coeficiente externo de transferência de calor entre o gás de combustão e a parede externa do tubo;

$r_t$  = raio interno do tubo;

$K_t$  = condutividade térmica do metal do tubo da serpentina;

$r_i$  = raio interno do tubo considerando a espessura da camada de deposição de coque;

$K_{\text{coque}}$  = condutividade térmica do coque depositado sobre a parede interna da serpentina;

$h_{w1}$  = coeficiente interno de transferência de calor.

Na figura seguinte temos o corte da seção transversal da serpentina.

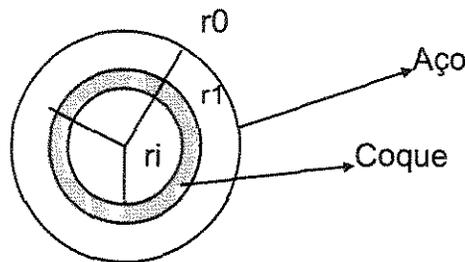


Figura 7 – Seção Transversal da Serpentina

Supondo  $R = r_0 = r_i$ ;  $h_{w0} = h_{w1} = h_w$  e resistência térmica do aço desprezíveis.

Resistência térmica do coque não desprezível

$$\frac{1}{RU} = \frac{1}{r_0 h_{w0}} + \frac{1}{r_t K_t} + \frac{1}{r_i K_{\text{coque}}} + \frac{1}{r_i h_{w1}}$$

$$\frac{1}{RU} = \frac{1}{R h_w} + \frac{1}{R K_{\text{coque}}} + \frac{1}{R h_w}$$

$$\frac{1}{U} = \frac{2}{h_w} + \frac{1}{K_{\text{coque}}}$$

$$\frac{2}{h_w} = \frac{1}{U} - \frac{1}{K_{\text{coque}}}$$

$$\frac{2}{h_w} = \frac{K_{\text{coque}} - U}{U \cdot K_{\text{coque}}}$$

$$h_w = \frac{2U K_{coque}}{K_{coque} - U}$$

Logo:

$$K_r \left. \frac{\partial T}{\partial r} \right|_{r=R} = \frac{2UK_{coque}}{K_{coque} - U} (T_w - T)$$

Substituindo na equação de balanço de energia

$$\rho C_p \frac{\partial T}{\partial t} + C_p \frac{\partial(G.T)}{\partial z} = \frac{2U}{R} \cdot \frac{K_{coque}}{K_{coque} - U} (T_w - T) + \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i^*$$

Usando o diâmetro do tubo e considerando escoamento uniforme, ou seja, a velocidade não varia com o comprimento, temos:

$$\rho C_p \frac{\partial T}{\partial t} + C_p G \frac{\partial T}{\partial z} = \frac{4U}{D} \cdot \frac{K_{coque}}{K_{coque} - U} (T_w - T) + \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i^*$$

### 2.3 - Dedução da Equação da Quantidade de Movimento

- Taxa de Acúmulo de Quantidade de Movimento:

$$(2 \cdot r \cdot \pi \cdot \Delta r \cdot \Delta z) \cdot \frac{\partial(\rho \cdot v)}{\partial t}$$

- Taxa Líquida de Momentum por transporte convectivo:

$$(2 \cdot \pi \cdot r \cdot \Delta r) \cdot \rho \cdot v \cdot v|_z - (2 \cdot \pi \cdot r \cdot \Delta r) \cdot \rho \cdot v \cdot v|_{z+\Delta z}$$

- Taxa Líquida de Momentum por transporte molecular:

$$2 \cdot \pi \cdot r \cdot \Delta z \cdot \tau_{rz}|_r - 2 \cdot \pi \cdot r \cdot \Delta z \cdot \tau_{rz}|_{r+\Delta r}$$

$$2 \cdot \pi \cdot r \cdot \Delta r \cdot \tau_{zz}|_z - 2 \cdot \pi \cdot r \cdot \Delta r \cdot \tau_{zz}|_{z+\Delta z}$$

- Taxa Líquida de Momentum devido à pressão do fluido e à força gravitacional por unidade de massa:

$$2\pi r \Delta r (P_z - P_{z+\Delta z}) + \rho \cdot g_z \cdot 2\pi r \Delta r \Delta z$$

A pressão em um fluido em movimento é definida pela equação de estado  $P = P(\rho, T)$  e é uma quantidade escalar.

Substituindo as expressões no balanço, dividindo a equação por  $2 \cdot \pi \cdot r \cdot \Delta r \cdot \Delta z$  e tomando o limite quando  $\Delta z$  e  $\Delta r$  tendem a zero:

$$\frac{\partial(\rho \cdot v)}{\partial t} + \frac{\partial}{\partial z}(\rho \cdot v \cdot v) = - \left[ \frac{1}{r} \cdot \frac{\partial}{\partial r}(r \cdot \tau_{rz}) + \frac{\partial \tau_{zz}}{\partial z} \right] - \frac{\partial P}{\partial z} + \rho \cdot g_z$$

Tem-se que:

$$\tau_{rz} = - \mu \left[ \frac{\partial v_z}{\partial r} + \frac{\partial v_r}{\partial z} \right]$$

$$\tau_{zz} = - \mu \left[ 2 \frac{\partial v_z}{\partial z} - \frac{2}{3} (\nabla_o \cdot v) \right]$$

Considerando que  $v_r = 0$ ,  $\nabla \cdot v = \frac{\partial v_z}{\partial z}$

Então:

$$\tau_{rz} = - \mu \cdot \frac{\partial v_z}{\partial r}$$

$$\tau_{zz} = - \frac{4}{3} \mu \frac{\partial v_z}{\partial z}$$

A massa específica é função apenas da temperatura

$$\rho \frac{\partial(v_z)}{\partial t} + \rho \frac{\partial}{\partial z}(v_z \cdot v_z) = - \left[ \frac{1}{r} \cdot \frac{\partial}{\partial r}(- \mu \cdot r \cdot \frac{\partial v_z}{\partial r}) + \frac{\partial}{\partial z} \left( - \frac{4}{3} \mu \frac{\partial v_z}{\partial z} \right) \right] - \frac{\partial P}{\partial z} + \rho \cdot g_z$$

$$\rho \frac{\partial v_z}{\partial t} + \rho \frac{\partial}{\partial z}(v_z \cdot v_z) = - \frac{4}{3} \mu \frac{\partial^2 v_z}{\partial z^2} - \frac{\partial P}{\partial z} + \rho \cdot g_z$$

$$\rho \frac{\partial v_z}{\partial t} + 2 \rho v_z \frac{\partial v_z}{\partial z} = \frac{4}{3} \mu \frac{\partial^2 v_z}{\partial z^2} - \frac{\partial P}{\partial z} + \rho \cdot g_z$$

Considerar-se-á a viscosidade ( $\mu$ ) desprezível. O escoamento é uniforme, logo  $v_z$  não varia com o comprimento Z:

$$\rho \left( \frac{\partial v_z}{\partial t} \right) + \left( \frac{\partial P}{\partial z} \right) = \rho \cdot g_z$$

## 2.4 - ADIMENSIONALIZAÇÃO DAS EQUAÇÕES

### 2.4.1 - Balanço de Massa

$$\rho \left( \frac{\partial W_j}{\partial t} \right) + G \left( \frac{\partial W_j}{\partial z} \right) = M_j \cdot \left( \sum_{i=1}^{NR} \nu_{j,i} r_i \right)$$

Onde:

$$z = \frac{z}{z_f}; z_f = \text{comprimento do reator (m)}$$

$$W_j = \frac{w_j}{w_{10}}; w_{10} = \text{composição de EDC na entrada da serpentina.}$$

$$\tau = \frac{T}{T_0}; T_0 = \text{temperatura na entrada da serpentina.}$$

$$t' = \frac{t \cdot G}{R \rho_0};$$

G = fluxo mássico da mistura gasosa.

R = raio da serpentina

$\rho_0$  = massa específica da mistura gasosa na entrada da serpentina.

$$\rho \frac{\partial(w_{10} \cdot w_j)}{\partial \left( \frac{t' R \rho_0}{G} \right)} + G \frac{\partial(w_{10} \cdot w_j)}{\partial(z \cdot z_f)} = M_j \left( \sum \nu_{j,i} r_i \right)$$

Multiplicando a equação por  $\left( \frac{R \cdot \rho_0}{G \cdot \rho w_{10}} \right)$  fica:

$$\frac{\partial W_j}{\partial t'} = - \left( \frac{R \cdot \rho_0}{\rho z_f} \right) \left( \frac{\partial W_j}{\partial z} \right) + \left( \frac{R \cdot \rho_0 \cdot M_j}{G \cdot \rho w_{10}} \right) \left( \sum \nu_{j,i} r_i \right)$$

### 2.4.2 - Balanço e Energia

$$\rho \cdot C_p \frac{\partial T}{\partial t} + G \cdot C_p \frac{\partial T}{\partial z} = \frac{4U}{D} (T_w - T) + \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i$$

Adimensionalizando fica:

$$\rho \cdot C_p \frac{\partial(\tau \cdot T_0)}{\partial\left(\frac{R\rho_0 t'}{G}\right)} + G \cdot C_p \frac{\partial(\tau \cdot T_0)}{\partial(Z_f \cdot z)} = \frac{4U}{D}(T_w - \tau \cdot T_0) + \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i$$

$$\frac{G \cdot \rho \cdot C_p \cdot T_0}{R \cdot \rho_0} \frac{\partial \tau}{\partial t'} + \frac{G \cdot C_p \cdot T_0}{Z_f} \frac{\partial \tau}{\partial z} = \frac{4U}{D}(T_w - \tau \cdot T_0) + \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i$$

Multiplicando por  $\frac{R \cdot \rho_0}{\rho G C_p T_0}$  fica:

$$\frac{\partial \tau}{\partial t'} + \frac{R \cdot \rho_0}{Z_f \cdot \rho} \frac{\partial \tau}{\partial z} = \frac{4U}{D} \frac{R \cdot \rho_0}{\rho G C_p T_0} (T_w - \tau \cdot T_0) + \frac{R \cdot \rho_0}{\rho G C_p T_0} \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i$$

$$\frac{\partial \tau}{\partial t'} = -\frac{R \cdot \rho_0}{Z_f \cdot \rho} \frac{\partial \tau}{\partial z} + \frac{2U \cdot \rho_0}{\rho G C_p T_0} (T_w - \tau \cdot T_0) + \frac{R \cdot \rho_0}{\rho G C_p T_0} \sum_{i=1}^{NR} (-\Delta H_i) \cdot R_i$$

### 2.4.3 - Balanço da Quantidade de Movimento

$$\frac{\partial P}{\partial z} = \rho \cdot g_z$$

Fazendo:

$$v_z' = \frac{v_z \cdot \rho}{G}$$

$$\text{e } P = \frac{P}{P_0} ; \quad P_0 = \text{Pressão da corrente na alimentação};$$

$$\frac{\partial\left(\frac{G \cdot v_z'}{\rho}\right)}{\partial\left(\frac{R\rho_0 t'}{G}\right)} + \frac{\partial(P \cdot P_0)}{\partial(z \cdot Z_f)} = \rho g_z$$

$$\frac{G^2}{R \cdot \rho_0} \frac{\partial v_z'}{\partial t'} + \frac{P_0}{Z_f} \frac{\partial P}{\partial z} = \rho g_z$$

Multiplicando a equação acima por  $\frac{R \cdot \rho_0}{G^2}$ , fica:

$$\frac{\partial v_z}{\partial t} + \frac{R \cdot \rho_o \cdot P_o}{G^2} \frac{\partial P}{Z_f \partial z} = \frac{R \cdot \rho_o \cdot \rho g_z}{G^2}$$

Sabe-se que:

$$\frac{\partial P}{\partial z} = \frac{\frac{\partial \left( \frac{1}{M_m} \right)}{\partial z} + \frac{1}{M_m} \left[ \frac{1}{T} \frac{\partial T}{\partial z} + F_r \right]}{\frac{1}{M_m P} - \frac{P}{\alpha \cdot G^2 \cdot R T}}$$

Adimensionalizando a equação acima, fica:

$$\frac{\partial P}{\partial z} = \frac{\frac{1}{Z_f} \frac{\partial \left( \frac{1}{M_m'} \right)}{\partial z} + \frac{1}{M_m'} \left[ \frac{1}{\tau Z_f} \frac{\partial \tau}{\partial z} + F_r \right]}{\frac{1}{M_m' P P_o} - \frac{P P_o}{\tau \cdot G^2 \cdot R T_o \alpha}}$$

Com estas equações a próxima etapa do trabalho é aplicar o método das diferenças finitas ou colocação ortogonal às equações de balanço de massa, energia e quantidade de movimento e após isto aplicar o método de Runge-Kutta para resolver o sistema de equações obtido.

## CAPÍTULO 3. MÉTODOS MATEMÁTICOS

### 3.1 - INTRODUÇÃO

Serão apresentados a seguir alguns métodos utilizados na solução de equações diferenciais (ordinárias e parciais), os quais servirão para resolver o conjunto de equações fenomenológicas que descrevem o processo de craqueamento térmico do 1,2-dicloroetano.

### 3.2 - DIFERENÇAS FINITAS

A aproximação por diferenças finitas é um dos métodos mais usados para integração numérica de equações diferenciais parciais (Carnaham, 1969; Constantinides, 1987). As derivadas parciais são substituídas por quocientes de diferenças das variáveis independentes e o resultado é usado para aproximação das derivadas. Em geral, o domínio de uma equação diferencial parcial é igual ao número de variáveis independentes. Vamos considerar, como simplificação, o caso especial de apenas duas variáveis independentes:

$$y = y(x, z)$$

Cujo domínio é bidimensional e pode ser representado numa superfície plana. Os valores de  $y$  em dois pontos  $(x+h', z+k')$  estão relacionados através da expansão em Taylor:

$$y(x+h', z+k') = y(x, z) + \left( h' \frac{\partial}{\partial x} + k' \frac{\partial}{\partial z} \right) y(x, z) + \frac{1}{2!} \left( h' \frac{\partial}{\partial x} + k' \frac{\partial}{\partial z} \right)^2 y(x, z) + \dots + \frac{1}{(n-1)!} \left( h' \frac{\partial}{\partial x} + k' \frac{\partial}{\partial z} \right)^{n-1} y(x, z) + R_n \quad (3.1)$$

Onde o último termo é dado por:

$$R_n = \frac{1}{(n)!} \left( h' \frac{\partial}{\partial x} + k' \frac{\partial}{\partial z} \right)^n y(x + \varepsilon h', z + \varepsilon k') + R_n \quad 0 < \varepsilon < 1 \quad (3.2)$$

Isto é,

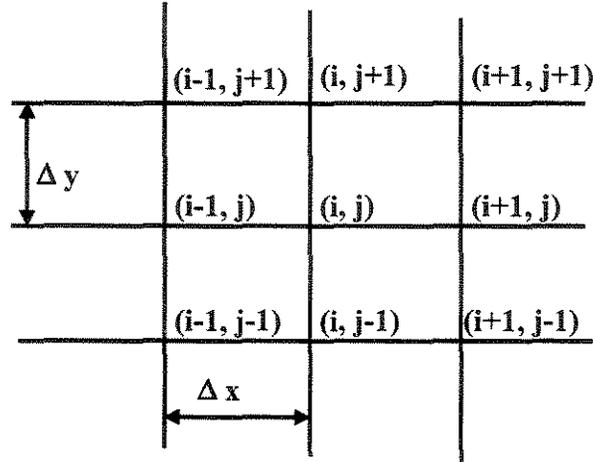
$$R_n = 0 \left[ (|h'| + |k'|)^n \right] \quad (3.3)$$

Pela equação (3.3) foi observado que existe uma constante positiva M tal que:

$$|R_n| \leq M \left[ (|h'| + |k'|)^n \right],$$

Com  $h'$  e  $k'$  tendendo para zero.

Considerando-se a grade de pontos a seguir:



Expandindo em série de Taylor  $y_{i-1, j}$  e  $y_{i+1, j}$  em torno do valor central  $y_{i, j}$ , obtém-se:

$$y_{i-1, j} = y_{i, j} - \Delta x y_x + \frac{(\Delta x)^2}{2!} y_{xx} - \frac{(\Delta x)^3}{3!} y_{xxx} + \frac{(\Delta x)^4}{4!} y_{xxxx}$$

$$y_{i+1, j} = y_{i, j} + \Delta x y_x + \frac{(\Delta x)^2}{2!} y_{xx} + \frac{(\Delta x)^3}{3!} y_{xxx} + \frac{(\Delta x)^4}{4!} y_{xxxx}$$

onde,  $y_x \equiv \frac{\partial y}{\partial x}$ ,  $y_{xx} \equiv \frac{\partial^2 y}{\partial x^2}$ , etc., e todas as derivadas são avaliadas no ponto  $(i, j)$ .

A partir destas duas expressões, isoladas e adicionadas ou subtraídas uma da outra, foram obtidas as seguintes expressões de diferenças finitas para a primeira e segunda derivadas no ponto  $(i, j)$ :

$$\frac{\partial y}{\partial x} = \frac{y_{i+1, j} - y_{i-1, j}}{\Delta x} + \theta(x) \tag{3.4}$$

$$\frac{\partial y}{\partial x} = \frac{y_{i, j} - y_{i-1, j}}{\Delta x} + \theta(x) \tag{3.5}$$

$$\frac{\partial y}{\partial x} = \frac{y_{i+1,j} - y_{i-1,j}}{2 \cdot \Delta x} + \theta [(\Delta x)^2] \quad (3.6)$$

$$\frac{\partial^2 y}{\partial x^2} = \frac{y_{i-1,j} - 2y_{i,j} + y_{i+1,j}}{2 \cdot \Delta x} + \theta [(\Delta x)^2] \quad (3.7)$$

As expressões (3.4), (3.5) e (3.6) são conhecidas como fórmula de diferenças finitas para frente, para trás e central respectivamente.

Formas semelhantes existem para  $\frac{\partial y}{\partial z}$  e  $\frac{\partial^2 y}{\partial z^2}$

Também se pode demonstrar que

$$\frac{\partial^2 y}{\partial x \partial z} = \frac{y_{i-1,j+1} - 2y_{i,j+1} + y_{i+1,j+1}}{4 \cdot \Delta x \Delta y} + \theta [(\Delta x + \Delta z)^2] \quad (3.8)$$

Por conveniência, o operador de diferença central pode ser usado ocasionalmente. Ele é definido por:

$$\delta_x y_{i,j} = \frac{y_{i+1/2,j} - y_{i-1/2,j}}{\Delta x} \quad (3.9)$$

portanto

$$\delta_x^2 y_{i,j} = \frac{y_{i-1/2,j} - 2y_{i,j} + y_{i+1/2,j}}{(\Delta x)^2} \quad (3.10)$$

Procedimento análogo pode ser desenvolvido para a primeira e segundas derivadas no ponto  $(i+1/2, j)$  obtendo-se as seguintes expressões:

$$\frac{\partial y}{\partial x} = \theta \delta_x y_{i,j} + (1-\theta) \delta_x y_{i+1,j} \quad (3.11)$$

$$\frac{\partial^2 y}{\partial x^2} = \theta \delta_x^2 y_{i,j} + (1-\theta) \delta_x^2 y_{i+1,j} \quad (3.12)$$

onde  $0 \leq \theta \leq 1$ .

Quando  $\theta$  é igual a 1/2 tem-se o método de Crank & Nicholson e pode-se escrever:

$$\frac{\partial y}{\partial x} = \frac{1}{2} \left[ \frac{y_{i,j+1} - y_{i,j-1}}{2 \cdot \Delta x} + \frac{y_{i+1,j+1} - y_{i+1,j-1}}{2 \cdot \Delta x} \right] \quad (3.13)$$

$$\frac{\partial^2 y}{\partial x^2} = \frac{1}{2} \left[ \frac{y_{i,j+1} - 2y_{i,j} + y_{i,j-1}}{(\Delta x)^2} + \frac{y_{i+1,j+1} - 2y_{i+1,j} + y_{i+1,j-1}}{(\Delta x)^2} \right] \quad (3.14)$$

### 3.3 - COLOCAÇÃO ORTOGONAL

O método da colocação ortogonal é um método de resíduo ponderado que é utilizado na solução aproximada de equações diferenciais e apresenta vantagens em relação a outros métodos de colocação (Finlayson, 1980; Villadson, 1978):

- Os pontos de colocação são selecionados automaticamente, evitando assim uma escolha arbitrária (possivelmente inadequada) pelo usuário, e
- O erro decresce de forma mais rápida com o aumento número de termos.

De um modo geral sua aplicação consiste na aproximação da solução por funções previamente escolhidas cujos coeficientes são ajustáveis. Estes coeficientes são determinados de modo a satisfazer às condições de contorno e a algum critério de minimização de erro entre a solução exata e a aproximada.

Três aspectos básicos caracterizam o método de colocação ortogonal:

a) A solução aproximada é expressa por polinômios ortogonais, do tipo

$$y = \sum_{j=1}^m a_j P_{j-1}(x) \quad (3.15)$$

b) Os pontos de colocação são as raízes do polinômio ortogonal de maior grau;

c) As variáveis dependentes são calculadas nos pontos de colocação ao invés dos coeficientes de expansão.

Considerando a equação diferencial:

$$f(y^n, y^{n-1}, \dots, y', y, x) = 0 \quad (3.16)$$

A solução aproximada desta equação será na forma de um polinômio

$$y_m(x) = \sum_{j=0}^m a_j x^j = a_0 + a_1 x + a_2 x^2 + \dots \quad (3.17)$$

Cujos coeficientes  $a_i$  deverão ser calculados de forma a satisfazer as condições de contorno da equação (3.16).

Substituindo a solução aproximada (3.17) na equação diferencial (3.16) obtém-se:

$$R(x, y_m) = f(y_m^n, y_m^{n-1}, \dots, y_m', y_m, x) \quad (3.18)$$

Onde  $R(x, y_m)$  é o resíduo que deve ser calculado de acordo com o critério de minimização estabelecido e representado pela integral ponderada dos resíduos.

$$\int_a^b W(x) R(x, y_m) dx = 0 \quad (3.19)$$

Utilizando a equação (3.19), é possível calcular os coeficientes do polinômio.

A função de ponderação  $W(x)$  é que determina os vários tipos de polinômios, sendo que na colocação ortogonal emprega-se a função delta de Dirac.

$$W(x) = \delta(x - x_j) \quad (3.20)$$

Combinando as equações (3.19) e (3.20) e considerando ainda as propriedades da função delta de Dirac tem-se:

$$\int_a^b W(x) R(x, y_m) dx = R(x_j, y_m) = 0 \quad (3.21)$$

Significando, neste caso, que o critério na determinação dos coeficientes do polinômio é a imposição de que o resíduo avaliado nos pontos de colocação sejam nulos.

No método da colocação ortogonal, a solução da equação (3.16) é dada por uma série de polinômios ortogonais cujos pontos de colocação são as raízes do polinômio ortogonal de maior grau.

Definem-se os polinômios  $P_M(x)$  de grau  $M$  e ordem  $M + 1$  como:

$$P_M(x) = \sum_{j=0}^m C_j x^j \quad (3.22)$$

Os coeficientes  $C_j$  da equação (3.22) são definidos de modo que  $P_1$  seja ortogonal a  $P_0$ ,  $P_2$  ortogonal a  $P_1$  e  $P_0$ , ou, de um modo geral,  $P_M$  ortogonal a  $P_K$ , sendo  $K \subseteq M-1$ . Na condição de ortogonalidade pode ser incluída uma função de ponderação  $W(x) \geq 0$ , de modo que:

$$\int_a^b W(x) P_K(x) P_m(x) dx = 0 \quad \begin{matrix} K=0,1,\dots,M-1 \\ M=1,2,\dots,N \end{matrix} \quad (3.23)$$

Geralmente escrevem-se as equações (3.17) e suas derivadas em notação matricial:

$$\underline{y} = \underline{Q} \underline{d} \quad (3.24)$$

$$\frac{d y}{d x} = \underline{C} \underline{d} \quad (3.25)$$

$$\frac{d^2 y}{d x^2} = \underline{D} \underline{d} \quad (3.26)$$

Onde

$$\left. \begin{matrix} Q_{j,i} = x_j^{i-1} \\ C_{j,i} = (i-1) x_j^{i-2} \\ D_{j,i} = (i-1)(i-2) x_j^{i-3} \end{matrix} \right\} \quad (3.27)$$

Pode-se determinar  $\underline{d}$  a partir da equação (3.24)

$$\underline{d} = \underline{Q}^{-1} \underline{y} \quad (3.28)$$

Substituindo a equação (3.28) nas equações (3.25) e (3.26) obtém-se:

$$\frac{d y}{d x} = \underline{C} \underline{Q}^{-1} \underline{y} \quad (3.29)$$

$$\frac{d^2 y}{dx^2} = \underline{\underline{D}} \underline{\underline{Q}}^{-1} \underline{y} \quad (3.30)$$

ou

$$\frac{dy(x_j)}{dx} = \sum_{i=1}^{n+2} A_{j,i} y(x_i) \quad (3.31)$$

$$\frac{d^2 y(x_j)}{dx^2} = \sum_{i=1}^{n+2} B_{j,i} y(x_i) \quad (3.32)$$

Deste modo, as derivadas em qualquer ponto de colocação podem ser expressas como uma somatória ponderada dos valores da função nos pontos de colocação.

As matrizes A e B dependem do polinômio ortogonal escolhido. Neste trabalho, serão utilizados os polinômios de Jacobi considerando serem os mesmos os mais freqüentemente empregados.

Os coeficientes  $A_{j,i}$  e  $B_{j,i}$  são obtidos a partir das raízes dos polinômios utilizando os algoritmos e sub-rotinas descritos por Villadsen (1978) que sugere o uso da dupla precisão (dezesseis algarismos significativos) na determinação dos pontos de colocação, para evitar erros de truncamento.

### 3.4 - MÉTODOS DE RUNGE-KUTTA

Os métodos de Runge-Kutta (Carnahan, 1969; Barroso et alli, 1987) que são utilizados na solução aproximada de equações diferenciais ordinárias têm três propriedades distintas:

- a) São métodos de um estágio, de modo que para determinar  $y_{n+1}$  precisa-se somente da informação disponível no ponto precedente,  $x_n, y_n$ ;
- b) Eles concordam com a série de Taylor até os termos em  $h^p$ , onde  $p$  assume valores distintos para métodos diferentes e é chamada a ordem do método;
- c) Eles não exigem o cálculo de quaisquer derivadas de  $f(x,y)$ , mas somente da própria função.

A terceira propriedade é que torna estes métodos mais práticos do que os que utilizam as séries de Taylor.

### 3.4.1 - MÉTODOS DE RUNGE-KUTTA DE SEGUNDA ORDEM

Seja

$$y_{j+1} = y_j + h\phi(x_j, y_j, h), \quad j=0,1,\dots,m-1 \quad (3.33)$$

fazendo-se  $\phi(x_j, y_j, h) = \alpha \cdot K_1 + \beta \cdot K_2$  e substituindo-se na equação

(3.33) resulta:

$$y_{j+1} = y_j + h \cdot (\alpha \cdot K_1 + \beta \cdot K_2) \quad j=0,1,\dots,m-1 \quad (3.34)$$

Sejam

$$\left. \begin{aligned} K_1 &= f(x_j, y_j) \\ K_2 &= f\left[x_j + ph, y_j + qh \cdot f(x_j, y_j)\right] \end{aligned} \right\} \quad (3.35)$$

As constantes  $\alpha, \beta, p$  e  $q$  nas equações (3.34) e (3.35) devem ser determinadas para que se obtenham métodos de Runge-Kutta de segunda ordem. Para isso os seguintes passos devem ser seguidos:

1º passo: expande-se  $K_2$  numa série de Taylor de duas variáveis, abandonando-se todos os termos a partir do termo em  $h^2$ , inclusive:

$$\begin{aligned} K_2 &= f(x_j, y_j) + ph \frac{\partial f}{\partial x}(x_j, y_j) + qhf(x_j, y_j) \frac{\partial f}{\partial x}(x_j, y_j) \\ &\quad + \sigma(h^2) \end{aligned} \quad (3.36)$$

Substituindo-se (3.35) e (3.36) em (3.34)

$$\begin{aligned} y_{j+1} &= y_j + h\left[(\alpha + \beta)f(x_j, y_j)\right] + h^2\left[\beta p \frac{\partial f}{\partial x}(x_j, y_j) + \beta q f(x_j, y_j) \frac{\partial f}{\partial x}(x_j, y_j)\right] \\ &\quad + \sigma(h^3) \end{aligned} \quad (3.37)$$

2º passo: expande-se a solução teórica  $y(x)$  em torno de  $x_j$  até o termo em  $h^2$ , inclusive:

$$y(x_j + h) = y(x_j) + \frac{h}{1!} f(x_j, y(x_j)) + \frac{h^2}{2!} f^1(x_j, y(x_j)) + \sigma(h^3) \quad (3.38)$$

Como

$$f^1(x_j, y(x_j)) = \frac{\partial f}{\partial x}(x_j, y(x_j)) + f(x_j, y(x_j)) \frac{\partial f}{\partial y}(x_j, y(x_j)) \quad (3.39)$$

Substituindo (3.39) em (3.38) tem-se:

$$y(x_j + h) = y(x_j) + hf(x_j, y(x_j)) + \frac{h^2}{2!} \left[ \frac{\partial f}{\partial x}(x_j, y(x_j)) + f(x_j, y(x_j)) \frac{\partial f}{\partial y}(x_j, y(x_j)) \right] + \sigma(h^3) \quad (3.40)$$

3º passo: comparam-se os termos de mesma potência, com relação a h, em (3.37) e (3.40) e faz-se  $y_j = y(x_j)$ .

$$\begin{aligned} f(x_j, y(x_j)) &= (\alpha + \beta) f(x_j, y(x_j)) \cdot \frac{1}{2} \left[ \frac{\partial f}{\partial x}(x_j, y(x_j)) + f(x_j, y(x_j)) \frac{\partial f}{\partial y}(x_j, y(x_j)) \right] \\ &= \beta p \frac{\partial f}{\partial x}(x_j, y_j) + \beta q f(x_j, y_j) \frac{\partial f}{\partial y}(x_j, y_j) \end{aligned}$$

Das duas igualdades acima, obtém-se o sistema não linear:

$$\left. \begin{aligned} \alpha + \beta &= 1 \\ \beta p &= \frac{1}{2} \\ \beta q &= \frac{1}{2} \end{aligned} \right\} \quad (3.41)$$

4º passo: resolve-se o sistema (3.41). O sistema possui quatro incógnitas e três equações; uma das incógnitas pode ser tomada arbitrariamente. Para a escolha  $\beta=1/2$  tem-se  $\alpha=1/2$ ,  $p=1$  e  $q=1$ . O Runge-Kutta de segunda ordem será:

$$\left. \begin{aligned} y_{j+1} &= y_j + \frac{h}{2}(K_1 + K_2), j = 0, 1, \dots, m-1 \\ K_1 &= f(x_j, y_j) \\ K_2 &= f\left[x_j + h, y_j + hf(x_j, y_j)\right] \end{aligned} \right\} \quad (3.42)$$

Para a escolha  $\beta = 1$ , tem-se  $\alpha = 0$ ,  $p=1/2$  e  $q=1/2$ . O Runge-Kutta de segunda ordem será:

$$\left. \begin{aligned} y_{j+1} &= y_j + h K_2, \quad j = 0, 1, \dots, m-1 \\ K_1 &= f(x_j, y_j) \\ K_2 &= f\left[x_j + \frac{h}{2}, y_j + \frac{h}{2}f(x_j, y_j)\right] \end{aligned} \right\} \quad (3.43)$$

O método expresso pelas equações (3.42) é conhecido como método de Euler melhorado e o das equações (3.43), como método de Euler modificado. É óbvio que o sistema (3.41) possibilita a obtenção de uma infinidade de métodos de Runge-Kutta de segunda ordem.

### 3.4.2 - MÉTODOS DE RUNGE-KUTTA DE ORDENS MAIS ELEVADAS

Os métodos de Runge-Kutta de ordens mais elevadas são obtidos de modo semelhante aos de segunda ordem. Os métodos de terceira ordem, por exemplo, possuem a função incremento:

$$\phi(x_j, y_j, h) = \alpha K_1 + \beta K_2 + \gamma K_3$$

onde  $K_1$ ,  $K_2$  e  $K_3$  aproximam derivadas em vários pontos do intervalo  $[x_j, x_{j+1}]$ .

Aqui se faz:

$$\begin{aligned} K_1 &= f(x_j, y_j) \\ K_2 &= f(x_j + ph, y_j + qhK_1) \\ K_3 &= f\left[x_j + uh, y_j + shK_2 + (u-s)hK_1\right] \end{aligned}$$

Para a determinação de  $\alpha, \beta, \gamma, p, q, s, u$  expandem-se  $K_2$  e  $K_3$  em torno de  $(x_j, y_j)$  numa série de Taylor de duas incógnitas. Expande-se também, a solução teórica  $y(x)$  em torno de  $x_j$ . Os coeficientes de mesma potência de  $h$  até  $h^3$  inclusive, são igualados chegando-se ao sistema:

$$\left. \begin{aligned} \alpha + \beta + \gamma + \delta &= 1 \\ qb + uc &= \frac{1}{2} \\ q^2b + u^2c &= \frac{1}{3} \\ cps &= \frac{1}{6} \end{aligned} \right\} \quad (3.44)$$

Duas incógnitas no sistema (3.44) são arbitrárias. O método de terceira ordem bastante conhecido é o seguinte:

$$\left. \begin{aligned} y_{j+1} &= y_j + \frac{h}{6}(K_1 + 4K_2 + K_3), \quad j = 0, 1, \dots, m-1 \\ K_1 &= f(x_j, y_j) \\ K_2 &= f(x_j + \frac{h}{2}, y_j + \frac{h}{2}K_1) \\ K_3 &= f(x_j + h, y_j + 2hK_2 - hK_1) \end{aligned} \right\} \quad (3.45)$$

Dentre os métodos de Runge-Kutta o mais difundido nas rotinas de cálculo é o de quarta ordem. Este método de Runge-Kutta pode ser definido pelas cinco seguintes equações:

$$\left. \begin{aligned} y_{j+1} &= y_j + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4), \quad j = 0, 1, \dots, m-1 \\ K_1 &= f(x_j, y_j) \\ K_2 &= f(x_j + \frac{h}{2}, y_j + \frac{h}{2}K_1) \\ K_3 &= f(x_j + \frac{h}{2}, y_j + \frac{h}{2}K_2) \\ K_4 &= f(x_j + h, y_j + hK_3) \end{aligned} \right\} \quad (3.46)$$

Este método é bem popular, pois é bastante exato, estável e relativamente fácil de programar.

## CAPÍTULO 4. RESULTADOS E DISCUSSÕES

### 4.1 – APLICAÇÃO DO MÉTODO DAS DIFERENÇAS FINITAS ÀS EQUAÇÕES DE BALANÇO

O método das diferenças finitas foi aplicado na variável  $z$  (comprimento da serpentina) no sistema de equações diferenciais parciais descritos pelo balanço de massa e energia do processo em estudo. Utilizou-se a variável comprimento pelo fato dela ter domínio finito, enquanto que as equações diferenciais no tempo foram resolvidas pelo método de Runge-Kutta, uma vez que a variável tempo tem domínio infinito.

A seguir tem-se a aplicação do método das diferenças finitas às de balanço de massa e energia adimensional:

Balanço de massa adimensional:

$$\frac{\partial W_j}{\partial t} = - \frac{R \rho_o}{\rho Z_f} \left( \frac{\partial W_j}{\partial z} \right) + \frac{R \rho_o M_j}{G \rho w_{10}} \left( \sum v_{j,i} r_i \right) \quad (4.1)$$

Balanço de energia adimensional:

$$\frac{\partial \tau}{\partial t} = - \frac{R \rho_o}{Z_f \rho} \left( \frac{\partial \tau}{\partial z} \right) + \frac{2U \rho_o}{\rho G C_p T_o} (T_w - \tau T_o) + \frac{R \rho_o}{\rho G C_p T_o} \left( \sum_{i=1}^{NR} (-\Delta H_i) R_i \right) \quad (4.2)$$

Discretizando-se o sistema de equações diferenciais parciais composto pelas equações (4.1) e (4.2) pela utilização da expressão da diferença finita para

trás dada pela equação (3.6), dada por  $\frac{\partial y}{\partial x} = \frac{y_{i+1,j} - y_{i-1,j}}{2 \cdot \Delta x} + \theta [(\Delta x)^2]$  obtém-se:

$$\left. \frac{\partial w_j}{\partial z} \right|_K = \left( \frac{w_{j,K} - w_{j,K-1}}{\Delta z} \right) \text{ para a derivada da fração mássica;} \quad (4.3)$$

e

$$\left. \frac{\partial \tau}{\partial z} \right|_K = \left( \frac{\tau_K - \tau_{K-1}}{\Delta z} \right) \text{ para a derivada da temperatura;} \quad (4.4)$$

Substituindo-se (4.3) em (4.1) resulta em:

$$\frac{\partial W_j}{\partial t'} = -\frac{R \rho_o}{\rho Z_t} \left( \frac{w_{j,K} - w_{j,K-1}}{\Delta z} \right) + \frac{R \rho_o M_j}{G \rho_{W10}} \left( \sum v_{j,i} r_i \right) \quad (4.5)$$

Substituindo-se (4.4) em (4.2) resulta em:

$$\frac{\partial \tau}{\partial t'} = -\frac{R \rho_o}{Z_t \rho} \left( \frac{\tau_K - \tau_{K-1}}{\Delta z} \right) + \frac{2U \rho_o}{\rho G C_p T_o} (T_w - \tau T_o) + \frac{R \rho_o}{\rho G C_p T_o} \left( \sum_{i=1}^{NR} (-\Delta H_i) R_i \right) \quad (4.6)$$

As frações mássicas dos componentes e a temperatura são avaliadas em cada ponto K, sendo K o número de pontos na malha. A malha utilizada neste trabalho é composta por Q pontos, onde Q pode variar de 10 a 100 pontos. Para se calcular a distância entre os pontos da malha precisa-se apenas dividir o comprimento total da serpentina por Q. O primeiro ponto da malha corresponde ao início da serpentina e é nele que são aplicadas as condições de contorno do sistema em estudo. O último ponto encontra-se na saída da serpentina.

Na elaboração do programa foram utilizadas as variáveis X(l) e Y(l) para identificar as frações mássicas dos vários componentes e temperatura do processo, segundo a seguinte denominação:

- X(l): Fração mássica do EDC;
- X(l+Q): Fração mássica do MVC;
- X(l+2Q): Fração mássica do HCl;
- X(l+3Q): Temperatura de processo;
- X(l+4Q): Pressão;
- X(l+5Q): Fração mássica do acetileno;
- X(l+6Q): Inverso do peso molecular médio;

Através da aplicação do método das diferenças finitas, o sistema de equações diferenciais parciais foi transformado num sistema de equações diferenciais ordinárias e o mesmo foi resolvido pelo método de Runge-Kutta. Os resultados obtidos são apresentados a seguir.

Vale salientar que os programas mencionados neste trabalho foram processados num Pentium 4(quatro) com processador de 1.60 GHz e 256 Mega de memória e Sistema Operacional Windows 2000 Professional.

#### 4.1.1 – Resultados da Aplicação - Método das Diferenças Finitas

A seguir são apresentados os gráficos dos resultados obtidos utilizando o método das diferenças finitas para a resolução do sistema de equações diferenciais parciais e o método de Runge-Kutta para o sistema de equações diferenciais ordinárias. A simulação foi realizada até que não houvesse mais variação da fração mássica dos componentes e temperatura de processo com o tempo, tendo atingido os resultados em regime estacionário, segundo referenciado por Ferreira (1996).

Na figura 9 são apresentados os resultados do perfil de concentração dos componentes em regime estacionário. Conforme esperado, a maior parte da reação ocorre na terça parte final da serpentina, onde as temperaturas da serpentina são maiores, o que é esperado teoricamente, pois quanto maior a temperatura maior a taxa da reação, cuja dependência é exponencial.

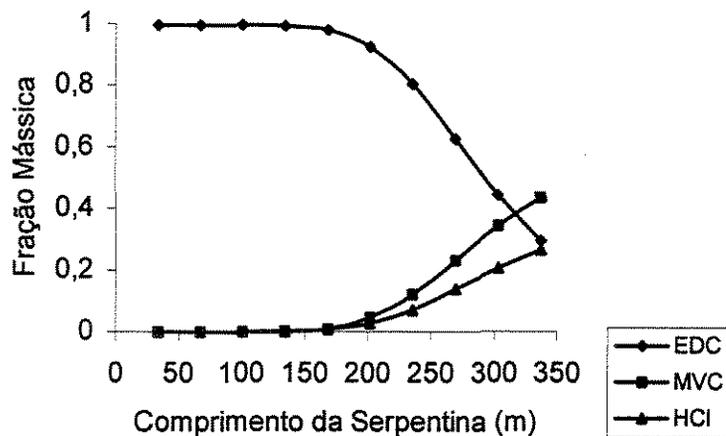


Figura 9 – Perfil de Concentração na Serpentina

Para o perfil de temperatura de processo, observa-se que as maiores temperaturas são obtidas nas zonas de maiores temperaturas de parede, correspondente à zona de radiação da fornalha. O perfil de temperatura de processo obtido encontra-se representado na figura 10. O escoamento da mistura gasosa ao longo da serpentina é caracterizado por um elevado número de Reynolds, o que faz com que a camada limite térmica fique completamente desenvolvida em um curto período de tempo.

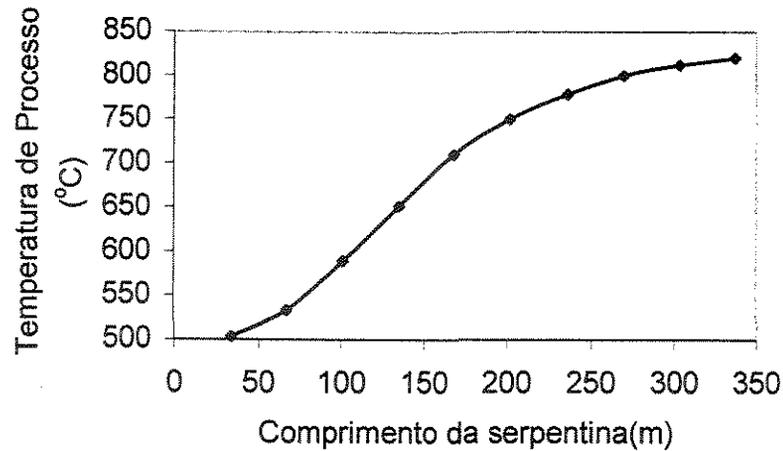


Figura 10 – Perfil de Temperatura de Processo na Serpentina

Na figura 11 tem-se a evolução da concentração do reagente e produtos com o tempo para a saída da serpentina. Observa-se que o processo atinge o estado estacionário em um tempo de aproximadamente 20 (vinte) segundos. Verifica-se que até os 20 primeiros segundos ocorre uma variação das frações mássicas do 1,2-dicloroetano, monocloreto de vinila, ácido clorídrico e acetileno, o que é característico de um processo em estado transiente. Depois de decorrido este tempo, as concentrações dos componentes ficam constantes com o tempo atingindo os valores característicos do estado estacionário.

O processo não é limitado pela transferência de massa, uma vez que em um pequeno período de tempo os perfis de concentração ficam completamente desenvolvidos.

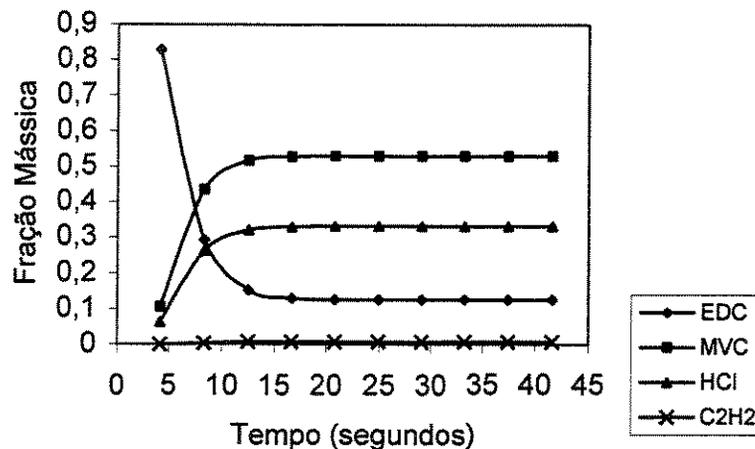


Figura 11 – Evolução do Perfil de Concentração no final da Serpentina.

Na figura 12 tem-se o perfil da temperatura de processo no final da serpentina com o tempo. Através desta figura pode-se observar como varia a temperatura final de processo com o tempo até ser atingido o estado estacionário.

Verifica-se que a dinâmica da transferência de calor é relativamente rápida, pois são necessários aproximadamente 10 (dez) segundos para se atingir o estado estacionário de regime completamente estabelecido para a transferência de calor. Comparando-se com os resultados do perfil de concentração mostrados anteriormente, observa-se que o processo de craqueamento térmico é controlado pela reação química, isto é, as taxas de transferência de calor são relativamente rápidas em relação às taxas de reação química.

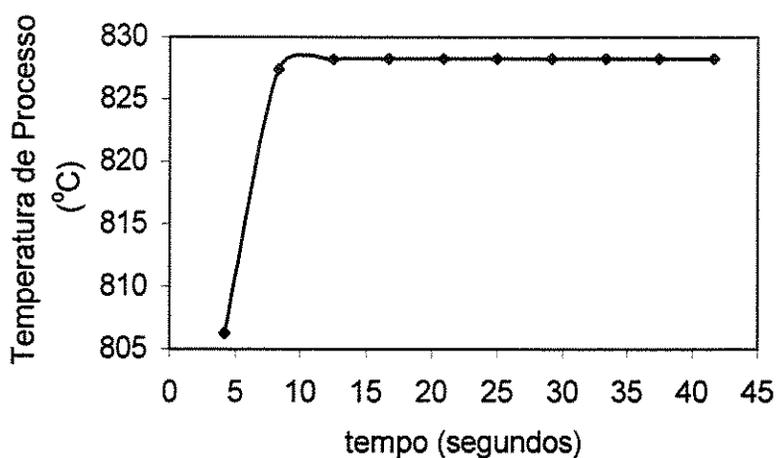


Figura 12 – Evolução do Perfil de Temperatura no final da Serpentina

O perfil de concentração do 1,2-dicloroetano ao longo da serpentina, para diversos tempos de processamento é mostrado na figura 13.

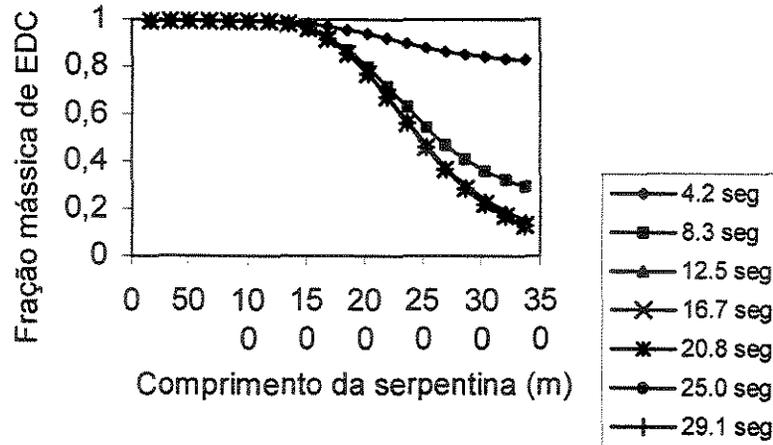


Figura 13 – Evolução do Perfil de Concentração do EDC ao final da Serpentina

Na figura 13 tem-se a evolução da concentração do 1,2-dicloroetano na serpentina. Através deste gráfico é possível visualizar o transiente do processo. Para tempos próximos a 20 (vinte) segundos verifica-se que há uma estabilização no perfil de concentração, o que sugere que já se tenha atingido o estado estacionário (as curvas de concentração ficam bem próximas). Analogamente, para o monocloreto de vinila, têm-se os resultados mostrados na figura 14.

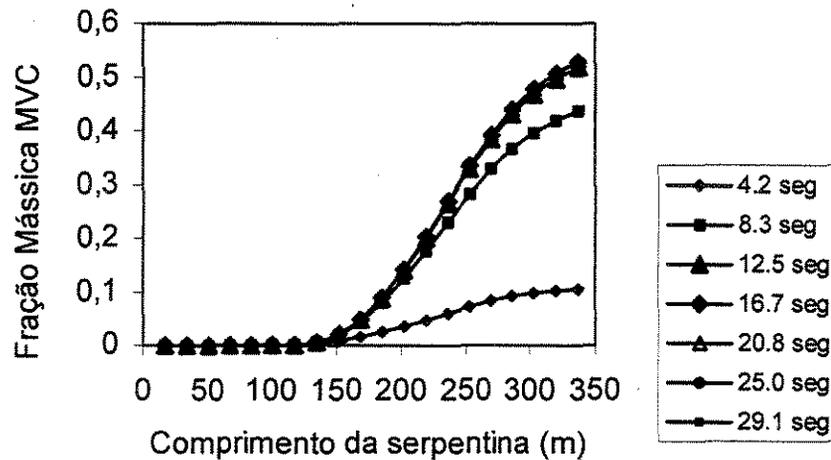


Figura 14 – Evolução do Perfil de Concentração de MVC no final da Serpentina

Observa-se que para o monocloreto de vinila o tempo necessário para atingir o estado estacionário é menor, em torno de 15 (quinze) segundos.

Analogamente ao monocloreto de vinila, obtém-se o perfil de concentração do etileno que é um subproduto da reação secundária.

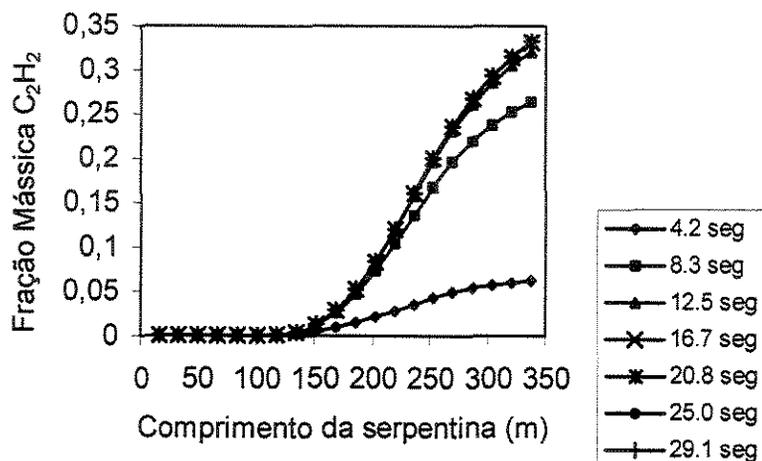


Figura 15 – Evolução do Perfil de Concentração do acetileno na serpentina.

Na figura 16 está representada a evolução da temperatura de processo ao longo da serpentina com o tempo. Observa-se que o transiente do processo de transferência de calor é bem mais rápido que o das reações químicas.

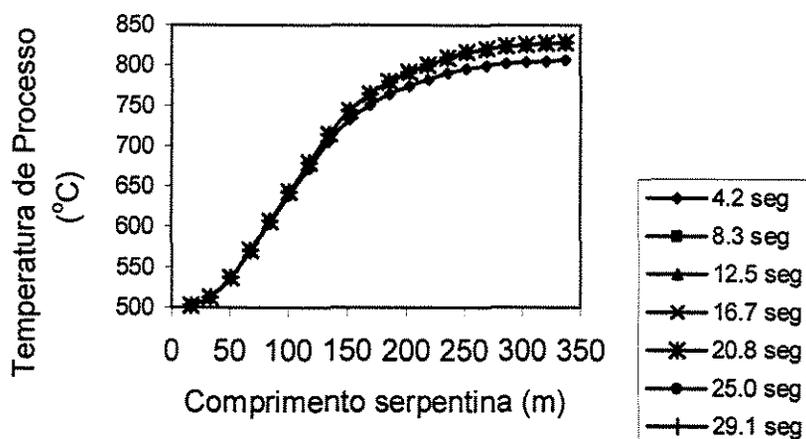


Figura 16 – Evolução do Perfil de Temperatura de Processo na serpentina

Na figura 17 apresenta-se o comportamento da conversão da reação principal e seletividade ao monocloreto de vinila com o tempo desde o estado transiente até o estado estacionário ser alcançado. Verifica-se que a conversão e seletividade chegam ao valor estacionário com aproximadamente 20 (vinte) segundos após ser iniciado o processo, em consequência da transferência de calor ter sido completamente estabelecida.

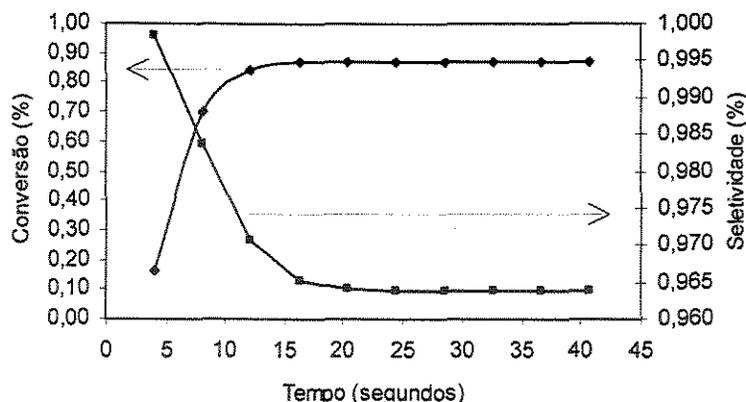


Figura 17 – Perfil da Conversão e Seletividade com o tempo.

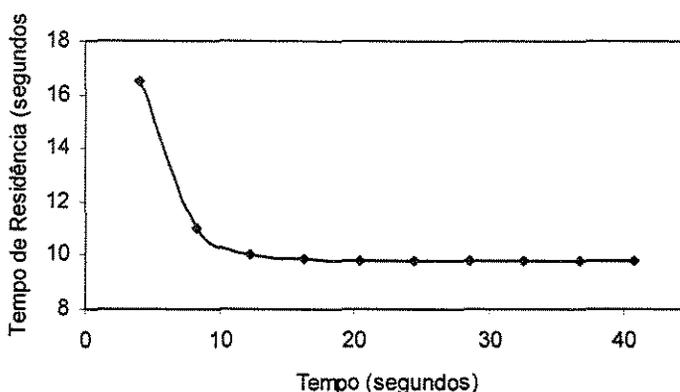


Figura 18 – Perfil do tempo de residência.

Através da figura 18 observa-se que decorrido um curto intervalo de tempo o perfil de temperatura ainda não está completamente desenvolvido devido às resistências à transferência de quantidade de movimento e calor no processo. Temperaturas menores correspondem a pequenos volumes de gás escoando pela serpentina, correspondendo a maiores tempos de residência e menores velocidades espaciais. Com o passar do tempo, existe uma maior transferência de calor para o sistema e conseqüentemente maiores volumes de gás escoam pela serpentina fazendo com que haja uma redução do tempo de residência na mesma com conseqüente aumento da velocidade espacial até atingir o estado estacionário.

#### 4.1.2 - Análise da Influência do Número de Pontos na Malha

Para avaliar a precisão dos resultados obtidos através do método de diferenças finitas aumentou-se o número de pontos na malha e compararam-se os resultados, até que os mesmos estivessem próximos, sendo representativos da solução do problema. Para o perfil de concentração do 1,2-dicloroetano os seguintes resultados foram obtidos:

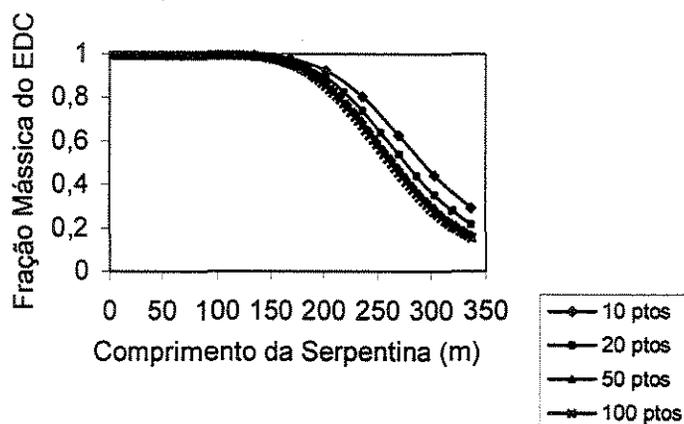


Figura 19 – Evolução do Perfil de Concentração do EDC na serpentina.

Observa-se que com 50 (cinquenta) pontos na malha os resultados são satisfatórios e que o aumento acentuado do número de pontos na malha faz com que a velocidade de processamento diminua, sem que ocorra uma melhora significativa dos resultados obtidos em relação aos resultados esperados industrialmente, segundo citado por Ferreira (1996). A seguir são apresentados os gráficos do perfil de concentração para os outros componentes:

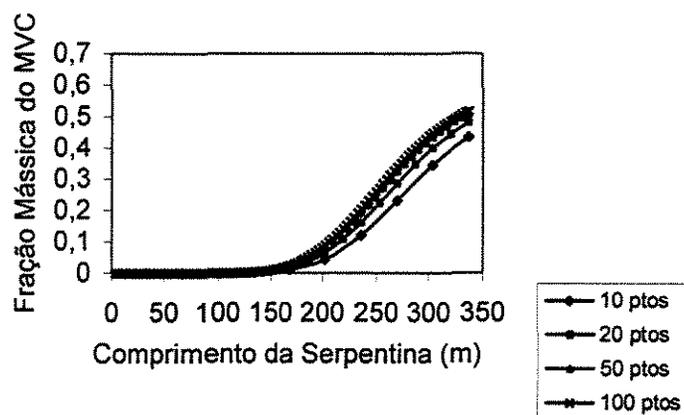


Figura 20 – Evolução do perfil de concentração do MVC na serpentina.

Na figura 21 é apresentado o perfil de temperatura de processo para diferentes números de pontos na malha.

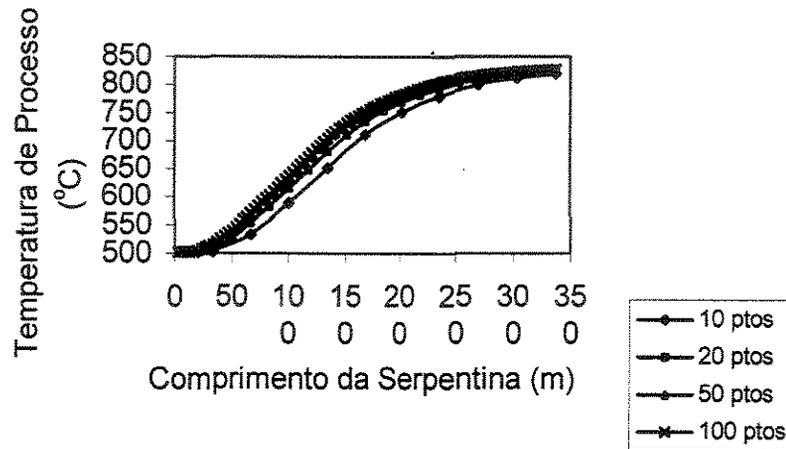


Figura 21 – Evolução do Perfil de Temperatura de Processo na serpentina.

Na figura 22 é mostrada a evolução do tempo de residência na fornalha com o tempo. Verifica-se que para tempos pequenos, o tempo de residência é elevado devido ao fato de que os perfis de transferência de quantidade de movimento, calor e massa ainda não estão completamente desenvolvidos. Para tempos maiores estes perfis tendem a se estabelecer, reduzindo o tempo de residência na fornalha e estabilizando num valor assintótico.

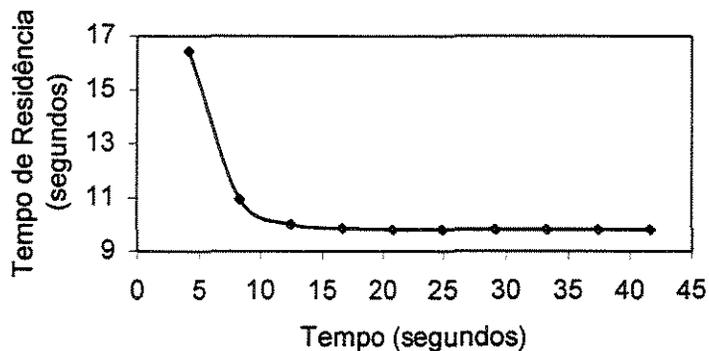


Figura 22 – Evolução do tempo de residência com o tempo

Através dos resultados mostrados anteriormente, conclui-se que 50 (cinquenta) pontos na malha são suficientes para simular o processo com o nível de precisão requerido, utilizando-se o método das diferenças finitas.

## 4.2 – APLICAÇÃO DO MÉTODO DA COLOCAÇÃO ORTOGONAL ÀS EQUAÇÕES DE BALANÇO

É necessário adimensionalizar as equações de balanço para normalizar as faixas dos gradientes, evitando deste modo problemas de divergência. Quando as variáveis são adimensionalizadas, pode-se trabalhar com faixas mais uniformes das mesmas, dentro dos limites pré-estabelecidos.

O método da colocação ortogonal deve ser aplicado numa variável que tenha faixa restrita, assim sendo aplicou-se a colocação ortogonal na variável  $Z$ , comprimento da serpentina. Este comprimento foi adimensionalizado de 0 (zero) a 1 (um). As raízes do polinômio ortogonal são fornecidas pelo programa DMATA.FOR<sup>1</sup> (Ravagnani – 1988) e ela depende do número de pontos internos de colocação e dos valores de alfa e beta, que refletem o quanto as raízes estão mais próximas dos contornos em  $z=0$  e  $z=L$ .

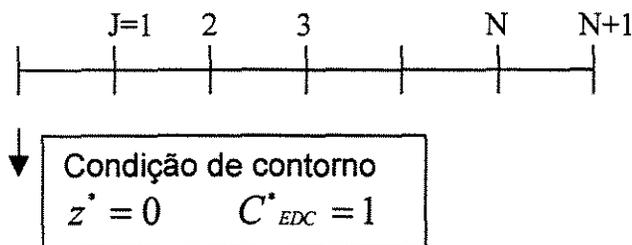
Aplicando o método da colocação ortogonal ao sistema, em termos de massa tem-se:

Condição de contorno na entrada:

$$z = 0 \quad C_{EDC} = C_0$$

Adimensionalizando fica:

$$z^* = 0 \quad C^*_{EDC} = 1$$



<sup>1</sup> Programa desenvolvido por Ravagnani, S. P. (1988), baseado em Algoritmo proposto por Villadsen (1970, 1978).

A derivada parcial da variável concentração em z, pode então ser substituída pelo somatório nos pontos de colocação:

$$\left. \frac{dC}{dz} \right|_j = \sum_{i=1}^{N+1} A_{j,i} C_i \quad (4.7)$$

Onde  $A_{j,i}$  são os coeficientes da matriz A, fornecida pelo programa DMATA.FOR que se encontra no anexo 3.2. Observa-se que no caso em estudo o problema é não simétrico.

Observa-se então que através do método da colocação ortogonal, conseguiu-se transformar uma equação ou sistemas de equações diferenciais parciais num sistema de equações diferenciais ordinárias, o qual pode ser transformado em um sistema de equações algébricas.

Para o Reator Tubular, Modelo Unidimensional, Dinâmico e Pseudo-homogêneo tem-se:

Condição de contorno:

$$z = 0 \quad C_{EDC} = C_0 \quad \forall t$$

Condição inicial:

$$t \subseteq 0 \quad C_{EDC} = 0 \quad \forall z$$

No presente caso, utiliza-se a fração mássica como forma de concentração dos componentes:

### **EQUAÇÕES ADIMENSIONALIZADAS:**

#### **4.2.1 - Balanço de Massa:**

$$\frac{\partial W_j}{\partial t'} = - \left( \frac{R \rho_o}{\rho Z_f} \right) \cdot \left( \frac{\partial W_j}{\partial Z} \right) + \left( \frac{R \rho_o M_j}{G \rho_{W10}} \right) \cdot \left( \sum v_{j,i} r_i \right) \quad (4.8)$$

Observa-se que:

$$\left. \frac{dC}{dz} \right|_j = \sum_{i=1}^{N+1} A_{j,i} C_i, \text{ ou em termos de fração mássica tem-se:} \quad (4.9)$$

$$\left. \frac{dW}{dz} \right|_j = \sum_{i=1}^{N+1} A_{j,i} W_i \quad (4.10)$$

Substituindo (4.10) em (4.8) fica:

$$\frac{\partial W_j}{\partial t'} = - \left( \frac{R \rho_o}{\rho Z_f} \right) \cdot \left( \sum_{i=1}^{N+1} A_{j,i} W_i \right) + \left( \frac{R \rho_o M_j}{G \rho w_{10}} \right) \left( \sum v_{j,i} r_i \right) \quad (4.11)$$

Onde:

$$W_j = \frac{w_j}{w_{10}} \text{ (composição adimensional);}$$

$w_{10}$  = composição de EDC na entrada da serpentina;

$$t' = \frac{t \cdot G}{R \rho_o} \text{ (tempo adimensional);}$$

$G$  = fluxo mássico da mistura gasosa ( $\text{kg/m}^2\text{h}$ );

$R$  = raio da serpentina (m);

$\rho_o$  = massa específica da mistura gasosa na entrada da serpentina ( $\text{kg/m}^3$ );

$$Z = \frac{z}{Z_f} \text{ (comprimento adimensional);}$$

$Z_f$  = comprimento do reator (m).

### 4.2.2 - Balanço de Energia:

Condição de contorno:

$$z = 0 \quad T = T_0 \quad \forall t$$

Condição inicial:

$$t \leq 0 \quad T = T_0 \quad \forall z$$

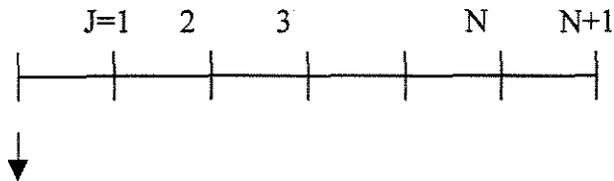
Aplicando o método da colocação ortogonal ao sistema, em termos de energia temos:

Condição de contorno na entrada:

$$z = 0 \quad T = T_0$$

Adimensionalizando fica:

$$z^* = 0 \quad \tau = 1$$



Condição de contorno

$$z^* = 0 \quad \tau = 1$$

A derivada parcial da variável concentração em  $z$ , pode então ser substituída pelo somatório nos pontos de colocação:

$$\left. \frac{d\tau}{dz} \right|_J = \sum_{i=1}^{N+1} A_{j,i} \tau_i \quad (4.12)$$

Onde  $A_{j,i}$  são os coeficientes da matriz  $A$ , fornecida pelo programa DMATA.FOR (Ravagnani, 1988) que se encontra no anexo 3.2.

A equação de balanço de energia adimensionalizada é:

$$\frac{\partial \tau}{\partial t'} = - \left( \frac{R \rho_o}{Z_t \rho} \right) \left( \frac{\partial \tau}{\partial z} \right) + \left( \frac{2U \rho_o}{\rho G C_p T_o} \right) (T_w - \tau T_o) + \left( \frac{R \rho_o}{\rho G C_p T_o} \right) \left( \sum_{i=1}^{NR} (-\Delta H_i) R_i \right) \quad (4.13)$$

Substituindo (4.12) em (4.13) fica:

$$\frac{\partial \tau}{\partial t'} = - \left( \frac{R \rho_o}{Z_t \rho} \right) \left( \sum_{i=1}^{N+1} A_{j,i} \tau_i \right) + \left( \frac{2U \rho_o}{\rho G C_p T_o} \right) (T_w - \tau T_o) + \left( \frac{R \rho_o}{\rho G C_p T_o} \right) \left( \sum_{i=1}^{NR} (-\Delta H_i) R_i \right) \quad (4.14)$$

Onde:

$$\tau = \frac{T}{T_o} \text{ (temperatura adimensional);}$$

$T_o$  = temperatura na entrada da serpentina (K);

$$t' = \frac{t \cdot G}{R \rho_o} \text{ (tempo adimensional).}$$

#### 4.2.3 – Resultados da Aplicação – Método da Colocação Ortogonal

Nas figuras 23 e 24 são apresentados os resultados obtidos do perfil de concentração do 1,2-dicloroetano, monocloreto de vinila, ácido clorídrico e acetileno utilizando cinco pontos de colocação ortogonal.

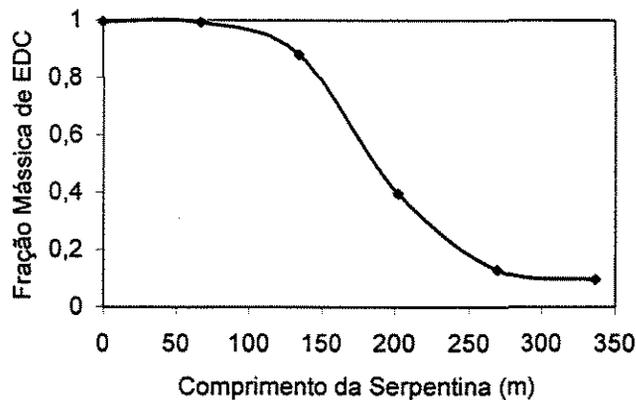


Figura 23 – Perfil de concentração do 1,2-dicloroetano.

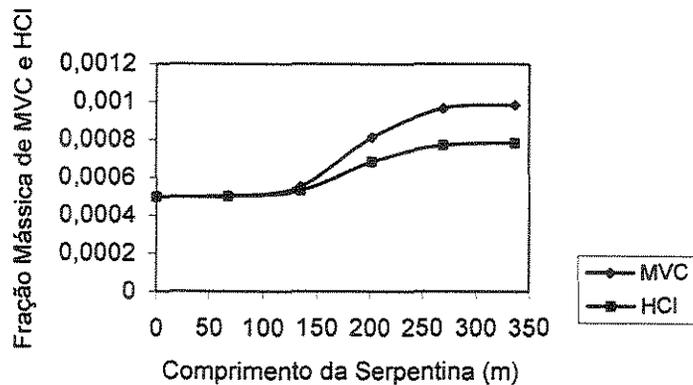


Figura 24 – Perfil de concentração do monocloreto de vinila e 1,2-dicloroetano.

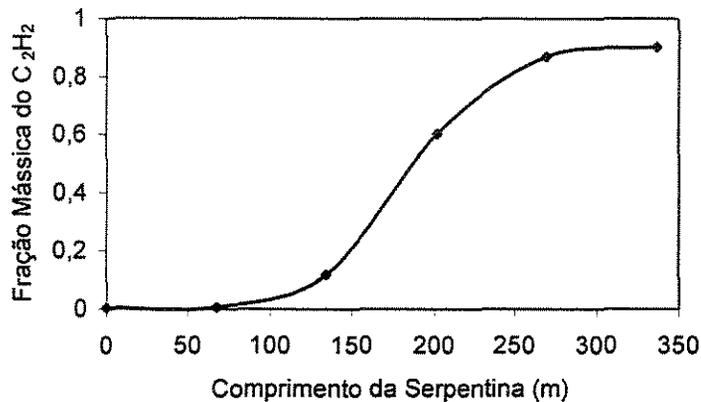


Figura 25 – Perfil de concentração do acetileno.

Comparando-se os perfis obtidos com os resultados das diferenças finitas, observa-se que os valores de concentração do monocloreto de vinila e ácido clorídrico foram menores que os obtidos nas diferenças finitas e os valores de concentração do acetileno foram muito maiores que os obtidos nas diferenças finitas. Foi introduzida uma variação do alfa e beta dos polinômios de Jacobi, entretanto os resultados não sofreram modificação significativa.

A seguir é apresentado o gráfico do perfil da temperatura de processo em função da posição na serpentina.

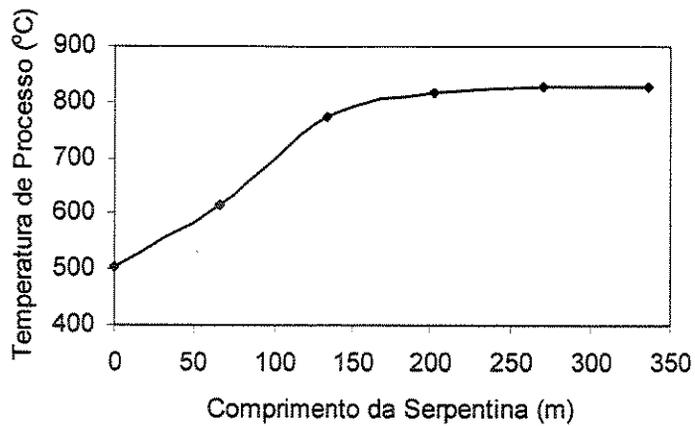


Figura 27 – Perfil de temperatura de processo.

Comparando-se os resultados com os obtidos por diferenças finitas, observa-se que o perfil de temperatura obtido por colocação ortogonal é similar ao obtido por diferenças finitas.

Foi implementado um aumento no número de pontos de colocação para avaliar a influência destes no perfil de concentração e temperatura do processo estudado. Os resultados são mostrados nas figuras que se seguem:

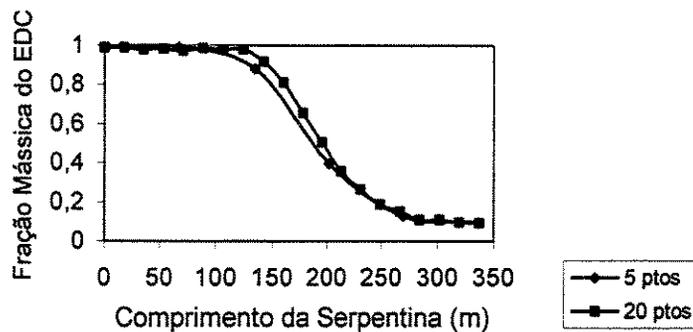


Figura 28 – Perfil de concentração do 1,2-dicloroetano.

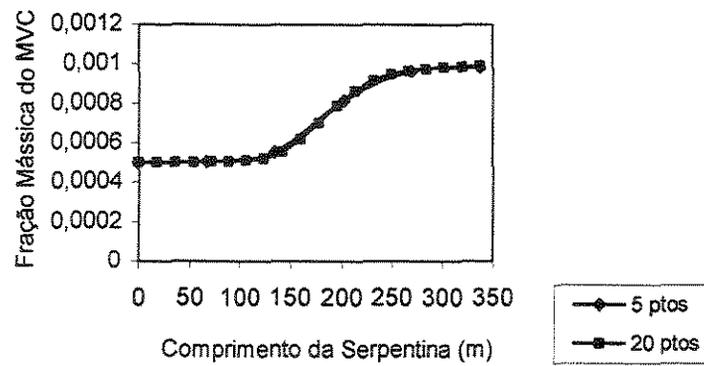


Figura 29 - Perfil de concentração do monoclóreto de vinila.

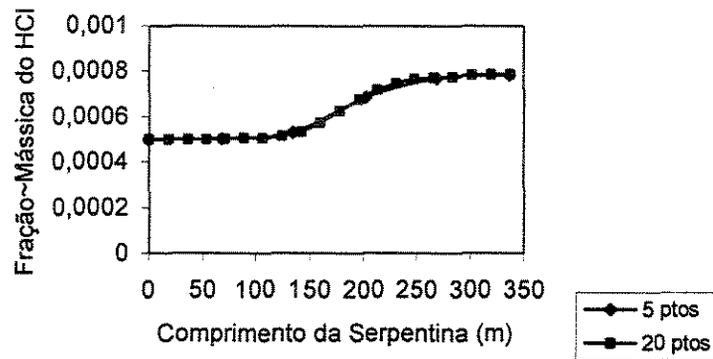


Figura 30 - Perfil de concentração do ácido clorídrico.

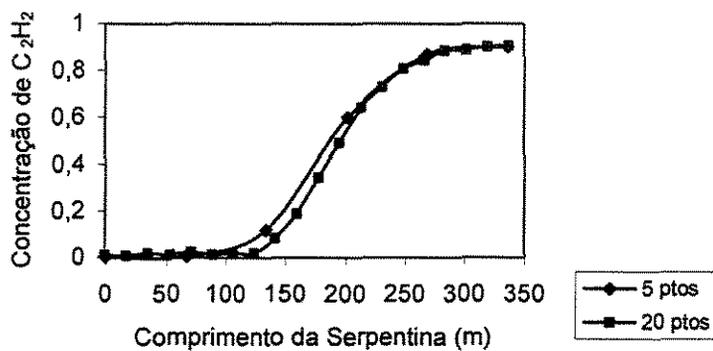


Figura 31 – Perfil de concentração do acetileno.

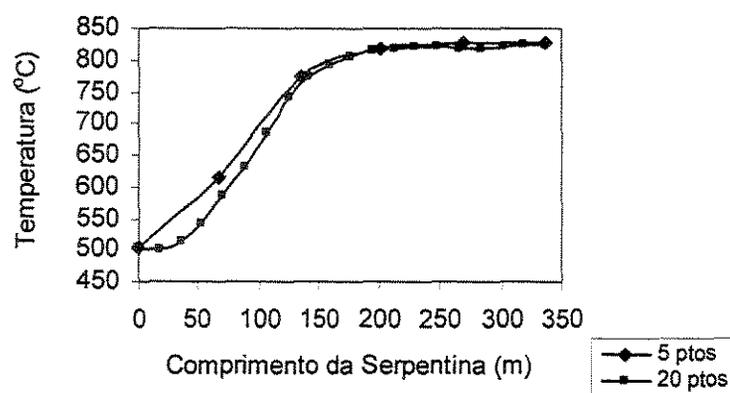


Figura 32 – Perfil de temperatura de processo.

Como evidenciado nos gráficos apresentados anteriormente, o aumento do número de pontos de colocação não influenciou significativamente nos resultados obtidos. Concluiu-se então que este método não foi adequado para a simulação do processo estudado. O método das diferenças finitas foi o mais adequado para a simulação do processo em estudo.

É adequado referenciar que a temperatura na zona de radiação da fornalha varia na faixa de 500 a 860 K (Kelvin) e normalmente a temperatura de reação é tomada como sendo um dos pontos da serpentina, que normalmente é o ponto correspondente ao segundo ponto da temperatura de parede TW2. Assim sendo a temperatura de processo neste ponto fica em torno de 773 K (500°C).

### 4.3 – REDES NEURAIS ARTIFICIAIS – ESTADO ESTACIONÁRIO

Neste item são apresentados os resultados da simulação do craqueamento térmico do 1,2-dicloroetano através das Redes Neurais Artificiais. Nele temos os resultados obtidos da simulação no estado estacionário fazendo variações na vazão mássica da carga, pressão de entrada, temperatura de convecção, temperatura de parede e como resultados tem-se a conversão da reação, tempo de residência e seletividade ao produto principal.

As condições no estado estacionário são mostradas a seguir:

Tabela 1. Condições utilizadas no estado estacionário.

Vm (kg/h)	Go (kg/m <sup>2</sup> h)	tc (°C)	tw <sub>1</sub> (°C)	tw <sub>2</sub> (°C)	tw <sub>3</sub> (°C)	tw <sub>4</sub> (°C)
30.842,00	2.208.679,40	240	370	510	552	557,5

As variações realizadas nas variáveis de entrada foram as seguintes:

Tabela 2. Variações utilizadas no treinamento da Rede.

Variável de entrada	Variação máxima	Variação mínima
Vazão mássica da carga	+20% ss <sup>2</sup>	-20% ss
Pressão de entrada	+10% ss	-10% ss
Temperatura de Parede	+10% ss	-20% ss

A rede neural artificial foi construída com 7 (sete) neurônios de entrada e 1 (um) neurônio de saída por vez.

Tabela 3. Variáveis de Entrada e Saída da Rede.

Entradas	Saídas
Vazão Mássica da Carga – VM – (kg/h)	Conversão da reação (%)
Pressão de Entrada – P0 – (kgf/cm <sup>2</sup> )	
Temperatura de convecção – tc – °C	Seletividade a MVC (%)
Temperatura de parede no ponto 1 (tw <sub>1</sub> ) – °C	
Temperatura de parede no ponto 2 (tw <sub>2</sub> ) – °C	Tempo de Residência (segundos)
Temperatura de parede no ponto 3 (tw <sub>3</sub> ) – °C	
Temperatura de parede no ponto 4 (tw <sub>4</sub> ) – °C	
	etc...

<sup>2</sup> ss- Estado Estacionário; as variações se referem às unidades citadas na Tabela 1.

Na tabela 4 tem-se a matriz de dados de entrada e saída a ser utilizada no treinamento da Rede Neural Artificial. Estes dados foram obtidos a partir do modelo de simulação.

Tabela 4. Matriz de Dados de Entrada e Saída para Treinamento da Rede.

Vazão mássica (t/h)	Pressão entrada (kgf/cm <sup>2</sup> )	Temperatura convecção (°C)	tw <sub>1</sub> (°C)	tw <sub>2</sub> (°C)	tw <sub>3</sub> (°C)	tw <sub>4</sub> (°C)	Conversão (%)	Tempo de Residência (s)	Seletividade (%)
24,67	23,33	240	370	510	552	557,5	60,43	23,53	97,02
27,76	23,33	240	370	510	552	557,5	57,06	20,98	97,40
30,84	23,33	240	370	510	552	557,5	54,05	18,92	97,69
33,93	23,33	240	370	510	552	557,5	51,34	17,21	97,92
37,01	23,33	240	370	510	552	557,5	48,86	15,76	98,11
24,67	23,33	264	407	561	607,2	613,25	92,52	20,74	65,79
27,76	23,33	264	407	561	607,2	613,25	90,10	18,50	74,88
30,84	23,33	264	407	561	607,2	613,25	87,64	16,69	80,44
33,93	23,33	264	407	561	607,2	613,25	85,20	15,19	84,14
37,01	23,33	264	407	561	607,2	613,25	82,80	13,92	86,74
24,67	23,33	252	388,5	535,5	579,6	585,38	78,50	22,07	91,50
27,76	23,33	252	388,5	535,5	579,6	585,38	75,15	19,69	92,97
30,84	23,33	252	388,5	535,5	579,6	585,38	72,05	17,76	94,01
33,93	23,33	252	388,5	535,5	579,6	585,38	69,17	16,16	94,78
37,01	23,33	252	388,5	535,5	579,6	585,38	66,49	14,81	95,38
24,67	23,33	228	351,5	484,5	524,4	529,6	42,06	25,09	98,81
27,76	23,33	228	351,5	484,5	524,4	529,6	39,19	22,35	98,94
30,84	23,33	228	351,5	484,5	524,4	529,6	36,68	20,13	99,04
33,93	23,33	228	351,5	484,5	524,4	529,6	34,46	18,30	99,13
37,01	23,33	228	351,5	484,5	524,4	529,6	32,47	16,75	99,20
24,67	23,33	216	333	459	496,8	501,75	25,98	26,67	99,52
27,76	23,33	216	333	459	496,8	501,75	23,86	23,73	99,57
30,84	23,33	216	333	459	496,8	501,75	22,04	21,35	99,61
33,93	23,33	216	333	459	496,8	501,75	20,45	19,38	99,64
37,01	23,33	216	333	459	496,8	501,75	19,05	17,73	99,67
24,67	23,33	204	314,5	433,5	469,2	473,875	13,84	28,18	99,82
27,76	23,33	204	314,5	433,5	469,2	473,875	12,51	25,03	99,84
30,84	23,33	204	314,5	433,5	469,2	473,875	11,40	22,50	99,85
33,93	23,33	204	314,5	433,5	469,2	473,875	10,45	20,41	99,87
37,01	23,33	204	314,5	433,5	469,2	473,875	9,62	18,65	99,88
24,67	23,33	192	296	408	441,6	446	6,21	29,56	99,94
27,76	23,33	192	296	408	441,6	446	5,54	26,23	99,95
30,84	23,33	192	296	408	441,6	446	4,98	23,55	99,95
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
27,76	20,997	264	407	561	607,2	613,25	88,12	16,63	79,45
30,84	20,997	264	407	561	607,2	613,25	85,45	14,98	83,75
33,93	20,997	264	407	561	607,2	613,25	82,81	13,61	86,68
37,01	20,997	264	407	561	607,2	613,25	80,22	12,45	88,78

### 4.3.1 - Funções de Ativação da Rede

Algumas funções de ativação podem ser usadas para gerar os sinais de saída de cada camada da rede. A adequação do uso depende do pré-tratamento dos dados e da necessidade de menores tempos de processamento da rede e nível de exatidão e faixa esperada dos resultados a serem calculados. As funções de ativação mais comumente usadas são a purelin, tansig e logsig, que são mostradas nas figuras seguintes.

Quando os dados de entrada da rede estão na faixa de  $-1$  (um) a  $+1$  (um) pode-se usar as funções purelin e tansig. Se os dados estiverem na faixa de  $0$  (zero) a  $1$  (um), deve-se usar a função logsig. A utilização destas funções também depende da faixa dos resultados esperados. Quando os resultados são de  $-1$ (um) a  $+1$  (um) deve-se usar as funções purelin ou tansig na camada de saída. Quando os resultados ficam na faixa de  $0$  (zero) a  $1$  (um) deve-se usar a função logsig.

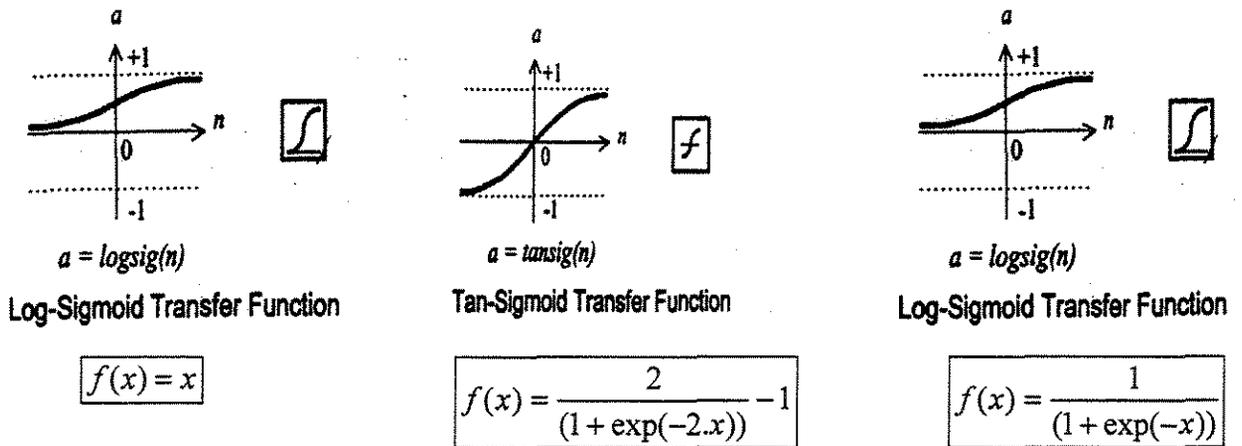


Figura 33 - Representação Gráfica e Algébrica das funções de ativação de RNA's.

### 4.3.2 – Predição da Conversão da Reação

No quadro 1 é apresentado o programa em MATLAB (Matsumoto 2002) da Rede Neural Artificial com 7 (sete) entradas e 1 (uma) saída por vez. Inicialmente os dados de entrada foram normalizados de  $-1$  (um) a  $+1$  (um) e as funções de ativação utilizadas nas camadas intermediária e de saída foram a tansig, logsig e purelin.

Quadro 1 – Programa em Matlab para predição da conversão da reação.

```
load('entradac.m')
p=entradac'
load('saidac.m')
t=saidac'
% normaliza os dados de p e t entre -1 e +1
% pn - entrada normalizada
% tn - saida normalizada
% minp - minimo valor de p
% maxp - maximo valor de p
% tn - saida normalizada
% mint - minimo valor de t
% maxt - maximo valor de t
[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t);
% gradiente descendente
minmax(pn);
net=newff(minmax(pn),[7,1],{'tansig','tansig'}, 'trainlm');
net.trainParam.show = 50;
net.trainParam.lr = 0.1;
net.trainParam.epochs = 3000;
net.trainParam.goal = 1.e-8;
net=train(net,pn,tn);
% Primeiro teste
an=sim(net,pn)
a = postmnmx(an,mint,maxt)
t
pause
[m b r]=postreg(a,t)
pause
net.iw{1,1};
net.lw{2,1};
net.b{1};
net.b{2};
```

Na figura 34 encontra-se representado o desempenho da rede neural treinada para predição da conversão da reação. O critério de parada foi o erro médio quadrático de  $1 \times 10^{-8}$  ou 3000 ciclos de treinamento, o que fosse atingido mais rápido. A curva contínua corresponde aos resultados de treinamento da rede, enquanto que a reta tracejada corresponde ao erro médio máximo desejado.

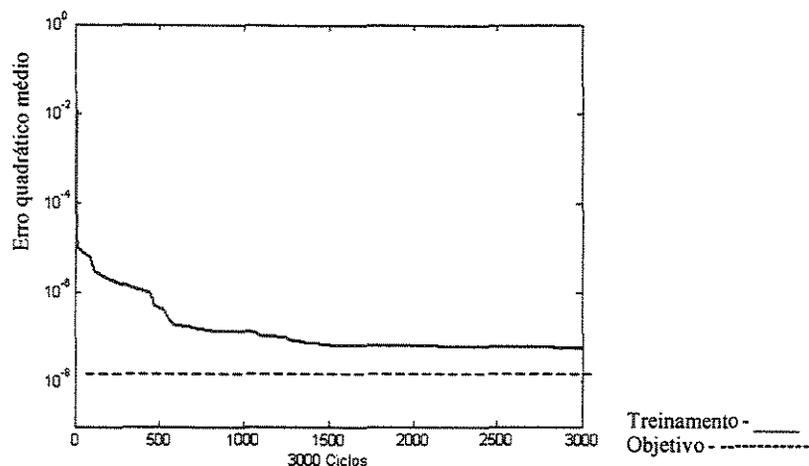


Figura 34 – Evolução do erro quadrático com os ciclos de treinamento.

#### 4.3.2.1 - Análise após Regressão

Na figura 35 temos a representação gráfica da análise após regressão entre os resultados de conversão preditos pela rede e os desejados, do conjunto de dados de entrada e saída não vistos no treinamento.

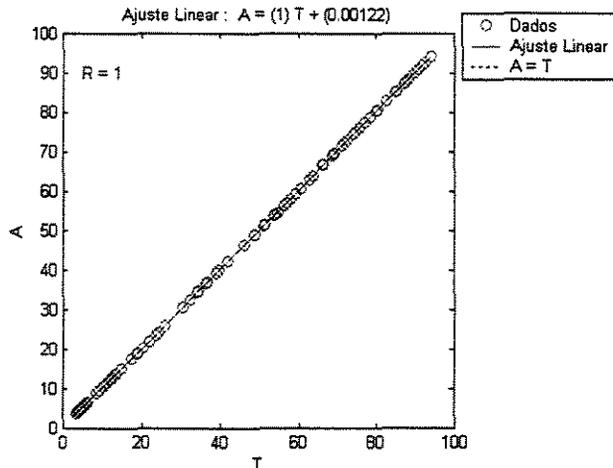


Figura 35 – Ajuste linear entre os dados preditos pela rede e os desejados.

Na análise após regressão, comparando-se os dados treinados pela rede e os desejados, foram obtidos os seguintes coeficientes:  $m = 0,9996$ ,  $b = 0,0186$ ,  $R = 1,0000$ . Observa-se que o coeficiente de correlação entre os dados é um e que a reta obtida foi a reta diagonal, ou seja, os valores preditos pela rede e os valores desejados foram bem próximos. O algoritmo de treinamento utilizado foi o

de Levenberg-Marquadt que utiliza a informação da Hessiana para a busca do mínimo global do erro.

#### 4.3.2.2 - Efeito das Funções de Ativação

Na tabela 5 encontram-se os resultados utilizando o Algoritmo de Levenberg-Marquadt com 7 (sete) neurônios na camada intermediária.

Tabela 5. Efeito das Funções de Ativação no treinamento da Rede.

Estrutura da rede	Erro Quadrático Médio	Coefficiente de correlação
tansig-tansig	6,65E-7	1,0000
tansig-purelin	6,45E-8	1,0000
tansig-logsig	0,3047	0,8968
logsig-logsig	0,3047	0,8968
logsig-tansig	9,12E-6	1,0000
<b>logsig-purelin</b>	<b>1,30E-8</b>	<b>1,0000</b>
purelin-purelin	0,00625	0,9930

Como visto anteriormente, utilizando a entrada da rede normalizada de  $-1$  (um) a  $+1$  (um), a melhor combinação em termos do erro quadrático médio e coeficiente de correlação entre os valores de conversão simulados pela rede e os desejados foi utilizando-se a função de ativação da camada escondida como sendo a "logsig" e a da camada de saída como sendo a "purelin".

#### 4.3.2.3 - Influência da Taxa de Aprendizado

Foi realizada uma avaliação do efeito da taxa de aprendizado no treinamento da rede para predição da conversão. Nas tabelas 6 e 7 e figura 33 são apresentados os resultados.

Tabela 6. Configuração e Parâmetros da rede.

No. de neurônios na camada oculta	7
Normalização dos Dados	0,0 a 1
Algoritmo de aprendizado	Gradiente Descendente
Função de ativação da primeira camada	logsig
Função de ativação da segunda camada	purelin
MSE (Erro Médio Quadrático)	1 E-8

Tabela 7. Resultados do Treinamento para diferentes taxas de aprendizado.

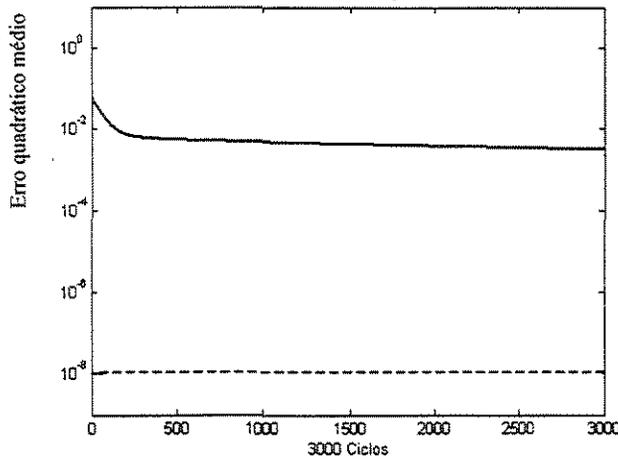
Taxa de Aprendizado $\eta$	Erro Quadrático Médio (MSE)	Análise após a regressão	Tempo de processamento (s)
0,05	3,13 E-3	m=0,985	1.000,6
		b=0,00564	
		R=0,991	
0,1	1,33 E-3	m=0,9860	1.170,8
		b=0,0059	
		R=0,9927	
0,3	1,32 E-3	m=0,9868	1.869,4
		b=0,0056	
		R=0,9928	
0,5	5,15 E-4	m=0,9943	1.636,6
		b=0,0024	
		R=0,9972	
1	420,23	m=3,1472E-4	1.271,4
		b=20,9215	
		R=0,3114	

Através dos resultados mostrados anteriormente, pode-se verificar que de maneira geral um aumento da taxa de aprendizado  $\eta$  favorece a um aumento do erro quadrático médio e uma redução do coeficiente de correlação entre os valores treinados pela rede e os desejados. Vale ressaltar que com uma taxa de aprendizado de 0,5, houve um aumento inicial do erro quadrático médio com os ciclos de treinamento e só depois ele tendeu a diminuir.

#### 4.3.2.3.1 - Comparação do Efeito da Taxa de Aprendizado

Foram comparados os gráficos obtidos da análise após regressão dos dados simulados e dos dados desejados de conversão. Verifica-se que para uma taxa de aprendizado muito alta a correlação entre os dados não é boa. Altas taxas de aprendizado aceleram o treinamento da rede, mas fazem com que haja uma desestabilização dos pesos sinápticos, limiares e gradientes. Vale salientar que os resultados obtidos são dependentes da estimativa inicial da rede, que pode variar de treinamento para treinamento. Observa-se também que com maiores taxas de aprendizado o erro no ciclo tende a aumentar, e depois vai diminuindo com o número de ciclos.

Treinamento da Rede para  $\eta=0,05$



Treinamento da rede para  $\eta=0,5$

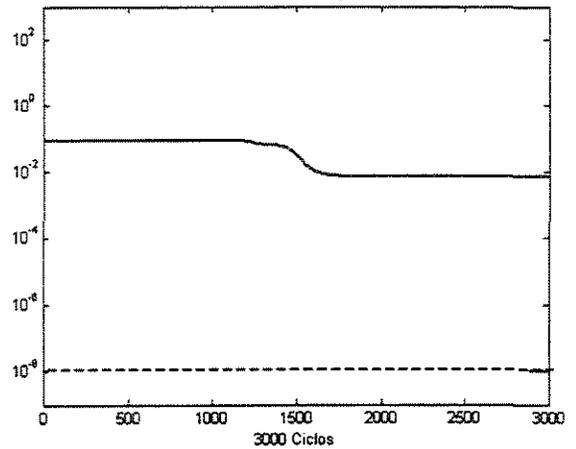
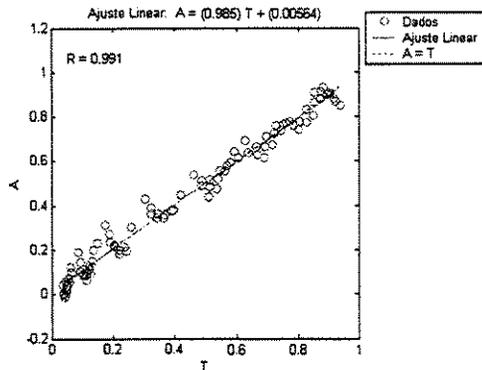


Figura 35 – Treinamento da Rede para Predição da Conversão – Efeito do  $\eta$ . Treinamento - \_\_\_\_\_  
Objetivo - - - - -

Análise após Regressão para  $\eta=0,05$



Análise após Regressão para  $\eta=0,5$

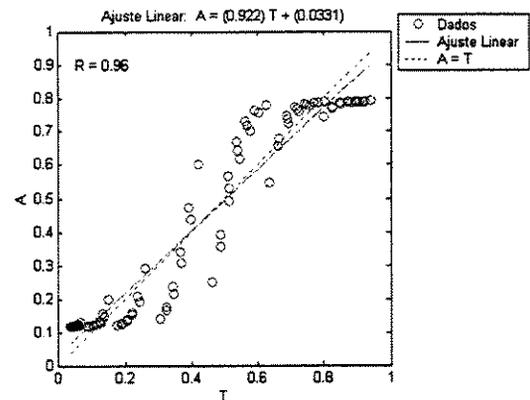


Figura 36 – Gráficos do ajuste linear para diferentes  $\eta$  no treinamento da Rede.

Com o aumento da taxa de aprendizado houve um aumento no tempo de processamento da rede.

#### 4.3.2.4 - Influência do Número de Neurônios da Camada Oculta

O número de neurônios utilizados na camada oculta pode influenciar significativamente no treinamento da rede. Avaliou-se a influência do número de neurônios na predição da conversão da reação. Função de ativação na camada oculta "tansig" e na camada de saída "purelin", com taxa de aprendizado de 1 (um).

Tabela 8. Resultados do Treinamento para diferentes nº. de neurônios.

No. de neurônios da camada oculta	MSE (Erro Quadrático Médio)	Tempo de processamento (s)
7	4,50 E-8	206,64
14	9,98 E-9	300,73
21	4,84 E-9	338,35
28	9,59 E-9	396,25
35	7,35 E-9	459,80
42	1,61 E-13	496,04
49	2,12 E-13	608,91
56	1,19 E-14	649,98

Verifica-se que com o aumento do número de neurônios na camada interna, há uma diminuição do erro quadrático médio, mas, em compensação, há um aumento no tempo de processamento do treinamento da rede. Verificou-se também que um maior número de neurônios na camada intermediária pode provocar um sobre-ajuste dos dados de treinamento.

No quadro 2 é apresentado o programa utilizado para predição da conversão da reação da reação com pré-processamento dos dados de zero a um, sete neurônios na camada oculta e utilizando-se as funções de ativação *logsig* e *purelin*. O algoritmo de treinamento utilizado foi o de Levenberg-Marquadt.

Quadro 2 – programa para predição de conversão.

```
load('entradapre.m')
p=entradapre'
load('saidapre.m')
t=saidapre'
% gradiente descendente
minmax(p);
net=newff(minmax(p),[7,1],{'logsig','purelin'}, 'traingd');
net.trainParam.show = 50;
net.trainParam.lr = .1;
net.trainParam.epochs = 3000;
net.trainParam.goal = 1.e-8;
net=train(net,p,t);
% Primeiro teste
a=sim(net,p)
t
tempo=cputime
pause
[m b r]=postreg(a,t)
net.iw{1,1};
net.lw{2,1};
net.b{1};
net.b{2};
```

Na figura 37 tem-se a representação gráfica da performance da rede de predição da conversão para a configuração da rede citada anteriormente.

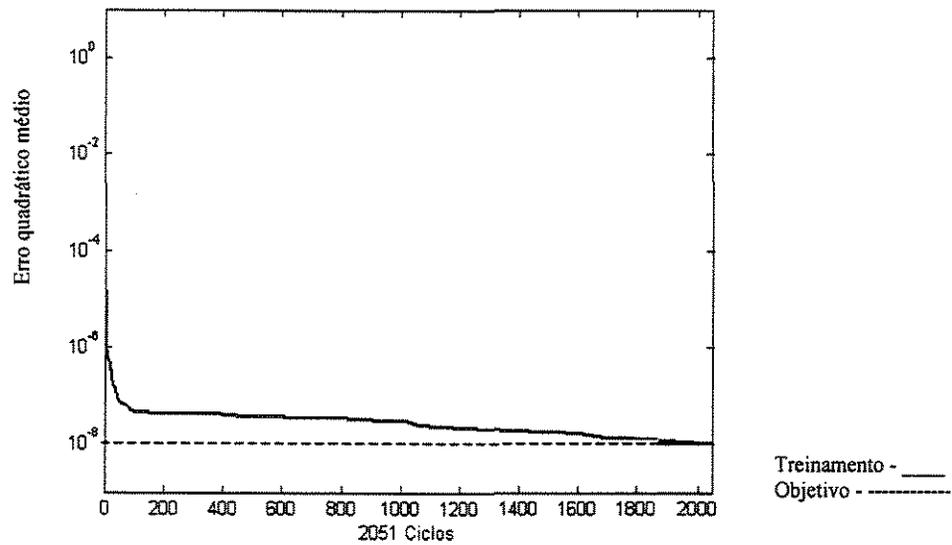


Figura 37 - Evolução do Erro Quadrático Médio com o nº de ciclos de treinamento.

O tempo de treinamento da rede utilizando a função “logsig” foi reduzido para 368,1 segundos e o erro quadrático médio atingiu  $10E^{-9}$  que foi alcançado em 2000 ciclos de treinamento.

A análise após regressão é mostrada na figura 38.

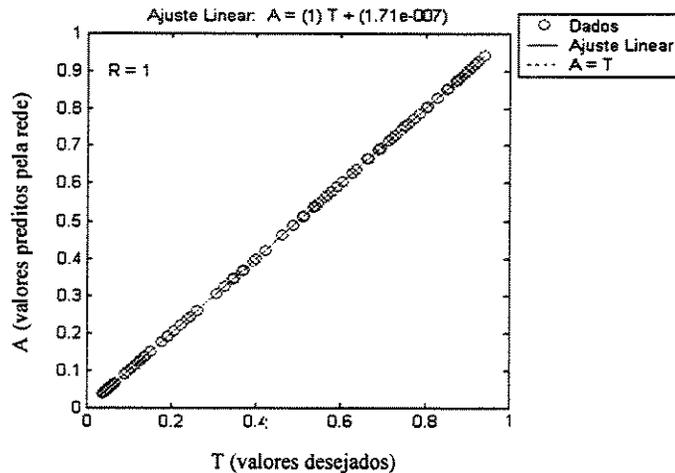


Figura 38 - Análise após regressão entre os resultados simulados e desejados.

Os parâmetros obtidos foram:  $m = 1,0000$ ;  $b = 1,71 E-7$ ;  $R = 1,0000$ .

#### 4.3.2.5 - Análise do Efeito do Algoritmo de Treinamento da Rede

Utilizando-se o programa e os dados citados anteriormente, treinou-se uma rede com diferentes algoritmos de treinamento para avaliar qual o método mais eficaz. A estrutura da rede está apresentada na tabela 9.

Tabela 9. Configuração da Rede para Algoritmos de Treinamento:

No. de neurônios na camada oculta	7
Taxa de Aprendizagem	0,1
Função de ativação da primeira camada	logsig
Função de ativação da segunda camada	purelin
Erro Quadrático Médio - MSE	1E-8

Na tabela 10 são apresentados os resultados obtidos para diferentes algoritmos de treinamento da rede. O algoritmo que apresentou melhor tempo de processamento da rede foi o do gradiente descendente, mas o erro médio quadrático foi elevado. Já o algoritmo de treinamento de Levenberg-Marquadt apresentou o menor erro quadrático médio e o segundo melhor tempo de processamento da rede.

Tabela 10. Resultados da rede para diferentes algoritmos de treinamento.

Algoritmo de Treinamento	Função	Erro Quadrático Médio	Tempo de processamento (s)	Coefficiente de correlação após regressão - R
Gradiente Descendente	traingd	2,25 E-3	2.340,3	0,9877
Gradiente Descendente com Momentum	traingdx	2,20 E-3	2.747,8	0,9879
Gradiente Conjugado Escalonado	trainscg	4,34 E-6	3.161,7	1,0000
Gradiente Conjugado Fletcher-Powell	traincgf	1,64 E-5 atingiu o passo mínimo e o erro não foi alcançado	3.361,0	0,9999
Secante em uma etapa	trainoss	2,46 E-4	3.559,2	0,9987
BFGS quasi-Newton	trainbfg	6,05 E-6 Atingiu o gradiente mínimo e o erro não foi alcançado	3.707,3	1,0000
Levenberg-Marquadt	trainlm	1,00 E-8	2.954,5	1,0000

### 4.3.2.6 - Influência do Algoritmo de Treinamento da Rede – Gradiente Descendente e Levenberg-Marquadt

A seguir são apresentados os gráficos de análise após regressão da predição da conversão utilizando-se o algoritmo do Gradiente Descendente e de Levenberg-Marquadt. Os resultados obtidos pelo treinamento da rede com o algoritmo de Levenberg-Marquadt foram melhores que os resultados obtidos com o algoritmo do Gradiente Descendente, isto porque o algoritmo de Levenberg-Marquadt leva em consideração a segunda derivada da função erro, otimizando a busca do mínimo global. Observa-se também que o coeficiente de correlação dos dados após o treinamento foi melhor no método de Levenberg-Marquadt ( $R=1,0000$ ) que no método do Gradiente Descendente ( $R=0,9800$ ).

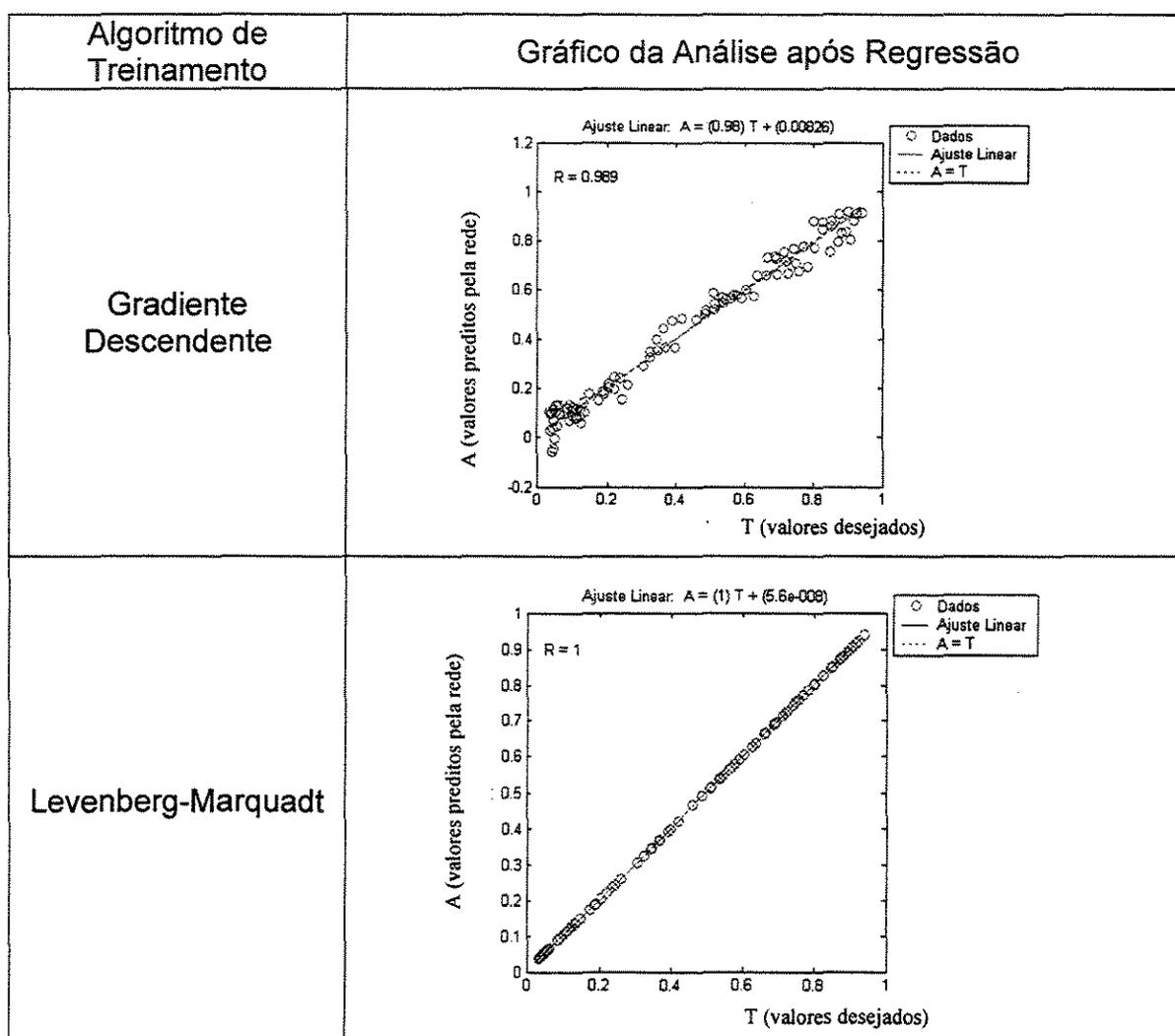
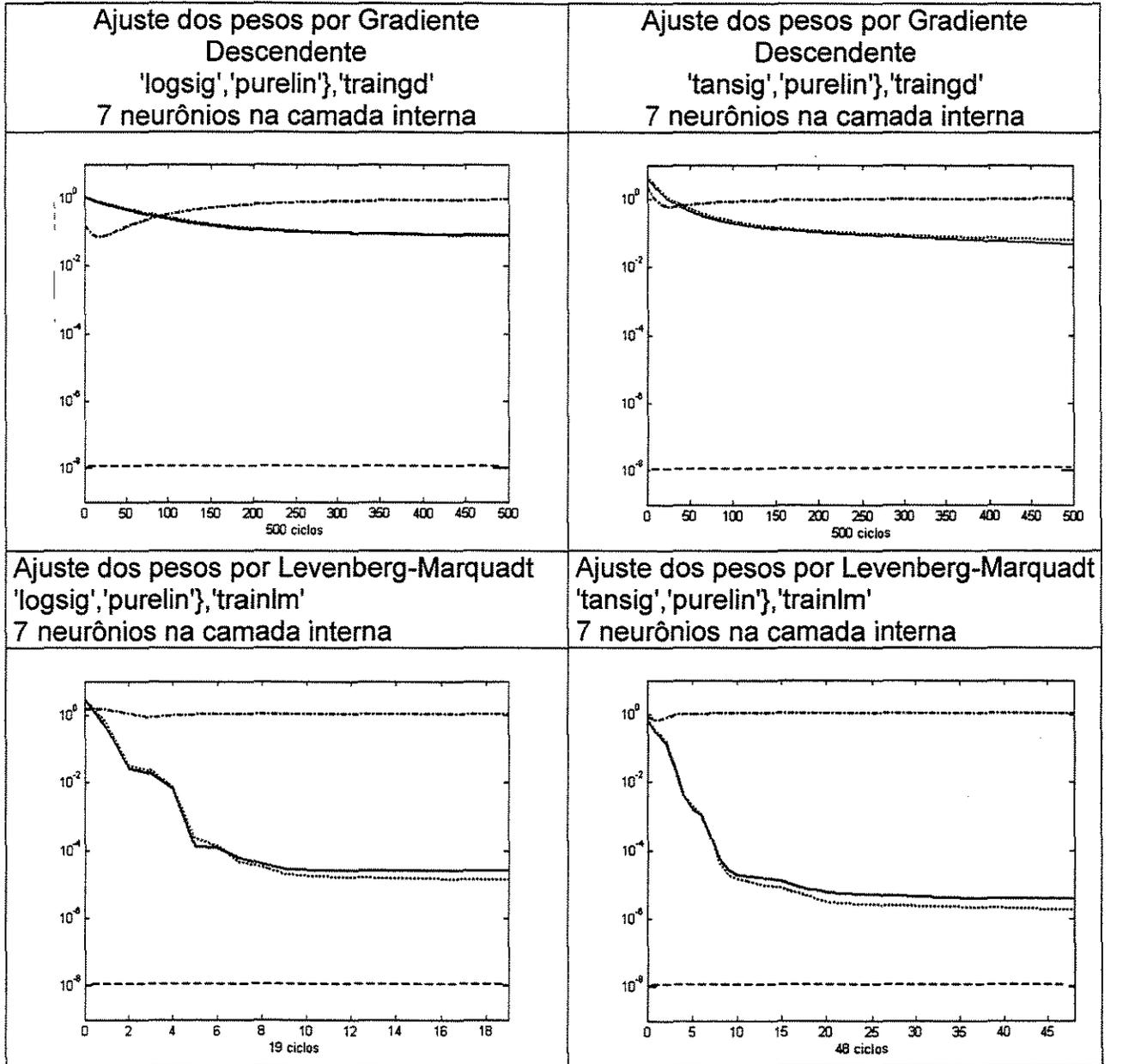


Figura 39 – Análise após Regressão com método de ajuste dos pesos pelo Gradiente Descendente e Levenberg-Marquadt.

Na figura 40 tem-se o comportamento do treinamento, teste e validação da rede para os algoritmos de treinamento da rede através do Gradiente Descendente e Levenberg-Marquadt.



Treinamento - .....Validação \_\_\_\_\_ Teste -.-.-.-.- Objetivo -

Figura 40 – Gráficos do Treinamento, Teste e Validação da Rede para GD e LM.

Através dos gráficos verifica-se que o Método de Levenberg-Marquadt fornece uma maior variação na trajetória de espaços dos erros quadráticos médios. O método do Gradiente Descendente fornece uma variação mais suave

na trajetória de espaço dos erros. De maneira geral a velocidade de processamento utilizando-se o método de Levenberg-Marquadt é maior que usando-se o método do Gradiente Descendente.

#### 4.3.2.7 - Aceleração do treinamento da Rede pela utilização dos pesos e bias de treinamentos anteriores

Para acelerar o treinamento ou o teste de uma outra rede, podem-se utilizar os pesos e bias de treinamentos anteriores. Faz-se necessário então, que uma vez treinada a rede, sejam armazenados os seus pesos e bias e estes sejam carregados num treinamento ou teste posterior.

O programa é descrito no quadro 3, onde os vetores,  $w_1$ ,  $w_2$ ,  $b_1$  e  $b_2$  armazenam os pesos e bias de treinamento anteriores.

Quadro 3 - Programa de processamento da rede com pesos e bias anteriores.

```
load('entradapre.m')
p=entradapre'
load('saidapre.m')
t=saidapre'
iitst = 2:4:95;
p=p(:,iitst); t= t(:,iitst);
% gradiente descendente
minmax(p);
net=newff(minmax(p),[7,1],{'logsig','logsig'}, 'trainlm');
net.iw{1,1}=w1;
net.lw{2,1}=w2;
net.b{1}=b1;
net.b{2}=b2;
net.trainParam.show = 50;
net.trainParam.lr = 0.5;
net.trainParam.epochs = 3000;
net.trainParam.goal = 1.e-8;
net=train(net,p,t);
pause
% Primeiro teste
a=sim(net,p)
t
tempo=cputime
pause
[m b r]=postreg(a,t)
pause
```

Nas figuras 41 e 42 tem-se o desempenho da rede treinada para predição da conversão com um subconjunto do conjunto total de dados. Ocorre uma otimização no tempo de treinamento e teste da rede. Observa-se que a rede atingiu o critério de erro médio quadrático mínimo desejado em apenas 20 (vinte) ciclos de treinamento.

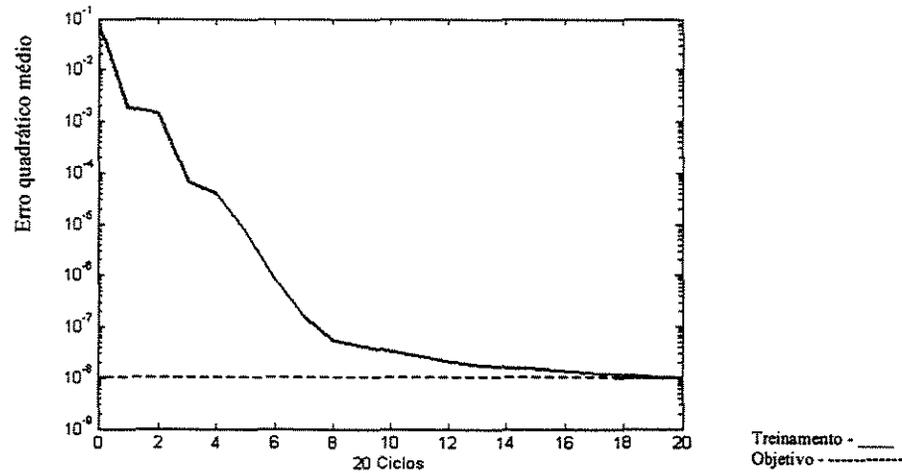


Figura 41 - Erro Quadrático médio para uma rede treinada com parâmetros anteriores.

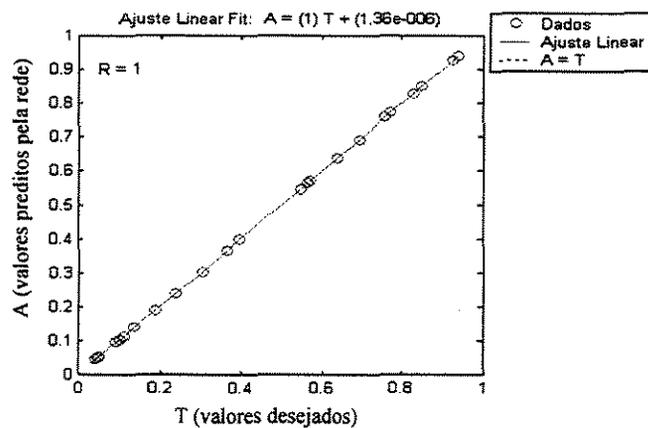


Figura 42 - Ajuste Linear para uma rede treinada com parâmetros anteriores.

### 4.3.3 - Predição da Seletividade

Inicialmente foi realizado o pré-processamento da entrada e saída da rede entre  $-1$  (um) a  $+1$  (um). Foram utilizadas as funções “tansig” na camada oculta e “tansig” na camada de saída. O ajuste dos pesos foi realizado pelo método do Gradiente Descendente, mostrados nas figuras 43 e 44. O programa encontra-se descrito no quadro 4.

Quadro 4 – Programa para predição da seletividade.

```

load('entradapre.m')
p=entradapre'
load('saidaseletpre.m')
t=saidaseletpre'
% normaliza os dados de p e t entre -1 e +1
% pn - entrada normalizada
% tn - saída normalizada
% minp - minimo valor de p
% maxp - maximo valor de p
% tn - saída normalizada
% mint - minimo valor de t
% maxt - maximo valor de t
[pn,minp,maxp,tn,mint,maxt]=premnmx(p,t);
% gradiente descendente
minmax(pn);
net=newff(minmax(pn),[7,1],{'tansig','tansig'}, 'traingd');
net.trainParam.show = 50;
net.trainParam.lr = 0.5;
net.trainParam.epochs = 3000;
net.trainParam.goal = 1.e-8;
net=train(net,pn,tn);
% Primeiro teste
an=sim(net,pn)
pause
a = postmnmx(an,mint,maxt)
t
[m b r]=postreg(a,t)

```

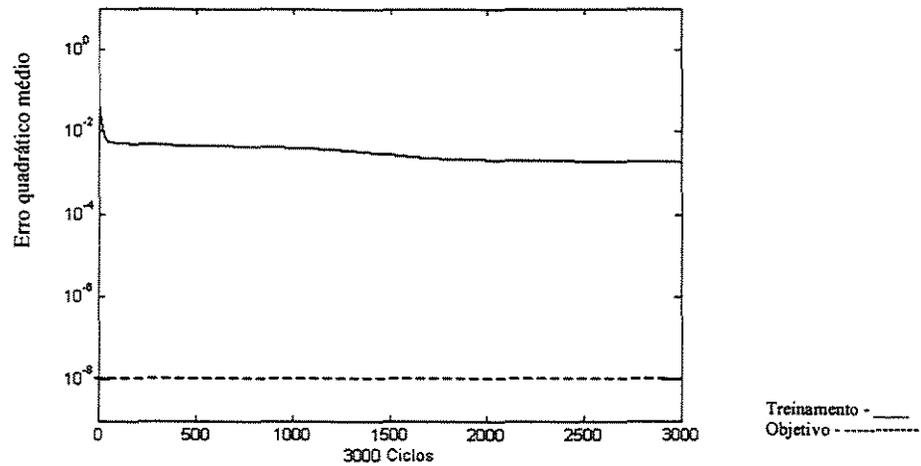


Figura 43 - Erro Quadrático Médio para predição da Seletividade.

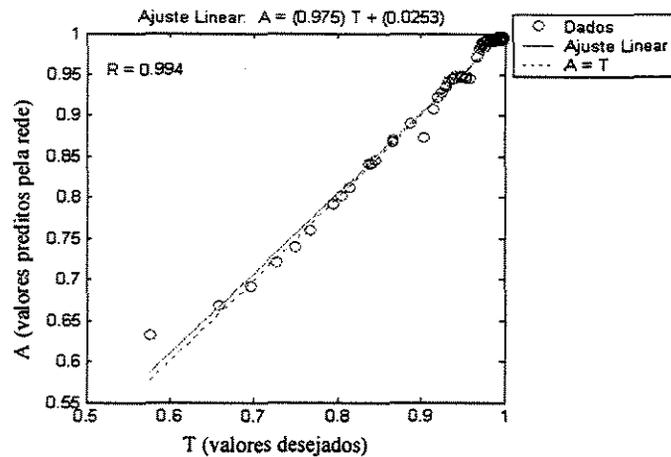


Figura 44 - Regressão dos Dados Simulados versus Desejados para a Seletividade.

Na análise após regressão, comparando-se os dados treinados pela rede com os desejados, foram obtidos os seguintes coeficientes:  $m = 0,999$ ,  $b = 0,0253$ ,  $R = 0,9943$ . Observa-se que o coeficiente de correlação entre os dados é 1 (um). O algoritmo de treinamento utilizado foi o do Gradiente Descendente.

Nas figuras 45 e 46 são mostrados os resultados do treinamento da rede e ajuste dos dados preditos e simulados. Foi utilizado o mesmo programa só que o algoritmo de treinamento usado foi o de Levenberg-Marquadt, que utiliza a informação da Hessiana para a busca do mínimo global do erro. Os seguintes resultados foram obtidos:

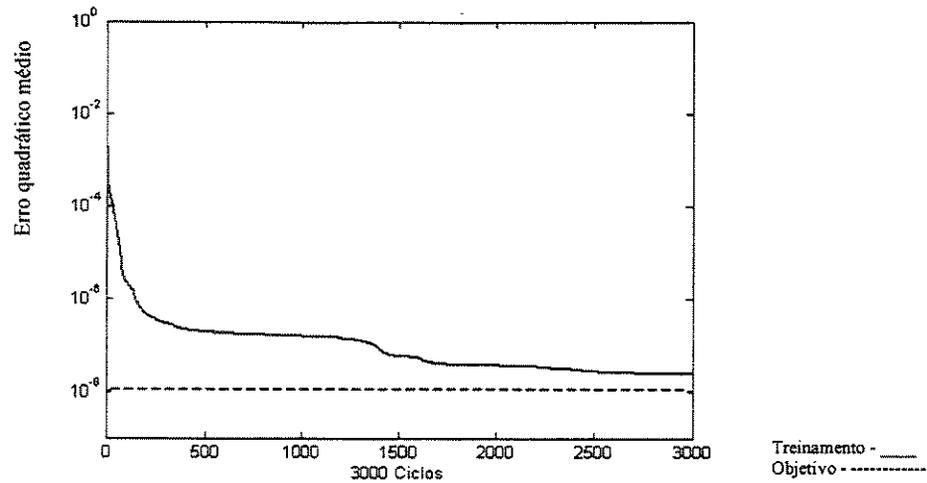


Figura 45 - Erro Quadrático Médio para a predição da Seletividade.

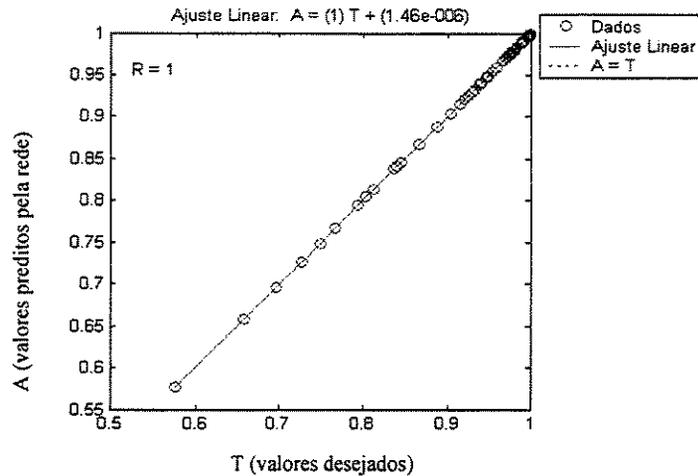


Figura 46 - Regressão dos Dados Simulados versus Desejados para a Seletividade.

Os coeficientes obtidos foram  $m = 1,0000$ ,  $b = 1,4644 \text{ E-}06$ ,  $R = 1,0000$ . Observa-se que o erro médio quadrático obtido pelo método de Levenberg-Marquadt ( $\text{MSE} = 2,3883 \text{ E-}8$ ) foi menor que o erro obtido utilizando-se o método do Gradiente Descendente ( $\text{MSE} = 0,0018$ ).

Na figura 47 são apresentados os resultados de treinamento da rede para seletividade utilizando-se a função “tansig” na camada de entrada e “logsig” na camada escondida, com algoritmo de treinamento de Levenberg-Marquadt.

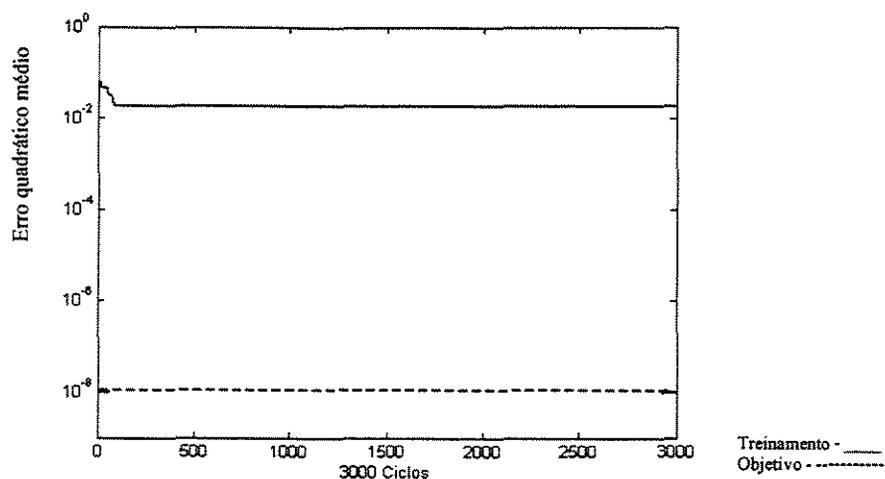


Figura 47 - Erro Quadrático Médio para a Seletividade.

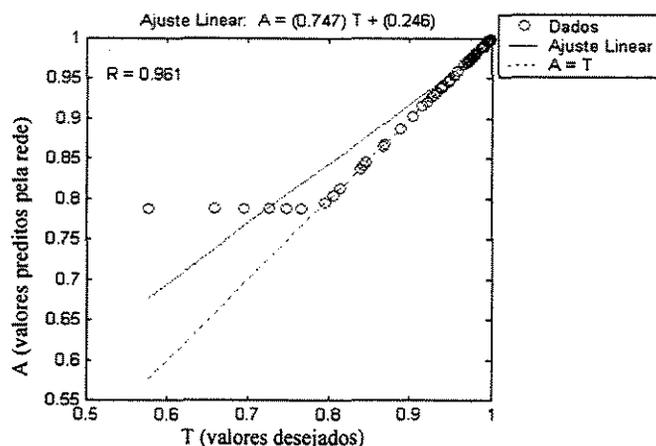


Figura 48 - Regressão dos Dados Simulados versus Desejados para a Seletividade.

Através da utilização da função “logsig” na camada intermediária observou-se uma pior predição da seletividade a monocloreto de vinila. A função “logsig” não foi apropriada para este conjunto de dados normalizados de  $-1$  (um) a  $+1$  (um). O erro quadrático médio obtido foi de 0,018 e os coeficientes para o ajuste linear foram  $m=0,7472$ ;  $b=0,2461$ ;  $R=0,9606$ .

Nas figuras 49 e 50 são apresentados os resultados para o treinamento da rede para seletividade utilizando-se a função “logsig” na camada intermediária e “purelin” na camada de saída, com algoritmo de treinamento de Levenberg-Marquadt.

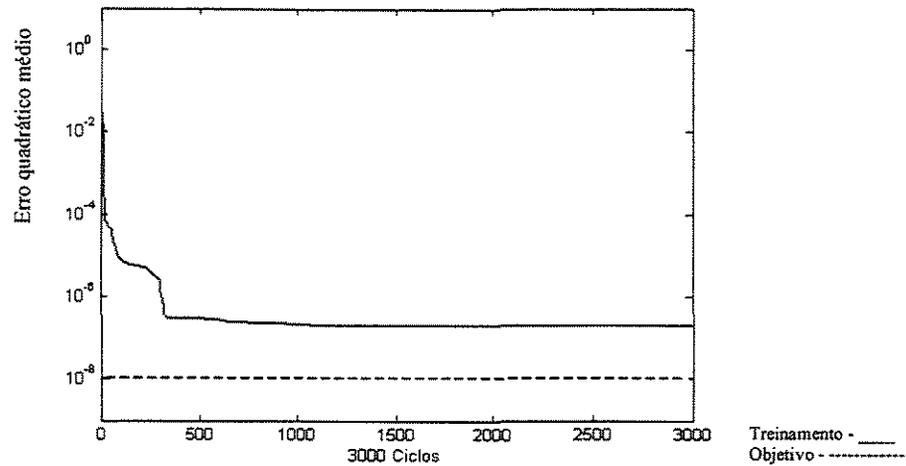


Figura 49 - Erro Quadrático Médio para a Seletividade.

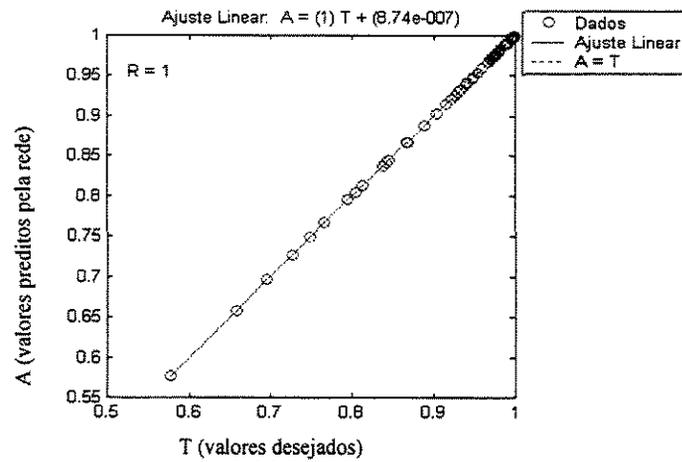


Figura 50 - Regressão dos Dados Simulados versus Desejados para a Seletividade.

#### 4.3.3.1 - Efeito do Pré-processamento dos Dados de Entrada da Rede

Como os resultados apresentados pela rede para a predição da seletividade não foram plenamente satisfatórios, foi realizado um pré-processamento dos dados de forma que eles ficassem na faixa de 0 (zero) a 1 (um), conforme mostrado no quadro 5. Inicialmente utilizou-se as funções de ativação “logsig” e “purelin” nas camadas intermediária e de saída respectivamente com 7 (sete) neurônios internos.

## Quadro 5 – Programa para predição de seletividade.

```

load('entradapre.m')
p=entradapre'
load('saidaseletpre.m')
t=saidaseletpre'
% pre-processamento dos dados de zero a um
% Levenberg-Marquadt
net=newff(minmax(p),[7, 1],{'logsig','purelin'}, 'trainlm');
net.trainParam.show = 50;
net.trainParam.lr = 0.1;
net.trainParam.epochs = 3000;
net.trainParam.goal = 1.e-8;
net=train(net,p,t);
% Primeiro teste
a=sim(net,p)
t
pause
[m b r]=postreg(a,t)
pause
net.iw{1,1};
net.lw{2,1};
net.b{1};
net.b{2};

```

Nas figuras 51 e 52 são apresentados os resultados do treinamento e ajuste de dados da rede para a predição da seletividade utilizando o pré-processamento de 0 (zero) a 1 (um).

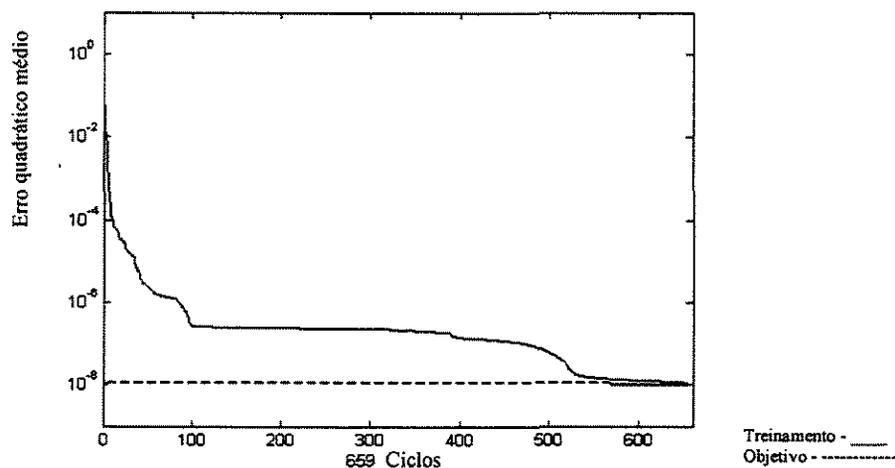


Figura 51 - Erro Quadrático Médio para a Seletividade.

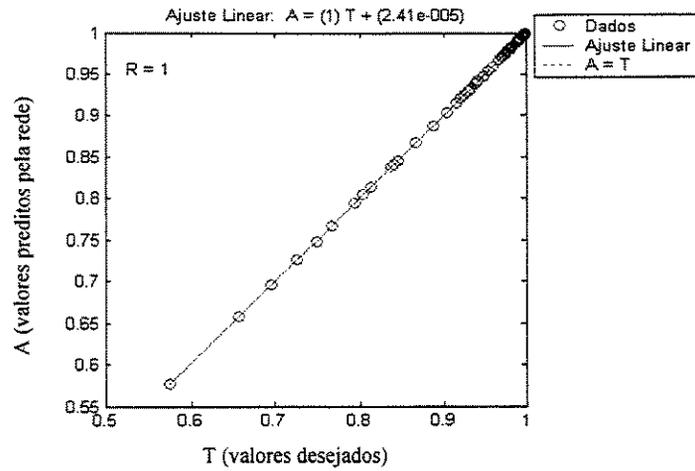


Figura 52 - Regressão dos Dados Simulados versus Desejados para a Seletividade.

Os coeficientes obtidos foram  $m = 1,0000$ ,  $b = 2,4092 E-5$ ,  $R = 1,0000$ . Observa-se que com a utilização dos dados pré-processados de 0 (zero) a 1 (um) e com as funções “logsig” na camada de entrada e “purelin” na camada intermediária, houve um aumento na velocidade de convergência reduzindo-se o número de ciclos de treinamento para 660 (seiscentos e sessenta).

#### 4.3.4 - Predição do Tempo de Residência

Realizou-se o treinamento da rede com o pré-processamento dos dados de entrada de 0 (zero) a 1 (um). O programa é apresentado no quadro 6 e os resultados são mostrados nas figuras 53 e 54. Os pesos e limiares foram ajustados pelo método de Levenberg-Marquadt.

Quadro 6 – Programa para predição do tempo de residência.

```
load('entradapre.m')
p=entradapre'
load('saidatpre.m')
t=saidatpre'
% pre-processamento dos dados de zero a um
% Levenberg-Marquadt
net=newff(minmax(p),[7,1],{'logsig','purelin'}, 'trainlm');
net.trainParam.show = 50;
net.trainParam.lr = 0.1;
net.trainParam.epochs = 3000;
net.trainParam.goal = 1.e-8;
net=train(net,p,t);
% Primeiro teste
a=sim(net,p)
t
[m b r]=postreg(a,t)
pause
net.iw{1,1};
net.lw{2,1};
net.b{1};
net.b{2};
```

Observa-se que para o treinamento da rede para predição do tempo de residência foram necessários aproximadamente 400 ciclos de treinamento e o ajuste linear coeficiente de correlação de 1 (um) na reta diagonal.

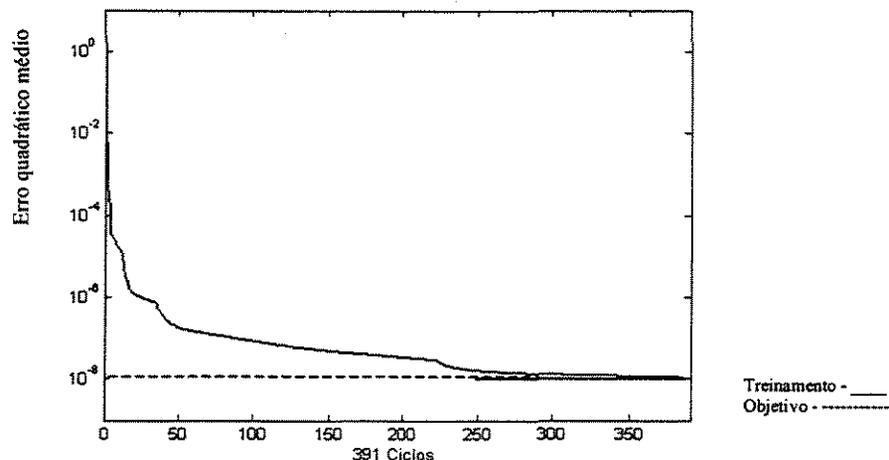


Figura 53 - Erro Quadrático Médio para o Tempo de Residência.

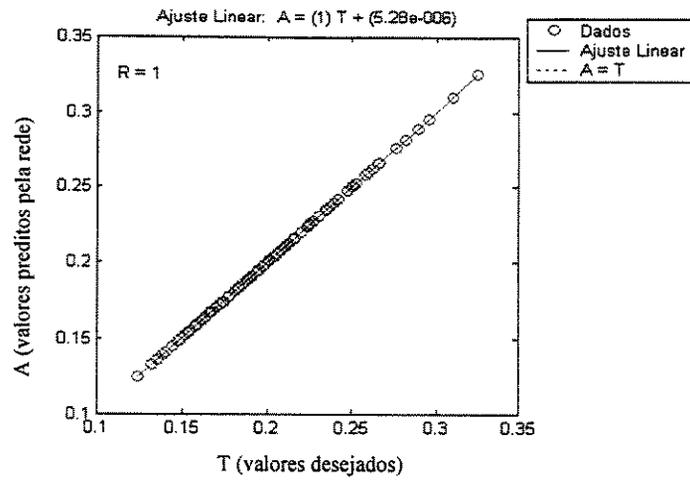


Figura 54 - Regressão dos Dados Simulados versus Desejados Tempo de Residência.

### 4.3.5 - Aplicação da PCA (Análise dos Componentes Principais) como Pré-tratamento nas Redes Neurais Artificiais

A técnica da análise dos componentes principais (PCA) foi aplicada para o treinamento da rede para predição da conversão com o objetivo de reduzir o tempo de processamento da rede. O programa está descrito no quadro 7.

Quadro 7 – Treinamento, Teste e Validação da Rede Neural para predição da conversão utilizando a PCA como pré-tratamento dos dados de entrada.

```
% leitura das entradas e saídas - 7 entradas e 1 saída
load('entradacpre.m')
p=entradac'
load('saidacpre.m')
t=saidac'
% normalização linear centrada na média dos dados de entrada e
saída entre -1 e 1
[pn,meanp,stdp,tn,meant,stdt]=prestd(p,t);
TT=t
pn
pause
tn
pause
% realização do PCA com 99% dos dados sendo representados
[Ptrans,TransMat]=prepca(pn,0.001);
[R,Q]=size(Ptrans)
R
Q
pause
% criação dos índices e separação dos conjuntos de teste
validação e treinamento
iitst=2:4:Q
iival=4:4:Q
iitr=[1:4:Q 2:4:Q]
% pegando os valores da matriz de entrada e saída
v.P=Ptrans(:,iival);v.T=tn(:,iival)
t.P=Ptrans(:,iitst);t.V=tn(:,iitst)
ptr=Ptrans(:,iitr);ttr=tn(:,iitr)
% treinamento da rede com o conjunto de treinamento
pause
net=newff(minmax(ptr),[14 1],{'tansig','purelin'},'trainlm');
net.trainParam.epochs=500;
net.trainParam.show=50;
net.trainParam.goal=1.e-9;
[net,tr]=train(net,ptr,ttr,[],[],v,t);
pause
% parte de verificação
plot(tr.epoch,tr.perf,tr.epoch,tr.vperf,tr.epoch,tr.tperf)
legend('treinamento','validacao','teste'),-1;
ylabel('erro quadratico');xlabel('ciclos')
```

O objetivo da Análise dos Componentes Principais (PCA) é reduzir a complexidade da representação do sistema. Através da utilização desta técnica no craqueamento térmico do 1,2-dicloroetano obtivemos os seguintes resultados:

Redução da dimensionalidade do problema de 7 (sete) para 3 (três), isto é, necessitou-se apenas de 3 (três) variáveis ou combinações de variáveis mais representativas do sistema para descrever a predição da conversão do 1,2-dicloroetano na reação principal. A análise dos componentes principais foi realizada com representação de 99% dos dados. Obteve-se um erro médio quadrado de treinamento de  $6,8 \text{ E-}7$ , entretanto os erros da validação e teste foram maiores.

Como visto através da figura 55, o erro quadrático médio do conjunto de teste ficou muito maior que o do conjunto de treinamento e validação.

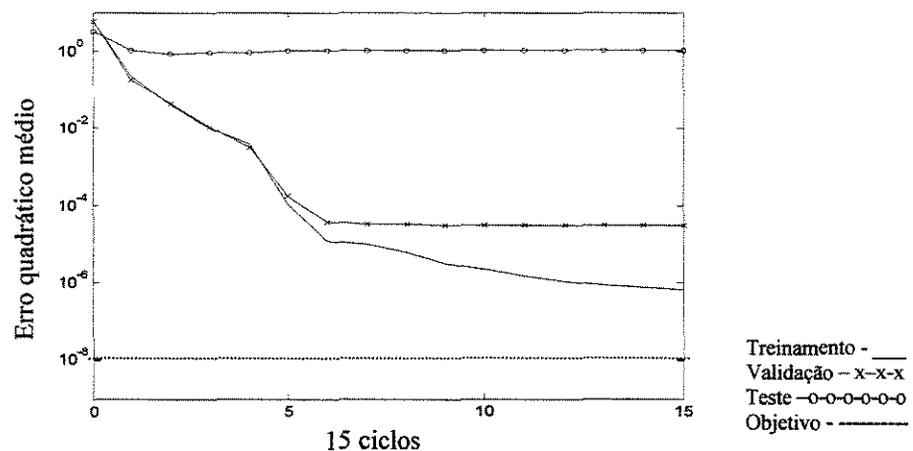


Figura 55 – Evolução Erro Quadrático Médio para os subconjuntos de treinamento, validação e teste.

Nas figuras 56, 57 e 58 temos o comportamento da conversão, seletividade e tempo de residência, respectivamente quando da aplicação da análise dos componentes principais (PCA) nos subconjuntos de teste, treinamento e validação com relação a cada variável de saída estudada.

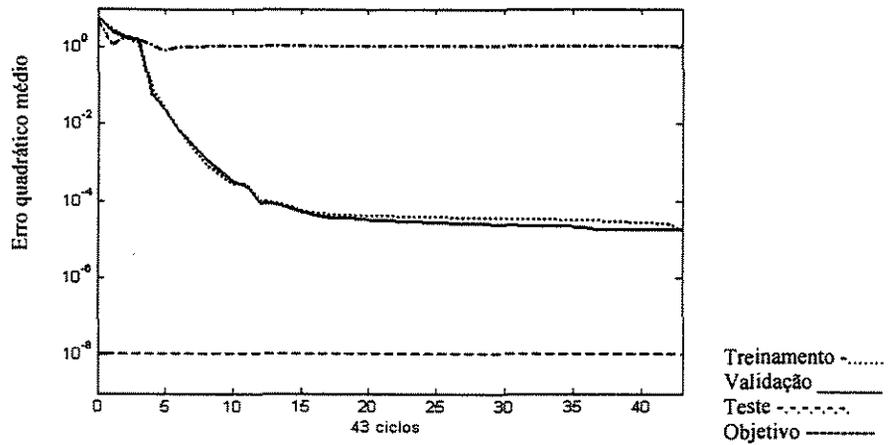


Figura 56 - Gráfico do Erro Quadrático Médio para a Conversão.

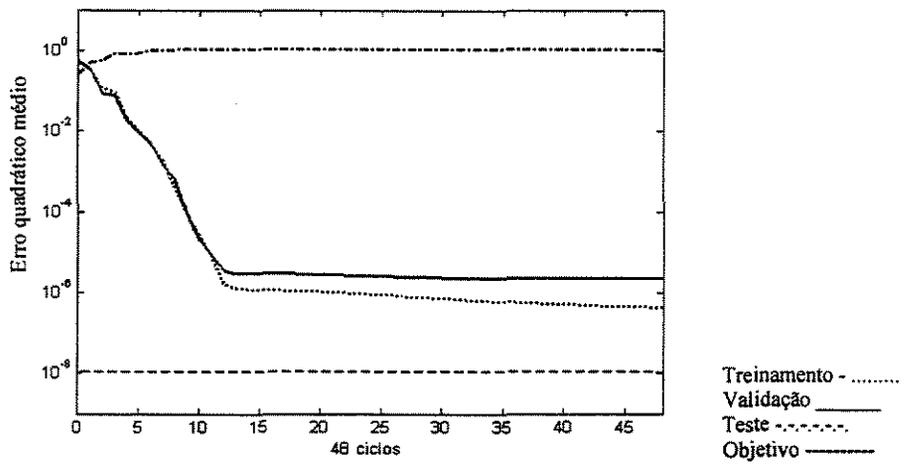


Figura 57 - Gráfico do Erro Quadrático Médio para a Seletividade.

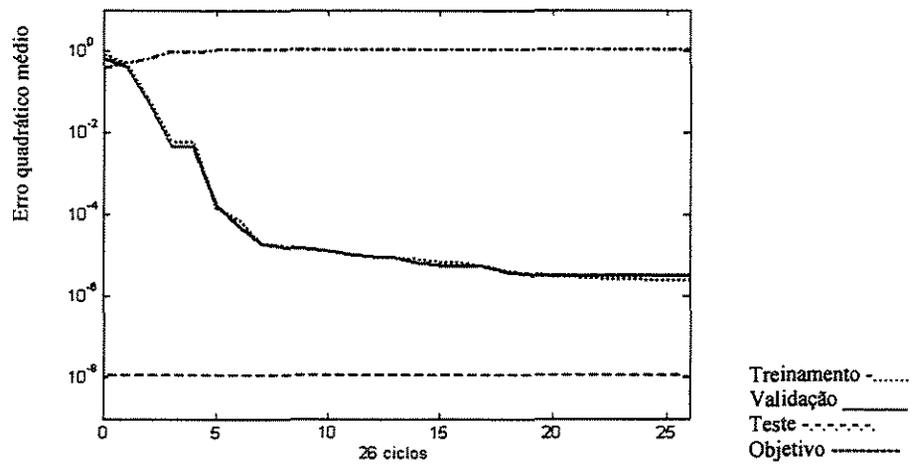


Figura 58 - Gráfico do Erro Quadrático Médio para o Tempo de residência.

Pelos gráficos apresentados anteriormente observa-se que a aplicação da análise de componentes principais otimizou o processamento das redes de predição da conversão, seletividade e tempo de residência, reduzindo o número de ciclos necessários ao treinamento, teste e validação da rede.

#### 4.4 – ANÁLISE DOS COMPONENTES PRINCIPAIS (PCA)

No quadro 8 é apresentado o programa em Matlab para o cálculo das componentes principais e os resultados obtidos na aplicação desta metodologia ao craqueamento térmico do 1,2-dicloroetano.

Quadro 8 – Programa de Análise dos Componentes Principais.

```
% Programa para realizar o PCA em um conjunto de dados de entrada
% 95 objetos e 7 variáveis de entrada
% calculo da matriz de dados centrada na media - Xc
n=95
load -ascii entrada.mat
X=entrada;
Xc=X-(ones(n,1)*mean(X));
% calculo da matriz de dados de entrada autoescalada - Xa
Xa=Xc./(ones(n,1)*std(X));
% calculo das médias e dos desvios padrão das matrizes Xc e Xa
mediaXc=mean(Xc);
mediaXa=mean(Xa);
dpXc=std(Xc);
dpXa=std(Xa);
% calculo das matrizes de covariância para os dois pré-processamentos:
C=Xc*Xc/(n-1);      % ou  C=cov(X)
R=Xa*Xa/(n-1);      % ou  R=corrcoef(X)
% calculo dos autovalores e autovetores da matriz de covariância do arquivo entrada.mat
[VC,LambdaC]=eig(C)
save LambdaC-VC.mat LambdaC VC
save -ascii LambdaC-VC.m LambdaC VC
% calculo dos autovalores e autovetores da matriz de correlação do arquivo entrada.mat
[VR,LambdaR]=eig(R)
save LambdaR-VR.mat LambdaR VR
save -ascii LambdaR-VR.m LambdaR VR
% criação dos vetores dos pesos - loadings nas componentes principais PC1 e PC2 -
Matriz C - Covariância.
v1C=VC(:,7)
v2C=VC(:,6)
% fazendo o gráfico de v2C contra v1C
%colordef white
plot(v1C,v2C,'k*')
% Nomes das variáveis para o arquivo X
vars=[' vme;' pce;' tce;'; tw1;' tw2;' tw3;' tw4']
text(v1C,v2C,vars)
title('Gráfico de v1C versus v2C - pesos')
xlabel('PC1')
ylabel('PC2')
saveas(gcf,'pesos C','fig')
% calculo dos escores nas duas primeiras componentes principais PC1 e PC2 na matriz X
t1C=Xa*v1C
t2C=Xa*v2C
plot(t1C,t2C,'k*')
exp = importdata('exp.xls')
text(t1C,t2C,exp)
title('Gráfico de t1C versus t2C - escores')
xlabel('t1C')
ylabel('t2C')
saveas(gcf,'escores C','fig')
```

```

% criação dos vetores dos pesos - loadings nas componentes principais PC1 e PC2 -
Matriz R - Coeficiente de correlação.
v1R=VR(:,7)
v2R=VR(:,6)
% fazendo o grafico de v1R contra v2R
%colordef white
plot(v1R,v2R,'k*')
text(v1R,v2R,vars)
title('Grafico de v1R versus v2R - pesos')
xlabel('PC1')
ylabel('PC2')
saveas(gcf,'pesos R','fig')

% calculo dos escores nas duas primeiras componentes principais PC1 e PC2 na matriz X
t1R=Xa*v1R
t2R=Xa*v2R
plot(t1R,t2R,'k*')
text(t1R,t2R,exp)
title('Grafico de t1R versus t2R - escores')
xlabel('t1R')
ylabel('t2R')
saveas(gcf,'escores R','fig')

% analise tridimensional - PC1, PC2 e PC3 - na matriz C
v3C=VC(:,5)
plot3(v1C,v2C,v3C,'k*')
text(v1C,v2C,v3C,vars)
title('Grafico de v1C vs v2C vs v3C - pesos')
xlabel('PC1')
ylabel('PC2')
zlabel('PC3')
saveas(gcf,'pesos tri C','fig')

% analise tridimensional - PC1, PC2 e PC3 - na matriz R
v3R=VR(:,5)
plot3(v1R,v2R,v3R,'k*')
text(v1R,v2R,v3R,vars)
title('Grafico de v1R vs v2R vs v3R - pesos')
xlabel('PC1')
ylabel('PC2')
zlabel('PC3')
saveas(gcf,'pesos tri R','fig')

```

Na tabela 11 são apresentados alguns dados de entrada utilizados na análise dos componentes principais. A matriz de dados de entrada sofre uma série de operações lineares da álgebra de forma a ser representada no espaço de máxima variância do conjunto de dados.

Os resultados do processamento da matriz de entrada são apresentados etapa por etapa na aplicação da análise dos componentes principais, mostrada a seguir:

Tabela 11 – Dados de entrada – Matriz  $X^2$ 

Vazão mássica (t/h)	Pressão de entrada (kgf/cm <sup>2</sup> )	Temperatura de convecção (°C)	TW <sub>1</sub> (°C)	TW <sub>2</sub> (°C)	TW <sub>3</sub> (°C)	TW <sub>4</sub> (°C)
24,67	23,33	240	370	510	552	557,5
27,76	23,33	240	370	510	552	557,5
30,84	23,33	240	370	510	552	557,5
33,93	23,33	240	370	510	552	557,5
37,01	23,33	240	370	510	552	557,5
24,67	23,33	264	407	561	607,2	613,2
27,76	23,33	264	407	561	607,2	613,2
30,84	23,33	264	407	561	607,2	613,2
33,93	23,33	264	407	561	607,2	613,2
37,01	23,33	264	407	561	607,2	613,2
24,67	23,33	252	388,5	535,5	579,6	585,4
27,76	23,33	252	388,5	535,5	579,6	585,4
30,84	23,33	252	388,5	535,5	579,6	585,4
33,93	23,33	252	388,5	535,5	579,6	585,4
37,01	23,33	252	388,5	535,5	579,6	585,4
24,67	23,33	228	351,5	484,5	524,4	529,6
27,76	23,33	228	351,5	484,5	524,4	529,6
30,84	23,33	228	351,5	484,5	524,4	529,6
33,93	23,33	228	351,5	484,5	524,4	529,6
37,01	23,33	228	351,5	484,5	524,4	529,6
33,93	23,33	216	333	459	496,8	501,8
⋮	⋮	⋮	⋮	⋮	⋮	⋮
30,84	25,663	240	370	510	552	557,5
33,93	25,663	240	370	510	552	557,5
37,01	25,663	240	370	510	552	557,5
24,67	25,663	252	388,5	535,5	579,6	585,4
27,76	20,997	264	407	561	607,2	613,2
30,84	20,997	264	407	561	607,2	613,2
33,93	20,997	264	407	561	607,2	613,2
37,01	20,997	264	407	561	607,2	613,2

Os resultados obtidos a partir da aplicação da análise dos componentes principais na matriz  $X$  são apresentados a seguir:

<sup>2</sup> Dados obtidos de Ferreira (1996).

**4.4.1 - Aplicação da Análise dos Componentes Principais à Matriz X**

Tabela 12. Autovalores da matriz das covariâncias - LambdaC

$\lambda_{c7}$	$\lambda_{c6}$	$\lambda_{c5}$	$\lambda_{c4}$	$\lambda_{c3}$	$\lambda_{c2}$	$\lambda_{c1}$
<b>0,00000</b>	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	<b>0,00000</b>	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	<b>0,00000</b>	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	<b>0,00005</b>	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	<b>3,40897</b>	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,00000	<b>19,22697</b>	0,00000
0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	<b>11803,66600</b>
				↓	↓	↓
				PC3	PC2	PC1

Os elementos da diagonal principal são os autovalores da matriz das covariâncias. Deve-se ordená-las do maior autovalor para o menor, assim tem-se a redução da dimensionalidade do problema, no caso de 7 (sete) dimensões para 3 (três). O sistema será representado então neste caso por 3 (três) componentes principais.

A matriz dos autovetores associada é dada por:

VC7	VC6	VC5	VC4	VC3	VC2	VC1
0,00000	0,00000	0,00000	0,00000	0,00000	1,00000	0,00000
0,00000	0,00000	0,00000	0,00064	-1,00000	0,00000	0,00060
0,16408	0,72072	-0,61464	0,14844	0,00023	0,00000	0,23200
0,68985	-0,53114	-0,24839	0,22884	0,00036	0,00000	0,35767
0,16445	0,34765	0,71383	0,31543	<b>0,00050</b>	0,00000	0,49301
-0,68567	-0,27854	-0,22579	0,34140	0,00054	<b>0,00000</b>	0,53361
0,00000	0,00000	0,00000	-0,84235	-0,00022	0,00000	<b>0,53893</b>
				↓	↓	↓
				L3	L2	L1

Onde os elementos da diagonal principal são os pesos, "loadings" ou autovetores principais, sendo que a numeração dos autovetores é fornecida do maior autovalor para o menor.

Matriz dos autovalores dos coeficientes de correlação - LambdaR

$\lambda_{r7}$	$\lambda_{r6}$	$\lambda_{r5}$	$\lambda_{r4}$	$\lambda_{r3}$	$\lambda_{r2}$	$\lambda_{r1}$
0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	<b>0,99844</b>	0,00000	0,00000
0,00000	0,00000	0,00000	0,00000	0,00000	<b>1,00000</b>	0,00000
0,00000	0,00000	0,00000	0,00000	0,00000	0,00000	<b>5,00156</b>
				↓	↓	↓
				PC3	PC2	PC1

Os elementos da diagonal principal são os autovalores da matriz dos coeficientes de correlação. Eles devem ser ordenados do maior autovalor para o menor, assim tem-se a redução da dimensionalidade do problema, no caso de 7 (sete) dimensões para 3 (três). O sistema será representado então neste caso por 3 (três) componentes principais.

Grau de representatividade de cada componente:

$$PC1 \% = \frac{\lambda_1}{\sum_{i=1}^p \lambda_i} = \frac{5,0156}{5,00156 + 1,0000 + 0,99844} = 71,45$$

$$PC2 \% = \frac{\lambda_2}{\sum_{i=1}^p \lambda_i} = \frac{1,0000}{5,00156 + 1,0000 + 0,99844} = 14,28$$

$$PC3 \% = \frac{\lambda_3}{\sum_{i=1}^p \lambda_i} = \frac{0,99844}{5,00156 + 1,0000 + 0,99844} = 14,26$$

A matriz dos autovetores associada é dada por:

V7R	V6R	V5R	V4R	V3R	V2R	V1R
0.00000	0.00000	0.00000	0.00000	0.00000	1.00000	0.00000
0.00000	0.00000	0.00000	0.00002	-0.99980	0.00000	0.01975
-0.29766	0.44968	0.67763	0.22361	0.00884	0.00000	0.44713
-0.35238	-0.79109	0.00125	0.22361	0.00884	0.00000	0.44713
-0.21162	0.40912	-0.73338	0.22361	<b>0.00884</b>	0.00000	0.44713
0.86165	-0.06770	0.05449	0.22361	0.00884	<b>0.00000</b>	0.44713
0.00000	0.00000	0.00000	-0.89443	0.00881	0.00000	<b>0.44713</b>
				↓	↓	↓
				L3	L2	L1

Os elementos da diagonal principal são os pesos, “loadings” ou autovetores principais, sendo que a numeração dos autovetores é fornecida do maior autovalor para o menor, de acordo com o grau de significância.

Na figura 59 são apresentados os resultados correspondentes a aplicação da análise das componentes principais (PCA) na matriz das covariâncias. Cada variável é representada nos eixos da primeira e segunda componentes principal, segundo seu grau de significância.

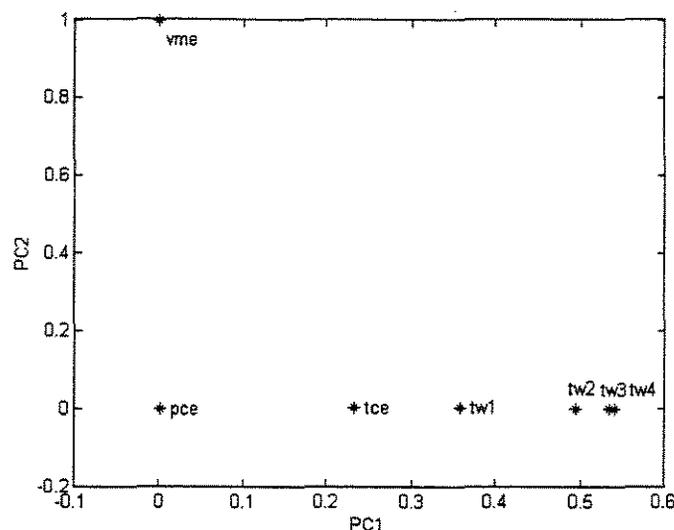


Figura 59 - Gráfico de v1C versus v2C – pesos nas PC1 e PC2.

vme:	Vazão mássica da carga na entrada (t/h);
pce:	Pressão da carga na entrada (kgf/cm <sup>2</sup> );
tce:	Temperatura de convecção na entrada (°C);
tw1:	Temperatura de parede no ponto 1 (°C);
tw2:	Temperatura de parede no ponto 2 (°C);
tw3:	Temperatura de parede no ponto 3 (°C);
tw4:	Temperatura de parede no ponto 4 (°C);

A figura 59 mostra que de todas as variáveis de entrada envolvidas no processo, as que têm maior influência na PC1 - componente principal 1, por ordem crescente de importância são pce e vme (igualmente), tce, tw1, tw2, tw3, e tw4. Ou seja, as variáveis tw3 e tw4 são as que tem maior influência na representação do processo em estudo.

A vazão mássica na entrada, é a variável que mais influencia na PC2 – componente principal 2. As outras variáveis têm pouca e igual influência nesta componente.

É interessante observar que a temperatura de parede e a vazão mássica da carga na entrada têm grande influência na conversão da reação e que, portanto os experimentos que tiverem maior PC1 e PC2 terão provavelmente maiores resultados de conversão, sendo que a influência da temperatura é exponencial e a da vazão mássica é linear.

A seguir tem-se a representação gráfica dos escores que estão relacionados aos objetos, que no caso são experimentos realizados.

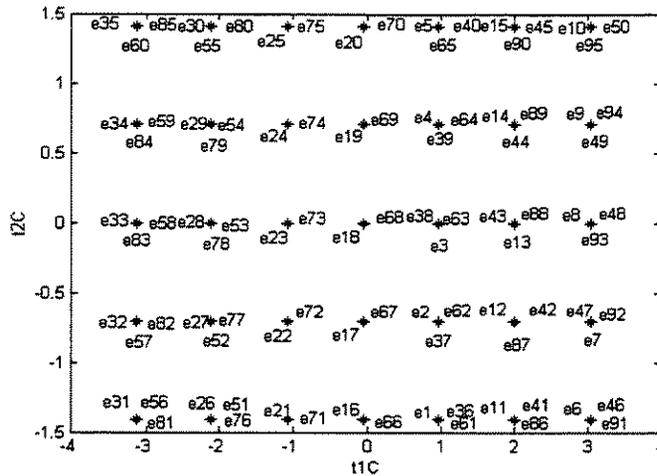


Figura 60 - Gráfico dos escores nas componentes principais PC1 e PC2.

Para tirar conclusões sobre as figuras 59 e 60, deve-se analisá-los paralelamente:

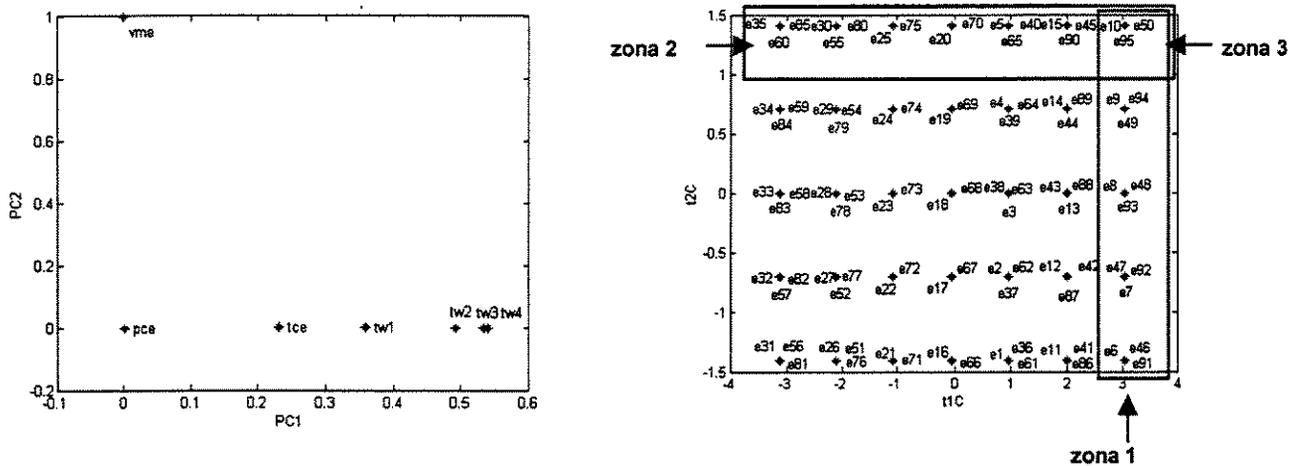


Figura 61 - Gráfico dos pesos e escores nas componentes PC1 e PC2.

Analisando a figura 61, observa-se que os experimentos delimitados pela zona 1 possuem maior influência da temperatura de parede, sendo experimentos caracterizados por uma grande PC1. Já os experimentos delimitados pela zona 2 possuem uma maior influência da vazão mássica da corrente da alimentação na entrada, sendo experimentos caracterizados por uma grande PC2.

Os experimentos delimitados pela zona 3 – e10, e50 e e95 possuem uma alta PC1 e PC2, ou seja, são influenciados significativamente pela temperatura de parede e vazão mássica da corrente de alimentação.

Realizando a mesma análise para os pesos e escores com relação à matriz dos coeficientes de correlação têm-se:

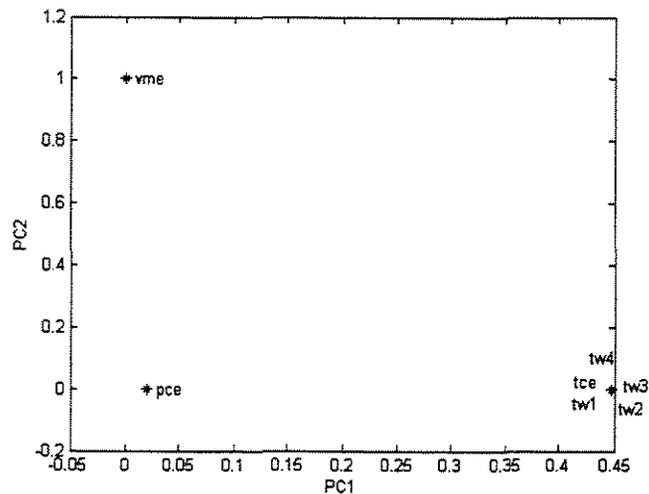


Figura 62 - Pesos nas componentes PC1 e PC2, com relação à matriz R.

Observa-se que no caso da matriz dos coeficientes de correlação, as variáveis relativas à temperatura tiveram o mesmo peso tanto na PC1 quanto na PC2. A análise com relação à matriz dos coeficientes de correlação (R), consegue agrupar mais o conjunto de dados. Isto se deve ao fato que variáveis de entrada com mesma unidade são agrupadas pela matriz dos coeficientes de correlação durante a etapa do auto-escalamento da mesma. Já a matriz da covariância não realiza este agrupamento porque os dados podem ter unidades diferentes sendo submetidos apenas à etapa de centrar os mesmos nas respectivas médias. Analisando o gráfico dos escores e dos pesos em relação à matriz dos coeficientes de correlação, apresentados na figura 63, tem-se:

Similarmente à análise da matriz das covariâncias, os experimentos que estão na zona 3 são os que apresentam maior PC1 e PC2, sendo um indicativo de que serão obtidas altas conversões para estes testes.

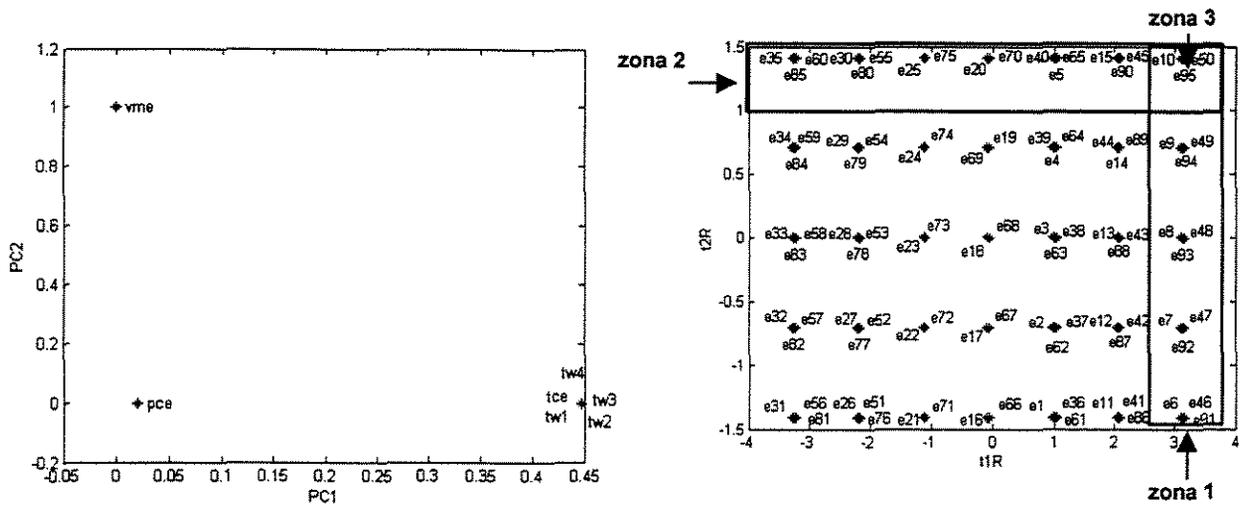


Figura 63 - Escores e pesos nas componentes PC1 e PC2 com relação a matriz R.

Na figura 64 tem-se a representação dos pesos nas três componentes principais, segundo a abordagem da matriz das covariâncias.

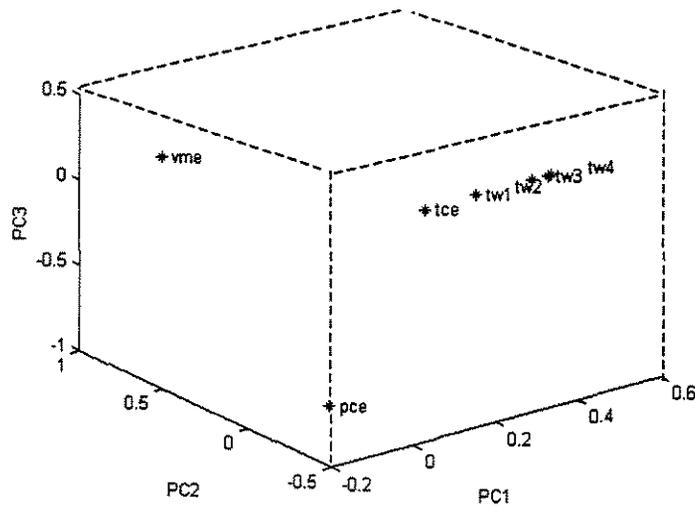


Figura 64 – Pesos nas 3 (três) componentes principais com relação a matriz C.

O grau de importância das variáveis nas componentes principais pela interpretação do gráfico é dado a seguir:

Tabela 12. Grau de significância das variáveis pela PCA

Componente Principal	Grau de importância das variáveis
PC1	tw4>tw3>tw2>tw1>tce>vme>pce
PC2	vme>pce=tce=tw1=tw2=tw3=tw4
PC3	vme>tce=tw1=tw2=tw3=tw4>pce

Analogamente o gráfico dos pesos em R fica:

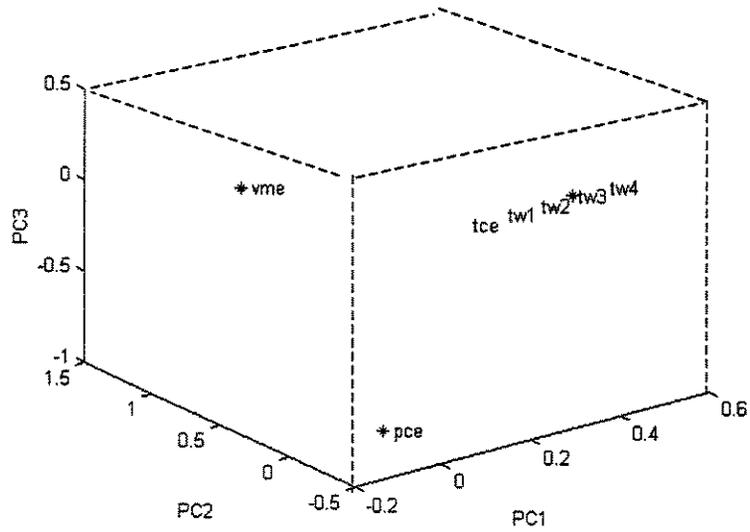


Figura 65 – Pesos nas 3 (três) PC's em relação a matriz dos coeficientes R.

## 4.5 – REGRESSÃO EM COMPONENTES PRINCIPAIS (PCR)

Com o objetivo de modelar a conversão da reação de craqueamento térmico do 1,2-dicloroetano através de uma abordagem empírica, aplicou-se a metodologia de Regressão em Componentes Principais. Foram utilizadas as variáveis de entrada usuais e conversão como variável de saída. A regressão em componentes principais é realizada em duas etapas. A primeira é a de construção do modelo e a segunda é a de utilização do mesmo. Os resultados destas etapas são apresentados a seguir:

### 4.5.1 - Construção do Modelo

A seguir são apresentados os resultados obtidos na construção do modelo

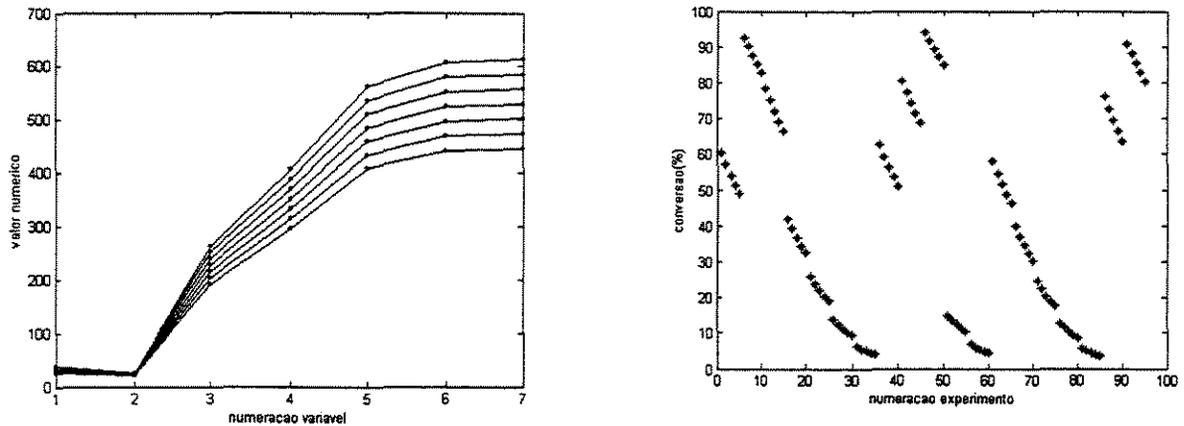


Figura 66 – Resultados da Construção do Modelo de Regressão.

Na figura 67 são representados os autovalores correspondentes ao grau de significância das componentes principais

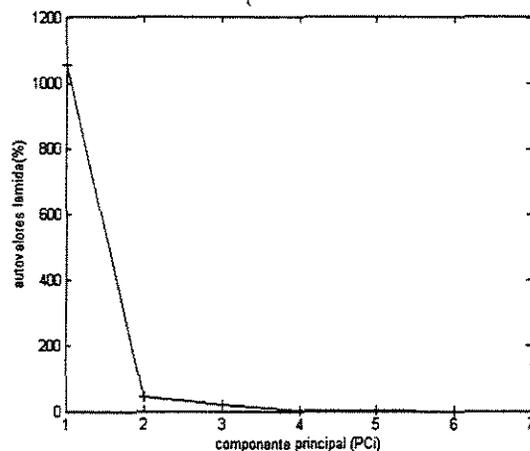


Figura 67 – Gráfico do Scree das Componentes Principais.

Através da análise do gráfico observa-se que três das componentes principais influenciam mais significativamente na correlação das variáveis de entrada, a saber PC1 com  $\lambda_1$ , PC2 com  $\lambda_2$  e PC3 com  $\lambda_3$ . Os outros valores não afetam significativamente a correlação entre as variáveis de entrada, dentro do grau de tolerância previamente estabelecido.

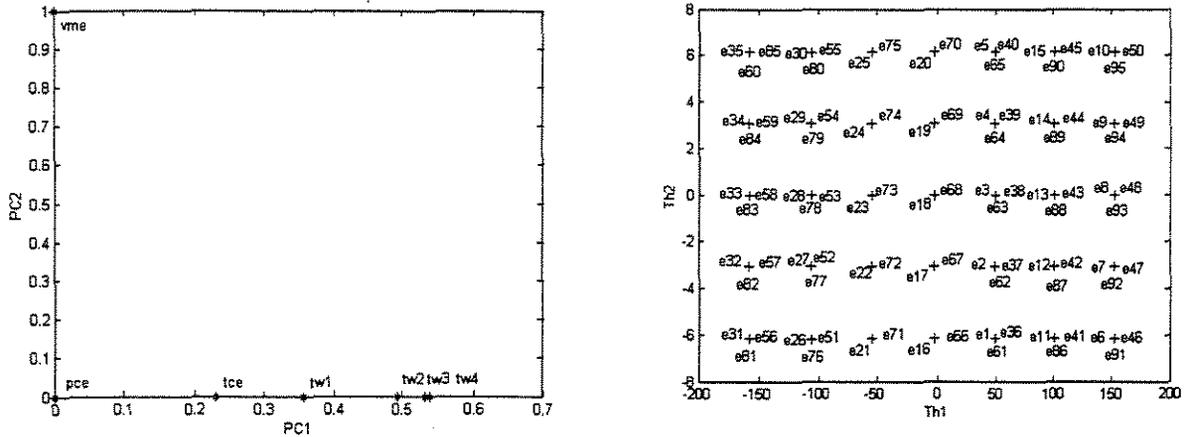


Figura 68 – Resultados da Construção do Modelo PCR

#### 4.5.2 - Utilização do Modelo

Na figura 69 têm-se os resultados obtidos na utilização do modelo para previsão da conversão da reação de craqueamento do 1,2-dicloroetano:

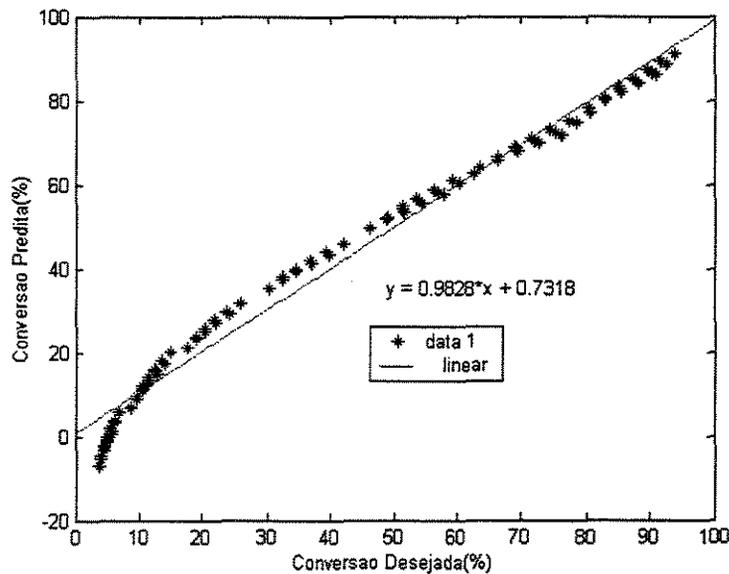


Figura 69 – Comparação conversão predita pelo PCR e conversão desejada.

Através da figura 69, observa-se que a relação entre a conversão predita e desejada não é puramente linear. Os resultados são melhores para médias e altas conversões. Para conversões abaixo de 10% os resultados são menores que os desejados.

No quadro 9 é apresentado o programa em Matlab para o desenvolvimento da Metodologia de Regressão por Componentes Principais:

Quadro 9 – Programa para Regressão em Componentes Principais.

```
% Regressao por Componentes Principais (PCR)
% Usar uma regressão em componente principais, com um modelo de três
componentes
% para prever a conversão do EDC de um conjunto de testes do arquivo
entrada.mat
% Gráficos dos dados do conjunto de treinamento:
entradatr=entrada
plot(entradatr,'k.-')
title('Gráfico das variáveis de entrada de treinamento')
xlabel('numeracao variavel')
ylabel('valor numerico')
saveas(gcf,'entrada treinamento','fig')
conv = importdata('conv.xls')
convtr=conv
plot(convtr,'k*')
title('Gráfico da conversao de treinamento')
xlabel('numeracao experimento')
ylabel('conversao(%)')
saveas(gcf,'conversao treinamento','fig')

%Construção do Modelo
% pré-processamento do conjunto de treinamento - centrando na média
E0tr=entradatr-(ones(95,1)*mean(entradatr));
f0tr=convtr-(ones(95,1)*mean(convtr));
plot(f0tr,'b*')

% Cálculo das componentes principais:
[U,S,V]=svd(E0tr);
% Número de componentes no modelo:
plot(diag(S), 'k+-') %(gráfico do scree)
title('Gráfico do Scree')
xlabel('componente principal (PCi)')
ylabel('autovalores lamida(%)')
saveas(gcf,'scree','fig')
Vh=V(:,1:3) % (matriz dos pesos)
Th=E0tr*Vh %(matriz dos escores)
% Gráficos do escores e dos pesos em duas componentes principais:
plot(Vh(:,1),Vh(:,2),'k*') % (pesos)
text(Vh(:,1),Vh(:,2),vars)
title('Gráfico de vh1 versus vh2 - pesos')
xlabel('PC1')
ylabel('PC2')
```

```

plot(Th(:,1),Th(:,2),'k+') % (escores)
text(Th(:,1),Th(:,2),exp)
title('Grafico de Th1 versus Th2 - escores')
xlabel('Th1')
ylabel('Th2')

% Regressão:

b=inv(Th*Th)*Th*f0tr; %(coeficientes)
f0ptr=Th*b; %(previsões no conjunto de treinamento)
plot(f0tr,f0ptr,'k+') % (gráficos das previsões contra os valores verdadeiros)
title('Grafico de Comparação entre os Valores Preditos pelo PCR e os valores reais')
xlabel('Conversao Desejada(%)')
ylabel('Conversao Predita(%)')
saveas(gcf,'comparacao predito desejado','fig')

% Utilização do modelo
% Carregando a matriz de dados de entrada de teste - entradatt - conjunto de testes
Xtt=importdata('entradatt.xls')
% Pré-processamento do conjunto de teste:
E0tt=Xtt-(ones(95,1)*mean(entradatr));
% Cálculo dos escores:
Tu=E0tt*Vh
% Estimativa das respostas:
f0u=Tu*b
% Conversão às unidades de partida:
% Carregando o vetor de saída - convtt
convtt=importdata('convtt.xls')
y0u=f0u+mean(convtr)
plot(convtt, y0u,'k*') %(gráfico)
title('Grafico de Comparação entre os Valores Preditos pelo PCR e os valores reais')
xlabel('Conversao Desejada(%)')
ylabel('Conversao Predita(%)')
rmseptt=sqrt((y0u-convtt)*(y0u-convtt)/94) %(erro médio de previsão no conjunto de teste)

```

Através da utilização da análise de componentes principais verificou-se que ela é uma metodologia importante na definição das variáveis mais significativas no sistema em estudo e conseqüente redução da dimensionalidade do problema. Já a regressão por componentes principais só se apresenta como adequada para pequenos intervalos da conversão - variável de saída utilizada neste estudo. Apesar de alguns valores preditos serem bem próximos dos valores desejados, especialmente em médias e altas conversões, observou-se que esta metodologia fez com que os valores de baixa conversão sejam preditos erroneamente.

## 5.1 – CONCLUSÕES

A modelagem e simulação de processos químicos se constituem num desafio para a implementação de melhorias nos processos, que nem sempre podem ser utilizados de forma experimental para verificação das suas respostas em relação as variáveis envolvidas nos mesmos.

Neste estudo o processo de craqueamento térmico do 1,2-dicloroetano foi representado por um modelo matemático dinâmico que considera os princípios da conservação da massa e energia na serpentina, onde ocorre a reação. As equações obtidas foram resolvidas numericamente pelos métodos das diferenças finitas e colocação ortogonal, sendo que o método de Runge-Kutta foi utilizado para as equações diferenciais ordinárias no tempo.

Dentre os métodos utilizados para resolver numericamente as equações diferenciais parciais, o que apresentou melhores resultados foi o método das diferenças finitas. Os resultados obtidos foram compatíveis com os esperados teoricamente apresentados por Ferreira (1996).

Foi realizada uma variação no número de pontos na malha de 10 a 100 e observou-se que com 50 e 100 pontos na malha os resultados obtidos foram bem próximos, sugerindo-se que apenas 50 pontos eram suficientes para simular o processo no grau de exatidão desejado.

No método da colocação ortogonal foi obtido um perfil de concentração dos produtos e subprodutos diferente do obtido no método das diferenças finitas, apesar do perfil de temperatura estar coerente com o esperado. Concluímos então que pelo fato do reator ser muito grande, o método da colocação ortogonal não ofereceu bons resultados. Sugere-se então aplicar o método da colocação ortogonal em elementos finitos.

Através dos resultados da simulação foi possível verificar alguns comportamentos teoricamente esperados no processo, tais como:

- ✓ Acompanhamento da evolução do processo desde o estado transiente até o estado estacionário, visualizando a estabilização dos perfis de concentração dos componentes e perfil de temperatura do processo.

- ✓ Verificou-se que o estabelecimento do perfil de temperatura de processo é relativamente rápido não se constituindo em etapa limitante para o craqueamento térmico do 1,2-dicloroetano.
- ✓ A temperatura de processo influencia significativamente no tempo de residência da mistura gasosa na serpentina. Através dos resultados obtidos foi possível visualizar a evolução do tempo de residência desde o estado transiente até o estado estacionário, onde havia uma estabilização do mesmo.
- ✓ A conversão da reação e seletividade a monocloreto de vinila evoluem rapidamente para o valor de estado estacionário, ao final da serpentina.
- ✓ A turbulência do escoamento, verificada pelos elevados valores de número de Reynolds, facilita os processos de transferência de quantidade de movimento, calor e massa, o que é verificado pelo pequenos intervalos de tempo necessários para a estabilização dos perfis de concentração e temperatura de processo.

Já na aplicação da simulação não convencional através das Redes Neurais Artificiais, foram obtidos bons resultados para a predição da conversão, seletividade e tempo de residência da reação, uma vez que os valores preditos pela rede foram compatíveis com os valores desejados, com coeficientes de correlação próximos a um, na reta diagonal. Foi utilizado o algoritmo de retroalimentação dos erros para ajuste dos pesos e limiares, sendo que o método mais eficiente no treinamento da rede foi o de Levenberg-Marquadt. A análise após regressão apresentou ótimos resultados, sendo que a predição da conversão foi a que apresentou os melhores resultados.

Verificou-se que o pré-processamento das Redes Neurais Artificiais influenciou significativamente no treinamento da rede. A depender do tipo de processamento utilizado, os resultados preditos pela rede apresentavam um melhor coeficiente de correlação com os resultados desejados.

Na predição da conversão ambos os tipos de pré-processamento estudados [0,1] e [-1, +1] conduziram a bons resultados, entretanto a normalização de zero a um necessitou de menor número de ciclos de treinamento. Foi realizado

um estudo sobre a influência das funções de ativação utilizadas no erro médio quadrático no treinamento da rede. Os menores erros quadráticos médios foram obtidos utilizando-se as funções “logsig” na camada oculta e “purelin” na camada de saída.

Quanto ao número de neurônios utilizados no processamento da rede, verificou-se que muitos neurônios geralmente conduzem a um menor erro quadrático médio, entretanto aumentam o tempo de processamento da rede. Sete neurônios foram suficientes para realizar o processamento da rede de predição da conversão com erro quadrático médio e tempo de processamento satisfatório.

Na análise do efeito da taxa de aprendizado no treinamento da rede, verificou-se que altas taxas de aprendizado geralmente promovem uma maior dispersão entre os dados preditos pela rede e os desejados, isto é, geralmente há uma redução no coeficiente de correlação entre estes dados. O aumento na taxa de aprendizado geralmente aumenta o tempo de processamento da rede, fazendo com que haja uma maior variação nos pesos sinápticos, limiars e gradientes. Isto pode conduzir a resultados mais discrepantes em relação aos valores desejados. Taxas de aprendizado de 0.1 a 0.5 foram adequadas para a predição da conversão, seletividade e tempo de residência do processo em estudo.

Na predição da seletividade a monocloreto de vinila observou-se que o pré-processamento que forneceu melhores resultados foi normalizando os dados de zero a um e utilizando-se as funções de ativação logsig na camada oculta e purelin na camada de saída. A taxa de aprendizagem de 0.1 foi satisfatória para o treinamento da rede com erro médio quadrático de  $1 \text{ E-}8$ , necessitando em torno de 700 ciclos de treinamento.

Na predição do tempo de residência obteve-se um bom coeficiente de correlação ( $R=1$ ) entre os resultados desejados e preditos pela rede. Utilizou-se uma normalização linear de zero a um e funções de ativação “logsig” na camada oculta e “purelin” na camada de saída. O critério de erro médio quadrático mínimo foi atingido em aproximadamente 400 (quatrocentos) ciclos de treinamento.

A análise dos Componentes Principais (PCA) conduziu a um aceleração significativo no treinamento da rede reduzindo a dimensionalidade do sistema a

dois componentes principais dentre os quais a temperatura de parede e a vazão mássica da carga tinham efeito mais significativo.

A conversão da reação foi modelada através da Regressão por Componentes Principais (PCR) e observamos que esta modelagem conduziu a bons resultados apenas para médias e altas conversões. Os resultados encontrados para baixas conversões foram insatisfatórios.

## 5.2 - SUGESTÕES PARA TRABALHOS FUTUROS

Em relação aos cálculos das propriedades da mistura gasosa:

1. Corrigir as correlações de cálculo de propriedades da mistura gasosa e equação de estado considerando a mistura gasosa como um gás não ideal e comparar os resultados obtidos pela simplificação de gás ideal.

Em relação ao modelo fenomenológico desenvolvido, temos as seguintes sugestões:

1. Incorporar ao modelo atual o modelo da zona de convecção da fornalha onde ocorre a vaporização do 1,2-dicloroetano.
2. Incorporar ao modelo atual um modelo de deposição de coque de forma que possam ser avaliadas quais as seções críticas de deposição de coque ao longo da serpentina.
3. Incorporar ao modelo o balanço de energia na fornalha, onde ocorre a queima do gás natural, de forma a se obter o perfil de temperatura de parede ao longo da serpentina, uma vez que neste trabalho o perfil de temperatura era pré-estabelecido.
4. Incorporar ao modelo atual a equação do balanço de energia na direção radial da serpentina com o objetivo de avaliar se a variação ao longo do raio da serpentina é realmente significativa ou não.
5. Incorporar ao modelo atual a equação de balanço de momentum a fim de verificar a influência da perda de carga nos resultados obtidos.

Em relação aos métodos matemáticos utilizados, têm-se as seguintes sugestões:

1. Aplicar o método da colocação ortogonal em elementos finitos de modo a avaliar se este método fornece melhores resultados que o método da colocação ortogonal utilizado no presente trabalho. Neste método divide-se a serpentina em elementos finitos e o método da colocação é aplicado em cada elemento, sendo que as condições de saída de um elemento finito são utilizadas como condições de entrada do próximo elemento finito.
2. Utilizar no perfil de temperatura de parede uma interpolação spline e comparar os resultados obtidos com a interpolação linear utilizada.

Em relação à aplicação da Metodologia de Redes Neurais Artificiais:

1. Aplicar as Redes Neurais Artificiais (RNA's) de mapeamento dinâmico às variáveis de processos de forma a obter relações entrada/saída em uma simulação dinâmica.
2. Desenvolver uma estratégia de controle para a temperatura de processo com o controlador sendo baseado em Redes Neurais Artificiais.
3. Desenvolver uma função de ativação da rede que incorpore a taxa de aprendizado como parâmetro da função e verificar se os resultados apresentados são melhores do que pela utilização das funções de ativação usuais.

Em relação à análise multivariada dos dados – Análise dos Componentes Principais (PCA) e Regressão por Componentes Principais (PCR):

1. Aplicar a Análise não Linear de Componentes Principais (NPCA) no sistema em estudo e comparar com os resultados obtidos na Análise por Componentes Principais (PCA), visto que o processo é não linear e isto pode influenciar significativamente na aceleração do treinamento da rede neural implementada.

2. Aplicar a Regressão por Componentes Principais (PCR) em faixas de baixa conversão, média e alta conversão, de forma a obter melhores previsões da conversão nas faixas estudadas.
3. Aplicar a regressão por Mínimos Quadrados Parciais (PLS) na modelagem para a previsão da conversão da reação e comparar com os resultados obtidos na Regressão por Componentes Principais (PCR).
4. Aplicar a regressão por Mínimos Quadrados Parciais (PLS) na previsão de somente uma variável (PLS1) e de duas variáveis (PLS2) e comparar os resultados obtidos.

**REFERÊNCIAS BIBLIOGRÁFICAS**

- ALBRIGHT, L. F.; "Manufacture of Vinyl Chloride"; Chemical Engineering, abril 10, 219-25, 1967.
- AZEVEDO, VALÉRIA LYRIO DE CASTRO; "Modelagem Dinâmica e Simulação de uma Fornalha utilizada no Craqueamento Térmico do 1,2-Dicloroetano" - Tese de Mestrado – UFBA – Mestrado em Engenharia Química – Outubro de 1998.
- BAGHAT, P. "An Introduction to Neural Nets". Chemical Engineering Progress, v. 45, 55-60, August, 1990.
- BARENDREGT, S.; DENTE, M.; RANZI, E.; DUIM, F.; "New Program Optimizes Cracking Furnaces"; Oil & Gas Journal Report, April 6, 90-4, 1981.
- BARROSO, L. C.; Barroso, M. M. de A; Campos Filho, F. F.; Carvalho, M. L. B. de; Maia, M. L., Cálculo Numérico (com aplicações), 2ª Edição. Editora Harbra (1987)
- BASF; "Procédé de Préparation du Chlorure de Vinyle"; Patente FR 2.246.525, outubro 1, 1974.
- CARNAHAN, B., LUTHER, H.A., WILKES, J.O., "Applied Numerical Methods", John Wiley & Sons, New York, 1969.
- CHAN, W. M., NASCIMENTO, C. A. O. "Modelagem de polimerização de olefinas em reator tubular de alta pressão através de rede neural". In: Congresso Brasileiro de Engenharia Química, 10, 1994, São Paulo. Anais... São Paulo, Associação Brasileira de Engenharia Química, v. 1, 623-628, 1994,.
- CITRA, S. P. "Neural Net Applications in Chemical Engineering". AI Expert, 20-25, November 1992.
- CITRA, S. P. "Use Neural Networks for Problems Solving". Chemical Engineering Progress, v. 89, n. 4, 44-55, April 1993.
- CONSTANTINIDES, ALKIS; "Applied Numerical Methods with Personal Computers", McGraw-Hill Inc., 1987.
- CYBENKO, G. "Continuous Valued Neural Networks with Two Hidden Layers Are Sufficient". Department of Computer Science, Tufts University, Medford, Ma (Technical Report), 1988.

DEMUTH, H., BEALE, M., "Neural Network Toolbox For Use with MATLAB<sup>®</sup>", The Math Works, Inc., 1996.

DENTE, M.; RANZI, E.; GOSENS, A. G.; "Detailed Prediction of Olefin Yields from Hydrocarbon Pyrolysis Through a Fundamental Simulation Model (SPYRO)"; Computers & Chemical Engineering, 3, 61-75, 1979.

DIRION, J. L., "Elaboration of neural network system for semi-batch reactor temperature control: an experimental study". Chemical Engineering and Processing, v. 35, 225-234, 1996.

EDGAR, T.F.; HIMMELBLAU, D.M.; Optimization of Chemical Processes, McGraw-Hill, 1988.

FERREIRA, HELIANILDES SILVA; "Modelagem e Simulação do Craqueamento Térmico do 1,2-Dicloroetano" - Tese de Mestrado – UFBA – Mestrado em Engenharia Química - fevereiro de 1996.

FILETI, ANA MARIA FRATTINI; "Controle em Destilação Batelada – Controle Adaptativo e Controle Preditivo com Modelo baseado em Redes Neurais Artificiais", Unicamp, Campinas, outubro, 1995.

FILHO, EDSON FÉLIX MENDES; Curso Introdutório sobre Redes Neurais – [http://elson\\_mendes.sites.uol.com.br/rn/rn\\_intro.html](http://elson_mendes.sites.uol.com.br/rn/rn_intro.html).

FINLAYSON, B.A, "Non Linear Analysis in Chemical Engineering" McGraw-Hill Book Co., New York, 1980.

FROMENT G. F.; "Kinetics and Reactor Design in the Thermal Cracking for Olefins Production"; Chemical Engineering Science, vol. 47, no. 9-11, 2163-2177, 1992.

GAO, F., WANG, F., LI, M., "A simple nonlinear controller with diagonal recurrent neural network". Chemical Engineering Science, v. 55, 1283-1288, 2000.

GELADI, P., E B. R. KOWALSKI, "Partial Least-Squares Regression: A Tutorial, "Analytica Chimica Acta 185, pp. 1-17, 1986.

HAHN, G. J. E W. Q. MEEKER. STATISTICAL Intervals: a Guide for Practitioners, Wiley Inter-Science, 1991.

HAYKIN, S., Neural Networks, Macmillan College Publishing Company, Inc., New York, 1994.

HEBB, D., – The Organization of Behavior – Wiley, New York, 1949.

HENSON, MICHAEL A., SEBORG, DALE E., Non-Linear Process Control, Prentice Hall, 1997.

HEYNDERICKX, G. J., CORNELIS, G. G.; FROMENT, G. F.; “ Circunferential Tube Skin Temperature Profiles in Thermal Cracking Coils”; AIChE Journal, 38, no. 12, 1905-12, dezembro 1992.

HIGGINS, W. A.; Technical Report, Know How Meeting Exchange, 3314-69, B. F. Goodrich, 1969.

HOPPFIELD, J. J. – “Neural Networks and Physical Systems with Emergent Collective Computational Abilities” – Proceedings of the National Academy of Sciences, vol. 79, 1982.

JACKSON, J. E., A User's Guide to Principal Components, John Wiley & Sons Inc, 1991.

JOHNSON, R. A. e D. W. Wichern, Applied Multivariate Statistical Analysis, Third Edition, Prentice Hall, 1992.

KRESTA J. V., J. F. MACGREGOR E T.E. MARLIN, “Multivariate Statistical Monitoring of Process Operating Performance”; The Canadian Journal Chem. Eng. , 69, pp. 35-47, 1991.

KÓVACS, ZSOLT L., Redes Neurais Artificiais, Fundamentos e Aplicações, Collegium Cognitio, 2ª. Edição, São Paulo, 1996.

KREKELER H.; Königstein, Germany, Farbwerke Hoechst Aktiengesellschaft, Patente US 2.724.006, junho 9, 1954.

LAPIDUS, L. "Digital Computation for Chemical Engineers", McGraw-Hill Book Co, New York, 1962.

LEE, J. K., KANG, K. H. E SONG, K. H., “Increase VCM output”, Hydrocarbon Processing, 63-66, June 2001.

- LEIB, T. M., MILLS, P. L., LEROU, J. J. Fast response distributed parameter fluidized bed reactor model for propylene partial oxidation using feed-forward neural network methods. *Chemical Engineering Science*, v. 51, n. 10, 2189-2198, 1996.
- LOPES, M. F. S., *Modelagem Matemática e Simulação de Reatores Catalíticos: Aplicação para a Reação de Oxidação do Orto-Xileno*, Tese de Doutorado, UNICAMP, São Paulo, 1994.
- MACGREGOR J. F., C. KIPARISSIDES E M. KOUTOUDI, "Process Monitoring and Diagnosis by Multi-block PLS Methods", *AIChE Journal*, 40, 5, pp 826-838, 1994.
- MARLET, J. M., LOPES FILHO, D. F., NASCIMENTO, C. A. O. "Comparação das abordagens de modelagem via redes neurais e estatística na análise de dados do processo de polimerização da acrilonitrila". In: *Congresso Brasileiro de Engenharia Química*, 10, 1994, São Paulo: Associação Brasileira de Engenharia Química, v. 1, 629-633, 1994.
- MATSUMOTO, ÉLIA YATHIE, *MATLAB 6.5 – Fundamentos de Programação*, Editora Érica, São Paulo, 2002.
- MCCULLOCH W. e PITTS W., – "A Logical Calculus of the Ideas Immanent in Nervous Activity" – *Bulletin of Mathematic Biophysics*, 1943.
- MOLGA, E., CHERBANSKI, R. Hibrid first – principle – neural – network – approach to modeling of the liquid – liquid reacting system. *Chemical Engineering Science*, v. 54, 2467-2473, 1999.
- MINSKY, M.; PAPERT, S.. *Perceptrons*. Cambridge, MA, MIT Press, 1969.
- NASCIMENTO, C. A. O., YAMAMOTO, C. L. "Modelagem de processos químicos via redes neurais". In: *Workshop em Aplicações de Redes Neurais*, 1993, S.I. [S. l.:s.n.], "não paginado", 1993.
- NASCIMENTO, C. A. O., CHAN W. M. Use of Neural Networks for Modeling of Olefin Polymerization in high pressure tubular reactors. *Journal of Applied Polymer Science*, v. 53, 1277-1289, 1994.

- NELSON, M. M., ILLINGWORTH, W. T., "A Practical Guide to Neural Nets", Addison-Wesley Publishing Company, Inc., Massachusetts, 1991.
- NETO, BENÍCIO DE BARROS – DQF – UFPE. Curso de Algoritmos de Calibração Multivariada, Universidade Federal da Bahia, Novembro de 2002.
- OLIVEIRA, FREDE, Curso de Simulação não convencional de Processos Químicos.– UFBA – MEQ – Mestrado em Engenharia Química. Julho 2001.
- PARAMESWARAN, A. K.; SHARMA, V. K.; KUNZRU, D.; " Modeling of Nafta Pyrolysis in Swaged Coils"; The Canadian Journal of Chemical Engineering, 66, 957-63, dezembro 1988.
- PLEHIERS, P. M.; REYNIERS, G. C.; FROMENT, G. F.; "Simulation of the Run Length of an Ethane Cracking Furnace"; Ind. Eng. Chem. Research, 29, 636-41, 1990.
- QUANTRILLE, T. E., LIU, Y. A., "Artificial Inteligence in Chemical Engineering", Academic Press, Inc., 1991.
- QI, H. , "A Hibrid neural network – first principles model for fixed – bed reactor". Chemical Engineering Science, v. 54, 2521-2526, 1999.
- RAE, R. W.; FRY, W. F.; "Vinyl Chloride by a Dehydrochlorination Process"; Patente UK 1.494.797, dezembro 14, 1977.
- RANZI E.; DENTE, M.; PIERUCCI, S.; GOOSSENS, A. G.; BARENDREGT, S.; Paper presented at the 13<sup>th</sup> European Symposium on "Routine Calculations and Use of Computers in Chemical Engineering"; Hungary, 1980.
- RANZI E.; GROTTOLI, M. G.; BUSSANI, G.; CHE S. C.; ZAHNG G.; "A New Simulation Program Predicts EDC Furnace Performences"; La Chimica & L'Industria, 75, 261-69; 1993.
- RANZI, E., DENTE, M.; FARAVELLI, T.;ROVAGLIO, M.; MULLICK, S.; BUSSANI, G.; "Mechanistic Modeling of Chlorinated Reacting Systems (Modeling of EDC Pyrolysis Furnaces), paper no. 31d, presented at the AIChE Summer National Conference, San Diego, 19-22 de agosto, 1990.

RANZI, E.; DENTE, M.; PIERUCCI, S.; BARENDREGT, S.; CRONIN, P.; "Coking Simulation Aids On-Stream Time", Oil & Gas Journal Report, 49-52; September 2, 1985.

RAVAGNANI, S. P., Modelos de Ordem Reduzida para Processos de Separação por Destilação Multicomponente em Colunas de Pratos. Aplicação a Sistemas Complexos, Tese de Doutorado, USP, São Paulo, 1988.

RICE, RICHARD G., DO, DUONG, D.; "Applied Mathematics and Modeling for Chemical Engineers, John Wiley & Sons, Inc., 1995.

RECHMEIER GERHARD; "Production of Vinyl Chloride by Thermal Cracking of 1,2-Dichloroethane" ; Patente US 3.903.182; Março, 12, 1974.

ROSELLA C. e MARIO G., "On the approximation error introduced using Principal Component Analysis in Neural Networks" , Nonlinear Analysis", 47, 5785-5794, 2001.

RUMMELHART D., HINTON, WILLIAMS– "Learning Internal Representations by Error Propagation, em Parallel Distributed Processing "– MIT Press, Cambridge, 1986.

SHEPHERD, G. M., "The Synaptic Organization of the Brain", 2nd ed., Oxford University Press, New York, 1979.

STEVANOVIC, S. J. "Neural Network for process analysis and optimization: modeling and applications". Computers Chemical Engineering , v. 18, no. 11/12, 1149-1155, February 1994.

SUNDARAM, K. M.; FROMENT, G. F.; "Two Dimensional Model for the Simulation of Tubular Reactors for Thermal Cracking"; Chemical Engineering Science, 35, 364-71, 1980.

SUNDARAM K. M.; FROMENT G. F.; " Kinetics of Coke Deposition in the Thermal Cracking of Propane"; Chemical Engineering Science, vol. 34, 635-644, 1979.

SUNDARAM K. M.; VAN DAMME, P. S.; FROMENT, G. F.; "Coke Deposition in the Thermal Cracking of Ethane"; AIChE Journal, 27,6, 946-51; 1981.

THE B.F. GOODRICH COMPANY, Patente US 938.84, outubro 9, 1963.

TONG H., CROWE, C. M., "Detecting Persistent Gross Errors by Sequential Analysis of Principal Components", Computers Chemical Engng., v. 20, Suppl. S733-738, 1996.

ULLMANN'S ENCYCLOPEDIA OF INDUSTRIAL CHEMISTRY. Flórida: Ed.Wolfgang Gerhartz, 1985. Volume A6, página 283. ISBN 0-89573-182-7.

VILLADSEN, J. , "Selected Approximation Methods for Chemical Engineering Problems", Lyngby, Danmarks Tekniske Hojskole, 1970.

VILLADSEN, J., MICHELSEN, M.L., "Solution of Differential Equations Models by Polinomial Approximation", Prentice Hall, New Jersey, 1978.

WANG, X. Z., "Neural nets, fuzzy sets and digraphs in safety and operability studies of refinery reaction process". Chemical Engineering Science, v. 51, n. 10 2169-2178, 1996.

WOLD, S., P. GELADI, K. ESBENSEN AND J. OHMAN, "Multi-way Principal Components and PLS Analysis"; Journal Chemometrics 1, pp. 41-56, 1987.

YAMAMOTO, C. I., BABA, E. H., NASCIMENTO, C. A. O. "Previsão do Comportamento de reator de síntese de amônia via rede neuronal". In: Simpósio Brasileiro de Automação Inteligente, 1, 1993, Rio Claro, SP. Anais... Rio Claro, SP: SBAI, 48-57, 1993.

ZHANG, J., Martin, E. B., Morris, A. J., "Process Monitoring using non-linear statistical techniques", Chemical Engineering Journal, 67, 181-189, 1997.

ZHOU, X., "Modeling of a fixed bed reactor using the K-L expansion and neural networks". Chemical Engineering Science, v. 51, n. 10, 2179-2188, 1996.



## ANEXO 1 – METODOLOGIA DE REDES NEURAIS ARTIFICIAIS – ALGORITMO DE TREINAMENTO BACKPROPAGATION

### Exemplo de processamento na Rede Neural Artificial – 3 camadas Método Backpropagation.

1. Leitura dos Dados de Entrada ( $I_k$ ) e Valores Desejados ( $d_k$ ):

$$I_i = [I_1 \quad I_2 \quad I_3] - \text{dados de entrada}$$

$$d_k = [d_1 \quad d_2 \quad d_3] - \text{valores desejados}$$

2. Transposição dos vetores  $I_k$  e  $d_k$

$$I_i = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} \quad \text{e} \quad d_k = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

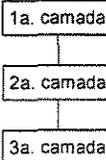
3. Inicialização dos pesos da 1ª camada –  $v_{i,j}$

$$v_{i,j} = \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{bmatrix}$$

4. Inicialização dos pesos da 2ª camada –  $w_{i,j}$

$$w_{i,j} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}$$

5. Inicialização dos limiares das 3 camadas –  $t_{i,j}$

$$T_{i,j} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} \begin{matrix} \longrightarrow \\ \longrightarrow \\ \longrightarrow \end{matrix}$$


6. Cálculo dos sinais  $x_i$  da 1ª camada:

$$x_i = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} I_1 \\ I_2 \\ I_3 \end{bmatrix} - \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}$$

## 7. Aplicação da função sigmóide:

$$a_i = f(x_i) = \frac{1}{1 + e^{(-x_i)}}$$

$$a_i = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix}$$

$$a_i = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + e^{(-x_1)}} \\ \frac{1}{1 + e^{(-x_2)}} \\ \frac{1}{1 + e^{(-x_3)}} \end{bmatrix}$$

8. Cálculo dos sinais  $x_j$  da 2ª camada:

$$x_j = \sum_{i=1}^L [(v_{ij} a_i) - T_{Bj}]$$

$$x_j = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} v_{11} & v_{12} & v_{13} \\ v_{21} & v_{22} & v_{23} \\ v_{31} & v_{32} & v_{33} \end{bmatrix}_{3 \times 3} \cdot \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}_{3 \times 1} - \left( \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}_{3 \times 3} \right)_{2.a. \text{ coluna todas as linhas}}$$

$$x_j = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} v_{11} \cdot a_1 + v_{12} \cdot a_2 + v_{13} \cdot a_3 \\ v_{21} \cdot a_1 + v_{22} \cdot a_2 + v_{23} \cdot a_3 \\ v_{31} \cdot a_1 + v_{32} \cdot a_2 + v_{33} \cdot a_3 \end{bmatrix}_{3 \times 1} - \left( \begin{bmatrix} T_{11} & T_{21} & T_{31} \\ T_{12} & T_{22} & T_{32} \\ T_{13} & T_{23} & T_{33} \end{bmatrix}_{3 \times 3} \right)_{2.a. \text{ coluna todas as linhas}}$$

$$x_j = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} v_{11} \cdot a_1 + v_{12} \cdot a_2 + v_{13} \cdot a_3 \\ v_{21} \cdot a_1 + v_{22} \cdot a_2 + v_{23} \cdot a_3 \\ v_{31} \cdot a_1 + v_{32} \cdot a_2 + v_{33} \cdot a_3 \end{bmatrix}_{3 \times 1} - \begin{bmatrix} T_{21} \\ T_{22} \\ T_{23} \end{bmatrix}_{3 \times 1}$$

$$x_j = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} v_{11} \cdot a_1 + v_{12} \cdot a_2 + v_{13} \cdot a_3 - T_{21} \\ v_{21} \cdot a_1 + v_{22} \cdot a_2 + v_{23} \cdot a_3 - T_{22} \\ v_{31} \cdot a_1 + v_{32} \cdot a_2 + v_{33} \cdot a_3 - T_{23} \end{bmatrix}_{3 \times 1}$$

$$x_2 = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

## 9. Aplicação da função sigmóide na 2ª camada:

$$b_j = f(x_j) = \frac{1}{1 + e^{(-x_j)}}$$

$$b_j = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix}$$

$$b_j = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + e^{(-x_1)}} \\ \frac{1}{1 + e^{(-x_2)}} \\ \frac{1}{1 + e^{(-x_3)}} \end{bmatrix}$$

10. Cálculo dos sinais  $x_k$  da 3ª camada:

$$x_k = \sum_{j=1}^n [(w_{jk} b_k) - T_{cn}]$$

$$x_k = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}_{3 \times 3} \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}_{3 \times 1} - \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}_{3 \times 3}' \quad (3a \text{ coluna da matriz transposta})$$

$$x_k = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} w_{11} \cdot b_1 + w_{12} \cdot b_2 + w_{13} \cdot b_3 \\ w_{21} \cdot b_1 + w_{22} \cdot b_2 + w_{23} \cdot b_3 \\ w_{31} \cdot b_1 + w_{32} \cdot b_2 + w_{33} \cdot b_3 \end{bmatrix}_{3 \times 1} - \begin{bmatrix} T_{11} & T_{21} & T_{31} \\ T_{12} & T_{22} & T_{32} \\ T_{13} & T_{23} & T_{33} \end{bmatrix}_{3 \times 3} \quad (3a \text{ coluna da matriz})$$

$$x_k = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} w_{11} \cdot b_1 + w_{12} \cdot b_2 + w_{13} \cdot b_3 \\ w_{21} \cdot b_1 + w_{22} \cdot b_2 + w_{23} \cdot b_3 \\ w_{31} \cdot b_1 + w_{32} \cdot b_2 + w_{33} \cdot b_3 \end{bmatrix}_{3 \times 3} - \begin{bmatrix} T_{31} \\ T_{32} \\ T_{33} \end{bmatrix}_{3 \times 1}$$

$$x_k = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}_{3 \times 1} = \begin{bmatrix} v_{11} \cdot b_1 + v_{12} \cdot b_2 + v_{13} \cdot b_3 - T_{31} \\ v_{21} \cdot b_1 + v_{22} \cdot b_2 + v_{23} \cdot b_3 - T_{32} \\ v_{31} \cdot b_1 + v_{32} \cdot b_2 + v_{33} \cdot b_3 - T_{33} \end{bmatrix}_{3 \times 3}$$

$$x_k = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

11. Aplicação da função sigmóide na 3ª camada:

$$c_k = f(x_k) = \frac{1}{1 + e^{-x_k}}$$

$$c_k = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ f(x_3) \end{bmatrix}$$

$$c_k = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} = \begin{bmatrix} \frac{1}{1 + e^{-x_1}} \\ \frac{1}{1 + e^{-x_2}} \\ \frac{1}{1 + e^{-x_3}} \end{bmatrix}$$

12. Cálculo do erro da 3ª camada:

$$e_k = c_k(1 - c_k)(d_k - c_k)$$

$$\begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \\ c_3 \end{bmatrix} \cdot \begin{bmatrix} 1 - c_1 \\ 1 - c_2 \\ 1 - c_3 \end{bmatrix} \cdot \begin{bmatrix} 1 - d_1 \\ 1 - d_2 \\ 1 - d_3 \end{bmatrix} \cdot \text{(operação numérica)}$$

13. Cálculo dos novos pesos  $w_{j,k}$  da 3ª camada:

$$w_{j,k \text{ novo}} = w_{j,k} + \eta_c \cdot b_j \cdot e_k$$

$$\begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}_{\text{novo}} = \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix} + \eta_c \cdot \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

14. Cálculo dos novos limiares  $t_{c,k}$

$$t_{c,k \text{ novo}} = t_{c,k} + \eta_c \cdot e_k$$

$$\begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix}_{\text{novo}} = \begin{bmatrix} T_{11} & T_{12} & T_{13} \\ T_{21} & T_{22} & T_{23} \\ T_{31} & T_{32} & T_{33} \end{bmatrix} + \eta_c \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}$$

15. Cálculo do erro da 2ª camada:

$$E_j = b_j (1 - b_j) \sum_{k=1}^n [(w_{jk} e_k)]$$

$$\begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \begin{bmatrix} 1 - b_1 \\ 1 - b_2 \\ 1 - b_3 \end{bmatrix} \begin{bmatrix} w_{11} & w_{12} & w_{13} \\ w_{21} & w_{22} & w_{23} \\ w_{31} & w_{32} & w_{33} \end{bmatrix}_{3 \times 3} \cdot \begin{bmatrix} e_1 \\ e_2 \\ e_3 \end{bmatrix}_{3 \times 1}$$

$$\begin{bmatrix} E_1 \\ E_2 \\ E_3 \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \cdot \begin{bmatrix} 1 - b_1 \\ 1 - b_2 \\ 1 - b_3 \end{bmatrix} \cdot \begin{bmatrix} w_{11} \cdot e_1 + w_{12} \cdot e_2 + w_{13} \cdot e_3 \\ w_{21} \cdot e_1 + w_{22} \cdot e_2 + w_{23} \cdot e_3 \\ w_{31} \cdot e_1 + w_{32} \cdot e_2 + w_{33} \cdot e_3 \end{bmatrix}$$

16. Cálculo do erro quadrático médio:

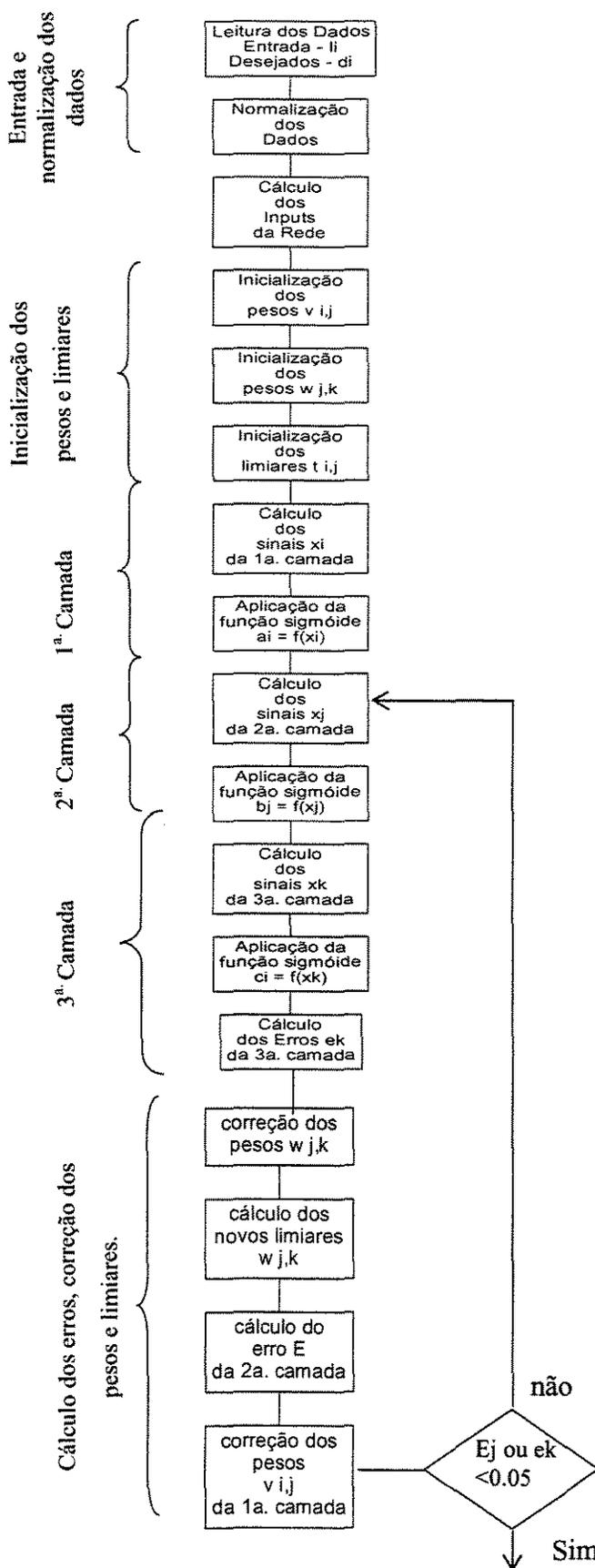
$$erro = \sum_{k=1 a 3} (d_k - c_k)^2$$

$$erro = \sum \begin{bmatrix} d_1 - c_1 \\ d_2 - c_2 \\ d_3 - c_3 \end{bmatrix}^2$$

$$erro = (d_1 - c_1)^2 + (d_2 - c_2)^2 + (d_3 - c_3)^2$$

17. Se o erro for maior ou igual a 0.001 voltar para a etapa 9, caso contrário pare e imprima os resultados.

Fluxograma da Metodologia de Redes Neurais Artificiais Equações do Algoritmo



1.  $I_i = [I_1 \ I_2 \ I_3 \ I_4 \dots I_L]$   
 $d_k = [d_1 \ d_2 \ d_3 \ d_4 \dots d_L]$

2.  $v_{ij}$

3.  $w_{j,k}$

4.  $t_{ij}$

5.  $x_i = I_i - T_{1i}$   
 $x_i = [x_1 \ x_2 \ x_3 \ x_4 \dots x_L]$

6.  $a_i = f(x_i) = \frac{1}{1 + e^{(-x_i)}}$

7.  $x_j = \sum_{i=1}^L [(v_{ij} a_i) - T_{Bj}]$

$x_j = [x_1 \ x_2 \ x_3 \ x_4 \dots x_m]$

8.  $b_j = f(x_j) = \frac{1}{1 + e^{(-x_j)}}$

9.  $x_k = \sum_{j=1}^n [(w_{jk} b_k) - T_{cn}]$

$x_k = [x_1 \ x_2 \ x_3 \ x_4 \dots x_n]$

10.  $c_k = f(x_k) = \frac{1}{1 + e^{(-x_k)}}$

11.  $e_k = c_k (1 - c_k) (d_k - c_k)$

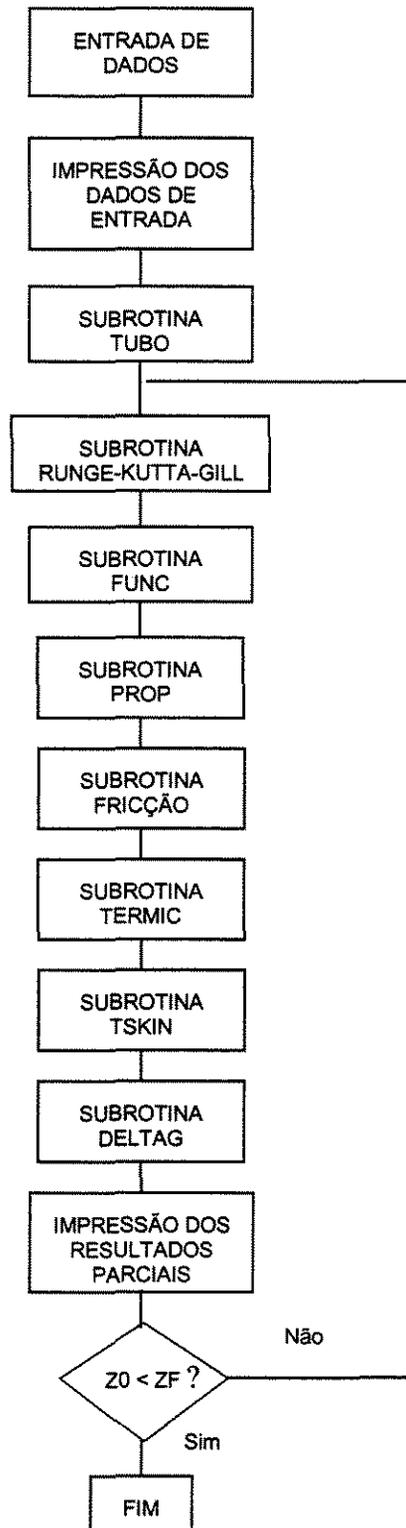
12.  $w_{j,k \text{ novo}} = w_{j,k} + \eta_c \cdot b_j \cdot e_k$

13.  $t_{c,k \text{ novo}} = t_{c,k} + \eta_c \cdot e_k$

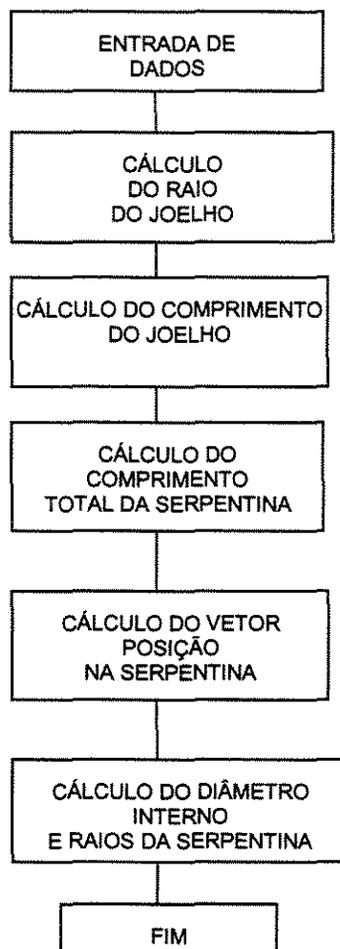
14.  $E_j = b_j (1 - b_j) \sum_{k=1}^n [(w_{jk} e_k)]$

## ANEXO 2 – FLUXOGRAMAS DOS PROGRAMAS UTILIZADOS

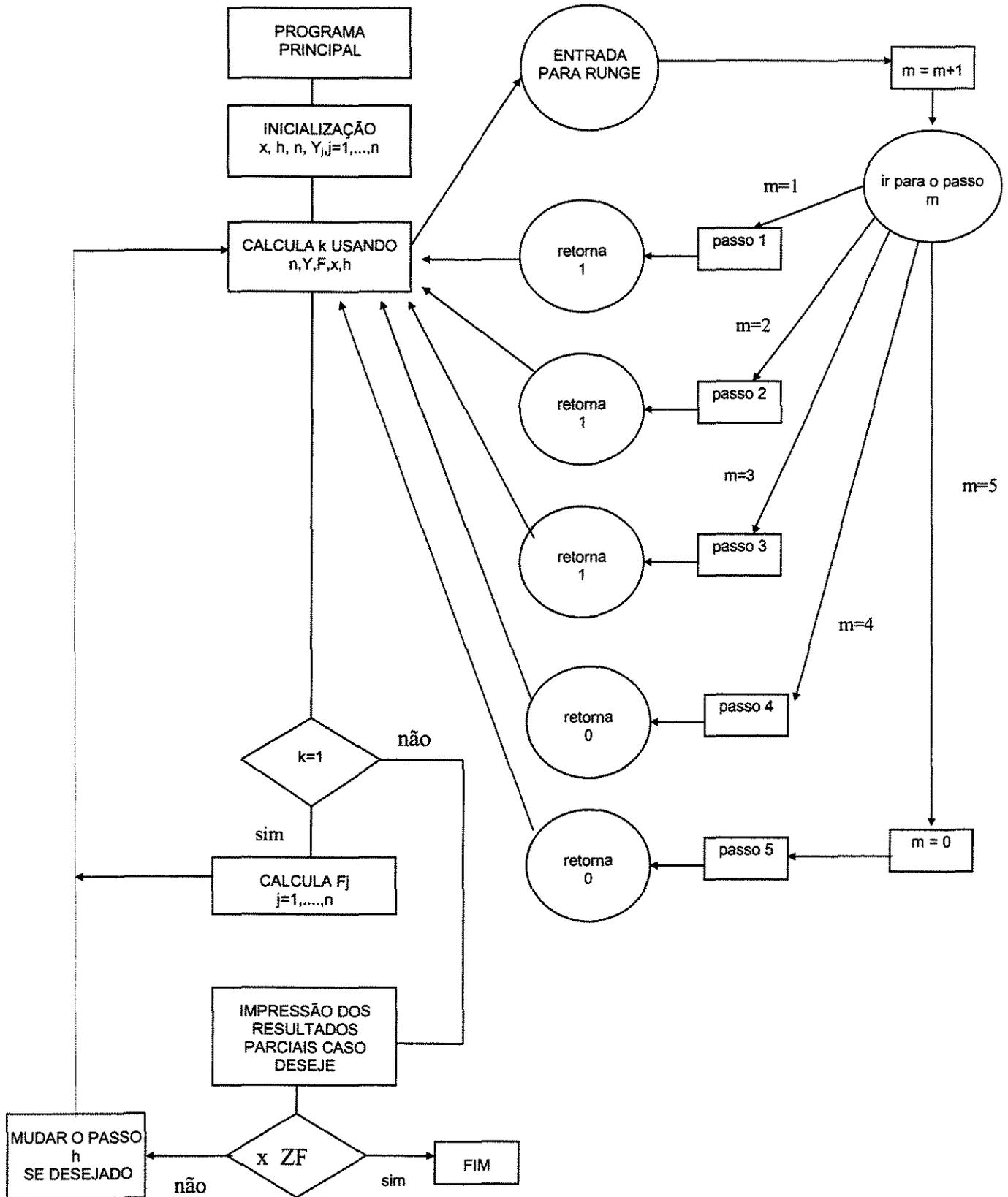
## 2.1 - FLUXOGRAMA DO PROGRAMA PRINCIPAL



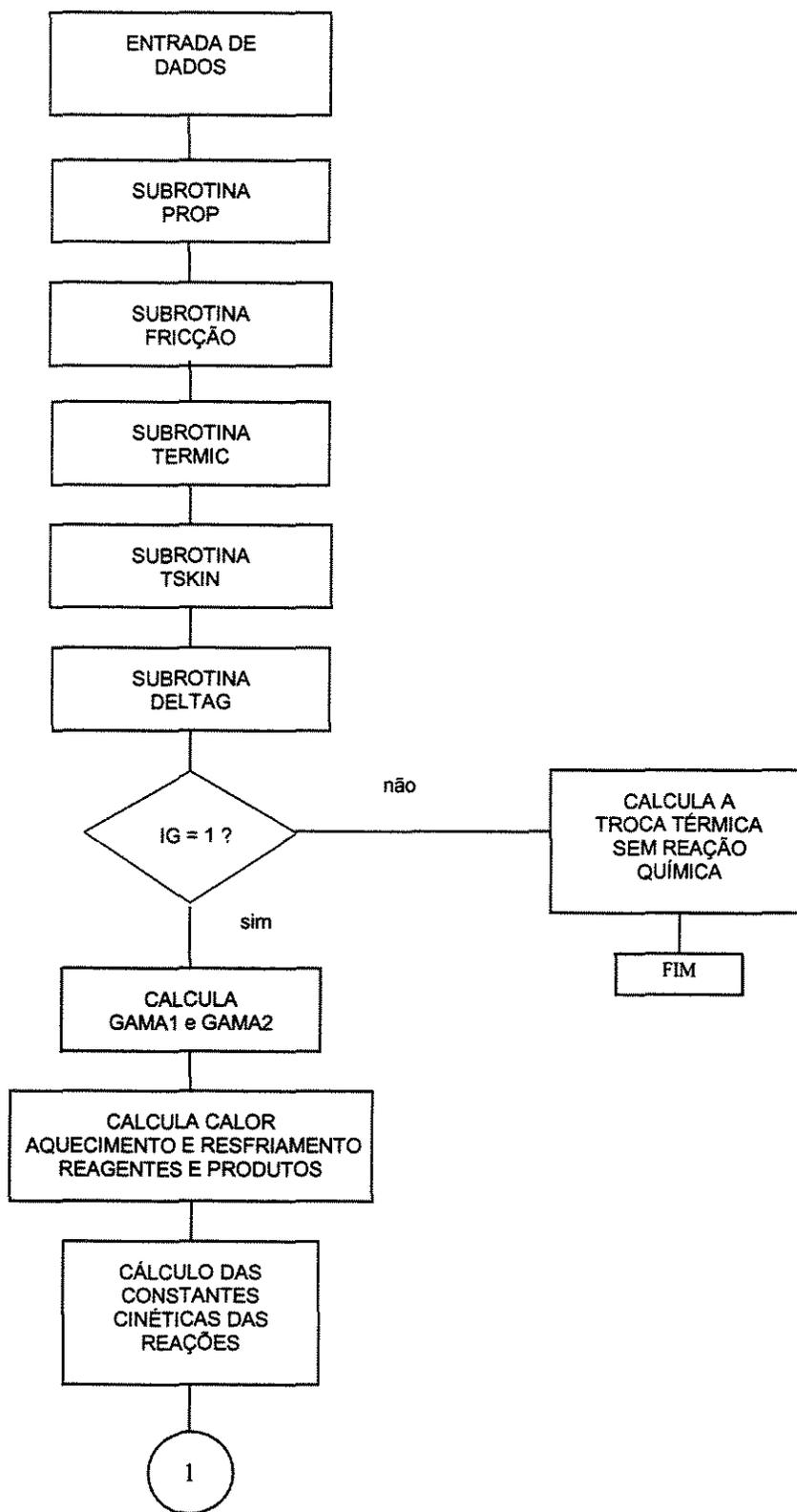
## 2.2 - FLUXOGRAMA DA SUBROTINA TUBO

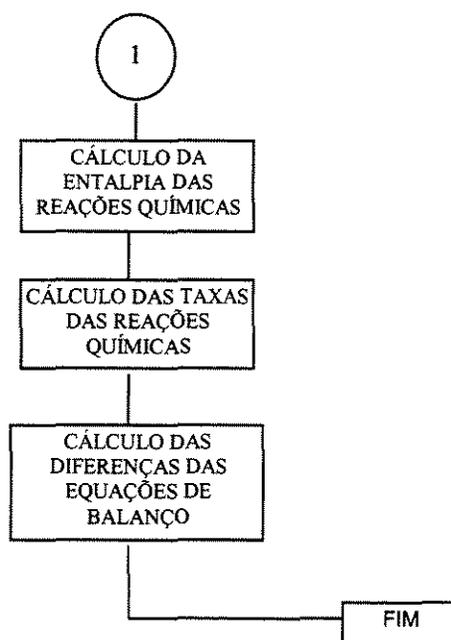


### 2.3 - FLUXOGRAMA DA SUBROTINA RUNGE-KUTTA-GILL

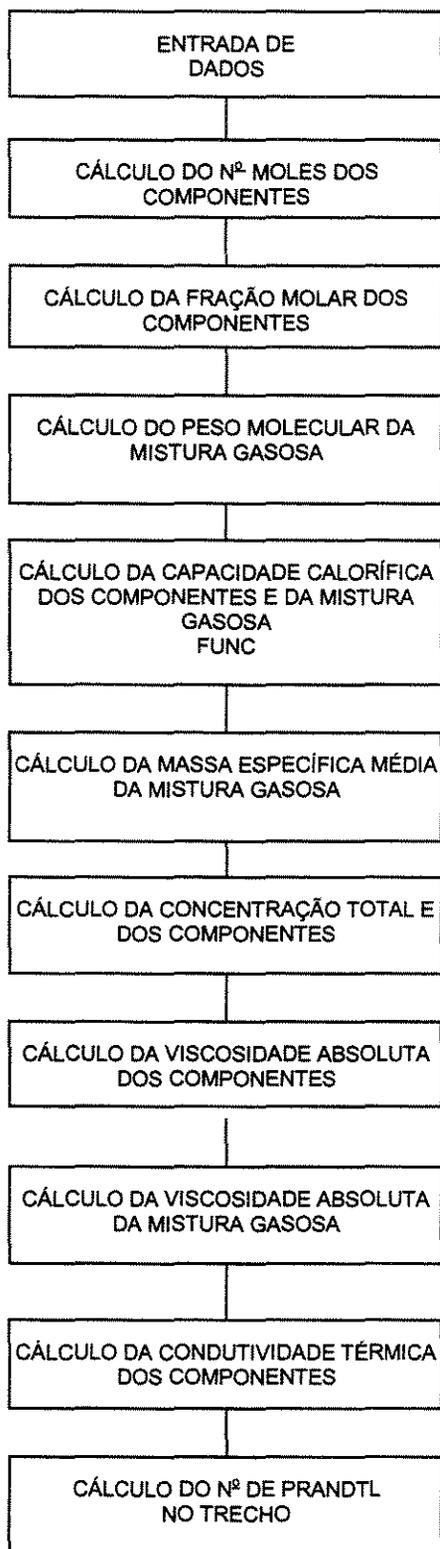


## 2.4 - FLUXOGRAMA DA SUBROTINA FUNC

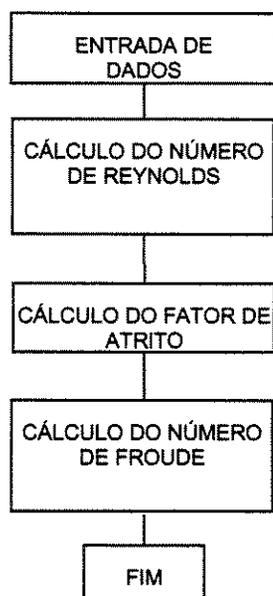




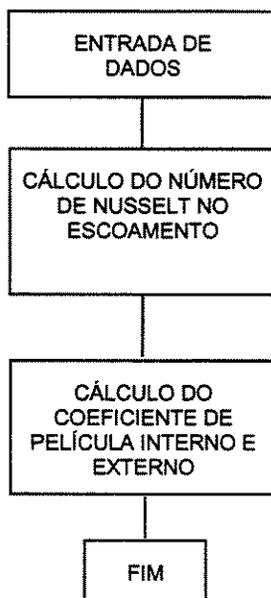
## 2.5 - FLUXOGRAMA DA SUBROTINA PROP



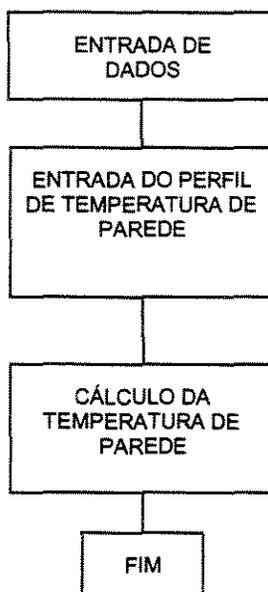
## 2.6 - FLUXOGRAMA DA SUBROTINA FRICÇÃO



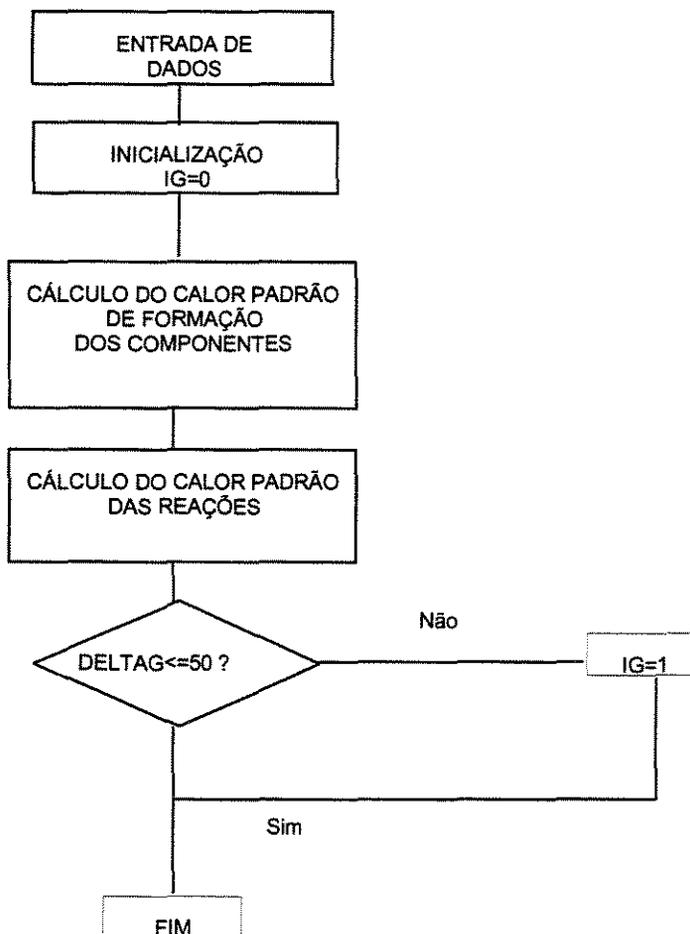
## 2.7 - FLUXOGRAMA DA SUBROTINA TERMIC



## 2.8 - FLUXOGRAMA DA SUBROTINA TSKIN



## 2.9 - FLUXOGRAMA DA SUBROTINA DELTAG



## ANEXO 3 – PROGRAMAS EM FORTRAN

### 3.1 - MÉTODO DAS DIFERENÇAS FINITAS

```

$ DEBUG
*
*****
*                               DIFFIN.FOR                               *
*                               *                                         *
*   MODELAGEM DINÂMICA E SIMULAÇÃO DE UMA FORNALHA PARA O             *
*   CRAQUEAMENTO TÉRMICO DO 1,2-DICLOROETANO.                         *
*   RESOLUÇÃO DAS EQ. DIFERENCIAIS PELO MÉTODO DE DIFERENÇAS         *
*   FINITAS E RUNGE-KUTTA-GILL DE QUARTA ORDEM                        *
*                               *                                         *
*   MODELO UNIDIMENSIONAL, HOMOGENEO, ESCOAMENTO PISTÃO,             *
*   SEM DISPERSÃO AXIAL DE CALOR E MASSA.                             *
*                               *                                         *
*                               *                                         *
*   ESQUEMA DA REAÇÃO CONSIDERADO:                                     *
*                               *                                         *
*           k1                                                         *
*   EDC  -----> MVC + HCl                                           *
*     1     5     6                                                     *
*                               *                                         *
*           k2                                                         *
*   MVC  -----> C2H2 + HCl                                           *
*     5     7     6                                                     *
*                               *                                         *
*   MODELO CINÉTICO: VIA REAÇÕES MOLECULARES                          *
*   BALANÇO DE MASSA DO 1,2-DICLOROETANO (EDC)                       *
*   BALANÇO DE MASSA DO MONOCLORETO DEE VINILA (MVC)                 *
*   BALANÇO DE MASSA DO ÁCIDO CLORÍDRICO (HCl)                       *
*   BALANÇO DE ENERGIA                                                *
*                               *                                         *
*   RESOLVENDO POR RUNGE-KUTTA-GILL, USAREMOS PARA CADA              *
*   EQUACAO DIFERENCIAL                                               *
*   Y(N+1) = Y(N) + 1/6(K1+K4) + 1/3(B*K2 + D*K3)                     *
*   K1 = h * f(Z(N),Y(N))                                              *
*   K2 = h * f(Z(N)+1/2 h,Y(N)+1/2 K1)                                *
*   K3 = h * f(Z(N)+1/2 h,Y(N)+A*K1+B*K2)                             *
*   K4 = h * f(Z(N)+h,Y(N)+C*K2+D*K3)                                  *
*   COM: A=(SQRT(2)-1)/2; B = (2-SQRT(2))/2; C = -SQRT(2)/2;         *
*         D = (2+SQRT(2))/2                                             *
*   SENDO QUE A,B,C e D SAO USADAS PARA MINIMIZAR OS ERROS DE        *
*   ARREDONDAMENTO                                                     *
*   CONSIDEROU-SE QUE NO ESCOAMENTO HA A VARIACAO AXIAL DO COEFICIENTE*
*   DE PELICULA INTERNO(HO), ISTO E DO NUD                            *
*                               *                                         *
*****
IMPLICIT REAL*8 (A-H,O-Z)
DOUBLE PRECISION SEL(2000),CONV(2000),FQR(2000)
COMMON / BLOC1 / NI,NC,NRE,DF
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO
COMMON / BLOC4 / CP(2000),CPM(2000),PMM(2000),AA(2000,2000)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(2000),HO(2000),HINT(2000),FI(2000,2000)
COMMON / BLOC8 / DCP1(2000),DCP2(2000)

```

```

COMMON / BLOC9 / A(2000),B(2000),PM(2000),RD(2000),RIN(2000),
SR1(2000),R2(2000)
COMMON / BLOC10 / AV(2000),BV(2000),CV(2000),VISA(2000)
COMMON / BLOC11 / AK(2000),BK(2000),CK(2000),DG(2000),Tw(2000)
COMMON / BLOC12 / UI(2000),ROM(2000),UNUD(2000)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(2000),TKM(2000),RE(2000),PR(2000),
$FF(2000),FR(2000),VISAMCP(2000)
COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(2000),BG(2000),CG(2000),TK(2000),FM(2000)
COMMON / BLOC17 / P(2000),Twp,CT1(2000),CT2(2000)
COMMON / BLOC18 / NT,NE,Q
COMMON / BLOC19 / V(2000),DELTAQ1(2000),DELTAQ2(2000),PROD(2000)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(2000),CS(2000),DGRE1(2000),DGRE2(2000)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA2(2000),RMULT(2000)
COMMON / BLOC23 / PMMEDC(2000),PMMMVC(2000),PMMHCI(2000),
$PMMCH(2000)
COMMON / BLOC24 / ANUM(2000),BDEN(2000),ALFA(2000),VISCN(2000)
COMMON / BLOC25 / VV(2000),CT(2000),TN(2000)
COMMON / BLOC26 / ANUMEDC(2000),ANUMMVC(2000),ANUMHCI(2000),
$ANUMCH(2000)
COMMON / BLOC27 / BDENEDC(2000),BDENMVC(2000),BDENHCI(2000),
$BDENCH(2000)
COMMON / BLOC28 / VISAMEDC(2000),VISAMMVC(2000),VISAMHCI(2000),
$VISAMCH(2000)
REAL ITEM
  Q=4
  NI=6
  NC=4
  NRE=2
  ZF=337.2579
  DF=ZF/(Q*ZF)
PASSO=0.05
G0=2208679.4
VM=30842.0
RG=1.987
NT=20
TL=16.46
TE=0.270
RB=180.
DE=0.1524
EP=0.00953
EC=0.000
CKT=3.
PMEDC=99.0
PMMVC=62.5
PMHCI=36.5
PMCH=26.0
  DO 1000 I=1,Q
  PM(I)=PMEDC
  PM(I+Q)=PMMVC
  PM(I+2*Q)=PMHCI
  PM(I+5*Q)=PMCH
1000 CONTINUE
  AEDC=4.363
  AMVC=1.419
  AHCI=7.000
  ACH=3.999
  DO 8 I=1,Q
  A(I)=AEDC
  A(I+Q)=AMVC
  A(I+2*Q)=AHCI
  A(I+5*Q)=ACH
8 CONTINUE
  BEDC=0.046856
  BMVC=0.048080
  BHCI=0.00006
  BCH=0.030168
  DO 207 I=1,Q
  B(I)=BEDC
  B(I+Q)=BMVC
  B(I+2*Q)=BHCI

```

```
B(I+5*Q)=BCH
207 CONTINUE
  AVEDC=-4.0E-9
  AVMVC=4.0E-9
  AVHCI=-2.0E-8
  AVCH=4.0E-9
  DO 10 I=1,Q
    AV(I)=AVEDC
    AV(I+Q)=AVMVC
    AV(I+2*Q)=AVHCI
    AV(I+5*Q)=AVCH
10 CONTINUE
  BVEDC=3.0E-5
  BVMVC=2.0E-5
  BVHCI=7.0E-5
  BVCH=2.0E-5
  DO 200 I=1,Q
    BV(I)=BVEDC
    BV(I+Q)=BVMVC
    BV(I+2*Q)=BVHCI
    BV(I+5*Q)=BVCH
200 CONTINUE
  CVEDC=0.002
  CVMVC=0.0057
  CVHCI=0.0072
  CVCH=0.0057
  DO 201 I=1,Q
    CV(I)=CVEDC
    CV(I+Q)=CVMVC
    CV(I+2*Q)=CVHCI
    CV(I+5*Q)=CVCH
201 CONTINUE
  AKEDC=-3.0E-11
  AKMVC=-1.0E-8
  AKHCI=2.0E-8
  AKCH=-3.0E-11
  DO 11 I=1,Q
    AK(I)=AKEDC
    AK(I+Q)=AKMVC
    AK(I+2*Q)=AKHCI
    AK(I+5*Q)=AKCH
11 CONTINUE
  BKEDC=4.0E-5
  BKMVC=6.0E-5
  BKHCI=3.0E-5
  BKCH=4.0E-5
  DO 202 I=1,Q
    BK(I)=BKEDC
    BK(I+Q)=BKMVC
    BK(I+2*Q)=BKHCI
    BK(I+5*Q)=BKCH
202 CONTINUE
  CKEDC=-0.0046
  CKMVC=-0.007
  CKHCI=0.0025
  CKCH=-0.0046
  DO 203 I=1,Q
    CK(I)=CKEDC
    CK(I+Q)=CKMVC
    CK(I+2*Q)=CKHCI
    CK(I+5*Q)=CKCH
203 CONTINUE
  AGEDC=-130.549
  AGMVC=31.557
  AGHCI=-92.209
  AGCH=227.187
  DO 12 I=1,Q
    AG(I)=AGEDC
    AG(I+Q)=AGMVC
    AG(I+2*Q)=AGHCI
    AG(I+5*Q)=AGCH
12 CONTINUE
  BGEDC=1.8562E-01
```





```

END IF
IF ((Z.GT.202.60).AND.(Z.LT.219.06)) THEN
    Twp= TWP2+5.0*DTW3
END IF
IF ((Z.GT.219.06).AND.(Z.LT.219.48)) THEN
    Twp= TWP2+5.5*DTW3
END IF
IF (Z.EQ.219.48) THEN
    Twp= TWP2+6.0*DTW3
END IF

```

```

*-----*
*                SECAO 4 DA SERPENTINA                *
*-----*

```

```

IF ((Z.GT.219.48).AND.(Z.LE.235.94)) THEN
    Twp=TWP3
END IF
IF ((Z.GT.235.94).AND.(Z.LT.236.36)) THEN
    Twp= TWP3+0.5*DTW4
END IF
IF ((Z.GE.236.36).AND.(Z.LE.252.82)) THEN
    Twp= TWP3+1.0*DTW4
END IF
IF ((Z.GT.252.82).AND.(Z.LT.253.25)) THEN
    Twp= TWP3+1.5*DTW4
END IF
IF ((Z.GE.253.25).AND.(Z.LE.269.71)) THEN
    Twp= TWP3+2.0*DTW4
END IF
IF ((Z.GT.269.71).AND.(Z.LT.270.13)) THEN
    Twp= TWP3+2.5*DTW4
END IF
IF ((Z.GE.270.13).AND.(Z.LE.286.59)) THEN
    Twp= TWP3+3.0*DTW4
END IF
IF ((Z.GT.286.59).AND.(Z.LT.287.02)) THEN
    Twp= TWP3+3.5*DTW4
END IF
IF ((Z.GT.287.02).AND.(Z.LT.303.48)) THEN
    Twp= TWP3+4.0*DTW4
END IF
IF ((Z.GE.303.48).AND.(Z.LE.303.90)) THEN
    Twp=TWP3+4.5*DTW4
END IF
IF ((Z.GT.303.90).AND.(Z.LT.320.36)) THEN
    Twp= TWP3+5.0*DTW4
END IF
IF ((Z.GT.320.36).AND.(Z.LT.320.78)) THEN
    Twp= TWP3+5.5*DTW4
END IF
IF ((Z.GT.320.78).AND.(Z.LE.337.2579)) THEN
    Twp= TWP3+6.0*DTW4
END IF
WRITE(6,*) K,Z,Twp
Tw(K)=Twp
write(*,*) K,Z,Tw(K)
GOTO 5
13 write(6,*) K,Tw(K)

```

```
***** VALOR INICIAL DE MVC, HCI, T , PT e 1/Mm *****
```

```

Y(Q+1)=0.0005
Y(2*Q+1)=0.0005
Y(3*Q+1)=513./1000.
Y(4*Q+1)=23.33/23.33
Y(6*Q+1)=0.01010101

```

```
***** VALOR INICIAL DE FRACAO MASSICA DE EDC *****
```

```

Y(1)=0.9950
W10 =0.9950
Y1A = Y(1)
Y9A = Y(Q+1)
TO = 1000.
ROME = Y(4*Q+1)*23.33*PM(1)/(0.082*1.023*Y(3*Q+1)*1000)
CALL TUBO

```

```

DO 1090 I=1,Q
  X(I)=Y(1)
  X(I+Q)=Y(Q+1)
  X(I+2*Q)=Y(2*Q+1)
  X(I+3*Q)=Y(3*Q+1)
  X(I+4*Q)=Y(4*Q+1)
  X(I+6*Q)=Y(6*Q+1)
  WRITE(*,*)X(I),X(I+Q),X(I+2*Q),X(I+3*Q),X(I+4*Q),
  $X(I+5*Q),X(I+6*Q)
1090 CONTINUE
  CONT=1
  ZM=7000
  NP=10
  NG=ZM/PASSO
  DO 100 TEM=PASSO,ZM,PASSO
  CALL RKG
  IF(CONT.NE.(NG/NP)) GOTO 130
  DO 1003 I=1,Q
  CONV(I) = (Y(1)-X(I))/Y(1)
1003 CONTINUE
  DO 1052 I=2,Q
  SEL(I) = ((X(I+Q)-Y9A)/(Y1A-X(I)))*PM(I)/PM(I+Q)
1052 CONTINUE
  DO 1004 I=1,Q
  X(I+5*Q) = 0.996-X(I)-X(I+Q)-X(I+2*Q)
1004 CONTINUE
*-----
*                CÁLCULO DO TEMPO DE RESIDENCIA NA SERPENTINA
*-----
C1=X(Q)*(PMM(Q)/PM(Q))*(X(5*Q)*23.33)/(0.082*(X(4*Q)*TO))
F1=X(Q)*VM/PM(Q)
TR=(3.1416*DI**2./4.)*ZF*3600./(F1/C1)
*-----
TEMP=(TEM*RT*ROME)/G0
*-----
*                IMPRESSÃO DOS RESULTADOS OBTIDOS
*-----
WRITE(*,150) TEM,TEMP
WRITE (8,150) TEM,TEMP
WRITE (7,150) TEM,TEMP
150  FORMAT(/,8X,'TEM=',F10.4,4X,'TEMP=',F12.6)
DO 250 I=1,Q
  WRITE(*,300) (X(J),J=I,I+6*Q,Q)
  WRITE(8,300) (X(J),J=I,I+6*Q,Q)
300  FORMAT(5X,F12.7,8F14.7,/)
  WRITE(7,500) SEL(I),CONV(I),TR,UI(I),HI(I),hint(I)
500  FORMAT(5X,F12.7,8F14.7,/)
250  CONTINUE
  CONT=0
130  CONT=CONT+1
100  CONTINUE
  DO 1050 I=1,Q
  FQR(I)=HINT(I)*(X(I+3*Q)*TO-Tw(I))
1050  CONTINUE
  CALL TEMPO (ITEMP)
  WRITE(*,*)
  WRITE(*,*) 'Tempo de execucao :',ITEMP
  STOP
  END
*-----
* SUBROTINA DE INTEGRACAO RUNGE-KUTTA-GILL DE QUARTA ORDEM
*-----
SUBROUTINE RKG
IMPLICIT REAL*8 (A-H,O-Z)
DOUBLE PRECISION C(4,2000)
COMMON / BLOC1 / NI,NC,NRE,DF
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO
COMMON / BLOC4 / CP(2000),CPM(2000),PMM(2000),AA(2000,2000)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(2000),HO(2000),HINT(2000),FI(2000,2000)
COMMON / BLOC8 / DCP1(2000),DCP2(2000)
COMMON / BLOC9 / A(2000),B(2000),PM(2000),RD(2000),RIN(2000),

```

```

$R1(2000),R2(2000)
COMMON / BLOC10 / AV(2000),BV(2000),CV(2000),VISA(2000)
COMMON / BLOC11 / AK(2000),BK(2000),CK(2000),DG(2000),Tw(2000)
COMMON / BLOC12 / UI(2000),ROM(2000),UNUD(2000)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(2000),TKM(2000),RE(2000),PR(2000),FF(2000)
$,FR(2000),VISAMCP(2000)
COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(2000),BG(2000),CG(2000),TK(2000),FM(2000)
COMMON / BLOC17 / P(2000),Twp,CT1(2000),CT2(2000)
COMMON / BLOC18 / NT,NE,Q
COMMON / BLOC19 / V(2000),DELTAQ1(2000),DELTAQ2(2000),PROD(2000)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(2000),CS(2000),DGRE1(2000),DGRE2(2000)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA2(2000),RMULT(2000)
COMMON / BLOC23 / PMMEDC(2000),PMMMVC(2000),PMMHCI(2000),
$PMMCH(2000)
COMMON / BLOC24 / ANUM(2000),BDEN(2000),ALFA(2000),VISCN(2000)
COMMON / BLOC25 / VV(2000),CT(2000),TN(2000)
COMMON / BLOC26 / ANUMEDC(2000),ANUMMVC(2000),ANUMHCI(2000),
$ANUMCH(2000)
COMMON / BLOC27 /BDENEDC(2000),BDENMVC(2000),BDENHCI(2000),
$BDENCH(2000)
COMMON / BLOC28 /VISAMEDC(2000),VISAMMVC(2000),VISAMHCI(2000),
$VISAMCH(2000)
CO = SQRT(2.0)
DO 10 I = 2 , Q
Y(I) = X(I)
Y(I+Q) = X(I+Q)
Y(I+2*Q) = X(I+2*Q)
Y(I+3*Q) = X(I+3*Q)
10 CONTINUE

CALL FUNC
DO 20 I = 2 , Q
C(1,I) = PASSO*F(I)
X(I) = X(I) + C(1,I)/2
C(1,I+Q) = PASSO*F(I+Q)
X(I+Q) = X(I+Q) + C(1,I+Q)/2
C(1,I+2*Q) = PASSO*F(I+2*Q)
X(I+2*Q) = X(I+2*Q) + C(1,I+2*Q)/2
C(1,I+3*Q) = PASSO*F(I+3*Q)
X(I+3*Q) = X(I+3*Q) + C(1,I+3*Q)/2
20 CONTINUE

CALL FUNC
DO 30 I = 2 , Q
C(2,I) = PASSO*F(I)
X(I) = Y(I) + (CO-1.0)*C(1,I)/2.0 + (2.0-CO)*C(2,I)/2.0
C(2,I+Q) = PASSO*F(I+Q)
X(I+Q) = Y(I+Q) + (CO-1.0)*C(1,I+Q)/2.0 + (2.0-CO)*C(2,I+Q)/2.0
C(2,I+2*Q) = PASSO*F(I+2*Q)
X(I+2*Q) = Y(I+2*Q) + (CO-1.0)*C(1,I+2*Q)/2.0 + (2.0-CO)*C(2,I+2*Q)
$/2.0
C(2,I+3*Q) = PASSO*F(I+3*Q)
X(I+3*Q) = Y(I+3*Q) + (CO-1.0)*C(1,I+3*Q)/2.0 + (2.0-CO)*C(2,I+3*Q)
$/2.0
30 CONTINUE

CALL FUNC
DO 40 I = 2 , Q
C(3,I) = PASSO*F(I)
X(I) = Y(I) - CO*C(2,I)/2.0 + (CO/2.0 + 1.0)*C(3,I)
C(3,I+Q) = PASSO*F(I+Q)
X(I+Q) = Y(I+Q) - CO*C(2,I+Q)/2.0 + (CO/2.0 + 1.0)*C(3,I+Q)
C(3,I+2*Q) = PASSO*F(I+2*Q)
X(I+2*Q) = Y(I+2*Q) - CO*C(2,I+2*Q)/2.0 + (CO/2.0 + 1.0)*C(3,I+
$/2*Q)
C(3,I+3*Q) = PASSO*F(I+3*Q)
X(I+3*Q) = Y(I+3*Q) - CO*C(2,I+3*Q)/2.0 + (CO/2.0 + 1.0)*C(3,I+
$/3*Q)
40 CONTINUE

```

```

CALL FUNC
DO 50 I = 2 , Q
  C(4,I) = PASSO*F(I)
  X(I)=Y(I)+(C(1,I)+C(4,I))/6.0+(2.0-CO)*C(2,I)/6.0+(CO/2.0+1.0)*
$   C(3,I)/3.0
  C(4,I+Q) = PASSO*F(I+Q)
  X(I+Q)=Y(I+Q)+(C(1,I+Q)+C(4,I+Q))/6.0+(2.0-CO)*C(2,I+Q)/6.0+
$(CO/2.0+1.0)*C(3,I+Q)/3.0
  C(4,I+2*Q) = PASSO*F(I+2*Q)
  X(I+2*Q)=Y(I+2*Q)+(C(1,I+2*Q)+C(4,I+2*Q))/6.0+(2.0-CO)*C(2,I+2*
$Q)/6.0+(CO/2.0+1.0)*C(3,I+2*Q)/3.0
  C(4,I+3*Q) = PASSO*F(I+3*Q)
  X(I+3*Q)=Y(I+3*Q)+(C(1,I+3*Q)+C(4,I+3*Q))/6.0+(2.0-CO)*C(2,I+3*
$Q)/6.0+(CO/2.0+1.0)*C(3,I+3*Q)/3.0
  WRITE(*,*) F(I),F(I+Q),F(I+2*Q),F(I+3*Q)
50 CONTINUE

RETURN
END

```

---

```

*
*           SUBROTINA TUBO
*   USADA PARA CALCULAR AS DIMENSOES DO TUBO DA SERPENTINA
*

```

---

```

SUBROUTINE TUBO
IMPLICIT REAL*8 (A-H,O-Z)
COMMON / BLOC1 / NI,NC,NRE,DF
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO
COMMON / BLOC4 / CP(2000),CPM(2000),PMM(2000),AA(2000,2000)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(2000),HO(2000),HINT(2000),FI(2000,2000)
COMMON / BLOC8 / DCP1(2000),DCP2(2000)
COMMON / BLOC9 / A(2000),B(2000),PM(2000),RD(2000),RIN(2000),
$R1(2000),R2(2000)
COMMON / BLOC10 / AV(2000),BV(2000),CV(2000),VISA(2000)
COMMON / BLOC11 / AK(2000),BK(2000),CK(2000),DG(2000),Tw(2000)
COMMON / BLOC12 / UI(2000),ROM(2000),UNUD(2000)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(2000),TKM(2000),RE(2000),PR(2000),FF(2000)
$,FR(2000),VISAMCP(2000)
COMMON / BLOC15 / TL,TE,RE,RB,RBM,BL
COMMON / BLOC16 / AG(2000),BG(2000),CG(2000),TK(2000),FM(2000)
COMMON / BLOC17 / P(2000),Twp,CT1(2000),CT2(2000)
COMMON / BLOC18 / NT,NE,Q
COMMON / BLOC19 / V(2000),DELTAQ1(2000),DELTAQ2(2000),PROD(2000)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(2000),CS(2000),DGRE1(2000),DGRE2(2000)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA2(2000),RMULT(2000)
COMMON / BLOC23 / PMMEDC(2000),PMMMVC(2000),PMMHCI(2000),
$PMMCH(2000)
COMMON / BLOC24 / ANUM(2000),BDEN(2000),ALFA(2000),VISCM(2000)
COMMON / BLOC25 / VV(2000),CT(2000),TN(2000)
COMMON / BLOC26 / ANUMEDC(2000),ANUMMVC(2000),ANUMHCI(2000),
$ANUMCH(2000)
COMMON / BLOC27 /BDENEDC(2000),BDENMVC(2000),BDENHCI(2000),
$BDENCH(2000)
COMMON / BLOC28 /VISAMEDC(2000),VISAMMVC(2000),VISAMHCI(2000),
$VISAMCH(2000)

```

---

```

*
*   Calculo do raio do joelho em metros(RBM)
*
*   RBM = (RB/360.)*3.1416*(TE-DE)
*
*   Calculo do comprimento do joelho(BL) em metros
*
*   BL = 3.1416*TE/2.
*   NJ=NT-1
*   NE=(NT-1)+NJ
*
*   Calculo do comprimento total da serpentina
*
*   ZF=NT*TL+NJ*BL

```

```

*
* Gerando o vetor que indica a posição na serpentina
* se esta no joelho ou no tubo reto
*
V(1)=TL
IK=1
DO 13 I=2,NE+1
  IF(IK.EQ.1) GOTO 14
  V(I) = V(I-1) + TL
  IK=1
  GOTO 13
14 V(I) = V(I-1) + BL
  IK=0
13 CONTINUE
DO 100 I=1,NE+1
  WRITE(12,*) V(I)
100 CONTINUE
*
* Calculo do Diâmetro Interno(DI) e raios das camadas
*
RO= DE/2.
DI= DE-2.*EP
RT= DI/2.
RI= RT-EC
RETURN
END
*
* SUBROTINA FUNC
*
SUBROUTINE FUNC
IMPLICIT REAL*8 (A-H,O-Z)
COMMON / BLOC1 / NI,NC,NRE,DF
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO
COMMON / BLOC4 / CP(2000),CPM(2000),PMM(2000),AA(2000,2000)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(2000),HO(2000),HINT(2000),FI(2000,2000)
COMMON / BLOC8 / DCP1(2000),DCP2(2000)
COMMON / BLOC9 / A(2000),B(2000),PM(2000),RD(2000),RIN(2000),
$R1(2000),R2(2000)
COMMON / BLOC10 / AV(2000),BV(2000),CV(2000),VISA(2000)
COMMON / BLOC11 / AK(2000),BK(2000),CK(2000),DG(2000),Tw(2000)
COMMON / BLOC12 / UI(2000),ROM(2000),UNUD(2000)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(2000),TKM(2000),RE(2000),PR(2000),FF(2000)
$,FR(2000),VISAMCP(2000)
COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(2000),BG(2000),CG(2000),TK(2000),FM(2000)
COMMON / BLOC17 / P(2000),Twp,CT1(2000),CT2(2000)
COMMON / BLOC18 / NT,NE,Q
COMMON / BLOC19 / V(2000),DELTAQ1(2000),DELTAQ2(2000),PROD(2000)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(2000),CS(2000),DGRE1(2000),DGRE2(2000)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA2(2000),RMULT(2000)
COMMON / BLOC23 / PMMEDC(2000),PMMMVC(2000),PMMHCI(2000),
$PMMCH(2000)
COMMON / BLOC24 / ANUM(2000),BDEN(2000),ALFA(2000),VISC(2000)
COMMON / BLOC25 / VV(2000),CT(2000),TN(2000)
COMMON / BLOC26 / ANUMEDC(2000),ANUMMVC(2000),ANUMHCI(2000),
$ANUMCH(2000)
COMMON / BLOC27 / BDENEDC(2000),BDENMVC(2000),BDENHCI(2000),
$BDENCH(2000)
COMMON / BLOC28 / VISAMEDC(2000),VISAMMVC(2000),VISAMHCI(2000),
$VISAMCH(2000)
CALL PROP
CALL REYNOLDS
CALL TERMIC
GAMA1=G0
DO 1007 I=1,Q
  GAMA2(I) = (4.*HINT(I))/(G0*CPM(I)*DI)
1007 CONTINUE
DO 1008 I=1,Q

```

```

DCP1(I) = ((A(2*Q+1)+A(Q+1)-A(1)) * (X(I+3*Q)*TO-298) + ((B(2*Q+1)+
SB(Q+1)-B(1)) * ((X(I+3*Q)*TO)**2-298**2)) / 2)
DCP2(I) = ((A(2*Q+1)+A(5*Q+1)-A(Q+1)) * (X(I+3*Q)*TO-298) + ((B(2*Q+1)+
SB(5*Q+1)-B(Q+1)) * ((X(I+3*Q)*TO)**2-298**2)) / 2)
1008 CONTINUE
*      WRITE(*,*) X(1+3*Q),X(2+3*Q),X(3+3*Q),X(4+3*Q),X(5+3*Q)
*      WRITE(*,*)
DO 32 I = 1,Q
  CT1(I) = S1*DEXP(-EA1/(RG*X(I+3*Q)*TO))
  CT2(I) = S2*DEXP(-EA2/(RG*X(I+3*Q)*TO))
  DELTAQ1(I) = -(DELTAHO1+DCP1(I))/(G0*CPM(I))
  DELTAQ2(I) = -(DELTAHO2+DCP2(I))/(G0*CPM(I))
32 CONTINUE
DO 33 I = 1,Q
  R1(I) = CT1(I)*X(I)*ROM(I)/PM(I)
  R2(I) = CT2(I)*X(I+Q)*ROM(I)/PM(I+Q)
33 CONTINUE
DO 35 I = 1,Q
  RMULT(I) = DELTAQ1(I)*R1(I)+DELTAQ2(I)*R2(I)
35 CONTINUE
DO 100 J = 2,Q
*
*
*      EQUAÇÃO DO BALANÇO DE MASSA DO EDC
*
  F(J) = ((RT*ROME*(-X(J-1)+X(J)))/(ZF*DF*ROM(J)))+(RT*ROME*PM(1)*
$(-R1(J)))/(G0*ROM(J))
*
*      EQUAÇÃO DO BALANÇO DE MASSA PARA O MVC
*
  F(J+Q) = ((RT*ROME*(-X(J+Q-1)+X(J+Q)))/(ZF*DF*ROM(J)))+(RT*ROME*
$PM(Q+1)*(R1(J)-R2(J)))/(G0*ROM(J))
*
*      EQUAÇÃO DO BALANÇO DE MASSA PARA O HCI
*
  F(J+2*Q) = ((RT*ROME*(-X(J+2*Q-1)+X(J+2*Q)))/(ZF*DF*ROM(J)))+(RT*
$ROME*PM(2*Q+1)*(R1(J)+R2(J)))/(G0*ROM(J))
*
*      BALANÇO DE ENERGIA
*
  F(J+3*Q) = -(RT*ROME*(-X(J+3*Q-1)+X(J+3*Q)))/(ZF*DF*ROM(J))+(2*2.6*
$UI(J)*ROME*(Tw(J)-(X(J+3*Q)*TO))/(G0*CPM(J)*TO*ROM(J))+
$(RT*ROME*(RMULT(J)))/(G0*CPM(J)*TO*ROM(J))
100 CONTINUE

37 RETURN
END
*
*      SUBROTINA PROP
*      USADA PARA AS PROPRIEDADES DOS COMPONENTES E DA MISTURA GASOSA*
*
SUBROUTINE PROP
IMPLICIT REAL*8 (A-H,O-Z)
COMMON / BLOC1 / NI,NC,NRE,DF
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO
COMMON / BLOC4 / CP(2000),CPM(2000),PMM(2000),AA(2000,2000)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(2000),HO(2000),HINT(2000),FI(2000,2000)
COMMON / BLOC8 / DCP1(2000),DCP2(2000)
COMMON / BLOC9 / A(2000),B(2000),PM(2000),RD(2000),RIN(2000),
$R1(2000),R2(2000)
COMMON / BLOC10 / AV(2000),BV(2000),CV(2000),VISA(2000)
COMMON / BLOC11 / AK(2000),BK(2000),CK(2000),DG(2000),Tw(2000)
COMMON / BLOC12 / UI(2000),ROM(2000),UNUD(2000)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(2000),TKM(2000),RE(2000),PR(2000),FF(2000)
$,FR(2000),VISAMCP(2000)
COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(2000),BG(2000),CG(2000),TK(2000),FM(2000)
COMMON / BLOC17 / P(2000),Twp,CT1(2000),CT2(2000)
COMMON / BLOC18 / NT,NE,Q
COMMON / BLOC19 / V(2000),DELTAQ1(2000),DELTAQ2(2000),PROD(2000)

```

```

COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(2000),CS(2000),DGRE1(2000),DGRE2(2000)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA2(2000),RMULT(2000)
COMMON / BLOC23 / PMMEDC(2000),PMMMVC(2000),PMMHCI(2000),
SPMMCH(2000)
COMMON / BLOC24 / ANUM(2000),BDEN(2000),ALFA(2000),VISCM(2000)
COMMON / BLOC25 / VV(2000),CT(2000),TN(2000)
COMMON / BLOC26 / ANUMEDC(2000),ANUMMVC(2000),ANUMHCI(2000),
$ANUMCH(2000)
COMMON / BLOC27 /BDENEDC(2000),BDENMVC(2000),BDENHCI(2000),
$BDENCH(2000)
COMMON / BLOC28 /VISAMEDC(2000),VISAMMVC(2000),VISAMHCI(2000),
$VISAMCH(2000)

```

---

\* Calculo do No. de moles e fração molar dos componentes

---

```

DO 2 I=1,Q
RN(I) = X(I)*VM/PM(I)
RN(I+Q) = X(I+Q)*VM/PM(I+Q)
RN(I+2*Q) = X(I+2*Q)*VM/PM(I+2*Q)
RN(I+5*Q) = X(I+5*Q)*VM/PM(I+5*Q)
2 CONTINUE

DO 3 I=1,Q
TN(I) = RN(I)+RN(I+Q)+RN(I+2*Q)+RN(I+5*Q)
3 CONTINUE

```

---

\* Calculo da fração molar dos componentes

---

```

DO 5 I=1,Q
FM(I) = RN(I)/TN(I)
FM(I+Q) = RN(I+Q)/TN(I)
FM(I+2*Q) = RN(I+2*Q)/TN(I)
FM(I+5*Q) = RN(I+5*Q)/TN(I)
5 CONTINUE

```

---

\* Calculo do Peso Molecular Médio da Mistura

---

```

DO 7 I=1,Q
PMMEDC(I)= FM(I)*PM(I)
PMMMVC(I+Q)= FM(I+Q)*PM(I+Q)
PMMHCI(I+2*Q)=FM(I+2*Q)*PM(I+2*Q)
PMMCH(I+5*Q)=FM(I+5*Q)*PM(I+5*Q)
PMM(I)=PMMEDC(I)+PMMMVC(I+Q)+PMMHCI(I+2*Q)+PMMCH(I+5*Q)
WRITE(*,*) PMM(I)
7 CONTINUE

```

---

\* Calculo da Capacidade Calorífica dos Componentes e Media

---

```

DO 10 I = 1,Q
CP(I) = (A(I)+B(I)*X(I+3*Q)*TO)/PM(I)
CP(I+Q) = (A(I+Q)+B(I+Q)*X(I+3*Q)*TO)/PM(I+Q)
CP(I+2*Q) = (A(I+2*Q)+B(I+2*Q)*X(I+3*Q)*TO)/PM(I+2*Q)
CP(I+5*Q) = (A(I+5*Q)+B(I+5*Q)*X(I+3*Q)*TO)/PM(I+5*Q)
10 CONTINUE
DO 11 I=1,Q
CPM(I) =FM(I)*CP(I)+ FM(I+Q)*CP(I+Q)+FM(I+2*Q)*CP(I+2*Q)+FM(I+5
$*Q)*CP(I+5*Q)
11 CONTINUE

```

---

\* Calculo da Densidade Media da Mistura

---

```

DO 12 I=1,Q
VV(I)=(G0*0.082*1.023*X(I+3*Q)*TO)/(1./X(I+6*Q))/(X(I+4*Q)*
$23.33)
ROM(I)=G0/VV(I)
12 CONTINUE

```

---

\* Calculo da Concentração Total e dos Componentes

---

```

DO 13 I=1,Q
CT(I) =(X(I+4*Q)*23.33)/(0.082*1.023*X(I+3*Q)*TO)
13 CONTINUE

```

```

DO 14 I=1,Q
  CS(I) = FM(I)*CT(I)
  CS(I+Q) = FM(I+Q)*CT(I)
  CS(I+2*Q) = FM(I+2*Q)*CT(I)
  CS(I+5*Q) = FM(I+5*Q)*CT(I)
14 CONTINUE
*
*-----
*   Calculo da Viscosidade Absoluta de cada Componente
*-----
*
DO 15 I=1,Q
  VISA(I) = AV(I)*(X(I+3*Q)*TO)**2+BV(I)*X(I+3*Q)*TO+CV(I)
  VISA(I+Q) = AV(I+Q)*(X(I+3*Q)*TO)**2+BV(I+Q)*X(I+3*Q)*TO+CV(
  $I+Q)
  VISA(I+2*Q) = AV(I+2*Q)*(X(I+3*Q)*TO)**2+BV(I+2*Q)*X(I+3*Q)*
  $TO+CV(I+2*Q)
  VISA(I+5*Q) = AV(I+5*Q)*(X(I+3*Q)*TO)**2+BV(I+5*Q)*X(I+3*Q)*
  $TO+CV(I+5*Q)
15 CONTINUE
*
*-----
*   Calculo da Viscosidade Absoluta da Mistura
*-----
*
*-----
*   Calculo dos Coeficientes FI(I,J) da regra de mistura
*-----
DO 16 I=1,Q
  FI(I,I)=1
  FI(I+Q,I+Q)=1
  FI(I+2*Q,I+2*Q)=1
  FI(I+5*Q,I+5*Q)=1
16 CONTINUE
DO 1060 I=1,Q
  FI(I,I+Q) = ((1.+(VISA(I)/VISA(I+Q))**0.5)*(PM(I+Q)/PM(I))**0.25)
  $**2.)/((8.*(1+PM(I)/PM(I+Q)))**0.5)
  FI(I,I+2*Q) = ((1.+(VISA(I)/VISA(I+2*Q))**0.5)*(PM(I+2*Q)/PM(I))**
  $0.25)**2.)/((8.*(1+PM(I)/PM(I+2*Q)))**0.5)
  FI(I,I+5*Q) = ((1.+(VISA(I)/VISA(I+5*Q))**0.5)*(PM(I+5*Q)/PM(I))**
  $0.25)**2.)/((8.*(1+PM(I)/PM(I+5*Q)))**0.5)
1060 CONTINUE
DO 1061 I=1,Q
  FI(I+Q,I) = ((1.+(VISA(I+Q)/VISA(I))**0.5)*(PM(I)/PM(I+Q))**0.25)
  $**2.)/((8.*(1+PM(I+Q)/PM(I))**0.5)
  FI(I+Q,I+2*Q) = ((1.+(VISA(I+Q)/VISA(I+2*Q))**0.5)*(PM(I+2*Q)/PM(I+
  $Q))**0.25)**2.)/((8.*(1+PM(I+Q)/PM(I+2*Q)))**0.5)
  FI(I+Q,I+5*Q) = ((1.+(VISA(I+Q)/VISA(I+5*Q))**0.5)*(PM(I+5*Q)/PM(I+
  $Q))**0.25)**2.)/((8.*(1+PM(I+Q)/PM(I+5*Q)))**0.5)
1061 CONTINUE
DO 1062 I=1,Q
  FI(I+2*Q,I) = ((1.+(VISA(I+2*Q)/VISA(I))**0.5)*(PM(I)/PM(I+2*Q))**
  $0.25)**2.)/((8.*(1+PM(I+2*Q)/PM(I))**0.5)
  FI(I+2*Q,I+Q) = ((1.+(VISA(I+2*Q)/VISA(I+Q))**0.5)*(PM(I+Q)/PM(I+2*
  $Q))**0.25)**2.)/((8.*(1+PM(I+2*Q)/PM(I+Q)))**0.5)
  FI(I+2*Q,I+5*Q) = ((1.+(VISA(I+2*Q)/VISA(I+5*Q))**0.5)*(PM(I+5*Q)/
  $PM(I+2*Q))**0.25)**2.)/((8.*(1+PM(I+2*Q)/PM(I+5*Q)))**0.5)
1062 CONTINUE
DO 1063 I=1,Q
  FI(I+5*Q,I) = ((1.+(VISA(I+5*Q)/VISA(I))**0.5)*(PM(I)/PM(I+5*Q))**
  $0.25)**2.)/((8.*(1+PM(I+5*Q)/PM(I))**0.5)
  FI(I+5*Q,I+Q) = ((1.+(VISA(I+5*Q)/VISA(I+Q))**0.5)*(PM(I+Q)/PM(I+5*
  $Q))**0.25)**2.)/((8.*(1+PM(I+5*Q)/PM(I+Q)))**0.5)
  FI(I+5*Q,I+2*Q) = ((1.+(VISA(I+5*Q)/VISA(I+2*Q))**0.5)*(PM(I+2*Q)/
  $PM(I+5*Q))**0.25)**2.)/((8.*(1+PM(I+5*Q)/PM(I+2*Q)))**0.5)
1063 CONTINUE
*
*-----
*   Calculo da viscosidade absoluta usando a regra de mistura
*-----
*
DO 20 I=1,Q
  PROD(I) = FM(I)+FM(I+Q)*FI(I,I+Q)+FM(I+2*Q)*FI(I,I+2*Q)+FM(I+5*Q)*
  $FI(I,I+5*Q)
  PROD(I+Q) = FM(I)*FI(I+Q,I)+FM(I+Q)+FM(I+2*Q)*FI(I+Q,I+2*Q)+FM(I+5*
  $Q)*FI(I+Q,I+5*Q)
  PROD(I+2*Q) = FM(I)*FI(I+2*Q,I)+FM(I+Q)*FI(I+2*Q,I+Q)+FM(I+2*Q)+FM(I
  $+5*Q)*FI(I+2*Q,I+5*Q)
  PROD(I+5*Q) = FM(I)*FI(I+5*Q,I)+FM(I+Q)*FI(I+5*Q,I+Q)+FM(I+2*Q)*FI(I

```

```

$+5*Q,I+2*Q)+FM(I+5*Q)
20 CONTINUE
DO 21 I=1,Q
VISAMEDC(I) = FM(I)*VISA(I)/PROD(I)
VISAMMVC(I+Q) = FM(I+Q)*VISA(I+Q)/PROD(I+Q)
VISAMHCI(I+2*Q) = FM(I+2*Q)*VISA(I+2*Q)/PROD(I+2*Q)
VISAMCH(I+5*Q) = FM(I+5*Q)*VISA(I+5*Q)/PROD(I+5*Q)
21 CONTINUE
DO 22 I=1,Q
VISAM(I)=VISAMEDC(I)+VISAMMVC(I+Q)+VISAMHCI(I+2*Q)+VISAMCH(I+5*Q)
22 CONTINUE
DO 23 I=1,Q
VISAMCP(I)=VISAM(I)
23 CONTINUE
*----- CONVERTENDO A VISAM DE cp PARA Kg/m/h -----
DO 1070 I=1,Q
VISAM(I)= VISAM(I)*3.6
1070 CONTINUE
*-----
*          Calculo da Condutividade Térmica de cada Componente
*-----
DO 24 I=1,Q
TK(I) = AK(I)*(X(I+3*Q)*TO)**2+BK(I)*X(I+3*Q)*TO+CK(I)
TK(I+Q) = AK(I+Q)*(X(I+3*Q)*TO)**2+BK(I+Q)*X(I+3*Q)*TO+CK(I+Q)
TK(I+2*Q) = AK(I+2*Q)*(X(I+3*Q)*TO)**2+BK(I+2*Q)*X(I+3*Q)*TO+CK(I
$+2*Q)
TK(I+5*Q) = AK(I+5*Q)*(X(I+3*Q)*TO)**2+BK(I+5*Q)*X(I+3*Q)*TO+CK(I
$+5*Q)
24 CONTINUE
*-----
*          Calculo da Condutividade Térmica da Mistura
*-----
DO 25 I=1,Q
ANUMEDC(I) = FM(I)*TK(I)*PM(I)**(1/3.)
ANUMMVC(I+Q) = FM(I+Q)*TK(I+Q)*PM(I+Q)**(1/3.)
ANUMHCI(I+2*Q) = FM(I+2*Q)*TK(I+2*Q)*PM(I+2*Q)**(1/3.)
ANUMCH(I+5*Q) = FM(I+5*Q)*TK(I+5*Q)*PM(I+5*Q)**(1/3.)
BDENEDC(I) =FM(I)*PM(I)**(1/3.)
BDENMVC(I+Q) =FM(I+Q)*PM(I+Q)**(1/3.)
BDENHCI(I+2*Q) =FM(I+2*Q)*PM(I+2*Q)**(1/3.)
BDENCH(I+5*Q) =FM(I+5*Q)*PM(I+5*Q)**(1/3.)
25 CONTINUE
DO 26 I=1,Q
ANUM(I) = ANUMEDC(I)+ANUMMVC(I+Q)+ANUMHCI(I+2*Q)+ANUMCH(I+5*Q)
BDEN(I) =BDENEDC(I)+BDENMVC(I+Q)+BDENHCI(I+2*Q)+BDENCH(I+5*Q)
TKM(I)=ANUM(I)/BDEN(I)
26 CONTINUE
*-----
*          Calculo do Numero de Prandtl(PR) no trecho
*-----
DO 27 I=1,Q
ALFA(I) = TKM(I)/(ROM(I)*CPM(I))
27 CONTINUE
DO 28 I=1,Q
VISCM(I) = VISAM(I)/ROM(I)
IF(VISCM(I).LE.0) WRITE(*,152) VISCM(I)
152 FORMAT(/,8X,"VISCM (I)",F10.4)
PR(I) = VISCM(I)/ALFA(I)
28 CONTINUE
RETURN
END
*-----
*          SUBROTINA REYNOLDS
*          USADA PARA CALCULAR O FATOR DE FRICCAO DE FANNING
*-----
SUBROUTINE REYNOLDS
IMPLICIT REAL*8 (A-H,O-Z)
COMMON / BLOC1 / NI,NC,NRE,DF
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO
COMMON / BLOC4 / CP(2000),CPM(2000),PMM(2000),AA(2000,2000)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2

```

```

COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(2000),HO(2000),HINT(2000),FI(2000,2000)
COMMON / BLOC8 / DCP1(2000),DCP2(2000)
COMMON / BLOC9 / A(2000),B(2000),PM(2000),RD(2000),RIN(2000),
$R1(2000),R2(2000)
COMMON / BLOC10 / AV(2000),BV(2000),CV(2000),VISA(2000)
COMMON / BLOC11 / AK(2000),BK(2000),CK(2000),DG(2000),Tw(2000)
COMMON / BLOC12 / UI(2000),ROM(2000),UNUD(2000)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(2000),TKM(2000),RE(2000),PR(2000),FF(2000)
$,FR(2000),VISAMCP(2000)
COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(2000),BG(2000),CG(2000),TK(2000),FM(2000)
COMMON / BLOC17 / P(2000),Twp,CT1(2000),CT2(2000)
COMMON / BLOC18 / NT,NE,Q
COMMON / BLOC19 / V(2000),DELTAQ1(2000),DELTAQ2(2000),PROD(2000)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(2000),CS(2000),DGRE1(2000),DGRE2(2000)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA2(2000),RMULT(2000)
COMMON / BLOC23 / PMMEDC(2000),PMMMVC(2000),PMMHCI(2000),
$PMMCH(2000)
COMMON / BLOC24 / ANUM(2000),BDEN(2000),ALFA(2000),VISCM(2000)
COMMON / BLOC25 / VV(2000),CT(2000),TN(2000)
COMMON / BLOC26 / ANUMEDC(2000),ANUMMVC(2000),ANUMHCI(2000),
$ANUMCH(2000)
COMMON / BLOC27 / BDENEDC(2000),BDENMVC(2000),BDENHCI(2000),
$BDENCH(2000)
COMMON / BLOC28 / VISAMEDC(2000),VISAMMVC(2000),VISAMHCI(2000),
$VISAMCH(2000)
*
*      Calculo do Numero de Reynolds no escoamento
*
DO 1010 I=1,Q
RE(I) = G0*DI/VISAM(I)
1010 CONTINUE
*
*      CALCULO DO FATOR DE FRICCAO PARA ESCOAMENTO TURBULENTO
*
DO 1011 I=1,Q
FF(I)=0.046*(RE(I)**(-0.2))
1011 CONTINUE
*
*      Calculo do Numero de FR identificando o trecho - se reto ou
*      curvo - a partir do vetor posição V() gerado no programa principal
*
DO 1012 I=1,Q
FR(I) = 2.*FF(I)/DI
1012 CONTINUE
RETURN
END
*
*
*      SUBROTINA TERMIC
*      USADA PARA CALCULAR OS COEFICIENTES DE TROCA TERMICA
*
SUBROUTINE TERMIC
IMPLICIT REAL*8 (A-H,O-Z)
COMMON / BLOC1 / NI,NC,NRE,DF
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO
COMMON / BLOC4 / CP(2000),CPM(2000),PMM(2000),AA(2000,2000)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(2000),HO(2000),HINT(2000),FI(2000,2000)
COMMON / BLOC8 / DCP1(2000),DCP2(2000)
COMMON / BLOC9 / A(2000),B(2000),PM(2000),RD(2000),RIN(2000),
$R1(2000),R2(2000)
COMMON / BLOC10 / AV(2000),BV(2000),CV(2000),VISA(2000)
COMMON / BLOC11 / AK(2000),BK(2000),CK(2000),DG(2000),Tw(2000)
COMMON / BLOC12 / UI(2000),ROM(2000),UNUD(2000)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(2000),TKM(2000),RE(2000),PR(2000),FF(2000)
$,FR(2000),VISAMCP(2000)
COMMON / BLOC15 / TL,TE,RE,RBM,BL

```

```

COMMON / BLOC16 / AG(2000),BG(2000),CG(2000),TK(2000),FM(2000)
COMMON / BLOC17 / P(2000),Twp,CT1(2000),CT2(2000)
COMMON / BLOC18 / NT,NE,Q
COMMON / BLOC19 / V(2000),DELTAQ1(2000),DELTAQ2(2000),PROD(2000)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(2000),CS(2000),DGRE1(2000),DGRE2(2000)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA2(2000),RMULT(2000)
COMMON / BLOC23 / PMMEDC(2000),PMMMVC(2000),PMMHCI(2000),
$PMMCH(2000)
COMMON / BLOC24 / ANUM(2000),BDEN(2000),ALFA(2000),VISCN(2000)
COMMON / BLOC25 / VV(2000),CT(2000),TN(2000)
COMMON / BLOC26 / ANUMEDC(2000),ANUMMVC(2000),ANUMHCI(2000),
$ANUMCH(2000)
COMMON / BLOC27 /BDENEDC(2000),BDENMVC(2000),BDENHCI(2000),
$BDENCH(2000)
COMMON / BLOC28 /VISAMEDC(2000),VISAMMVC(2000),VISAMHCI(2000),
$VISAMCH(2000)

*
* -----
*   Calculo do Numero de Nusselt (NUD) no trecho
*   Equacao de Pethukov, Kirilov e Popov
*   Faixa: 0.5 < PR(I) < 2000, 10.000 < RED(I) < 5.000.000
* -----
DO 1015 I=1,Q
UNUD(I)=(FF(I)/2.)*RE(I)*PR(I)/(1.07+12.7*((FF(I)/2.)**0.5)*(PR(I)
$(2/3.)-1.))
HI(I) = UNUD(I)*TKM(I)/DI
HO(I) = HI(I)
1015 CONTINUE
*
* -----
*   Calculo do Coeficiente Global de troca UI - baseado em AInterna
* -----
DO 1016 I=1,Q
UI(I) = (1./RI)/(1./(RI*HI(I))+DLOG(RT/RI)/CKT+1./(RO*HO(I)))
1016 CONTINUE
*
* -----
*   Calculo do Coeficiente Interno hint pela equação do Kern
* -----
hint = 0.023*(RE**0.8)*(PR**0.4)*TKM/DI
DO 1017 I=1,Q
hint(I)=2.*UI(I)*1.30
1017 CONTINUE
RETURN
END
*
* -----
*   INTERFACE ROUTINE FROM DEC TO PC FOR GETTING TIME
*   SUBROTINA TEMPO
* -----
SUBROUTINE TEMPO(X)
*
* RETURN CURRENT TIME - MIDNIGHT - X
INTEGER*2 IHOURL, IMINUT, ISECON, IHUND
REAL*4  RHOUR, RMINUT, RSECON, RHUND
REAL*4  X1, X
CALL GETTIM(IHOURL, IMINUT, ISECON, IHUND)
RHOUR  = FLOAT( IHOURL )
RMINUT = FLOAT( IMINUT )
RSECON = FLOAT( ISECON )
RHUND  = FLOAT( IHUND )
X1     = RHOUR*3600.0 + RMINUT*60.0 + RSECON +
&      RHUND/100.0
X      = X1 - X
RETURN
END

```

### 3.2 - PROGRAMA PARA O CÁLCULO DAS RAÍZES DA COLOCAÇÃO ORTOGONAL E DOS COEFICIENTES A(I,J) - DMATA.FOR

```

INTERFACE TO FUNCTION SYSTEM[C]
+   (STRING)
  INTEGER*2 SYSTEM
  CHARACTER*1 STRING[REFERENCE]
  END

  IMPLICIT REAL*8(A-H,O-Z)
  INTEGER*2 SYSTEM
  DIMENSION A(30),A1(30),R1(30),GA(30,61),C(30,30),Q(30,30),X(30)
C
  I=SYSTEM('del mata.dat'//char(0))
  OPEN(UNIT=08,FILE='MATA.DAT',STATUS='NEW')
C
  WRITE(*,10)
10  FORMAT(///,3X,'DE O N. DE PONTOS INTERNOS (I2) : ')
  READ(*,20) N
20  FORMAT(I2)
  ND=N+2
C
  WRITE(*,30)
30  FORMAT(/,3X,'DE O VALOR DE ALFA : ')
  READ(*,*) ALF
  WRITE(*,50)
50  FORMAT(/,3X,'DE O VALOR DE BETA : ')
  READ(*,*) BET
C
  WRITE(*,60) ND,ALF,BET
60  FORMAT(//,3X,'ND = ',I2,3X,'ALFA = ',F5.2,3X,'BETA = ',F5.2,/,
* 3X,'PONTOS DE COLOCACAO : ',/)
C
  CALL COEPOT(N,ALF,BET,A)
  IP1=N+1
  I2=N/2
  A2=DSQRT(DABS(A(I2)))
  DO 70 J=1,IP1
70  A1(J)=A(J)/A2
C
  CALL GRAEF(A1,N,R1)
  CALL NEWHOR(A,N,R1)
C
  X(1)=0.0
  X(N+2)=1.0
  DO 80 I=1,N
80  X(I+1)=R1(I)
C
  WRITE(*,90) (X(I),I=1,ND)
90  FORMAT(3X,15(F11.8))
C
C   DEFINICAO DAS MATRIZES Q E C
C
  DO 100 I=1,ND
  DO 100 J=1,ND
  Q(J,I)=1.0
  IF(I.NE.1) Q(J,I)=X(J)**(I-1)
  C(J,I)=0.0
  IF(I.EQ.1) GO TO 100
  C(J,I)=(I-1)
  IF(I.NE.2) C(J,I)=(I-1)*X(J)**(I-2)
100  CONTINUE
C
C   INVERSAO DA MATRIZ Q - DEFINICAO DA MATRIZ A
C
  DO 110 I=1,ND
  DO 110 J=1,ND
  GA(I,J)=Q(I,J)
  IA=J+ND
  GA(I,IA)=0.0

```

```

      IF(I.EQ.J) GA(I,IA)=1.0
110  CONTINUE
C
C   CHAMADA DA SUBROTINA GAUSS
C
      NT2=ND+ND
      CALL GAUSS(ND,ND,NT2,GA)
C
C   MONTAGEM DA INVERSA DE Q
C
      DO 120 I=1,ND
      DO 120 J=1,ND
      IA=J+ND
      Q(I,J)=GA(I,IA)
120  CONTINUE
C
C   MULTIPLICACAO DE C X Q(-1)
C
      DO 140 I=1,ND
      DO 140 J=1,ND
      S=0.0
      DO 130 K=1,ND
130  S=S+C(I,K)*Q(K,J)
140  GA(I,J)=S
C
C   GRAVACAO DOS RESULTADOS NO DISQUETE - MATA.DAT
C
      WRITE(8,150) N
150  FORMAT(1X,I2)
      WRITE(8,160) ALF,BET
160  FORMAT(2(1X,F5.2))
      WRITE(8,170) (X(I),I=1,ND)
170  FORMAT(1X,15(F24.18))
      DO 190 I=1,ND
      WRITE(8,180) (GA(I,J),J=1,ND);
180  FORMAT(1X,15(F28.18))
190  CONTINUE
      END
C
C
C
C   SUBROUTINE COEPOT(N,ALF,BET,A)
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION A(30),C(30)
      DIMENSION IB(30,30)
C
      P=1.0
      NP1=N+1
      DO 10 I=1,NP1
      DO 10 J=1,NP1
10  IB(I,J)=0
      C(1)=1.0
      DO 20 J=1,N
      P=P*(NP1-J)*((NP1-J)+ALF)/J/(J+BET)
20  C(J+1)=P
      DO 30 I=1,NP1
      IB(I,I)=1
30  IB(1,I)=1
      DO 40 I=2,N
      DO 40 J=1,N
40  IB(I,J+1)=IB(I,J)+IB(I-1,J)
      DO 60 J=1,NP1
      SUM=0.0
      NP=N+2-J
      DO 50 I=1,NP
      NIP=NP+1-I
      IJ=I+J-1
50  SUM=SUM+C(NIP)*IB(J,IJ)
60  A(J)=SUM*(-1.0)**J
      RETURN
      END

```

```

SUBROUTINE NEWHOR(A1,N,R1)
C
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION X(30),CC(30),R1(30),S3(30),A1(30),R(30)
C
NT=N+1
DO 10 I=1,NT
X(I)=A1(I)
10  CC(I)=A1(I)
DO 20 J=1,N
NTJ=NT-J
20  S3(J)=R1(NTJ)
DO 30 J=1,N
30  R1(J)=S3(J)
S3(1)=CC(1)
A1(1)=CC(1)
DO 140 K=1,2
NK=NT
DO 120 I=1,N
40  DO 50 J=2,NK
S3(J)=S3(J-1)*R1(I)+X(J)
50  A1(J)=A1(J-1)*R1(I)+S3(J)
R(I)=R1(I)-S3(NK)/A1(NK-1)
IF(DABS(R1(I)-R(I))-5.0D-10) 70,70,60
60  R1(I)=R(I)
GO TO 40
70  IF (K-1) 80,80,120
80  NK=NK-1
IF(I-N+1) 100,90,90
90  R(N)=S3(2)/S3(1)
GO TO 130
100 DO 110 J=2,NK
110 X(J)=S3(J)
120 CONTINUE
130 DO 140 J=1,N
R1(J)=R(J)
140 X(J)=CC(J)
AK=R(1)
J=2
150 DO 170 I=J,N
IF(R(I)-AK) 160,160,170
160 AK=R(I)
R(I)=R(J-1)
R(J-1)=AK
170 CONTINUE
AK=R(J)
J=J+1
IF(J-N) 150,150,180
180 DO 190 I=1,N
190 R1(I)=R(I)
RETURN
END

```

C

C

C

```

SUBROUTINE GRAEF(A,N,R1)
C
IMPLICIT REAL*8(A-H,O-Z)
DIMENSION R(30),B(30),A(30),R1(30)
C
KL=1
NT=N+1
DO 10 M=1,N
10  R(M)=6.0
DO 150 M=1,40
IF(KL) 160,160,20
20  DO 30 I=2,NT
30  B(I)=A(I-1)*A(I-1)
DO 70 I=2,NT
IP=I
IM=I-2
AK=1.0
40  IF(IM)70,70,50

```

```

50 IF(IP-NT)60,60,70
60 B(I)=B(I)-2.0*A(IM)*A(IP)*AK
   IM=IM-1
   IP=IP+1
   AK=-AK
   GO TO 40
70 CONTINUE
   DO 80 I=2,NT
80 A(I-1)=B(I)
   A(NT)=A(NT)*A(NT)
   DO 100 I=1,NT
   IF(DABS(A(I))-1.0D-18)90,90,100
90 KL=0
100 CONTINUE
   POT=2.0**M
   DO 110 I=1,N
110 R1(I)=(A(I+1)/A(I))**(1.0/POT)
   DO 120 J=1,N
   IF(DABS(R(J)/R1(J)-1.0)-1.0D-09) 120,120,130
120 CONTINUE
   GO TO 160
130 DO 140 J=1,N
140 R(J)=R1(J)
150 CONTINUE
160 RETURN
END
C     A SUBROTINA GAUSS RESOLVE SISTEMA DE EQUACOES ALGEBRICAS
C     LINEARES PELO METODO DE ELIMINACAO DE GAUSS COM PIVOTA-
C     CAO DE LINHA. PARA RESOLVER O PROBLEMA QX=U ONDE Q E UMA
C     MATRIZ (N X N) E U (N X NS), O PROGRAMA USA UMA MATRIZ
C     A QUE E FORMADA POR Q NAS SUAS PRIMEIRAS N COLUNAS E POR
C     U NAS SEGUINTE NS COLUNAS. A SOLUCAO DO SISTEMA RETORNA
C     NA POSICAO DE U NA MATRIZ A.
C     ENTRAR COM O VALOR DE ND QUE E O NUMERO DE LINHAS DA MA-
C     TRIZ A (=N) E NCOL O NUMERO DE COLUNAS DA MATRIZ A QUE
C     E = A N+NS.
C
C     SUBROUTINE GAUSS(N,NS,NCOL,A)
C
C     IMPLICIT REAL*8(A-H,O-Z)
C     DIMENSION A(30,61)
C     N1 = N + 1
C     NT = N + NS
C     IF (N.EQ.1) GO TO 50
C     DO 40 I=2,N
C     IP = I - 1
C     I1 = IP
C     X = DABS(A(I1,I1))
C     DO 10 J=I,N
C     IF(DABS(A(J,I1)).LT.X) GO TO 10
C     X = DABS(A(J,I1))
C     IP = J
10 CONTINUE
   IF (IP.EQ.I1) GO TO 30
   DO 20 J=I1,NT
   X = A(I1,J)
   A(I1,J) = A(IP,J)
20 A(IP,J) = X
30 DO 40 J=I,N
   X = A(J,I1) / A(I1,I1)
   DO 40 K=I,NT
40 A(J,K) = A(J,K) - X * A(I1,K)
50 DO 70 IP=1,N
   I = N1 - IP
   DO 70 K=N1,NT
   A(I,K) = A(I,K) / A(I,I)
   IF (I.EQ.1) GO TO 70
   I1 = I-1
   DO 60 J=1,I1
60 A(J,K) = A(J,K) - A(I,K) * A(J,I)
70 CONTINUE
   RETURN
   END

```

### 3.3 – MÉTODO DA COLOCAÇÃO ORTOGONAL

\$ DEBUG

```

*
*-----*
*           COLOCACAO ORTOGONAL.FOR           *
*-----*
* MODELAGEM DINÂMICA E SIMULAÇÃO DE UMA FORNALHA PARA O CRAQUEAMENTO *
* TÉRMICO DO 1,2-DICLOROETANO. *
* RESOLUÇÃO DAS EQ. DIFERENCIAIS PELO MÉTODO DE COLOCAÇÃO ORTOGONAL *
* COM 3 PONTOS INTERNOS E RUNGE-KUTTA-GILL DE QUARTA ORDEM *
*-----*
* MODELO UNIDIMENSIONAL, HOMOGENEO, ESCOAMENTO PISTÃO, *
* SEM DISPERSÃO AXIAL DE CALOR E MASSA. *
*-----*
*           ESQUEMA DA REAÇÃO CONSIDERADO: *
*           k1 *
*           EDC -----> MVC + HCl *
*           1         5 6 *
*           k2 *
*           MVC -----> C2H2 + HCl *
*           5         7 6 *
*-----*
* MODELO CINÉTICO: VIA REAÇÕES MOLECULARES *
*-----*
* BALANÇO DE MASSA DO 1,2-DICLOROETANO (EDC) *
*-----*
* BALANÇO DE MASSA DO MONOCLORETO DEE VINILA (MVC) *
*-----*
* BALANÇO DE MASSA DO ÁCIDO CLORÍDRICO (HCl) *
*-----*
* BALANÇO DE ENERGIA *
*-----*
* RESOLVENDO POR RUNGE-KUTTA-GILL, USAREMOS PARA CADA EQUACAO *
* DIFERENCIAL *
*-----*
* Y(N+1) = Y(N) + 1/6(K1+K4) + 1/3(B*K2 + D*K3) *
*-----*
* K1 = h * f(Z(N),Y(N)) *
* K2 = h * f(Z(N)+1/2 h,Y(N)+1/2 K1) *
* K3 = h * f(Z(N)+1/2 h,Y(N)+A*K1+B*K2) *
* K4 = h * f(Z(N)+h,Y(N)+C*K2+D*K3) *
*-----*
* COM: A=(SQRT(2)-1)/2; B = (2-SQRT(2))/2; C = -SQRT(2)/2; *
*       D = (2+SQRT(2))/2 *
*-----*
* SENDO QUE A,B,C e D SAO USADAS PARA MINIMIZAR OS ERROS DE ARREDONDAMENTO *
*-----*
* CONSIDEROU-SE QUE NO ESCOAMENTO HA A VARIACAO AXIAL DO COEFICIENTE *
* DE PELICULA INTERNO(HO), ISTO E DO NUD *
*-----*
*
*-----*
* IMPLICIT REAL*8 (A-H,O-Z) *
* DOUBLE PRECISION CA(100,100) *
* DOUBLE PRECISION SEL(500),CONV(500),FQR(500),Z,ZP(100) *
* DOUBLE PRECISION XX(30) *
* COMMON / BLOC1 / NI,NC,NRE *
* COMMON / BLOC2 / PT,RG,FA,CKT,IG *
* COMMON / BLOC3 / G0,VM,ROME,W10,TO,YPI,YPMI,YTI *
* COMMON / BLOC4 / CP(500),CPM(500),PMM(500),AA(500,500) *
* COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2 *
* COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO *
* COMMON / BLOC7 / HI(500),HO(500),HINT(500),FI(500,500) *
* COMMON / BLOC8 / DCP1(500),DCP2(500) *
* COMMON / BLOC9 / A(500),B(500),PM(500),RD(500),RIN(500), *
* $R1(500),R2(500) *
* COMMON / BLOC10 / AV(500),BV(500),CV(500),VISA(500)

```

```

COMMON / BLOC11 / AK(500),BK(500),CK(500),DG(500),Tw(50)
COMMON / BLOC12 / UI(500),ROM(500),UNUD(500)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(500),TKM(500),RE(500),PR(500),
$FF(500),FR(500),VISAMCP(500)
COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(500),BG(500),CG(500),TK(500),FM(500)
COMMON / BLOC17 / P(500),Twp,CT1(500),CT2(500)
COMMON / BLOC18 / NT,NE,MR1,MR2
COMMON / BLOC19 / V(500),DELTAQ1(500),DELTAQ2(500),PROD(500)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(500),CS(500),DGRE1(500),DGRE2(500)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA1(500),GAMA2(500),
$GAMA2E(500),GAMA3E(500),RMULT(500)
COMMON / BLOC23 / PMMEDC(500),PMMMVC(500),PMMHCI(500),PMMCH(500)
COMMON / BLOC24 / ANUM(500),BDEN(500),ALFA(500),VISCM(500)
COMMON / BLOC25 / VV(500),CT(500),TN(500)
COMMON / BLOC26/ANUMEDC(500),ANUMMVC(500),ANUMHCI(500),ANUMCH(500)
COMMON / BLOC27 / BDENEDC(500),BDENMVC(500),BDENHCI(500),
$BDENCH(500)
COMMON / BLOC28 / VISAMEDC(500),VISAMMVC(500),VISAMHCI(500),
$VISAMCH(500)
COMMON / BLOC29 / CA
REAL ITEMP
* arquivo CORK1.DAD contem o numero de pontos de colocacao, os pontos de colocacao
* propriamente ditos e os coeficientes da matriz A(i,j)
OPEN(UNIT=10,FILE='CORK1.DAD',STATUS='OLD')
* arquivo de resultados
OPEN(UNIT=12,FILE='VETOR.DAT',STATUS='UNKNOWN')
OPEN(UNIT=9,FILE='RESULT.DAT',STATUS='UNKNOWN')
OPEN(UNIT=8,FILE='8.DAT',STATUS='UNKNOWN')
OPEN(UNIT=7,FILE='7.DAT',STATUS='UNKNOWN')
READ(10,*) MR
MR1 = MR + 1
MR2 = MR + 2
READ(10,*)(XX(I),I=1,MR2)
* fazendo a mudanca dos indices das raizes para 1 a nc+1 - o primeiro ponto sempre é zero.
DO 9 I=1,MR1
XX(I)=XX(I+1)
9 CONTINUE
*
* leitura dos coeficientes da matriz A(i,j), CA(i,j)
DO 10 I=1,MR2
READ(10,*)(CA(I,J),J=1,MR2)
WRITE(*,*)(CA(I,J),J=1,MR2)
10 CONTINUE
NI=6
NC=4
NRE=2
NP=10
PASSO=0.05
ZF=337.2579
* tempo máximo de processamento - ZM=TF
ZM=7000
YT1=0.503
TI=503.
TO=1000.
G0=2208679.4
VM=30842.0
RG=1.987
NT=20
TL=16.46
TE=0.270
RB=180.
DE=0.1524
EP=0.00953
EC=0.000
CKT=3.
PMEDC=99.0
PMMVC=62.5
PMHCI=36.5
PMCH=26.0
DO 1000 I=1,MR2

```

```
PM(I)=PMEDC
PM(I+MR2)=PMMVC
PM(I+2*MR2)=PMHCI
PM(I+5*MR2)=PMCH
1000 CONTINUE
AEDC=4.363
AMVC=1.419
AHCI=7.000
ACH=3.999
DO 11 I=1,MR2
A(I)=AEDC
A(I+MR2)=AMVC
A(I+2*MR2)=AHCI
A(I+5*MR2)=ACH
11 CONTINUE
BEDC=0.046856
BMVC=0.048080
BHCI=0.00006
BCH=0.030168
DO 207 I=1,MR2
B(I)=BEDC
B(I+MR2)=BMVC
B(I+2*MR2)=BHCI
B(I+5*MR2)=BCH
207 CONTINUE
AVEDC=-4.0E-9
AVMVC=4.0E-9
AVHCI=-2.0E-8
AVCH=4.0E-9
DO 12 I=1,MR2
AV(I)=AVEDC
AV(I+MR2)=AVMVC
AV(I+2*MR2)=AVHCI
AV(I+5*MR2)=AVCH
12 CONTINUE
BVEDC=3.0E-5
BVMVC=2.0E-5
BVHCI=7.0E-5
BVCH=2.0E-5
DO 200 I=1,MR2
BV(I)=BVEDC
BV(I+MR2)=BVMVC
BV(I+2*MR2)=BVHCI
BV(I+5*MR2)=BVCH
200 CONTINUE
CVEDC=0.002
CVMVC=0.0057
CVHCI=-0.0072
CVCH=0.0057
DO 201 I=1,MR2
CV(I)=CVEDC
CV(I+MR2)=CVMVC
CV(I+2*MR2)=CVHCI
CV(I+5*MR2)=CVCH
201 CONTINUE
AKEDC=-3.0E-11
AKMVC=-1.0E-8
AKHCI=2.0E-8
AKCH=-3.0E-11
DO 13 I=1,MR2
AK(I)=AKEDC
AK(I+MR2)=AKMVC
AK(I+2*MR2)=AKHCI
AK(I+5*MR2)=AKCH
13 CONTINUE
BKEDC=4.0E-5
BKMVC=6.0E-5
BKHCI=3.0E-5
BKCH=4.0E-5
DO 202 I=1,MR2
BK(I)=BKEDC
BK(I+MR2)=BKMVC
BK(I+2*MR2)=BKHCI
```

```

BK(I+5*MR2)=BKCH
202 CONTINUE
CKEDC=-0.0046
CKMVC=-0.007
CKHCI=0.0025
CKCH=-0.0046
DO 203 I=1,MR2
CK(I)=CKEDC
CK(I+MR2)=CKMVC
CK(I+2*MR2)=CKHCI
CK(I+5*MR2)=CKCH
203 CONTINUE
AGEDC=-130.549
AGMVC=31.557
AGHCI=-92.209
AGCH=227.187
DO 14 I=1,MR2
AG(I)=AGEDC
AG(I+MR2)=AGMVC
AG(I+2*MR2)=AGHCI
AG(I+5*MR2)=AGCH
14 CONTINUE
BGEDC=1.8562E-01
BGMVC=3.2864E-02
BGHCI=-1.1226E-02
BGCH=-6.1675E-02
DO 204 I=1,MR2
BG(I)=BGEDC
BG(I+MR2)=BGMVC
BG(I+2*MR2)=BGHCI
BG(I+5*MR2)=BGCH
204 CONTINUE
CGEDC=1.3775E-05
CGMVC=1.8788E-05
CGHCI=2.6966E-06
CGCH=4.3960E-06
DO 205 I=1,MR2
CG(I)=CGEDC
CG(I+MR2)=CGMVC
CG(I+2*MR2)=CGHCI
CG(I+5*MR2)=CGCH
205 CONTINUE
S1=3.600E13
EA1=39600.
DELTAHO1=13149.
S2=3.4812E14
EA2=49500.
DELTAHO2=27150.
* ***** VALOR INICIAL DE Temperatura de Skin *****
* ***** Leitura da temperatura de conveccao *****
TC=503.

* ***** Leitura dos pontos de temperatura na serpentina *****
TWP1=643.
TWP2=783.
TWP3=825.
TWP4=830.5

*
* _____
* TSKIN *
* USADA PARA O PERFIL DE TEMPERATURA DE PAREDE NO TUBO *
* _____
*
* PERFIL DE TEMPERATURA DE PAREDE (TW) BASEADO EM DADOS DE PROCESSO
*
* _____
* Perfil de temperatura do ultimo ponto da conveccao ate o primeiro
* ponto da serpentina - TC ate TWP(1) - 4 tubos de serpentina *****
*
* _____
* OPEN(UNIT=6,FILE='6.DAT',STATUS='UNKNOWN')
* _____ VALOR INICIAL DE Temperatura de Skin _____
* _____ Leitura da temperatura de conveccao _____

```

```

DTW1 = (TWP1-TC)/3.
DTW2 = (TWP2-TWP1)/4.
DTW3 = (TWP3-TWP2)/6.
DTW4 = (TWP4-TWP3)/6.
Z=0
K=1
Twp=TC
Tw(K)=Twp
write(*,*) K,Z,Tw(K)
DO 30 J=1,MR1
  Z=XX(J)*ZF
  ZP(J)=Z
  IF (Z.GT.337.5) GOTO 26
  K=K+1

```

\*    SECAO 1 DA SERPENTINA

```

IF(Z.LE.16.46) THEN
  Twp=TC
  END IF
IF ((Z.GT.16.46).AND.(Z.LT.16.88)) THEN
  Twp= TC+0.5*DTW1
  END IF
IF ((Z.GE.16.88).AND.(Z.LE.33.34)) THEN
  Twp= TC+1.0*DTW1
  END IF
IF ((Z.GT.33.34).AND.(Z.LT.33.76)) THEN
  Twp= TC+1.5*DTW1
  END IF
IF ((Z.GE.33.76).AND.(Z.LE.50.22)) THEN
  Twp= TC+2.0*DTW1
  END IF
IF ((Z.GT.50.22).AND.(Z.LT.50.64)) THEN
  Twp= TC+2.5*DTW1
  END IF
IF (Z.EQ.50.64) THEN
  Twp= TC+3.0*DTW1
  END IF

```

\*    SECAO 2 DA SERPENTINA

```

IF ((Z.GT.50.64).AND.(Z.LE.67.10)) THEN
  Twp= TWP1
  END IF
IF ((Z.GT.67.10).AND.(Z.LT.67.50)) THEN
  Twp= TWP1+0.5*DTW2
  END IF
IF ((Z.GE.67.50).AND.(Z.LE.83.98)) THEN
  Twp= TWP1+1.0*DTW2
  END IF
IF ((Z.GT.83.98).AND.(Z.LT.84.40)) THEN
  Twp= TWP1+1.5*DTW2
  END IF
IF ((Z.GE.84.40).AND.(Z.LE.100.87)) THEN
  Twp= TWP1+2.0*DTW2
  END IF
IF ((Z.GT.100.87).AND.(Z.LT.101.29)) THEN
  Twp= TWP1+2.5*DTW2
  END IF
IF ((Z.GE.101.29).AND.(Z.LE.117.75)) THEN
  Twp= TWP1+3.0*DTW2
  END IF
IF ((Z.GT.117.75).AND.(Z.LT.118.18)) THEN
  Twp= TWP1+3.5*DTW2
  END IF
IF (Z.EQ.118.18) THEN
  Twp= TWP1+4.0*DTW2
  END IF

```

\*    SECAO 3 DA SERPENTINA

```

IF ((Z.GT.118.18).AND.(Z.LE.134.64)) THEN
  Twp=TWP2
  END IF
IF ((Z.GT.134.64).AND.(Z.LT.135.06)) THEN

```

```

    Twp= TWP2+0.5*DTW3
  END IF
  IF ((Z.GE.135.06).AND.(Z.LE.151.52)) THEN
    Twp= TWP2+1.0*DTW3
  END IF
  IF ((Z.GT.151.52).AND.(Z.LT.151.95)) THEN
    Twp= TWP2+1.5*DTW3
  END IF
  IF ((Z.GE.151.95).AND.(Z.LE.168.41)) THEN
    Twp= TWP2+2.0*DTW3
  END IF
  IF ((Z.GT.168.41).AND.(Z.LT.168.83)) THEN
    Twp= TWP2+2.5*DTW3
  END IF
  IF ((Z.GE.168.83).AND.(Z.LE.185.29)) THEN
    Twp= TWP2+3.0*DTW3
  END IF
  IF ((Z.GT.185.29).AND.(Z.LT.185.72)) THEN
    Twp= TWP2+3.5*DTW3
  END IF
  IF ((Z.GT.185.72).AND.(Z.LT.202.18)) THEN
    Twp= TWP2+4.0*DTW3
  END IF
  IF ((Z.GE.202.18).AND.(Z.LE.202.60)) THEN
    Twp= TWP2+4.5*DTW3
  END IF
  IF ((Z.GT.202.60).AND.(Z.LT.219.06)) THEN
    Twp= TWP2+5.0*DTW3
  END IF
  IF ((Z.GT.219.06).AND.(Z.LT.219.48)) THEN
    Twp= TWP2+5.5*DTW3
  END IF
  IF (Z.EQ.219.48) THEN
    Twp= TWP2+6.0*DTW3
  END IF

```

\* SECAO 4 DA SERPENTINA

```

  IF ((Z.GT.219.48).AND.(Z.LE.235.94)) THEN
    Twp=TWP3
  END IF
  IF ((Z.GT.235.94).AND.(Z.LT.236.36)) THEN
    Twp= TWP3+0.5*DTW4
  END IF
  IF ((Z.GE.236.36).AND.(Z.LE.252.82)) THEN
    Twp= TWP3+1.0*DTW4
  END IF
  IF ((Z.GT.252.82).AND.(Z.LT.253.25)) THEN
    Twp= TWP3+1.5*DTW4
  END IF
  IF ((Z.GE.253.25).AND.(Z.LE.269.71)) THEN
    Twp= TWP3+2.0*DTW4
  END IF
  IF ((Z.GT.269.71).AND.(Z.LT.270.13)) THEN
    Twp= TWP3+2.5*DTW4
  END IF
  IF ((Z.GE.270.13).AND.(Z.LE.286.59)) THEN
    Twp= TWP3+3.0*DTW4
  END IF
  IF ((Z.GT.286.59).AND.(Z.LT.287.02)) THEN
    Twp= TWP3+3.5*DTW4
  END IF
  IF ((Z.GT.287.02).AND.(Z.LT.303.48)) THEN
    Twp= TWP3+4.0*DTW4
  END IF
  IF ((Z.GE.303.48).AND.(Z.LE.303.90)) THEN
    Twp=TWP3+4.5*DTW4
  END IF
  IF ((Z.GT.303.90).AND.(Z.LT.320.36)) THEN
    Twp= TWP3+5.0*DTW4
  END IF
  IF ((Z.GT.320.36).AND.(Z.LT.320.78)) THEN
    Twp= TWP3+5.5*DTW4
  END IF

```

```

      END IF
      IF ((Z.GT.320.78).AND.(Z.LE.337.2579)) THEN
          Twp= TWP3+6.0*DTW4
      END IF

      Tw(K)=Twp
      write(*,*) K,Z,Tw(K)
30 CONTINUE
26   write(6,*) K

```

---

```

*
*   VALORES INICIAIS
* atribuição de valores iniciais no ponto 1
      Y(1)=0.9950
      Y(MR2+1)=0.0005
      Y(2*MR2+1)=0.0005
      Y(3*MR2+1)=503./1000.
      Y(4*MR2+1)=23.33/23.33
      Y(6*MR2+1)=0.01010101
*
*   atribuição de valores para o ponto 2 em diante da serpentina em t=0
      DO 16 I= 1,MR2
          X(I) = Y(1)
          X(I+MR2) = Y(MR2+1)
          X(I+2*MR2) = Y(2*MR2+1)
          X(I+3*MR2) = Y(3*MR2+1)
          X(I+4*MR2) = Y(4*MR2+1)
          X(I+6*MR2) = Y(6*MR2+1)
      16 CONTINUE

```

---

```

***** VALOR INICIAL DE FRACAO MASSICA DE EDC *****
      W10 =0.9950
      Y1A = Y(1)
      Y9A = Y(MR2+1)
      TO = 1000.
      ROME = Y(4*MR2+1)*23.33*PM(1)/(0.082*1.023*Y(3*MR2+1)*1000)
      CALL TUBO
      CONT=1
          ZM=7000
          NG=ZM/PASSO
          DO 100 TEM=PASSO,ZM,PASSO
              CALL RKG
          IF(CONT.NE.(NG/NP)) GOTO 130
          DO 1003 I=1,MR2
              CONV(I) = (Y(1)-X(I))/Y(1)
          1003 CONTINUE
          DO 1052 I=2,MR2
              SEL(I) = ((X(I+MR2)-Y9A)/(Y1A-X(I)))*PM(I)/PM(I+MR2)
          1052 CONTINUE
          DO 1004 I=1,MR2
              X(I+5*MR2)=1.-X(I)-X(I+MR2)-X(I+2*MR2)
          1004 CONTINUE
***** CÁLCULO DO TEMPO DE RESIDENCIA NA SERPENTINA *****
      C1=X(MR2)*(PMM(MR2)/PM(MR2))*(X(5*MR2)*23.33)/(0.082*(X(4*MR2)*
      $TO))
      F1=X(MR2)*VM/PM(MR2)
      TR=(3.1416*DI**2./4.)*ZF*3600./(F1/C1)
*****
      TEMP=(TEM*RT*ROME)/G0
*****
*****IMPRESSÃO DOS RESULTADOS OBTIDOS*****
      WRITE(*,150) TEM,TEMP
      WRITE (8,150) TEM,TEMP
      WRITE (7,150) TEM,TEMP
      WRITE (9,150) TEM,TEMP

      WRITE(9,18)
      18 FORMAT(8X,'PTO COLO.',2X,'COMP',5X,'YEDC',5X,'YMVC',5X,'YHCI',
      $5X,'YC2H2',5X,'T')
      150 FORMAT(/,8X,'TEM=',F10.4,4X,'TEMP=',F12.6)
*****IMPRESSÃO DOS RESULTADOS OBTIDOS*****
      DO 250 I=1,MR2
          WRITE(*,300) (X(J),J=I,I+6*MR2,MR2)
          WRITE(8,300) (X(J),J=I,I+6*MR2,MR2)
      300 FORMAT(5X,F12.7,8F14.7,/)

```

```

WRITE(7,500) SEL(I),CONV(I),TR,UI(I),HI(I),hint(I)
500 FORMAT(5X,F12.7,8F14.7,/)
250 CONTINUE
CONT=0
130 CONT=CONT+1
100 CONTINUE
*****
DO 1050 I=1,MR2
FQR(I)=-HINT(I)*(X(I+3*MR2)*TO-Tw(I))
1050 CONTINUE
CALL TEMPO (ITEMP)
WRITE(*,*)
WRITE(*,*) 'Tempo de execucao :',ITEMP
STOP
END
*****
*
*
* SUBROTINA DE INTEGRACAO RUNGE-KUTTA-GILL DE QUARTA ORDEM
*
*****

SUBROUTINE RKG
IMPLICIT REAL*8 (A-H,O-Z)
DOUBLE PRECISION CA(100,100)
DOUBLE PRECISION C(4,100)
COMMON / BLOC1 / NI,NC,NRE
COMMON / BLOC2 / PT,RG,FA,CKT,IG,
COMMON / BLOC3 / G0,VM,ROME,W10,TO,YPI,YPMI,YTI
COMMON / BLOC4 / CP(500),CPM(500),PMM(500),AA(500,500)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(500),HO(500),HINT(500),FI(500,500)
COMMON / BLOC8 / DCP1(500),DCP2(500)
COMMON / BLOC9 / A(500),B(500),PM(500),RD(500),RIN(500),
SR1(500),R2(500)
COMMON / BLOC10 / AV(500),BV(500),CV(500),VISA(500)
COMMON / BLOC11 / AK(500),BK(500),CK(500),DG(500),Tw(50)
COMMON / BLOC12 / UI(500),ROM(500),UNUD(500)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(500),TKM(500),RE(500),PR(500),FF(500),
$FR(500),VISAMCP(500)
COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(500),BG(500),CG(500),TK(500),FM(500)
COMMON / BLOC17 / P(500),Twp,CT1(500),CT2(500)
COMMON / BLOC18 / NT,NE,MR1,MR2
COMMON / BLOC19 / V(500),DELTAQ1(500),DELTAQ2(500),PROD(500)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(500),CS(500),DGRE1(500),DGRE2(500)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA1(500),GAMA2(500),
$GAMA2E(500),GAMA3E(500),RMULT(500)
COMMON / BLOC23 / PMMEDC(500),PMMMVC(500),PMMHCI(500),PMMCH(500)
COMMON / BLOC24 / ANUM(500),BDEN(500),ALFA(500),VISCM(500)
COMMON / BLOC25 / VV(500),CT(500),TN(500)
COMMON / BLOC26 / ANUMEDC(500),ANUMMVC(500),ANUMHCI(500),ANUMCH(500)
COMMON / BLOC27 / BDENEDC(500),BDENMVC(500),BDENHCI(500),BDENCH(500)
COMMON / BLOC28 / VISAMEDC(500),VISAMMVC(500),VISAMHCI(500),
$VISAMCH(500)
COMMON / BLOC29 / CA
CO = SQRT(2.0)
DO 10 I = 2 , MR2
Y(I) = X(I)
Y(I+MR2)= X(I+MR2)
Y(I+2*MR2)=X(I+2*MR2)
Y(I+3*MR2)=X(I+3*MR2)
10 CONTINUE
CALL FUNC
DO 20 I = 2 , MR2
C(1,I) = PASSO*F(I)
X(I) = X(I) + C(1,I)/2
C(1,I+MR2) = PASSO*F(I+MR2)
X(I+MR2) = X(I+MR2) + C(1,I+MR2)/2
C(1,I+2*MR2) = PASSO*F(I+2*MR2)
X(I+2*MR2) = X(I+2*MR2) + C(1,I+2*MR2)/2

```



```

COMMON / BLOC15 / TL,TE,RE,RBM,BL
COMMON / BLOC16 / AG(500),BG(500),CG(500),TK(500),FM(500)
COMMON / BLOC17 / P(500),Twp,CT1(500),CT2(500)
COMMON / BLOC18 / NT,NE,MR1,MR2
COMMON / BLOC19 / V(500),DELTAQ1(500),DELTAQ2(500),PROD(500)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(500),CS(500),DGRE1(500),DGRE2(500)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA1(500),GAMA2(500),
$GAMA2E(500),GAMA3E(500),RMULT(500)
COMMON / BLOC23 / PMMEDC(500),PMMMVC(500),PMMHCI(500),PMMCH(500)
COMMON / BLOC24 / ANUM(500),BDEN(500),ALFA(500),VISCM(500)
COMMON / BLOC25 / VV(500),CT(500),TN(500)
COMMON / BLOC26 / ANUMEDC(500),ANUMMVC(500),ANUMHCI(500),
$ANUMCH(500)
COMMON / BLOC27 /BDENEDC(500),BDENMVC(500),BDENHCI(500),
$BDENCH(500)
COMMON / BLOC28 /VISAMEDC(500),VISAMMVC(500),VISAMHCI(500),
$VISAMCH(500)
COMMON / BLOC29 / CA
*****
*   Calculo do raio do joelho em metros(RBM)
*****
RBM = (RB/360.)*3.1416*(TE-DE)
*****
*   Calculo do comprimento do joelho(BL) em metros
*****
BL = 3.1416*TE/2.
NJ=NT-1
NE=(NT-1)+NJ
*****
*   Calculo do comprimento total da serpentina
*****
ZF=NT*TL+NJ*BL
*****
*   Gerando o vetor que indica a posicao na serpentina
*   se esta no joelho ou no tubo reto
*****
V(1)=TL
IK=1
DO 13 I=2,NE+1
  IF(IK.EQ.1) GOTO 14
  V(I) = V(I-1) + TL
  IK=1
  GOTO 13
14  V(I) = V(I-1) + BL
  IK=0
13  CONTINUE
DO 100 I=1,NE+1
  WRITE(12,*) V(I)
100 CONTINUE
*   *****
*   Calculo do Diametro Interno(DI) e raios das camadas
*   *****
RO= DE/2.
DI= DE-2.*EP
RT= DI/2.
RI= RT-EC
RETURN
END
*****
*
*   SUBROTINA FUNC
*
*
*
*****
SUBROUTINE FUNC
  IMPLICIT REAL*8 (A-H,O-Z)
  DOUBLE PRECISION CA(100,100)
  COMMON / BLOC1 / NI,NC,NRE
  COMMON / BLOC2 / PT,RG,FA,CKT,IG
  COMMON / BLOC3 / G0,VM,ROME,W10,TO,YPI,YPMI,YTI
  COMMON / BLOC4 / CP(500),CPM(500),PMM(500),AA(500,500)
  COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2

```

```

COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(500),HO(500),HINT(500),FI(500,500)
COMMON / BLOC8 / DCP1(500),DCP2(500)
COMMON / BLOC9 / A(500),B(500),PM(500),RD(500),RIN(500),R1(500),
SR2(500)
COMMON / BLOC10 / AV(500),BV(500),CV(500),VISA(500)
COMMON / BLOC11 / AK(500),BK(500),CK(500),DG(500),Tw(50)
COMMON / BLOC12 / UI(500),ROM(500),UNUD(500)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(500),TKM(500),RE(500),PR(500),FF(500),
SFR(500),VISAMCP(500)
COMMON / BLOC15 / TL,TE,RE,REB,RBM,BL
COMMON / BLOC16 / AG(500),BG(500),CG(500),TK(500),FM(500)
COMMON / BLOC17 / P(500),Twp,CT1(500),CT2(500)
COMMON / BLOC18 / NT,NE,MR1,MR2
COMMON / BLOC19 / V(500),DELTAQ1(500),DELTAQ2(500),PROD(500)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(500),CS(500),DGRE1(500),DGRE2(500)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA1(500),GAMA2(500),
$GAMA2E(500),GAMA3E(500),RMULT(500)
COMMON / BLOC23 / PMMEDC(500),PMMMVC(500),PMMHCI(500),PMMCH(500)
COMMON / BLOC24 / ANUM(500),BDEN(500),ALFA(500),VISCM(500)
COMMON / BLOC25 / VV(500),CT(500),TN(500)
COMMON / BLOC26 / ANUMEDC(500),ANUMMVC(500),ANUMHCI(500),
$ANUMCH(500)
COMMON / BLOC27 /BDENEDC(500),BDENMVC(500),BDENHCI(500),
$BDENCH(500)
COMMON / BLOC28 /VISAMEDC(500),VISAMMVC(500),VISAMHCI(500),
$VISAMCH(500)
COMMON / BLOC29 / CA
CALL PROP
CALL REYNOLDS
CALL TERMIC
DO 1008 I=2,MR2
DCP1(I) = ((A(2*MR2+1)+A(MR2+1)-A(1))*(X(I+3*MR2)*TO-298)+
$((B(2*MR2+1)+B(MR2+1)-B(1))*((X(I+3*MR2)*TO)**2-298**2))/2)
DCP2(I) = ((A(5*MR2+1)+A(2*MR2+1)-A(MR2+1))*(X(I+3*MR2)*TO-298)+
$((B(5*MR2+1)+B(2*MR2+1)-B(MR2+1))*((X(I+3*MR2)*TO)**2-298**2))/2)
1008 CONTINUE
DO 32 I = 2,MR2
CT1(I) = S1*DEXP(-EA1/(RG*X(I+3*MR2)*TO))
CT2(I) = S2*DEXP(-EA2/(RG*X(I+3*MR2)*TO))
DELTAQ1(I) = -(DELTAHO1+DCP1(I))/(G0*CPM(I))
DELTAQ2(I) = -(DELTAHO2+DCP2(I))/(G0*CPM(I))
32 CONTINUE
DO 33 I=2,MR2
R1(I) = CT1(I)*X(I)*ROM(I)/PM(I)
R2(I) = CT2(I)*X(I+MR2)*ROM(I)/PM(I+MR2)
33 CONTINUE
DO 35 I=2,MR2
RMULT(I) = DELTAQ1(I)*R1(I)+DELTAQ2(I)*R2(I)
35 CONTINUE
* ----- calculos para o 1,2-dicloroetano
DO 101 I=2,MR2
SM=0.
DO 20 J=1,MR2
SM = SM + CA(I,J)*X(J)
20 CONTINUE
F(I) = -(RI*ROME/(ZF*ROM(I)))*SM + (RI*ROME*PM(1)*(-R1(I))
$/(G0*ROM(I)))
* cálculo para o monocloreto de vinila
SM=0.
DO 60 J=1,MR2
SM = SM + CA(I,J)*X(J+MR2)
60 CONTINUE
F(I+MR2) = -(RI*ROME/ZF*ROM(I))*SM + (RI*ROME*PM(MR2+1)*
$(R1(I)-R2(I))/(G0*ROM(I)))
* cálculo para o ácido clorídrico
SM=0.
DO 21 J=1,MR2
SM = SM + CA(I,J)*X(J+2*MR2)
21 CONTINUE
F(I+2*MR2) = -(RI*ROME/ZF*ROM(I))*SM + (RI*ROME*PM(2*MR2+1)*

```

```

$(R1(I)+R2(I))/(G0*ROM(I))
* cálculo para a temperatura - balanço de energia
  SM=0.
  DO 25 J=1,MR2
  SM = SM + CA(I,J)*X(J+3*MR2)
25 CONTINUE
  F(I+3*MR2)=- (RI*ROME/(ZF*ROM(I)))*SM-(2.*2.6*UI(I)*
  $ROME)*(TO*X(I+3*MR2)-Tw(I))/(ROM(I)*G0*CPM(I)*TO)+
  $RI*ROME*RMULT(I)/(ROM(I)*G0*CPM(I)*TO)
101 CONTINUE
  RETURN
  END
*****
*
* SUBROTINA PROP
* USADA PARA AS PROPRIEDADES DOS COMPONENTES E DA MISTURA GASOSA
*
*****
SUBROUTINE PROP
IMPLICIT REAL*8 (A-H,O-Z)
  DOUBLE PRECISION CA(100,100)
  COMMON / BLOC1 / NI,NC,NRE
  COMMON / BLOC2 / PT,RG,FA,CKT,IG
  COMMON / BLOC3 / G0,VM,ROME,W10,TO,YPI,YPMI,YTI
  COMMON / BLOC4 / CP(500),CPM(500),PMM(500),AA(500,500)
  COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
  COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
  COMMON / BLOC7 / HI(500),HO(500),HINT(500),FI(500,500)
  COMMON / BLOC8 / DCP1(500),DCP2(500)
  COMMON / BLOC9 / A(500),B(500),PM(500),RD(500),RIN(500),
  $R1(500),R2(500)
  COMMON / BLOC10 / AV(500),BV(500),CV(500),VISA(500)
  COMMON / BLOC11 / AK(500),BK(500),CK(500),DG(500),Tw(50)
  COMMON / BLOC12 / UI(500),ROM(500),UNJD(500)
  COMMON / BLOC13 / RI,RT,RO
  COMMON / BLOC14 / VISAM(500),TKM(500),RE(500),PR(500),FF(500),
  $FR(500),VISAMCP(500)
  COMMON / BLOC15 / TL,TE,RB,RBM,BL
  COMMON / BLOC16 / AG(500),BG(500),CG(500),TK(500),FM(500)
  COMMON / BLOC17 / P(500),Twp,CT1(500),CT2(500)
  COMMON / BLOC18 / NT,NE,MR1,MR2
  COMMON / BLOC19 / V(500),DELTAQ1(500),DELTAQ2(500),PROD(500)
  COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
  COMMON / BLOC21 / RN(500),CS(500),DGRE1(500),DGRE2(500)
  COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA1(500),GAMA2(500),
  $GAMA2E(500),GAMA3E(500),RMULT(500)
  COMMON / BLOC23 / PMMEDC(500),PMMMVC(500),PMMHCI(500),
  $PMMCH(500)
  COMMON / BLOC24 / ANUM(500),BDEN(500),ALFA(500),VISCM(500)
  COMMON / BLOC25 / VV(500),CT(500),TN(500)
  COMMON / BLOC26 / ANUMEDC(500),ANUMMVC(500),ANUMHCI(500),
  $ANUMCH(500)
  COMMON / BLOC27 / BDENEDC(500),BDENMVC(500),BDENHCI(500),
  $BDENCH(500)
  COMMON / BLOC28 / VISAMEDC(500),VISAMMVC(500),VISAMHCI(500),
  $VISAMCH(500)
  COMMON / BLOC29 / CA
*****
* Cálculo do No. de moles e fração molar dos componentes
*****
  DO 2 I=1,MR2
  RN(I) = X(I)*VM/PM(I)
  RN(I+MR2) = X(I+MR2)*VM/PM(I+MR2)
  RN(I+2*MR2) = X(I+2*MR2)*VM/PM(I+2*MR2)
  RN(I+5*MR2) = X(I+5*MR2)*VM/PM(I+5*MR2)
  2 CONTINUE
  DO 3 I=1,MR2
  TN(I) = RN(I)+RN(I+MR2)+RN(I+2*MR2)+RN(I+5*MR2)
  3 CONTINUE
*
* Cálculo da fração molar dos componentes
*
*****

```

```

DO 5 I=1,MR2
  FM(I) = RN(I)/TN(I)
  FM(I+MR2) =RN(I+MR2)/TN(I)
  FM(I+2*MR2) = RN(I+2*MR2)/TN(I)
  FM(I+5*MR2) =RN(I+5*MR2)/TN(I)
5 CONTINUE
*
* *****
*   Calculo do Peso Molecular Medio da Mistura
* *****
DO 7 I=1,MR2
  PMMEDC(I)= FM(I)*PM(I)
  PMMMVC(I+MR2)= FM(I+MR2)*PM(I+MR2)
  PMMHCI(I+2*MR2)=FM(I+2*MR2)*PM(I+2*MR2)
  PMMCH(I+5*MR2)=FM(I+5*MR2)*PM(I+5*MR2)
  PMM(I)=PMMEDC(I)+PMMMVC(I+MR2)+PMMHCI(I+2*MR2)+PMMCH(I+5*MR2)
7 CONTINUE
*
* *****
*   Calculo da Capacidade Calorifica dos Componentes e Media
* *****
DO 10 I = 1,MR2
  CP(I) = (A(I)+B(I)*X(I+3*MR2)*TO)/PM(I)
  CP(I+MR2) = (A(I+MR2)+B(I+MR2)*X(I+3*MR2)*TO)/PM(I+MR2)
  CP(I+2*MR2) = (A(I+2*MR2)+B(I+2*MR2)*X(I+3*MR2)*TO)/PM(I+2*MR2)
  CP(I+5*MR2) = (A(I+5*MR2)+B(I+5*MR2)*X(I+3*MR2)*TO)/PM(I+5*MR2)
10 CONTINUE
DO 11 I=1,MR2
  CPM(I) =FM(I)*CP(I)+ FM(I+MR2)*CP(I+MR2)+FM(I+2*MR2)*CP(I+2*MR2)+
  $FM(I+5*MR2)*CP(I+5*MR2)
11 CONTINUE
*
* *****
*   Calculo da Densidade Media da Mistura
* *****
DO 12 I=1,MR2
  VV(I)=(G0*0.082*1.023*X(I+3*MR2)*TO)/(1./X(I+6*MR2))/X(I+4*MR2)*
  $23.33)
  ROM(I)=G0/VV(I)
12 CONTINUE
*
* *****
*   Calculo da Concentração ToTAL e dos Componentes
* *****
DO 13 I=1,MR2
  CT(I) =(X(I+4*MR2)*23.33)/(0.082*1.023*X(I+3*MR2)*TO)
13 CONTINUE
DO 14 I=1,MR2
  CS(I) = FM(I)*CT(I)
  CS(I+MR2) = FM(I+MR2)*CT(I)
  CS(I+2*MR2) = FM(I+2*MR2)*CT(I)
  CS(I+5*MR2) = FM(I+5*MR2)*CT(I)
14 CONTINUE
*
* *****
*   Calculo da Viscosidade Absoluta de cada Componente
* *****
DO 15 I=1,MR2
  VISA(I) = AV(I)*(X(I+3*MR2)*TO)**2+Bv(I)*X(I+3*MR2)*TO+CV(I)
  VISA(I+MR2) = AV(I+MR2)*(X(I+3*MR2)*TO)**2+Bv(I+MR2)*X(I+3*MR2)*TO
  $+CV(I+MR2)
  VISA(I+2*MR2) = AV(I+2*MR2)*(X(I+3*MR2)*TO)**2+Bv(I+2*MR2)*
  $X(I+3*MR2)*TO+CV(I+2*MR2)
  VISA(I+5*MR2) = AV(I+5*MR2)*(X(I+3*MR2)*TO)**2+Bv(I+5*MR2)*
  $X(I+3*MR2)*TO+CV(I+5*MR2)
15 CONTINUE
*
* *****
*   Calculo da Viscosidade Absoluta da Mistura
* *****
* *****
*   Calculo dos Coeficientes FI(I,J) da regra de mistura
* *****
DO 16 I=1,MR2
  FI(I,I)=1
  FI(I+MR2,I+MR2)=1
  FI(I+2*MR2,I+2*MR2)=1
  FI(I+5*MR2,I+5*MR2)=1

```

```

16  CONTINUE
    DO 1060 I=1,MR2
      FI(I,I+MR2) = ((1.+(VISA(I)/VISA(I+MR2))**0.5)*(PM(I+MR2)/PM(I))**
$0.25)**2.)/((8.*(1+PM(I)/PM(I+MR2)))**0.5)
      FI(I,I+2*MR2) = ((1.+(VISA(I)/VISA(I+2*MR2))**0.5)*(PM(I+2*MR2)/
$PM(I))**0.25)**2.)/((8.*(1+PM(I)/PM(I+2*MR2)))**0.5)
      FI(I,I+5*MR2) = ((1.+(VISA(I)/VISA(I+5*MR2))**0.5)*(PM(I+5*MR2)/
$PM(I))**0.25)**2.)/((8.*(1+PM(I)/PM(I+5*MR2)))**0.5)
1060 CONTINUE
    DO 1061 I=1,MR2
      FI(I+MR2,I) = ((1.+(VISA(I+MR2)/VISA(I))**0.5)*(PM(I)/PM(I+MR2))**
$0.25)**2.)/((8.*(1+PM(I+MR2)/PM(I)))**0.5)
      FI(I+MR2,I+2*MR2) = ((1.+(VISA(I+MR2)/VISA(I+2*MR2))**0.5)*
$(PM(I+2*MR2)/PM(I+
$MR2))**0.25)**2.)/((8.*(1+PM(I+MR2)/PM(I+2*MR2)))**0.5)
      FI(I+MR2,I+5*MR2) = ((1.+(VISA(I+MR2)/VISA(I+5*MR2))**0.5)*(PM(I+5*
$MR2)/PM(I+MR2))**0.25)**2.)/((8.*(1+PM(I+MR2)/PM(I+5*MR2)))**0.5)
1061 CONTINUE
    DO 1062 I=1,MR2
      FI(I+2*MR2,I) = ((1.+(VISA(I+2*MR2)/VISA(I))**0.5)*(PM(I)/PM(I+2*
$MR2))**0.25)**2.)/((8.*(1+PM(I+2*MR2)/PM(I)))**0.5)
      FI(I+2*MR2,I+MR2) = ((1.+(VISA(I+2*MR2)/VISA(I+MR2))**0.5)*(PM(I+MR
$1)/PM(I+2*MR2))**0.25)**2.)/((8.*(1+PM(I+2*MR2)/PM(I+MR2)))**0.5)
      FI(I+2*MR2,I+5*MR2) = ((1.+(VISA(I+2*MR2)/VISA(I+5*MR2))**0.5)*
$(PM(I+5*MR2)/PM(I+2*MR2))**0.25)**2.)/((8.*(1+PM(I+2*MR2)/PM(I+5*
$MR2)))**0.5)
1062 CONTINUE
    DO 1063 I=1,MR2
      FI(I+5*MR2,I) = ((1.+(VISA(I+5*MR2)/VISA(I))**0.5)*(PM(I)/
$PM(I+5*MR2))**0.25)**2.)/((8.*(1+PM(I+5*MR2)/PM(I)))**0.5)
      FI(I+5*MR2,I+MR2) = ((1.+(VISA(I+5*MR2)/VISA(I+MR2))**0.5)*
$(PM(I+MR2)/PM(I+5*MR2))**0.25)**2.)/((8.*(1+PM(I+5*MR2)/
$PM(I+MR2)))**0.5)
      FI(I+5*MR2,I+2*MR2) = ((1.+(VISA(I+5*MR2)/VISA(I+2*MR2))**0.5)*
$(PM(I+2*MR2)/PM(I+5*MR2))**0.25)**2.)/((8.*(1+PM(I+5*MR2)/
$PM(I+2*MR2)))**0.5)
1063 CONTINUE
*****
*   Calculo da viscosidade absoluta usando a regra de mistura
*
*****
    DO 20 I=1,MR2
      PROD(I) = FM(I)+FM(I+MR2)*FI(I,I+MR2)+FM(I+2*MR2)*FI(I,I+2*MR2)+
$FM(I+5*MR2)*FI(I,I+5*MR2)
      PROD(I+MR2) = FM(I)*FI(I+MR2,I)+FM(I+MR2)+FM(I+2*MR2)*
$FI(I+MR2,I+2*MR2)+FM(I+5*MR2)*FI(I+MR2,I+5*MR2)
      PROD(I+2*MR2) = FM(I)*FI(I+2*MR2,I)+FM(I+MR2)*FI(I+2*MR2,I+MR2)+
$FM(I+2*MR2)+FM(I+5*MR2)*FI(I+2*MR2,I+5*MR2)
      PROD(I+5*MR2) = FM(I)*FI(I+5*MR2,I)+FM(I+MR2)*FI(I+5*MR2,I+MR2)+
$FM(I+2*MR2)*FI(I+5*MR2,I+2*MR2)+FM(I+5*MR2)
20 CONTINUE
    DO 21 I=1,MR2
      VISAMEDC(I) = FM(I)*VISA(I)/PROD(I)
      VISAMMVC(I+MR2) = FM(I+MR2)*VISA(I+MR2)/PROD(I+MR2)
      VISAMHCI(I+2*MR2) = FM(I+2*MR2)*VISA(I+2*MR2)/PROD(I+2*MR2)
      VISAMCH(I+5*MR2) = FM(I+5*MR2)*VISA(I+5*MR2)/PROD(I+5*MR2)
21 CONTINUE
    DO 22 I=1,MR2
      VISAM(I) = VISAMEDC(I)+VISAMMVC(I+MR2)+VISAMHCI(I+2*MR2)+VISAMCH(I+
$5*MR2)
22 CONTINUE
    DO 23 I=1,MR2
      VISAMCP(I) = VISAM(I)
23 CONTINUE
*****CONVERTENDO A VISAM DE cp PARA Kg/m/h *****
    DO 1070 I=1,MR2
      VISAM(I) = VISAM(I)*3.6
1070 CONTINUE
*****
*   Calculo da Condutividade Termica de cada Componente
*
*****
    DO 24 I=1,MR2
      TK(I) = AK(I)*(X(I+3*MR2)*TO)**2+BK(I)*X(I+3*MR2)*TO+CK(I)

```

```

TK(I+MR2) = AK(I+MR2)*(X(I+3*MR2)*TO)**2+BK(I+MR2)*X(I+3*MR2)*TO+
$CK(I+MR2)
TK(I+2*MR2) = AK(I+2*MR2)*(X(I+3*MR2)*TO)**2+BK(I+2*MR2)*
$X(I+3*MR2)*TO+CK(I+2*MR2)
TK(I+5*MR2) = AK(I+5*MR2)*(X(I+3*MR2)*TO)**2+BK(I+5*MR2)*
$X(I+3*MR2)*TO+CK(I+5*MR2)
24 CONTINUE
*
*      Calculo da Condutividade Termica da Mistura
*
DO 25 I=1,MR2
ANUMEDC(I) = FM(I)*TK(I)*PM(I)**(1/3.)
ANUMMVC(I+MR2) = FM(I+MR2)*TK(I+MR2)*PM(I+MR2)**(1/3.)
ANUMHCI(I+2*MR2) = FM(I+2*MR2)*TK(I+2*MR2)*PM(I+2*MR2)**(1/3.)
ANUMCH(I+5*MR2) = FM(I+5*MR2)*TK(I+5*MR2)*PM(I+5*MR2)**(1/3.)
BDENEDC(I) =FM(I)*PM(I)**(1/3.)
BDENMVC(I+MR2) =FM(I+MR2)*PM(I+MR2)**(1/3.)
BDENHCI(I+2*MR2) =FM(I+2*MR2)*PM(I+2*MR2)**(1/3.)
BDENCH(I+5*MR2) =FM(I+5*MR2)*PM(I+5*MR2)**(1/3.)
25 CONTINUE
DO 26 I=1,MR2
ANUM(I) = ANUMEDC(I)+ANUMMVC(I+MR2)+ANUMHCI(I+2*MR2)+ANUMCH(I+5*
$MR2)
BDEN(I) =BDENEDC(I)+BDENMVC(I+MR2)+BDENHCI(I+2*MR2)+BDENCH(I+5*
$MR2)
TKM(I)=ANUM(I)/BDEN(I)
26 CONTINUE
*
*      Calculo do Numero de Prandtl(PR) no trecho
*
DO 27 I=1,MR2
ALFA(I) = TKM(I)/(ROM(I)*CPM(I))
27 CONTINUE
DO 28 I=1,MR2
VISC(I) = VISAM(I)/ROM(I)
PR(I) = VISC(I)/ALFA(I)
28 CONTINUE
RETURN
END
*
*
*      SUBROTINA REYNOLDS
*      USADA PARA CALCULAR O FATOR DE FRICCAO DE FANNING
*
*
SUBROUTINE REYNOLDS
IMPLICIT REAL*8 (A-H,O-Z)
DOUBLE PRECISION CA(100,100)
COMMON / BLOC1 / NI,NC,NRE
COMMON / BLOC2 / PT,RG,FA,CKT,IG
COMMON / BLOC3 / G0,VM,ROME,W10,TO,YPI,YPMI,YTI
COMMON / BLOC4 / CP(500),CPM(500),PMM(500),AA(500,500)
COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
COMMON / BLOC7 / HI(500),HO(500),HINT(500),FI(500,500)
COMMON / BLOC8 / DCP1(500),DCP2(500)
COMMON / BLOC9 / A(500),B(500),PM(500),RD(500),RIN(500),
$R1(500),R2(500)
COMMON / BLOC10 / AV(500),BV(500),CV(500),VISA(500)
COMMON / BLOC11 / AK(500),BK(500),CK(500),DG(500),Tw(50)
COMMON / BLOC12 / UI(500),ROM(500),UNUD(500)
COMMON / BLOC13 / RI,RT,RO
COMMON / BLOC14 / VISAM(500),TKM(500),RE(500),PR(500),FF(500),
$FR(500),VISAMCP(500)
COMMON / BLOC15 / TL,TE,RB,RBM,BL
COMMON / BLOC16 / AG(500),BG(500),CG(500),TK(500),FM(500)
COMMON / BLOC17 / P(500),Twp,CT1(500),CT2(500)
COMMON / BLOC18 / NT,NE,MR1,MR2
COMMON / BLOC19 / V(500),DELTAQ1(500),DELTAQ2(500),PROD(500)
COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
COMMON / BLOC21 / RN(500),CS(500),DGRE1(500),DGRE2(500)
COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA1(500),GAMA2(500),
$GAMA2E(500),GAMA3E(500),RMULT(500)

```

```

COMMON / BLOC23 / PMMEDC(500),PMMMVC(500),PMMHCI(500),
$PMMCH(500)
COMMON / BLOC24 / ANUM(500),BDEN(500),ALFA(500),VISCM(500)
COMMON / BLOC25 / VV(500),CT(500),TN(500)
COMMON / BLOC26 / ANUMEDC(500),ANUMMVC(500),ANUMHCI(500),
$ANUMCH(500)
COMMON / BLOC27 / BDENEDC(500),BDENMVC(500),BDENHCI(500),
$BDENCH(500)
COMMON / BLOC28 / VISAMEDC(500),VISAMMVC(500),VISAMHCI(500),
$VISAMCH(500)
COMMON / BLOC29 / CA
*
* *****
*      Calculo do Numero de Reynolds no escoamento
* *****
      DO 1010 I=1,MR2
      RE(I) = G0*DI/VISAM(I)
1010 CONTINUE
* *****
*      CALCULO DO FATOR DE FRICCAO PARA ESCOAMENTO TURBULENTO
* *****
      DO 1011 I=1,MR2
*      write(*,*) RE(I)
      FF(I)=0.046*(RE(I)**(-0.2))
1011 CONTINUE
* *****
*      Calculo do Numero de FR identificando o trecho - se reto ou
*      curvo - a partir do vetor posicao V() gerado no programa principal
* *****
      DO 1012 I=1,MR2
      FR(I) = 2.*FF(I)/DI
1012 CONTINUE
      RETURN
      END
*
* *****
*
*      SUBROTINA TERMIC
*      USADA PARA CALCULAR OS COEFICIENTES DE TROCA TERMICA
*
* *****
SUBROUTINE TERMIC
  IMPLICIT REAL*8 (A-H,O-Z)
  DOUBLE PRECISION CA(100,100)
  COMMON / BLOC1 / NI,NC,NRE
  COMMON / BLOC2 / PT,RG,FA,CKT,IG
  COMMON / BLOC3 / G0,VM,ROME,W10,TO,YPI,YPMI,YTI
  COMMON / BLOC4 / CP(500),CPM(500),PMM(500),AA(500,500)
  COMMON / BLOC5 / S1,S2,EA1,EA2,DELTAHO1,DELTAHO2
  COMMON / BLOC6 / DE,DI,EP,EC,ZF,PASSO
  COMMON / BLOC7 / HI(500),HO(500),HINT(500),FI(500,500)
  COMMON / BLOC8 / DCP1(500),DCP2(500)
  COMMON / BLOC9 / A(500),B(500),PM(500),RD(500),RIN(500),
  $R1(500),R2(500)
  COMMON / BLOC10 / AV(500),BV(500),CV(500),VISA(500)
  COMMON / BLOC11 / AK(500),BK(500),CK(500),DG(500),Tw(50)
  COMMON / BLOC12 / UI(500),ROM(500),UNUD(500)
  COMMON / BLOC13 / RI,RT,RO
  COMMON / BLOC14 / VISAM(500),TKM(500),RE(500),PR(500),FF(500),
  $FR(500),VISAMCP(500)
  COMMON / BLOC15 / TL,TE,RE,RB,RBM,BL
  COMMON / BLOC16 / AG(500),BG(500),CG(500),TK(500),FM(500)
  COMMON / BLOC17 / P(500),Twp,CT1(500),CT2(500)
  COMMON / BLOC18 / NT,NE,MR1,MR2
  COMMON / BLOC19 / V(500),DELTAQ1(500),DELTAQ2(500),PROD(500)
  COMMON / BLOC20 / TC,DTW1,DTW2,DTW3,DTW4
  COMMON / BLOC21 / RN(500),CS(500),DGRE1(500),DGRE2(500)
  COMMON / BLOC22 / F(2000),Y(2000),X(2000),GAMA1(500),GAMA2(500),
  $GAMA2E(500),GAMA3E(500),RMULT(500)
  COMMON / BLOC23 / PMMEDC(500),PMMMVC(500),PMMHCI(500),PMMCH(500)
  COMMON / BLOC24 / ANUM(500),BDEN(500),ALFA(500),VISCM(500)
  COMMON / BLOC25 / VV(500),CT(500),TN(500)
  COMMON / BLOC26 / ANUMEDC(500),ANUMMVC(500),ANUMHCI(500),
  $ANUMCH(500)

```

```

COMMON / BLOC27 /BDENEDC(500),BDENMVC(500),BDENHCI(500),
$BDENCH(500)
COMMON / BLOC28 /VISAMEDC(500),VISAMMVC(500),VISAMHCI(500),
$VISAMCH(500)
COMMON / BLOC29 / CA
*****
*   Calculo do Numero de Nusselt (NUD) no trecho
*   Equacao de Pethukov, Kirilov e Popov
*   Faixa: 0.5 < PR(I) < 2000, 10.000 < RED(I) < 5.000.000
*   *****
      DO 1015 I=1,MR2
        UNUD(I)=(FF(I)/2.)*RE(I)*PR(I)/(1.07+12.7*((FF(I)/2.)**0.5)*(PR(I)
$**(2/3.)-1.))
        HI(I) = UNUD(I)*TKM(I)/DI
        HO(I) = HI(I)
1015   CONTINUE
*   *****
*   Calculo do Coeficiente Global de troca UI - baseado em Alnterna
*   *****
      DO 1016 I=1,MR2
        UI(I) = (1./RI)/(1./(RI*HI(I))+DLOG(RT/RI)/CKT+1./(RO*HO(I)))
1016   CONTINUE
*   *****
*   Calculo do Coeficiente Interno hint pela equaç/Eo do Kern
*   *****
*   hint = 0.023*(RE**0.8)*(PR**0.4)*TKM/DI
      DO 1017 I=1,MR2
        hint(I)=2.*UI(I)*1.30
1017   CONTINUE
      RETURN
      END
*****
*   INTERFACE ROUTINE FROM DEC TO PC FOR GETTING TIME
*   SUBROTINA TEMPO
*****
SUBROUTINE TEMPO(X)
*   RETURN CURRENT TIME - MIDNIGHT - X
INTEGER*2 IHOOR, IMINUT, ISECON, IHUND
REAL*4  RHOUR, RMINUT, RSECON, RHUND
REAL*4  X1, X
CALL GETTIM(IHOOR, IMINUT, ISECON, IHUND)
RHOUR  = FLOAT( IHOOR )
RMINUT = FLOAT( IMINUT)
RSECON = FLOAT( ISECON)
RHUND  = FLOAT( IHUND )
X1     = RHOUR*3600.0 + RMINUT*60.0 + RSECON +
&      RHUND/100.0
X = X1 - X
RETURN
END

```

## ANEXO 4 – DADOS PROCESSADOS NA RNA

Vazão mássica (t/h)	Pressão entrada (Kgf/cm <sup>2</sup> )	Temp. convecção (°C)	TW <sub>1</sub> (°C)	TW <sub>2</sub> (°C)	TW <sub>3</sub> (°C)	TW <sub>4</sub> (°C)	Conversão (%)	Tempo de Residência (seg)	Seletividade (%)
24,67	23,33	240	370	510	552	557,5	60,43	23,53	97,02
27,76	23,33	240	370	510	552	557,5	57,06	20,98	97,40
30,84	23,33	240	370	510	552	557,5	54,05	18,92	97,69
33,93	23,33	240	370	510	552	557,5	51,34	17,21	97,92
37,01	23,33	240	370	510	552	557,5	48,86	15,76	98,11
24,67	23,33	264	407	561	607,2	613,25	92,52	20,74	65,79
27,76	23,33	264	407	561	607,2	613,25	90,10	18,50	74,88
30,84	23,33	264	407	561	607,2	613,25	87,64	16,69	80,44
33,93	23,33	264	407	561	607,2	613,25	85,20	15,19	84,14
37,01	23,33	264	407	561	607,2	613,25	82,80	13,92	86,74
24,67	23,33	252	388,5	535,5	579,6	585,38	78,50	22,07	91,50
27,76	23,33	252	388,5	535,5	579,6	585,38	75,15	19,69	92,97
30,84	23,33	252	388,5	535,5	579,6	585,38	72,05	17,76	94,01
33,93	23,33	252	388,5	535,5	579,6	585,38	69,17	16,16	94,78
37,01	23,33	252	388,5	535,5	579,6	585,38	66,49	14,81	95,38
24,67	23,33	228	351,5	484,5	524,4	529,6	42,06	25,09	98,81
27,76	23,33	228	351,5	484,5	524,4	529,6	39,19	22,35	98,94
30,84	23,33	228	351,5	484,5	524,4	529,6	36,68	20,13	99,04
33,93	23,33	228	351,5	484,5	524,4	529,6	34,46	18,30	99,13
37,010	23,33	228	351,5	484,5	524,4	529,6	32,47	16,75	99,20
24,67	23,33	216	333	459	496,8	501,75	25,98	26,67	99,52
27,76	23,33	216	333	459	496,8	501,75	23,86	23,73	99,57
30,84	23,33	216	333	459	496,8	501,75	22,04	21,35	99,61
33,93	23,33	216	333	459	496,8	501,75	20,45	19,38	99,64
37,01	23,33	216	333	459	496,8	501,75	19,05	17,73	99,67
24,67	23,33	204	314,5	433,5	469,2	473,875	13,84	28,18	99,82
27,76	23,33	204	314,5	433,5	469,2	473,875	12,51	25,03	99,84
30,84	23,33	204	314,5	433,5	469,2	473,875	11,40	22,50	99,85
33,93	23,33	204	314,5	433,5	469,2	473,875	10,45	20,41	99,87
37,01	23,33	204	314,5	433,5	469,2	473,875	9,62	18,65	99,88
24,67	23,33	192	296	408	441,6	446	6,21	29,56	99,94
27,76	23,33	192	296	408	441,6	446	5,54	26,23	99,95
30,84	23,33	192	296	408	441,6	446	4,98	23,55	99,95
33,93	23,33	192	296	408	441,6	446	4,52	21,35	99,96
37,01	23,33	192	296	408	441,6	446	4,12	19,50	99,96
24,67	25,663	240	370	510	552	557,5	62,69	25,86	96,74
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
30,84	25,663	252	388,5	535,5	579,6	585,38	74,35	19,55	93,29
33,93	25,663	252	388,5	535,5	579,6	585,38	71,54	17,81	94,18
33,93	20,997	264	407	561	607,2	613,25	82,81	13,61	86,68
37,01	20,997	264	407	561	607,2	613,25	80,22	12,45	88,78