

SIMULAÇÃO DE TRANSIENTES EM REATORES DE PESQUISA

por

HORÁCIO NAKATA

RESUMO -- No presente trabalho foi desenvolvido o programa PC-REATOR para simulação de transientes de operação de reatores de pesquisa tipo MTR. O simulador PC-REATOR foi escrito em Pascal para o microcomputador IBM-PC, e apresenta grande flexibilidade na sua utilização, podendo ser interrompido a qualquer instante para teste e modificação de parâmetros. Com esse programa os operadores podem adquirir experiência sem os riscos e prejuízos que acarretariam em operação real dos reatores de pesquisa.

INTRODUÇÃO

Os reatores de pesquisa tipo MTR podem ser operados manualmente ou automaticamente, observando-se os critérios de segurança pré-definidos.

Para a mudança de potência a movimentação das barras de controle e de segurança pode ser feita manualmente pelos operadores, observando o período mínimo de multiplicação de potência, ou pode ser feita através da malha de controle automático, a qual é ajustada para não ultrapassar o período mínimo de multiplicação.

É apresentado, no presente trabalho, um programa de simulação de reatores de pesquisa tipo MTR, a qual servirá para ensino e treinamento de operadores, através de simulação dos principais efeitos observados durante um transiente de mudança de potência.

No programa desenvolvido o núcleo é representado por um volume de controle onde é resolvida a equação de cinética puntual em dois grupos de precursoros. Os efeitos de realimentação de potência considerados são a temperatura do refrigerante e a temperatura do combustível, com os efeitos do Xenônio e do Samário desprezados, uma vez que a duração da simulação não deve ser mais do que alguns minutos.

A potência gerada pelo decaimento de produtos de fissão é aproximada pela função de Borst-Wheeler (ref. 2), e a cada instante o programa calcula o acúmulo e o decaimento dos produtos de fissão.

A reatividade das barras de controle é simulada pela curva integral de reatividade representada por senóide deformada, ou por qualquer outra função desejada.

O controle automático é acionável por meio de teclado, ativando a malha de controle de potência com limitação do período de multiplicação de potência.

As temperaturas do combustível e do refrigerante são calculadas com a conservação de energia em um único volume de controle axial, sob as condições de controle previamente conhecidas no estado estacionário.

O programa por ser interrompido a qualquer momento para mudança de qualquer parâmetro ou regime de controle. O usuário pode exercitar várias

opções de acordo com o transcorrer do transiente para fins de treinamento e aprendizado, podendo assim adquirir experiência sem incorrer em riscos e prejuízos reais ao reator.

METODOLOGIA UTILIZADA NO PROGRAMA PC-REATOR

O programa PC-REATOR resolve a equação de difusão na modelagem de cinética puntual, com dois grupos de precursores.

As equações são explicitamente escritas como :

$$\frac{d}{dt} n(t) = \frac{\rho(t) - \beta_{ef}}{\lambda_o} n(t) + \sum_{i=1}^2 \lambda_i C_i(t) \quad (1)$$

$$\frac{d}{dt} C_i(t) = -\lambda_i C_i(t) + \frac{\beta_i}{\lambda_o} n(t), \quad i=1,2 \quad (2)$$

onde :

- $n(t)$ = densidade de neutrons,
- $\rho(t)$ = reatividade total do núcleo,
- β_{ef} = fração efetiva de neutrons atrasados,
- λ_o = vida média dos neutrons,
- β_i = fração efetiva de neutrons do grupo i ,
- λ_i = constante de decaimento do grupo i ,
- $C_i(t)$ = concentração dos precursores do grupo i .

Em testes comparativos a aproximação em dois grupos de precursores mostrou-se bastante satisfatório em relação aos resultados obtidos com 6 grupos de precursores, em transientes com duração de algumas dezenas de segundos.

A reatividade $\rho(t)$ inclui as reatividades da barra de controle, realimentação do moderador e realimentação do combustível, i.é.,

$$\rho(t) = \rho_b(t) + \rho_M(t) + \rho_F(t) \quad (3)$$

A reatividade da barra de controle, $\rho_b(t)$, é calculada através da curva integral aproximada pela função,

$$\rho(t) = \frac{\rho_{\max}}{2} \left[1 - \cos\left(\frac{\pi z(t)}{H}\right) \right] f(z(t), a) \quad (4)$$

com opções para deformação da curva para o fundo do núcleo ou para o topo do núcleo, escolhendo-se o parâmetro da função $f(z(t), a)$ convenientemente.

A deformação para o topo do núcleo é conseguida com,

$$f(z(t), a) = a + \frac{z(t)}{H} (1-a) \quad (5)$$

onde a = parâmetro variando de 0 a 1, com 0 obtendo maior pico e com 1 sem deformação.

A deformação para o fundo do núcleo é conseguida com,

$$f(z(t), a) = 1 - \frac{z(t)}{H} (1-a) \quad (6)$$

onde a = parâmetro com as mesmas propriedades da Equação 5.

A reatividade da barra de controle $\tilde{\epsilon}$ é obtida por diferença de reatividade definida como :

$$\rho_b(t) = \rho(t) - \rho(t_0). \quad (7)$$

As reatividades $\rho_M(t)$ e $\rho_F(t)$, realimentação do moderador e do combustível, são dadas, respectivamente, por:

$$\rho_M(t) = \alpha_M (T_M(t) - T_{M0}) \quad (8)$$

$$\rho_F(t) = \alpha_F (T_F(t) - T_{F0}) \quad (9)$$

onde :

α_M, α_F = coeficientes de reatividade do moderador e do combustível,

$T_M(t), T_F(t)$ = temperatura do moderador e do combustível, e

T_{M0}, T_{F0} = temperaturas iniciais do moderador e do combustível.

O calor residual devido à operação do reator antes do início da simulação é calculado pela função de Borst-Wheeler (ref.2),

$$P_{\text{residual}}(t, T) = 0,0665 \left[t^{-0.2} - (t+T)^{-0.2} \right] \frac{W}{W_{\text{total}}}$$

onde :

t = tempo (> 1 segundo) após desligamento do reator, e

T = tempo (s) de operação antes da simulação.

No instante t , a potência residual é calculada integrando-se a potência de decaimento devida a partículas beta e raios gama, dada por (ref.2),

$$\frac{d}{dt'} P_{\gamma, \beta}(t, t') = 0.0133 (t-t')^{-1.2} \frac{W}{W_{\text{total}}}$$

onde t' = instante da ocorrência de fissão, em segundos.

As temperaturas são calculadas com um modelo simplificado do núcleo, em um volume de controle, com as equações de conservação de energia dadas por:

$$M_F C_F \frac{d}{dt} T_F(t) = f_F P(t) + A_{FF} K_F (T_F(t) - T_M(t)) \quad (10)$$

$$M_M C_M \frac{d}{dt} T_M(t) = (1-f_F) P(t) - W C_M (T_{M_{\text{out}}}(t) - T_{M_{\text{in}}}(t)) \quad (11)$$

onde:

M_F, M_M = massas do combustível e do moderador,

C_F, C_M = calores específicos do combustível e do moderador,

f_F = fração da potência gerada no combustível,

- $P(t)$ = potência total do núcleo,
 A_F = área de transferência de calor para o refrigerante
 K_F = coeficiente efetivo de transmissão de calor por convecção
 W = fluxo máximo de refrigerante
 $T_{M_{out}}(t)$ = temperatura do refrigerante na saída do núcleo
 $T_{M_{in}}(t)$ = temperatura do refrigerante na entrada do núcleo.

O esquema numérico utilizado no programa PC-REATOR é o Runge-Kutta clássico de 4ª ordem, com aproximação "PROMPT-JUMP" na equação da concentração de neutrons, desprezando-se o termo da derivada. Essa aproximação é válida para taxas de inserção de reatividade da ordem de 0,2 dólar/segundo, até o máximo de 0,9 dólar (referência 1).

O controle do reator no programa PC-REATOR pode ser manual ou automático.

Em controle automático, pode-se calibrar o ganho do período, de modo a manter o período acima do valor desejado. A Figura 1 mostra o diagrama de controle implementado no programa.

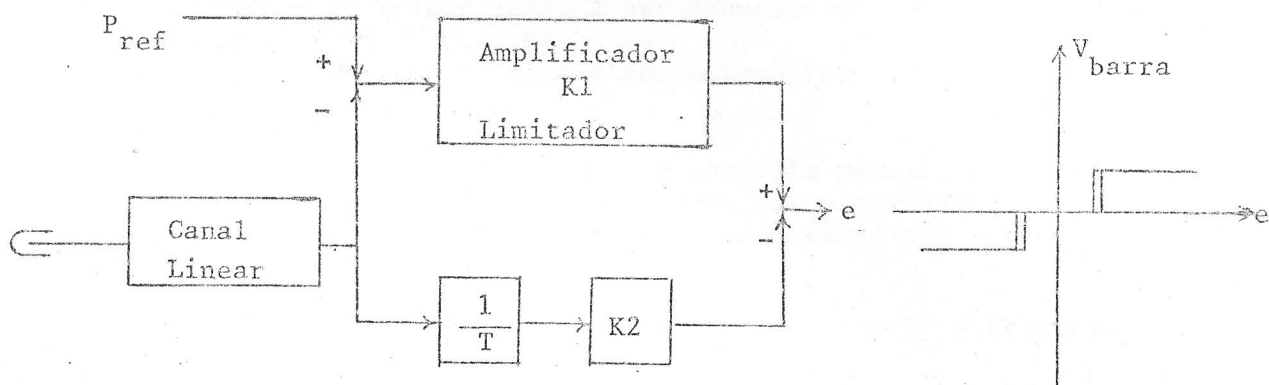


Figura 1 - Diagrama de Controle Automático do Reator IEA-R1.

O controle do reator é feito por sinal proporcional ao erro, compensado pelo sinal proporcional ao inverso do período. O amplificador limitador tem ganho K_1 , porém o sinal transmitido é limitado a e_{max} , para evitar altas taxas de aumento de potência. O período do reator é mantido abaixo do limite escolhido, através da calibração do ganho K_2 , o qual amplifica o inverso do período do reator. O valor de K_1 no reator IEA-R1 é 100 e o limite é aproximadamente 0,5% da potência nominal.

DESCRIÇÃO DAS ENTRADAS DO PC-REATOR

As opções do programa PC-REATOR são, na maior parte, pré-definidas, porém, pode-se durante a execução, redefiní-las pelo teclado. Dentre as opções mais importantes estão a escolha do número de gráficos e o número de curvas por gráfico para o acompanhamento do transiente.

Os índices requeridos para a escolha das curvas são tabelados a seguir:

<u>Índice</u>	<u>Variável</u>
1-	Concentração dos precursores do grupo 1,
2-	Concentração dos precursores do grupo 2,
7-	Potência Nuclear Normalizada,
8-	Posição da Barra de Controle,
9-	Temperatura do Combustível ($^{\circ}\text{C}$)
10-	Período do Reator (s)
11-	Temperatura do Refrigerante na Entrada do Núcleo ($^{\circ}\text{C}$)
12-	Temperatura Média do Refrigerante ($^{\circ}\text{C}$)
14-	Temperatura do Refrigerante na Saída do Núcleo ($^{\circ}\text{C}$)
15-	Potência Residual do Reator (Normalizada)
16-	Potência Total do Reator (Normalizada)

Independentemente do número de gráficos escolhidos o programa mostra a cada instante o valor dessas variáveis para melhor definição dos valores que variam lentamente.

Qualquer tecla pode interromper temporariamente a execução, podendo-se escolher entre quatro alternativas: V/B/A/E. As opções são descritas como :

- E - encerra a execução do programa,
- V - qualquer variável pode ser modificado pelo usuário,
- B - o controle do reator passa a ser manualmente, e o usuário pode movimentar a barra ou pará-la,
- A - o controle do reator passa a ser automático, e o valor da potência desejada pode ser mudado.

Tanto em controle automático como em controle manual, o programa calcula a cada instante o período do reator e compara com o período mínimo de segurança. Se ultrapassado esse limite as barras de segurança são inseridas e o calor residual passa a ser computado.

Um exemplo de simulação de operação do Reator IEA-R1 é ilustrado no apêndice com controle automático e controle manual.

REFERÊNCIAS

1. HETRICK, D.L., DYNAMICS OF NUCLEAR REACTORS, The University of Chicago Press, 1971.
2. LAMARSH, J.R., INTRODUCTION TO NUCLEAR REACTOR THEORY, Addison-Wesley, 1966.

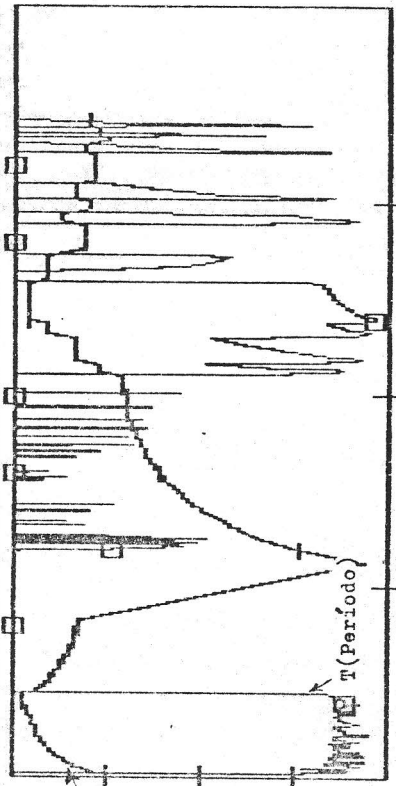
APÊNDICE

Exemplo de saída no vídeo durante controle automático e controle manual em mudança de potência de 20% → 100% → 20% → 100% da potência nominal.

MUDAR: Variavel/Barra controle/Automatico/ou Encerra? (entre: V/B/A/E) : e

BC - 800.00
T - 200.00 seg

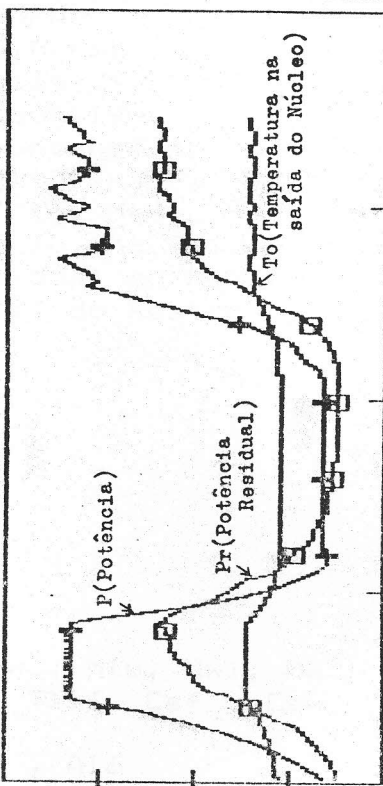
BC (Posição da Barra de Controle)



BC - 600.00
T - 10.00 seg

0.00 125.00 250.00 375.00

To - 50.00 °C
Pr - 0.10
P - 1.20



To - 20.00 °C
Pr - 0.00
P - 0.00

0.00 125.00 250.00 375.00

Controle Automático de 20% a 100% de 100% a 20%
Controle Manual de 20% a 100%

P/ =	102.182
T =	311.407
Ro =	0.02518
V =	0.000
BC =	762.000
CI =	4496.069
C2 =	505.747
Ti =	28.000
Tm =	29.508
To =	31.015
Tf =	36.402

T(s) = 433.600

```

program REATOR(input,output);
  { EXPERIMENTAL PARA VERIFICAR CALOR RESIDUAL POR DT }
  { JUNHO,29,1988 }
  { simulacao dinamica do nucleo em 2 GRUPOS DE PRECURSORES}
  { Modelo para simular IEA-R1 com controle de barras }
  { Junho 1988 - Horacio Nakata }

  { Variav[1] := Conc. Prec. Grupo 1
  Variav[2] := Conc. Prec. Grupo 2
  Variav[3] := Conc. Prec. Grupo 3
  Variav[4] := Conc. Prec. Grupo 4
  Variav[5] := Conc. Prec. Grupo 5
  Variav[6] := Conc. Prec. Grupo 6
  Variav[7] := Pot. Nuclear Normalizada
  Variav[8] := Pos. das Barras de Controle
  Variav[9] := Temp. do Combustivel
  Variav[10] := Período Instantaneo do Reator
  Variav[11] := Temp. do Refrig. na Entrada do Reator
  Variav[12] := Temp. do Refrig. Media do Reator
  Variav[14] := Temp. do Refrig. na Saída do Reator
  Variav[15] := Potencia Residual do Reator
  Variav[16] := Potencia Total do Reator
  Variav[17] := }

const
  Pi = 3.14159 ;
  Nimp = 9 ;
  NumEq = 21 ;
  NumCon = 5 ;
  NumCurv = 21 ;

type
  Parlist = record
    Afw1 , Afw2, Kfw , Ff, Fw1, Fw2, Mfu, Mw1, Mw2 : real ;
    Q0 , Fresid, Tresid, Beta, Tlif, Csf , Csw : real ;
    Tf0 , Tw0 , Ref, Pot , Wa : real ;
    A11 , A12 , A13 , A14 , A15 , A16 : real ;
    Be1 , Be2 , Be3 , Be4 , Be5 , Be6 : real ;
  end ;

  Contlist = record
    Romax, Nb, Nbarra0, Top_Peak, Bot_Peak, Rpeak : real ;
    PotRef, Atraso, Veloc_Barra, dTemp_entrada : real ;
    T_entrada, Alfaw, Alfaf, Banda1, Banda2 : real ;
    K1, K2, ErroMax : real ;
  end ;

  VetEst = array[1..NumEq] of real;
  VetCon = array[1..NumCon] of real;
  Nome = string[80];
  Varia = string[40];
  Registro = array[1..NumCurv] of real;
  variavel = array[1..NumCurv] of varia;

var
  Y , Yp , Yt : VetEst;
  U , Er , Er0 , Ur , Ur0 : VetCon;
  Rofict , Temporef : real;
  P : Parlist;
  i : Integer;
  car : char;
  C : contlist;
  Tempo , Temposim , Deltat : real;
  PotAcum1, PotAcum2 : array[1..10] of real;

```

```

Checa_periodo           : integer;
Periodo_Critico         : real;
Numdt                   : integer;
Temporel, Potrel        : real;
UmNdt , PotNdt          : array[1..500] of real;
Shutdown                : boolean;
Flag, Auto              : Integer;
Ident , Maxmin          : Text;
Dados                   : file of Registro;
Titulo                  : Nome;
NumVar, fim             : Integer;
NumReg, Ninterv         : Integer;
ArqVar                  : Varia;
ArqDat                  : Varia;
Arq, Imp, Video         : Registro;
Variav                  : Variavel;
Xplt, Xpltmn, Xpltmx, Dvx : real;
Xesq, Xdir, Px, Px0, Ngraf, Contador, Npx, Delcont : Integer ;
Yplt, Ypltmn, Ypltmx, Dvy : array[1..20,1..20] of real;
Ypltmn1, Ypltmx1       : array[1..20] of real;
Ysuper, Yinfer, Yref, Npy, Ncurv : array[1..2] of integer;
Ind, Py, Py0           : array[1..20,1..20] of integer;
OpGrav, OpGraf, OpImpr : string[1];

```

Procedure ENTRADA:

```

var
  i, EntI                : integer;
  EntrR1, EntrR2, EntrR3 : real;
  Arq1                   : string[15];
begin
  with P do begin
    ClrScr;
    for i := 1 to 4 do writeln;
    Writeln('          RTI - SISTEMAS DE CONTROLE          ');
    Writeln('          =====          ');
    Writeln;
    Writeln('          Julho/85          ');
    Writeln;
    Writeln('          Simulacao Dinamica do Reator          ');
    Writeln('          -----          ');
    Writeln;
    Write('Entre Tempo de Duracao da Simulacao:') ; readln(EntrR1);
    If EntrR1 <> 0.0 then Temposim := EntrR1;
    Write('Entre Intervalo de Integracao          :') ; readln(EntrR2);
    If EntrR2 <> 0.0 then Deltat := EntrR2;
    Write('Entre Potencia Consumida          :') ; readln(EntrR3);
    If EntrR3 <> 0.0 then pot := EntrR3;

    Write('Entre <S>im para Gravar no Disco          :') ; readln(OpGrav);
    If OpGrav = 'S' then
      begin
        Write('Entre o Nome do Arquivo de Identificacao : ');
        readln(Arq1);
        If Arq1 <> ' ' then ArqVar := Arq1;
        Write('Entre o Nome do Arquivo de Dados          : ');
        readln(Arq1);
        If Arq1 <> ' ' then ArqDat := Arq1;
      end;
    Write('Entre <S>im p/ Imprimir Resultados : ') ; readln(OpImpr);
    Write('Entre Numero de Intervalos a Suprimir :') ; readln(EntI);
    If EntI <> 0.0 then Ninterv := EntI;
    Writeln('Entre Titulo para a Simulacao          : ') ; readln(Titulo);
  end;
end;

```

end;

```
Procedure INICIALIZACAO(var F: Parlist);
```

```
var
```

```
  I : Integer;
```

```
  Df: String[11];
```

```
begin
```

```
  With F do begin
```

```
    ClrScr;
```

```
    For I := 1 to 4 do writeln;
```

```
    Writeln('          RTI - SISTEMAS DE CONTROLE           ');
```

```
    Writeln('          =====                               ');
```

```
    Writeln;
```

```
    Writeln('          Junho/88                               ');
```

```
    Writeln;
```

```
    Writeln('          Simulacao Dinamica do Reator           ');
```

```
    Writeln('          -----                               ');
```

```
    Temposim :=500.0;          Deltat := 0.2;          pot := 0.2 ;
```

```
    Ninterv := 1;          OpGrav := 'N';          OpGraf := 'S';
```

```
    OpImpr := 'N';          ArqVar := 'b:Arq.Ide'; Shutdown:=False;
```

```
    ArqDat := 'b:Arq.Dat';
```

```
    gotoxy(26,13); writeln('D E F A U L T');
```

```
    gotoxy(26,14); writeln('=====');
```

```
    gotoxy(20,16); writeln('Tempo de Duracao da Simulacao : ',
```

```
      Temposim:4:1);
```

```
    gotoxy(20,17); writeln('Intervalo de Integracao : ',
```

```
      Deltat:1:4);
```

```
    gotoxy(20,18); writeln('Potencia Consumida : ',
```

```
      pot:4:1);
```

```
    gotoxy(20,19); writeln('Opcao de Gravacao Ativada ? : ',
```

```
      OpGrav);
```

```
    gotoxy(20,20); writeln('Opcao de Impressao Ativada ? : ',
```

```
      OpImpr);
```

```
    gotoxy(20,21); writeln('Opcao de Grafico Ativada ? : ',
```

```
      OpGraf);
```

```
    gotoxy(20,22); writeln('Numeros de Intervalos Suprimidos: ',
```

```
      Ninterv:2);
```

```
    gotoxy(20,24); write ('ASSUMIR VALORES DO DEFAULT? <S/N>:');
```

```
      Readln(Df);
```

```
    If ( Df = 'N' ) or ( Df = 'n' ) then ENTRADA;
```

```
    Variav[1] := ' Conc. Prec. Grupo 1           ';
```

```
    Variav[2] := ' Conc. Prec. Grupo 2           ';
```

```
    Variav[3] := ' Conc. Prec. Grupo 3           ';
```

```
    Variav[4] := ' Conc. Prec. Grupo 4           ';
```

```
    Variav[5] := ' Conc. Prec. Grupo 5           ';
```

```
    Variav[6] := ' Conc. Prec. Grupo 6           ';
```

```
    Variav[7] := ' Pot. Nuclear Normalizada       ';
```

```
    Variav[8] := ' Pos. das Barras de Controle     ';
```

```
    Variav[9] := ' Temp. do Combustivel           ';
```

```
    Variav[10] := ' Período Instantaneo do Nucleo  ';
```

```
    Variav[11] := ' Temp. do Refrig. na Entrada do Nucleo ';
```

```
    Variav[12] := ' Temp. do Refrig. Media do Nucleo ';
```

```
    Variav[14] := ' Temp. do Refrig. na Saida do Nucleo ';
```

```
    Variav[15] := ' Potencia Residual do Nucleo ';
```

```
    Variav[16] := ' Potencia Total do Nucleo ';
```

```
    Variav[17] := ' ';
```

```
    Variav[18] := ' ';
```

```
  end;
```

```
end;
```

```
procedure PARAMETROS (var F : parlist ; var C : contlist ) ;
```

```
begin
```

```
  with F do begin
```

```
with C do begin
```

```

Q0 := 2.0E+06 ; { Potencia do Nucleo }
Fresid := 0.0665; { fracao da potencia residual }
Tresid := 30*86400.0; { 30 dias de operacao continua }
Numdt := 100; { No. de dt para Calor Residual } {max 500}

```

```
{ Parametros de Controle }
```

```

Nb := 1000.0 ; Nbarra0 := 0.75*Nb ; Romax := 0.03000 ;
Top_Peak := 1 ; Bot_Peak := 0 ; Rpeak := 0.72 ;
Veloc_Barra := 4.0 ; Periodo_Critico:=15.0; PotRef:= Pot ;
Atraso := 0.001 ; Auto := 0 ;
Banda1 := 0.20 ; Banda2 := 0.20 ; ErroMax:= 0.50 ;
K1 := 100.0 ; K2 := 11.0 ;

```

```
{ Parametros de Cinetica }
```

```

A11 := 0.0252 ; A12 := 0.566 ; A13 := 0.115 ;
A14 := 0.311 ; A15 := 1.4 ; A16 := 3.87 ;
Be1 := 0.002319 ; Be2 := 0.005486 ; Be3 := 0.001466 ;
Be4 := 0.003175 ; Be5 := 0.0009984 ; Be6 := 0.0002028 ;
Tlif := 0.0000183 ; Beta:= Be1 + Be2 ; { aprox. 2 grupos }

```

```
{ Parametros Termohidraulicos }
```

```

dTemp_entrada:= 0.0 ; T_entrada:= 28.0 ; Wa := 162.0 ;
Alfaf := -0.9E-05 ; Alfaw := -13.0E-05 ;
Ff := 0.97 ; Fw2 := 0.015 ; Fw1 := 0.015 ;
Mfu := 29.6 ; Mw1 := 33.0 ; Mw2:= 33.0 ;
Afw1 := 19.05 ; Afw2:= 19.05 ;
Csw := 4180.0 ; Csf := 760.0 ;
Kfw := 7560.0 ;
Tf0 := 30.0 ; Tw0:= 30.0 ;

```

```
end;
```

```
end;
```

```
end;
```

```

procedure CONDICOES(var Y:VetEst ;var Ur:VetCon ;var U:VetCon ;
var Er:VetCon;var Er0:Vetcon;var P:parlist;
var C:ContList;var Ur0:VetCon);

```

```
var
```

```

Aux1,A1,A2,A3,A4,A5,A6,A7,A8 : real;
Rob,Rot,Reat : real;

```

```
begin
```

```
with P do begin
```

```
with C do begin
```

```
{ Neutronica }
```

```

Potrel:= Fresid*Pot;
Y[15] := Fresid*(1.0-exp(-0.2*ln(1.0+Tresid)))*Pot; {Calor Residual}
Y[7] := Pot - Y[15]; { Potencia Nuclear }
Y[16] := Pot; { Potencia Total }
For i:= 1 to 10 do begin
PotAcum1[i] := Y[7];
PotAcum2[i] := Y[7]; end;

```

```
{ Concentracao de Precursores dos 6 Grupos }
```

```

Y[1] := (Be1*Y[7])/(Tlif*A11);
Y[2] := (Be2*Y[7])/(Tlif*A12);
Y[3] := (Be3*Y[7])/(Tlif*A13);
Y[4] := (Be4*Y[7])/(Tlif*A14);
Y[5] := (Be5*Y[7])/(Tlif*A15);
Y[6] := (Be6*Y[7])/(Tlif*A16);

```

```
Y[8] := TRUNC(Nbarra0);
```

```
{ Temperaturas do Primario }
```

```

Y[11] := T_entrada;           (Temperatura de Entrada )
Y[14] := Y[11] + Pot*Q0/(Ww*Csw); (Temperatura de Saída )
Y[12] := 0.5*(Y[11] + Y[14]);  (Temperatura Média )
                                (Temperatura do Combustível)
Y[9] := (Ff*Y[7]+Y[15])*Q0/((Afw1 + Afw2)*Kfw) + Y[12];

```

{Reatividade das Barras}

```

Aux1 := Trunc(Y[8])/Nb;
Rob := 0.5*Romax*(1.0 - cos(Pi*Aux1));
If Top_Peak = 1 then Rob := Rob*(Rpeak + Aux1*(1.0 - Rpeak));
If Bot_Peak = 1 then Rob := Rob*(1.0 - Aux1*(1.0 - Rpeak));
Rob := Rob/Beta;

```

{Reatividade Térmica}

```
Rot := (Alfaf*(Y[9] - Tfo) + Alfaw*(Y[12] - Two) )/Beta ;
```

{Reatividade Fictícia}

```
Rofict := -Rob - Rot ;
```

{reatividade}

```
Reat := Rob + Rot + Rofict;
Y[20] := Reat;
Y[10] := 9999.999;

```

```
U[1] := 0.0; ErO[1] := 0.0; UrO[1] := 0.0; ErO[3] := 0.0;
                                UrO[3] := 0.0; UrO[4] := 0.0;

```

```
for i:=3 to 6 do Y[i] := 0.0;
for i:= 17 to 18 do Y[i] := 0.0;
for i:= 1 to 21 do Yp[i] := 0.0;

```

end;
end;

end;

```
procedure DERIVADAS(var Y:VetEst;var U:VetCon;var P:parlist;var Yp:VetEst;
var Yt:VetEst);
```

var

```
Aux1, Reat, Rob, Rot : real;
```

begin

```
with P do begin
with C do begin
```

{ ** Reator ** }

{ * Barras de Controle * }

{ Reatividade das Barras }

```

If Y[8] > Nb then Y[8] := Nb;
If Y[8] < 0.0 then Y[8] := 0.0;
Aux1 := Trunc(Y[8])/Nb;
If Aux1 > 1.0 then Aux1 := 1.0;
Rob := 0.5*Romax*(1.0 - cos(Pi*Aux1));
If Top_Peak = 1 then Rob := Rob*(Rpeak + Aux1*(1.0 - Rpeak));
If Bot_Peak = 1 then Rob := Rob*(1.0 - Aux1*(1.0 - Rpeak));
Rob := Rob/Beta;

```

{ Reatividade do Refrigerante }

```
Rot := (Alfaf*(Yt[9] - Tfo) + Alfaw*(Yt[12] - Two) )/Beta ;
```

{ Reatividade Total }

```
Reat := Rob + Rot + Rofict;
Y[20] := Reat;
```

{ ** Neutronica ** }

{ * Nucleo * }

{ Potencia Nuclear }

```
Y[7] := (Tl1f*( A11*Yt[1] + A12*Yt[2] ))/(Beta*(1.0 - Reat)) ;
```

```
Y[16] := Y[7] + Y[15]; ( Potencia Total )
```

{ Concentração dos Precursores do Grupo 1 a 6 }

```
Yp[1] := Be1*Yt[7]/Tl1f - A11*Yt[1] ;
```

```

Yp[12] := Be2*Yt[7]/T1if - A12*Yt[2] ;
(Yp[3] := Be3*Yt[7]/T1if - A13*Yt[3] ;
Yp[4] := Be4*Yt[7]/T1if - A14*Yt[4] ;
Yp[5] := Be5*Yt[7]/T1if - A15*Yt[5] ;
Yp[6] := Be6*Yt[7]/T1if - A16*Yt[6] ;) {Aprox. 2 grupos precursores}

      ( ** Termohidraulica ** )
      ( * Primario * )

      { Temperatura do Combustivel }
Yp[9] := ((Ff*Yt[7]+Yt[15])*Q0 - (Afw1 + Afw2)*Kfw*(Yt[9] - Yt[12]))/
      (Mfu*CsF) ;

      { Temperatura do Refrigerante do Nucleo }
Yp[12] := ((Fw1 + Fw2)*Yt[7]*Q0 - 2.0*Wa*Csw*(Yt[12] - Yt[11]) +
      (Afw1 + Afw2)*Kfw*(Yt[9] - Yt[12]))/((Mw1+Mw2)*Csw) ;

      { Temperatura de Saida do Nucleo }
Y[14] := 2.0*Yt[12] - Yt[11] ;

      { Taxa de Resfriamento de Entrada }
Yp[11] := dTemp_entrada; { C/seg}
      end;
      end;
end;

procedure INTEGRACAO (var Y:VetEst; var Yt:VetEst;
      var U:VetCon; var P:parlist ) ;
var
  Yp1, Yp2, Yp3, Yp4 : array[1..NumEq] of real ;
begin
  For i := 1 to NumEq do begin
    Yt[i] := Y[i] ;
  end;

  Derivadas(Y,U,P,Yp,Yt);

  For i := 1 to NumEq do begin
    Yp1[i] := Yp[i]*Deltat;
    Yt[i] := Y[i] + Yp1[i]/2.0;
  end;

  Derivadas(Y,U,P,Yp,Yt);
  For i := 1 to NumEq do begin
    Yp2[i] := Yp[i]*Deltat;
    Yt[i] := Yt[i] + Yp2[i]/2.0;
  end;

  Derivadas(Y,U,P,Yp,Yt);
  For i := 1 to NumEq do begin
    Yp3[i] := Yp[i]*Deltat;
    Yt[i] := Yt[i] + Yp3[i]/2.0;
  end;

  Derivadas(Y,U,P,Yp,Yt);
  For i := 1 to NumEq do begin
    Yp4[i] := Yp[i]*Deltat;
    Yt[i] := Yt[i] + (Yp1[i] + 2.0*(Yp2[i] + Yp3[i]) + Yp4[i])/6.0;
  end;
end;

end;

procedure LIMPAR_LINHA;
begin
  Gotoxy( 1,1); Write(' ');
  Gotoxy(41,1); Write(' ');
  Gotoxy(1,1);

```

end;

```
procedure MOSTRADOR(var Y: VetEst; var U: VetCon);
```

```
var aux1 : real;
```

```
begin
```

```
aux1 := 100.0*Y[16];
```

```
Gotoxy( 1, 1);
```

```
Gotoxy(69, 3); Write('F%=', aux1 :8:3);
```

```
Gotoxy(69, 5); Write('T =', Y[10]:8:3);
```

```
Gotoxy(69, 7); Write('Ro=', Y[20]:8:5);
```

```
Gotoxy(69, 9); Write('V =', Yp[8]:8:3);
```

```
Gotoxy(69, 11); Write('BC=', Y[ 8]:8:3);
```

```
Gotoxy(69, 13); Write('C1=', Y[ 1]:8:3);
```

```
Gotoxy(69, 15); Write('C2=', Y[ 2]:8:3);
```

```
Gotoxy(69, 17); Write('Ti=', Y[11]:8:3);
```

```
Gotoxy(69, 19); Write('Tm=', Y[12]:8:3);
```

```
Gotoxy(69, 21); Write('To=', Y[14]:8:3);
```

```
Gotoxy(69, 23); Write('Tf=', Y[ 9]:8:3);
```

```
end;
```

```
procedure QUADRO_MOSTRADOR;
```

```
begin
```

```
Draw(539, 10, 635, 10, 1);
```

```
Draw(539, 188, 635, 188, 1);
```

```
Draw(539, 10, 539, 188, 1);
```

```
Draw(635, 10, 635, 188, 1);
```

```
end;
```

```
procedure MODIFICACOES(var Y: VetEst; var U: VetCon; var P: parlist);
```

```
var
```

```
aux1 : real;
```

```
temp1, temp2, indice : integer;
```

```
modo : string[1];
```

```
label repete;
```

```
begin
```

```
with P do begin
```

```
with C do begin
```

```
Limpar_Linha;
```

```
Write('MUDAR: Variavel/Barra controle/Automatico/ou Encerra?',  
      '(entre: V/B/A/E):');
```

```
Read(modo);
```

```
If (modo='E') or (modo='e') then Fim := 0;
```

```
Shutdown := False;
```

```
If (modo='V') or (modo='v') then
```

```
begin
```

```
temp1 := 0;
```

```
While temp1 = 0 do
```

```
begin
```

```
Limpar_Linha;
```

```
Write('Entre indice:'); Read(temp2);
```

```
repete: If (temp2<=0) or (temp2 > NumEq) then
```

```
begin
```

```
Limpar_Linha;
```

```
Write('Indice FORA DO RANGE: 0 -', NumEq:3, ' Entre de novo:')
```

```
Read(temp2); goto repete;
```

```
end else
```

```
begin
```

```
Limpar_Linha;
```

```
Write('Variav[temp2], '=' , Y[temp2]:8:3, ' NOVO VALOR=');
```

```
Read(Y[temp2]);
```

```
Limpar_Linha;
```

```
Write('Mudar OUTRA VARIAVEL?(entre 1=Nao ou 0=Sim):');
```

```
Read(temp1);
```

```
end;
```

```
end; ( while )
```

```
end; { modo V }
Limpar_Linha;
```

```
If (modo='B') or (modo='b') then
begin
  Limpar_Linha;
  Write(' BARRA DE CONTROLE=',Yp[8]:8:3,
        ' Entre VELOCIDADE:(-1,0,1):');
  Read(temp2);
  case temp2 of -1: Yp[8]:= - Veloc_Barra;
                0: Yp[8]:=  0.0          ;
                1: Yp[8]:=  Veloc_Barra;
  end;
end; {modo B}
Limpar_Linha;
```

```
If (modo='A') or (modo='a') then
begin

  Limpar_Linha;
  Aux1 := 100.0*Y[16];
  Write(' POTENCIA ATUAL =',Aux1:8:3,' Entre NOVA POTENCIA:');
  Read(Aux1); PotRef := Aux1*0.01;
  Auto := 1;
end else Auto := 0;
Limpar_Linha;
end;
end;
end;
```

```
procedure CONTROLE(var Y:VetEst ;var U:VetCon ;var Er:VetCon;var Er0:Vetcon;
var Ur:VetCon;var Ur0:VetCon;var F:parlist;var C:ContList);
```

```
erro, Fg : real;
in
with F do begin
with C do begin
```

```
  ( Controle Automatico das Barras )
  (Canal de Potencia)
```

```
Er[1] := PotRef - Y[16];
```

```
( Ur[1] := Ur0[1] + (Er[1] - Ur0[1])*DeltaT/Atraso;
Ur0[1] := Ur[1]; ) { nao utilizado para IEAR-1 }
```

```
Erro := Ur[1]; ( Erro na Potencia )
```

```
ER-1) Erro := K1*Er[1];
```

```
If Erro > ErroMax then Erro := ErroMax;
( Compensacao do Periodo )
```

```
If Erro > 0.0 then begin
  Erro := Erro - K2/Y[10];
```

```
If Erro < 0.0 then Erro := 0.0; end;
```

```
( Programa de Velocidade das Barras )
```

```
If Erro >= 0.0 then ( Erro Positivo )
begin
```

```
  If Erro <= Banda1 then Yp[8] := 0.0;
```

```
  If (Erro > Banda1) and (Erro<= Banda2) then
  begin
```

```
    If Yp[8] < Veloc_Barra then Yp[8] := 0.0;
```

```
    If Yp[8] >= Veloc_Barra then Yp[8] := Veloc_Barra;
```

```
  end;
```

```
  If (Erro > Banda2 ) then Yp[8] := Veloc_Barra;
```

```
end;
```

```
If Erro < 0.0 then ( Erro Negativo )
```

```

begin
  If Erro >= -Banda1 then Yp[8] := 0.0;
  If (Erro < -Banda1) and (Erro >= -Banda2) then
    begin
      If Yp[8] > -Veloc_Barra then Yp[8] := 0.0;
      If Yp[8] <= -Veloc_Barra then Yp[8] := -Veloc_Barra;
    end;
  If (Erro < -Banda2) then Yp[8] := -Veloc_Barra;
end;
end;
end;
; ( controle )

```

```

procedure SETAR(Y,Yp:VetEst; U:VetCon; P:Parlist);
begin

```

```

  with P do begin
    Arq[1] := Y[1];
    Arq[2] := Y[2];
    Arq[3] := Y[3];
    Arq[4] := Y[4];
    Arq[5] := Y[5];
    Arq[6] := Y[6];
    Arq[7] := Y[7];
    Arq[8] := Y[8];
    Arq[9] := Y[9];
    Arq[10] := Y[10];
    Arq[11] := Y[11];
    Arq[12] := Y[12];
    Arq[13] := Y[13];
    Arq[14] := Y[14];
    Arq[15] := Y[15];
    Arq[16] := Y[16];
    Arq[17] := Y[17];
    Arq[18] := Y[18];

```

```

    Imp[1] := Y[7];
    Imp[2] := Y[8];
    Imp[3] := Y[9];
    Imp[4] := Y[10];
    Imp[5] := Y[11];
    Imp[6] := Y[12];
    Imp[7] := Y[14];
    Imp[8] := Y[15];
    Imp[9] := Y[16];

```

```

    Video[1] := Y[1];
    Video[2] := Y[2];
    Video[3] := Y[3];
    Video[4] := Y[4];
    Video[5] := Y[5];
    Video[6] := Y[6];
    Video[7] := Y[7];
    Video[8] := Y[8];
    Video[9] := Y[9];
    Video[10] := Y[10];
    Video[11] := Y[11];
    Video[12] := Y[12];
    Video[14] := Y[14];
    Video[15] := Y[15];
    Video[16] := Y[16];

```

```

  end;

```

```

end;

```

```

procedure IMPRESSAO(Y,Yp:VetEst; U:Vetcon; Tempo:real);
begin

```

```

  Writeln(Lst,'=====>', Tempo);

```

```

Writeln;
For i := 1 to Nimp do begin
    Writeln(Lst, Imp[i]);
end;
end;

```

```

procedure IDENTIFICACAO;

```

```

var
    I: Integer;
begin
    Writeln(Ident, Titulo);
    Writeln(Ident, NumCurv);
    Writeln(Ident, NumReg);
    Writeln(Ident, Deltat);
    For I := 1 to NumCurv do begin
        Writeln(Ident, Variav[I]);
    end;
    Close(Ident);
end;

```

```

procedure SAIDA;

```

```

begin
    Assign(Ident, ArqVar);
    Assign(Dados, ArqDat);
    Rewrite(Ident);
    Rewrite(Dados);
end;

```

```

procedure ARQUIVO_DADOS;

```

```

begin
    Write(Dados, Arq);
end;

```

```

Procedure Monta_UmNdt;

```

```

begin
    UmNdt[1] := 1.0;
    For i := 2 to Numdt do UmNdt[i] := exp(-1.2*ln(1.0+(i-1)*Deltat));
    For i := 1 to Numdt do PotNdt[i] := 0.0;
end;

```

```

Procedure CALOR_RESIDUAL;

```

```

var Aux1 : real;
begin
    with P do begin
        with C do begin
            Y[15] := PotRel*( exp(-0.2*ln(Temporel))
                            -exp(-0.2*ln(Temporel+Tresid)));
            For i := 2 to Numdt do PotNdt[Numdt+2-i] := PotNdt[Numdt+1-i];
            PotNdt[1] := Y[16];
            Aux1 := 0.0;
            for i := 1 to Numdt do Aux1 := Aux1 + PotNdt[i]*UmNdt[i];
            Y[15] := Y[15] + Fresid*0.2*Aux1*Deltat;

            Temporel := Temporel + Deltat;
            If Temporel > (Numdt*Deltat) then begin
                Temporel := 1.0;
                Tresid := Tresid + Numdt*Deltat;
                PotRel := Y[15];
                For i := 1 to Numdt do PotNdt[i] := 0.0; end;
            end;
        end;
    end;
end;

```

```

Procedure CALCULA_PERIODO;

```

```

var Aux1, Aux2 : real;
    N, N1 : integer;

```

```

Begin
Checa_periodo := Checa_Periodo + 1;
If Checa_Periodo = 3 then Checa_Periodo := 0;

N := 5; { maximo 10 }
N1 := N-1;
with P do begin
  For i := 1 to N1 do PotAcum2[i] := PotAcum2[i+1];
  PotAcum2[N] := PotAcum1[1];
  For i := 1 to N1 do PotAcum1[i] := PotAcum1[i+1];
  PotAcum1[N] := Y[7];
  If Checa_periodo = 0 then
    begin
      Aux1 := 0.0;
      For i := 1 to N do Aux1 := Aux1 + PotAcum1[i];
      Aux2 := 0.0;
      For i := 1 to N do Aux2 := Aux2 + PotAcum2[i];
      If Abs(Aux1-Aux2) > 1.0E-20 then
        Y[10] := (N*Deltat*Aux1)/(Aux1 - Aux2);

      If Y[10] > 9999.999 then Y[10] := 9999.999;
      If Y[10] <= 0.0 then Y[10] := 9999.999;
      If (Y[10] < Periodo_Critico) and (Shutdown=False) then begin
        Shutdown:=TRUE;
        Gotoxy(10,1);
        Write(' CUIDADO: PERIODO CRITICO ! BARRAS DE CONTROLE CAIRAM' );
        Y[8] := 0.0; Yp[8] := 0.0; end;
      end; { If }
    end;
End; { Calcula_Periodo }

```

{PROGRAMA PRINCIPAL}

Graf_Vid.Pas}

IN

```

HiRes; HiResColor(15);
Inicializacao(P);
Parametros(P,D);
CONDICOES(Y,Ur,U,Er,Er0,P,C,Ur0);

Tempo := 0.0 ;
NumReg := 0;
Flag := 0;

If (OpGrav = 'S' ) or (OpGrav = 's' ) then Saida;
Setar(Y,Yp,U,P);
Definicao_Curvas;
Inicializacao_Grafico;
Seta_Variaveis;
HiRes; HiResColor(15);
Monta_Quadro;
QUADRO_MOSTRADOR;
Escala;
Monta_UmNdt;
Temporel := 1.0;
fim := 1;
Checa_Periodo := 0;

while fim = 1 do
  begin
    if tempo >= temposim then fim := 0;
    If KeyPressed then MODIFICACOES(Y,U,P);
    INTEGRACAO(Y,Yt,U,P);
    If KeyPressed then MODIFICACOES(Y,U,P);
    CALCULA_PERIODO;
  end;

```

```
    If KeyPressed then MODIFICACOES(Y,U,P);
CALOR_RESIDUAL;
    If KeyPressed then MODIFICACOES(Y,U,P);
MOSTRADOR(Y,U);
    If KeyPressed then MODIFICACOES(Y,U,P);
If Auto=1 then CONTROLE(Y,U,Er,ErO,Ur,UrO,P,C);
If (OpImpr = 'S') or (OpImpr = 's') and ((Numreg mod Ninterv) = 0)
then IMPRESSAO(Y,Yp,U,Tempo);
gotoxy(68,25); write('T(s)=', tempo:7:3); gotoxy(1,1);
Tempo := Tempo + Deltat;
    If KeyPressed then MODIFICACOES(Y,U,P);
SETAR(Y,Yp,U,P);
CALCULA_IMPRIME;
Xplt := Xplt + Deltat;
NumReg := NumReg + 1;
If (OpGrav = 'S' ) or (OpGrav = 's')
    and ((Numreg mod Ninterv) = 0)
    then ARQUIVO_DADOS;
    If KeyPressed then MODIFICACOES(Y,U,P);
end;

If (OpGrav = 'S' ) or (OpGrav = 's') then begin
    IDENTIFICACAO; CLOSE(Dados); end;
```