

BR 90 22 825

*SSN 0101-3084



CNEN/SP

ipen Instituto de Pesquisas
Energéticas e Nucleares

PC-REATOR – PROGRAMA PARA SIMULAÇÃO DE TRANSIENTES

Horácio NAKATA

IPEN-PUB -- 273.

PUBLICAÇÃO IPEN 273

OUTUBRO/1989

SÃO PAULO
1989

PUBLICAÇÃO IPEN 273

OUTUBRO/1989

PC-REATOR – PROGRAMA PARA SIMULAÇÃO DE TRANSIENTES

Horácio NAKATA

DEPARTAMENTO DE TECNOLOGIA DE REATORES

**CNEN/SP
INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES
SÃO PAULO – BRASIL**

Série PUBLICAÇÃO IPEN

INIS Categories and Descriptors

E36.00

MTR REACTOR

P CODES

SIMULATION

TRANSIENTS

IPEN - Doc - 3467

Aprovado para publicação em 23/08/89.

Nota: A redação, ortografia, conceitos e revisão final são de responsabilidade do(s) autor(es).

PC-REATOR - PROGRAMA PARA SIMULAÇÃO DE TRANSIENTES

Horácio NAKATA

**COMISSÃO NACIONAL DE ENERGIA NUCLEAR - CNEN/SP
INSTITUTO DE PESQUISAS ENERGÉTICAS E NUCLEARES - IPEN
Caixa Postal 11049 - Pinheiros
05499 - São Paulo - BRASIL**

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 microcomputadores IBM-PC, e apresenta grande flexibilidade na sua utilização, podendo ser interrompido a qualquer instante para teste e modificação de parâmetros de operação. Com esse simulador os operadores podem adquirir experiência sem incorrer nos riscos que acarretariam em operação real dos reatores de pesquisa.

PC-REATOR - REACTOR CORE TRANSIENT SIMULATION CODE

Horácio NAKATA

**COMISSÃO NACIONAL DE ENERGIA NUCLEAR - CNEN/SP
INSTITUTO DE PESQUISAS ENERGETICAS E NUCLEARES- IPEN
Caixa Postal 11049 - Pinheiros
05499 - São Paulo - BRASIL**

ABSTRACT

PC-REATOR, a reactor core transient simulation code has been developed for the real-time operator training on a IBM-PC microcomputer. The program presents capabilities for on-line exchange of the operating parameters during the transient simulation, by friendly keyboard instructions. The model is based on the point-kinetics approximation, with 2 delayed neutron precursors and up to 11 decay power generating groups.

1. 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 pode ser feita manualmente pelos operadores, observando-se 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, o 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 precursores. 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 por 11 grupos de produtos de fissão, 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 pode ser interrompido a qualquer momento para mudança de qualquer parâmetro ou regime de operação. 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.

2. METODOLOGIA UTILIZADA

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

As equações são explicitamente escritas como:

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

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

onde:

- $P(t)$ = potência total do núcleo,
- $\rho(t)$ = reatividade total do núcleo,
- $\beta(t)$ = fração efetiva de neutrons atrasados,
- Λ = vida média dos neutrons,
- β_i = fração efetiva de neutrons atrasados do grupo i ,
- λ_i = constante de decaimento dos precursores de neutrons atrasados 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 satisfatória em relação aos resultados obtidos com 6 grupos de precursores, em transientes com duração de algumas dezenas de segundos.

A potência residual é calculada através de equacionamento dos produtos de fissão em até 11 grupos de decaimento, utilizando os parâmetros da referência 2, os quais representam com boa aproximação a curva de decaimento experimental. As equações são apresentadas abaixo.

$$\frac{d}{dt} H_j(t) = -\lambda_j^H H_j(t) + E_j P(t) \quad , j=1, \dots, 11 \quad (3)$$

onde

- $H_j(t)$ = calor de decaimento do grupo j ,
- λ_j^H = constante de decaimento do grupo j ,
- E_j = fração efetiva de energia de decaimento do grupo j .

A potência total do núcleo em qualquer instante é calculada pela soma da potência nuclear instantânea e o calor de decaimento dos produtos de fissão como,

$$P_{ef}(t) = \left(1 - \sum_{j=1}^{11} E_j \right) P(t) + \sum_{j=1}^{11} \lambda_j^H H_j. \quad (4)$$

Para se obter tempo de processamento razoavelmente prático sem comprometimento dos resultados finais, pesquisou-se o número mínimo de grupos de calor de decaimento chegando-se à conclusão que 6 grupos representam adequadamente o comportamento do calor residual em transientes de algumas centenas de segundos.

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

$$\rho(t) = \rho_b(t) + \rho_f(t) + \rho_m(t), \quad (5)$$

onde

$$\begin{aligned} \rho_b(t) &= \text{reatividade da barra de controle,} \\ \rho_f(t) &= \text{reatividade do combustível} \\ \rho_m(t) &= \text{reatividade do moderador.} \end{aligned}$$

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

$$\rho(t) = \frac{\rho_{\max}}{2} \left[1 - \cos\left(\frac{\pi z(t)}{H}\right) \right] f(z(t), a), \quad (6)$$

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,

$$f(z(t), a) = a + \frac{z(t)}{H} (1 - a) \quad (7)$$

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,

$$f(z(t), a) = 1 - \frac{z(t)}{H} (1 - a) \quad (8)$$

onde a é o parâmetro com as mesmas propriedades da equação anterior.

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

$$\rho_m(t) = a_m [T_m(t) - T_{mo}] \quad (9)$$

$$\rho_f(t) = a_f [T_f(t) - T_{fo}] \quad (10)$$

onde

- a_m = coeficiente de reatividade do moderador,
- a_f = coeficiente de reatividade do combustível,
- $T_m(t)$ = temperatura do moderador,
- $T_f(t)$ = temperatura do combustível,
- T_{mo} = temperatura inicial do moderador,
- T_{fo} = temperatura inicial do combustível.

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

$$M_f C_f \frac{d}{dt} T_f(t) = F_f P_{ef}(t) + A_f K_f [T_f(t) - T_m(t)] \quad (11)$$

$$M_m C_m \frac{d}{dt} T_m(t) = (1 - F_f) P_{ef}(t) + W C_m [T_{m_{out}}(t) - T_{m_{in}}(t)] \quad (12)$$

onde

- M_f = massa do combustível,
- M_m = massa do moderador,
- C_f = calor específico do combustível,
- C_m = calor específico do moderador,
- F_f = fração da potência gerada no combustível,
- A_f = área de transferência de calor do combustível,
- K_f = coeficiente efetivo de transmissão de calor por convecção,

W = fluxo mássico 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 quarta ordem, com aproximação *Prompt-Jump* na equação da potência total, 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/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 a malha de controle implementada no programa.

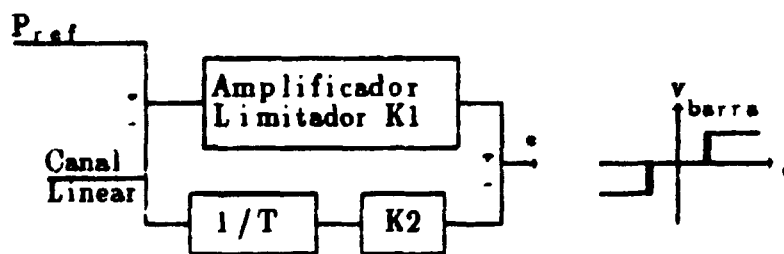


Figura 1 - Malha de Controle Automático do PC-REATOR.

O controle do reator é feito por sinal proporcional ao erro, compensado pelo sinal proporcional ao inverso do período. O amplificador limitador tem ganho $K1$, 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 $K2$, o qual amplifica o inverso do período do reator. O valor de $K1$ nos reatores tipo MTR é 100 e o limite é aproximadamente 0,5% da potência nominal.

3. DESCRIÇÃO DAS ENTRADAS DO PC-REATOR

As opções do programa PC-REATOR são, na maior parte, pré-definidas, porém, pode-se, durante a execução, redefini-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 (segundos),
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,
16	Potência total do reator.

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	modifica valor de variável,
B	controle manual da velocidade da barra de controle,
A	controle automático para novo nível de potência.

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 programa continua calculando a potência de decaimento e das fissões de neutrons atrasados.

4. RESULTADOS E CONCLUSÕES

O método utilizado no programa PC-REATOR, é bastante simples e é capaz de representar em tempo real transientes nucleares mais importantes do ponto de vista dos operadores. A metodologia produz praticamente os mesmos resultados que os obtidos por solução da cinética puntual pelo método de Hansen/1/ em 6 grupos de precursores de neutrons atrasados, e o nível de calor residual proveniente de produtos de fissão são praticamente coincidentes com os resultados da American Nuclear Society/2/.

O programa PC-REATOR fornece a visualização dos parâmetros mais importantes durante o transiente permitindo julgamento mais racional do operador quanto aos fenômenos físicos envolvidos no processo, facilitando dessa

forma tomada de ação corretiva consciente durante o desenvolvimento da simulação de transientes típicos de operação.

Um exemplo de simulação de operação do Reator IEA-R1 é ilustrado na Figura 2 com controle automático e controle manual. Em controle automático o nível de potência foi aumentado de 20% para 100% da nominal em aproximadamente 50 segundos, ao passo que em controle manual o mesmo aumento não foi possível de ser efetuado em menos de 100 segundos, ao fim dos quais a potência não se encontra ainda estabilizada.

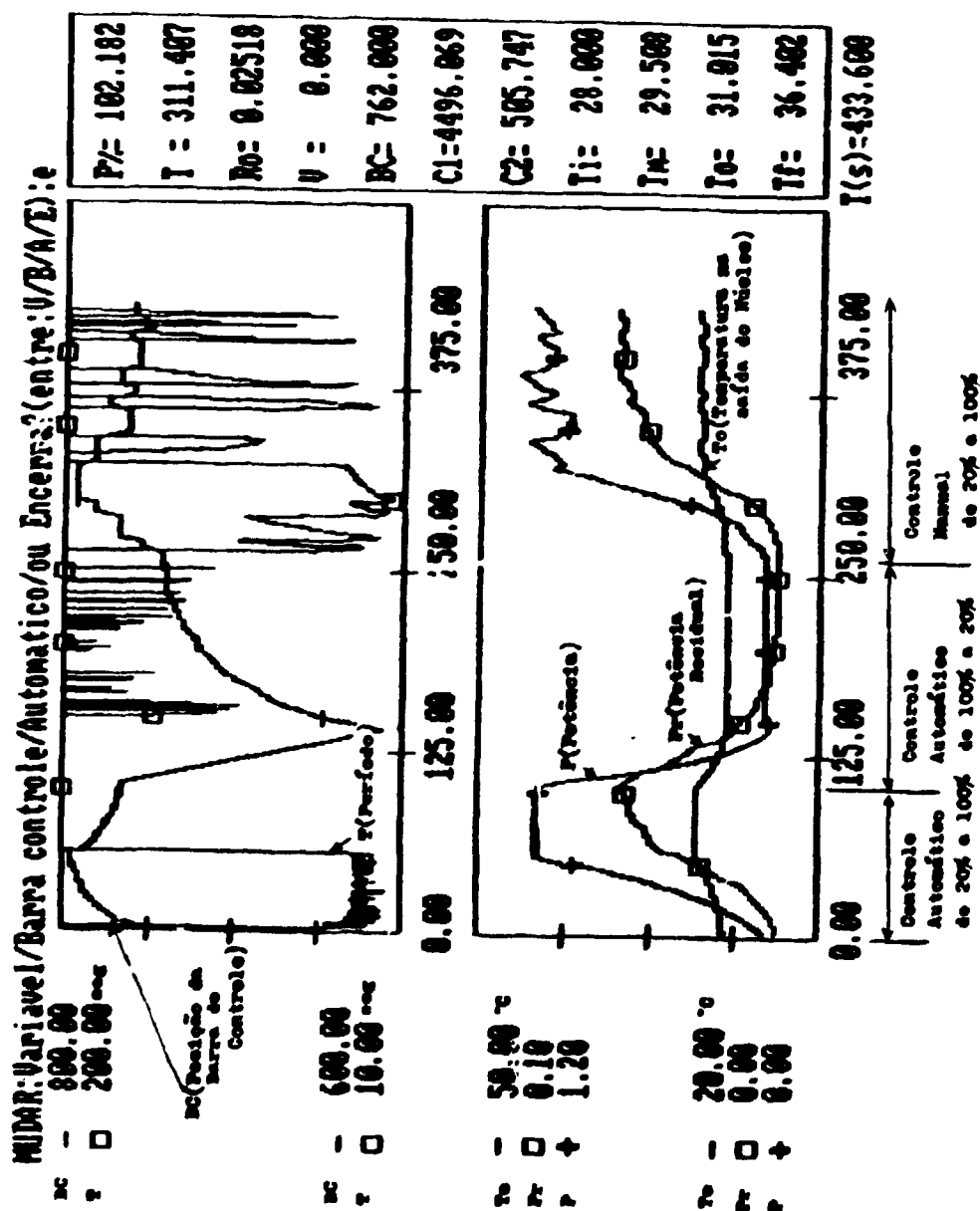


Figura 2 -- Exemplo de Saída no Vídeo Durante Simulação com PC-REATOR

5. REFERÊNCIAS BIBLIOGRÁFICAS

- 1 - HETRICK, D.L. Dynamics of nuclear reactors. Chicago, University Press, 1971.
- 2 - TRAC-PD2 - An advanced best estimate computer program for PWR-LOCA analysis. Los Alamos, NM, Los Alamos Scientific Laboratory, Apr., 1981. (NUREG/CR-2054, LA-8709-MS).

APÊNDICE**Listagem do Programa PC-REATOR em Pascal para IBM-PC.**

```

Program REATOR(input,output)
  ( simulacao dinamica do nucleo em 2 GRUPOS DE PRECURSORES)
  ( Modelo para simular REATOR com controle de barras )
  ( dados do REATOR IEA-R1 )
  ( Com Valor Residual do TRAC/PF1 Out/88 Horacio Nakata )
  ( 5 GRUPOS MAIS CONSTANTE )

  (
    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) :=
    Variav(18) :=
    Variav(19) :=
    Variav(20) :=
    Variav(21) := Potencia Residual do Grupo 1
    Variav(22) := Potencia Residual do Grupo 2
    Variav(23) := Potencia Residual do Grupo 3
    Variav(24) := Potencia Residual do Grupo 4
    Variav(25) := Potencia Residual do Grupo 5
    Variav(26) := Potencia Residual do Grupo 6
    Variav(27) := Potencia Residual do Grupo 7
    Variav(28) := Potencia Residual do Grupo 8
    Variav(29) := Potencia Residual do Grupo 9
    Variav(30) := Potencia Residual do Grupo 10
    Variav(31) := Potencia Residual do Grupo 11
  )

Const
  Pi = 3.14159
  Nimp = 9
  NumEq = 31
  NumCon = 5
  NumCurv = 31
  NumRes = 5 ( MAX 11 grupos, mas 6 grupos eh bom )
  ( de 6 a 11 considerados constantes )

type
  ParList = record
    Afw1 , Afw2 , Kfw , Ff , Fw1 , Fw2 , Mfu , Mw1 , Mw2 : real ;
    Q0 , Beta , Tinf , Cef , Csw : real ;
    Tfo , Tw0 , Ref , Pot , We : real ;
    A11 , A12 , A13 , A14 , A15 , A16 : real ;
    Be1 , Be2 , Be3 , Be4 , Be5 , Be6 : real ;
    Crea , Frea : array(1..11) of real ;
    FreeTotal : real ;
  end ;

  ConList = record
    Romax , Nb , Nbarra0 , Top_Peak , Bot_Peak , Rpeak : real ;
    PotRef , Atreso , Valoc_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;
  YresCte              : real;
  U , Er , Er0 , Ur , Ur0 : VetCon;
  Rofict , Temporef    : real;
  P                   : Parlist;
  I , II              : integer;
  car                 : char;
  C                   : contlist;
  Tempo , Temposim , Deitac : real;
  PotAcum1 , PotAcum2    : array(1..10) of real;
  Chece_periodes       : integer;
  Periode_Critico      : real;
  UmNet , PotNet       : 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;
  Variev              : Varievel;
  Xpit , Xpitmn , Xpitmx , Ovx : real;
  Xseq , Xdir , Px , Px0 , Ngraf , Contador , Npx , Delcont : integer;
  Ypit , Ypitmn , Ypitmx , Ovy : array(1..20,1..20) of real;
  Ypitmn1 , Ypitmx1    : 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;
  OpGraf , OpGraf , OpImpr : string(1);

Procedure ENTRADA;
var
  I , Ent1             : integer;
  Entr1 , Entr2 , Entr3 : real;
  Arq1                 : string(15);
begin
  with P do begin
    ClrScr;
    for I := 1 to 4 do writeLn;
    WriteLn('          RT1 - SISTEMAS DE CONTROLE          ');
    WriteLn('          *****');
    WriteLn;
    WriteLn('          Julho/85          ');
    WriteLn;
    WriteLn('          Simulacao Dinamica do Reator          ');
    WriteLn('          -----');
    WriteLn;
    Write('Entre Tempo de Duracao da Simulacao:'); readLn(Ent1);
    If Ent1 (<) 0.0 then Temposim := Ent1;
    Write('Entre Intervalo de Integracao          '); readLn(Entr2);
    If Entr2 (<) 0.0 then Deitac := Ent2;
    Write('Entre Potencia Consumida          '); readLn(Entr3);
    If Entr3 (<) 0.0 then pot := Entr3;
  end;
end;

```



```

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(Ent1);
if Ent1 <> 0.0 then Nintery := Ent1;
Write('Entre Titulo para a Simulacao :') ; readln(Titulo);
end;

end;

Procedure INICIALIZACAO(var P: Parlist);
var
  I : Integer;
  Of: String(1);
begin
  With P do begin
    ClrScr;
    For I := 1 to 4 do writeln;
    Write('          RTI - SISTEMAS JE CONTROLE          ');
    Write('          *****          ');
    Write('          ');
    Write('          Junho/88          ');
    Write('          ');
    Write('          Simulacao Dinamica do Reator          ');
    Write('          -----          ');

    Temposim := 100.0 ; Deltat := 0.2 ; pot := 0.2 ;
    Nintery := 1 ; OpGrav := 'N' ; OpGraf := 'S' ;
    OpImpr := 'N' ; ArqVar := 'b:Arq.1de' ; Shutdown:=False ;
    ArqDat := 'b:Arq.0at' ;

    gotoxy(26,13); writeln('C E F A U L T');
    gotoxy(28,14); writeln(' * * * * *');
    gotoxy(20,16); writeln('Tempo de Duracao de 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: ',
      Nintery:2);
    gotoxy(20,24); write ('ASSUMIR VALORES DO DEFAULT? <S/N>:');
    Readln(Of);
    if ( Of = 'N' ) or ( Of = 'n' ) then ENTRADA;

    Variev(1) := ' Conc. Prec. Grupo 1          ';
    Variev(2) := ' Conc. Prec. Grupo 2          ';
    Variev(3) := ' Conc. Prec. Grupo 3          ';
    Variev(4) := ' Conc. Prec. Grupo 4          ';
    Variev(5) := ' Conc. Prec. Grupo 5          ';
    Variev(6) := ' Conc. Prec. Grupo 6          ';
    Variev(7) := ' Pot. Nuclear Normalizado          ';
  end;
end;

```

```

Variav(8) := Pos. das Barras de Controle
Variav(9) := Temp. do Combustivel
Variav(10) := Período Instantâneo do Reator
Variav(11) := Temp. do Refrig. na Entrada do Reator
Variav(12) := Temp. do Refrig. Média 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) :=
Variav(18) :=
Variav(19) :=
Variav(20) :=
Variav(21) := Potencia Residual do Grupo 1
Variav(22) := Potencia Residual do Grupo 2
Variav(23) := Potencia Residual do Grupo 3
Variav(24) := Potencia Residual do Grupo 4
Variav(25) := Potencia Residual do Grupo 5
Variav(26) := Potencia Residual do Grupo 6
Variav(27) := Potencia Residual do Grupo 7
Variav(28) := Potencia Residual do Grupo 8
Variav(29) := Potencia Residual do Grupo 9
Variav(30) := Potencia Residual do Grupo 10
Variav(31) := Potencia Residual do Grupo 11
end.

end.

procedure PARAMETROS (var P : parlist ; var C : conlist ) ;
var I : Integer.
begin
  with P do begin
    with C do begin

      Q0 := 2.0E+08 ; [ Potencia do Nucleo ]

      [ Parametros de Controle ]
      Nb := 1000.0 ; Mbarra0 := 0.75*Nb ; Romax := 0.03000 ;
      Top_Peak := 1 ; Bot_Peak := 0 ; Rpeak := 0.72 ;
      Veloc_Barra := 4.0 ; Período_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 Cinética ]
      A11 := 0.0252 ; A12 := 0.588 ; A13 := 0.115 ;
      A14 := 0.311 ; A15 := 1.4 ; A16 := 3.87 ;
      Be1 := 0.002319 ; Be2 := 0.005485 ; Be3 := 0.001486 ;
      Be4 := 0.003175 ; Be5 := 0.0009984 ; Be6 := 0.0002028 ;
      Tlif := 0.0000183 ; Beta := Be1 + Be2 ; [ aprox. 2 grupos ]

      [ Parametros de Calor Residual ]
      Cres( 1) := 1.772 ; Cres( 2) := 0.5774 ; Cres( 3) := 0.08743 ;
      Cres( 4) := 0.008214 ; Cres( 5) := 4.738E-04 ; Cres( 6) := 4.810E-05 ;
      Cres( 7) := 5.344E-08 ; Cres( 8) := 5.728E-07 ; Cres( 9) := 1.038E-07 ;
      Cres(10) := 2.958E-08 ; Cres(11) := 7.585E-10 ;
      Frec( 1) := 0.00298 ; Frec( 2) := 0.00825 ; Frec( 3) := 0.0155 ;
      Frec( 4) := 0.01935 ; Frec( 5) := 0.01165 ; Frec( 6) := 0.00645 ;
      Frec( 7) := 0.00231 ; Frec( 8) := 0.00184 ; Frec( 9) := 0.00085 ;
      Frec(10) := 0.00043 ; Frec(11) := 0.00057 ;
      FrecTotal := 0.6 ;
      For I := 1 to NumRee do FrecTotal := FrecTotal + Frec(I) ;

      [ Parametros Termohidraulicos ]
      dTemp_entrada := 0.0 ; T_entrada := 28.0 ; Wc := 162.0 ;
      A1f1 := -0.9E-08 ; A1f2 := -13.0E-05 ;
      Ff := 0.97 ; Fw2 := 0.015 ; Fw1 := 0.015 ;
      Mfu := 29.8 ; Mw1 := 33.0 ; Mw2 := 23.0 ;
    end ;
  end ;
end.

```

```

Afw1 := 19.05      Afw2 := 19.05
Csw  := 4180.0     Cef  := 760.0
Vfw  := 7580.0     Tw0 := 30.0
Tf0  := 30.0

end.
end.
end.

procedure CONDIC0ES (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
[ Zerando... ]
U(1) := 0.0; Er0(1) := 0.0; Ur0(1) := 0.0; Er0(3) := 0.0;
Ur0(3) := 0.0; Ur0(4) := 0.0;
for i := 1 to NumEq do Y(i) := 0.0;
for i := 1 to NumEq do Yp(i) := 0.0;

[ Neutronics ]
YresCte := 0.0;
For i := (NumRes + 1) to 11 do
YresCte := YresCte + Fres(i)*Pot. [ Color Res. Const. ]
Y(15) := FresTotal*Pot + YresCte. [ Color Residual ]
Y(7) := Pot. [ Potencia Nuclear ]
Y(16) := Pot. [ Potencia Total ]
For i := 1 to 10 do begin
PotAcum1(i) := Y(7). [ acumuladores p/ periodo ]
PotAcum2(i) := Y(7). end.

[ Concentracao de Precursores dos 8 Grupos ]
Y(1) := (Be1*Y(7))/(T11*f*A11);
Y(2) := (Be2*Y(7))/(T11*f*A12);
Y(3) := (Be3*Y(7))/(T11*f*A13);
Y(4) := (Be4*Y(7))/(T11*f*A14);
Y(5) := (Be5*Y(7))/(T11*f*A15);
Y(6) := (Be6*Y(7))/(T11*f*A16);

[ Color Residual ]
For i := 1 to NumRes do begin
ii := 20 + i;
Y(ii) := (Fres(ii)/Cres(ii))*Y(7). end.

Y(8) := TRUNC(Nbarr0);

[ Temperaturas do Primario ]
Y(11) := T_entrada. [ Temperatura de Entrada ]
Y(14) := Y(11) + Pot*Q0/(Wp*Csw). [ Temperatura de Saída ]
Y(12) := 0.5*(Y(11) + Y(14)). [ Temperatura Media ]
[ Temperatura de Combustivel ]
Y(9) := (Fp*Y(16))*Q0/((Afw1 + Afw2)*Kfw) + Y(12);

[ Reatividade dos Barras ]
Aux1 := TRUNC(Y(8))/Nb;
Rob := 0.5*Reome*(1.0 - exp(-P1*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/Bete.

[ Reatividade Termica ]

```

```

Rot := (A1fafx(Y(9) - TFD) + A1fawx(Y(12) - TwD) )/Beta .
      (Reatividade Ficticial)
Rofict := -Rob - Rot .
      (reatividade)
Reat := Rob + Rot + Rofict .
Y(20) := Reat
Y(10) := 9999.999

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(P)*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 := (A1fafx(Yt(9) - TFD) + A1fawx(Yt(12) - TwD) )/Beta .
      [ Reatividade Total ]
      Reat := Rob + Rot + Rofict;
      Y(20) := Reat;

      ( * Nucleo * )
      ( Potencia Nuclear )
      Y( 7 ) := (T11f*( A11*Yt(1) + A12*Yt(2) ))/(Beta*(1.0 - Reat));

      [ Calor Residual ]
      for i := 1 to NumRes do begin
        ii := 20 + i;
        Yp(ii) := Fres( i)*Yt(7) - Cres( i)*Yt(ii) . end;

      Y(15) := 0.0;
      for i := 1 to NumRes do Y(15) := Y(15) + Cres(i)*Yt(20 + i);
      Y(15) := Y(15) + YresCte;

      [ Potencia Total ]
      Y(16) := (1.0 - FresTotal)*Yt(7) + Y(15);

      [ Concentraçao dos Precursores do Grupo 1 e 2 ]
      Yp(1) := Bc1*Yt(7)/T11f - A11*Yt(1) .
      Yp(2) := Bc2*Yt(7)/T11f - A12*Yt(2) .

      ( ** Termohidraulica ** )
      ( * Primerio * )

      [ Temperatura do Combustivel ]
      Yp(9) := ((Ff*Yt(16))*Q0 - (Afw1 + Afw2)*Kfw*(Yt(9) - Yt(12)))/
      (Mfu*Csf) ;

      [ Temperatura de Saída do Nucleo ]
    end;
  end;
end;

```

```

Yp(14) := ((Fw1 + Fw2)*Yt(18)*00 - Ww*Gsw*(Yt(14) - Yt(11)) -
           (Afw1 + Afw2)*Kfw*(Yt(9) - Yt(12)))/((Mw1+Mw2)*Gsw) .
           { Temperatura Media do Nucleo }

Y(12) := 0.5*(Yt(11) + Yt(14)).
           { 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:par1let) ;
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) := Y(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) := Y(i) + Yp3(i)/2.0;
  end.

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

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

procedure MOSTRARSM (var Y:VetEst. var U:VetCon);
var aux1:real;
begin
  aux1 := 100.0*Y(18);
  Gotoxy(1,1);
  Gotoxy(88, 3); Write('P%', aux1:8:2);
  Gotoxy(88, 8); Write('T =', Y(10):8:2);
  Gotoxy(88, 7); Write('Ro =', Y(20):8:8);
  Gotoxy(88, 8); Write('V =', Y(8):8:3);

```

```

Gotoxy(89,11).Write('C0=',Y( 8):8:3);
Gotoxy(89,13).Write('C1=',Y( 11):8:3);
Gotoxy(89,15).Write('C2=',Y( 21):8:3);
Gotoxy(89,17).Write('T1=',Y(11):8:3);
Gotoxy(89,19).Write('T2=',Y(12):8:3);
Gotoxy(89,21).Write('T0=',Y(14):8:3);
Gotoxy(89,23).Write('Tf=',Y( 9):8:3);

eng.

procedure QUADRO_MOSTRADOR;
begin
  Draw(539, 10, 635, 10, 1);
  Draw(539, 188, 635, 188, 1);
  Draw(549, 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='a') 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(' Variavel(temp2), '=',Y(temp2):8:3,' NOVO VALOR=');
                  Read(Y[temp2]);
                  Limpar_Linha;
                  Write('Mudar OUTRA VARIAVEL?(entre 1=Na0 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=',Y(8):8:3,
                    ' Entre VELOCIDADE:(-1,0,1):');
              Read(temp2);
              case temp2 of -1: Y(8):= - Veloc_Barra;
                           0: Y(8):=  0.0
                           ;
            end;
          end;
        end;
    end;
  end;
end.

```

```

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 = ',Aux1:0:3,' Entre NOVA POTENCIA: ');
Read(Aux1); PotRef := Aux1*0.61.
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 Lr:VetCon; var Ur0:VetCon; var P:parList; var C:ContList);
var
erro, Fg : real;
begin
with P do begin
with C do begin

[ Controle Automatico das Brras ]
[ Canal de Potencia ]
Er(1) := PotRef - Y(16).

[ Ur(1) := Ur0(1) + (Er(1) - Ur0(1))*DeltaT/Atrese.
Ur0(1) := Ur(1). ] [ nao utilizado para iEAR=1 ]

Erro := Ur(1); [ Erro na Potencia ]
(iEAR=1) Erro := K1*Er(1).
if Erro > ErroMax then Erro := ErroMax.
[ Compensacao do Período ]
if Erro > 0.0 then begin
Erro := Erro - K2/Y(10);
if Erro < 0.0 then Erro := 0.0. end.

[ Programa de Velocidade das Brras ]
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.
end.
end.
end.
end.
end.
end.
end.

```

```

end i 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
  Writein(Lst, '====>', Tempo);
  WriteLn;
  For i := 1 to Nimp do begin
    Writein(Lst, Imp(i));
  end;
end;

procedure IDENTIFICAO;
var
  i: Integer;
begin
  Writein(Ident, Titulo);
  Writein(Ident, NumCurv);
  Writein(Ident, NumReg);
  Writein(Ident, DeltaT);
  For i := 1 to NumCurv do begin
    Writein(Ident, Var[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 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; { maxima 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*Delitat*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) < Período_Critico) and (Shutdown=False) then begin
                    Shutdown:=TRUE;
                    Gotoxy(10,1);
                    Write(' CUIDADO: PERÍODO CRÍTICO ! BARRAS DE CONTROLE CAIRAM' );
                    Y(8) := 0.0; Y(9) := 0.0; end;
                end; { if }
            end;
    end; { Calcula_Periodo }

    (PROGRAMA PRINCIPAL)

[8] Graf_Vid.Pas)

BEGIN

    HIRes, HIResColor(15);
    Inicializacao(P);
    Parametros(P,C);
    CONDICÕES(Y,Ur,U,Er,Er0,P,C,Ur0);

    Tempo := 0.0;
    NumReg := 0;
    flag := 0;

    If (OpGrav = 'S') or (OpGrav = 's') then Salda;
    Seta(Y,Yp,U,P);
    Definicao_Curvas;
    Inicializacao_Grafico;
    Seta_Variaveis;
    HIRes, HIResColor(15);
    Monta_Quadro;
    QUADRO_MOSTRADOR;
    Escala;
    fim := 1;
    Checa_Periodo := 0;
    while fim = 1 do
        begin
            if tempo >= tempoim then fim := 0;

```

```

      if KeyPressed then MODIFICACOES(Y,U,P).
      INTEGRACAO(Y,Yt,U,P)
      if KeyPressed then MODIFICACOES(Y,U,P).
      CALCULA_PERIODO
      if KeyPressed then MODIFICACOES(Y,U,P).
      MOSTRAR(Y,U)
      if KeyPressed then MODIFICACOES(Y,U,P).
      if Auto=1 then CONTROLE(Y,U,Er,ErD,Ur,UrD,P,G).
      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).
      if KeyPressed then MODIFICACOES(Y,U,P).
      CALCULA_IMPRIME.
      if KeyPressed then MODIFICACOES(Y,U,P).
      Xpit := Xpit + 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
      IDENTIFICAO. CLOSE(Dados). end.
    TextMode.
  END.

```