

INTRODUÇÃO À COMPUTAÇÃO CIENTÍFICA

VIº Trabalho: PVC

Tobias J. D. E. Rosa

2022

1º Período

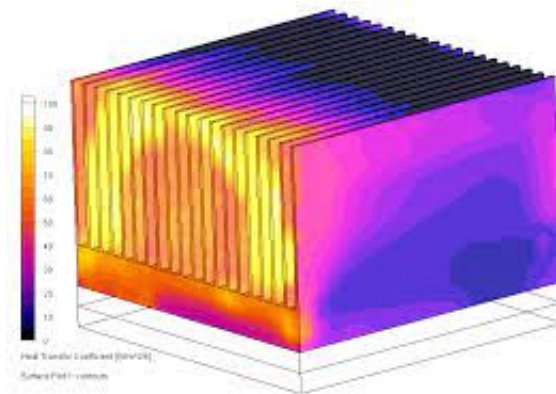
Resumo

O presente trabalho introduz brevemente o método das diferenças finitas e apresenta uma formulação para o desenvolvimento de um programa em Fortran para solução do problema de condução de calor bidimensional em regime permanente, se caracterizando como um problema de valor de contorno (PVC) com diferentes condições físicas para cada extremidade da chapa.

Solução do Problema

Um problema comum de distribuição de calor bidimensional é a distribuição de calor em um dissipador térmico eletrônico, como o ilustrado na Fig.1. Se analisarmos apenas a seção transversal de cada barra do dissipador, podemos modelar essa seção como uma placa bidimensional homogênea de espessura constante.

Figura 1 - Dissipador térmico eletrônico.



Modelagem

A equação geral da transferência de calor é escrita da seguinte forma:

$$\nabla^2 T + \frac{q}{k} = \frac{1}{\alpha} \cdot \frac{\partial T}{\partial t} \quad (1)$$

onde ∇ é o laplaciano, α é a difusividade térmica, k é a condutividade térmica e q é o fluxo de calor. No atual modelo, pretendemos analisar a distribuição espacial de calor em uma única chapa, nas direções x e z , com y sendo o comprimento longitudinal da barra de seção retangular, neste caso considerado unitário. A distribuição temporal de temperatura será desconsiderada, assumindo a condição de regime permanente em que a variação de temperatura no tempo é igual a zero e a equação de governo é simplificada para a Equação (2).

$$\frac{\partial^2}{\partial x^2} T + \frac{\partial^2}{\partial z^2} T = -\frac{q}{k} \quad (2)$$

Sendo a equação de governo uma equação diferencial parcial, um dos métodos possíveis e relativamente simples para solução é o método das diferenças finitas, em que a seção é discretizada em um número finito de pontos, sendo o número de equações igual ao número de pontos.

Método das diferenças finitas

O método das diferenças finitas consiste em substituir as derivadas explícitas por aproximações escritas na forma de equações algébricas, de forma a evitar o problema de resolver uma equação diferencial (RUGGIERO, 2008).

As diferenças podem ser consideradas atrasada, adiantada e centrada, em função dos pontos utilizados para serem calculadas, isto é, se a temperatura é calculada utilizando a diferença entre os dois pontos anteriores ao que se deseja obter a temperatura, é dita atrasada, se utilizar os dois pontos sucessores, é dita avançada e se utilizar os pontos anterior e posterior, é dita centrada.

Equações para a derivada de **primeira Ordem**:

Diferenças atrasadas:

Direção x:

$$T'(x_i) \simeq \frac{T_i - T_{i-1}}{dx} \quad (3)$$

Direção z:

$$T'(z_i) \simeq \frac{T_i - T_{i-nx}}{dz} \quad (4)$$

Diferenças avançadas:

Direção x:

$$T'(x_i) \simeq \frac{T_{i+1} - T_i}{dx} \quad (5)$$

Direção z:

$$T'(z_i) \simeq \frac{T_{i+1} - T_i}{dz} \quad (6)$$

Equações para as diferenças centradas para a derivada de **segunda ordem**:

Direção x:

$$T''(x_i) \simeq \frac{T_{i+1} - 2T_i + T_{i-1}}{dx^2} \quad (7)$$

Direção Z:

$$T''(z_i) \simeq \frac{T_{i+nx} - 2T_i + T_{i-nx}}{dz^2} \quad (8)$$

em que T_i é a temperatura no ponto de interesse, T_{i+1} é a temperatura do ponto à direita, T_{i-1} a temperatura do ponto à esquerda, T_{i-nx} a temperatura no ponto acima, T_{i+nx} a temperatura no ponto abaixo e dx e dz são os incrementos da largura e altura da seção calculados em função da largura e altura efetivas Lx e Lz , e do número de pontos nx e nz , nas direções x e z , respectivamente.

$$dx = \frac{Lx}{nx} \quad (9)$$

$$dz = \frac{Lz}{nz} \quad (10)$$

A fórmula que utiliza diferenças centradas é mais precisa pois o erro é na ordem de h^2 , e como h é sempre menor que 1, esta fórmula acaba sendo mais precisa que as demais (RUGGIERO, 2008). Entretanto, nas extremidades não é possível utilizar diferenças centradas, pois faltará o ponto anterior ou posterior, nestes casos, podem ser usadas as diferenças avançadas ou atrasadas.

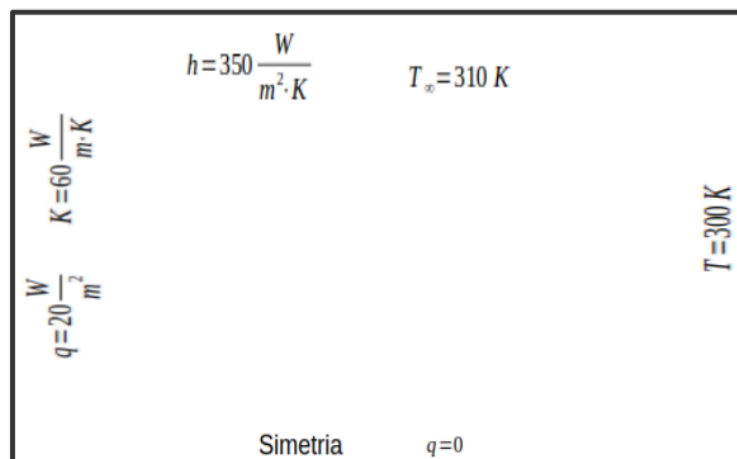
Condição de contorno

Primeiramente foi introduzida a equação de governo, a EDP que representa a distribuição espacial de calor na chapa, posteriormente foi introduzido o método das diferenças finitas como alternativa para a solução do problema, entretanto, o problema físico trata-se de uma chapa com diferentes condições de contorno, sendo ainda necessária a discretização do espaço bidimensional e além disso, a necessidade de vincular o espaço físico com a temperatura, de forma que seja possível gerar gráficos que ilustram a distribuição de temperatura por meio de intensidade de cores.

A chapa sendo simétrica e as condições de contorno também sendo aplicadas simetricamente, permite a modelagem apenas de uma das extremidades até o eixo de simetria, onde o fluxo de calor é nulo, ou seja, $q = 0$. Esta técnica de modelagem pode reduzir pela metade o esforço computacional e consequentemente o tempo de simulação, já que cada ponto extra do domínio significa uma equação a mais no sistema linear a ser resolvido.

A chapa a ser modelada recebe um fluxo de calor por condução com $q = 20 \frac{W}{m^2}$ na extremidade esquerda e por convecção na extremidade superior com $h = 350 \frac{W}{m^2 \cdot K}$ e $T_\infty = 310 K$, enquanto que a extremidade direita se mantém a uma temperatura constante de $T_{direita} = 300 K$, como ilustrado na Fig. 2.

Figura 2 - Condições de contorno nas extremidades.



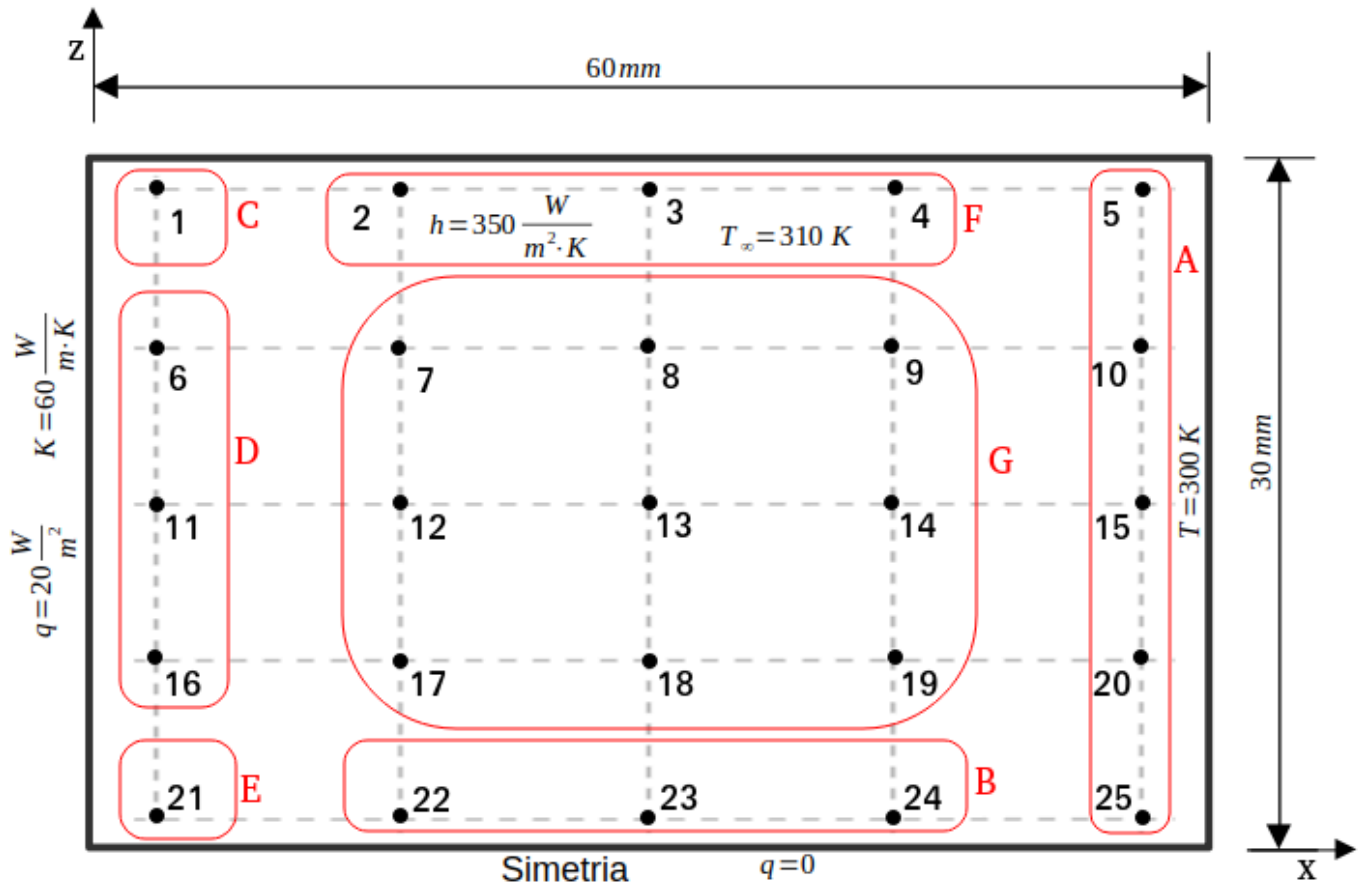
Discretização do domínio

Sabendo que haverá uma equação para cada ponto da malha, e muitos destes pontos tendo características comuns, é possível criar através de programação, rotinas que possam gerar automaticamente um sistema de equações de dimensão $nx^2 \times nx^2$ qualquer. Para ilustrar o procedimento de discretização do domínio, utilizaremos como exemplo um valor de $nx = 5$, correspondente a 25 pontos e 25 equações.

É importante destacar que o domínio deve ser discretizado de forma que as grandezas físicas essenciais sejam priorizadas com relação a suas derivadas. No problema a ser resolvido, a condição de contorno da extremidade direita é uma grandeza física essencial, temperatura. Por isso foi a primeira a ser definida, sendo precedente em toda a extremidade direita, ilustrada na Fig. 3 pela letra **A**. Posteriormente foram

definidas as demais condições de contorno na sequência **B**, **C**, **D**, **E**, **F** e por fim **G**, conforme também ilustrado na Fig 3.

Figura 3 - Discretização do domínio.



A temperatura nas bordas sofrerá influência das condições de contorno, como ilustrado na Fig. 3, enquanto que nos demais pontos no interior da malha sofrerá influência apenas de sua vizinhança, partindo do princípio de que não há geração de calor no interior da malha.

Retomando a equação de governo, Equação (2) e a escrevendo em função das derivadas algébricas de segunda ordem apresentadas na introdução do método das diferenças finitas, Equações (7) e (8), chegamos a seguinte forma da equação de governo:

$$\frac{1}{dz^2} \cdot T_{i-nx} + \frac{1}{dx^2} \cdot T_{i+1} - \left(\frac{2}{dx^2} + \frac{2}{dz^2} \right) \cdot T_i + \frac{1}{dz^2} \cdot T_{i+nx} + \frac{1}{dx^2} \cdot T_{i-1} = \frac{g}{k} \quad (11)$$

em que g é a geração de cada condição de contorno, ou zero para os pontos do interior da malha. É importante destacar que a posição de armazenamento em cada rotina deve ser feita com muito cuidado, pois a disposição física da matriz deve ser correlacionada com a da geometria da chapa. A Equação (11) para cada ponto da malha ficará da seguinte forma:

Rotina A:

A rotina inicia em $i = nx$, tem incremento de $i \cdot nx$ a cada iteração e finaliza quando $i \cdot nx = nx^2$, garantindo que a rotina grave valores apenas na última coluna da matriz.

Como já mencionado, a extremidade direita receberá os valores da própria temperatura, portanto as equações serão:

$$T_i = T_{direita} \quad (12)$$

Rotina B:

A rotina B, preencherá todas as posições da última linha, com exceção das extremidades, e as equações serão da forma:

$$\frac{1}{dz^2} \cdot T_{i-nx} + \frac{1}{dx^2} \cdot T_{i+1} - \left(\frac{2}{dx^2} + \frac{1}{dz^2} \right) \cdot T_i + \frac{1}{dx^2} \cdot T_{i-1} = -\frac{q}{k} = 0 \quad (13)$$

Vértice C:

$$\frac{1}{dx^2} \cdot T_{i+1} - \left(\frac{1}{dx^2} + \frac{1}{dz^2} \right) \cdot T_i + \frac{1}{dz^2} \cdot T_{i+nx} = -\frac{q}{k} \quad (14)$$

Rotina D:

$$\frac{1}{dz^2} \cdot T_{i-nx} + \frac{1}{dx^2} \cdot T_{i+1} - \left(\frac{1}{dx^2} + \frac{2}{dz^2} \right) \cdot T_i + \frac{1}{dz^2} \cdot T_{i+nx} = -\frac{q}{k} \quad (15)$$

Vértice E:

$$\frac{1}{dz^2} \cdot T_{i-nx} + \frac{1}{dx^2} \cdot T_{i+1} - \left(\frac{1}{dx^2} + \frac{1}{dz^2} \right) \cdot T_i = -\frac{q}{k} \quad (16)$$

Rotina F:

$$\frac{1}{dx^2} \cdot T_{i+1} - \left(\frac{2}{dx^2} + \frac{1}{dz^2} + h \cdot dz + 1 \right) \cdot T_i + \frac{1}{dz^2} \cdot T_{i+nx} + \frac{1}{dx^2} \cdot T_{i-1} = h \cdot dz \cdot T_{inf}$$

Rotina G:

$$\frac{1}{dz^2} \cdot T_{i-nx} + \frac{1}{dx^2} \cdot T_{i+1} - \left(\frac{2}{dx^2} + \frac{2}{dz^2} \right) \cdot T_i + \frac{1}{dz^2} \cdot T_{i+nx} + \frac{1}{dx^2} \cdot T_{i-1} = 0 \quad (17)$$

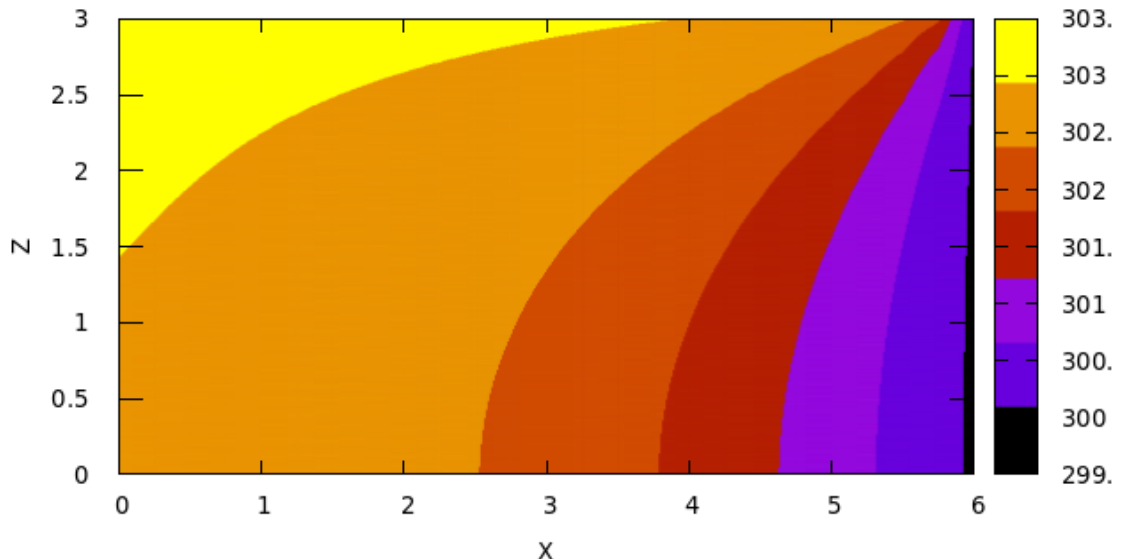
Sistema linear

As rotinas resultam em um sistema linear com nx^2 equações e incógnitas, que são armazenadas na forma matricial. O método de solução do sistema linear utilizado foi a biblioteca da fgs1.

Resultados e discussões

Após a solução do sistema, os dados são salvos em um arquivo txt e na sequência são lidos pelo Gnuplot para gerar os gráficos. Foram realizadas diversas simulações com uma certa variação da densidade da malha, o valor de $nx = 25$, foi o que mais se aproximou dos resultados esperados, como ilustrado na Fig. 4

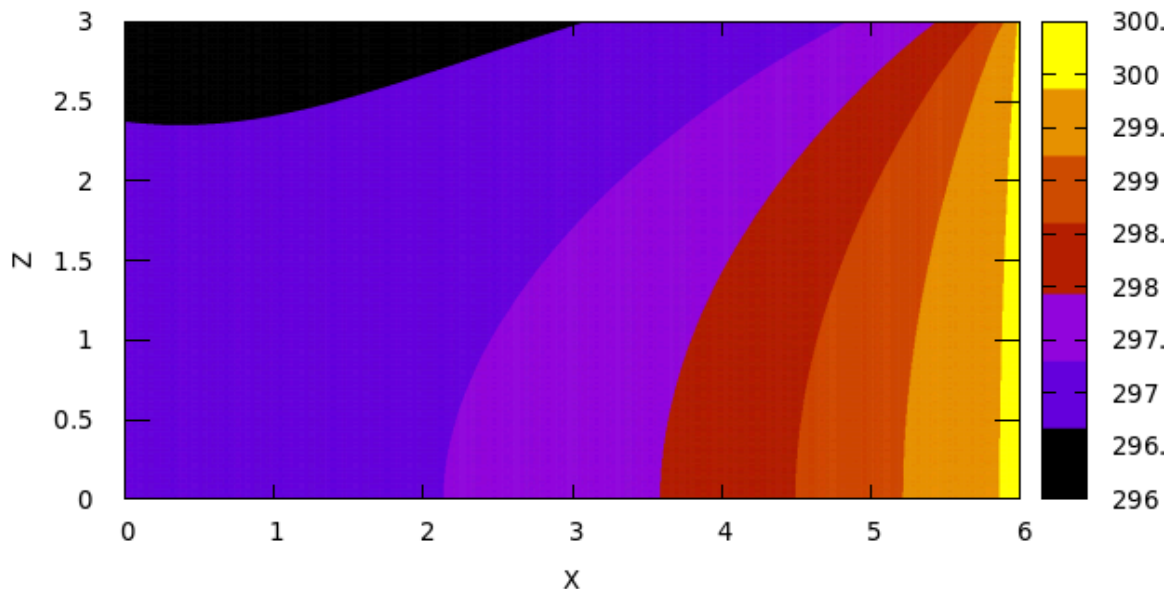
Figura 4 - Simulação para $nx = 25$.



O resultado apresentou um intervalo de temperatura condizente com o esperado, entretanto, o gradiente de cores teve uma pequena variação, que pode ser justificada pelas diferentes formas utilizadas para resolver o sistema linear e pelo refinamento da malha.

Também foram realizadas simulações para valores de $nx = 50, 75$ e 80 . O gráfico referente a simulação para $nx = 50$ é apresentado na Fig. 5.

Figura 5 - Simulação da distribuição de calor na chapa com $nx = 50$.



Para valores de $nx > 80$, como o tempo de simulação estava crescendo exponencialmente, apresentado na Tab. 1, não foi possível obter resultados pois a máquina antiga não correspondia, entretanto, utilizar malha muito refinada, também pode significar perda de precisão, como observado no problema da barra resolvido em sala de aula e também observado na Fig. 5, em que o intervalo de temperatura mudou de [299:303]K para [296:300]K com relação a Fig. 4.

Tabela 1 - Tempo de simulação para diferentes densidades de malha.

nx	25	50	75	80
Tempo [s]	0,52	33,35	359	539,5

Conclusões

Problemas de diferenças finitas são muito particulares, pois cada problema tem suas características específicas o que demanda uma programação, de certa forma personalizada, ou seja a forma como é montado o sistema é particular de cada problema. Neste problema foi possível entender as vantagens de se utilizar diferenças finitas, principalmente em função de dispensar a solução de EDP's na forma analítica. Além disso, foi observado que refinar a malha apesar de aumentar o esforço computacional, nem sempre significa ganho de precisão do resultado. O que pode melhorar de fato o resultado é melhorar a precisão da solução do sistema linear.

De modo geral, foi possível encontrar um resultado satisfatório para a distribuição de calor em uma chapa similar a uma aleta de um dissipador térmico, com as extremidades sob diferentes condições de contorno.

Referências

RUGGIERO, Marcia A. Gomes; LOPES, Vera Lucia da Rocha. *Cálculo numérico: aspectos teóricos e computacionais*. São Paulo: Pearson Makron Books, 2008.

INCROPERA, Frank P. et al. *Fundamentos de Transferência de Calor e Massa*. 6ª edição. Rio de Janeiro: LTC, 2008.

CÓDIGOS-FONTE

programa.f90

```
!gfortran -I/usr/local/include/fgsl programa.f90 -lfgsl -lm ; ./a.out ; gnuplot plot.plt
program diferencas_finitas
  use fgsl
  implicit none
  integer(fgsl_size_t) :: n,i,j,nx,ctd
  integer(fgsl_int) :: status, signum
  type(fgsl_matrix) :: M_a
  type(fgsl_vector) :: V_b, V_x
  real(fgsl_double), dimension (:), allocatable :: Vet_T,Vet_B
  real(fgsl_double), dimension (:,:), allocatable :: Mat_T
  type(fgsl_permutation) :: p
  real(8) :: dx,dz,Lx,Lz,T_direita,T_inf,k,q,h,t0,t1,t2
  open(2,file="dados.txt",status='unknown')

  nx=25
  n=nx**2
  allocate(Mat_T(n,n),Vet_T(n),Vet_B(n))

  !      DEFININDO PARÂMETROS
  Lx=6
  Lz=3
  dx=Lx/(nx-1)
  dz=Lz/(nx-1)
  k=60
  q=20
  h=350
  T_direita=300
```

T_inf=310

call cpu_time(t0)

! DISCRETIZANDO O DOMÍNIO - MONTANDO A MATRIZ A E VETOR B

! ROTINA A (Lateral direita)

do i=1,nx

!print*,i=i,i*nx

Mat_T(i*nx,i*nx) = 1

Vet_B(i*nx) = T_direita

end do

! ROTINA B (Linha inferior exceto as extremidades)

do i=nx**2-nx+2,nx*nx-1

!print*,i=i,i

Mat_T(i,i) = -2/dx**2 - 1/dz**2

Mat_T(i,i-1) = 1/dx**2

Mat_T(i,i+1) = 1/dx**2

Mat_T(i,i-nx) = 1/dz**2

Vet_B(i) = 0

end do

! VÉRTICE C (vertice superior esquerdo)

i=1

!print*,i=i,i

Mat_T(i,i) = -1/dx**2 - 1/dz**2 !-(h*dz+1)

Mat_T(i,i+1) = 1/dx**2

Mat_T(i,i+nx) = 1/dz**2

Vet_B(i) = - q/k !- h*dz*T_inf

! ROTINA D (Lateral esquerda, exceto os vértices)

do i=2,nx-1

!print*,i=i,(i-1)*nx+1

Mat_T((i-1)*nx+1,(i-1)*nx+1) = -1/dx**2 - 2/dz**2

Mat_T((i-1)*nx+1,(i-1)*nx+2) = 1/dx**2

Mat_T((i-1)*nx+1,(i-1)*nx+1-nx) = 1/dz**2

Mat_T((i-1)*nx+1,(i-1)*nx+1+nx) = 1/dz**2

Vet_B((i-1)*nx+1) = -q/k

end do

! VÉRTICE E (Vértice inferior esquerdo)

i=nx**2-nx+1

!print*, 'i=', i

Mat_T(i,i) = -1/dx**2 - 1/dz**2

Mat_T(i,i-nx) = 1/dz**2

Mat_T(i,i+1) = 1/dx**2

Vet_B(i) = -q/k

! ROTINA F (Linha superior, exceto os vértices)

do i=2,nx-1

!print*, 'i=', i

Mat_T(i,i) = -2/dx**2 - 1/dz**2 -(h*dz+1)

Mat_T(i,i-1) = 1/dx**2

Mat_T(i,i+1) = 1/dx**2

Mat_T(i,i+nx) = 1/dz**2

Vet_B(i) = - h*dz*T_inf

end do

! ROTINA G (Interior da malha Toda a malha, exceto o contorno)

i=nx+1

do ctd=1,nx-2

do i = ctd*nx+2,(ctd+1)*nx-1

!print*, 'i=', i

Mat_T(i,i) = -2/dx**2 -2/dz**2

Mat_T(i,i-1) = 1/dx**2

Mat_T(i,i+1) = 1/dx**2

Mat_T(i,i+nx) = 1/dz**2

Mat_T(i,i-nx) = 1/dz**2

Vet_B(i)=0

end do

end do

! IMPRIMINDO A MATRIZ

!write(*,*)'MATRIZ'

!do i=1 ,n

! write(*,*)Mat_T(i,:), Vet_B(i)

!end do

call cpu_time(t1)

! RESOLVENDO O SISTEMA

```
Mat_T = transpose(Mat_T)
M_a = fgsl_matrix_init(type=1.0_fgsl_double)
V_b = fgsl_vector_init(type=1.0_fgsl_double)
V_x = fgsl_vector_init(type=1.0_fgsl_double)
p = fgsl_permutation_alloc(n)
status = fgsl_matrix_align(Mat_T, n, n, n, M_a)
status = fgsl_vector_align(Vet_B, n, V_b, n, 0_fgsl_size_t, 1_fgsl_size_t)
status = fgsl_vector_align(Vet_T, n, V_x, n, 0_fgsl_size_t, 1_fgsl_size_t)
status = fgsl_linalg_LU_decomp (M_a, p, signum)
status = fgsl_linalg_LU_solve (M_a, p, V_b, V_x)
call fgsl_matrix_free(M_a)
call fgsl_vector_free(V_b)
call fgsl_vector_free(V_x)
call fgsl_permutation_free(p)
call cpu_time(t2)
print*, 'Tempo para definir a matriz', t1-t0
print*, 'Tempo para resolver o sistema', t2-t1
!print*, 'VETOR SOLUÇÃO'
!print*, Vet_T
```

!VINCULANDO O VETOR SOLUÇÃO COM A POSIÇÃO DOS EIXOS CARTEZIANOS

```
ctd = 0
do j=1,nx
  write(2,*) !Linha em branco para o Gnuplot
  do i=1,nx
    ctd = ctd+1
    write(2,*)(i-1)*dx,(nx-j)*dz,Vet_T(ctd)
    !write(*,*)(j-1)*dx,(nx-i)*dz,Vet_T(ctd)
  end do
end do
close(2)
deallocate(Mat_T,Vet_T,Vet_B)
end program diferencas_finitas
```