UFRJ

AUGUSTO GUIMARÃES RODRIGUES DE LIMA

DRE: 119025393

2)

(a) Substituindo a aproximação

$y''(x_k) \approx (y(x_{k+1}) - 2y(x_k) + y(x_{i-1}))/h^2$

na equação e como h = 5/3 (temos 6 intervalos entre 0 e 10, logo h é igual a (10 – 0)/6), obteremos:

$(100/9)\, y''(x_k) = y(x_{k+1}) - 2y(x_k) + y(x_{i-1})$

$(100/9)\, 4x = y(x_{k+1}) - 2y(x_k) + y(x_{i-1})$

Sabemos que y(0) = 5 e y(10) = 20, então montamos o sistema abaixo:

$(100/9)(5/3) = y(x_2) - 2y(x_1) + 5$

$(100/9)(10/3) = y(x_3) - 2y(x_2) + y(x_1)$

$(100/9)(5) = y(x_4) - 2y(x_3) + y(x_2)$

$(100/9)(20/3) = y(x_5) - 2y(x_4) + y(x_3)$

$(100/9)(25/3) = 20 - 2y(x_5) + y(x_4)$

$365/27 = y(x_2) - 2y(x_1)$

$1000/27 = y(x_3) - 2y(x_2) + y(x_1)$

$500/9 = y(x_4) - 2y(x_3) + y(x_2)$

$2000/27 = y(x_5) - 2y(x_4) + y(x_3)$

$1960/27 = -2y(x_5) + y(x_4)$

Questão 3:

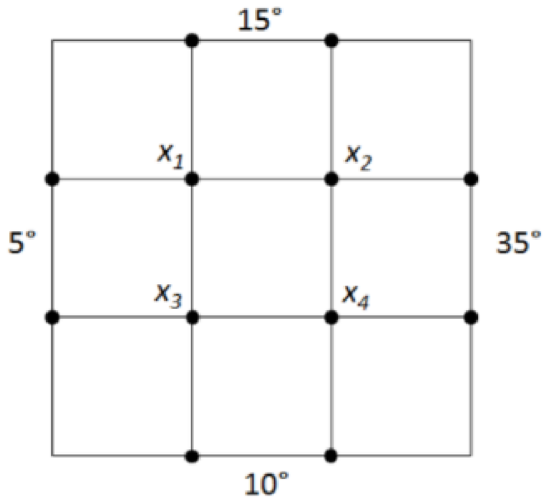
Questão realizada pelos alunos

Augusto Guimarães - DRE:119025393,

Luiz Rodrigo Lace - DRE: 118049873 e

Lívia Barbosa Fonseca - DRE: 118039721.

(a) Nesse exercício queremos descobrir a temperatura em diferentes lugares no interior de um lago. Sabendo que quando o calor está em equilíbrio, a temperatura em cada vértice no interior do lago é média das temperaturas dos 4 vértices vizinhos. Vamos modelar o problema dado como um sistema linear do tipo Ax = b.



Com base nos dados informados pelo problema, vamos obter o seguinte conjunto de equações:

$$x_1 = (x_2+x_3+5+15)/4$$

$$x_2 = (x_1+x_4+15+35)/4$$

$$x_3 = (x_1+x_4+5+10)/4$$

$$x_4 = (x_2+x_3+35+10)/4$$

Arrumando essas equações obtemos:

$4x_1 = x_2+x_3+5+15 \Rightarrow -4x_1 + x_2+x_3 = -20$

$4x_2 = x_1+x_4+15+35 \Rightarrow -4x_2 + x_1+x_4 = -50$

$4x_3 = x_1+x_4+5+10 \Rightarrow -4x_3 + x_1+x_4 = -15$

$4x_4 = x_2+x_3+35+10 \Rightarrow -4x_4 + x_2+x_3 = -45$

Com essas equações, podemos montar as seguintes matrizes que atendem o formato o formato Ax = B

$$\begin{bmatrix} -4 & 1 & 1 & 0 \\ 1 & -4 & 0 & 1 \\ 1 & 0 & -4 & 1 \\ 0 & 1 & 1 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} -20 \\ -50 \\ -15 \\ -45 \end{bmatrix}$$

(d) Nessa questão queremos descobrir o maior número de nós que conseguimos discretizar e rodar na nossa máquina em menos de 2 minutos usando as funções criadas para a resolução de um sistema via método LU.

Para descobrirmos o tempo que a discretização em questão está demorando, vamos utilizar a seguinte biblioteca:

```
using Dates
#Importando a biblioteca
import Dates
```

Para calcularmos o tempo que a função está demorando, vamos marcar o tempo do início do processo e quando o processo foi finalizado e em seguida realizar um tempo menos o outro, essa diferença será o tempo que a função está demorando para ser rodada.

Vamos começar chutando n = 2025 (lembrando que n é um quadrado perfeito):

```
A = criação_da_matriz(2025)
b = matriz_resultado(2025)

#Inicio do processo
rightnow = Dates.Time(Dates.now())

#Chamando a função que calcula o sistema
sistema_denso(A,b)

#Final do processo
```

```
rightnow2 = Dates.Time(Dates.now())

#Duração
dif = (rightnow2 - rightnow)
```

Obtemos:

```
A = criação_da_matriz(2025)
b = matriz_resultado(2025)

rightnow = Dates.Time(Dates.now())
sistema_denso(A,b)
rightnow2 = Dates.Time(Dates.now())

dif = (rightnow2 - rightnow)
```
✓ 1m 55.9s

```
115929000000 nanoseconds
```

Sabendo que 1 minuto vale 6e+10 nanosegundo, temos:

```
[28] 115929000000/6e+10
```
✓ 0.5s

```
... 1.93215
```

Encontramos que com n = 2025, nossa função demora 1,93215 minutos.

Agora vamos tentar com n = 2116:

```
A = criação_da_matriz(2116)
b = matriz_resultado(2116)

rightnow = Dates.Time(Dates.now())
sistema_denso(A,b)
rightnow2 = Dates.Time(Dates.now())

dif = (rightnow2 - rightnow)
```

✓ 2m 19.8s

```
139792000000 nanoseconds
```

Transformando em minuto temos:

```
139792000000/6e+10
```

✓ 0.2s

```
2.3298666666666668
```

Podemos observar que com n = 2116, o tempo estipulado foi maior do que 2 minutos. Dessa forma, o maior numero que nós que conseguimos discretizar em menos de 2 minutos é 2025.