MAP 2320 – MÉTODOS NUMÉRICOS EM EQUAÇÕES DIFERENCIAIS II

2º Semestre - 2019

Prof. Dr. Luis Carlos de Castro Santos

lsantos@ime.usp.br

3. Approximate the solutions to the following elliptic partial differential equations, using Algorithm 12.1:

a. $$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0, \qquad 0 < x < 1, \quad 0 < y < 1;$$

$$u(x,0) = 0, \quad u(x,1) = x, \qquad 0 \le x \le 1;$$

$$u(1,y) = y, \qquad 0 \le y \le 1.$$

Use $h = k = 0.2$, and compare the results to the actual solution $u(x,y) = xy$.

Vamos repetir o problema com h = k = 1/3

Vamos mudar uma condição de contorno para du/dx=y em x=0

O que deve acontecer ?

Forward Finite-Difference

First Derivative	Error
$f'(x_i) = \dfrac{f(x_{i+1}) - f(x_i)}{h}$	$O(h)$
$f'(x_i) = \dfrac{-f(x_{i+2}) + 4f(x_{i+1}) - 3f(x_i)}{2h}$	$O(h^2)$
Second Derivative	
$f''(x_i) = \dfrac{f(x_{i+2}) - 2f(x_{i+1}) + f(x_i)}{h^2}$	$O(h)$
$f''(x_i) = \dfrac{-f(x_{i+3}) + 4f(x_{i+2}) - 5f(x_{i+1}) + 2f(x_i)}{h^2}$	$O(h^2)$
Third Derivative	
$f'''(x_i) = \dfrac{f(x_{i+3}) - 3f(x_{i+2}) + 3f(x_{i+1}) - f(x_i)}{h^3}$	$O(h)$
$f'''(x_i) = \dfrac{-3f(x_{i+4}) + 14f(x_{i+3}) - 24f(x_{i+2}) + 18f(x_{i+1}) - 5f(x_i)}{2h^3}$	$O(h^2)$
Fourth Derivative	
$f''''(x_i) = \dfrac{f(x_{i+4}) - 4f(x_{i+3}) + 6f(x_{i+2}) - 4f(x_{i+1}) + f(x_i)}{h^4}$	$O(h)$
$f''''(x_i) = \dfrac{-2f(x_{i+5}) + 11f(x_{i+4}) - 24f(x_{i+3}) + 26f(x_{i+2}) - 14f(x_{i+1}) + 3f(x_i)}{h^4}$	$O(h^2)$

Backward Finite-Divided Difference Formulas

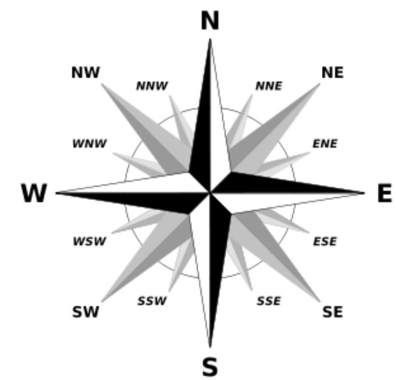
First Derivative	**Error**
$f'(x_i) = \dfrac{f(x_i) - f(x_{i-1})}{h}$	$O(h)$
$f'(x_i) = \dfrac{3f(x_i) - 4f(x_{i-1}) + f(x_{i-2})}{2h}$	$O(h^2)$
Second Derivative	
$f''(x_i) = \dfrac{f(x_i) - 2f(x_{i-1}) + f(x_{i-2})}{h^2}$	$O(h)$
$f''(x_i) = \dfrac{2f(x_i) - 5f(x_{i-1}) + 4f(x_{i-2}) - f(x_{i-3})}{h^2}$	$O(h^2)$
Third Derivative	
$f'''(x_i) = \dfrac{f(x_i) - 3f(x_{i-1}) + 3f(x_{i-2}) - f(x_{i-3})}{h^3}$	$O(h)$
$f'''(x_i) = \dfrac{5f(x_i) - 18f(x_{i-1}) + 24f(x_{i-2}) - 14f(x_{i-3}) + 3f(x_{i-4})}{2h^3}$	$O(h^2)$
Fourth Derivative	
$f''''(x_i) = \dfrac{f(x_i) - 4f(x_{i-1}) + 6f(x_{i-2}) - 4f(x_{i-3}) + f(x_{i-4})}{h^4}$	$O(h)$
$f''''(x_i) = \dfrac{3f(x_i) - 14f(x_{i-1}) + 26f(x_{i-2}) - 24f(x_{i-3}) + 11f(x_{i-4}) - 2f(x_{i-5})}{h^4}$	$O(h^2)$


12
1
2
3
4
5
6
7
8
9
10
11

			iterações							
n	0	1	2	3	4	8	9	10	sol	erro_2
u_1_1	0	-0,01852	-0,00116	0,040196	0,068479	0,105983	0,108101611	0,109345	0,111111	3,11799E-06
u_2_1	0	0,078704	0,1739	0,195114	0,206251	0,220318	0,22110518	0,221567	0,222222	4,29513E-07
u_1_2	0	0,041667	0,120467	0,164322	0,188722	0,218291	0,219916447	0,22087	0,222222	1,82968E-06
u_2_2	0	0,363426	0,406925	0,423192	0,432077	0,442986	0,44358874	0,443942	0,444444	2,52011E-07
									norma_2	2,37E-03

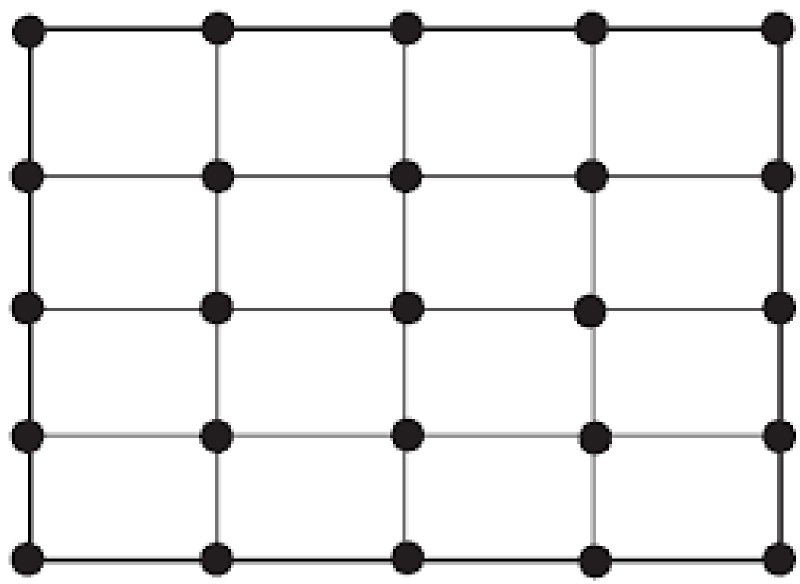
Problema de Poisson com condições mistas

$$u_{xx} + u_{yy} = f(x, y) \qquad a \le x \le b$$
$$c \le y \le d$$



$$u(x, d) = g_N(x)$$

$$u(a, y) = g_W(y)$$



$$\frac{\partial u}{\partial x}(b, y) = g_E(x)$$

$$u(x, c) = g_S(x)$$

A discretização da equação resulta na mesma expressão para os pontos internos:

$$u_{i,j} = \frac{1}{\mu}\left\{u_{i+1,j} + u_{i-1,j} + \lambda\left(u_{i,j+1} + u_{i,j-1}\right) - h^2 f_{i,j}\right\}$$

A alteração ocorre na necessidade de aproximar a derivada no contorno por uma expressão algébrica.

A fórmula de diferenças finitas regressiva de 2ª ordem pode ser utilizada.

$$\left(\frac{\partial u}{\partial x}\right)_{i,j} = \frac{3u_{i,j} - 4u_{i-1,j} + u_{i-2,j}}{2h} = g_E(y_j)$$

Logo

$$u_{i,j} = \frac{1}{3}\left(2hg_E(y_j) + 4u_{i-1,j} - u_{i-2,j}\right)$$

No problema exemplo essa expressão deve ser aplicada para o contorno i=m

MAP2320

```
INPUT a,b,c,d -> extremos do domínio
      m,n     -> no. de intervalos nas direções x e y respectivamente
      tol     -> tolerância do critério de parada do método iterativo
      itmax   -> no. máximo de iterações (para evitar loop infinito)

h=(b-a)/m, k=(d-c)/n, lambda = (h*h)/(k*k), mu =2*(1+lambda)


for i = 0,m          -> definição as coordenadas da malha
 x(i) = a + i*h
endfor

for j = 0,n
 y(j) = c + j*k
endfor

for i = 0,m
for j = 0,n
 u(i,j) = 0.                   -> inicialização da solução
 f(i,j) = f(x(i),y(j))         -> definição do termo forçante
endfor
endfor

for i = 0,m          -> aplicação das condições de contorno de Dirichlet
 u(i,0) = gs(x(i))
 u(i,n) = gn(x(i))
endfor

for j = 1,n-1
 u(0,j) = gw(y(j))
Endfor

for j = 1,n-1-> aplicação da condições de contorno de Neumann para a condição inicial
 u(m,j) = (2*h*ge(y(j))+4*u(m-1,j)-u(m-2,j))/3
endfor
```

```
norm = 1e+02          -> valor inicial arbitrário da norma
l = 1                 -> início do loop do método de Gauss-Seidel

"TAG"

for i = 1,m-1                    -> laço para os pontos interiores
for j = 1,n-1
 u(i,j) = (u(i+1,j)+u(i-1,j)+lambda*(u(i,j+1)+u(i,j-1))-h*h*f(i,j))/(mu)
endfor
Endfor

for j = 1,n-1                   -> aplicação da condições de contorno de Neumann
 u(m,j) = (2*h*ge(y(j))+4*u(m-1,j)-u(m-2,j))/3
endfor

z = max(u)             -> norma do máximo da linguagem escolhida

if (abs(z-norm).le. tol) then
 write(u)              -> comando de impressão/plotagem/saída da linguagem escolhida
 write("erro=", abs(z-norm))
 STOP
else
 norm = z
endif

if (l.ge.itmax) then
 write(u)              -> comando de impressão/plotagem/saída da linguagem escolhida
 write("erro=", abs(z-norm))
 write ("WARNING - no. max. Iterações atingido")
 STOP
endif

l = l + 1

Goto "TAG"
```

Example 2

Use the Poisson finite-difference method with $n = 6$, $m = 5$, and a tolerance of 10^{-10} to approximate the solution to

$$\frac{\partial^2 u}{\partial x^2}(x, y) + \frac{\partial^2 u}{\partial y^2}(x, y) = xe^y, \quad 0 < x < 2, \quad 0 < y < 1,$$

with the boundary conditions

$$u(0, y) = 0, \quad u(2, y) = 2e^y, \quad 0 \le y \le 1,$$
$$u(x, 0) = x, \quad u(x, 1) = ex, \quad 0 \le x \le 2,$$

and compare the results with the exact solution $u(x, y) = xe^y$.

Vamos realizar o problema com n = 3, m = 3


12
1
2
3
4
5
6
7
8
9
10
11

...

	iterações								sol	erro_2
n	0	1	2	3	8	9	10		sol	erro_2
u_1_1	0	0,25633	0,663263	0,909938	1,01117	1,011250145	1,01127		0,930408	0,006538645
u_2_1	0	0,817403	1,74558	1,934142	1,991208	1,991248256	1,991258		1,860817	0,017015039
u_1_2	0	0,812982	1,197626	1,33727	1,389498	1,389538368	1,389548		1,298489	0,008291746
u_2_2	0	2,218709	2,628445	2,717834	2,745883	2,745903139	2,745908		2,596979	0,022179974
	norma		1,159	0,352	0,0004	0,0001	2,5E-05		norma_2	0,232433654

Norma_2 iterações sucessivas

Norma_2 erro sol. exata


</tbody></table>
</tbody></table>

2o Trabalho Computacional

Entrega 24/11/2019 11:59

A entrega consiste num relatório **substanciado** contendo os resultados, preferencialmente na forma de gráficos e/ou tabelas, e do código fonte com instruções sobre como obter os resultados. Os arquivos devem estar numa pasta compactada com indicação pelo menos do nome do autor do trabalho, e devem ser enviados apenas por e-mail.

Considere a equação de Poisson,

$$u_{xx} + u_{yy} = f(x,y), \quad 0 < x < 1,\ 0 < y < 1$$

Proponha uma solução envolvendo senos, cossenos e exponenciais (no mínimos duas funções combinadas de forma livre) e deduza a forma de f(x,y).

Resolva o problema com condições de contorno de Dirichlet (dadas pela sua solução proposta) em duas faces do domínio, e de Neumann nas outras duas, usando como solução inicial u(x,y)=0.

a) Estabeleça uma sequência de valores de h (= k) cada vez menores e obtenha a convergência do método iterativo para uma mesma tolerância fixa, usando como critério convergência alguma norma da diferença de soluções consecutivas. Compare a norma do erro da sua solução convergida com a solução exata manufaturada.

b) Escolha um valor de h fixo, e repita o estudo para tolerâncias cada vez menores.

c) Comente os resultados dos estudos, comparando normas, número de iterações, tempos de CPU, etc... Use preferencialmente tabelas e gráficos. Esse item terá peso maior que os anteriores, logo dedique-se a análise dos resultados.

MAP 2320 – MÉTODOS NUMÉRICOS EM EQUAÇÕES DIFERENCIAIS II

2º Semestre - 2019

Roteiro do curso

- Introdução
- Séries de Fourier
- **Método de Diferenças Finitas**
- Equação do calor transiente (parabólica)
- **Equação de Poisson (elíptica)**
- Equação da onda (hiperbólica)