

Departamento de Matemática Pura e aplicada -  
 UFRGS  
 Lista de Exercícios para Cálculo Numérico  
 Autor: Prof. João Batista Carvalho

Matp06 - Métodos Numéricos para Equações Diferenciais  
 Notas de aula - 2003/1 (adaptadas para 2012/1)

Conteúdo programático da disciplina:

1. Propagação de erros numéricos;
2. Derivação e integração numérica;
3. Métodos de passo simples e múltiplo; explícitos e implícitos.
4. Métodos de discretização parcial; o Método das Linhas.
5. Revisão de pré-requisitos básicos.
6. Análise de estabilidade numérica: Fourier, Von Neumann, espectral.
7. Convergência e o teorema de Lax-Wendroff, Lax-Richtmeyer.
8. Solução numérica de equações parabólicas.
9. Solução numérica de equações hiperbólicas.
10. Solução numérica de equações elípticas.

# 1 Propagação de erros numéricos e estimativas

Em Análise Numérica duas fontes de erros numéricos se sobressaem: erros de arredondamento e erros de truncamento. Definições precisas dessas fontes de erros dependem do problema em questão .

**Exemplo 1.1 : Recursão escalar.**

Considere a avaliação numérica da seguinte recursão :

$$\begin{cases} x_{n+1} = 4x_n - 1 & , n > 1 \\ x_1 = 1/3 \end{cases}$$

A solução analítica é  $x_n = 1/3$  para todo  $n \geq 0$ .

A solução numérica em Gnu-Fortran 77 (g77), usando precisão dupla em ambiente IBM PC e Linux, nos dá a tabela seguinte.

$n$	$x_n$	$n$	$x_n$
1	0.333333333	21	0.333251953
2	0.333333333	22	0.333007812
3	0.333333333	23	0.33203125
4	0.333333333	24	0.328125
5	0.333333333	25	0.3125
6	0.333333333	26	0.25
7	0.333333333	27	0.
8	0.333333333	28	-1.
9	0.333333333	29	-5.
10	0.333333333	30	-21.
11	0.333333333	31	-85.
12	0.333333333	32	-341.
13	0.333333332	33	-1365.
14	0.333333328	34	-5461.
15	0.333333313	35	-21845.
16	0.333333254	36	-87381.
17	0.333333015	37	-349525.
18	0.333332062	38	-398101.
19	0.333328247	39	-5592405.
20	0.333312988	40	-22369621.

```

program exemplo1
double precision x
integer n,i
c
n = 40
x = 1.0D0/3
do i=1,n
    write(*,'(i3,x,f20.8)') i,x
    x = 4*x - 1.0D0
end do
end

# Script em Maple
printlevel:=0;
n:=40; x:=1.0/3;
for i from 1 to n do
    print(i,x); x:=4*x-1.0;
end do;
    
```

A explicação do fenômeno é simples: por não ter uma representação binária finita, o número real 1/3 não pode ser representado exatamente no computador:

$$\frac{1}{3} = \frac{1}{4-1} = \frac{1/4}{1-1/4} = \frac{1}{4} + \frac{1}{4^2} + \frac{1}{4^3} + \dots$$

$$\Rightarrow \frac{1}{3} = \frac{1}{2^2} + \frac{1}{2^4} + \frac{1}{2^6} + \dots = (0.01010101\dots)_2$$

e assim temos que  $x_1 = 1/3 + \epsilon$ , onde  $\epsilon$  é um número de magnitude muito pequena que depende da aritmética ponto flutuante utilizada pelo computador.

Dessa forma, existe uma **única** fonte de erro, e uma propagação desastrosa:

$$x_1 - 1/3 = \epsilon$$

$$x_2 - 1/3 = 4(x_1 + \epsilon) - 1 - 1/3 = 4\epsilon$$

$$x_3 - 1/3 = 4(x_2 + \epsilon) - 1 - 1/3 = 4^2\epsilon$$

e de uma forma generalizada temos  $x_n - 1/3 = 4^{n-1}\epsilon$ . Dessa forma o erro numérico é quadruplicado a cada iteração .

**Definição 1.1** Erros que ocorrem devido ao fato do computador trabalhar apenas com números com representação binária finita são chamados erros de arredondamento.

**Definição 1.2** Um número real  $a$  possui representação no computador a ser denotada por  $fl(a)$  ou  $\hat{a}$ .

**Exemplo 1.2 : Avaliação de funções transcendentess.**

Diferentemente das funções algébricas, funções transcendentess não podem ser avaliadas como um número finito de operações aritméticas (+, -, \*, ÷).

- Considere a avaliação numérica da raiz quadrada de um número positivo  $a$  dado. Essa tarefa, ao computador, normalmente é feita através da recursão :

$$x_{n+1} = \frac{x_n}{2} + \frac{a}{2x_n}, n > 1$$

onde o valor inicial  $x_0$  é adequadamente definido em termos de  $a$ .

Novamente, **erros de arredondamento** estão presentes em todos os estágios da computação . Entretanto, uma característica dessa recursão é a rápida convergência ao valor desejado, o que minimiza tremendamente a propagação dos erros de arredondamento.

Pode-se mostrar que  $fl(\sqrt{fl(a)}) = fl(\sqrt{a})$ , ou seja, a resposta dada pelo computador ainda é a melhor possível em aritmética finita.

- Considere a avaliação numérica da exponencial de um número real  $a$  dado.

Ao computador, a exponencial pode ser calculada através de sua **série ou expansão de Taylor**:

$$e^a = \sum_{i=0}^{\infty} \frac{a^i}{i!} = 1 + a + \frac{a^2}{2!} + \frac{a^3}{3!} + \dots +$$

Os termos parciais

$$s_n = \sum_{i=0}^n \frac{a^i}{i!} = 1 + a + \frac{a^2}{2} + \frac{a^3}{6} + \dots + \frac{a^n}{n!}$$

são então avaliados em IBM PC Gnu-Fortran77:

$n$	$s_n$	$ s_n - e^a $
1	2.609437912434101	0.23905620875E + 01
2	3.904583109424218	0.10954168905E + 01
3	4.599401703471827	0.40059829652E + 00
4	4.878968550352922	0.12103144964E + 00
5	4.968957646838940	0.31042353161E - 01
6	4.993096290770655	0.69037092293E - 02
7	4.998646240584719	0.13537594153E - 02
8	4.999762778040077	0.23722195992E - 03
9	4.999962444452445	0.37555547555E - 04
10	4.99994579521835	0.54204781641E - 05
11	4.99999281285380	0.71871461937E - 06
12	4.99999911885088	0.88114910923E - 07
13	4.99999989955171	0.10044828080E - 07
14	4.99999998930096	0.10699032771E - 08
15	4.9999999893068	0.10693135266E - 09
16	4.99999998933	0.10065726031E - 10
17	4.9999999999104	0.89528384706E - 12
18	4.9999999999924	0.75495165675E - 13
19	4.9999999999993	0.62172489379E - 14
20	4.9999999999998	0.88817841970E - 15
21	4.9999999999998	0.88817841970E - 15
22	4.9999999999998	0.88817841970E - 15
23	4.9999999999998	0.88817841970E - 15
24	4.9999999999998	0.88817841970E - 15
25	4.9999999999998	0.88817841970E - 15

```

program exemplo2
  double precision
  a,a_n,s_n,e_n
  integer nmax,n,i
  c
  a = log(5.0D0)
  nmax = 25
  a_n = 1.0D0
  s_n = 1.0D0
  do n=1,nmax
    a_n = a*a_n/n
    s_n = s_n + a_n
    e_n = abs(exp(a)-s_n)
    write(*, '(i3,x,f30.16,x,e20.12)')
  n,s_n,e_n
  end do
end

```

Além dos inevitáveis erros de arredondamento, ao calcular cada  $s_n$ , temos também um **erro de truncamento**, pelo fato de não estarmos considerando os termos finais (cauda) da série.

Nesse exemplo, os erros de arredondamento podem ser desprezados em comparação aos erros de truncamento (diferentes ordens de magnitude).

## 2 Derivação e integração numéricas

### Exemplo 2.1 : Avaliação numérica de derivadas.

Seja o problema de calcular numericamente  $f'(1)$  para uma dada função contínua  $f(x) : I \rightarrow \mathbb{R}$ ,  $1 \in I$ .

A estratégia é aproximar a derivada por diferentes **fórmulas de diferenças finitas**:

$$d_1(h) = \frac{f(1+h) - f(1)}{h}, \quad d_2(h) = \frac{f(1+h) - f(1-h)}{2h}$$

$$d_3(h) = \frac{f(1+2h) - f(1+h) + f(1)}{h}$$

e gostaríamos de saber se alguma das fórmulas acima deveria ser preferida às demais.

Lembramos a **Série de Taylor** para  $f(x)$  em  $x = 1$ :

$$f(1+h) = f(1) + hf'(1) + \frac{h^2 f''(1)}{2!} + \frac{h^3 f^{(3)}(1)}{3!} + \dots + \frac{h^n f^{(n)}(1)}{n!} + \dots$$

assumindo que  $f(x)$  seja analítica em alguma vizinhança de  $x = 1$ .

Temos então

$$d_1(h) = \frac{1}{h} \left[ f(1) + hf'(1) + \frac{h^2 f''(1)}{2!} + \frac{h^3 f^{(3)}(1)}{3!} + \dots - f(1) \right] =$$

$$f'(1) + \frac{hf''(1)}{2!} + \frac{h^2 f^{(3)}(1)}{3!} + \frac{h^3 f^{(4)}(1)}{4!} + \dots +$$

ao passo que

$$d_2(h) = \frac{1}{2h} \left[ f(1) + hf'(1) + \frac{h^2 f''(1)}{2!} + \frac{h^3 f^{(3)}(1)}{3!} + \dots + \frac{h^n f^{(n)}(1)}{n!} + \dots \right.$$

$$\left. - \left( f(1) - hf'(1) + \frac{h^2 f''(1)}{2!} - \frac{h^3 f^{(3)}(1)}{3!} + \dots + \frac{(-1)^n h^n f^{(n)}(1)}{n!} + \dots \right) \right] =$$

$$f'(1) + \frac{h^2 f^{(3)}(1)}{3!} + \frac{h^5 f^{(5)}(1)}{5!} + \dots + \frac{h^7 f^{(7)}(1)}{7!} + \dots$$

e

$$d_3(h) = \frac{1}{h} \left[ \left( f(1) + 2hf'(1) + \frac{(2h)^2 f''(1)}{2!} + \frac{(2h)^3 f^{(3)}(1)}{3!} + \frac{(2h)^4 f^{(4)}(1)}{4!} + \dots \right) \right.$$

$$\left. - \left( f(1) + hf'(1) + \frac{h^2 f''(1)}{2!} + \frac{h^3 f^{(3)}(1)}{3!} + \frac{h^4 f^{(4)}(1)}{4!} + \dots \right) + f(1) \right] =$$

$$\frac{f(1)}{h} + f'(1) + \frac{3hf''(1)}{2!} + \frac{7h^2 f^{(3)}(1)}{3!} + \frac{15h^3 f^{(4)}(1)}{4!} + \dots$$

As quantidades

$$e_1(h) = d_1(h) - f'(1)$$

$$e_2(h) = d_2(h) - f'(1)$$

$$e_3(h) = d_3(h) - f'(1)$$

são **erros de truncamento** pois são resultado da aproximação de um operador de dimensão infinita (a derivada) por uma diferença finita.

Vemos que apenas as duas primeiras aproximações podem ser úteis na aproximação de  $f'(1)$  se  $f(1) \neq 0$ .

**Definição 2.1** No contexto de nosso problema, uma aproximação é **consistente** se o correspondente erro de truncamento tende a zero ao  $h$  tender a zero.

As aproximações  $d_1$  e  $d_2$  são consistentes, enquanto que  $d_3$  é inconsistente.

Aém disso, assintoticamente,

$$d_1(h) = f'(1) + O(h) \text{ ao } h \rightarrow 0$$

$$d_2(h) = f'(1) + O(h^2) \text{ ao } h \rightarrow 0$$

e então  $d_1$  é um aproximação consistente de **primeira ordem**, ao passo que  $d_2$  é uma aproximação consistente de **segunda ordem**.

A tabela abaixo mostra a computação dessas duas aproximações para o caso em que  $f(x) = \exp(x)$ .

$h$	$e_1(h)$	$e_2(h)$	
0.1E+01	0.1952E+01	0.4762E+00	program exemplo3
0.1E+00	0.1406E+00	0.4533E-02	double precision
0.1E-01	0.1364E-01	0.4530E-04	h,d1,d2,de,one
0.1E-02	0.1360E-02	0.4530E-06	integer i
0.1E-03	0.1359E-03	0.4531E-08	one=1.0D0
0.1E-04	0.1359E-04	0.5859E-10	h = one
0.1E-05	0.1359E-05	-0.1635E-09	de= exp(one)
0.1E-06	0.1399E-06	0.5859E-10	do i=1,16
0.1E-07	-0.6603E-08	-0.6603E-08	d1=(exp(one+h)-
0.1E-08	0.2154E-06	-0.6603E-08	exp(one))/h
0.1E-09	0.1548E-05	-0.6727E-06	d2=(exp(one+h)-exp(one-
0.1E-10	0.3263E-04	0.1043E-04	h))/(2*h)
0.1E-11	0.4323E-03	0.2103E-03	write(*,'(e9.1,1x,e13.5,1x,e13.5)')
0.1E-12	-0.4559E-03	-0.4559E-03	. h,d1-de,d2-de
0.1E-13	-0.9338E-02	-0.9338E-02	h = h/10;
0.1E-14	0.3903E+00	0.1683E+00	end do
			end

**Definição 2.2** Chama-se de *Quadratura Numérica* o problema de avaliar, numericamente,

$$I = \int_a^b f(x)dx$$

para uma dada função  $f(x)$ , e para um intervalo  $[a, b]$ , para os quais sabemos que essa quantidade realmente existe, isto é, quando  $f$  é dita ser integrável no intervalo  $[a, b]$ .

Neste capítulo aplicaremos técnicas matemáticas na solução do problema da quadratura numérica de uma função dada, ou a partir de aproximação de uma tabela de dados.

**Métodos de Integração definida:** quanto à natureza, podem ser

- Analíticos: são os que usam o Teorema Fundamental do Cálculo ou fórmulas de recorrência a integrais mais simples;
- Numéricos: discutidos nesta disciplina;
- Semi-analíticos ou semi-numéricos:  $f$  é aproximada por funções  $\phi_i$  mais simples, como por exemplo, via Aproximação Funcional, e então as quantidades

$$\int_a^b \phi_i(x)dx,$$

que aproximam a quantidade que procuramos, são calculadas analiticamente.

**Métodos de Integração definida:** podem ser determinísticos ou probabilísticos

- determinísticos: a função  $f$  é avaliada em um conjunto suficientemente grande de pontos  $x_i$ ,  $i = 0, 1, \dots, n$ , que podem ter um espaçamento fixo (quadratura fixa), ou podem ter espaçamento variável (quadratura adaptativa). Construímos uma fórmula de integração a partir do polinômio interpolador correspondente;
- probabilísticos: os que valem-se da caracterização de integral definida de uma  $f$  em algum intervalo como "área com sinal" entre o gráfico dessa  $f$  e os eixos coordenados. Por causa da caracterização como área, métodos Monte-Carlo podem ser usados.

**Quadratura adaptativa:** usando uma *partição* de pontos  $x_0 < x_1 < x_2 < \dots < x_n$ , escrevemos

$$\int_a^b f(x)dx = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x)dx$$

onde

- fazemos os intervalos menores nas regiões onde  $f$  varia mais bruscamente (maior erro nas fórmulas de integração);
- fazemos os intervalos maiores nas regiões onde  $f$  é mais suave (menor erro nas fórmulas de integração).

Nas próximas seções, desenvolveremos estratégias consagradas de quadratura determinística:

- fórmulas de Newton-Côtes (quadratura newtoniana), cujo grande retrato é

$$\int_a^b f(x)dx = \sum_{i=0}^n w_i f(x_i); \quad (1)$$

- quadratura gaussiana, cujo grande retrato é

$$\int_a^b f(x)w(x)dx = \sum_{i=0}^n w_i f(x_i); \quad (2)$$

bem como suas aplicações a

- quadratura recursiva;
- quadratura de integrandos com singularidade.

## 2.1 Fórmulas de Newton-Côtes

São derivadas usando interpolação por polinômios de Lagrange.

Vantagem: Sendo  $\phi$  um interpolador de grau  $n$  nos  $n + 1$  pontos

$$x_0 = a, x_1, x_2, \dots, x_n = b,$$

então a correspondente fórmula será EXATA toda vez que  $f$  for um polinômio de grau menor ou igual a  $n$ .

$n = 1$  : Regra do Trapézio

$$\begin{aligned} \phi(x) &= \frac{(x-b)}{a-b} f(a) + \frac{(x-a)}{b-a} f(b) \\ \int_a^b f(x) &\approx \int_a^b \left( \frac{x-b}{a-b} f(a) + \frac{x-a}{b-a} f(b) \right) dx = \\ f(a) \int_a^b \frac{x-b}{a-b} dx + f(b) \int_a^b \frac{x-a}{b-a} dx &= f(a) \left[ \frac{(x-b)^2}{2(a-b)} \right]_a^b + f(b) \left[ \frac{(x-a)^2}{2(b-a)} \right]_a^b = \\ f(a) \frac{-(a-b)^2}{2(a-b)} + f(b) \frac{(b-a)^2}{2(b-a)} & \end{aligned}$$

e a após simplificação obtemos

$$\int_a^b f(x)dx \approx \frac{f(a) + f(b)}{2}(b-a) \quad (3)$$

$n = 2$ : Regra de Simpson, seja  $c = (a + b)/2$ ,  $2h = b - a$ ,

$$\phi(x) = \frac{(x-c)(x-b)}{(a-c)(a-b)}f(a) + \frac{(x-a)(x-b)}{(c-a)(c-b)}f(c) + \frac{(x-a)(x-c)}{(b-a)(b-c)}f(b)$$

e assim

$$\begin{aligned} \int_a^b f(x)dx &\approx \int_a^b \left( \frac{(x-c)(x-b)}{(a-c)(a-b)}f(a) + \frac{(x-a)(x-b)}{(c-a)(c-b)}f(c) + \frac{(x-a)(x-c)}{(b-a)(b-c)}f(b) \right) dx = \\ &\frac{f(a)}{(a-c)(a-b)} \int_a^b (x-c)(x-b)dx + \frac{f(c)}{(c-a)(c-b)} \int_a^b (x-a)(x-b)dx + \\ &\frac{f(b)}{(b-a)(b-c)} \int_a^b (x-a)(x-c)dx \end{aligned}$$

e assim

$$\begin{aligned} \int_a^b f(x)dx &\approx \frac{f(a)}{(a-c)(a-b)} \left[ \frac{(c-a)(b-a)^2}{2} - \frac{(b-a)^3}{6} \right] + \\ &\frac{f(c)}{(c-a)(c-b)} \frac{(a-b)^3}{6} + \frac{f(b)}{(b-a)(b-c)} \left[ \frac{(b-c)(b-a)^2}{2} - \frac{(b-a)^3}{6} \right] = \\ &\frac{f(a)}{(-h)(-2h)} \left[ \frac{(h)(2h)^2}{2} - \frac{(2h)^3}{6} \right] + \frac{f(c)}{(h)(-h)} \cdot \frac{(-2h)^3}{6} + \\ &\frac{f(b)}{(2h)(h)} \left[ \frac{(h)(2h)^2}{2} - \frac{(2h)^3}{6} \right] = \frac{f(a)}{2h^2} \cdot \frac{4h^3}{6} - \frac{f(c)}{h^2} \cdot \frac{-8h^3}{6} + \frac{f(b)}{2h^2} \cdot \frac{4h^3}{6} \end{aligned}$$

e a após simplificação obtemos

$$I = \int_a^b f(x)dx \approx \frac{f(a) + 4f(c) + f(b)}{6}(b-a) \quad (4)$$

**Exemplo 2.2** : Avaliar numericamente, em Scilab,

$$I = \int_0^{\pi^2/4} \text{sen}(\sqrt{1+x})dx$$

(a) usando a Regra do Trapézio; (b) usando a Regra de Simpson.

A sequência de comandos em Scilab:

```

--> a = 0; b =
--> function u = f(x)
-->   u = sin(sqrt(1+x)); endfunction
--> It = (f(a) + f(b))*(b-a)/2
--> Is = (f(a) + 4*f(c) + f(b))*(b-a)/6

```

produz as aproximações  $I = 2.2198498$  e  $I = 2.3801058$ , respectivamente para as regras de Trapézio e Simpson.

**Teorema:** Se  $f$  é uma função contínua em um intervalo  $[a, b]$ , e se  $f^{(2)}(x)$  é limitada em  $[a, b]$ , então a regra do Trapézio possui erro de truncamento local

$$E = \frac{h^3}{12} |f^{(2)}(\xi)|, \xi \in (a, b) \quad (5)$$

onde  $h = b - a$ .

**Demonstração:** omitida

**Teorema:** Se  $f$  é uma função contínua em um intervalo  $[a, b]$ , e se  $f^{(4)}(x)$  é limitada em  $[a, b]$ , então a regra de Simpson possui erro de truncamento local dado por

$$E = \frac{h^5}{90} |f^{(4)}(\xi)|, \xi \in (a, b) \quad (6)$$

onde  $h = (b - a)/2$ .

**Demonstração:** omitida

### Quadratura Composta Uniforme

No contexto de quadratura newtoniana, ou seja, das fórmulas de Newton Côtes, dependendo do tamanho do intervalo de integração e da natureza não-polinomial da função  $f$ , as fórmulas básicas apresentadas podem nos dar aproximações bem rudimentares (baixa exatidão) para o valor exato da integral definida.

Como estratégia para vencer esse obstáculo, podemos usar Regra do Trapézio e Regra de Simpson (ou qualquer outra fórmula de quadratura) em intervalos menores, aproveitando a propriedade aditiva da integral definida.

Nesse sentido, duas estratégias para solução desse problema são apresentadas.

### Quadratura Composta Fixa

Uma vez escolhido um número  $N$  de intervalos, escrevemos

$$\int_a^b f(x)dx = \int_{x_0}^{x_1} f(x)dx + \int_{x_1}^{x_2} f(x)dx + \int_{x_2}^{x_3} f(x)dx + \dots + \int_{x_{N-1}}^{x_N} f(x)dx$$

onde  $x_0 = a$ ,  $x_N = b$  e  $x_i - x_{i-1} = (b - a)/N$ . Assim, fórmulas de baixa ordem (por exemplo, Trapézio e Simpson) são aplicadas em cada sub-intervalo.

Quadratura Composta do Trapézio:

$$I = \sum_{i=1}^N \frac{f(x_{i-1}) + f(x_i)}{2} (x_i - x_{i-1}) \quad (7)$$

onde  $h = (b - a)/N$ .

Quadratura Composta de Simpson:

$$I = \sum_{i=1}^N \frac{f(x_{i-1}) + 4f(x_{i-1/2}) + f(x_i)}{6} (x_i - x_{i-1}) \quad (8)$$

onde  $h = (b - a)/N$ ,  $x_{i-1/2} = (x_{i-1} + x_i)/2$  (índice fracionário).

**Exemplo 2.3** (Regra Composta do Trapézio): Avaliar numericamente a integral

$$I = \int_0^{\pi^2/4} \text{sen}(\sqrt{1+x}) dx$$

usando a Regra Composta do Trapézio e 3 intervalos.

A sequência de comandos em *Scilab*:

```
--> N=3; x0=0; x3 = %pi*%pi/4; h = (x3-x0)/N;
--> x1 = x0 + h; x2 = x0 + 2*h;
--> function u = f(x)
--> u = sin(sqrt(1+x)); endfunction
--> I1 = (f(x0) + f(x1))*(x1-x0)/2;
--> I2 = (f(x1) + f(x2))*(x2-x1)/2;
--> I3 = (f(x2) + f(x3))*(x3-x2)/2;
--> I = I1 + I2 + I3
```

produz a aproximação  $I = 2.3636473$ .

**Exemplo 2.4** (Regra Composta de Simpson) : Avaliar numericamente a integral

$$I = \int_0^{\pi^2/4} \text{sen}(\sqrt{1+x}) dx$$

usando a Regra Composta de Simpson e 3 intervalos.

A sequência de comandos em *Scilab*:

```
--> N=3; x0=0; x3 = %pi*%pi/4; h = (x3-x0)/N;
--> x1 = x0 + h; x2 = x0 + 2*h;
--> x0m = (x0+x1)/2; x1m = (x1+x2)/2; x2m = (x2+x3)/2;
```

```
--> function u = f(x)
--> u = sin(sqrt(x)); endfunction
--> I1 = (f(x0) + 4*f(x0m) + f(x1))*(x1-x0)/6;
--> I2 = (f(x1) + 4*f(x1m) + f(x2))*(x2-x1)/6;
--> I3 = (f(x2) + 4*f(x2m) + f(x3))*(x3-x2)/6;
--> I = I1 + I2 + I3
```

produz a aproximação  $I = 2.3829249$ .

**Quadratura Composta Recursiva - quadratura de Romberg**

No contexto da Quadratura Composta Fixa, um problema que surge é : Como encontrar o  $N$  adequado antes de começar o procedimento ?

A estratégia de Quadratura Composta Recursiva vem corrigir essa dificuldade, através da avaliação de uma sequência de aproximações

$$I_n = \sum_{i=1}^n \int_{x_{i-1}}^{x_i} f(x) dx, \text{ onde } x_i - x_{i-1} = \frac{b-a}{n}$$

se possível usando, recursivamente, as aproximações anteriores ao cálculo de cada  $I_n$ . Em cada intervalo, alguma fórmula de quadratura de baixa ordem (Trapézio ou Simpson) deve ser usada. Além disso, devemos testar a convergência dos  $I_n$  ao  $n$  crescer.

**Regra Composta Recursiva do Trapézio**

Dada uma função  $f(x)$  definida no intervalo  $[a, b]$ , seja  $s_n$  a aproximação de

$$\int_a^b f(x) dx$$

que usa a Regra Composta do Trapézio e  $2^n$  intervalos. Para a determinação de  $s_{n+1}$ , 1 ponto deve ser adicionado em cada um dos  $2^n$  intervalos, determinando então  $2 \cdot 2^n = 2^{n+1}$  novos intervalos. Precisamos avaliar a função  $f$  APENAS nos  $2^n$  novos pontos adicionados, desde que seja possível fazer apenas uma correção no valor de  $s_n$ .

**Algoritmo:** (Regra Composta Recursiva Trapezoidal )

**Entrada:** função  $f$ , números reais  $a$  e  $b$ , parâmetro de tolerância TOL.

**Saída:** Aproximação  $s_n$  para  $\int_a^b f(x) dx$

**Passo 1:**  $n = 0$ ;  $h_0 = b - a$ ;  $s_0 = (f(a) + f(b))h_0/2$ ; segue=1

**Passo 2:** Enquanto segue=1

**Passo 3:**  $n = n + 1$ ;

**Passo 4:**  $h_n = \frac{h_{n-1}}{2}$ ,  $g_n = \sum_{i \geq 1, \text{ ímpar}}^{2^n - 1} f(a + i \cdot h_n)$

Tabela 1: Aplicação da Regra Composta Recursiva do Trapézio.

$n$	$s_n$	$\delta_n$
0	2.2198498	–
1	2.3400418	0.988
2	2.372051	1.569
3	2.3802356	2.163
4	2.3822954	2.762
5	2.3828113	3.364
6	2.3829403	3.966
7	2.3829725	4.567
8	2.3829806	5.170

**Passo 5:**  $s_n = \frac{s_{n-1}}{2} + g_n h_n$

**Passo 6:** Se TOL já foi alcançado, segue  $\leftarrow 0$ .

Fim Enquanto

Retorne  $s_n$

**Exemplo 2.5 :** Avaliar numericamente, em Scilab,

$$I = \int_0^{\pi^2/4} \text{sen}(\sqrt{1+x}) dx$$

usando a Regra Recursiva do Trapézio, com 5 casas decimais significativas corretas.

A sequência de comandos em *Scilab*

```
--> function u=f(x)
--> u = sin(sqrt(1+x));endfunction
--> getf('digse.sci');
--> a = 0; b = %pi^2/4;
--> n=0; hn = b-a; sn=(f(a)+f(b))*hn/2;
--> n=n+1;hn=hn/2;gn=0;for i=1:2:(2**n-1);gn=gn+f(a+i*hn);end;
    sa=sn;sn=sn/2+gn*hn;[n sn digse(sa,sn)]
```

permite construir a tabela 1, onde observamos uma convergência linear de aproximadamente 0.60 digse/iter.

### Regra Composta Recursiva de Simpson

Dada uma função  $f(x)$  definida no intervalo  $[a, b]$ , seja  $s_n$  a aproximação de

$$\int_a^b f(x) dx$$

que usa a Regra Composta de Simpson e  $2^n$  intervalos. Para a determinação de  $s_{n+1}$ , 1 ponto deve ser adicionado em cada um dos  $2^n$  intervalos, determinando então  $2 \cdot 2^n = 2^{n+1}$  novos intervalos. Precisamos avaliar a função  $f$  APENAS nos  $2^n$  novos pontos adicionados, desde que seja possível fazer apenas uma correção no valor de  $s_n$ .

**Algoritmo:** (Regra Composta Recursiva de Simpson)

**Entrada:** função  $f$ , números reais  $a$  e  $b$ , parâmetro de tolerância TOL.

**Saída:** Aproximação  $s_n$  para  $\int_a^b f(x) dx$

**Passo 1:**  $n = 0$ ;  $h_0 = (b - a)/2$ ;  $d_0 = f(a) + f(b)$ ;

**Passo 2:**  $g_0 = f(a + h_0)$ ;  $s_0 = \frac{h_0(d_0 + 4g_0)}{3}$ ; segue = 1;

**Passo 3:** Enquanto segue=1

**Passo 3:**  $n = n + 1$ ;

**Passo 4:**  $h_n = \frac{h_{n-1}}{2}$ ,  $d_n = d_{n-1} + 2g_{n-1}$

**Passo 5:**  $g_n = \sum_{i \geq 1, \text{ ímpar}} f(a + i \cdot h_n)$

**Passo 6:**  $s_n = \frac{h_n(d_n + 4g_n)}{3}$

**Passo 7:** Se TOL já foi alcançado, segue  $\leftarrow 0$ .

Fim Enquanto

Retorne  $s_n$

**Exemplo 2.6 :** Avaliar numericamente, em Scilab,

$$I = \int_0^{\pi^2/4} \text{sen}(\sqrt{1+x}) dx$$

usando a Regra Composta Recursiva de Simpson, com 4 casas decimais significativas corretas.

A sequência de comandos em *Scilab*

```
--> function u=f(x)
--> u = sin(sqrt(1+x));endfunction
--> getf('digse.sci');
--> a = 0; b = %pi^2/4;
--> n=0;hn=(b-a)/2;dn=f(a)+f(b);
--> gn=f(a+hn);sn = (dn+4*gn)*hn/3
--> n=n+1; hn=hn/2; dn=dn+2*gn; gn=0; for i=1:2:(2**(n+1)-1);
```

Tabela 2: Aplicação da Regra Composta Recursiva de Simpson.

$n$	$s_n$	$\delta_n$
0	2.3801058	–
1	2.3827208	2.659
2	2.3829638	3.690
3	2.382982	4.817
4	2.3829832	5.994

gn=gn+f(a+i\*hn);end; sa=sn; sn=(dn+4\*gn)\*hn/3; [n sn digse(sa,sn)]  
 permite construir a tabela 2, onde observamos uma convergência linear de aproximadamente 1.20 digse/iter.

## 2.2 Quadratura Gaussiana

Observamos que a expressão de Taylor com resto de Lagrange para uma função  $f(x)$  sendo interpolada em um conjunto de nós distintos  $\{x_0, x_1, x_2, \dots, x_n\}$ ,  $x_0 \geq a, x_n \leq b$  é dada por

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{f''(x_0)(x - x_0)^2}{2} + \dots + \frac{f^{(n)}(x_0)(x - x_0)^n}{n!} + \frac{f^{(n+1)}(\xi)(x - x_0)(x - x_1) \dots (x - x_n)}{(n + 1)!}, \quad (9)$$

para algum  $\xi$  entre  $x$  e  $x_0$ , e uma vez que sabemos que o resto de Lagrange deve anular-se para  $x = x_j, j = 0, 1, 2, \dots, n$ . Dessa forma, temos

$$f(x) = \phi(x) + E_{tr}(x), \quad (10)$$

onde

$$\phi(x) = \sum_{k=0}^n L_k(x)f(x_k) \quad (11)$$

$$L_k(x) = \prod_{\substack{0 \leq j \leq n \\ j \neq k}} \frac{(x - x_j)}{(x_k - x_j)}$$

é a interpoladora Lagrangiana de grau  $\leq n$ , sobre  $\{x_0, x_1, \dots, x_n\}$ , e

$$lE_{tr}(x) = p_n(x)q_n(x) \quad (12)$$

$$p_n(x) = (x - x_0)(x - x_1) \dots (x - x_n) \quad (13)$$

$$q_n(x) = \frac{f^{(n+1)}(\xi)}{(n + 1)!}. \quad (14)$$

Sendo  $w(x)$  uma função peso escolhida, que deve satisfazer  $w(x) \geq 0$  em  $[a, b]$ , integramos a equação (10), obtendo

$$I = \int_a^b f(x)w(x)dx = \int_a^b \phi(x)w(x)dx + \int_a^b E_{tr}(x)w(x)dx. \quad (15)$$

Nosso objetivo nesta seção será apresentar estratégias que existem para minimizar

$$\int_a^b E_{tr}(x)w(x)dx = \int_a^b p_n(x)q_n(x)w(x)dx \quad (16)$$

para um dado intervalo  $[a, b]$ , usando projeção em espaços polinomiais gerados por bases ortogonais conhecidas. Deverá ser sempre lembrado que é a estratégia de projeção que determinará o valor dos parâmetros da respectiva fórmula de quadratura

$$I = \int_a^b f(x)w(x)dx \approx \int_a^b \sum_{k=0}^n L_k(x)f(x_k)w(x)dx = \sum_{k=0}^n w_k f(x_k) \quad (17)$$

onde

$$w_k = \int_a^b L_k(x)w(x)dx. \quad (18)$$

determina os pesos da fórmula de quadratura.

Será importante definirmos ortogonalidade generalizada (com *peso*) em um intervalo  $[a, b]$ : dois polinômios  $p(x)$  e  $q(x)$  são ortogonais em um intervalo  $[a, b]$ , com peso  $w(x)$ , se

$$\int_a^b p(x)q(x)w(x)dx = 0. \quad (19)$$

### Quadratura de Gauss-Legendre

Apresentamos os polinômios de Legendre, definidos recursivamente no intervalo  $[-1, 1]$  por

$$P_0(x) = 1, P_1(x) = x$$

$$P_{n+1}(x) = \left(\frac{2n+1}{n+1}\right)xP_n(x) - \frac{n}{n+1}P_{n-1}(x), \quad n \geq 1 \quad (20)$$

Estes polinômios têm a propriedade de ser ortogonais no intervalo  $[-1, 1]$ , com peso  $w(x) \equiv 1$ , isto é,

$$\int_{-1}^1 P_n(x)P_j(x)dx = \begin{cases} 0, & n \neq j \\ \frac{1}{2^n}, & n = j \end{cases} \quad (21)$$

Decompondo os polinômios  $p_n$  e  $q_n$ , definidos em (13)-(14) em termos dos polinômios de Legendre, escrevemos

$$p_n(x) = \sum_{k=0}^{n+1} b_k P_k(x) \quad (22)$$

$$q_n(x) = \sum_{j=0}^n c_j P_j(x). \quad (23)$$

Dessa forma, temos

$$p_n(x)q_n(x) = \left( \sum_{k=0}^{n+1} b_k P_k(x) \right) \left( \sum_{j=0}^n c_j P_j(x) \right) = \sum_{k=0}^n \sum_{j=0}^n b_k c_j P_k(x) P_j(x) + b_{n+1} \sum_{j=0}^n c_j P_k(x) P_j(x)$$

e então, substituindo essa expressão em (10) e movendo para fora da integral as constantes

$$\int_{-1}^1 p_n(x)q_n(x)dx = \sum_{k=0}^n \sum_{j=0}^n b_k c_j \int_{-1}^1 P_k(x)P_j(x)dx + b_{n+1} \sum_{j=0}^n c_j \int_{-1}^1 P_k(j)P_j(x)dx = \sum_{k=0}^n b_k c_k \int_{-1}^1 P_k^2(x)dx$$

em virtude das relações de ortogonalidade (14). Assim, podemos anular a expressão acima desde que estabeleçamos

$$b_0 = b_1 = \dots = b_k = 0,$$

o que implica  $p_n(x) = b_{n+1}P_{n+1}(x)$  e que por sua vez implica que as abscissas  $x_0, x_1, \dots, x_n$  sejam as  $n+1$  raízes de  $P_{n+1}(x)$ , o polinômio de Legendre de grau  $n+1$ . Também implica (veja Sperandio 2003, Ralston 1978) que a correspondente fórmula de quadratura será exata para polinômios de grau menor ou igual a  $(2n+1)$ .

A tabela 3 mostra, para  $n = 1, 2, 3$ , as raízes dos polinômios de Legendre de grau  $n+1$ .

Dessa forma, a fórmula de quadratura de Gauss-Legendre para  $n+1$  pontos pode ser escrita

$$\int_{-1}^1 f(x)dx \approx \sum_{k=0}^n w_k f(x_k) \quad (24)$$

onde  $x_0, x_1, x_2, \dots, x_n$  são definidos na tabela 3, para  $n = 1, 2, 3$ .

Tabela 3: Parâmetros da Quadratura de Gauss-Legendre.

$n$	$x_k$	$w_k$
1	$-\sqrt{1/3}$	1
	$\sqrt{1/3}$	1
2	$-\sqrt{3/5}$	5/9
	0	8/9
	$\sqrt{3/5}$	5/9
3	$-\sqrt{3/7 + (6/7)\sqrt{2/15}}$	$1/2 - (1/18)\sqrt{15/2}$
	$-\sqrt{3/7 - (6/7)\sqrt{2/15}}$	$1/2 + (1/18)\sqrt{15/2}$
	$\sqrt{3/7 - (6/7)\sqrt{2/15}}$	$1/2 + (1/18)\sqrt{15/2}$
	$\sqrt{3/7 + (6/7)\sqrt{2/15}}$	$1/2 - (1/18)\sqrt{15/2}$

Em particular, ressaltamos os casos  $n = 1$  e  $n = 2$ , que são os mais usados: ( $n = 1$ ) Quadratura de Gauss-Legendre usando 2 pontos.

$$\int_{-1}^1 f(x)dx \approx f\left(-\sqrt{\frac{1}{3}}\right) + f\left(\sqrt{\frac{1}{3}}\right) \quad (25)$$

exata para polinômios de grau  $\leq 3$ .

( $n = 2$ ) Quadratura de Gauss-Legendre usando 3 pontos.

$$\int_{-1}^1 f(x)dx \approx \frac{5}{9}f\left(-\sqrt{\frac{3}{5}}\right) + \frac{8}{9}f(0) + \frac{5}{9}f\left(\sqrt{\frac{3}{5}}\right) \quad (26)$$

exata para polinômios de grau  $\leq 5$ .

**Exemplo 2.7 :** Encontrar a fórmula de Gauss-Legendre para

$$\int_0^1 f(x)dx$$

usando 2 pontos.

**Solução:** pensamos em uma substituição  $u = \alpha x + \beta$ , onde então

$$\begin{cases} -1 = \alpha(0) + \beta \\ 1 = \alpha(1) + \beta \end{cases} \Rightarrow \alpha = 2, \beta = -1 \Rightarrow u = 2x - 1.$$

Fazendo a substituição  $u = 2x - 1, du = 2dx$

$$\int_0^1 f(x)dx = \int_{-1}^1 f\left(\frac{u+1}{2}\right) \frac{du}{2} = \int_{-1}^1 g(u)du,$$

onde

$$g(u) = \frac{f((u+1)/2)}{2}.$$

Caso  $f$  seja polinômio de grau  $n$  sabemos que  $g$  também será polinômio de grau  $n$ .

Assim

$$\int_0^1 f(x)dx = g\left(\frac{-1}{\sqrt{3}}\right) + g\left(\frac{1}{\sqrt{3}}\right) = \frac{1}{2}f\left(\frac{-1/\sqrt{3}+1}{2}\right) + \frac{1}{2}f\left(\frac{1/\sqrt{3}+1}{2}\right)$$

e assim a fórmula

$$\int_0^1 f(x)dx = \frac{1}{2}f\left(\frac{3-\sqrt{3}}{6}\right) + \frac{1}{2}f\left(\frac{3+\sqrt{3}}{6}\right) \quad (27)$$

é exata para todos os polinômios de grau  $\leq 3$ .

**Exemplo 2.8** : Avaliar numericamente, em Scilab,

$$I = \int_0^1 \ln(1 + \sqrt{x}) dx$$

usando o método de Gauss-Legendre para 2 pontos.

Usaremos a fórmula obtida no exemplo anterior. Os comandos em Scilab

```
--> w0= 0.5; w1 = 0.5;
--> function u = f(x)
-->   u = log( 1 + sqrt(x) ); endfunction
--> x0= (3-sqrt(3))/6; x1 = (3+sqrt(3))/6;
--> I = w0*f(x0) + w1*f(x1)
```

produz a aproximação  $I = 0.5068943$ .

**Exemplo 2.9** : Avaliar numericamente

$$\int_0^{\pi^2/4} \text{sen}(\sqrt{1+x}) dx$$

usando quadratura de Gauss-Legendre com 2 pontos.

Para obtenção da fórmula apropriada, é necessária uma mudança de variável do intervalo  $[0, \pi^2/4]$  para o intervalo  $[-1, 1]$  :  $u = \alpha x + \beta$  onde

$$\begin{aligned} x = 0 &\Rightarrow \alpha(0) + \beta = -1 \Rightarrow \beta = -1 \\ x = \pi^2/4 &\Rightarrow \alpha\pi^2/4 - 1 = 1 \Rightarrow \alpha = 2/(\pi^2/4) = 8/(\pi^2) \end{aligned}$$

e então  $u = 8x/\pi^2 - 1$ ,  $du = 8/\pi^2 dx$ , e

$$\int_0^{\pi^2/4} f(x)dx = \int_{-1}^1 f\left(\frac{\pi^2(u+1)}{8}\right) \frac{\pi^2}{8} du = \int_{-1}^1 \frac{\pi^2}{8} f\left(\frac{\pi^2(u+1)}{8}\right) du.$$

Usando (27), obtemos

$$\int_0^{\pi^2/4} f(x)dx = \frac{\pi^2}{8} \left[ f\left(\frac{\pi^2}{8} \left(\frac{-1}{\sqrt{3}} + 1\right)\right) + f\left(\frac{\pi^2}{8} \left(\frac{1}{\sqrt{3}} + 1\right)\right) \right]$$

e segue que a fórmula

$$\int_0^{\pi^2/4} f(x)dx = 1.2337f(0.521423) + 1.2337f(1.94598)$$

é exata para polinômios de grau menor ou igual a 3.

A sequência de comandos em Scilab:

```
--> function u = f(x)
-->   u = sin(sqrt(1+x));endfunction
--> w0 = 1.2337; w1 = w0;
--> x0 = 0.521423; x1 = 1.94598;
--> I = w0*f(x0) + w1*f(x1)
```

produz a aproximação  $I = 2.3848163$ .

**Quadratura de Gauss-TChebyshev**

Apresentamos os polinômios de TChebyshev  $T_n(x) = \cos(n * \arccos(x))$ ,  $n \geq 0$ , definidos no intervalo  $[-1, 1]$ . Sabidamente, satisfazem a recursão

$$\begin{aligned} T_0(x) &= 1, T_1(x) = x \\ T_{n+1}(x) &= 2xT_n(x) - T_{n-1}(x), \quad n \geq 1. \end{aligned} \quad (28)$$

Estes polinômios têm a propriedade de ser ortogonais no intervalo  $[-1, 1]$ , com peso  $w(x) = (1 - x^2)^{-1/2}$ , isto é,

$$\int_{-1}^1 T_n(x)T_j(x) \frac{1}{\sqrt{1-x^2}} dx = \begin{cases} 0, & n \neq j \\ \frac{\pi}{2}, & n = j \neq 0 \\ \pi, & n = j = 0 \end{cases} \quad (29)$$

Decompondo os polinômios  $p_n$  e  $q_n$ , definidos em (13)-(14) em termos dos polinômios de TChebyshev, escrevemos

$$p_n(x) = \sum_{k=0}^{n+1} b_k T_k(x) \quad (30)$$

$$q_n(x) = \sum_{j=0}^n c_j T_j(x). \quad (31)$$

Dessa forma, analogamente ao desenvolvimento na seção (2.2), usamos a propriedade de ortogonalidade dada em (29) para obter  $p_n(x) = b_{n+1} T_{n+1}(x)$  e que por sua vez implica que as abscissas  $x_0, x_1, \dots, x_n$  sejam as  $n+1$  raízes de  $T_{n+1}(x)$ , o polinômio de TChebyshev de grau  $n+1$ . Também implica que a correspondente fórmula de quadratura será exata para polinômios de grau menor ou igual a  $(2n+1)$ .

Pode-se mostrar que as raízes de  $T_{n+1}(x)$  são dadas por

$$x_k = \cos\left(\frac{\pi(2k+1)}{2(n+1)}\right) \quad (32)$$

para  $k = 0, 1, 2, \dots, n$ . Pode-se mostrar também que os pesos são dados por

$$w_k = \frac{\pi}{n+1} \quad (33)$$

para  $k = 0, 1, 2, \dots, n$ . Dessa forma, a fórmula de quadratura de Gauss-TChebyshev para  $n+1$  pontos pode ser escrita

$$\int_{-1}^1 \frac{f(x)}{\sqrt{1-x^2}} dx \approx \frac{\pi}{n+1} \sum_{k=0}^n f(x_k) \quad (34)$$

onde  $x_0, x_1, x_2, \dots, x_n$  são definidos por (32).

**Exemplo 2.10** : *Avaliar numericamente*

$$\int_0^{\pi^2/4} \text{sen}(\sqrt{1+x}) dx$$

usando quadratura de Gauss-TChebyshev com 4 pontos.

Na obtenção da fórmula apropriada, usamos a mudança de variável obtida no Exemplo (2.9):

$$u = 8x/\pi^2 - 1, \quad du = 8/\pi^2 dx,$$

e assim

$$\int_0^{\pi^2/4} f(x) dx = \int_{-1}^1 f\left(\frac{\pi^2(u+1)}{8}\right) \frac{\pi^2}{8} du = \int_{-1}^1 \frac{\pi^2}{8} f\left(\frac{\pi^2(u+1)}{8}\right) du.$$

e assim

$$\int_0^{\pi^2/4} f(x) dx = \int_{-1}^1 \frac{g(u)}{\sqrt{1-u^2}} du$$

onde

$$g(u) = \frac{\pi^2 \sqrt{1-u^2}}{8} f\left(\frac{\pi^2(u+1)}{8}\right).$$

A sequência de comandos em *Scilab*

```
--> function y = f(x)
--> y = sin(sqrt(1+x)); endfunction
--> function y = g(u)
--> y = %pi^2*sqrt(1-u*u)*f(%pi^2*(u+1)/8)/8;
--> endfunction
--> // n = 3 //
--> u0 = cos(%pi*(2*0+1)/8);
--> u1 = cos(%pi*(2*1+1)/8);
--> u2 = cos(%pi*(2*2+1)/8);
--> u3 = cos(%pi*(2*3+1)/8);
--> I = (%pi/4)*(g(u0) + g(u1) + g(u2) + g(u3))
```

produz a aproximação  $I = 2.4401498$ .

**Quadratura de Gauss-Laguerre**

Apresentamos os polinômios de Laguerre, definidos em  $[0, \infty)$  como sendo soluções não-singulares da equação diferencial  $xy'' + (1-x)y' + ny = 0$ . Sabidamente, esta equação diferencial tem soluções não-singulares somente se  $n$  é inteiro não-negativo.

Polinômios de Laguerre, que satisfazem a relação recursiva

$$L_0(x) = 1, L_1(x) = 1 - x$$

$$L_{n+1}(x) = \frac{1}{n+1} [(2n+1-x)L_n(x) - nL_{n-1}(x)], \quad n \geq 1, \quad (35)$$

têm a propriedade de ser ortogonais no intervalo  $[0, \infty)$ , com peso  $w(x) = e^{-x}$ , isto é,

$$\int_0^{\infty} L_n(x)L_j(x)e^{-x} dx = \begin{cases} 0, & n \neq j \\ 1, & n = j \end{cases} \quad (36)$$

Tabela 4: Parâmetros da Quadratura de Gauss-Laguerre.

$n$	$x_k$	$w_k$
1	0.585786437627	0.853553390593
	3.414213562373	0.14644669407
2	0.415774556783	0.711093009929
	2.294280360279	0.278517733569
	6.289945082937	0.103892565016
3	0.322547689619	0.603154104342
	1.745761101158	0.357418692438
	4.536620296921	$0.388879085150 \cdot 10^{-1}$
	9.395070912301	$0.539294705561 \cdot 10^{-3}$
4	0.2635603197	0.5217556106
	1.413403059	0.39866681108
	3.59642577	$0.75942244868 \cdot 10^{-1}$
	7.0858100059	$0.36117586799 \cdot 10^{-2}$
	12.64080084423	$0.23369972386 \cdot 10^{-4}$

Decompondo os polinômios  $p_n$  e  $q_n$ , definidos em (13)-(14) em termos dos polinômios de Laguerre, escrevemos

$$p_n(x) = \sum_{k=0}^{n+1} b_k L_k(x) \quad (37)$$

$$q_n(x) = \sum_{j=0}^n c_j L_j(x). \quad (38)$$

Dessa forma, analogamente ao desenvolvimento na seção 2.2, usamos a propriedade de ortogonalidade dada em (36) para obter  $p_n(x) = b_{n+1} L_{n+1}(x)$  e que por sua vez implica que as abscissas  $x_0, x_1, \dots, x_n$  sejam as  $n+1$  raízes de  $L_{n+1}(x)$ , o polinômio de Laguerre de grau  $n+1$ . Também implica que a correspondente fórmula de quadratura será exata para polinômios de grau menor ou igual a  $(2n+1)$ .

Pode-se mostrar que as raízes de  $L_{n+1}(x)$  e os pesos  $w_k$ ,  $k = 0, 1, \dots, n$  para esta quadratura são dadas conforme a tabela 4.

Dessa forma, a fórmula de quadratura de Gauss-Laguerre para  $n+1$  pontos pode ser escrita

$$\int_0^{\infty} f(x) e^{-x} dx \approx \sum_{k=0}^n w_k f(x_k) \quad (39)$$

onde  $x_0, x_1, x_2, \dots, x_n$  são definidos pela tabela 4.

**Exemplo 2.11** : Avaliar numericamente

$$\int_e^{\infty} \frac{dx}{1+x^{5/2}}$$

usando quadratura de Gauss-Laguerre com 4 pontos.

A mudança de variável  $u = x - e$ ,  $du = dx$  permite escrevermos

$$\int_e^{\infty} \frac{dx}{1+x^{5/2}} = \int_0^{\infty} \frac{du}{1+(u+e)^{5/2}} = \int_0^{\infty} g(u) e^{-u} du$$

onde  $g(u) = \frac{e^u}{1+(u+e)^{5/2}}$ .

A sequência de comandos em *Scilab*

```
--> function y = g(u)
--> y = exp(u)/(1 + (u+exp(1))^2.5); endfunction
--> u0 = 0.322547689619; w0 = 0.603154104342;
--> u1 = 1.745761101158; w1 = 0.357418692438;
--> u2 = 4.536620296921; w2 = 0.388879085150*10^(-1);
--> u3 = 9.395070912301; w3 = 0.539294705561*10^(-3);
--> I = w0*g(u0) + w1*g(u1) + w2*g(u2) + w3*g(u3)
```

produz a aproximação  $I = 0.1342564$ .

**Quadratura de Gauss-Hermite**

Apresentamos os polinômios de Hermite como sendo soluções não-singulares da equação diferencial  $xy'' - 2xy' + 2ny = 0$ .

Polinômios de Hermite, que satisfazem a relação recursiva

$$\begin{aligned} H_0(x) &= 1 \\ H_{n+1}(x) &= 2xH_n(x) - H'_n(x), \quad n \geq 0, \end{aligned} \quad (40)$$

têm a propriedade de ser ortogonais no intervalo  $(-\infty, \infty)$ , com peso  $w(x) = e^{-x^2}$ , isto é,

$$\int_{-\infty}^{\infty} H_n(x) H_j(x) e^{-x^2} dx = \begin{cases} 0, & n \neq j \\ \sqrt{\pi} n! 2^n, & n = j \end{cases} \quad (41)$$

Decompondo os polinômios  $p_n$  e  $q_n$ , definidos em (13)-(14) em termos dos polinômios de Hermite, escrevemos

$$p_n(x) = \sum_{k=0}^{n+1} b_k H_k(x) \quad (42)$$

$$q_n(x) = \sum_{j=0}^n c_j H_j(x). \quad (43)$$

Tabela 5: Parâmetros da Quadratura de Gauss-Hermite.

$n$	$x_k$	$w_k$
1	-0.7071067811	0.8862269255
	0.7071067811	0.8862269255
2	-1.22447448714	0.2954089752
	0.00000000000	1.1816359006
	1.22447448714	0.2954089752
3	-1.6506801239	0.0813128354
	-0.5246476233	0.8049140900
	0.5246476233	0.8049140900
	1.6506801239	0.0813128354
4	-2.021828705	0.0199532421
	-0.9585724646	0.3936193232
	0	0.9453087205
	0.9585724646	0.3936193232
	2.021828705	0.0199532421

Dessa forma, analogamente ao desenvolvimento na seção 2.2, usamos a propriedade de ortogonalidade dada em (41) para obter  $p_n(x) = b_{n+1}H_{n+1}(x)$  e que por sua vez implica que as abscissas  $x_0, x_1, \dots, x_n$  sejam as  $n + 1$  raízes de  $H_{n+1}(x)$ , o polinômio de Hermite de grau  $n + 1$ . Também implica que a correspondente fórmula de quadratura será exata para polinômios de grau menor ou igual a  $(2n + 1)$ .

Pode-se mostrar que as raízes de  $H_{n+1}(x)$  e os pesos  $w_k$ ,  $k = 0, 1, \dots, n$  para esta quadratura são dadas conforme a tabela 5.

Dessa forma, a fórmula de quadratura de Gauss-Hermite para  $n + 1$  pontos pode ser escrita

$$\int_{-\infty}^{\infty} f(x)e^{-x^2} dx \approx \sum_{k=0}^n w_k f(x_k) \quad (44)$$

onde  $x_0, x_1, x_2, \dots, x_n$  são definidos pela tabela 5.

**Exemplo 2.12 :** Avaliar numericamente

$$\int_{\text{infy}}^{\infty} \frac{e^{-|x|} dx}{1+x^3}$$

usando quadratura de Gauss-Hermite com 4 pontos.

Não há necessidade de mudança de variável. Temos

$$\int_{-\infty}^{\infty} \frac{e^{-|x|} dx}{1+x^3} = \int_{-\infty}^{\infty} g(x)e^{-x^2} dx$$

$$\text{onde } g(x) = \frac{e^{x^2} e^{-|x|}}{1+x^3}.$$

A sequência de comandos em *Scilab*

```
--> function y = g(u)
--> y = exp(u*u - abs(u))/(1 + u*u*u); endfunction
--> u0=-1.6506801239; w0 = 0.0813128354;
--> u1=-0.5246476233; w1=0.8049140900;
--> u2= 0.5246476233; w2= 0.8049140900;
--> u3=1.6506801239; w3=0.0813128354;
--> I = w0*g(u0) + w1*g(u1) + w2*g(u2) + w3*g(u3)
```

produz a aproximação  $I = 1.2564603$ .

### Quadratura de integrandos com singularidade

Ao empregarmos quadratura gaussiana de qualquer tipo e ordem, nenhum dos nós  $x_0, x_1, \dots, x_n$  coincide com qualquer dos extremos  $x = a$  ou  $x = b$ .

Estratégias de quadratura gaussiana (que pela razão acima é classificada como uma estratégia de quadratura *aberta*) será muito útil no cálculo numérico de integrais definidas de funções  $f(x)$  que não estão definidas em  $x = a$  ou  $x = b$ , mas possuem integral no intervalo  $[a, b]$ . Os próximos exemplos mostram situações como essas.

**Exemplo 2.13** Encontre uma aproximação numérica para a integral mal comportada

$$I = \int_0^2 \frac{e^x}{\sqrt{x(2-x)}} dx$$

usando quadratura de Gauss-TChebyshev e 6 pontos.

Aplicando a quadratura de Gauss-TChebyshev, fazemos a mudança de variável  $u = x - 1$ ,  $du = dx$ , que implica

$$I = \int_{-1}^1 \frac{e^{u+1}}{\sqrt{(u+1)(1-u)}} du = \int_{-1}^1 \frac{g(u)}{\sqrt{1-u^2}} dx$$

onde

$$g(u) = \frac{\sqrt{1-u^2} e^{u+1}}{\sqrt{(u+1)(1-u)}}.$$

A sequência de comandos em *Scilab*

```
--> n = 5; wk = %pi/(n+1);
--> function y = g(u)
--> y = sqrt(1-u*u)*exp(u+1)/sqrt((u+1)*(1-u));endfunction
--> u0 = cos( 1*%pi/(2*(n+1)) ); u1=cos(3*%pi/(2*(n+1)));
--> u2 = cos( 5*%pi/(2*(n+1)) ); u3 = cos( 7*%pi/(2*(n+1)) );
--> u4 = cos( 9*%pi/(2*(n+1)) ); u5 = cos( 11*%pi/(2*(n+1)) );
--> I = wk*( g(u0) + g(u1) + g(u2) + g(u3) + g(u4) + g(u5) )
```

produz a aproximação  $I = 10.811866$ .

**Exemplo 2.14** : *Aproxime numericamente a integral mal comportada*

$$I = \int_{-\pi/4}^{\pi/4} \frac{\tan(x)}{e^x \sqrt[3]{x}} dx$$

usando quadratura de Gauss-Legendre e 4 pontos.

Definimos dois sub-intervalos de integração, separados pelo ponto de singularidade  $x = 0$ :

$$I = \int_{-\pi/4}^0 \frac{\tan(x)}{e^x \sqrt[3]{x}} dx + \int_0^{\pi/4} \frac{\tan(x)}{e^x \sqrt[3]{x}} dx$$

e assim estão definidas duas quantidades

$$I_1 = \int_{-\pi/4}^0 \frac{\tan(x)}{e^x \sqrt[3]{x}} dx$$

$$I_2 = \int_0^{\pi/4} \frac{\tan(x)}{e^x \sqrt[3]{x}} dx$$

a serem calculadas. Para a primeira integral, a mudança de variável proposta é  $u = \alpha x + \beta$ , onde

$$\begin{cases} -1 = \alpha(-\pi/4) + \beta \\ 1 = \alpha(0) + \beta \end{cases} \Rightarrow \alpha = \frac{8}{\pi}, \beta = 1$$

e assim  $x = \frac{\pi(u-1)}{8}$ ,  $dx = \pi du/8$ , o que implica

$$I_1 = \int_{-\pi/4}^0 \frac{\tan(x)}{e^x \sqrt[3]{x}} dx = \int_{-1}^1 f\left(\frac{\pi(u-1)}{8}\right) \frac{\pi du}{8} \Rightarrow I_1 = \int_{-1}^1 g(u) du$$

onde  $g(u) = \frac{\pi f\left(\frac{\pi(u-1)}{8}\right)}{8}$ .

Para a segunda integral, a mudança de variável proposta é  $u = \alpha x + \beta$ , onde

$$\begin{cases} -1 = \alpha(0) + \beta \\ 1 = \alpha(\pi/4) + \beta \end{cases} \Rightarrow \alpha = \frac{8}{\pi}, \beta = -1$$

e assim  $x = \frac{\pi(u+1)}{8}$ ,  $dx = \pi du/8$ , o que implica

$$I_2 = \int_0^{\pi/4} \frac{\tan(x)}{e^x \sqrt[3]{x}} dx = \int_{-1}^1 f\left(\frac{\pi(u+1)}{8}\right) \frac{\pi du}{8} \Rightarrow I_2 = \int_{-1}^1 p(u) du$$

onde  $p(u) = \frac{\pi f\left(\frac{\pi(u+1)}{8}\right)}{8}$ .

A sequência de comandos em *Scilab*

```
--> function u = cbrt(x)
--> u = sign(x)*abs(x)^(1/3);endfunction
--> function y = f(x)
--> y = tan(x)/(exp(x)*cbrt(x)); endfunction
--> function y = g(u)
--> y = %pi*f( %pi*(u-1)/8 ); endfunction
--> function y = p(u)
--> y = %pi*f( %pi*(u+1)/8 ); endfunction
--> u0 = -sqrt(3/7 + (6/7)*sqrt(2/15)); w0 = 1/2 -(1/18)*sqrt(15/2);
--> u1 = -sqrt(3/7 - (6/7)*sqrt(2/15)); w1 = 1/2 +(1/18)*sqrt(15/2);
--> u2 = sqrt(3/7 - (6/7)*sqrt(2/15)); w2 = 1/2 +(1/18)*sqrt(15/2);
--> u3 = sqrt(3/7 + (6/7)*sqrt(2/15)); w3 = 1/2 -(1/18)*sqrt(15/2);
--> I1 = w0*g(u0) + w1*g(u1) + w2*g(u2) + w3*g(u3)
--> I2 = w0*p(u0) + w1*p(u1) + w2*p(u2) + w3*p(u3)
```

produz as aproximações  $I_1 = 0.7529783$ ,  $I_2 = 0.2751271$  e assim  $I = 1.0281055$ .

### 3 Métodos de passo simples e múltiplo; explícitos e implícitos.

**Exemplo 3.1** : *Solução numérica de problemas iniciais ordinários*

Consideremos o problema de valor inicial

$$\begin{cases} dx/dt - x(x-4) = 0, x \in (0,4) \\ x(0) = 1/10 \end{cases}$$

A derivada primeira será aproximada pelas fórmulas em diferenças finitas centrais:

$$d_1 = \frac{x(t+h) - x(t-h)}{2h}, \quad d_2 = \frac{x(t+h) - x(t)}{h}$$

que já mostramos serem consistentes e possuírem erro de truncamento de segunda e primeira ordens, respectivamente.

O intervalo  $[0, 1]$  será discretizado por  $0 = t_0 < t_1 < \dots < t_N = 1$  onde  $N$  é um inteiro fixo e  $t_i = i\Delta t$ ,  $i = 0, 1, 2, \dots, N$ , onde  $\Delta t = 1/N$ .

Definimos  $x_i = x(t_i)$ ,  $i = 0, 1, 2, \dots, N$ .

Aplicando a fórmula da aproximação  $d_1$ :

$$\begin{aligned} \frac{x_{i+1} - x_{i-1}}{2\Delta t} - x_i(x_i - 4) &= 0, \quad i > 0 \\ \frac{x_1 - x_0}{\Delta t} - x_0(x_0 - 4) &= 0 \\ x_0 &= 1/10 \end{aligned}$$

e temos então o esquema numérico:

$$\begin{cases} x_0 = 1/10, & x_1 = x_0 + \Delta t x_0(x_0 - 4) \\ x_{i+1} = x_i + \Delta t x_i(x_i - 4), & i > 0 \end{cases} \quad (E1)$$

também chamado **esquema central**.

Aplicando a fórmula da aproximação  $d_2$ :

$$\begin{aligned} \frac{x_{i+1} - x_i}{\Delta t} - x_i(x_i - 4) &= 0, \quad i \geq 0 \\ x_0 &= 1/10 \end{aligned}$$

e temos então o esquema numérico:

$$\begin{cases} x_0 = 1/10 \\ x_{i+1} = x_i + \Delta t x_i(x_i - 4), & i \geq 0 \end{cases} \quad (E2)$$

também chamado **esquema de Euler**.

**Definição 3.1** Um método (esquema) numérico é **consistente** se o correspondente erro de truncamento tende a zero quando o tamanho do passo tende a zero.

Os dois esquemas propostos são consistentes uma vez que as respectivas aproximações usadas são consistentes.

Manipulando nosso problema de valor inicial analiticamente:

$$dx/dt = x(x - 4) \Rightarrow \int_{x_0}^x \frac{du}{u(u - 4)} = \int_0^t ds$$

e então

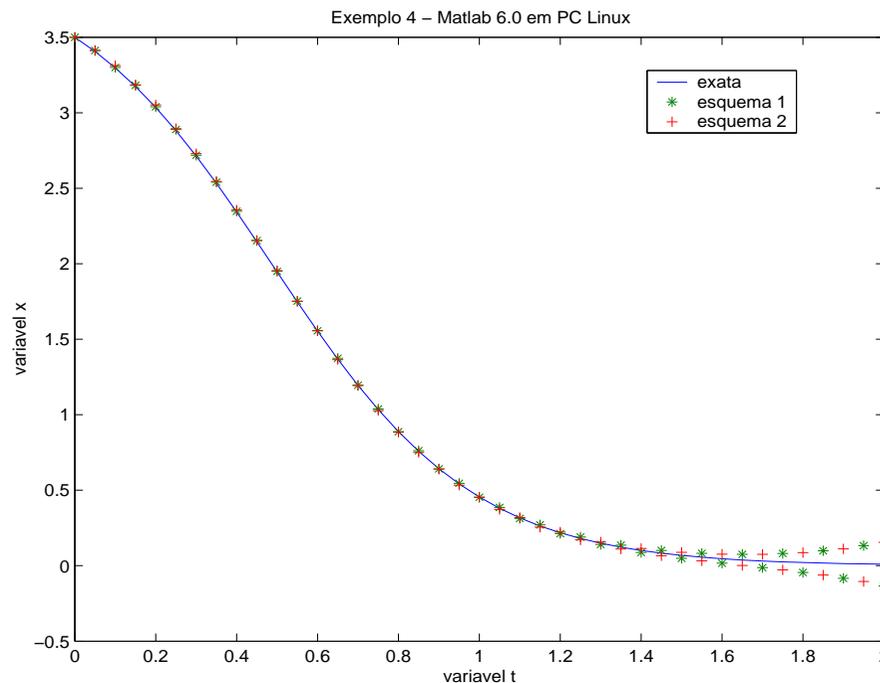
$$\frac{1}{4} \ln \left| \frac{x - 4}{x} \right| - \frac{1}{4} \ln \left| \frac{x_0 - 4}{x_0} \right| = t.$$

Assim,

$$\frac{(x - 4)x_0}{x(x_0 - 4)} = e^{4t} \Rightarrow x(t) = \frac{4x_0}{x_0 + (4 - x_0)e^{4t}}$$

e temos uma expressão analítica para a solução  $x(t)$ .

A figura abaixo mostra o resultado da solução numérica em Matlab.



**Exemplo 3.2** : encontrar solução numérica para o PVI

$$\begin{cases} \frac{dy}{dt} = \frac{\cos(t)}{t + y^2} \\ y(1) = 1/2 \end{cases}$$

no intervalo  $[1, 10]$ , usando o método de Euler e  $\Delta t = 0.1$ .

O esquema numérico

$$\begin{cases} \frac{y_{i+1} - y_i}{\Delta t} = \frac{\cos(t_i)}{t_i + y_i^2}, & i = 0, 1, 2, \dots, n - 1 \\ y_0 = 0.5 \end{cases}$$

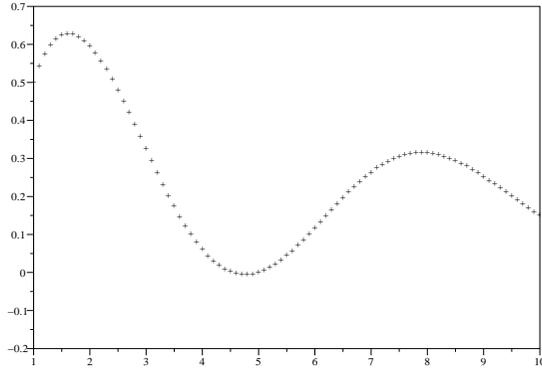


Figura 1: Solução numérica do exemplo 3.2 via Método de Euler.

é proposto. Dessa forma, estabelecemos a recursão

$$\begin{cases} y_0 = 0.5 \\ y_{i+1} = y_i + \frac{(\Delta t) \cos(t_i)}{t_i + y_i^2}, \quad i = 0, 1, 2, \dots, n-1 \end{cases}$$

A sequência de comandos em *Scilab*

```
--> h=0.1; n = (10-1)/h; t=[1:h:10]';
--> y = 0*t; y(1)=0.5; // inicializacao //
--> for i=1:n
-->     ti= 1 + (i-1)*h; yi = y(i);
-->     y(i+1)=yi + h*cos(ti)/(ti + yi*yi);
--> end
--> [t y]
--> plot2d(t,y,-1)
```

produzem a Figura 1.

### Exemplo 3.3 : Solução numérica da equação de difusão

Considere o problema de condições de contorno

$$\begin{aligned} v_t &= \nu v_{xx}, \quad x \in (0, 1), \quad t > 0 \\ v(x, 0) &= f(x), \quad x \in [0, 1] \\ v(0, t) &= a(t), \quad v(1, t) = b(t), \quad t \geq 0 \end{aligned}$$

onde  $f(0) = a(0)$  e  $f(1) = b(0)$ . Tipicamente esse problema modela a condução unidimensional de calor em uma barra limitada representada pelo segmento  $[0, 1]$ . A temperatura da barra é controlada nos extremos  $x = 0$  e  $x = 1$  por meio de duas funções  $a(t)$  e  $b(t)$ . A temperatura inicial (instante  $t = 0$ ) é dada por uma função  $f(x)$  e o parâmetro  $\nu$  é chamado **coeficiente de condutividade térmica**.

A discretização a ser adotada é:  $v_k^n = v(x_k, t_n)$ , onde  $t_n = n\Delta t$ ,  $n = 0, 1, 2, \dots, N$  e  $x_k = k \cdot \Delta x$ ,  $k = 0, 1, 2, \dots, K$ , onde  $\Delta t > 0$  e  $\Delta x = 1/K$ .

A derivada parcial  $v_t$  será aproximada pela diferença finita:

$$\frac{v_k^{n+1} - v_k^n}{\Delta t}$$

que verifica

$$\frac{v_k^{n+1} - v_k^n}{\Delta t} = \frac{\partial v}{\partial t}(x_k, t_n) + \frac{\Delta t}{2} \frac{\partial^2 v(x_k, \eta_1)}{\partial t^2}$$

Por outro lado, derivada parcial  $v_{xx}$  será aproximada pela diferença central:

$$\frac{v_{k+1}^n - 2v_k^n + v_{k-1}^n}{(\Delta x)^2}$$

que verifica

$$\frac{v_{k+1}^n - 2v_k^n + v_{k-1}^n}{(\Delta x)^2} = \frac{\partial^2 v(x_k, t_n)}{\partial x^2} + \frac{(\Delta x)^2}{4!} \left( \frac{\partial^4 v(\xi_1, t_n)}{\partial x^4} + \frac{\partial^4 v(\xi_2, t_n)}{\partial x^4} \right).$$

O erro de truncamento de nosso método é

$$\begin{aligned} \tau_k^n &= v_t(x_k, t_n) - \nu v_{xx}(x_k, t_n) = \frac{v_k^{n+1} - v_k^n}{\Delta t} - \frac{\nu}{(\Delta x)^2} (v_{k+1}^n - 2v_k^n + v_{k-1}^n) - \\ &\quad \frac{\Delta t}{2} \frac{\partial^2 v}{\partial t^2}(x_k, \eta_1) + \frac{\nu(\Delta x)^2}{4!} \left( \frac{\partial^4 v(\xi_1, t_n)}{\partial x^4} + \frac{\partial^4 v(\xi_2, t_n)}{\partial x^4} \right) \end{aligned}$$

e então, se as derivadas parciais acima forem limitadas,

$$\tau_k^n = e_k^n + O(\Delta t, (\Delta x)^2)$$

onde  $e_k^n$  é um erro de arredondamento a ser desprezado. **Dessa forma, o erro de truncamento é de primeira ordem no tempo e segunda ordem no espaço.**

Nosso problema discretizado é:

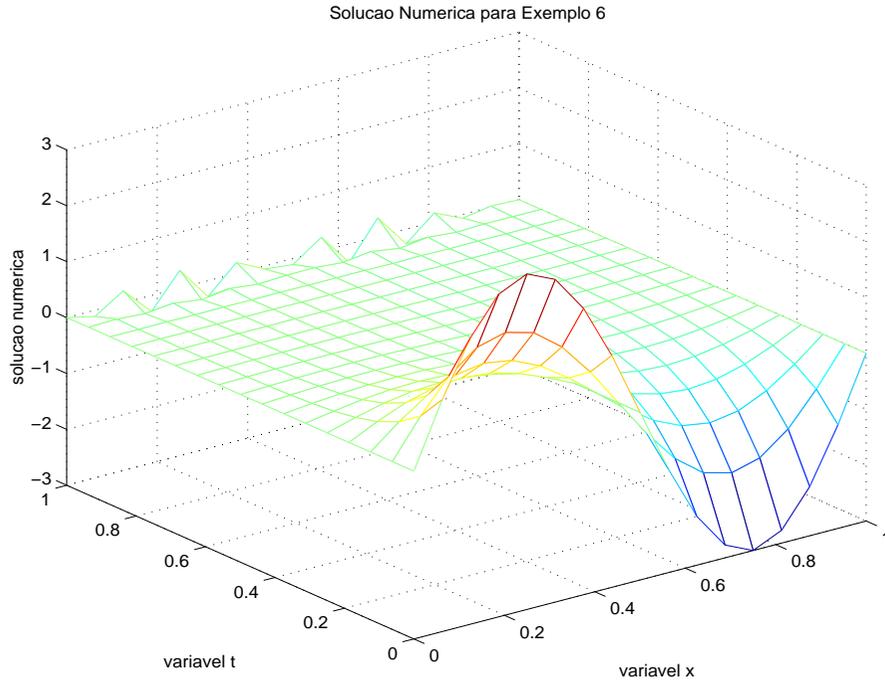
$$\begin{aligned} \frac{v_k^{n+1} - v_k^n}{\Delta t} &= \nu \cdot \frac{v_{k+1}^n - 2v_k^n + v_{k-1}^n}{(\Delta x)^2}, \quad k = 1, 2, \dots, K, \quad n = 0, 1, 2, \dots \\ v_k^0 &= f(k\Delta x), \quad k = 0, 1, 2, \dots, K \\ v_0^n &= a(n\Delta t), \quad v_K^n = b(n\Delta t), \quad n = 0, 1, 2, \dots \end{aligned}$$

e assim temos o esquema

$$\begin{aligned} v_k^0 &= f(k\Delta x) \quad , k = 1, 2, \dots, K-1 \\ v_0^n &= a(n\Delta t) \quad , v_K^n = b(n\Delta t) \quad , n = 0, 1, 2, \dots \\ v_k^{n+1} &= rv_{k+1}^n + (1-2r)v_k^n + rv_{k-1}^n \quad , k = 1, 2, \dots, K \quad , n = 0, 1, 2, \dots \end{aligned}$$

onde  $r = \nu\Delta t/(\Delta x)^2$ .

A solução numérica em Matlab, para  $N = K = 15$ ,  $\Delta x = \Delta t = 0.0625$ ,  $f(x) = 3\text{sen}(2\pi x)$  e  $a(t) = b(t) = 0$ , é mostrada abaixo:



### Exemplo 3.4 : O esquema Leapfrog para a equação de difusão

Para a solução numérica da equação de difusão, agora aproximaremos a derivada parcial  $v_t$  pela diferença finita central:

$$\frac{v_k^{n+1} - v_k^{n-1}}{2\Delta t}$$

que verifica

$$\frac{v_k^{n+1} - v_k^{n-1}}{2\Delta t} = \frac{\partial v}{\partial t}(x_k, t_n) + \frac{(\Delta t)^2}{3} \frac{\partial^3 v(x_k, \eta_1)}{\partial t^3}$$

Entretanto, derivada parcial  $v_{xx}$  continuará aproximada pela diferença central:

$$\frac{v_{k+1}^n - 2v_k^n + v_{k-1}^n}{(\Delta x)^2}$$

que verifica

$$\frac{v_{k+1}^n - 2v_k^n + v_{k-1}^n}{(\Delta x)^2} = \frac{\partial^2 v(x_k, t_n)}{\partial x^2} + \frac{(\Delta x)^2}{4!} \left( \frac{\partial^4 v(\xi_1, t_n)}{\partial x^4} + \frac{\partial^4 v(\xi_2, t_n)}{\partial x^4} \right).$$

O erro de truncamento desse método é

$$\begin{aligned} \tau_k^n &= v_t(x_k, t_n) - \nu v_{xx}(x_k, t_n) = \frac{v_k^{n+1} - v_k^{n-1}}{2\Delta t} - \frac{\nu}{(\Delta x)^2} (v_{k+1}^n - 2v_k^n + v_{k-1}^n) - \\ &\quad \frac{(\Delta t)^2}{3} \frac{\partial^3 v}{\partial t^3}(x_k, \eta_1) + \frac{\nu(\Delta x)^2}{4!} \left( \frac{\partial^4 v(\xi_1, t_n)}{\partial x^4} + \frac{\partial^4 v(\xi_2, t_n)}{\partial x^4} \right) \end{aligned}$$

e então, se as derivadas parciais acima forem limitadas,

$$\tau_k^n = e_k^n + O((\Delta t)^2, (\Delta x)^2)$$

onde  $e_k^n$  é um erro de arredondamento a ser desprezado. **Dessa forma, o erro de truncamento é de segunda ordem no tempo e no espaço.**

**Nosso método é agora explícito de passo múltiplo  $m = 2$  (3 estágios).**

**Uma dificuldade com esquemas de passo múltiplo é que eles normalmente requerem um número adicional de valores iniciais ou então devem ser inicializados por esquemas mais simples.**

No nosso exemplo, os valores no instante correspondendo a  $n = 1$  não podem ser calculados usando a aproximação central para  $v_t$ , uma vez que precisaríamos de dados correspondendo a  $n = -1$ .

Nestes casos, os dois remédios mais usados são :

- valores fantasmas correspondendo a  $n = -1$  são inseridos obedecendo algum critério traduzindo bom senso físico do problema;
- valores correspondendo a  $n = 1$  são calculados usando a aproximação clássica de primeira ordem para  $v_t$ .

Adotando a segunda estratégia acima, nosso problema discretizado é:

$$\begin{aligned} \frac{v_k^1 - v_k^0}{\Delta t} &= \nu \cdot \frac{v_{k+1}^0 - 2v_k^0 + v_{k-1}^0}{(\Delta x)^2}, \quad k = 1, 2, \dots, K \\ \frac{v_k^{n+1} - v_k^n}{2\Delta t} &= \nu \cdot \frac{v_{k+1}^n - 2v_k^n + v_{k-1}^n}{(\Delta x)^2}, \quad k = 1, 2, \dots, K, \quad n = 1, 2, \dots \\ v_k^0 &= f(k\Delta x), \quad k = 0, 1, 2, \dots, K \\ v_0^n &= a(n\Delta t), \quad v_K^n = b(n\Delta t), \quad n = 0, 1, 2, \dots \end{aligned}$$

e assim temos o esquema numérico (**Leapfrog**)

$$\begin{aligned} v_k^0 &= f(k\Delta x), \quad k = 1, 2, \dots, K-1 \\ v_0^n &= a(n\Delta t), \quad v_K^n = b(n\Delta t), \quad n = 0, 1, 2, \dots \\ v_k^1 &= rv_{k+1}^0 + (1-2r)v_k^0 + rv_{k-1}^0, \quad k = 1, 2, \dots, K, \quad n = 0, 1, 2, \dots \\ v_k^{n+1} &= v_k^{n-1} + 2r(v_{k+1}^n - 2v_k^n + v_{k-1}^n), \quad k = 1, 2, \dots, K, \quad n = 0, 1, 2, \dots \end{aligned}$$

onde novamente  $r = \nu\Delta t/(\Delta x)^2$ .

### Exemplo 3.5 : Escolha inteligente dos parâmetros $\Delta t$ e $\Delta x$ .

Ainda no contexto do problem de difusão, analisaremos as fórmulas de aproximação das quantidades  $v_t$  e  $v_{xx}$  usadas no Exemplo 3.3 com mais detalhe:

$$\frac{v_k^{n+1} - v_k^n}{\Delta t} = \frac{\partial v}{\partial t}(x_k, t_n) + \frac{\Delta t}{2} \frac{\partial^2 v(x_k, t_n)}{\partial t^2} + O((\Delta t)^2)$$

ao passo que

$$\frac{v_{k+1}^n - 2v_k^n + v_{k-1}^n}{(\Delta x)^2} = \frac{\partial^2 v(x_k, t_n)}{\partial x^2} + \frac{(\Delta x)^2}{12} \frac{\partial^4 v(x_k, t_n)}{\partial x^4} + O((\Delta x)^4).$$

Dessa forma, o erro de truncamento do esquema do Exemplo 3.3 é

$$\begin{aligned} \tau_k^n &= v_t(x_k, t_n) - \nu v_{xx}(x_k, t_n) = \frac{v_k^{n+1} - v_k^n}{\Delta t} - \frac{\nu}{(\Delta x)^2} (v_{k+1}^n - 2v_k^n + v_{k-1}^n) - \\ &\quad \frac{\Delta t}{2} \frac{\partial^2 v}{\partial t^2}(x_k, t_n) + \frac{\nu(\Delta x)^2}{12} \frac{\partial^4 v(x_k, t_n)}{\partial t^4} + O((\Delta t)^2, (\Delta x)^4) \end{aligned}$$

e assim, desprezando erros de arredondamento,

$$\tau_k^n = -\frac{\Delta t}{2} \frac{\partial^2 v}{\partial t^2}(x_k, t_n) + \frac{\nu(\Delta x)^2}{12} \frac{\partial^4 v(x_k, t_n)}{\partial t^4} + O((\Delta t)^2, (\Delta x)^4)$$

Entretanto, se assumirmos que  $v_t(x, t)$  e  $v_{xx}(x, t)$  são funções contínuas, então

$$\frac{\partial^2 v(x_k, t_n)}{\partial t^2} = \frac{\partial}{\partial t} \left( \nu \frac{\partial^2 v(x_k, t_n)}{\partial x^2} \right) = \nu \frac{\partial^2}{\partial x^2} \left( \frac{\partial v(x_k, t_n)}{\partial t} \right)$$

e assim

$$\frac{\partial^2 v(x_k, t_n)}{\partial t^2} = \nu^2 \frac{\partial^4 v(x_k, t_n)}{\partial x^4}.$$

Dessa forma

$$\tau_k^n = \frac{\nu}{2} \left( \frac{(\Delta x)^2}{6} - \nu\Delta t \right) \frac{\partial^4 v(x_k, t_n)}{\partial x^4} + O((\Delta t)^2, (\Delta x)^4).$$

**Conclusão :** se tomarmos  $\Delta t, \Delta x$  tais que

$$\frac{\Delta t}{(\Delta x)^2} = \frac{1}{6\nu}$$

e se a quarta derivada espacial for limitada, então o erro de truncamento do esquema do Exemplo 3.3 é de segunda ordem no tempo e quarta ordem no espaço.

**Introdução 3.1 :** *Tratamento numérico de condições de contorno; condição de Neumann*

No problema de difusão visto anteriormente, ressaltamos suas condições de contorno

$$\begin{aligned} v(0, t) &= a(t), \quad t \geq 0 \\ v(1, t) &= b(t), \quad t \geq 0. \end{aligned}$$

Essas condições de contorno são chamadas de **condições de Dirichlet**, sendo as de mais simples implementação numérica.

Muito frequentemente, para uma barra representada pelo segmento  $\{x : 0 \leq x \leq L\}$ , o problema físico nos dá condições de contorno do tipo

$$\begin{aligned} v_x(0, t) &= a(t) \\ v_x(L, t) &= b(t) \end{aligned}$$

chamadas **condições de contorno de Neumann** e que traduzem um controle do fluxo de calor nas extremidades da barra.

Entretanto, condições de Neumann não podem ser exatamente implementadas e conseqüentemente satisfeitas ao computador. Se consideramos, por exemplo, a condição  $v_x(0, t) = a(t)$ , vemos que uma abordagem bastante natural é usarmos

$$\frac{v_1^n - v_0^n}{\Delta x} = a(n\Delta t), \quad n = 1, 2, 3, \dots$$

e então teríamos  $v_0^n = v_1^n - \Delta x a(n\Delta t)$ ,  $n = 1, 2, 3, \dots$ . Essa expressão ainda não permite o cálculo de  $v_0^n$ , mas permite a eliminação de  $v_0^n$  das demais equações que compõem o esquema numérico.

**Problema:** pode mostrar-se que o erro de truncamento da solução será também determinado pela ordem da aproximação das condições de contorno.

Dessa forma, como essa aproximação é de primeira ordem no espaço, mesmo usando aproximações mais exatas (maior ordem) na equação diferencial parcial, a solução ao problema de contorno ainda teria um erro de truncamento de primeira ordem no espaço.

**Possível saída:** aumentar a ordem da aproximação das condições de contorno.

Assim, introduzimos os valores  $v_{-1}^n$  e usamos então a aproximação

$$\frac{v_1^n - v_{-1}^n}{2\Delta x} = a(n\Delta t), n = 1, 2, 3, \dots$$

e então  $v_1^n = v_{-1}^n + 2\Delta x a(n\Delta t)$ ,  $n = 1, 2, \dots$ . Essa expressão será usada na equação

$$\frac{v_0^{n+1} - v_0^n}{\Delta t} = \nu \frac{v_1^n - 2v_0^n + v_{-1}^n}{(\Delta x)^2}$$

obtida da aproximação da equação diferencial quando  $k = 0$ .

Dessa forma, temos

$$\begin{aligned} \frac{v_0^{n+1} - v_0^n}{\Delta t} &= \nu \frac{v_1^n - 2v_0^n + v_{-1}^n - 2\Delta x a(n\Delta t)}{(\Delta x)^2} \\ \implies v_0^{n+1} &= v_0^n + \frac{2\nu(v_1^n - v_0^n)}{(\Delta x)^2} - 2\nu\Delta x a(n\Delta t) \end{aligned}$$

e essa última equação permitirá calcular  $v_0^n$ ,  $n = 1, 2, 3, \dots$

**Definição 3.2** Diferenças finitas na variável espacial  $x$  serão denotadas pela função  $\delta(\cdot)$ , da seguinte forma:

$$\begin{aligned} \delta_+(v_k^n) &= v_{k+1}^n - v_k^n, \quad \delta_-(v_k^n) = v_k^n - v_{k-1}^n \\ \delta^1(v_k^n) &= v_{k+1}^n - v_{k-1}^n, \quad \delta^2(v_k^n) = v_{k+1}^n - 2v_k^n + v_{k-1}^n \end{aligned}$$

Dessa forma, o **esquema de Euler** para  $v_t = \nu v_{xx}$  pode ser descrito por

$$\frac{v_k^{n+1} - v_k^n}{\Delta t} = \frac{\nu}{(\Delta x)^2} \delta^2(v_k^n) \implies v_k^{n+1} = v_k^n + \frac{\nu\Delta t}{(\Delta x)^2} \delta^2(v_k^n),$$

ao passo que o **esquema Leapfrog** para  $v_t = \nu v_{xx}$  pode ser descrito por

$$\frac{v_k^{n+1} - v_k^{n-1}}{2\Delta t} = \frac{\nu\delta^2(v_k^n)}{(\Delta x)^2} \implies v_k^{n+1} = v_k^{n-1} + 2\frac{\nu\Delta t}{(\Delta x)^2} \delta^2(v_k^n).$$

**Definição 3.3** O método de Crank-Nicolson para  $v_t = \nu v_{xx}$  é descrito por

$$\frac{v_k^{n+1} - v_k^n}{\Delta t} = \frac{\nu}{(\Delta x)^2} \frac{\delta^2(v_k^{n+1}) + \delta^2(v_k^n)}{2}.$$

**Exemplo 3.6** : O esquema implícito de Crank-Nicolson para o problema de difusão

Dessa forma,  $v_t = \nu v_{xx}$  é discretizado via:

$$v_k^{n+1} - \frac{\nu\Delta t}{2(\Delta x)^2} \delta^2 v_k^{n+1} = v_k^n + \frac{\nu\Delta t \delta^2(v_k^n)}{2(\Delta x)^2}$$

e vemos que esse esquema é **implícito** de **2 estágios**.

Nosso problema discretizado é então :

$$\begin{aligned} v_k^{n+1} - \frac{r}{2} \delta^2(v_k^{n+1}) &= v_k^n + \frac{r}{2} \delta^2(v_k^n), \quad k = 1, 2, \dots, K; n = 0, 1, 2, \dots \\ v_k^0 &= f(k\Delta x), \quad k = 0, 1, 2, \dots, K \\ v_0^n &= a_n = a(n\Delta t), v_K^n = b_n = b(n\Delta t), \quad n = 0, 1, 2, 3, \dots \end{aligned}$$

A primeira equação é então considerada em três casos:

(A)  $k = 1$ :

$$(1+r)v_1^{n+1} - \frac{r}{2}v_2^{n+1} = \frac{r}{2}a_n + (1-r)v_1^n + \frac{r}{2}v_2^n + \frac{r}{2}a_{n+1}$$

(B)  $k = 2, 3, 4, \dots, K-2$ :

$$-\frac{r}{2}v_{k-1}^{n+1} + (1+r)v_k^{n+1} - \frac{r}{2}v_{k+1}^{n+1} = \frac{r}{2}v_{k-1}^n + (1-r)v_k^n + \frac{r}{2}v_{k+1}^n.$$

(C)  $k = K-1$ :

$$-\frac{r}{2}v_{K-2}^{n+1} + (1+r)v_{K-1}^{n+1} = \frac{r}{2}v_{K-2}^n + (1-r)v_{K-1}^n + \frac{r}{2}b_n + \frac{r}{2}b_{n+1}$$

o que significa que temos  $K-1$  equações nas  $(K-1)$  variáveis  $v_1^{n+1}, v_2^{n+1}, \dots, v_{K-1}^{n+1}$ .

Matricialmente, temos  $A\mathbf{v}^{n+1} = \mathbf{z}^n$  onde

$$A = \begin{bmatrix} 1+r & -r/2 & 0 & \dots & & & & \\ -r/2 & 1+r & -r/2 & 0 & \dots & & & \\ 0 & -r/2 & 1+r & -r/2 & 0 & \dots & & \\ & 0 & r/2 & \dots & \dots & \dots & & \\ & & 0 & \dots & 1+r & -r/2 & & \\ & & & \dots & -r/2 & 1+r & & \end{bmatrix}$$

$$\mathbf{v}^{n+1} = \begin{bmatrix} v_1^{n+1} \\ v_2^{n+1} \\ \dots \\ v_{K-2}^{n+1} \\ v_{K-1}^{n+1} \end{bmatrix}, \quad \mathbf{z}^n = \begin{bmatrix} \frac{r}{2}a_n + (1-r)v_1^n + \frac{r}{2}v_2^n + \frac{r}{2}a_{n+1} \\ \dots \\ \frac{r}{2}v_{k-1}^n + (1-r)v_k^n + \frac{r}{2}v_{k+1}^n \\ \dots \\ \frac{r}{2}v_{K-2}^n + (1-r)v_{K-1}^n + \frac{r}{2}b_n + \frac{r}{2}b_{n+1} \end{bmatrix}$$

A matriz  $A$  acima é uma matriz constante (não muda de um passo para outro) esparsa do tipo tridiagonal. Dessa forma, seu armazenamento, fatorização e a solução de  $Av^{n+1} = \mathbf{z}^n$  podem ser feitos usando métodos especializados para esse tipo de estrutura.

**Definição 3.4** Nos chamados métodos (esquemas) numéricos de **passo múltiplo**, os valores nos  $m$  instantes anteriores  $n, n-1, n-2, \dots, n-m+1$  são usados na determinação dos valores do  $(n+1)$ -ésimo instante, onde  $m \in \mathbb{N}, m > 1$ . Neste caso, dizemos que

- o método tem  $m$  passos, ou então que
- o método tem  $m+1$  estágios.

Se acima tivermos  $m=1$ , então temos os chamados métodos (esquemas) numéricos de **passo simples** ou 2 estágios.

**Definição 3.5** Um método (esquema) numérico é dito **explícito** se os valores no  $(n+1)$ -ésimo instante são determinados explicitamente. Caso contrário, o método (esquema) numérico é dito ser **implícito**.

Um exemplo de esquema numérico explícito de passo simples é o apresentado no Exemplo 5, onde os valores no  $(n+1)$ -ésimo instante são diretamente determinados apenas pelos valores do  $n$ -ésimo instante.

Para a equação diferencial ordinária  $y' = f(t, y)$ , um esquema de  $m$  passos é

$$\frac{y_{i+1} + \alpha_1 y_i + \alpha_2 y_{i-1} + \dots + \alpha_m y_{i+1-m}}{h} = \beta_0 f(t_{i+1}, y_{i+1}) + \dots + \beta_m f(t_{i+1-m}, y_{i+1-m}) \quad (45)$$

### 3.1 Métodos Implícitos ou Corretores de Adams

Os esquemas implícitos de Adams-Moulton (Corretores) são uma família de esquemas de  $m$  passos onde

$$\alpha_1 = -1, \quad \alpha_2 = \alpha_3 = \dots = \alpha_m = 0$$

e os coeficientes  $\beta_0, \dots, \beta_m$  são determinados de maneira a maximizar a ordem do erro de truncamento.

A tabela 6 mostra os valores desses coeficientes para  $m = 1, 2, 3$ .

Tabela 6: Parâmetros para os esquemas Corretores de Adams

$m$	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$T$
1	1/2	1/2			$O(h^2)$
2	5/12	8/12	-1/12		$O(h^3)$
3	9/24	19/24	-5/24	1/24	$O(h^4)$

Aqui o parâmetro de discretização é denotado por  $h$ . Por exemplo, o esquema corretor de 2 passos de Adams é

$$\frac{y_{i+1} - y_i}{h} = \beta_0 f(t_{i+1}, y_{i+1}) + \beta_1 f(t_i, y_i) + \beta_2 f(t_{i-1}, y_{i-1}).$$

Neste caso, pode-se mostrar que o erro de truncamento é dado por:

$$e_{i,h} = \frac{y_{i+1} - y_i}{h} - \beta_0 f(t_{i+1}, y_{i+1}) - \beta_1 f(t_i, y_i) - \beta_2 f(t_{i-1}, y_{i-1}) =$$

$$(1 - \beta_0 - \beta_1 - \beta_2)y'(t_i) + (1 - 2\beta_0 + 2\beta_2)\frac{hy''(t_i)}{2} + (1 - 3\beta_0 - 3\beta_2)\frac{h^2y^{(3)}(t_i)}{6} + (1 - 4\beta_0 + 4\beta_2)\frac{h^3y^{(4)}(t_i)}{24} + O(h^4)$$

Dessa forma, os parâmetros  $\beta_0, \beta_1$  e  $\beta_2$  devem satisfazer

$$\begin{aligned} 1 - \beta_0 - \beta_1 - \beta_2 &= 0 \\ 1 - 2\beta_0 + 2\beta_2 &= 0 \\ 1 - 3\beta_0 - 3\beta_2 &= 0 \end{aligned}$$

o que resulta

$$\beta_0 = \frac{5}{12}, \quad \beta_1 = \frac{8}{12}, \quad \beta_2 = \frac{-1}{12}.$$

Como  $1 - 4\beta_0 + 4\beta_2 \neq 0$ , o erro de truncamento é de ordem 3.

**Dificuldade com esquemas de passo múltiplo:** eles normalmente requerem um número adicional de valores para sua inicialização.

**Exemplo 3.7** : encontrar solução numérica para o PVI

$$\begin{cases} \frac{dy}{dt} = \frac{y}{1+t^p} \\ y(0) = 1 \end{cases} .$$

no intervalo  $[0, 20]$ , para  $p = 1.5$ , usando o método Corretor de Adams de ordem 2 e  $h = 0.1$ .

Temos  $f(t, y) = y/(1+t^{3/2})$ , e assim

$$\begin{cases} \frac{y_{i+1} - y_i}{h} = \frac{1}{2} \frac{y_i}{1+t_i^{3/2}} + \frac{1}{2} \frac{y_{i+1}}{1+t_{i+1}^{3/2}}, i = 0, 1, 2, \dots, n-1 \\ y_0 = 1 \end{cases} .$$

e re-escrevemos a primeira equação:

$$\left(1 - \frac{h}{2(1+t_{i+1}^{3/2})}\right) y_{i+1} = \left(1 + \frac{h}{2(1+t_i^{3/2})}\right) y_i$$

o que implica no esquema numérico

$$\begin{cases} y_0 = 1 \\ y_{i+1} = \left(1 - \frac{h}{2(1+t_{i+1}^{3/2})}\right)^{-1} \left(1 + \frac{h}{2(1+t_i^{3/2})}\right) y_i, i = 0, 1, 2, \dots, n-1 \end{cases}$$

e a sequência de comandos em *Scilab*

```
--> h = 0.1; t=[0:h:20]; n = 20/h;
--> p=1.5; y=0*t; y(1)=1;
--> for i=1:n
-->   ti = (i-1)*h; ti1=ti+h;
-->   di = 1 + h/(2+2*ti*sqrt(ti)); ui= 1 - h/(2+2*ti1*sqrt(ti1));
-->   y(i+1) = di*y(i)/ui;
--> end
```

produz a Figura 2.

### 3.2 Métodos Explícitos ou Preditores de Adams

Os esquemas explícitos de Adams-Bashforth (Preditores) são uma família de esquemas de  $m$  passos onde

$$\alpha_1 = -1, \alpha_2 = \alpha_3 = \dots = \alpha_m = 0$$

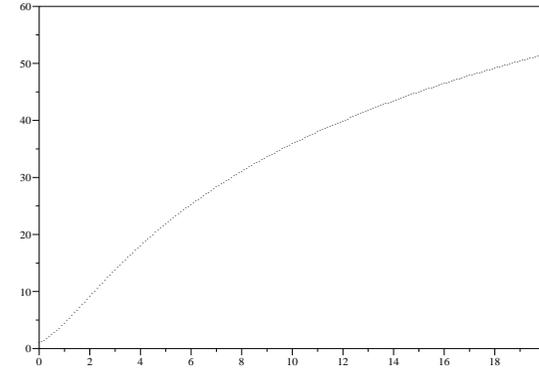


Figura 2: Solução numérica do exemplo 3.7 via esquema corretor de ordem 2.

Tabela 7: Parâmetros para os esquemas Previsores de Adams

$m$	$\beta_0$	$\beta_1$	$\beta_2$	$\beta_3$	$\beta_4$	$T$
1	0	1				$O(h)$
2	0	$3/2$	$-1/2$			$O(h^2)$
3	0	$23/12$	$-16/12$	$5/12$		$O(h^3)$
4	0	$55/24$	$-59/24$	$37/24$	$-9/24$	$O(h^4)$

e os coeficientes  $\beta_0, \dots, \beta_m$  são determinados de maneira a maximizar a ordem do erro de truncamento sob a restrição  $\beta_0 = 0$ .

A Tabela 7 mostra os valores desses coeficientes para  $m = 1, 2, 3, 4$ .

**Exemplo 3.8** : encontrar solução numérica para o PVI

$$\begin{cases} \frac{dy}{dt} = \frac{\cos(t)}{t+y^2} \\ y(1) = 1/2 \end{cases} .$$

no intervalo  $[1, 10]$ , usando o método Predictor de ordem 3 de Adams e  $h = 0.1$ .

Temos  $f(t, y) = \cos(t)/(t+y^2)$ , e assim

$$\begin{cases} \frac{y_{i+1} - y_i}{h} = \frac{23 \cos(t_i)}{12 t_i + y_i^2} - \frac{16 \cos(t_{i-1})}{12 t_{i-1} + y_{i-1}^2} + \frac{5 \cos(t_{i-2})}{12 t_{i-2} + y_{i-2}^2}, & i = 2, \dots, n-1 \\ \frac{y_2 - y_0}{2h} = \frac{\cos(t_1)}{t_1 + y_1^2} \\ \frac{y_1 - y_0}{h} = \frac{\cos(t_0)}{t_0 + y_0^2} \\ y_0 = 1/2 \end{cases}$$

e re-escrevemos a primeira equação:

$$y_{i+1} = y_i + \frac{23h \cos(t_i)}{12 t_i + y_i^2} - \frac{16h \cos(t_{i-1})}{12 t_{i-1} + y_{i-1}^2} + \frac{5h \cos(t_{i-2})}{12 t_{i-2} + y_{i-2}^2}$$

o que implica no esquema numérico

$$\begin{cases} y_0 = 1/2 \\ y_1 = y_0 + h \cos(t_0)/(t_0 + y_0^2) \\ y_2 = y_0 + 2h \cos(t_1)/(t_1 + y_1^2) \\ y_{i+1} = y_i + \frac{23h \cos(t_i)}{12 t_i + y_i^2} - \frac{16h \cos(t_{i-1})}{12 t_{i-1} + y_{i-1}^2} + \frac{5h \cos(t_{i-2})}{12 t_{i-2} + y_{i-2}^2}, & i = 2, \dots, n-1 \end{cases}$$

A sequência de comandos em *Scilab*

```
--> h = 0.1; t=[1:h:10];n = (10-1)/h;
--> y=0*t;y(1)=0.5;
--> function u=f(t,y)
--> u = cos(t)/(t + y*y); endfunction
--> y(2) = y(1) + h*f(t(1),y(1));
--> y(3) = y(1) + 2*h*f(t(2),y(2));
--> for i=3:n
--> y(i+1)= y(i) + 23*h*f(t(i),y(i))/12 - 16*h*f(t(i-1),y(i-1))/12 +
--> 5*h*f(t(i-2),y(i-2))/12;
--> end
--> plot2d(x,y,-1)
```

produzem a Figura 3.

### 3.3 Métodos Preditor-corretor de Adams

Esses métodos são combinações dos métodos explícitos (preditores) e implícitos (corretores) de Adams.

Estratégia: primeiramente, aplicamos o esquema predictor para obter uma aproximação  $y^p$  de  $y_{i+1}$ , após, usamos  $y^p$  no lugar de  $y_{i+1}$  no lado recorrente da equação.

$$y_i \xrightarrow{P} y^p \xrightarrow{C} y_{i+1}$$

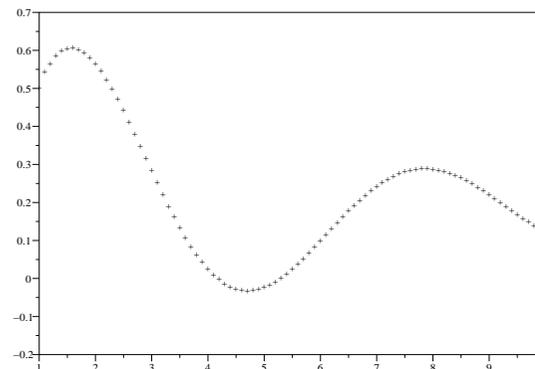


Figura 3: Solução numérica do exemplo 3.8 via esquema predictor de ordem 3.

**Exemplo 3.9 :** Resolver numericamente o PVI

$$\begin{cases} dy/dt = y(4 - y) \\ y(0) = 1/10 \end{cases}$$

no intervalo  $[0, 5]$ , usando o método Preditor-corretor de Adams de ordem 2, e  $N = 20$ .

Propomos o esquema numérico

$$y_0 = 0.1, y_1 = y_0 + \Delta t y_0 (4 - y_0)$$

$$\begin{cases} y_p = y_i + \frac{3\Delta t y_i (4 - y_i)}{2} - \frac{\Delta t y_{i-1} (4 - y_{i-1})}{2} \\ y_{i+1} = y_i + \frac{\Delta t y_p (4 - y_p)}{2} + \frac{\Delta t y_i (4 - y_i)}{2} \end{cases}$$

para  $i = 1, 2, 3, \dots, N - 1$

A sequência de comandos em *Scilab*

```
--> N = 20; h = (5-0)/N;
--> t=[0:h:5];
--> function u=f(t,y)
--> u = y*(4-y);endfunction
--> y=0*t; y(1) = 0.1; y(2) = y(1) + h*f( t(1), y(1) );
--> for i=2:N
--> yp = y(i) + 3*h*f(t(i),y(i))/2 - h*f(t(i-1),y(i-1))/2;
```

```

--> y(i+1)=y(i) + h*f(t(i+1),yp)/2 + h*f(t(i),y(i))/2;
--> end
--> ye=4*ones(t)./(ones(t) + 39*exp(-4*t));
--> [t y ye]
--> plot2d([t t],[y ye],[-2 1])

```

produz a Tabela 8 e a Figura 4, que mostra o resultado da avaliação desse esquema numérico:

Tabela 8: Aplicação do esquema Previsor-Corretor de Adams

$t_i$	$y_i$	exata	$t_i$	$y_i$	exata
0.	0.1	0.1	2.75	4.0133827	3.9973962
0.25	0.1975	0.2606322	3.	4.0036113	3.9990417
0.5	0.4834080	0.6371379	3.25	3.9993494	3.9996474
0.75	1.0775639	1.35976	3.5	3.9986079	3.9998703
1.	1.9709386	2.3333004	3.75	3.9991186	3.9999523
1.25	2.8258465	3.1676145	4.	3.9996869	3.9999824
1.5	3.4324052	3.6474009	4.25	3.9999855	3.9999935
1.75	3.7938897	3.8626317	4.5	4.0000674	3.9999976
2.	3.9667782	3.9483437	4.75	4.0000542	3.9999991
2.25	4.0231938	3.9808403	5.	4.0000238	3.9999997
2.5	4.0255656	3.9929301			

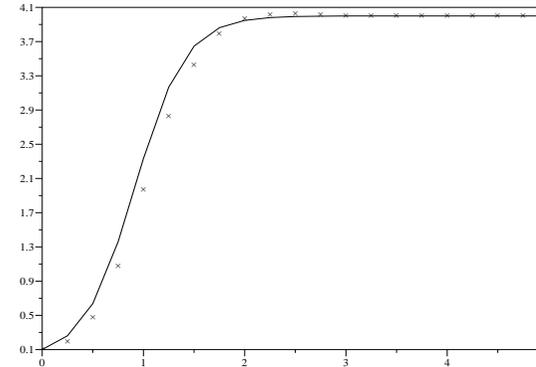


Figura 4: Solução numérica do exemplo 3.9 via esquema previsor-corretor de ordem 2 de Adams.

**Introdução 3.2** : Métodos de Runge-Kutta para solução de EDO de primeira ordem

Estes são provavelmente os métodos de alta-ordem mais fáceis de programar. São métodos de passo simples que, ao contrário dos métodos de passo múltiplo, são auto-inicializáveis e não enfrentam dificuldade ao variarmos o passo  $h$ . Por outro lado, métodos de Runge-Kutta requerem muitas avaliações da função  $f$  por passo.

• **Métodos de Runge-Kutta**

Esquema clássico de Runge-Kutta de quarta ordem:

$$\begin{aligned}
 k_1 &= f(t_n, y_n) \\
 k_2 &= f\left(t_n + \frac{h}{2}, y_n + \frac{hk_1}{2}\right) \\
 k_3 &= f\left(t_n + \frac{h}{2}, y_n + \frac{hk_2}{2}\right) \\
 k_4 &= f(t_n + h, y_n + hk_3) \\
 y_{n+1} &= y_n + \frac{h(k_1 + 2k_2 + 2k_3 + k_4)}{6}
 \end{aligned}$$

• **Método RKF45**

O método de Runge-Kutta-Fehlberg para solução numérica de

$$\frac{dy}{dt} = f(t, y)$$

possui um erro de truncamento de quarta ordem, além de fornecer uma estimativa de quinta ordem desse erro de truncamento.

Novamente o parâmetro de discretização é denotado por  $h$ .

$$\begin{aligned}
k_1 &= hf(t_n, y_n) \\
k_2 &= hf\left(t_n + \frac{h}{4}, y_n + \frac{k_1}{4}\right) \\
k_3 &= hf\left(t_n + \frac{3h}{8}, y_n + \frac{3k_1}{32} + \frac{9k_2}{32}\right) \\
k_4 &= hf\left(t_n + \frac{12h}{13}, y_n + \frac{1932k_1}{2197} - \frac{7200k_2}{2197} + \frac{7296k_3}{2197}\right) \\
k_5 &= hf\left(t_n + h, y_n + \frac{439k_1}{216} - 8k_2 + \frac{3680k_3}{513} - \frac{845k_4}{4104}\right) \\
k_6 &= hf\left(t_n + \frac{h}{2}, y_n - \frac{8k_1}{27} + 2k_2 - \frac{3544k_3}{2565} + \frac{1859k_4}{4104} - \frac{11k_5}{40}\right) \\
y_{n+1} &= y_n + \frac{25k_1}{216} + \frac{1408k_3}{2565} + \frac{2197k_4}{4104} - \frac{k_5}{5} \\
e_n &= \frac{1}{h} \left| \frac{k_1}{360} - \frac{128k_3}{4275} - \frac{2197k_4}{75240} + \frac{k_5}{50} + \frac{2k_6}{55} \right|
\end{aligned}$$

## 4 Métodos de discretização parcial; o Método das Linhas

Os métodos numéricos vistos anteriormente são de **discretização total**, pois tanto as **variáveis de espaço** ( $x$ ) quanto a variável temporal ( $t$ ) são discretizadas.

**Introdução 4.1** : *Método das Linhas.*

O Método das Linhas é um método de discretização parcial, onde apenas as variáveis espaciais são discretizadas. Essa estratégia, dá origem a um sistema de equações diferenciais ordinárias na variável  $t$ . Esse sistema de equações é então resolvido usando-se métodos clássicos disponíveis na literatura.

**Exemplo 4.1** : *Método das Linhas para o problema de difusão*

Após a discretização

$$x_k = k\Delta x, v_k(t) = v(x_k, t); k = 0, 1, 2, \dots, K$$

e a escolha de uma aproximação de diferenças finitas para a derivada parcial  $v_{xx}$ , a equação parcial  $v_t = \nu v_{xx}$  escreve-se

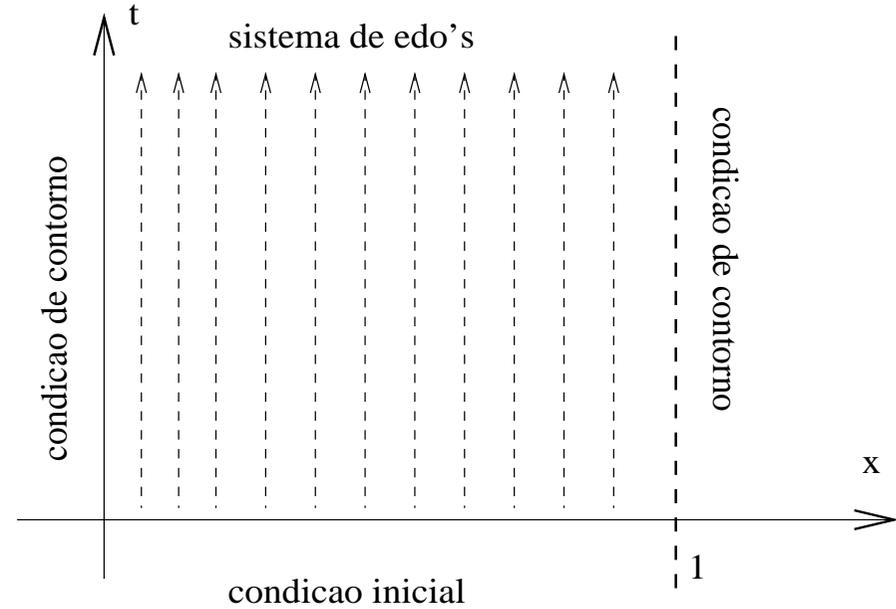
$$\frac{dv_k}{dt} = f(v_k, x_k, t).$$

Se a derivada parcial  $v_{xx}$  for aproximada pela diferença

$$\frac{v_{k+1}(t) - 2v_k(t) + v_{k-1}(t)}{(\Delta x)^2} = \frac{\delta^2(v_k)}{(\Delta x)^2}$$

então temos

$$f(v_k, x_k, t) = \frac{\delta^2(v_k)}{(\Delta x)^2}.$$



Assim, temos os sistema de equações diferenciais ordinárias

$$\frac{d}{dt} \begin{bmatrix} v_1 \\ v_2 \\ \dots \\ v_{K-2} \\ v_{K-1} \end{bmatrix} = \begin{bmatrix} \rho(v_2 - 2v_1) + \rho a(t) \\ \rho(v_3 - 2v_2 + v_1) \\ \dots \\ \rho(v_{K-1} - 2v_{K-2} + v_{K-3}) \\ \rho(-2v_{K-1} + v_{K-3}) + \rho b(t). \end{bmatrix}$$

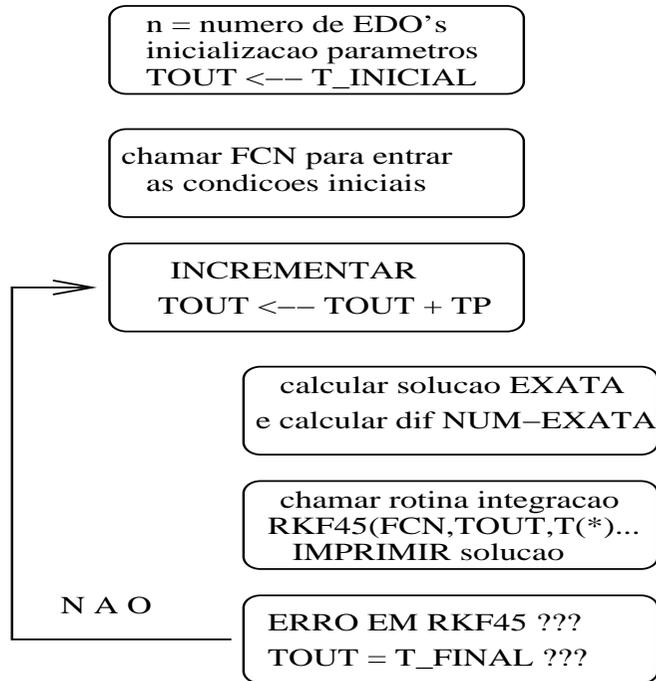
com a condição inicial

$$\begin{aligned}
v_1(0) &= f(\Delta x) \\
v_2(0) &= f(2\Delta x) \\
\dots & \\
v_{K-1}(0) &= f((K-1)\Delta x).
\end{aligned}$$

**Implementação do Método das Linhas: Vários métodos clássicos para solução de sistemas de equações  $y' = f(y, t)$  podem ser usados.**

Entre eles:

- Método de Euler, Trapézio
- Métodos de Adams-Moulton e Adams-Bashforth
- Métodos Preditor-Corretor de Adams
- Método de Runge-Kutta, Runge-Kutta-Fehlberg



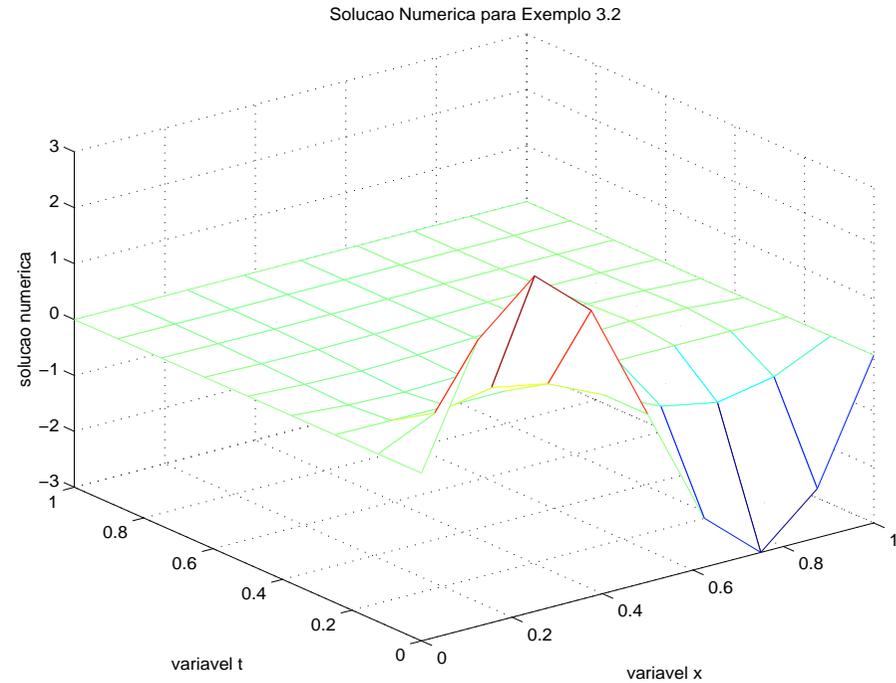
onde  $f(x) = 3\text{sen}(2\pi x)$  e  $a(t) = b(t) = 0$ .

O Método das Linhas é então aplicado e o sistema de equações diferenciais ordinárias é resolvido usando o método preditor-corretor de segunda ordem de Adams:

$$y^p = y_n + \frac{3hf(t_n, y_n)}{2} + \frac{hf(t_{n-1}, y_{n-1})}{2}$$

$$y_{n+1} = y_n + \frac{hf(t_{n+1}, y^p)}{2} + \frac{hf(t_n, y_n)}{2}.$$

A solução numérica em Matlab, para  $N = K = 8$ ,  $\Delta x = \Delta t = 0.125$ , é mostrada abaixo



**Exemplo 4.2 :** Implementação do Método das Linhas para problema de difusão .

Considere o problema de condições de contorno

$$v_t = \nu v_{xx}, \quad x \in (0, 1), \quad t > 0$$

$$v(x, 0) = f(x), \quad x \in [0, 1]$$

$$v(0, t) = a(t), \quad v(1, t) = b(t), \quad t \geq 0$$

## 5 Revisão de pré-requisitos

### 5.1 Revisão de análise real e funcional

Em espaços de dimensão finita, usaremos as normas

$$\|u\|_2 = \sqrt{\sum_{k=1}^N |u_k|^2} \quad , \quad \|u\|_{2,\Delta x} = \sqrt{\sum_{k=1}^N |u_k|^2 \Delta x} \quad , \quad \|u\|_\infty = \max_{1 \leq k \leq N} |u_k|$$

Em espaços de seqüências infinito-dimensionais, como os espaços  $\ell_p$ :

$$\ell_p = \left\{ u = (\dots, -u_{-1}, u_0, u_1, \dots) : \left( \sum_{k=-\infty}^{\infty} |u_k|^p \right)^{1/p} < \infty \right\}$$

com norma

$$\|u\|_p = \left( \sum_{k=-\infty}^{\infty} |u_k|^p \right)^{1/p}$$

e em particular, quando  $p = 2$ ,

$$\ell_2 = \left\{ u = (\dots, u_{-1}, u_0, u_1, \dots) : \sum_{k=-\infty}^{\infty} |u_k|^2 < \infty \right\}$$

$$\|u\|_2 = \sqrt{\sum_{k=-\infty}^{\infty} |u_k|^2}$$

e **norma-energia**

$$\|u\|_{2,\Delta x} = \sqrt{\sum_{k=-\infty}^{\infty} |u_k|^2 \Delta x} .$$

Quando  $p = \infty$  temos os espaço das seqüências limitadas

$$\ell_\infty = \left\{ u = (\dots, u_{-1}, u_0, u_1, \dots) : \sup_{-\infty \leq k \leq \infty} |u_k| < \infty \right\}$$

com norma

$$\|u\|_\infty = \sup_{-\infty \leq k \leq \infty} |u_k| .$$

Além disso, no contexto de transformadas integrais usaremos o espaço de funções complexas

$$L_2(R) = \left\{ v : R \rightarrow \mathbb{C} : \int_R |v(x)|^2 dx < \infty \right\}$$

de funções de quadrado integrável a Lebesgue, e norma

$$\|v\|_2 = \sqrt{\int_R |v(x)|^2 dx} .$$

Para vetores ou seqüências bi-indexadas em  $\mathbb{R}^{N \times N}$

$$\|\{u_{jk}\}\|_{2,\Delta x} = \sqrt{\sum_{j=1}^{N_x} \sum_{k=1}^{N_y} |u_{jk}|^2 \Delta x \Delta y}$$

$$\|\{u_{jk}\}\|_{2,\Delta x} = \sqrt{\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} |u_{jk}|^2 \Delta x \Delta y}$$

e para o espaço  $L_2(R)$  bi-dimensional

$$\|v\|_2 = \sqrt{\int_{R \times R} |v(x, y)|^2 dx dy} .$$

Para seqüências de vetores

$$\mathbf{u} = (\dots, u_{-1}, u_0, u_1, \dots) \quad , \quad u_j \in \mathbb{R}^k \text{ ou } \mathbb{C}^k$$

$$\|\mathbf{u}\|_{2,\Delta x} = \sqrt{\sum_{j=-\infty}^{\infty} \|u_j\|_2^2 \Delta x}$$

e para funções vetoriais  $v : R \rightarrow \mathbb{C}^N$ :

$$\|\mathbf{v}\|_2 = \sqrt{\int_R \|v(x)\|_2^2 dx} .$$

## 5.2 Revisão de álgebra matricial

Operadores  $A : X \rightarrow X$ , onde  $X$  é um espaço linear com norma  $\|\cdot\|$ .

**Exemplo:**  $X$  tem dimensão finita e  $A$  é um operador linear

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \xrightarrow{A} \begin{bmatrix} x_1 - 2x_2 + x_3 \\ x_1 + x_3 \\ x_2 - x_3 \end{bmatrix}$$

com representação matricial

$$A = \begin{bmatrix} 1 & -2 & 1 \\ 1 & 0 & 1 \\ 0 & 1 & -1 \end{bmatrix}$$

onde toleraremos o abuso de notação :  $A(x) = Ax$ .

A norma de  $A$  induzida por  $\|\cdot\|$  é:

$$\|A\| = \sup_{\|x\| \leq 1} \|Ax\|.$$

**Lema 5.1** *Se  $A$  é uma matriz  $N \times N$  hermitiana ( $A = A^*$ ) então a norma espectral é definida por*

$$\|A\|_2 = \sigma(A)$$

onde  $\sigma(A)$  é o raio espectral de  $A$ , definido por

$$\sigma(A) = \max\{|\lambda| : \exists x \in \mathbb{C}^N, Ax = \lambda x\}.$$

**Definição 5.1** *Os valores singulares de uma matriz  $A$  são as raízes quadradas dos autovalores de  $A^T A$ .*

**Definição 5.2** *Se  $A$  é uma matriz complexa qualquer então definimos*

$$\|A\|_2 = \sqrt{\sigma(A^* A)}.$$

Assim, se  $A$  é uma matriz real qualquer

$$\|A\|_2 = \sqrt{\sigma(A^T A)}.$$

Dessa forma, a norma espectral de uma matriz real  $A$  é determinada pelo seu maior valor singular.

Como o cálculo dessa norma requer computação pesada, algumas alternativas podem ser consideradas.

**1. Matrizes especiais.** Algumas matrizes tem autovalores e autovetores conhecidos analiticamente, o que facilita bastante a análise numérica. Considere

$$T = \begin{bmatrix} b & c & \dots & & \\ a & b & c & \dots & \\ & \dots & \dots & \dots & \\ & & \dots & a & b \end{bmatrix}$$

de dimensão  $N \times N$ . Os autovalores e correspondentes autovetores são :

$$\lambda_j = b + 2c\sqrt{\frac{a}{c}} \cos\left(\frac{j\pi}{N+1}\right)$$

$$u_j(k) = 2\left(\sqrt{\frac{a}{c}}\right)^k \text{sen}\left(\frac{kj\pi}{N+1}\right), \quad k = 1, \dots, N$$

para  $j = 1, 2, \dots, N$ .

**2. Normas Matriciais Equivalentes.** Para vetores de  $n$  componentes, uma consequência da definição das normas vetoriais  $\|\cdot\|_p$  é

- $\|x\|_2 \leq \|x\|_1 \leq \sqrt{n}\|x\|_2$
- $\|x\|_\infty \leq \|x\|_2 \leq \sqrt{n}\|x\|_\infty$
- $\|x\|_\infty \leq \|x\|_1 \leq n\|x\|_\infty$

Uma consequência direta para estimação da norma matricial  $\|A\|_2$  (norma espectral) são as fórmulas

- $\frac{1}{\sqrt{n}}\|A\|_\infty \leq \|A\|_2 \leq \sqrt{n}\|A\|_\infty$
- $\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2$
- $\|A\|_2 \leq \sqrt{\|A\|_1 \|A\|_\infty}$

onde  $\|A\|_F$  é a chamada **norma Frobenius** de  $A$ , definida por

$$\|A\|_F = \sqrt{\sum_{i=1}^n \sum_{j=1}^n |a_{ij}|^2}$$

e que verifica  $\|A\|_F^2 = \text{tr}(A^T A)$  (traço matricial).

## 6 A noção de estabilidade de um esquema numérico

Um esquema de valor inicial é estável: pequenos erros nas condições iniciais correspondem a pequenos erros na solução

**Definição 6.1** *Um esquema de passo simples*

$$\mathbf{u}^{n+1} = Q\mathbf{u}^n, \quad n \geq 0$$

é **estável** com respeito à norma  $\|\cdot\|$  se existem constantes positivas  $\Delta x_0$  e  $\Delta t_0$  e constantes não negativas  $K$  e  $\beta$  tais que

$$\|\mathbf{u}^{n+1}\| \leq K e^{\beta t} \|\mathbf{u}^0\|, \quad n = 0, 1, 2, \dots$$

para  $0 \leq t = (n+1)\Delta t$ ,  $0 < \Delta x \leq \Delta x_0$  e  $0 < \Delta t \leq \Delta t_0$ .

**Lema 6.1** *O esquema acima é estável com respeito à norma  $\|\cdot\|$  se e somente se existem constantes  $K$  e  $\beta$  tais que*

$$\|Q^{n+1}\| \leq Ke^{\beta t}, n = 0, 1, 2, \dots$$

para  $0 \leq t = (n+1)\Delta t$ ,  $0 < \Delta x \leq \Delta x_0$  e  $0 < \Delta t \leq \Delta t_0$ .

**Demonstração :**

$\Leftarrow$  Supomos que  $\|Q^{n+1}\| \leq Ke^{\beta t}, n = 0, 1, \dots$ . Então

$$u^{n+1} = Qu^n = Q^2u^{n-1} = \dots = Q^{n+1}u^0$$

e assim

$$\|u^{n+1}\| = \|Q^{n+1}u^0\| \leq \|Q^{n+1}\| \|u^0\| \leq Ke^{\beta t} \|u^0\|$$

e então o esquema numérico é estável na norma  $\|\cdot\|$ .

$\Rightarrow$  Supomos que o esquema numérico é estável na norma  $\|\cdot\|$ . Então existem constantes  $K$  e  $\beta$  tais que  $\|u^{n+1}\| \leq Ke^{\beta t} \|u^0\|$ ,  $n = 0, 1, 2, \dots$ . Por definição

$$\|Q^{n+1}\| = \sup_{\|u\| \leq 1} \|Q^{n+1}u\| = \sup_{\|u\| \leq 1} Ke^{\beta t} \|u\| \leq Ke^{\beta t}$$

como queríamos mostrar.

**Exemplo 6.1 :** *estabilidade do esquema de Euler para equação de difusão .*

Considere o esquema numérico

$$u_k^{n+1} = (1 - 2r)u_k^n + r(u_{k+1}^n + u_{k-1}^n)$$

para a solução de  $v_t = \nu v_{xx}$ . Mostraremos que esse esquema é **condicionalmente estável** com relação à norma do supremo.

Observamos que se  $1 - 2r \geq 0$  então

$$|u_k^{n+1}| \leq (1 - 2r)|u_k^n| + r|u_{k+1}^n| + r|u_{k-1}^n| \leq (1 - 2r + r + r)\|u^n\|_\infty$$

e assim, tomando o supremo para  $0 \leq k \leq \infty$ :

$$\|u^{n+1}\|_\infty \leq \|u^n\|_\infty, n = 0, 1, 2, \dots$$

o que implica  $\|u^{n+1}\| \leq \|u^0\|$ ,  $n = 0, 1, 2, \dots$

Dessa forma, se  $0 \leq r \leq 1/2$ , a condição de estabilidade é satisfeita para  $K = 1, \beta = 0$ .

**Exemplo 6.2 :** *estabilidade do esquema de Euler para  $v_t + av_x = 0$*

Considere a equação de onda unidimensional

$$v_t + av_x = 0, x > 0, t > 0$$

com condição de contorno  $v(0, t) = g(t)$  e condição inicial  $v(x, 0) = f(x)$ .

O esquema de Euler para essa equação é

$$\frac{v_k^{n+1} - v_k^n}{\Delta t} + a \frac{v_{k+1}^n - v_k^n}{\Delta x} = 0$$

e assim temos

$$\begin{aligned} v_0^n &= g(t_n), n = 0, 1, 2, \dots \\ v_k^0 &= f(x_k), k = 0, 1, 2, \dots \\ v_k^{n+1} &= (1 + R)v_k^n - Rv_{k+1}^n, k = 0, 1, 2, \dots, n = 0, 1, 2, \dots \end{aligned}$$

onde  $R = a\Delta t/\Delta x$ .

Observamos que

$$\begin{aligned} \sum_{k=-\infty}^{\infty} |v_k^{n+1}|^2 &= \sum_{k=-\infty}^{\infty} |(1 + R)v_k^n - Rv_{k+1}^n|^2 \\ \sum_{k=-\infty}^{\infty} |v_k^{n+1}|^2 &\leq \sum_{k=-\infty}^{\infty} \{ |1 + R|^2 |v_k^n|^2 + 2|1 + R||R| |v_k^n| |v_{k+1}^n| + |R|^2 |v_{k+1}^n|^2 \} \\ \sum_{k=-\infty}^{\infty} |v_k^{n+1}|^2 &\leq \sum_{k=-\infty}^{\infty} \{ |1 + R|^2 |v_k^n|^2 + 2|1 + R||R| (|v_k^n|^2 + |v_{k+1}^n|^2) + |R|^2 |v_{k+1}^n|^2 \} \\ \sum_{k=-\infty}^{\infty} |v_k^{n+1}|^2 &\leq \sum_{k=-\infty}^{\infty} (|1 + R|^2 + 2|1 + R||R| + |R|^2) |v_k^n|^2 \\ \sum_{k=-\infty}^{\infty} |v_k^{n+1}|^2 &\leq (|1 + R| + |R|)^2 \sum_{k=-\infty}^{\infty} |v_k^n|^2 \end{aligned}$$

e dessa forma, na norma  $\ell_2$  temos

$$\|u^{n+1}\|_2 \leq K_1 \|u^n\|_2, \text{ onde } K_1 = |1 + R| + |R|$$

e assim

$$\|u^{n+1}\|_2 \leq K_1^{n+1} \|u^0\|_2, n = 0, 1, 2, \dots$$

Dessa forma, o esquema é estável na norma  $\ell_2$  se e somente se existe uma constante  $K$  tal que

$$(|1 + R| + |R|)^{n+1} \leq K$$

o que somente é possível ( $K = 1$ ) se  $|1 + R| + |R| \leq 1$ .

Conclusões : com relação à norma  $\ell_2$ ,

- se  $R > 0$  então  $|1 + R| + |R| > 1$  e o esquema é incondicionalmente instável;
- se  $R < 0$  então o sistema é estável se e somente se  $|1 + R| \leq 1 + R$ ; ou seja  $1 + R < 0 \rightarrow -1 \leq R < 0$  (condicionalmente estável).

### Introdução 6.1 : Critérios para análise numérica da estabilidade

Três modalidades de análise numérica da estabilidade serão discutidas nesta disciplina:

- Análise de Fourier;
- Análise de Von Neumann;
- Análise espectral;

**Definição 6.2** A transformada discreta de Fourier de uma sequência  $u \in \ell_2$  é a função  $\hat{u} \in L_2[-\pi, \pi]$  definida por

$$\hat{u}(\xi) = \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-im\xi} u_m$$

para  $\xi \in [-\pi, \pi]$ .

**Lema 6.2** Se  $u \in \ell_2$  e  $\hat{u}$  é a transformada discreta de Fourier de  $u$ , então

(a) (Transformada Inversa)

$$u_m = \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{im\xi} \hat{u}(\xi) d\xi.$$

(b) (Identidade de Parseval)

$$\|\hat{u}\|_2 = \|u\|_2$$

**Prova:** (a)

$$\begin{aligned} \frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{ik\xi} \hat{u}(\xi) d\xi &= \frac{1}{\sqrt{2\pi}} e^{ik\xi} \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-im\xi} u_m d\xi = \\ &= \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} u_m \int_{-\pi}^{\pi} e^{i(k-m)\xi} d\xi \end{aligned}$$

e assim

$$\frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{ik\xi} \hat{u}(\xi) d\xi = \frac{1}{2\pi} \sum_{m=-\infty}^{\infty} u_m (2\pi \delta_{km}) = u_k.$$

(b) Temos

$$\begin{aligned} \|\hat{u}\|_2^2 &= \int_{-\pi}^{\pi} |\hat{u}(\xi)|^2 d\xi = \int_{-\pi}^{\pi} \overline{\hat{u}(\xi)} \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} e^{-im\xi} u_m d\xi = \\ &= \frac{1}{\sqrt{2\pi}} \sum_{m=-\infty}^{\infty} u_m \int_{-\pi}^{\pi} e^{-im\xi} \overline{\hat{u}(\xi)} d\xi \end{aligned}$$

e dessa forma

$$\|\hat{u}\|_2^2 = \sum_{m=-\infty}^{\infty} u_m \overline{\frac{1}{\sqrt{2\pi}} \int_{-\pi}^{\pi} e^{im\xi} \hat{u}(\xi) d\xi} = \sum_{m=-\infty}^{\infty} u_m \overline{u_m} = \|u\|_2^2.$$

**Lema 6.3** : Estratégia para Análise de Fourier

A sequência  $\{u^n\}$  é estável em  $\ell_{2,\Delta x}$  se e somente se a sequência  $\{\hat{u}^n\}$  é estável em  $L_2([-\pi, \pi])$ .

**Exemplo 6.3** : análise de Fourier do esquema de Euler para equação de difusão

Consideramos

$$u_k^{n+1} = ru_{k-1}^n + (1 - 2r)u_k^n + ru_{k+1}^n, \quad -\infty < k < \infty$$

onde  $r = \nu \Delta t / (\Delta x)^2$ , e aplicamos a Transformada Discreta de Fourier

$$\begin{aligned} \hat{u}^{n+1} &= \frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} e^{-ik\xi} (ru_{k-1}^n + (1 - 2r)u_k^n + ru_{k+1}^n) = \\ &= \frac{1}{\sqrt{2\pi}} \sum_{k=-\infty}^{\infty} [re^{-i(k+1)\xi} + (1 - 2r) \exp^{-ik\xi} + re^{-i(k-1)\xi}] u_k^n \end{aligned}$$

e assim

$$\hat{u}^{n+1}(\xi) = [re^{-i\xi} + (1 - 2r) + re^{i\xi}] \hat{u}^n(\xi)$$

e então

$$\hat{u}^{n+1}(\xi) = (2r \cos(\xi) + (1 - 2r)) \hat{u}^n(\xi) = (1 - 4r \sin^2(\xi/2)) \hat{u}^n(\xi).$$

Dessa forma, o esquema de Euler para  $v_t = \nu v_{xx}$  é estável na norma energia em  $\ell_{2,\Delta x}$  se e somente se

$$|1 - 4r\text{sen}^2(\xi/2)| \leq 1.$$

Temos então

$$-1 \leq 1 - 4r\text{sen}^2(\xi/2) \Leftrightarrow \text{sen}^2(\xi/2) \leq 2/r \quad \forall \xi \in [-\pi, \pi].$$

Assim, uma condição necessária e suficiente para a estabilidade é  $0 < r \leq 1/2$ .

**Definição 6.3** A quantidade  $\rho$  em  $\hat{u}^{n+1}(\xi) = \rho u^{n+1}(\xi)$  é chamada de símbolo do esquema numérico correspondente. Assumiremos que  $\rho$  é uma função contínua de  $\xi$ . Muitas vezes assumiremos que  $\rho(\xi)$  é derivável ou mesmo duas vezes derivável.

**Exemplo 6.4** : análise de Fourier do esquema de Euler para  $v_t + av_x = 0$

Considere o esquema de Euler para  $v_t + av_x = 0, a < 0$ :

$$u_k^{n+1} = (1 + R)u_k^n - Ru_{k+1}^n, k = 0, \pm 1, \pm 2, \dots$$

onde  $R = a\Delta t/\Delta x$ .

Aplicando a TDF temos

$$\hat{u}^{n+1}(\xi) = (1 + R)\hat{u}^n(\xi) - Re^{i\xi}\hat{u}^n(\xi)$$

e assim  $\rho(\xi) = 1 + R - R\cos(\xi) - iR\text{sen}(\xi)$ .

Como  $\rho(\xi)$  é complexo:

$$|\rho(\xi)|^2 = (1 + R)^2 - 2R(1 + R)\cos(\xi) + R^2.$$

O valores máximo e mínimo da expressão à direita correspondem a  $\xi = \pm\pi$  e  $\xi = 0$ .

Como  $R < 0$ , temos

$$|1 + 2R| \leq |\rho(\xi)| \leq |1|$$

e então

$$-1 \leq 1 + 2R \Rightarrow -1 \leq R < 0$$

é a condição para estabilidade do esquema numérico.

**Exemplo 6.5** : estabilidade do esquema de Crank-Nicolson para equação de difusão

Considere o esquema implícito de Crank-Nicolson para  $v_t = \nu v_{xx}$ :

$$-\frac{r}{2}u_{k+1}^{n+1} + (1 + r)u_k^{n+1} - \frac{r}{2}u_{k-1}^{n+1} = \frac{r}{2}u_{k-1}^n + (1 - r)u_k^n + \frac{r}{2}u_{k+1}^n, k = 0, 1, 2, \dots$$

Aplicando a TDF,

$$-\frac{r}{2}e^{-i\xi}\hat{u}^{n+1} + (1 + r)\hat{u}^{n+1} - \frac{r}{2}e^{i\xi}\hat{u}^{n+1} = \frac{r}{2}e^{-i\xi}\hat{u}^n + (1 - r)\hat{u}^n + \frac{r}{2}e^{i\xi}\hat{u}^n$$

e então

$$\begin{aligned} \left[-\frac{r}{2}(e^{-i\xi} + e^{i\xi}) + 1 + r\right]\hat{u}^{n+1} &= \left[\frac{r}{2}(e^{-i\xi} + e^{i\xi}) + 1 - r\right]\hat{u}^n \\ (-r\cos(\xi) + 1 + r)\hat{u}^{n+1} &= (r\cos(\xi) + 1 - r)\hat{u}^n \\ (1 + 2r\text{sen}^2(\xi/2))\hat{u}^{n+1} &= (1 - 2r\text{sen}^2(\xi/2))\hat{u}^n \end{aligned}$$

e temos

$$\rho(\xi) = \frac{1 - 2r\text{sen}^2(\xi/2)}{1 + 2r\text{sen}^2(\xi/2)}.$$

Assim, como  $r > 0$ ,

$$|\rho(\xi)| \leq \frac{|1| + |2r\text{sen}^2(\xi/2)|}{1 + 2r\text{sen}^2(\xi/2)} \leq \frac{1 + 2r\text{sen}^2(\xi/2)}{1 + 2r\text{sen}^2(\xi/2)} = 1$$

e esse esquema numérico é **incondicionalmente estável** na norma  $\ell_{2,\Delta x}$ .

**Lema 6.4** : condição de von Neumann para estabilidade usando TDF

O esquema de diferenças  $u^{n+1} = Qu^n$  é estável com respeito à norma energia se e somente se existem constantes positivas  $\Delta t_0, \Delta x_0$  e  $C$  tais que

$$|\rho(\xi)| \leq 1 + C\Delta t$$

para  $0 < \Delta t \leq \Delta t_0, 0 < \Delta x \leq \Delta x_0$  e todo  $\xi \in [-\pi, \pi]$ .

**Demonstração** :

⇐ Lembramos  $|\rho(\xi)| \leq 1 + C\Delta t \leq e^{C\Delta t}$ , e então

$$|\rho(\xi)|^{n+1} \leq e^{(n+1)C\Delta t}.$$

Dessa forma

$$\begin{aligned} |\rho(\xi)|^{n+1} |\hat{u}^0(\xi)|^2 &\leq \left(e^{(n+1)C\Delta t}\right)^2 |\hat{u}^0(\xi)|^2 \\ \Rightarrow \int_{-\pi}^{\pi} |\hat{u}^{n+1}(\xi)|^2 d\xi &\leq \int_{-\pi}^{\pi} e^{2(n+1)C\Delta t} |\hat{u}^0(\xi)|^2 d\xi \\ \Rightarrow \|\hat{u}^{n+1}(\xi)\| &\leq e^{(n+1)C\Delta t} \|\hat{u}^0(\xi)\| \end{aligned}$$

e o esquema numérico é estável na norma  $\ell_2$ . Para a norma energia, temos

$$\|u^{n+1}\|_{2,\Delta x} = \sqrt{\Delta x}\|u^{n+1}\|_2 = \sqrt{\Delta x}\|\hat{u}^{n+1}\|_2 \leq$$

$$\sqrt{\Delta x} e^{b(n+1)\Delta t} \|\hat{u}^0\|_2 = \sqrt{\Delta x} e^{b(n+1)\Delta t} \|u^0\|_2 = e^{b(n+1)\Delta t} \|u^0\|_{2,\Delta x}$$

e a estabilidade também segue para a norma energia.

⇒ Omitida. Prova por contradição em J. Thomas.

**Exemplo 6.6** : *estabilidade do esquema de Euler para  $v_t = \nu v_{xx} + bv$*

Considere o esquema de Euler

$$u_k^{n+1} = ru_{k-1}^n + (1 - 2r + b\Delta t)u_k^n + ru_{k+1}^n, k = \pm 1, \pm 2, \dots$$

para a solução de  $v_t = v_{xx} + \nu v_{xx} + bv, b > 0$ .

Tomando a transformada de Fourier

$$\begin{aligned} \hat{u}^{n+1} &= re^{-\xi i} \hat{u}^n + (1 - 2r + b\Delta t) \hat{u}^n + re^{\xi i} \hat{u}^n \\ \hat{u}^{n+1} &= (1 - 2r + b\Delta t + 2r \cos(\xi)) \hat{u}^n \end{aligned}$$

e assim o símbolo  $\rho(\xi)$  satisfaz

$$\rho(\xi) = (1 - 4r \sin^2(\xi/2)) + b\Delta t.$$

Dessa forma, se  $r < 1/2$  temos

$$|\rho(\xi)| \leq 1 + b\Delta t \leq e^{b\Delta t}$$

e assim esse esquema numérico é condicionalmente estável na norma energia, com condição  $r < 1/2$ .

**Exemplo 6.7** : *Estabilidade de esquemas de passo duplo*

Considere o esquema de Leapfrog para  $v_t = \nu v_{xx}$ :

$$u_k^{n+1} - 2r(u_{k-1}^n - 2u_k^n + u_{k+1}^n) - u_k^{n-1} = 0$$

Aplicando a TDF

$$\begin{aligned} \hat{u}^{n+1} - 2r(e^{i\xi} - 2 + e^{-i\xi})\hat{u}^n - \hat{u}^{n-1} &= 0 \\ \hat{u}^{n+1} + 4r(1 - \cos(\xi))\hat{u}^n - \hat{u}^{n-1} &= 0 \\ \hat{u}^{n+1} + 8r \sin^2(\xi/2)\hat{u}^n - \hat{u}^{n-1} &= 0. \end{aligned}$$

Dessa vez, as soluções  $\hat{u}^n$  verificarão

$$\|u^n\| \leq e^{\beta t} \|u^0\|$$

se e somente se as raízes  $\lambda_1$  e  $\lambda_2$  da equação característica

$$\lambda^2 + 8r \sin^2(\xi/2)\lambda - 1 = 0$$

satisfazerem:

- $\lambda_1 \neq \lambda_2$ ,  $|\lambda_1| \leq 1 + C_1\Delta t$  e  $|\lambda_2| \leq 1 + C_2\Delta t$ ;
- $\lambda_1 = \lambda_2$  e  $|\lambda_1| = |\lambda_2| < 1$ .

Entretanto, da equação característica concluímos  $\lambda_1\lambda_2 = -1$  o que impossibilita a segunda alternativa acima. Além disso, seu discriminante é

$$\Delta = 64r^2 \sin^4(\xi/2) - 4(1)(-1) \geq 4$$

e sempre teremos duas raízes reais distintas

$$\begin{aligned} \lambda_1 &= \sqrt{16r^2 \sin^4(\xi/2) + 1} - 4r \sin^2(\xi/2) \\ \lambda_2 &= -\sqrt{16r^2 \sin^4(\xi/2) + 1} - 4r \sin^2(\xi/2) \end{aligned}$$

que satisfazem  $|\lambda_1||\lambda_2| = 1$ .

Após estudo dos sinais das derivadas da função  $p(u) = \sqrt{1+u^2} - u$  concluímos que essa função é monótona decrescente e então  $0 < p(u) \leq p(0) = 1$ . Assim segue  $|\lambda_1| \leq 1$ .

Após estudo dos sinais das derivadas da função  $q(u) = -\sqrt{1+u^2} - u$  concluímos  $q''(u) < 0$ , ou seja, a função é côncava para baixo. Então  $q(u) \leq q(0) + q'(0)u$  e segue

$$|\lambda_2| \geq 1 + 4r \sin^2(\xi/2)$$

e a estimativa  $|\lambda_2| \leq 1 + C\Delta t$  não pode valer.

Dessa forma, concluímos que esse esquema é incondicionalmente instável.

**Introdução 6.2** : *Análise de von Neumann*

A estabilidade de um esquema numérico de passo simples é estudada usando harmônicos discretos de Fourier; para tal, substituímos

$$u_k^n = \epsilon^n e^{ijk\pi\Delta x}$$

e restringimos os parâmetros de tal forma que  $|\epsilon| \leq 1$ .

**Exemplo 6.8** : *Análise de von Neumann para o esquema de Euler para  $v_t = \nu v_{xx}$*

Considere  $u_k^{n+1} = ru_{k-1}^n + (1 - 2r)u_k^n + ru_{k+1}^n$ .

Temos

$$\begin{aligned} \epsilon^{n+1} e^{ijk\pi\Delta x} &= \epsilon^n e^{ijk\pi\Delta x} (re^{-ijk\pi\Delta x} + (1 - 2r) + re^{ijk\pi\Delta x}) \\ \Rightarrow \epsilon &= 1 - 2r(1 - \cos(j\pi\Delta x)) = 1 - 4r \sin^2\left(\frac{j\pi\Delta x}{2}\right) \end{aligned}$$

e então o esquema é estável se e somente  $|\epsilon| \leq 1$ , ou seja  $r \leq 1/2$ .

### Introdução 6.3 : Análise Espectral

No contexto de um esquema numérico de passo simples é o estabelecimento de estimativas sobre os autovalores da matriz  $u^{n+1} = Qu^n, n = 0, 1, 2, \dots$ , a ferramenta da análise espectral  $Q$ , e conseqüentemente sobre sua norma induzida.

Relembramos o lema (6.1):

**Lema:** Um esquema numérico de passo simples  $u^{n+1} = Qu^n, n = 0, 1, 2, \dots$  é estável com respeito à norma  $\|\cdot\|$  se e somente se existem constantes  $K$  e  $\beta$  tais que

$$\|Q^{n+1}\| \leq Ke^{\beta t}, n = 0, 1, 2, \dots$$

para  $0 \leq t = (n+1)\Delta t, 0 < \Delta x \leq \Delta x_0$  e  $0 < \Delta t \leq \Delta t_0$ .

### Lema 6.5 (Teorema dos Discos de Gerschgorin)

Seja  $Q = (q_{ij})$  uma matriz  $L \times L$  e seja

$$\rho_s = \sum_{j=1; j \neq s} |q_{sj}|$$

a soma dos valores absolutos dos elementos da  $s$ -ésima linha que estão fora da diagonal principal. Então para cada autovalor  $\lambda$  de  $Q$  existe um  $s$  tal que

$$|\lambda - q_{ss}| \leq \rho_s.$$

**Prova:** Omitida.

### Lema 6.6 Se $A$ é uma matriz $N \times N$ então

$$\|A\|_2 = \sqrt{\sigma(A^T A)} \geq \max\{|\lambda| : \exists x \in \mathbb{C}^N, Ax = \lambda x\}.$$

Se  $A$  é hermitiana, então a igualdade se verifica.

### Exemplo 6.9 : Análise espectral do esquema de Crank-Nicolson para $v_t = \nu v_{xx}$

O esquema de Crank-Nicolson para  $v_t = \nu v_{xx}$  escreve-se

$$Au^{n+1} = Bu^n, n = 0, 1, 2, \dots$$

onde

$$A = \begin{bmatrix} 1+r & -r/2 & 0 & \dots & & & \\ -r/2 & 1+r & -r/2 & 0 & \dots & & \\ & 0 & -r/2 & 1+r & -r/2 & 0 & \dots \\ & & 0 & r/2 & \dots & \dots & \dots \\ & & & 0 & \dots & 1+r & -r/2 \\ & & & & \dots & -r/2 & 1+r \end{bmatrix}$$

$$B = \begin{bmatrix} 1-r & r/2 & 0 & \dots & & & \\ r/2 & 1-r & r/2 & 0 & \dots & & \\ & 0 & r/2 & 1-r & r/2 & 0 & \dots \\ & & 0 & r/2 & \dots & \dots & \dots \\ & & & 0 & \dots & 1-r & r/2 \\ & & & & \dots & r/2 & 1-r \end{bmatrix}$$

e dessa forma temos que estimar a norma espectral da matriz  $Q = A^{-1}B$ . Observamos que  $B = 2I - A$  e então  $Q = 2A^{-1} - I$ . Isso implica que  $Q$  é simétrica, e então  $\|Q\|_2 = \sigma(Q)$  (raio espectral).

Dessa forma, autovalores  $\mu$  e  $\lambda$  das matrizes  $Q$  e  $A$  se relacionam por

$$\mu = \frac{2}{\lambda} - 1.$$

Pelo Teorema dos Discos de Gerschgorin, temos

$$|\lambda - (1+r)| \leq r$$

e assim  $1 \leq \lambda \leq 1+2r$ . Finalmente,

$$\lambda \geq 1 \Rightarrow 0 < \frac{1}{\lambda} \leq 1 \Rightarrow -1 < \mu = \frac{2}{\lambda} - 1 \leq 1$$

e dessa forma  $0 < \sigma(Q) = \|Q\|_2 \leq 1$  e o esquema numérico é incondicionalmente estável.

### Exemplo 6.10 : Análise espectral do esquema de Euler para $v_t + av_x = 0$

Considere novamente o esquema numérico

$$u_k^{n+1} = (1+R)u_k^n - Ru_{k+1}^n, k = 0, 1, 2, \dots, K-1$$

$$u_K^{n+1} = 0, u_k^0 = f(k\Delta x), k = 0, 1, 2, \dots, K.$$

No formato matricial, temos  $u^{n+1} = Qu^n, n = 0, 1, 2, \dots$  onde  $u^0$  é determinado da condição inicial e

$$Q = \begin{bmatrix} 1+R & -R & 0 & \dots & & \\ 0 & 1+R & -R & 0 & & \\ & & \dots & \dots & \dots & \\ & \dots & 0 & 1+R & -R & \\ & & \dots & 0 & 1+R & \end{bmatrix}$$

Como  $Q$  é triangular superior, todos os seus autovalores  $\mu$  serão iguais a  $1+R$ , e então  $\sigma(Q) = |1+R|$ .

Como  $Q$  não é simétrica, temos em geral  $\sigma(Q) \leq \|Q\|_2$ , e a restrição

$$\sigma(Q) = |1+R| \leq 1$$

nos dará apenas uma condição necessária para a estabilidade. De fato, temos  $-1 \leq 1+R \leq 1 \Rightarrow -2 \leq R \leq 0$ .

Pela Análise de Fourier, já havíamos concluído que uma condição necessária e suficiente para a estabilidade é  $-1 \leq R \leq 0$ .

## 7 Convergência e o teorema de Lax.

**Definição 7.1** Um esquema numérico de diferenças finitas  $L_k^n u_k^n = g_k^n$  que aproxima uma equação diferencial parcial  $Lu = g$  é **pontualmente convergente** se para todo  $x$  e  $t$

$$u_k^n \rightarrow u(x, t)$$

ao  $\Delta x \rightarrow 0, \Delta t \rightarrow 0$  e  $(k\Delta x, (n+1)\Delta t) \rightarrow (x, t)$ .

**Definição 7.2** Um esquema numérico de diferenças finitas  $L_k^n u_k^n = g_k^n$  que aproxima uma equação diferencial parcial  $Lu = g$  é **convergente** se para todo  $t$

$$\|(u_k)^n - u(x_k, t)\| \rightarrow 0$$

ao  $\Delta x \rightarrow 0$  e  $\Delta t \rightarrow 0$ .

**Lema 7.1** : Teorema da Equivalência de Lax

(i) Um esquema numérico consistente de passo simples para um problema linear diferencial parcial bem-posto de valores iniciais é convergente se e somente se é estável. Em caso afirmativo, a ordem de convergência é a mesma ordem de consistência.

(ii) Um esquema numérico consistente de passo simples para um problema linear diferencial parcial bem-posto de valores iniciais e valores de contorno é convergente somente se o esquema de aproximação da EDP for estável. Em caso afirmativo, a ordem de convergência é a mesma ordem de consistência do problema.

### Observações :

- Um problema de valores iniciais é bem-posto se sua solução depende continuamente sobre suas condições iniciais.
- Pela parte (ii), a estabilidade do esquema numérico que aproxima a EDP pode não ser suficiente para a convergência do esquema numérico considerando as condições de contorno. Isso é porque as CC poderiam estar sendo "mal" aproximadas, e a noção de estabilidade definida nesta disciplina não leva isso em consideração .
- Da mesma forma que a convergência pode ser definida pontualmente ou uniformemente (via norma), tal vale para a consistência. Dessa forma, a parte (i) do Teorema acima pode ser entendida:

consistência	estabilidade	convergência
pontual ordem $(p, q)$	norma $\ \cdot\ $	pontual ordem $(p, q)$
norma ordem $(p, q)$	norma $\ \cdot\ $	norma ordem $(p, q)$

**Exemplo 7.1** : Convergência do esquema de Euler para problema de difusão

Já mostramos que o esquema  $u_k^{n+1} = ru_{k-1}^n + (1-2r)u_k^n + ru_{k+1}^n$ , para a aproximação de  $v_t = \nu v_{xx}$ , é consistente de primeira ordem no tempo e segunda ordem no espaço (consistência pontual). Também já mostramos que esse esquema é estável se e somente se  $r \leq 1/2$ .

**Condições de contorno de Dirichlet:**

$$v(0, t) = a(t), v(1, t) = b(t), t \geq 0$$

o esquema numérico tem duas equações

$$\begin{aligned} v_0^n &= a(n\Delta t) \\ v_K^n &= b(n\Delta t) \quad , n = 0, 1, 2, \dots \end{aligned}$$

que são aproximações **exatas** das condições de contorno.

Dessa forma, a parte (i) do Teorema de Lax pode em verdade ser aplicada, garantindo que o esquema acima é pontualmente convergente à solução exata do problema de difusão acima se e somente se  $r \leq 1/2$ .

**Condições de contorno de Neumann:**

$$v_x(0, t) = a(t), v_x(1, t) = b(t), t \geq 0$$

o esquema numérico deverá aproximar  $v_x(0, t)$  e  $v_x(1, t)$  usando fórmulas de diferenças finitas.

Neste caso, o parte (ii) Teorema de Lax apenas nos diz que uma condição necessária para a convergência já foi satisfeita, mas ainda não podemos garantir tal convergência.

⇒ Uma análise de consistência das aproximações para as condições de contorno, e respectivas ordens, se faz necessária para que se possa concluir a convergência da solução numérica do problema como um todo, bem como a ordem de convergência.

**Boas notícias:** O resultado da análise espectral independe das condições de contorno. Dessa forma, se nosso esquema for consistente e  $Q$  for simétrica, as condições necessária e suficiente dadas pela análise espectral são condições para a convergência do problema de valores iniciais e valores de contorno como um todo.

**Exemplo 7.2 :** *Análise espectral do esquema de Euler para o problema de difusão com CC mistas*

Seja o problema de valores iniciais e condições de contorno

$$\begin{aligned} v_t &= \nu v_{xx}, \quad x \in (0, 1), t > 0 \\ v(x, 0) &= f(x) = x(2-x), \quad x \in [0, 1] \\ v(0, t) &= 0, \quad t \geq 0 \\ v_x(1, t) &= 0, \quad t \geq 0 \end{aligned}$$

cujas solução pode ser encontrada:

$$v(x, t) = \sum_{m=1}^{\infty} a_m e^{-\lambda_m t} \sin\left(\frac{(2m+1)\pi x}{2}\right)$$

onde os coeficientes  $a_m$  e os autovalores  $\lambda_m$  são determinados por:

$$\begin{aligned} a_m &= \frac{\int_0^1 x(2-x) \sin\left(\frac{(2m+1)\pi x}{2}\right) dx}{\int_0^1 \sin^2\left(\frac{(2m+1)\pi x}{2}\right) dx} \\ \lambda_m &= \frac{(2m+1)^2 \pi \nu}{4} \end{aligned}$$

A condição de contorno  $v_x(1, t) = 0$  será tratada usando a aproximação de segunda ordem

$$\frac{u_{K-1}^n - u_{K+1}^n}{2\Delta x} = 0$$

que nos dá  $u_{K+1}^n = u_{K-1}^n$  então, usando a aproximação de Euler para  $k = K$ ,  $n = 0, 1, 2, \dots$ :

$$u_K^{n+1} = ru_{K-1}^n + (1-2r)u_K^n + ru_{K+1}^n$$

que então implica

$$u_K^{n+1} = (1-2r)u_K^n + 2ru_{K-1}^n.$$

Dessa forma, o esquema adotado será

$$\begin{aligned} u_k^{n+1} &= ru_{k-1}^n + (1-2r)u_k^n + ru_{k+1}^n, \quad k = 1, 2, 3, \dots; n = 0, 1, 2, \dots \\ u_k^0 &= f(k\Delta x), \quad k = 0, 1, 2, \dots \\ u_0^n &= 0, \quad n = 0, 1, 2, \dots \\ u_K^{n+1} &= (1-2r)u_K^n + 2ru_{K-1}^n, \quad n = 0, 1, 2, \dots \end{aligned}$$

Pode-se mostrar que a EDP é aproximada por esquema consistente de primeira ordem no tempo e segunda no espaço.

Analisaremos a equação que aproxima a CC em  $x = 1$ :

$$\begin{aligned} \tau_K^n &= \frac{\partial u(1, t_n)}{\partial x} = \frac{u_{K+1}^n - u_K^n}{\Delta x} - \frac{\partial^2 u(1, t_n)}{\partial x^2} \frac{\Delta x}{2} - \frac{\partial^3 u(1, t_n)}{\partial x^3} \frac{(\Delta x)^2}{3!} + O((\Delta x)^3) \\ \tau_K^n &= \frac{1}{\Delta x} \left[ \frac{u_K^{n+1} - (1-2r)u_K^n - u_K^n}{2r} - \frac{\partial^2 u(1, t_n)}{\partial x^2} \frac{\Delta x}{2} - \frac{\partial^3 u(1, t_n)}{\partial x^3} \frac{(\Delta x)^2}{3!} + O((\Delta x)^3) \right] \\ \tau_K^n &= \frac{1}{\Delta x} \frac{u_K^{n+1} - u_K^n}{2r} - \frac{\partial^2 u(1, t_n)}{\partial x^2} \frac{\Delta x}{2} - \frac{\partial^3 u(1, t_n)}{\partial x^3} \frac{(\Delta x)^2}{3!} + O((\Delta x)^3) \\ \tau_K^n &= \frac{\Delta x}{2\nu} \left[ \frac{\partial u(1, t_n)}{\partial t} + \frac{\partial^2 u(1, t_n)}{\partial t^2} \frac{\Delta t}{2} + O((\Delta t)^2) \right] - \frac{\partial^2 u(1, t_n)}{\partial x^2} \frac{\Delta x}{2} - \frac{\partial^3 u(1, t_n)}{\partial x^3} \frac{(\Delta x)^2}{3!} + O((\Delta x)^3) \\ \tau_K^n &= \frac{\Delta x}{2\nu} \left[ \nu \frac{\partial^2 u(1, t_n)}{\partial x^2} + \frac{\partial^2 u(1, t_n)}{\partial t^2} \frac{\Delta t}{2} + O((\Delta t)^2) \right] - \frac{\partial^2 u(1, t_n)}{\partial x^2} \frac{\Delta x}{2} + O((\Delta x)^2) \end{aligned}$$

e concluímos  $\tau_K^n = O(\Delta t \Delta x) + O((\Delta x)^2)$ .

**Surpresa: o esquema numérico como um todo pode não ser de segunda ordem no espaço...**

Para decidirmos estabilidade, e então convergência, observamos que todo o esquema acima pode ser escrito na forma  $u^{n+1} = Qu^n$ ,  $n = 0, 1, 2, \dots$ , onde

$$u^n = \begin{bmatrix} u_1^n & u_2^n & \dots & u_{K-1}^n & u_K^n \end{bmatrix}^T$$

$$Q = \begin{bmatrix} 1-2r & r & & \dots & & \\ r & 1-2r & r & & \dots & \\ 0 & r & 1-2r & r & \dots & \\ & & \dots & \dots & r & \\ & \dots & 0 & 2r & 1-2r & \end{bmatrix}$$

e onde  $u^0$  é determinado usando as condições iniciais.

Entretanto, pode-se mostrar que

$$S^{-1}QS = \begin{bmatrix} 1-2r & r & & \dots & & \\ r & 1-2r & r & & \dots & \\ 0 & r & 1-2r & r & \dots & \\ & & \dots & \dots & \sqrt{2}r & \\ & \dots & 0 & \sqrt{2}r & 1-2r & \end{bmatrix}$$

para

$$S = \text{diag} ( 1 \quad 1 \quad \dots \quad 1 \quad \sqrt{2} )$$

e então  $Q$  é similar a uma matriz que possui autovalores

$$\lambda_j = 1 - 4r \text{sen}^2 \left( \frac{(2j+1)\pi}{4K} \right)$$

e então  $r \leq 1/2$  é uma condição necessária e suficiente para a convergência do esquema numérico.

## 8 Solução numérica de equações parabólicas.

**Introdução 8.1** : *Modelo parabólico básico em duas dimensões .*

O problema de difusão (calor) em duas dimensões escreve-se:

$$\begin{aligned} v_t &= \nu(v_{xx} + v_{yy}) + F(x, y, t), (x, y) \in R, t > 0 \\ v(x, y, t) &= g(x, y, t), (x, y) \in \partial R, t > 0 \\ v(x, y, 0) &= f(x, y), (x, y) \in \bar{R} \end{aligned} .$$

A estratégia de solução via diferenças finitas requer discretizações no tempo e no espaço. A novidade é que agora teremos também um parâmetro  $\Delta y$  para a discretização da variável espacial  $y$ .

A ferramenta matemática básica para análise é o Teorema de Taylor em duas variáveis: se  $f(x, y, t)$  é uma função analítica em alguma vizinhança do ponto  $(x, y, t)$  então

$$f(x+h_x, y, t) = f(x, y, t) + \frac{\partial f(x, y, t)}{\partial x} h_x + \frac{\partial^2 f(x, y, t)}{\partial x^2} \frac{(h_x)^2}{2!} + \frac{\partial^3 f(x, y, t)}{\partial x^3} \frac{(h_x)^3}{3!} + \dots$$

ou ainda

$$f(x, y+h_y, t) = f(x, y, t) + \frac{\partial f(x, y, t)}{\partial y} h_y + \frac{\partial^2 f(x, y, t)}{\partial y^2} \frac{(h_y)^2}{2!} + \frac{\partial^3 f(x, y, t)}{\partial y^3} \frac{(h_y)^3}{3!} + \dots$$

ou ainda

$$f(x, y, t+h_t) = f(x, y, t) + \frac{\partial f(x, y, t)}{\partial t} h_t + \frac{\partial^2 f(x, y, t)}{\partial t^2} \frac{(h_t)^2}{2!} + \frac{\partial^3 f(x, y, t)}{\partial t^3} \frac{(h_t)^3}{3!} + \dots$$

A discretização será feita via

$$\begin{aligned} u(x_j, y_k, t_n) &= u_{jk}^n, \quad F(x, y, t) = F_{jk}^n \\ x_j &= j\Delta x, \quad y_k = k\Delta y, \quad t_n = n\Delta t \end{aligned}$$

e agora precisaremos usar um sub-índice nas fórmulas  $\delta$  para diferenças finitas:

$$\begin{aligned} \delta_x^1 &= \frac{v_{j+1,k}^n - v_{j-1,k}^n}{2} \\ \delta_y^2 &= v_{j,k+1}^n - 2v_{j,k}^n + v_{j,k-1}^n (\Delta y)^2 \end{aligned}$$

e assim por diante.

Dessa forma, usando diferenças simétricas de segunda ordem para  $v_{xx}$  e  $v_{yy}$ , e de primeira ordem para  $v_t$ , quando  $R = [0, 1] \times [0, 1]$ , vemos que  $v_t = \nu(v_{xx} + v_{yy}) + F(x, t)$  discretiza-se:

$$\frac{v_{jk}^{n+1} - v_{jk}^n}{\Delta t} = \nu \left( \frac{v_{j+1,k}^n - 2v_{j,k}^n + v_{j-1,k}^n}{(\Delta x)^2} + \frac{v_{j,k+1}^n - 2v_{j,k}^n + v_{j,k-1}^n}{(\Delta y)^2} \right) + F_{jk}^n$$

para  $n = 0, 1, 2, 3, \dots; j = 1, 2, 3, \dots, J-1; k = 1, 2, 3, \dots, K-1$ , e temos

$$v_{jk}^{n+1} = v_{jk}^n + (r_x \delta_x^2 + r_y \delta_y^2) v_{jk}^n + \Delta t F_{jk}^n$$

onde  $r_x = \nu \Delta t / (\Delta x)^2$ ,  $r_y = \nu \Delta t / (\Delta y)^2$ .

A condição inicial  $v(x, y, 0) = f(x, y)$ ,  $(x, y) \in \bar{R}$  se escreve

$$v_{j,k}^0 = f(j\Delta x, k\Delta y), j = 0, 1, \dots, J; k = 0, 1, \dots, K.$$

Ao passo que a condição de contorno  $v(x, y, t) = g(x, y, t)$ ,  $(x, y) \in \partial R, t > 0$  escreve-se

$$\begin{aligned} v_{0,k}^n &= g(0, k\Delta y, n\Delta t), \quad k = 0, 1, \dots, K, n \geq 0 \\ v_{j,k}^n &= g(1, k\Delta y, n\Delta t), \quad k = 0, 1, \dots, K, n \geq 0 \\ v_{j,0}^n &= g(j\Delta x, 0, n\Delta t), \quad j = 0, 1, \dots, J, n \geq 0 \\ v_{j,K}^n &= g(j\Delta x, 1, n\Delta t), \quad j = 0, 1, \dots, J, n \geq 0 \end{aligned}$$

**Introdução 8.2** : Aspectos computacionais e desempenho

**Estrutura de dados:** Usaremos *matrizes* multidimensionais ou **hiper-matrizes**

$A = A(j, k, n)$ .

<b>Matlab 6:</b>	<b>g77:</b>	<b>gcc</b>
<code>A = zeros(100,100,80);</code>	<code>real A(100,100,80)</code>	<code>float A[100][100][80];</code>
<code>size(A)</code>	<code>integer n</code>	<code>int n;</code>
<code>ans</code>	<code>do n=1,20</code>	<code>for (n = 0; n &lt; 80; n++)</code>
<code>100 100 80</code>	<code>A(1,2,n) = 1E0;</code>	<code>A[1][1][n]=1.0E+0;</code>
	<code>end do</code>	

**Estratégia para computação de desempenho (speed-up):** agora temos a possibilidade de fazer operações **vetoriais** ou mesmo **matriciais**, o que aumenta o desempenho do código, mas que depende das características de cada compilador.

**Estratégia para otimização de memória usada (workspace):** agora temos a possibilidade de usar muito mais memória do que o estritamente necessário, dependendo do propósito da computação .

Exemplo: se somente a solução em  $t = t_f$  nos interessa (rumo à solução estacionária, por exemplo), podemos tentar armazenar nossos dados em uma matriz simples, que então deveria ser atualizada em cada etapa do laço mais externo em  $n$ .

Usaremos **Matlab** para explicar como implementar

$$v_{jk}^{n+1} = v_{jk}^n + (r_x \delta_x^2 + r_y \delta_y^2) v_{jk}^n + \Delta t F_{jk}^n$$

**Código nível 0: (totalmente escalar)**

```

for n=1:n_max
  for k=2:k_max
    for j=2:j_max
      ...
      dxx = v(j+1,k,n) - 2*v(j,k,n) + v(j-1,k,n);
      dyy = v(j,k+1,n) - 2*v(j,k,n) + v(j,k-1,n);
      F_jkn = ...;
      v(j,k,n+1) = v(j,k,n) + (rx*dxx + ry*dyy) + delta_t * F_jkn
    end
  end
end

```

**Código nível 1: (alguma vetorização )**

```

for n=1:n_max
  for k=2: k_max

```

```

      dyy = v(2:j_max,k+1,n) - 2*v(2:j_max,k,n) + v(2:j_max,k-1,n);
      v(2:j_max,k,n+1) = v(2:j_max,k,n) + ry*dyy;
    end
  for j=2: j_max
      dxx = v(j+1,2:k_max,n) - 2*v(j,2:k_max,n) + v(j-1,2:k_max,n);
      v(j,2:k_max,n+1) = v(j,2:k_max,n) + rx*dxx;
      Fj_kn = F(2:k_max,y_k,t_n);
      v(j,2:k_max,n+1) = v(j,2:k_max,n) + delta_t * Fj_kn;
    end

```

**Código nível 2: (matriz-vetor)** Definimos

$$v_{_xx} = v_{yy} = \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}^T$$

```

for n=1:n_max
  for k=2: k_max
    v(2:j_max,k,n+1) = v(2:j_max,k,n) + ry*v(2:j_max,k-1:k+1,n)*v_yy;
  end
  for j=2: j_max
    v(j,2:k_max,n+1) = v(j,2:k_max,n) + rx*v_xx^T*v(j-1:j+1,2:k_max,n);
    Fj_kn = F(x_j, (2:k_max)Δy, t_n);
    v(j,2:k_max,n+1) = v(j,2:k_max,n) + delta_t * Fj_kn;
  end
end

```

**Código nível 3: (matriz-matriz)** Definimos duas matrizes de ordems  $(j_{_max} - 1) \times (j_{_max} + 1)$  e  $(k_{_max} + 1) \times (k_{_max} - 1)$  respectivamente:

$$M_{_xx} = \begin{bmatrix} 1 & -2 & 1 & & & \\ & 1 & -2 & 1 & & \\ & & \dots & \dots & \dots & \\ & & & 1 & -2 & 1 \end{bmatrix} \quad M_{_yy} = \begin{bmatrix} 1 & & & & & \\ 2 & 1 & & & & \\ 1 & -2 & 1 & & & \\ & 1 & -2 & & & \\ & & \dots & 1 & & \\ & & & 1 & -2 & \\ & & & & & 1 \end{bmatrix}$$

```

for n=1:n_max
  v(2:j_max,2:k_max,n+1) = v(2:j_max,2:k_max,n) +
  ry*v(2:j_max,1:k_max+1,n)*M_yy;
  v(2:j_max,2:k_max,n+1) = v(2:j_max,2:k_max,n) +
  M_xx*v(1:j_max+1,2:k_max,n);
  v(2:j_max,2:k_max,n+1) = v(2:j_max,2:k_max,n) + dt *
  F((2:j_max)Δx,(2:k_max)Δy, t_n)
end

```

**Exemplo 8.1** : *esquema de Euler para problema de difusão em duas dimensões*

Considere a solução numérica, via esquema de Euler, do problema

$$\begin{aligned} v_t &= \nu(v_{xx} + v_{yy}), & (x, y) \in (0, 1) \times (0, 1), t > 0 \\ v(x, y, 0) &= \text{sen}(\pi x)\text{sen}(2\pi y), & (x, y) \in [0, 1] \times [0, 1] \\ v(0, y, t) &= v(1, y, t) = 0 & , y \in (0, 1), t > 0 \\ v(x, 0, t) &= v(x, 1, t) = 0 & , x \in (0, 1), t > 0 \end{aligned}$$

usando os comandos básicos de inicialização e computação descritos acima.

Escrevendo a solução da EDP como

$$v(x, y, t) = \exp(\lambda t)\psi(x, y)$$

vemos que a função  $\psi(x, y)$  deve satisfazer

$$\nabla^2(\psi(x, y)) = \lambda\psi(x, y)$$

e a técnica de expansão por autofunções assume

$$\psi(x, y) = \sum_{m,n=0}^{\infty} A_{mn}p_m(x)q_n(y)$$

onde então  $p_m$  e  $q_n$  satisfazem

$$\begin{aligned} p_m''(x) &= \alpha_m p_m(x), & p_m(0) = p_m(1) = 0 \\ q_n''(x) &= \beta_n q_n(x), & q_n(0) = q_n(1) = 0. \end{aligned}$$

Dessa forma

$$\psi(x, y) = \sum_{m,n=1}^{\infty} A_{mn}\text{sen}(\pi m x)\text{sen}(\pi n y)$$

e assim temos autovalores  $\lambda_{mn} = -(n^2 + m^2)\pi\nu$ .

Por outro lado, a condição inicial nos dá

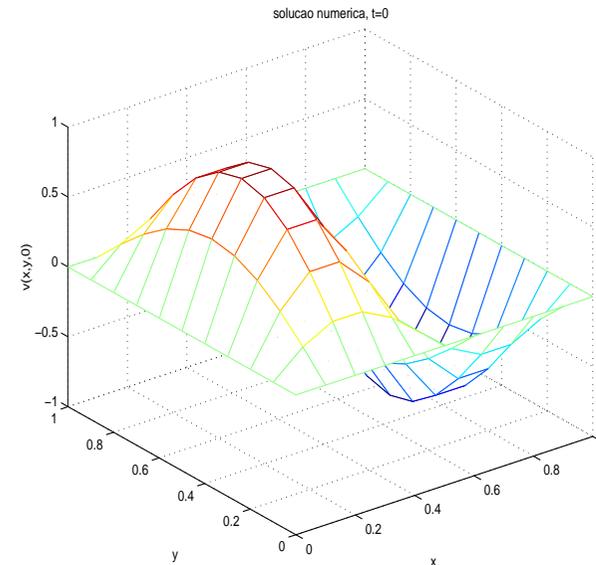
$$\text{sen}(\pi x)\text{sen}(2\pi y) = \sum_{m,n=1}^{\infty} A_{mn}\text{sen}(\pi m x)\text{sen}(\pi n y)$$

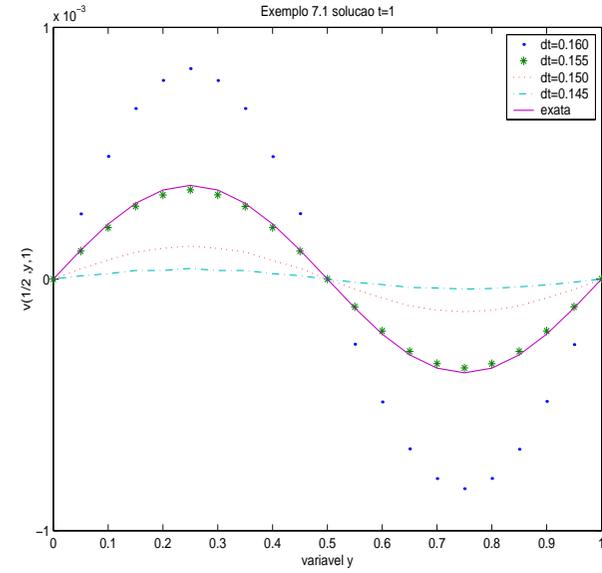
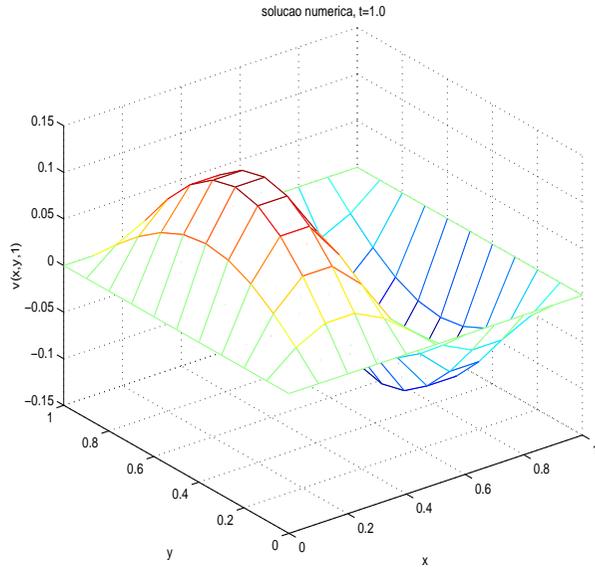
e então a solução exata de nosso problema é

$$v(x, y, t) = \exp(-5\pi^2\nu t)\text{sen}(\pi x)\text{sen}(2\pi y).$$

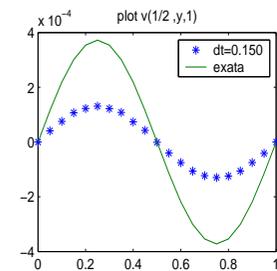
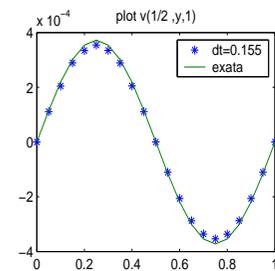
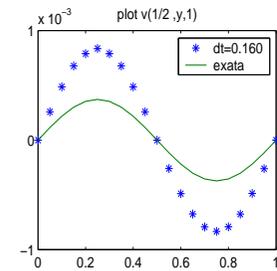
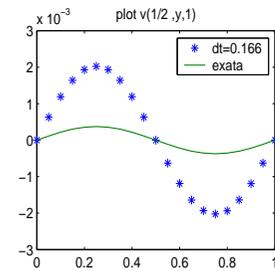
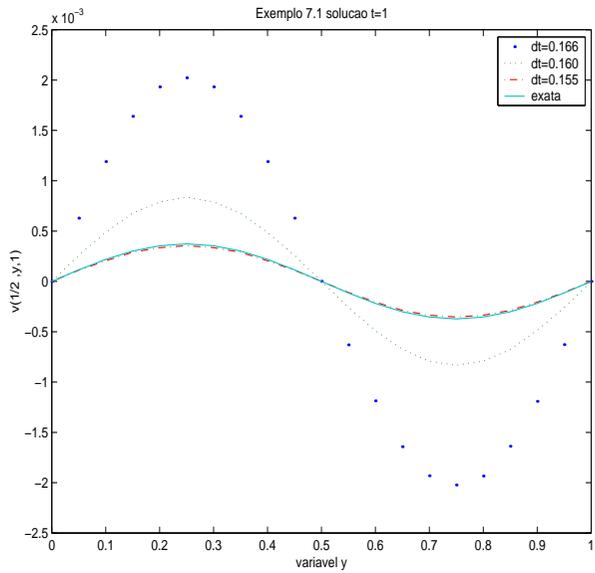
Os gráficos abaixo mostram as soluções numéricas correspondendo a  $t = 0$  e  $t = 1$ ; foram usados  $\Delta x = \Delta y = 0.1$ ,  $\Delta t = 0.20$ .

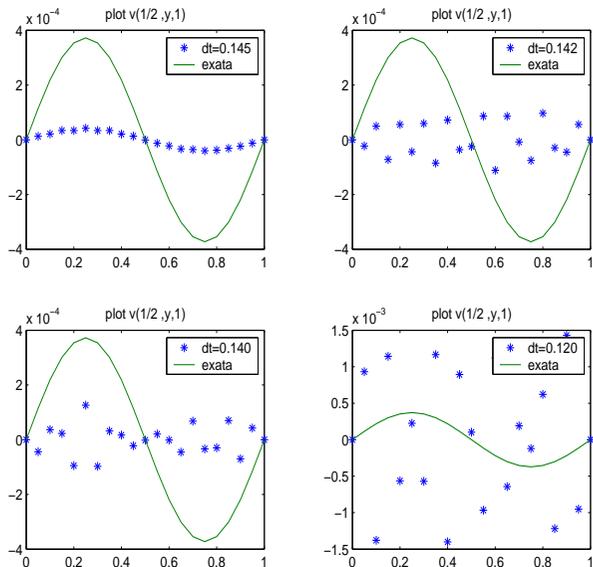
Em particular, vemos que, quando  $\Delta x$  e  $\Delta y$  estão fixos, existe um valor  $\Delta t_0$  para  $\Delta t$  que produz uma melhor aproximação à solução exata, e valores  $\Delta t$  menores produzem resultados desastrosos. Esse fenômeno tem a haver com a acumulação de erros de arredondamento.





Os gráficos abaixo mostram a aproximação à solução exata ao longo das retas  $x = 1/2$  e  $t = 1$  quando variamos  $\Delta t$ .





**Exemplo 8.2 :** Análise de consistência do esquema de Euler para  $v_t = \nu(v_{xx} + v_{yy})$ .

Negligenciando erro de arredondamento, temos

$$\tau_{jk}^n = \frac{u(x_j, j_k, t_{n+1}) - u(x_j, y_k, t_n)}{\Delta t} - \nu \frac{u(x_{j+1}, y_k, t_n) - 2u(x_j, y_k, t_n) + u(x_{j-1}, y_k, t_n)}{(\Delta x)^2} - \nu \frac{u(x_j, y_{k+1}, t_n) - 2u(x_j, y_k, t_n) + u(x_j, y_{k-1}, t_n)}{(\Delta y)^2}$$

$$\tau_{jk}^n = \frac{1}{\Delta t} \left( \Delta t \frac{\partial u}{\partial t}(x_j, y_k, t_n) + \frac{(\Delta t)^2}{2} \frac{\partial^2 u}{\partial t^2}(x_j, y_k, t_n) + O((\Delta t)^3) \right) - \frac{\nu}{(\Delta x)^2} \left( \frac{2(\Delta x)^2}{2!} \frac{\partial^2 u}{\partial x^2}(x_j, y_k, t_n) + \frac{2(\Delta x)^4}{4!} \frac{\partial^4 u}{\partial x^4}(x_j, y_k, t_n) + \frac{2(\Delta x)^6}{6!} \frac{\partial^6 u}{\partial x^6}(x_j, y_k, t_n) + O(\frac{(\Delta x)^8}{8!}) \right) - \frac{\nu}{(\Delta y)^2} \left( \frac{2(\Delta y)^2}{2!} \frac{\partial^2 u}{\partial y^2}(x_j, y_k, t_n) + \frac{2(\Delta y)^4}{4!} \frac{\partial^4 u}{\partial y^4}(x_j, y_k, t_n) + \frac{2(\Delta y)^6}{6!} \frac{\partial^6 u}{\partial y^6}(x_j, y_k, t_n) + O(\frac{(\Delta y)^8}{8!}) \right)$$

e dessa forma

$$\tau_{jk}^n = \frac{\Delta t}{2} \frac{\partial^2 u}{\partial t^2}(x_j, y_k, t_n) - \frac{(\Delta x)^2}{12} \frac{\partial^4 u}{\partial x^4}(x_j, y_k, t_n) - \frac{(\Delta y)^2}{12} \frac{\partial^4 u}{\partial y^4}(x_j, y_k, t_n) + O((\Delta x)^4, (\Delta y)^4, (\Delta t)^2)$$

e o esquema numérico é de ordem 1 no tempo e dois no espaço.

**Exemplo 8.3 :** Análise de estabilidade do esquema de Euler para  $v_t = \nu(v_{xx} + v_{yy})$ .

Negligenciando o termo não-homogêneo, consideramos

$$u_{jk}^{n+1} = u_{jk}^n + (r_x \delta_x^2 + r_y \delta_y^2) u_{jk}^n$$

ou

$$u_{jk}^{n+1} = u_{jk}^n + r_x(u_{j+1,k}^n - 2u_{jk}^n + u_{j-1,k}^n) + r_y(u_{j,k+1}^n - 2u_{jk}^n + u_{j,k-1}^n).$$

Aplicando a Transformada Discreta de Fourier em duas dimensões :

$$\hat{u}(\xi, \eta) = \frac{1}{2\pi} \sum_{j,k=-\infty}^{\infty} e^{-ij\xi - ik\eta} u_{jk}$$

onde  $\xi, \eta \in [-\pi, \pi]$  e onde a Identidade de Parseval  $\|u_{jk}\|_2 = \|\hat{u}\|_{L_2}$  vale, temos

$$\hat{u}^{n+1} = \hat{u}^n + r_x(e^{i\xi} - 2 + e^{-i\xi})\hat{u}^n + r_y(e^{i\eta} - 2 + e^{-i\eta})\hat{u}^n$$

$$\hat{u}^{n+1} = \hat{u}^n - 2r_x(1 - \cos(\xi))\hat{u}^n - 2r_y(1 - \cos(\eta))\hat{u}^n$$

$$\hat{u}^{n+1} = (1 - 4r_x \sin^2(\xi/2) - 4r_y \sin^2(\eta/2))\hat{u}^n$$

e o símbolo  $\rho(\xi, \eta)$  é dado por

$$\rho(\xi, \eta) = 1 - 4r_x \sin^2(\xi/2) - 4r_y \sin^2(\eta/2).$$

Entretanto, tomando derivadas parciais de  $\rho$  com relação a  $\xi$  e  $\eta$ , vemos que o máximo dessa função ocorre em  $(\xi, \eta) = (0, 0)$ , e que o mínimo ocorre em  $(\xi, \eta) = (\pi, \pi)$ .

A condição  $\rho(\xi, \eta) \geq -1$  então implica e requer  $r_x + r_y \leq 1/2$ .

Assim, esse esquema é estável na norma energia

$$\|u_{jk}\|_{2,\Delta x} = \sqrt{\sum_{j,k=-\infty}^{\infty} |u_{jk}|^2 \Delta x \Delta y}$$

se e somente se  $r_x + r_y \leq 1/2$ . Dessa forma, pelo Teorema de Lax, o esquema de Euler para solução do problema bidimensional de difusão, com condições de Dirichlet, é convergente se e somente se  $r_x + r_y \leq 1/2$ .

**Exemplo 8.4 :** problema de difusão 2D com condições de contorno de Neumann

Considere o problema de valores iniciais e condições de contorno

$$v_t = \nu(v_{xx} + v_{yy}) + 10, \quad (x, y) \in (0, 1) \times (0, 1), t > 0$$

$$v(x, y, 0) = f(x, y), \quad (x, y) \in [0, 1] \times [0, 1]$$

$$v_x(0, y, t) = v_x(1, y, t) = 0, \quad y \in (0, 1), t > 0$$

$$v_y(x, 0, t) = v_y(x, 1, t) = 0, \quad x \in (0, 1), t > 0$$

onde  $f(x, y) = \sin^2(\pi x)\sin^2(\pi y)$ . A condição de compatibilidade

$$\begin{aligned}\frac{\partial f}{\partial x}(0, y) &= \frac{\partial f}{\partial x}(1, y) = 0 \\ \frac{\partial f}{\partial y}(x, 0) &= \frac{\partial f}{\partial y}(x, 1) = 0\end{aligned}$$

é claramente satisfeita. O esquema de Euler com tratamento de segunda ordem para CC será adotado.

**Solução Exata:** Observamos que

$$f(x, y) = \frac{1 - \cos(2\pi x)}{2} \cdot \frac{1 - \cos(2\pi y)}{2} = \frac{1 - (\cos(2\pi x) + \cos(2\pi y)) + \cos(2\pi x)\cos(2\pi y)}{4}$$

e assim

$$f(x, y) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} A_{mn} \cos(m\pi x) \cos(n\pi y)$$

onde  $A_{00} = 1/4$ ,  $A_{20} = A_{02} = -1/4$ ,  $A_{22} = 1/4$  e os demais  $A_{mn}$  são nulos.

A solução exata é então

$$v(x, y, t) = \frac{1}{4} - \exp(-4\pi^2 \nu t) \frac{\cos(2\pi x) + \cos(2\pi y)}{4} + \exp(-8\pi^2 \nu t) \frac{\cos(2\pi x)\cos(2\pi y)}{4}.$$

**Solução Numérica:** A aproximação da EDP:

$$u_{jk}^{n+1} = u_{jk}^n + r_x(u_{j+1,k}^n - 2u_{jk}^n + u_{j-1,k}^n) + r_y(u_{j,k+1}^n - 2u_{jk}^n + u_{j,k-1}^n) + 10\Delta t.$$

A condição inicial:

$$u_{jk}^0 = f(j\Delta x, k\Delta y).$$

As condições de contorno:

$$\begin{aligned}\frac{u_{1,k}^n - u_{-1,k}^n}{\Delta x} &= 0, \quad \frac{u_{J+1,k}^n - u_{J-1,k}^n}{\Delta x} = 0, \\ \frac{u_{j,1}^n - u_{j,-1}^n}{\Delta y} &= 0, \quad \frac{u_{j,K+1}^n - u_{j,K-1}^n}{\Delta y} = 0.\end{aligned}$$

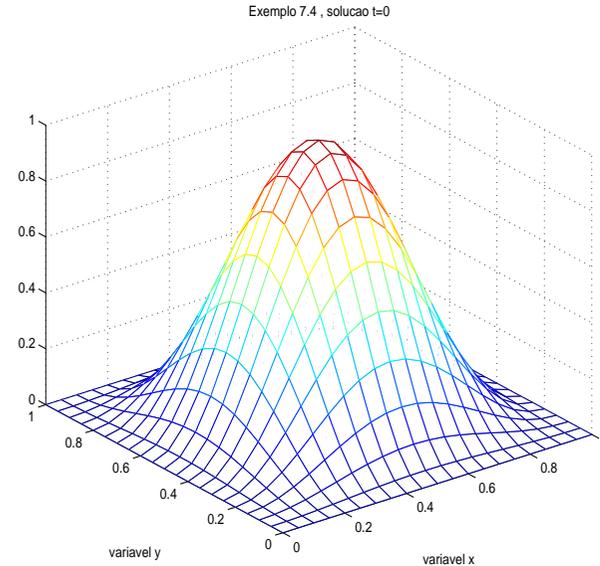
e então temos  $u_{-1,k}^n = u_{1,k}^n$ ,  $u_{J+1,k}^n = u_{J-1,k}^n$ ,  $u_{j,-1}^n = u_{j,1}^n$  e  $u_{j,K+1}^n = u_{j,K-1}^n$ . Essas relações são substituídas em

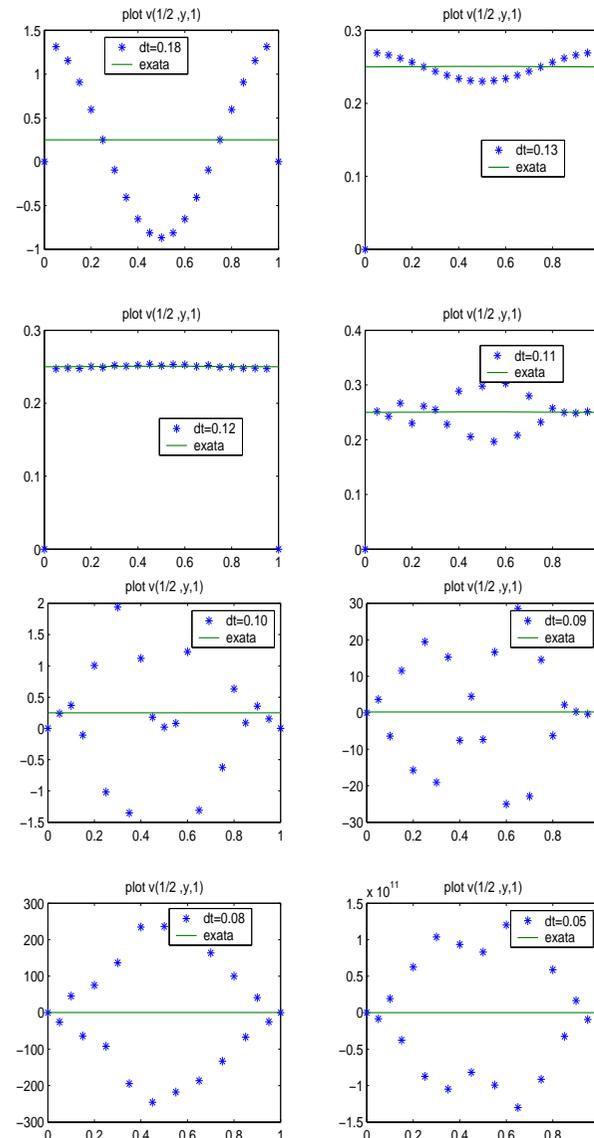
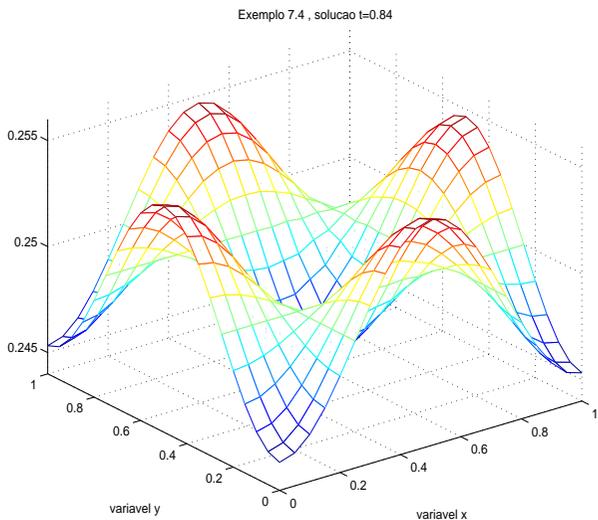
resultando

$$\begin{aligned}u_{0k}^{n+1} &= u_{0k}^n + r_x(u_{1k}^n - 2u_{0k}^n + u_{-1,k}^n) + r_y(u_{0,k+1}^n - 2u_{0k}^n + u_{0,k-1}^n) + 10\Delta t. \\ u_{Jk}^{n+1} &= u_{Jk}^n + r_x(u_{J+1,k}^n - 2u_{Jk}^n + u_{J-1,k}^n) + r_y(u_{J,k+1}^n - 2u_{Jk}^n + u_{J,k-1}^n) + 10\Delta t. \\ u_{j0}^{n+1} &= u_{j0}^n + r_x(u_{j+1,0}^n - 2u_{j0}^n + u_{j-1,0}^n) + r_y(u_{j,1}^n - 2u_{j0}^n + u_{j,-1}^n). \\ u_{jK}^{n+1} &= u_{jK}^n + r_x(u_{j+1,K}^n - 2u_{jK}^n + u_{j-1,K}^n) + r_y(u_{j,K+1}^n - 2u_{jK}^n + u_{j,K-1}^n)\end{aligned}$$

$$\begin{aligned}u_{00}^{n+1} &= u_{00}^n + r_x(2u_{10}^n - 2u_{00}^n) + r_y(2u_{0,1}^n - 2u_{00}^n) + 10\Delta t \\ u_{0k}^{n+1} &= u_{0k}^n + r_x(2u_{1k}^n - 2u_{0k}^n) + r_y(u_{0,k+1}^n - 2u_{0k}^n + u_{0,k-1}^n) + 10\Delta t. \\ u_{0K}^{n+1} &= u_{0K}^n + r_x(2u_{1K}^n - 2u_{0K}^n) + r_y(-2u_{0K}^n + 2u_{0,K-1}^n) + 10\Delta t \\ u_{jK}^{n+1} &= u_{jK}^n + r_x(u_{j+1,K}^n - 2u_{jK}^n + u_{j-1,K}^n) + r_y(-2u_{jK}^n + 2u_{j,K-1}^n) + 10\Delta t \\ u_{jK}^{n+1} &= u_{jK}^n + r_x(-2u_{jK}^n + 2u_{j-1,K}^n) + r_y(-2u_{jK}^n + 2u_{j,K-1}^n) + 10\Delta t \\ u_{jk}^{n+1} &= u_{jk}^n + r_x(-2u_{jk}^n + 2u_{j-1,k}^n) + r_y(u_{j,k+1}^n - 2u_{jk}^n + u_{j,k-1}^n) + 10\Delta t \\ u_{j0}^{n+1} &= u_{j0}^n + r_x(-2u_{j0}^n + 2u_{j-1,0}^n) + r_y(2u_{j,1}^n - 2u_{j0}^n) + 10\Delta t \\ u_{j0}^{n+1} &= u_{j0}^n + r_x(u_{j+1,0}^n - 2u_{j0}^n + u_{j-1,0}^n) + r_y(2u_{j,1}^n - 2u_{j0}^n) + 10\Delta t.\end{aligned}$$

Os gráficos abaixo mostram as soluções numéricas correspondendo a  $t = 0$  e  $t = 0.84$ ; foram usados  $\Delta x = \Delta y = 0.05$ ,  $\Delta t = 0.12$ .





Os gráficos abaixo mostram a aproximação à solução exata ao longo das retas  $x = 1/2$  e  $t = 1$  quando variamos  $\Delta t$ .

Novamente vemos que, quando  $\Delta x$  e  $\Delta y$  estão fixos, existe um valor  $\Delta t_0$  para  $\Delta t$  que produz uma melhor aproximação à solução exata, e valores  $\Delta t$  menores produzem resultados desastrosos.

Por causa das condições de contorno que traduzem um isolamento térmico, temos que a solução exata

$$v(x, y, t) = \frac{1}{4} - \exp(-4\pi^2 \nu t) \frac{\cos(2\pi x) + \cos(2\pi y)}{4} + \exp(-8\pi^2 \nu t) \frac{\cos(2\pi x) \cos(2\pi y)}{4}$$

converge à superfície constante igual a  $1/4$  ao  $t \rightarrow \infty$ , o que pode ser observado acima.

**Exemplo 8.5 :** O esquema de Crank-Nicolson para o problema de difusão 2D

Considere o problema

$$\begin{aligned} v_t &= \nu(v_{xx} + v_{yy}) + F(x, y, t), & (x, y) \in (0, 1) \times (0, 1), t > 0 \\ v(x, y, 0) &= f(x, y), & (x, y) \in [0, 1] \times [0, 1] \\ v(0, y, t) &= v(1, y, t) = 0, & y \in (0, 1), t > 0 \\ v(x, 0, t) &= v(x, 1, t) = 0, & x \in (0, 1), t > 0 \end{aligned}$$

O esquema de Crank-Nicolson para aproximação da EDP é:

$$\frac{u_{jk}^{n+1} - u_{jk}^n}{\Delta t} = \nu \left( \frac{\delta_x^2}{(\Delta x)^2} + \frac{\delta_y^2}{(\Delta y)^2} \right) \frac{u_{jk}^{n+1} + u_{jk}^n}{2} + \frac{F_{jk}^{n+1} + F_{jk}^n}{2}$$

e então temos

$$\left( 1 - \frac{r_x \delta_x^2}{2} - \frac{r_y \delta_y^2}{2} \right) u_{jk}^{n+1} - \frac{\Delta t F_{jk}^{n+1}}{2} = \left( 1 + \frac{r_x \delta_x^2}{2} + \frac{r_y \delta_y^2}{2} \right) u_{jk}^n + \frac{\Delta t F_{jk}^n}{2}$$

ou ainda

$$\begin{aligned} -\frac{r_y}{2} u_{j,k-1}^{n+1} - \frac{r_x}{2} u_{j-1,k}^{n+1} + (1 + r_x + r_y) u_{jk}^{n+1} - \frac{r_x}{2} u_{j+1,k}^{n+1} - \frac{r_y}{2} u_{j,k+1}^{n+1} - \frac{\Delta t F_{jk}^{n+1}}{2} = \\ + \frac{r_y}{2} u_{j,k-1}^n + \frac{r_x}{2} u_{j-1,k}^n + (1 - r_x - r_y) u_{jk}^n + \frac{r_x}{2} u_{j+1,k}^n + \frac{r_y}{2} u_{j,k+1}^n + \frac{\Delta t F_{jk}^n}{2} \end{aligned}$$

Se  $F = 0$ , escrevemos como  $Au^{n+1} = Bu^n, n = 0, 1, 2, \dots$  onde

$$u^n = \begin{bmatrix} u_{11}^n & u_{21}^n & u_{31}^n & u_{12}^n & u_{22}^n & u_{32}^n & u_{13}^n & u_{23}^n & u_{33}^n \end{bmatrix}^T$$

$$A = \begin{bmatrix} d & -r_x/2 & & -r_y/2 & & & & & \\ -r_x/2 & d & -r_x/2 & & & & & & \\ & -r_x/2 & d & & & & -r_y/2 & & \\ -r_y/2 & & & d & -r_x/2 & & & -r_y/2 & \\ & -r_y/2 & & -r_x/2 & d & -r_x/2 & & & -r_y/2 \\ & & -r_y/2 & & -r_x/2 & d & & & \\ & & & -r_y/2 & & & d & -r_x/2 & \\ & & & & -r_y/2 & & -r_x/2 & d & -r_x/2 \\ & & & & & -r_y/2 & & -r_x/2 & d \end{bmatrix}$$

$$B = \begin{bmatrix} c & r_x/2 & & r_y/2 & & & & & \\ r_x/2 & c & r_x/2 & & r_y/2 & & & & \\ & r_x/2 & c & & & r_y/2 & & & \\ r_y/2 & & & c & r_x/2 & & r_y/2 & & \\ & r_y/2 & & r_x/2 & c & r_x/2 & & r_y/2 & \\ & & r_y/2 & & r_x/2 & c & & & r_y/2 \\ & & & r_y/2 & & & c & r_x/2 & \\ & & & & r_y/2 & & r_x/2 & c & r_x/2 \\ & & & & & r_y/2 & & r_x/2 & c \end{bmatrix}$$

onde  $d = 1 + r_x + r_y$ ,  $c = 1 - r_x - r_y$  e  $u^0$  é determinado pela condição inicial.

**Exemplo 8.6 :** Análise de Fourier do esquema de Crank-Nicolson para eq difusão 2D

Desconsiderando o termo não -homogêneo, temos

$$\begin{aligned} \left( 1 - \frac{r_x}{2}(2 \cos(\xi) - 2) - \frac{r_y}{2}(2 \cos(\eta) - 2) \right) \hat{u}^{n+1} = \\ \left( 1 + \frac{r_x}{2}(2 \cos(\xi) - 2) + \frac{r_y}{2}(2 \cos(\eta) - 2) \right) \hat{u}^n \end{aligned}$$

e então

$$(1 + 2r_x \sin^2(\xi/2) + 2r_y \sin^2(\eta/2)) \hat{u}^{n+1} = (1 - 2r_x \sin^2(\xi/2) - 2r_y \sin^2(\eta/2)) \hat{u}^n$$

para  $n = 0, 1, 2, \dots$ . O símbolo desse esquema numérico é

$$\rho(\xi, \eta) = \frac{1 - 2r_x \sin^2(\xi/2) - 2r_y \sin^2(\eta/2)}{1 + 2r_x \sin^2(\xi/2) + 2r_y \sin^2(\eta/2)}$$

e então claramente temos  $|\rho(\xi, \eta)| \leq 1 \quad \forall \xi, \eta \in [-\pi, \pi]$  e o esquema é incondicionalmente estável na norma energia.

**Tarefa:** Mostre que, a exemplo do caso unidimensional, o esquema 2D de Crank-Nicolson é de segunda ordem no tempo e no espaço. Definir erro de truncamento usando  $t_{n+1.2} \dots$

**Introdução 8.3 :** Esquemas ADI (Implícitos de Direção Alternada)

Para o problema de difusão unidimensional, esquemas numéricos de dois tipos foram trabalhados:

- $u^{n+1} = Qu^n, n = 0, 1, 2, \dots$ : explícitos, de mais fácil implementação numérica mas apenas condicionalmente estáveis;



$$B_2 = \begin{bmatrix} 1 - r_x & r_x/2 & & & & \\ r_x/2 & 1 - r_x & r_x/2 & & & \\ & r_x/2 & 1 - r_x & r_x/2 & & \\ & & \dots & \dots & r_x/2 & \\ & & & r_x/2 & 1 - r_x & \\ & & & & & \end{bmatrix}$$

• **Condição inicial:**

$$u_{jk}^0 = f(j\Delta x, k\Delta y),$$

$j = 0, 1, 2, \dots, J; k = 0, 1, 2, \dots, K$ .

**Condições de contorno:** Considerando uma CC de Dirichlet em  $x = 0$ , a implementação mais imediata é

$$u_{0k}^{n+1/2} = g(0, k\Delta y, (n + 1/2)\Delta t).$$

Alternativamente, considerando

$$\begin{aligned} \left(1 - \frac{r_x \delta_x^2}{2}\right) u_{jk}^{n+1/2} &= \left(1 + \frac{r_y \delta_y^2}{2}\right) u_{jk}^n \\ \left(1 + \frac{r_x \delta_x^2}{2}\right) u_{jk}^{n+1/2} &= \left(1 - \frac{r_y \delta_y^2}{2}\right) u_{jk}^{n+1} \end{aligned}$$

e adicionando implica

$$u_{jk}^{n+1/2} = \frac{1}{2} \left(1 - \frac{r_y \delta_y^2}{2}\right) u_{jk}^{n+1} + \frac{1}{2} \left(1 + \frac{r_y \delta_y^2}{2}\right) u_{jk}^n$$

e vemos que uma expressão alternativa é

$$u_{0k}^{n+1/2} = \frac{1}{2} \left(1 - \frac{r_y \delta_y^2}{2}\right) g(0, k\Delta y, (n + 1)\Delta t) + \frac{1}{2} \left(1 + \frac{r_y \delta_y^2}{2}\right) g(0, k\Delta y, n\Delta t).$$

**Pode-se mostrar que a expressão acima aproxima a expressão mais imediata com um erro de ordem  $(\Delta t)^2$ . Dessa forma, ambas as expressões podem ser usadas para implementação da CC em  $x = 0$ .**

Conclusões análogas são derivadas para as demais CC.

• Considerando uma CC de Neumann  $v_y(x, 0, t) = g^N(x, t)$  em  $y = 0$ :

Aproximação de primeira ordem: assumimos

$$u_{j0}^{n+1} = u_{j1}^{n+1} - \Delta t g^N(j\Delta x, (n + 1)\Delta t)$$

e eliminamos  $u_{j0}^{n+1}$  das equações .

Aproximação de segunda ordem: introduzimos  $u_{j,-1}^{n+1}$  tal que

$$u_{j,-1}^{n+1} = u_{j,1}^{n+1} - 2\Delta y g^N(j\Delta x, (n + 1)\Delta t)$$

extendemos as equações vindas da EDP para  $k = 0$  e eliminamos  $u_{j,-1}^{n+1}$  usando a relação acima.

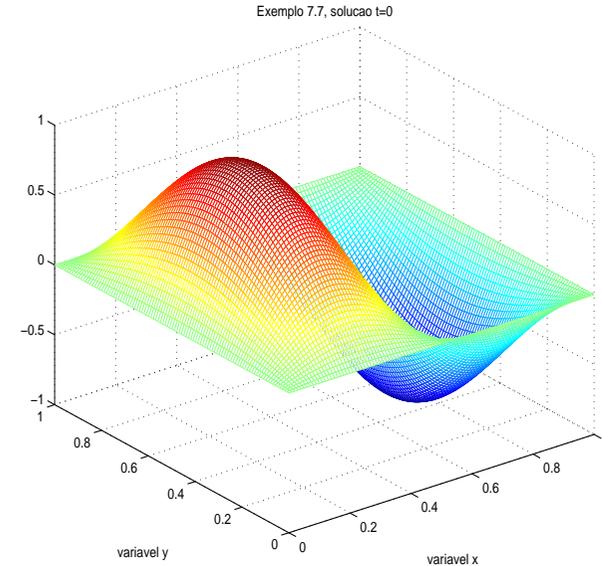
Um enfoque alternativo será proposto na lista de exercícios.

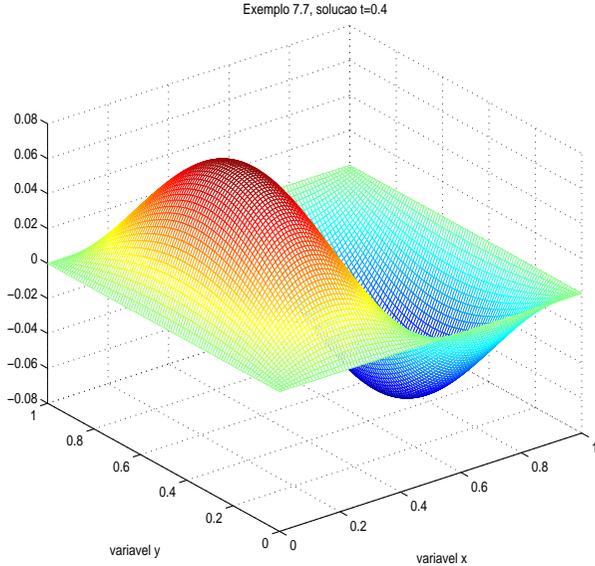
Considere o problema de valores iniciais e condições de contorno

$$\begin{aligned} v_t &= \nu(v_{xx} + v_{yy}), & (x, y) &\in (0, 1) \times (0, 1), t > 0 \\ v(x, y, 0) &= f(x, y), & (x, y) &\in [0, 1] \times [0, 1] \\ v(0, y, t) &= v(1, y, t) = 0, & y &\in (0, 1), t > 0 \\ v(x, 0, t) &= v(x, 1, t) = 0, & x &\in (0, 1), t > 0 \end{aligned}$$

onde  $f(x, y) = \text{sen}(\pi x)\text{sen}(2\pi y)$ . A condição de compatibilidade é claramente satisfeita.

As figuras abaixo mostram as soluções numéricas do problema acima, via esquema de Peaceman-Rachford, para  $t = 0$  e  $t = 0.4$ . Foi usado  $\Delta x = \Delta y = 0.01$ ,  $\Delta t = 0.1$  e  $\nu = 1/6$ .





### Introdução 8.5 : Métodos de Fatorização Aproximada

Considere o esquema 2D de Crank-Nicolson

$$\left(1 - \frac{r_x \delta_x^2}{2} - \frac{r_y \delta_y^2}{2}\right) u_{jk}^{n+1} = \left(1 + \frac{r_x \delta_x^2}{2} + \frac{r_y \delta_y^2}{2}\right) u_{jk}^n$$

que já mostramos (exercício-lista) ser de segunda ordem no tempo e no espaço.

A idéia é reescrever o lado esquerdo, usando a fatorização

$$\left(1 - \frac{r_x \delta_x^2}{2}\right) \left(1 - \frac{r_y \delta_y^2}{2}\right) u_{jk}^{n+1} \quad (\dagger)$$

que é apenas aproximada pois estaríamos introduzindo o termo

$$\frac{r_x r_y}{4} \delta_x^2 \delta_y^2 (u_{jk}^{n+1})$$

ao qual corresponderia, em termos de análise infinitesimal, uma parcela

$$\frac{(\Delta t)^2}{4} \frac{\partial^4 u(x_j, y_k, t_{n+1/2})}{\partial^2 x \partial^2 y} + O((\Delta x)^4, (\Delta y)^4, (\Delta t)^3).$$

Dessa forma, como  $(\dagger)$  deve ser dividida por  $\Delta t$  na análise de consistência, vemos que o esquema

$$\left(1 - \frac{r_x \delta_x^2}{2}\right) \left(1 - \frac{r_y \delta_y^2}{2}\right) u_{jk}^{n+1} = \left(1 + \frac{r_x \delta_x^2}{2} + \frac{r_y \delta_y^2}{2}\right) u_{jk}^n$$

tem a mesma ordem espacial de Crank-Nicolson, mas ordem 1 no tempo.

Entretanto, se adicionarmos o termo

$$\frac{r_x r_y}{4} \delta_x^2 \delta_y^2 (u_{jk}^n)$$

no lado direito e fatorizarmos analogamente, obtemos um esquema numérico que difere de Crank-Nicolson por um termo

$$\frac{r_x r_y}{4} \delta_x^2 \delta_y^2 (u_{jk}^{n+1} - u_{jk}^n).$$

Esse termo é de ordem 3 no tempo e ordem superior no espaço, e dessa forma o esquema resultante

$$\left(1 - \frac{r_x \delta_x^2}{2}\right) \left(1 - \frac{r_y \delta_y^2}{2}\right) u_{jk}^{n+1} = \left(1 + \frac{r_x \delta_x^2}{2}\right) \left(1 + \frac{r_y \delta_y^2}{2}\right) u_{jk}^n,$$

que é o esquema de Peaceman Rachford, possui a mesma ordem de consistência que o esquema 2D de Crank-Nicolson.

Dessa forma, o esquema de Peaceman Rachford é a fatorização aproximada do esquema de Crank-Nicolson.

### Exemplo 8.8 : o Esquema de Douglas-Rachford para Equação de Difusão 2D

O esquema de Douglas-Rachford é a fatorização aproximada do esquema implícito de Euler

$$(1 - r_x \delta_x^2 - r_y \delta_y^2) u_{jk}^{n+1} = u_{jk}^n.$$

A fatorização aproximada nos dá

$$(1 - r_x \delta_x^2)(1 - r_y \delta_y^2) u_{jk}^{n+1} = u_{jk}^n$$

e então o termo

$$r_x r_y \delta_x^2 \delta_y^2 u_{jk}^{n+1}$$

foi adicionado. A exemplo do que fizemos para obter o esquema de Peaceman-Rachford a partir do de Crank-Nicolson, contra-adicionamos o termo

$$r_x r_y \delta_x^2 \delta_y^2 u_{jk}^n$$

no lado direito da equação, e temos

$$(1 - r_x \delta_x^2)(1 - r_y \delta_y^2)u_{jk}^{n+1} = (1 + r_x r_y \delta_x^2 \delta_y^2)u_{jk}^n.$$

A forma mais usada do esquema de **Douglas-Rachford** é

$$\begin{aligned} (1 - r_x \delta_x^2)u_{jk}^{n+1/2} &= (1 + r_y \delta_y^2)u_{jk}^n \\ (1 - r_y \delta_y^2)u_{jk}^{n+1} &= u_{jk}^{n+1/2} - r_y \delta_y^2 u_{jk}^n. \end{aligned}$$

## 9 Solução Numérica de Equações Hiperbólicas

Equações ou sistemas de equações hiperbólicas governam fenômenos importantes como fluxos aerodinâmicos, incluindo fluxos através de meio porosos e fluxos atmosféricos.

### Introdução 9.1 : Fluxos Unidimensionais

Um sistema linear geral de equações parciais unidimensionais à coeficientes constantes pode ser escrito

$$V_t = Av_x, x \in \mathbb{R}, t > 0 \quad (46)$$

onde  $v$  é um vetor de incógnitas e  $A$  é uma matriz  $K \times K$  de coeficientes. Se  $A$  é uma matriz diagonalizável, então dizemos que o sistema acima é **fortemente hiperbólico**. Neste caso, existe uma matriz  $S$  de autovetores tal que  $A = S^{-1}\Lambda S$ , onde  $\Lambda$  é uma matriz diagonal. Assim,

$$Sv_t = SAV_x = \Lambda Sv_x$$

e então, sendo  $u(x, t) = Sv(x, t)$ , temos

$$\frac{\partial u_i}{\partial t} = \lambda_i \frac{\partial u_i}{\partial x}, i = 0, 1, \dots, K$$

onde  $\lambda_i$  são os elementos diagonais de  $\Lambda$  (autovalores de  $A$ ).

### Introdução 9.2 : Curvas Características

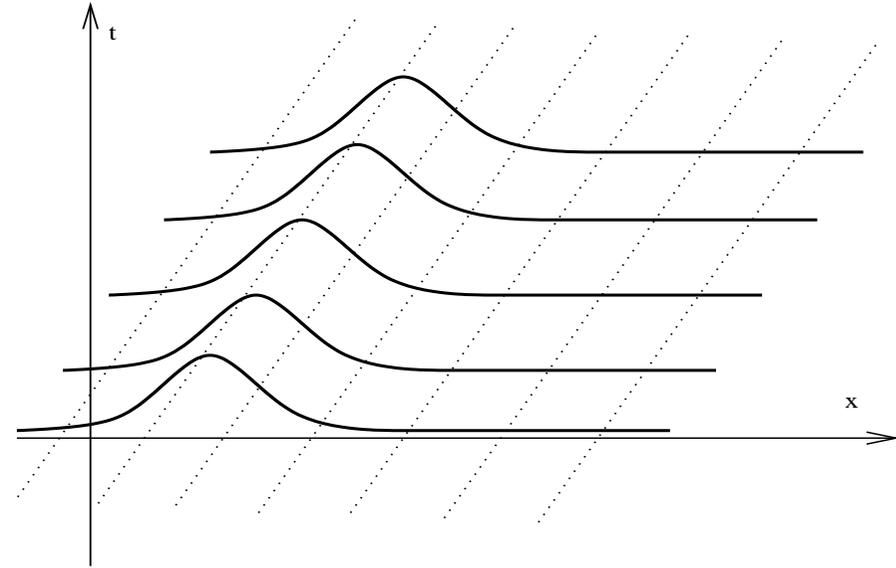
Considere o problema da equação de onda unidimensional

$$\begin{aligned} v_t + av_x &= 0, & x \in \mathbb{R}, t > 0 \\ v(x, 0) &= f(x), & x \in \mathbb{R} \end{aligned}$$

Vemos que a primeira equação pode ser escrita como

$$\begin{bmatrix} a & 1 \end{bmatrix} \begin{bmatrix} v_x \\ v_t \end{bmatrix} = 0$$

Ou seja, a solução é constante ao longo das retas  $x - at = c$ , onde  $c \in \mathbb{R}$  são constantes. Dessa forma, a solução exata é  $v(x, t) = f(x - at)$ ,  $x \in \mathbb{R}, t \geq 0$ .



**Definição 9.1** O domínio de dependência analítico da solução em um ponto  $(x, t)$  (no domínio da EDP) é o ponto  $s_0 = x - at$ .

### Introdução 9.3 : Solução Numérica de $v_t + av_x = 0$

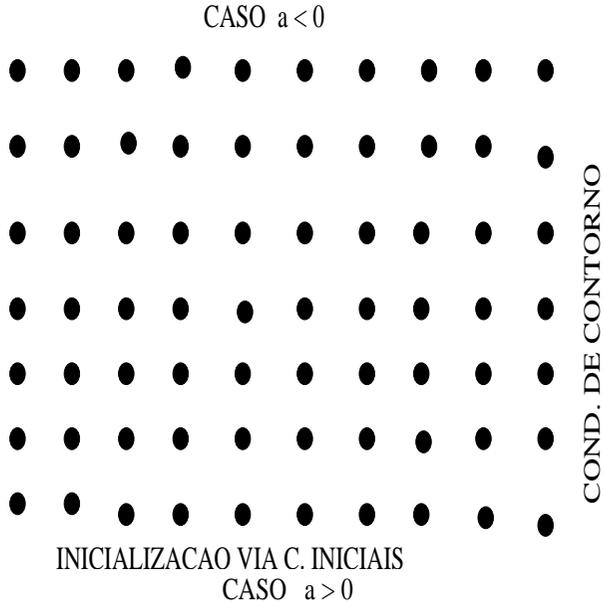
Esquemas unidimensionais:

esquema	fórmula	símbolo	estabilidade
FTFS	$u_k^{n+1} = u_k^n - R(u_{k+1}^n - u_k^n)$	$1 + 2R\text{sen}^2(\xi/2) - iR\text{sen}(\xi)$	$-1 \leq R \leq 0$
FTBS	$u_k^{n+1} = u_k^n - R(u_k^n - u_{k-1}^n)$	$1 - 2R\text{sen}^2(\xi/2) - iR\text{sen}(\xi)$	$0 \leq R \leq 1$
FTCS	$u_k^{n+1} = u_k^n - (R/2)(u_{k+1}^n - u_{k-1}^n)$	$1 - iR\text{sen}(\xi)$	$R = 0$
BTFS	$(1 - R)u_k^{n+1} + Ru_{k+1}^{n+1} = u_k^n$	$\frac{1}{1 - 2R\text{sen}^2(\xi/2) + iR\text{sen}(\xi)}$	$R \leq 0$ ou $R \geq 1$

### Implementação numérica:

Apesar de o problema ser definido numa região infinita  $[0, +\infty) \times [0, +\infty)$  OU  $(-\infty, 0] \times [0, +\infty]$ , a implementação numérica assumirá  $(x, t) \in [0, K\Delta x] \times [0, N\Delta t]$  ou  $(x, t) \in [-K\Delta x, 0] \times [0, N\Delta t]$ .

Estrutura de dados :  $v(n, k)$ ,  $n = 1, 2, 3, \dots, N + 1$ ;  $k = 1, 2, 3, \dots, K + 1$ .



Por exemplo: para o esquema FTFS, o domínio de dependência numérica  $D_n$  de um ponto  $(k\Delta x, n\Delta t)$  é o intervalo  $[k\Delta x, (k+n)\Delta x]$ .

Por exemplo: para o esquema FTBS, o domínio de dependência numérica  $D_n$  de um ponto  $(k\Delta x, n\Delta t)$  é o intervalo  $[(k-n)\Delta x, k\Delta x]$ .

**Exemplo 9.1** : Implementação do esquema FTBS para  $v_t + av_x = 0$

Considere o problema

$$\begin{aligned} v_t + av_x &= 0, & 0 < x < 3, t > 0 \\ v(x, 0) &= f(x), & 0 \leq x \leq 3 \\ v(0, t) &= g(t), & t \geq 0 \end{aligned}$$

onde  $a = 1/6$ ,  $f(x) = \text{sen}(\pi x)$ ,  $g(t) \equiv 0$ .

**Condição de compatibilidade:**

$$v(0, 0) = f(0) = g(0)$$

que é claramente verificada.

**Inicialização via CI:**  $u_k^0 = f(x_k), k = 0, 1, \dots, K$ .

**Inicialização via CC:**  $u_0^n = g(t_n), n = 0, 1, \dots, N$ .

**Iteração principal:**

for  $n = 0 : N\_max - 1$

for  $k = 1 : K\_max$

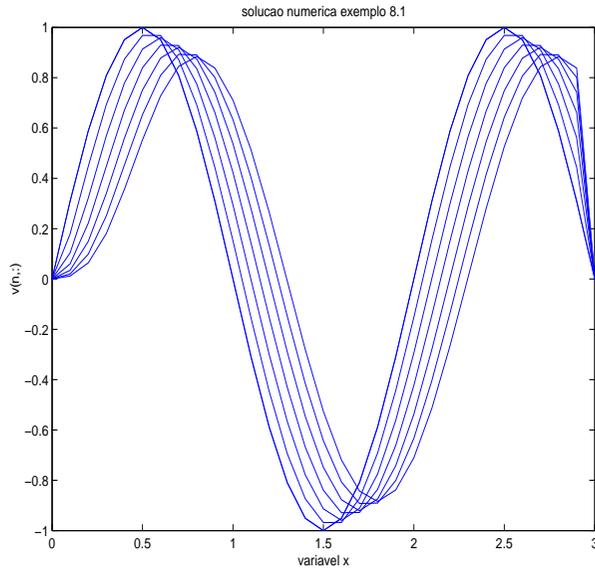
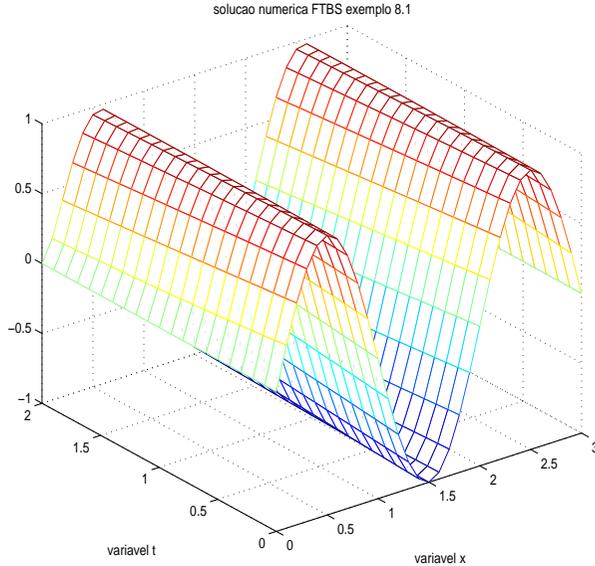
$$u_k^{n+1} = (1 - R)u_k^n + Ru_{k-1}^n$$

end

end

As figuras abaixo mostram o resultado da implementação numérica (Matlab) do esquema acima, para  $a = 1/6$ ,  $\Delta t = \Delta x = 1/10$ .

**Definição 9.2** O domínio de dependência de um esquema numérico em um ponto  $(k\Delta x, n\Delta t)$  da discretização é o intervalo ao longo da condição inicial que é usado na computação de  $u_k^n$ .



**Definição 9.3 :** a Condição de Courant-Friedrichs-Levy

Uma equação diferencial parcial e um esquema numérico associado são ditos satisfazerem a condição de Courant-Friedrichs-Levy (CFL) se para todo ponto  $P$  do domínio da EDP temos que o domínio de dependência analítica  $D_a(P)$  está contido

no domínio de dependência numérica  $D_n(P)$ .

**Lema 9.1 :** Aplicação da condição de CFL

A condição de Courant-Friedrichs-Levy é necessária para a convergência de um esquema numérico.

**Justificativa:** Caso contrário, existe um ponto  $P$  tal que  $D_a(P) \not\subseteq D_n(P)$ . Seja  $Q \in D_a(P)$  tal que  $Q \notin D_n(P)$ . A solução analítica no ponto  $P$  depende da informação contida em uma vizinhança do ponto  $Q$ , mas essa informação nunca será captada pelo esquema numérico, e portanto não poderá haver convergência.

**Consequência:** podemos usar CFL para determinar condições necessárias para a convergência de esquemas consistente, isto é, temos uma ferramenta para encontrar condições necessárias para a estabilidade.

Por exemplo, para o esquema FTBS e um ponto  $(x, t) = (k\Delta x, n\Delta t)$  temos  $D_a(x, t) = \{s_0\}$ , onde  $s_0 = k\Delta x - an\Delta t = (k - Rn)\Delta x$ . Dessa forma, é necessário que

$$(k - n)\Delta x \leq (k - Rn)\Delta x \leq k\Delta x$$

e então

$$k - n \leq k - Rn \leq k \Rightarrow -n \leq -Rn \leq 0 \Rightarrow -1 \leq -R \leq 0 \Rightarrow 0 \leq R \leq 1$$

como já sabemos.

**Introdução 9.4 :** Esquema de Lax-Wendroff para  $v_t + av_x = 0$

A proposta é modificar o esquema FTCS da tabela dada anteriormente, de maneira a estabilizá-lo.

**Estratégia:**

$$v_{tt} = (-av_x)_t = -av_{xt} = -av_{tx} = -a(-av_x)_x = a^2v_{xx}$$

**Equacionamento e análise de consistência:**

$$\begin{aligned} v(x_k, t_{n+1}) &= v(x_k, t_n) + v_t(x_k, t_n)\Delta t + v_{tt}(x_k, t_n)\frac{(\Delta t)^2}{2} + O((\Delta t)^3) \\ v(x_k, t_{n+1}) &= v(x_k, t_n) - av_x(x_k, t_n)\Delta t + a^2v_{xx}(x_k, t_n)\frac{(\Delta t)^2}{2} + O((\Delta t)^3) \\ v(x_k, t_{n+1}) &= v(x_k, t_n) - a\left(\frac{v(x_{k+1}, t_n) - v(x_{k-1}, t_n)}{2\Delta x} + O((\Delta x)^2)\right)\Delta t + \\ &+ a^2\left(\frac{v(x_{k+1}, t_n) - 2v(x_k, t_n) + v(x_{k-1}, t_n)}{(\Delta x)^2} + O((\Delta x)^2)\right)\frac{(\Delta t)^2}{2} + O((\Delta t)^3) \end{aligned}$$

Dessa forma, desconsiderando-se erros de arredondamento, o esquema numérico

$$u_k^{n+1} = u_k^n - \frac{R}{2}\delta^1(u_k^n) + \frac{R^2}{2}\delta^2(u_k^n),$$

onde  $R = a\Delta t/\Delta x$ , possui um erro de truncamento

$$\tau_k^n = \frac{1}{\Delta t} [O((\Delta x)2\Delta t, (\Delta x)^2(\Delta t)^2, (\Delta t)^3)]$$

ou seja, de primeira ordem no tempo e segunda no espaço.

**Análise de estabilidade:** Aplicando a TDF,

$$\hat{u}^{n+1} = \left(1 - \frac{R}{2}(2i\text{sen}(\xi)) + \frac{R^2}{2}(-2 + 2\cos(\xi))\right) \hat{u}^n,$$

ou seja,

$$\rho(\xi) = 1 - 2R^2\text{sen}^2(\xi/2) - iR\text{sen}(\xi).$$

Vemos que

$$\begin{aligned} |\rho(\xi)|^2 &= (1 - 2R^2\text{sen}^2(\xi/2))^2 + R^2\text{sen}^2(\xi) \\ |\rho(\xi)|^2 &= 1 - 4R^2\text{sen}^2(\xi/2) + 4R^4\text{sen}^4(\xi/2) + R^2\text{sen}^2(\xi) \end{aligned}$$

e como  $\text{sen}^2(\xi) = 4\text{sen}^2(\xi/2)\cos(\xi/2)$  temos

$$\begin{aligned} -4\text{sen}^2(\xi/2) + \text{sen}^2(\xi) &= -4\text{sen}^2(\xi/2) + 4\text{sen}^2(\xi/2)\cos^2(\xi/2) = \\ &= -4\text{sen}^2(\xi/2)(1 - \cos^2(\xi/2)) = -4\text{sen}^4(\xi/2) \end{aligned}$$

e assim

$$|\rho(\xi)|^2 = 1 - 4R^2\text{sen}^4(\xi/2) + 4R^4\text{sen}^4(\xi/2).$$

Como possíveis máximos estão em  $\xi=0, \pm\pi$ , calculamos

$$\begin{aligned} |\rho(0)|^2 &= 1 \\ |\rho(\pm\pi)|^2 &= (1 - 2R^2)^2 \end{aligned}$$

e dessa forma concluímos que o esquema de Lax-Wendroff é estável na norma-energia se e somente se  $-1 \leq 1 - 2R^2$ , isto é,  $R^2 \leq 1$ , isto é,  $|R| \leq 1$ .

**Introdução 9.5 :** o Esquema de Crank-Nicolson para Equação de Onda 1D

O esquema de Crank-Nicolson para equação de onda unidimensional:

$$\begin{aligned} \frac{u_k^{n+1} - u_k^n}{\Delta t} + a \left( \frac{u_{k+1}^{n+1} - u_{k-1}^{n+1}}{4\Delta x} + \frac{u_{k+1}^n - u_{k-1}^n}{4\Delta x} \right) &= 0 \\ u_k^{n+1} - u_k^n + R \left( \frac{u_{k+1}^{n+1} - u_{k-1}^{n+1}}{4} + \frac{u_{k+1}^n - u_{k-1}^n}{4} \right) &= 0 \\ -\frac{R}{4}u_{k-1}^{n+1} + u_k^{n+1} + \frac{R}{4}u_{k+1}^{n+1} &= \frac{R}{4}u_{k-1}^n + u_k^n - \frac{R}{4}u_{k+1}^n \end{aligned}$$

Aplicando a TDF, temos

$$\begin{aligned} \left(1 + \frac{R}{4}(2i\text{sen}(\xi))\right) \hat{u}^{n+1} &= \left(1 - \frac{R}{4}(2i\text{sen}(\xi))\right) \hat{u}^n \\ \rho(\xi) &= \frac{1 - iR\text{sen}(\xi)/2}{1 + iR\text{sen}(\xi)/2} \end{aligned}$$

de onde concluímos estabilidade marginal incondicional, pois  $|\rho(\xi)|^2 \equiv 1$ .

**Introdução 9.6 :** Equações Parciais Hiperbólicas 2D

A equação diferencial parcial hiperbólica em duas dimensões é

$$v_t + av_x + bv_y = 0$$

que possui uma condição inicial do tipo  $v(x, y, 0) = f(x, y)$ .

Uma solução desse problema inicial é  $v(x, y, t) = f(x - at, y - bt)$ .

A solução é então constante ao longo da reta  $x - at = x_0$ ,  $y - at = y_0$ , e novamente temos uma *propagação*.

Dependendo da direção dessa reta (sinais de  $a$  e  $b$ ), condições de contorno como, por exemplo,

$$\begin{aligned} v(x, 0, t) &= g_1(x, t) & x \in [0, +\infty) & t \geq 0 \\ v(0, y, t) &= g_2(y, t) & y \in [0, +\infty) & t \geq 0 \end{aligned}$$

devem ser adotadas para domínios espaciais finitos ou semi-finitos.

**Exemplo 9.2 :** Esquema FTBS para  $v_t + av_x + bv_y = 0$

Considere o problema

$$\begin{aligned} v_t + av_x + bv_y &= 0, & (x, y) \geq 0, t > 0 \\ v(x, y, 0) &= f(x, y), & x \geq 0, y \geq 0 \\ v(x, 0, t) &= g_1(x, t) & x \geq 0, t \geq 0 \\ v(0, y, t) &= g_2(y, t) & y \geq 0, t \geq 0 \end{aligned}$$

onde  $a = b = 2$ ,  $f(x, y) = \text{sen}(\pi x/3)\text{sen}(\pi y/4)$ ,  $g_1(x, t) = g_2(y, t) \equiv 0$ .

O esquema numérico usa aproximações de primeira ordem; avançada para  $v_t$  e retrasadas para  $v_x$  e  $v_y$ :

$$\frac{u_{jk}^{n+1} - u_{jk}^n}{\Delta t} + a \frac{u_{jk}^n - u_{j-1,k}^n}{\Delta x} + b \frac{u_{jk}^n - u_{j,k-1}^n}{\Delta y}$$

e então definimos

$$R_x = a \frac{\Delta t}{\Delta x}, R_y = b \frac{\Delta t}{\Delta y}$$

para escrever na forma compacta

$$u_{jk}^{n+1} = (1 - R_x \delta_{x-} - R_y \delta_{y-}) u_{jk}^n.$$

Aplicando a TDF em duas dimensões, temos

$$\rho(\xi, \eta) = 1 - R_x(1 - e^{-i\xi}) - R_y(1 - e^{-i\eta})$$

e assim

$$|\rho(\xi, \eta)|^2 = (1 - 2R_x \text{sen}^2(\xi/2) - 2R_y \text{sen}^2(\eta/2))^2 + (R_x \text{sen}(\xi) + R_y \text{sen}(\eta))^2.$$

Procurando por pontos de máximo em  $[-\pi, \pi] \times [-\pi, \pi]$ :

$$\frac{\partial |\rho|^2}{\partial \xi} = (1 - 2R_x \text{sen}^2(\xi/2) - 2R_y \text{sen}^2(\eta/2))(-2R_x) \text{sen}(\xi/2) \cos(\xi/2) + (R_x \text{sen}(\xi) + R_y \text{sen}(\eta)) R_x \cos(\xi) = 0$$

$$\frac{\partial |\rho|^2}{\partial \eta} = (1 - 2R_x \text{sen}^2(\xi/2) - 2R_y \text{sen}^2(\eta/2))(-2R_y) \text{sen}(\eta/2) \cos(\eta/2) + (R_x \text{sen}(\xi) + R_y \text{sen}(\eta)) R_y \cos(\eta) = 0$$

ou seja

$$\begin{aligned} -(1 - 2R_x \text{sen}^2(\xi/2) - 2R_y \text{sen}^2(\eta/2)) \text{sen}(\xi) + (R_x \text{sen}(\xi) + R_y \text{sen}(\eta)) \cos(\xi) &= 0 \\ -(1 - 2R_x \text{sen}^2(\xi/2) - 2R_y \text{sen}^2(\eta/2)) \text{sen}(\eta) + (R_x \text{sen}(\xi) + R_y \text{sen}(\eta)) \cos(\eta) &= 0 \end{aligned}$$

o que implica

$$\begin{aligned} (R_x \cos(\xi) - 1 + 2R_x \text{sen}^2(\xi/2) + 2R_y \text{sen}^2(\eta/2)) \text{sen}(\xi) + R_y \cos(\xi) \text{sen}(\eta) &= 0 \\ R_x \cos(\eta) \text{sen}(\xi) + (R_y \cos(\eta) - 1 + 2R_x \text{sen}^2(\xi/2) + 2R_y \text{sen}^2(\eta/2)) \text{sen}(\eta) &= 0 \end{aligned}$$

e possíveis máximos existem em  $(0, 0)$ ,  $(0, \pm\pi)$ ,  $(\pm\pi, 0)$  e  $(\pm\pi, \pm\pi)$ , entre outros.

Entretanto

$$\begin{aligned} |\rho(0, 0)| &= 1, |\rho(0, \pm\pi)|^2 = (1 - 2R_y)^2, |\rho(\pm\pi, 0)|^2 = (1 - 2R_x)^2 \\ |\rho(\pm\pi, \pm\pi)|^2 &= (1 - 2R_x - 2R_y)^2 \end{aligned}$$

e condições necessárias para a convergência são :

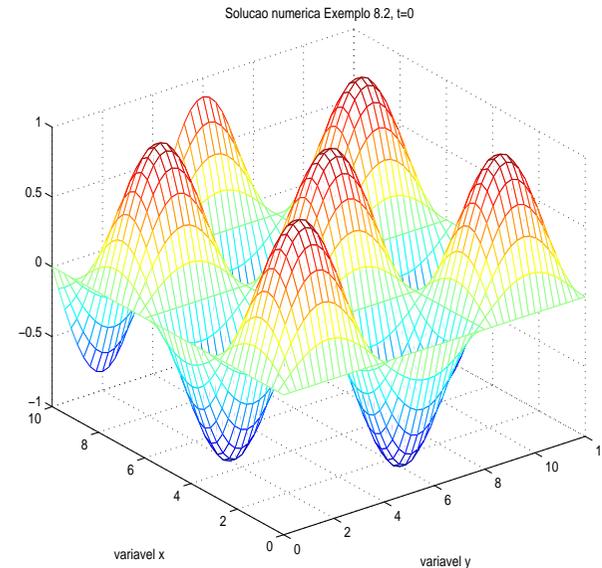
$$|1 - 2R_x| \leq 1 \Leftrightarrow 0 \leq R_x \leq 1$$

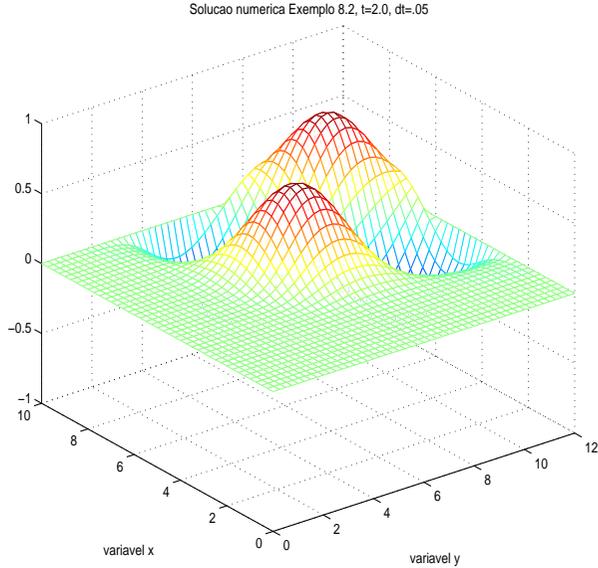
$$|1 - 2R_y| \leq 1 \Leftrightarrow 0 \leq R_y \leq 1$$

$$|1 - 2R_x - 2R_y| \leq 1 \Leftrightarrow 0 \leq R_x + R_y \leq 1$$

e portanto  $0 \leq R_x + R_y \leq 1$  é condição necessária para a estabilidade e portanto para a convergência desse esquema.

As figuras abaixo mostram o resultado da implementação numérica (Matlab) para os instantes  $t = 0$  e  $t = 2.0$ , respectivamente. Foi usado  $\Delta x = \Delta y = 1/4$ ,  $\Delta t = 1/20$ .





**Definição 9.4** O domínio de dependência numérica da solução (depende do esquema numérico) em um ponto  $(j\Delta x, k\Delta y, n\Delta t)$  é o menor retângulo de  $\mathbb{R}^2$  que contém todos os pontos das CI dos quais a computação de  $u_{jk}^n$  depende.

**Lema 9.2** : Aplicação da condição de CFL em duas dimensões

A condição de Courant-Friedrichs-Levy é necessária para a convergência de um esquema numérico para equação parcial hiperbólica bidimensional

**Justificativa:** a mesma.

**Exemplo:** temos 3 categorias a distinguir:

•  $u_{jk}^{n+1} = a_1 u_{j-1,k}^n + a_2 u_{j,k-1}^n + a_3 u_{jk}^n + a_4 u_{j+1,k}^n + a_5 u_{j,k+1}^n$   
 com  $D_n = [(j-n)\Delta x, (j+n)\Delta x] \times [(k-n)\Delta y, (k+n)\Delta y]$   
 CFL implica

$$\begin{aligned} (j-n)\Delta x &\leq (x-at) \leq (j+n)\Delta x \\ (k-n)\Delta y &\leq (y-bt) \leq (k+n)\Delta y \end{aligned}$$

implica

$$\begin{aligned} (j-n)\Delta x &\leq (j-R_x n)\Delta x \leq (j+n)\Delta x \\ (k-n)\Delta y &\leq (k-R_y n)\Delta y \leq (k+n)\Delta y \end{aligned}$$

implica

$$\begin{aligned} -n &\leq -R_x n \leq n \Leftrightarrow -1 \leq R_x \leq 1 \\ -n &\leq -R_y n \leq n \Leftrightarrow -1 \leq R_y \leq 1 \end{aligned}$$

•  $u_{jk}^{n+1} = a_3 u_{jk}^n + a_4 u_{j+1,k}^n + a_5 u_{j,k+1}^n$   
 com  $D_n = [j\Delta x, (j+n)\Delta x] \times [k\Delta y, (k+n)\Delta y]$ .  
 Analogamente, CFL implica  $-1 \leq R_x \leq 0, -1 \leq R_y \leq 0$ .  
 •  $u_{jk}^{n+1} = a_1 u_{j-1,k}^n + a_2 u_{j,k-1}^n + a_3 u_{jk}^n$   
 com  $D_n = [(j-n)\Delta x, j\Delta x] \times [(k-n)\Delta y, k\Delta y]$ .  
 Analogamente, CFL implica  $0 \leq R_x \leq 1, 0 \leq R_y \leq 1$ .

**Introdução 9.7** : Um esquema ADI para  $v_t + av_x + bv_y = 0$

Considere o esquema **localmente unidimensional** para  $v_t + av_x + bv_y = 0$ :

$$\begin{aligned} (1 + R_x \delta_x^0) u_{jk}^{n+1/2} &= u_{jk}^n \\ (1 + R_y \delta_y^0) u_{jk}^{n+1} &= u_{jk}^{n+1/2} \end{aligned}$$

**Exercício:** mostre que esse esquema é consistente de primeira ordem no tempo e segunda no espaço.

Aplicando a TDF em

$$(1 + R_x \delta_x^0) (1 + R_y \delta_y^0) u_{jk}^{n+1} = u_{jk}^n$$

temos

$$(1 + iR_x \text{sen}(\xi))(1 + iR_y \text{sen}(\eta)) \hat{u}^{n+1} = \hat{u}^n$$

e assim

$$\rho(\xi) = \frac{1}{(1 + iR_x \text{sen}(\xi))(1 + iR_y \text{sen}(\eta))}$$

que verifica

$$|\rho(\xi)|^2 = \frac{1}{(1 + R_x^2 \text{sen}^2(\xi))(1 + R_y^2 \text{sen}^2(\eta))}$$

de onde a estabilidade incondicional é concluída.

**Introdução 9.8** : o Esquema de Beam-Warming para  $v_t + av_x + bv_y = 0$

O esquema de Beam-Warming para  $v_t + av_x + bv_y = 0$  é o resultado da fatorização aproximada aplicada ao esquema bidimensional de Crank-Nicolson

$$\frac{u_{jk}^{n+1} - u_{jk}^n}{\Delta t} + \frac{a}{2\Delta x} \delta_x^1 (u_{jk}^{n+1} + u_{jk}^n) + \frac{b}{2\Delta y} \delta_y^1 (u_{jk}^{n+1} + u_{jk}^n) = 0$$

ou

$$\left(1 + \frac{R_x \delta_x^1}{2} + \frac{R_y \delta_y^1}{2}\right) u_{jk}^{n+1} = \left(1 - \frac{R_x \delta_x^1}{2} - \frac{R_y \delta_y^1}{2}\right) u_{jk}^n.$$

Resulta então

$$\left(1 + \frac{R_x \delta_x^1}{2}\right) \left(1 + \frac{R_y \delta_y^1}{2}\right) u_{jk}^{n+1} = \left(1 - \frac{R_x \delta_x^1}{2}\right) \left(1 - \frac{R_y \delta_y^1}{2}\right) u_{jk}^n.$$

Esse esquema é comumente usado na forma:

$$\begin{aligned} \left(1 + \frac{R_x \delta_x^1}{2}\right) u_{jk}^* &= \left(1 - \frac{R_x \delta_x^1}{2}\right) \left(1 - \frac{R_y \delta_y^1}{2}\right) u_{jk}^n \\ \left(1 + \frac{R_y \delta_y^1}{2}\right) u_{jk}^{n+1} &= u_{jk}^* \end{aligned}$$

Aplicando a TDF

$$\begin{aligned} \left(1 + \frac{R_x}{2} i \text{sen}(\xi)\right) \hat{u}^* &= \left(1 - \frac{R_x}{2} i \text{sen}(\xi)\right) \left(1 - \frac{R_y}{2} i \text{sen}(\eta)\right) \hat{u}^n \\ \left(1 + \frac{R_y}{2} i \text{sen}(\eta)\right) \hat{u}^{n+1} &= \hat{u}^* \end{aligned}$$

e então

$$\rho(\xi, \eta) = \frac{(1 - iR_x \text{sen}(\xi)/2)(1 - iR_y \text{sen}(\eta)/2)}{(1 + iR_x \text{sen}(\xi)/2)(1 + iR_y \text{sen}(\eta)/2)}$$

que claramente verifica  $|\rho(\xi, \eta)| \equiv 1$ , e temos estabilidade marginal incondicional.

## 10 Solução numérica de equações elípticas