

Matemática Numérica II

Pedro H A Konzen

31 de março de 2024

Licença

Este trabalho está licenciado sob a Licença Atribuição-CompartilhaIgual 4.0 Internacional Creative Commons. Para visualizar uma cópia desta licença, visite http://creativecommons.org/licenses/by-sa/4.0/deed.pt_BR ou mande uma carta para Creative Commons, PO Box 1866, Mountain View, CA 94042, USA.

Prefácio

O site [notaspedrok.com.br](https://www.notaspedrok.com.br) é uma plataforma que construí para o compartilhamento de minhas notas de aula. Essas anotações feitas como preparação de aulas é uma prática comum de professoras/es. Muitas vezes feitas a rabiscos em rascunhos com validade tão curta quanto o momento em que são concebidas, outras vezes, com capricho de um diário guardado a sete chaves. Notas de aula também são feitas por estudantes - são anotações, fotos, prints, entre outras formas de registros de partes dessas mesmas aulas. Essa dispersão de material didático sempre me intrigou e foi o que me motivou a iniciar o site.

Com início em 2018, o site contava com apenas três notas incipientes. De lá para cá, conforme fui expandido e revisando os materiais, o site foi ganhando acessos de vários locais do mundo, em especial, de países de língua portuguesa. No momento, conta com 13 notas de aula, além de minicursos e uma coleção de vídeos e áudios.

As notas de **Matemática Numérica II** abordam tópicos introdutórios sobre métodos numéricos para derivação e integração de funções e resolução de equações diferenciais. Códigos exemplos são apresentados em linguagem [Python](#).

Aproveito para agradecer a todas/os que de forma assídua ou esporádica contribuem com correções, sugestões e críticas! ;)

Pedro H A Konzen

<https://www.notaspedrok.com.br>

Conteúdo

Capa	i
Licença	ii
Prefácio	iii
Sumário	vi
1 Derivação	1
1.1 Derivadas de Primeira Ordem	1
1.1.1 Diferenças Finitas por Polinômio de Taylor	3
1.2 Derivadas de Segunda Ordem	9
1.3 Diferenças Finitas por Polinômios Interpoladores	12
1.3.1 Fórmulas de dois pontos	13
1.3.2 Fórmulas de cinco pontos	15
2 Técnicas de extrapolação	18
2.1 Extrapolação de Richardson	18
2.1.1 Sucessivas extrapolações	23
2.1.2 Exercícios	24
3 Integração	26
3.1 Regras de Newton-Cotes	26
3.1.1 Regras de Newton-Cotes Fechadas	27
3.1.2 Regras de Newton-Cotes Abertas	31
3.1.3 Exercício	32
3.2 Regras Compostas de Newton-Cotes	34

3.2.1	Regra Composta do Ponto Médio	35
3.2.2	Regra Composta do Trapézio	37
3.2.3	Regra Composta de Simpson	38
3.3	Quadratura de Romberg	41
3.4	Grau de Exatidão	43
3.4.1	Regra do Ponto Médio	43
3.4.2	Regra de Simpson	45
3.4.3	Exercícios	47
3.5	Quadratura Gauss-Legendre	47
3.5.1	Intervalos de integração arbitrários	54
3.5.2	Exercícios	55
3.6	Quadraturas gaussianas com pesos	56
3.6.1	Quadratura de Gauss-Chebyshev	57
3.6.2	Quadratura de Gauss-Laguerre	58
3.6.3	Quadratura de Gauss-Hermite	59
3.7	Método de Monte Carlo	62
4	Problema de Valor Inicial	64
4.1	Método de Euler	64
4.1.1	Análise Numérica	68
4.1.2	Sistemas de Equações	73
4.1.3	Equações de Ordem Superior	76
4.1.4	Exercícios	78
4.2	Métodos de Taylor de Alta Ordem	80
4.2.1	Análise Numérica	83
4.2.2	Exercícios	85
4.3	Métodos de Runge-Kutta	86
4.3.1	Métodos de Runge-Kutta de ordem 2	87
4.3.2	Método de Runge-Kutta de ordem 4	92
4.3.3	Exercícios	93
4.4	Método de Euler Implícito	95
4.4.1	Análise Numérica	97
4.4.2	Exercícios	99
4.5	Métodos de Passo Múltiplo	101
4.5.1	Métodos de Adams-Bashforth	102
4.5.2	Métodos de Adams-Moulton	107
4.5.3	Exercícios	110
4.6	Método adaptativo com controle de erro	112

4.6.1	Exercícios	115
5	Problema de Valor de Contorno	116
5.1	Método de Diferenças Finitas	116
5.1.1	Exercícios	121
5.2	Método de Elementos Finitos	123
5.2.1	Exercícios	129
5.3	Método de Volumes Finitos	130
5.3.1	Exercícios	135
5.4	Problemas Não-Lineares	136
5.4.1	Exercícios	141
6	Equações Diferenciais Parciais	143
6.1	Equação de Poisson	143
6.1.1	Exercícios	150
6.2	Equação do Calor	151
6.2.1	Exercícios	157
6.3	Equação da Onda	159
6.3.1	Exercício	164
	Respostas dos Exercícios	166
	Bibliografia	174

Capítulo 1

Derivação

Neste capítulo, estudamos os métodos fundamentais de derivação numérica de funções.

1.1 Derivadas de Primeira Ordem

A derivada de uma função f num ponto x é, por definição,

$$f'(x) := \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}. \quad (1.1)$$

Assim sendo e assumindo $h > 0$ ¹ próximo de zero, temos que $f'(x)$ pode ser aproximada pela **fórmula de diferenças finitas**

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \quad (1.2a)$$

$$=: D_h f(x). \quad (1.2b)$$

Geometricamente, isto é análogo a aproximar a declividade da reta tangente ao gráfico da função f no ponto $(x, f(x))$ pela declividade da reta secante ao gráfico da função f pelos pontos $(x, f(x))$ e $(x+h, f(x+h))$ (consulte a Figura 1.1).

¹Para fixar notação, assumiremos $h > 0$ ao longo deste capítulo.

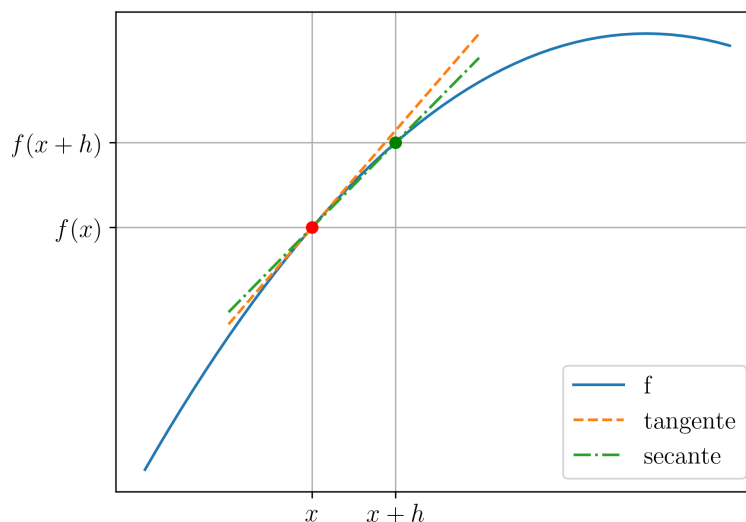


Figura 1.1: Interpretação geométrica da aproximação da derivada pela razão fundamental.

Exemplo 1.1.1. A derivada de $f(x) = \sin(x)$ no ponto $\pi/3$ é $f'(\pi/3) = \cos(\pi/3) = 0.5$. Agora, usando a aproximação pela fórmula de diferenças finitas (1.2), temos

$$f' \left(\frac{\pi}{3} \right) \approx D_h f \left(\frac{\pi}{3} \right) \quad (1.3a)$$

$$= \frac{f \left(\frac{\pi}{3} + h \right) - f \left(\frac{\pi}{3} \right)}{h} \quad (1.3b)$$

$$= \frac{\sin \left(\frac{\pi}{3} + h \right) - \sin \left(\frac{\pi}{3} \right)}{h}. \quad (1.3c)$$

Na Tabela 1.1 temos os valores desta aproximação para diferentes escolhas da passo h .

Tabela 1.1: Valores aproximados da derivada de $f(x) = \sin(x)$ no ponto $x = \pi/3$ usado a fórmula de diferenças finitas (1.2).

h	$Df(\pi/3)$
10^{-1}	4.55902e-1
10^{-2}	4.95662e-1
10^{-3}	4.99567e-1
10^{-5}	4.99996e-1
10^{-7}	5.00000e-1
10^{-10}	5.00000e-1

```

1 import numpy as np
2
3 def dfdx(f, x, h=1e-7):
4     df = (f(x+h) - f(x))/h
5     return df
6
7 f = lambda x: np.sin(x)
8 x = np.pi/3
9 h = 1e-7
10 dfdx = dfdx(f, x, h)

```

1.1.1 Diferenças Finitas por Polinômio de Taylor

Vamos estudar o desenvolvimento de **fórmulas de diferenças finitas** via polinômios de Taylor.

Fórmula de Diferenças Finitas Progressiva de Ordem h

A aproximação por polinômio de Taylor de grau 1 de uma dada função f em torno no ponto x é

$$f(x+h) = f(x) + hf'(x) + O(h^2). \quad (1.4)$$

Isolando $f'(x)$, obtemos

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h). \quad (1.5)$$

Isto nos fornece a chamada **fórmula de diferenças finitas progressiva de ordem h**

$$D_{+,h}f(x) := \frac{f(x+h) - f(x)}{h}. \quad (1.6)$$

Observemos que a ordem da fórmula se refere a do **erro de truncamento com respeito ao passo h** .

Exemplo 1.1.2. Consideremos o problema de aproximar a derivada da função $f(x) = \sin(x)$ no ponto $\pi/3$. Usando a fórmula de diferenças finitas progressiva de ordem h obtemos

$$f'\left(\frac{\pi}{3}\right) \approx D_{+,h}f(x) \quad (1.7a)$$

$$= \frac{f\left(\frac{\pi}{3} + h\right) - f\left(\frac{\pi}{3}\right)}{h} \quad (1.7b)$$

$$= \frac{\sin\left(\frac{\pi}{3} + h\right) - \sin\left(\frac{\pi}{3}\right)}{h}. \quad (1.7c)$$

Na Tabela 1.2 temos os valores desta aproximação para diferentes escolhas de h , bem como, o erro absoluto da aproximação de $f'(\pi/3)$ por $D_{+,h}f(\pi/3)$.

Tabela 1.2: Resultados referente ao Exemplo 1.1.2.

h	$D_{+,h}f(\pi/3)$	$ f'(\pi/3) - D_{+,h}f(\pi/3) $
10^{-1}	4.55902e-1	4.4e-2
10^{-2}	4.95662e-1	4.3e-3
10^{-3}	4.99567e-1	4.3e-4
10^{-5}	4.99996e-1	4.3e-6
10^{-10}	5.00000e-1	4.1e-8

Código 1.1: dfp_h.py

```

1 import numpy as np
2
3 def dfp_h(f, x, h=1e-7):
4     df = (f(x+h) - f(x))/h
5     return df
6

```

```

7 f = lambda x: np.sin(x)
8 x = np.pi/3
9 h = 1e-1
10 dfdx = dfp_h(f, x, h)

```

Observação 1.1.1. (**Erro de Truncamento.**) No Exemplo 1.1.2, podemos observar que o erro absoluto na aproximação de $f'(x)$ por $D_{+,h}f(x)$ decresce conforme a ordem do erro de truncamento para valores moderados de h (consulte a Tabela 1.2). Agora, para valores de h muito pequenos (por exemplo, $h = 10^{-10}$), o erro $|f'(x) - D_{+,h}f(x)|$ não segue mais a tendência de decaimento na mesma ordem do de truncamento. Isto se deve a dominância dos erros de arredondamento para valores muito pequenos de h .

Fórmula de Diferenças Finitas Regressiva de Ordem h

Substituindo h por $-h$ no polinômio de Taylor de grau 1 (1.4), temos

$$f(x - h) = f(x) - hf'(x) + O(h^2), \quad (1.8)$$

donde obtemos a **fórmula de diferenças finitas regressiva de ordem h**

$$D_{-,h}f(x) := \frac{f(x) - f(x - h)}{h}. \quad (1.9)$$

Exemplo 1.1.3. Consideremos o problema de aproximar a derivada da função $f(x) = \sin(x)$ no ponto $\pi/3$. Usando a fórmula de diferenças finitas regressiva de ordem h obtemos

$$f'\left(\frac{\pi}{3}\right) \approx D_{-,h}f(x) \quad (1.10a)$$

$$= \frac{f\left(\frac{\pi}{3}\right) - f\left(\frac{\pi}{3} - h\right)}{h} \quad (1.10b)$$

$$= \frac{\sin\left(\frac{\pi}{3}\right) - \sin\left(\frac{\pi}{3} - h\right)}{h}. \quad (1.10c)$$

Na Tabela 1.3 temos os valores desta aproximação para diferentes escolhas de h , bem como, o erro absoluto da aproximação de $f'(\pi/3)$ por $D_{-,h}f(\pi/3)$.

Código 1.2: dfr_h.py

Tabela 1.3: Resultados referente ao Exemplo 1.1.3.

h	$D_{-,h}f(\pi/3)$	$ f'(\pi/3) - D_{-,h}f(\pi/3) $
10^{-1}	$5.42432e-1$	$4.2e-2$
10^{-2}	$5.04322e-1$	$4.3e-3$
10^{-3}	$5.00433e-1$	$4.3e-4$
10^{-5}	$5.00004e-1$	$4.3e-6$
10^{-10}	$5.00000e-1$	$4.1e-8$

```

1 import numpy as np
2
3 def dfr_h(f, x, h=1e-7):
4     df = (f(x) - f(x-h))/h
5     return df
6
7 f = lambda x: np.sin(x)
8 x = np.pi/3
9 h = 1e-1
10 dfdx = dfr_h(f, x, h)

```

Fórmula de Diferenças Finitas Central de Ordem h^2

Usando o polinômio de Taylor de grau 2 para aproximar a função $f(x)$ em torno de x , temos

$$f(x+h) = f(x) + hf'(x) + \frac{h}{2}f''(x) + O(h^3) \quad (1.11)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h}{2}f''(x) + O(h^3). \quad (1.12)$$

Então, subtraindo esta segunda equação da primeira, temos

$$f(x+h) - f(x-h) = 2hf'(x) + O(h^3). \quad (1.13)$$

Então, isolando $f'(x)$, obtemos

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2), \quad (1.14)$$

Isto nos fornece a chamada **fórmula de diferenças finitas central de ordem h^2**

$$D_{0,h^2}f(x) := \frac{f(x+h) - f(x-h)}{2h}. \quad (1.15)$$

Exemplo 1.1.4. Consideremos o problema de aproximar a derivada da função $f(x) = \sin(x)$ no ponto $\pi/3$. Usando a fórmula de diferenças finitas central de ordem h^2 obtemos

$$f'\left(\frac{\pi}{3}\right) \approx D_{0,h^2}f(x) \quad (1.16a)$$

$$= \frac{f\left(\frac{\pi}{3} + h\right) - f\left(\frac{\pi}{3} - h\right)}{2h} \quad (1.16b)$$

$$= \frac{\sin\left(\frac{\pi}{3} + h\right) - \sin\left(\frac{\pi}{3} - h\right)}{2h}. \quad (1.16c)$$

Na Tabela 1.4 temos os valores desta aproximação para diferentes escolhas de h , bem como, o erro absoluto da aproximação de $f'(\pi/3)$ por $D_{0,h^2}f(\pi/3)$.

Tabela 1.4: Resultados referente ao Exemplo 1.1.4.

h	$D_{0,h^2}f(\pi/3)$	$ f'(\pi/3) - D_{0,h^2}f(\pi/3) $
10^{-1}	4.99167e-1	8.3e-04
10^{-2}	4.99992e-1	8.3e-06
10^{-3}	5.00000e-1	8.3e-08
10^{-5}	5.00000e-1	8.3e-10
10^{-10}	5.00000e-1	7.8e-12

Código 1.3: dfc_h2.py

```

1 import numpy as np
2
3 def dfc_h2(f, x, h=1e-7):
4     df = (f(x+h) - f(x-h))/(2*h)
5     return df
6
7 f = lambda x: np.sin(x)
8 x = np.pi/3
9 h = 1e-1
10 dfdx = dfc_h2(f, x, h)

```

Exercícios

E.1.1.1. Considere a função $f(x) = \cos(x)$. Use a fórmula de diferenças finitas progressiva de ordem h para computar a aproximação de $f'(\pi/3)$ com 5 dígitos significativos corretos.

E.1.1.2. Considere a função $f(x) = \cos(x)$. Use a fórmula de diferenças finitas regressiva de ordem h para computar a aproximação de $f'(\pi/3)$ com 5 dígitos significativos corretos.

E.1.1.3. Considere a função $f(x) = \cos(x)$. Use a fórmula de diferenças finitas central de ordem h^2 para computar a aproximação de $f'(\pi/3)$ com 5 dígitos significativos corretos.

E.1.1.4. Calcule aproximações da derivada de

$$f(x) = \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} + x \quad (1.17)$$

no ponto $x = 2.5$ dadas pelas seguintes fórmulas de diferenças finitas com $h = 10^{-2}$:

- a) progressiva de ordem h .
- b) regressiva de ordem h .
- c) central de ordem h^2 .

E.1.1.5. Considere a seguinte tabela de pontos

i	x_i	y_i
1	2.0	1.86
2	2.1	1.90
3	2.2	2.01
4	2.3	2.16
5	2.4	2.23
6	2.5	2.31

Calcule aproximações de dy/dx usando diferenças finitas centrais de ordem h^2 quando possível e, caso contrário, diferenças finitas progressiva ou regressiva conforme o caso.

E.1.1.6. Use uma combinação de polinômios de Taylor de grau 2 para desenvolver a fórmula de diferenças finitas progressiva de ordem h^2

$$D_{+,h^2}(x) := \frac{1}{2h} [-3f(x) + 4f(x+h) - f(x+2h)]. \quad (1.18)$$

Então, aplique-a para computar $f'(\pi/3)$ com $f(x) = \sin(x)$ e verifique o comportamento do erro $|D_{+,h^2}(\pi/3) - f'(\pi/3)|$ em relação à ordem de truncamento da fórmula.

E.1.1.7. Use uma combinação de polinômios de Taylor de grau 2 para desenvolver a fórmula de diferenças finitas regressiva de ordem h^2

$$D_{-,h^2}(x) := \frac{1}{2h} [3f(x) - 4f(x-h) + f(x-2h)]. \quad (1.19)$$

Então, aplique-a para computar $f'(\pi/3)$ com $f(x) = \sin(x)$ e verifique o comportamento do erro $|D_{-,h^2}(\pi/3) - f'(\pi/3)|$ em relação à ordem de truncamento da fórmula.

E.1.1.8. Refaça as computações do Exercício 1.1.5 usando fórmulas de diferenças finitas de ordem h^2 para todos os pontos.

1.2 Derivadas de Segunda Ordem

Diferentemente do usual em técnicas analíticas, no âmbito da matemática numérica é preferível obter aproximações diretas de derivadas de segunda ordem, em vez de utilizar aproximações sucessivas de derivadas. Na sequência, desenvolvemos e aplicaremos uma fórmula de diferenças finitas central para a aproximação de derivadas de segunda ordem.

Consideremos os seguintes polinômios de Taylor¹ de grau 3 de $f(x)$ em torno do ponto x

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(x) + O(h^4), \quad (1.20)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f'''(x) + O(h^4). \quad (1.21)$$

$$(1.22)$$

Somando estas duas equações, obtemos

$$f(x+h) + f(x-h) = 2f(x) + h^2f''(x) + O(h^4). \quad (1.23)$$

Então, isolando $f''(x)$ temos

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2). \quad (1.24)$$

Isto nos leva a definição da **fórmula de diferenças finitas de ordem h^2 para a derivada segunda**

$$D_{0,h^2}^2 f(x) := \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (1.25)$$

Exemplo 1.2.1. Consideramos o problema de computar a derivada segunda de $f(x) = x^2 + \sin x$ no ponto $x = \pi/6$. Analiticamente, $f''(\pi/6) = 2 - \sin(\pi/6) = 1,5$. Numericamente, vamos explorar as seguintes duas aproximações:

- a) Aplicação de sucessivas diferenças finitas centrais de ordem h^2 para derivada primeira, i.e.

$$f''(x) \approx D_{0,h^2} D_{0,h^2} f(x) \quad (1.26a)$$

$$= \frac{D_{0,h^2} f(x+h) - D_{0,h^2} f(x-h)}{2h} \quad (1.26b)$$

- b) Aplicação da fórmula de diferenças finitas central de ordem h^2 para a derivada segunda, i.e.

$$f''(x) \approx D_{0,h^2}^2 f(x) \quad (1.27a)$$

$$= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (1.27b)$$

Na Tabela 1.5 temos os valores computados em ambos os casos e seus respectivos erros absolutos para diversas escolhas de h . Observamos que a aplicação

Tabela 1.5: Resultados referente ao Exemplo 1.2.1. Notação: $\delta_{DD} := |f''(\pi/6) - D_{0,h^2}D_{0,h^2}f(\pi/6)|$ e $\delta_{D^2} := |f''(\pi/6) - D_{0,h^2}^2f(\pi/6)|$.

h	$D_{0,h^2}D_{0,h^2}f(\pi/6)$	δ_{DD}	$D_{0,h^2}^2f(\pi/6)$	δ_{D^2}
10^{-1}	1.50166	1.7e-03	1.50042	4.2e-04
10^{-2}	1.50002	1.7e-05	1.50000	4.2e-06
10^{-3}	1.50000	1.7e-07	1.50000	4.2e-08
10^{-5}	1.50000	1.2e-07	1.50000	1.2e-07

da diferença finita D_{0,h^2}^2 fornece resultados mais precisos (para valores moderados de h) do que as sucessivas aplicações de D_{0,h^2} . De fato, uma rápida inspeção de (1.26) mostra que

$$D_{0,h^2}D_{0,h^2}f(x) = \underbrace{\frac{f(x+2h) - 2f(x) + f(x-2h)}{4h^2}}_{D_{0,(2h)^2}^2f(x)}. \quad (1.28)$$

Código 1.4: d2fc_h2.py

```

1 import numpy as np
2
3 def d2fc_h2(f, x, h=1e-7):
4     df = (f(x+h) - 2*f(x) + f(x-h))/h**2
5     return df
6
7 f = lambda x: x**2 + np.sin(x)
8 x = np.pi/6
9 h = 1e-1
10 d2fdx2 = d2fc_h2(f, x, h)
11 print(f'{h}: d2fdx2 = {d2fdx2:.5e}, erro = {np.fabs(d2fdx2-1.5):.1e}')
```

Exercícios

E.1.2.1. Use a fórmula de diferenças finitas central de ordem h^2 para computar aproximações da segunda derivada de

$$f(x) = \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} + x \quad (1.29)$$

1.3. DIFERENÇAS FINITAS POR POLINÔMIOS INTERPOLADORES

no ponto $x = 2,5$. Para tanto, use os passos

a) $h = 10^{-1}$

b) $h = 10^{-2}$

c) $h = 10^{-3}$

d) $h = 10^{-4}$

Por fim, com base nos resultados obtidos, qual foi o maior passo que forneceu a aproximação com precisão de pelo menos 5 dígitos significativos? Justifique sua resposta.

E.1.2.2. Considere a função $f(x) = e^x \ln(x+1) - x$. Use a fórmula de diferenças finitas central de ordem h^2 para computar a aproximação de $f''(1)$ com 6 dígitos significativos corretos.

E.1.2.3. Considere a seguinte tabela de pontos

i	1	2	3	4	5	6
x_i	2,0	2,1	2,2	2,3	2,4	2,5
y_i	1,86	1,90	2,01	2,16	2,23	2,31

Calcule a aproximação d^2y/dx^2 no ponto $x = 2,2$ usando a fórmula de diferenças finitas central de ordem h^2 .

1.3 Diferenças Finitas por Polinômios Interpoladores

Vamos estudar como obter fórmulas de diferenças finitas por polinômios interpoladores. Seja $p(x)$ o polinômio interpolador dos pontos $\{(x_i, f(x_i))\}_{i=1}^{n+1}$ de uma dada função $f(x)$, com $x_1 < x_2 < \dots < x_{n+1}$. Então, pelo teorema de Lagrange temos

$$f(x) = p(x) + R_{n+1}(x), \quad (1.30)$$

onde $R(x)$ é o erro na aproximação de $f(x)$ por $p(x)$ e tem a forma

$$R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=1}^{n+1} (x - x_j). \quad (1.31)$$

onde $\xi = \xi(x)$.

Deste modo, a ideia para obtermos as fórmulas de diferenças é aproximarmos $f'(x)$ por $p'(x)$. Entretanto, isto nos coloca a questão de estimarmos o erro $|f'(x) - p'(x)|$. Por sorte temos o seguinte teorema.

Teorema 1.3.1. *Seja $p(x)$ o polinômio interpolador de uma dada função $f(x)$ pelo pontos $\{(x_i, f(x_i))\}_{i=1}^{n+1}$, com $x_1 < x_2 < \dots < x_{n+1}$. Se $f(x)$ é $(n+1)$ continuamente diferenciável, então o resíduo $R_{n+1}^{(k)}(x) = f^{(k)}(x) - p^{(k)}(x)$ é*

$$R_{n+1}^{(k)} = \frac{f^{(n+1)}(\eta)}{(n+1-k)!} \prod_{j=1}^{n+1-k} (x - \xi_j), \quad (1.32)$$

onde ξ_j é um ponto tal que $x_j < \xi_j < x_{j+k}$, $j = 1, 2, \dots, n+1+k$, e $\eta = \eta(x)$ é algum ponto no intervalo de extremos x e ξ_j .

Demonstração. Veja [3, Ch.6, Sec.5]. □

1.3.1 Fórmulas de dois pontos

Para obtermos fórmulas de diferenças finitas de dois pontos consideramos $p(x)$ o polinômio interpolador de Lagrange de $f(x)$ pelos pontos $(x_1, f(x_1))$ e $(x_2, f(x_2))$, com $x_1 < x_2$, i.e.

$$f(x) = p(x) + R_2(x) \quad (1.33)$$

$$= f(x_1) \frac{x - x_2}{x_1 - x_2} + f(x_2) \frac{x - x_1}{x_2 - x_1} + R_2(x). \quad (1.34)$$

Denotando $h = x_2 - x_1$, temos

$$f(x) = f(x_1) \frac{x - x_2}{-h} + f(x_2) \frac{x - x_1}{h} + R_2(x). \quad (1.35)$$

1.3. DIFERENÇAS FINITAS POR POLINÔMIOS INTERPOLADORES

e, derivando com respeito a x

$$f'(x) = \frac{f(x_2) - f(x_1)}{h} + R_2^{(1)}(x), \quad (1.36)$$

onde $R_2^{(1)}(x)$ é dado conforme o Teorema 1.3.1.

Agora, escolhendo $x = x_1$, temos $x_2 = x_1 + h = x + h$ e, obtemos a **fórmula de diferenças finitas progressiva de ordem h**

$$f(x) = \underbrace{\frac{f(x+h) - f(x)}{h}}_{D_{+,h}f(x)} + O(h). \quad (1.37)$$

Se escolhermos $x = x_2$, temos $x_1 = x_2 - h = x - h$, obtemos a **fórmula de diferenças finitas regressiva de ordem h**

$$f(x) = \underbrace{\frac{f(x) - f(x-h)}{h}}_{D_{-,h}f(x)} + O(h). \quad (1.38)$$

Fórmulas de três pontos

Para obtermos **fórmulas de diferenças finitas de três pontos consideramos o polinômio interpolador de Lagrange de $f(x)$ pelos pontos $(x_1, f(x_1))$, $(x_2, f(x_2))$ e $(x_3, f(x_3))$, $x_1 < x_2 < x_3$, i.e.**

$$f(x) = f(x_1) \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} \quad (1.39)$$

$$+ f(x_2) \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} \quad (1.40)$$

$$+ f(x_3) \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} + R_3(x). \quad (1.41)$$

Derivando em relação a x , obtemos

$$f'(x) = f(x_1) \frac{(x_2 - x_3)(2x - x_2 - x_3)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} \quad (1.42)$$

$$+ f(x_2) \frac{(x_1 - x_3)(-2x + x_1 + x_3)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} \quad (1.43)$$

$$+ f(x_3) \frac{(x_1 - x_2)(2x - x_1 - x_2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} + R_3^{(1)}(x). \quad (1.44)$$

Aqui, podemos escolher por obter fórmulas de diferenças com passo constante ou não. Por exemplo, denotando $h_1 = x_2 - x_1$ e $h_2 = x_3 - x_2$ e escolhendo $x = x_1$, temos $x_2 = x + h_1$ e $x_3 = x + h_1 + h_2$. Fazendo estas substituições na expressão acima, obtemos seguinte fórmula de diferenças finitas progressiva

$$D_{+,h_1,h_2}f(x) = \frac{1}{h_1h_2(h_1+h_2)} (-h_2(2h_1+h_2)f(x) \quad (1.45)$$

$$+ (h_1+h_2)^2 f(x+h_1) \quad (1.46)$$

$$- h_1^2 f(x+h_1+h_2)). \quad (1.47)$$

Agora, assumindo um passo constante $h = h_1 = h_2$, obtemos a **fórmula de diferenças progressiva de ordem h^2**

$$D_{+,h^2}f(x) = \frac{1}{2h} [-3f(x) + 4f(x+h) - f(x+2h)]. \quad (1.48)$$

Escolhendo $x = x_2$, $x_1 = x - h$ e $x_3 = x + h$ na equação (1.42), obtemos a **fórmula de diferenças finitas central de ordem h^2**

$$D_{0,h^2} = \frac{1}{2h} [f(x+h) - f(x-h)]. \quad (1.49)$$

Por fim, escolhendo $x = x_3$, $x_1 = x - 2h$ e $x_2 = x - h$ na equação (1.42), obtemos a **fórmula de diferenças finitas regressiva de ordem h^2**

$$D_{-,h^2} = \frac{1}{2h} [3f(x) - 4f(x-h) + f(x-2h)]. \quad (1.50)$$

1.3.2 Fórmulas de cinco pontos

Aqui, usamos o polinômio interpolador de Lagrange da função $f(x)$ pelos pontos $(x_1, f(x_1))$, $(x_2, f(x_2))$, $(x_3, f(x_3))$ e $(x_5, f(x_5))$, com $x_1 < x_2 < x_3 <$

$x_4 < x_5$. Isto nos fornece

$$f(x) = \sum_{i=1}^5 f(x_i) \left(\prod_{j=1, j \neq i}^5 \frac{x - x_j}{x_i - x_j} \right) + R_5(x). \quad (1.51)$$

Calculando a derivada em relação a x , temos

$$f'(x) = \sum_{i=1}^5 f(x_i) \left(\sum_{\substack{j=1 \\ j \neq i}}^5 \prod_{\substack{k=1 \\ k \neq i, k \neq j}}^5 \frac{x - x_k}{x_i - x_k} \right) + R_5^{(1)}(x). \quad (1.52)$$

Por exemplo, substituindo $x_1 = x - 2h$, $x_2 = x - h$, $x_3 = x$, $x_4 = x + h$ e $x_5 = x + 2h$ na equação acima, obtemos fórmula de diferenças finitas central de ordem h^4

$$D_{+,h^4} f(x) := \frac{1}{12h} [f(x - 2h) - 8f(x - h) + 8f(x + h) - f(x + 2h)]. \quad (1.53)$$

Exercícios

E.1.3.1. Use a fórmula de diferenças finitas central de ordem h^4 para computar a aproximação da derivada de

$$f(x) = \frac{\sin(x + 2) - e^{-x^2}}{x^2 + \ln(x + 2)} + x \quad (1.54)$$

no ponto $x = 2,5$ com passo $h = 0,1$.

E.1.3.2. Obtenha as seguintes fórmulas de diferenças finitas de 5 pontos com passo h constante e com:

- a) 4 pontos para frente.
- b) 1 ponto para trás e 3 pontos para frente.
- c) 2 pontos para trás e 2 pontos para frente.

d) 3 pontos para traz e 1 pontos para frente.

e) 4 pontos para traz.

E.1.3.3. Considere a seguinte tabela de pontos

i	1	2	3	4	5	6
x_i	2,0	2,1	2,2	2,3	2,4	2,5
y_i	1,86	1,90	2,01	2,16	2,23	2,31

Calcule a aproximação dy/dx nos pontos tabelados usando as fórmulas de diferenças finitas obtidas no exercício anteriores (Exercício 1.3.2). Para tanto, dê preferência para fórmulas centrais sempre que possível.

Capítulo 2

Técnicas de extrapolação

[[tag:revisar]]

Neste capítulo, estudamos algumas técnicas de extrapolação, as quais serão usadas nos próximos capítulos.

2.1 Extrapolação de Richardson

[[tag:revisar]]

Seja $F_1(h)$ uma aproximação de I tal que

$$I = F_1(h) + \underbrace{k_1h + k_2h^2 + k_3h^3 + O(h^4)}_{\text{erro de truncamento}}. \quad (2.1)$$

Então, dividindo h por 2, obtemos

$$I = F_1\left(\frac{h}{2}\right) + k_1\frac{h}{2} + k_2\frac{h^2}{4} + k_3\frac{h^3}{8} + O(h^4). \quad (2.2)$$

Agora, de forma a eliminarmos o termo de ordem h das expressões acima, subtraímos (2.1) de 2 vezes (2.2), o que nos leva a

$$I = \underbrace{\left[F_1\left(\frac{h}{2}\right) + \left(F_1\left(\frac{h}{2}\right) - F_1(h) \right) \right]}_{F_2(h)} - k_2\frac{h^2}{2} - k_3\frac{3h^3}{4} + O(h^4). \quad (2.3)$$

Ou seja, denotando

$$F_2(h) := F_1\left(\frac{h}{2}\right) + \left(F_1\left(\frac{h}{2}\right) - F_1(h)\right) \quad (2.4)$$

temos que $N_2(h)$ é uma aproximação de I com erro de truncamento da ordem de h^2 , uma ordem a mais de $N_1(h)$. Ou seja, esta combinação de aproximações de ordem de truncamento h nos fornece uma aproximação de ordem de truncamento h^2 .

Analogamente, consideremos a aproximação de I por $N_2(h/2)$, *i.e.*

$$I = F_2\left(\frac{h}{2}\right) - k_2 \frac{h^2}{8} - k_2 \frac{3h^3}{32} + O(h^4) \quad (2.5)$$

Então, subtraindo (2.3) de 4 vezes (2.5) de, obtemos

$$I = \underbrace{\left[3F_2\left(\frac{h}{2}\right) + \left(F_2\left(\frac{h}{2}\right) - F_2(h)\right)\right]}_{F_3(h)} + k_3 \frac{3h^3}{8} + O(h^4). \quad (2.6)$$

Observemos, ainda, que $N_3(h)$ pode ser reescrita na forma

$$F_3(h) = F_2\left(\frac{h}{2}\right) + \frac{F_2\left(\frac{h}{2}\right) - F_2(h)}{3}, \quad (2.7)$$

a qual é uma aproximação de ordem h^3 para I .

Para fazermos mais um passo, consideramos a aproximação de I por $F_3(h/2)$, *i.e.*

$$I = F_3\left(\frac{h}{2}\right) + k_3 \frac{3h^3}{64} + O(h^4). \quad (2.8)$$

E, então, subtraindo (2.6) de 8 vezes (2.8), temos

$$I = \underbrace{\left[F_3\left(\frac{h}{2}\right) + \left(\frac{F_3\left(\frac{h}{2}\right) - F_3(h)}{7}\right)\right]}_{F_4(h)} + O(h^4). \quad (2.9)$$

Ou seja,

$$F_4(h) = \left[F_3\left(\frac{h}{2}\right) + \frac{F_3\left(\frac{h}{2}\right) - F_3(h)}{7} \right] \quad (2.10)$$

é uma aproximação de I com erro de truncamento da ordem h^4 . Estes cálculos nos motivam o seguinte teorema.

Teorema 2.1.1. *Seja $F_1(h)$ uma aproximação de I com erro de truncamento da forma*

$$I - F_1(h) = \sum_{i=1}^n k_i h^i + O(h^{n+1}). \quad (2.11)$$

Então, para $j \geq 2$,

$$F_j(h) := F_{j-1}\left(\frac{h}{2}\right) + \frac{F_{j-1}\left(\frac{h}{2}\right) - F_{j-1}(h)}{2^{j-1} - 1} \quad (2.12)$$

é uma aproximação de I com erro de truncamento da forma

$$\begin{aligned} I - F_j(h) &= \sum_{i=j}^n (-1)^{j-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{j-2} (2^{i-l-1} - 1)}{2^{(j-1)(i-j+1)} d_j} k_i h^i \\ &\quad + O(h^{n+1}), \end{aligned} \quad (2.13)$$

onde d_j é dado recursivamente por $d_{j+1} = 2^{j-1} d_j$, com $d_2 = 1$.

Demonstração. Fazemos a demonstração por indução. O resultado para $j = 2$ segue de (2.3). Assumimos, agora, que vale

$$\begin{aligned} I - F_j(h) &= (-1)^{j-1} \frac{(2^{j-1} - 1) \prod_{l=1}^{j-2} (2^{j-l-1} - 1)}{2^{(j-1)} d_j} k_j h^j \\ &\quad + \sum_{i=j+1}^n (-1)^{j-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{j-2} (2^{i-l-1} - 1)}{2^{(j-1)(i-j+1)} d_j} k_i h^i \\ &\quad + O(h^{n+1}). \end{aligned} \quad (2.14)$$

para $j \geq 2$. Então, tomamos

$$\begin{aligned} I - F_j \left(\frac{h}{2} \right) &= (-1)^{j-1} \frac{(2^{j-1} - 1) \prod_{l=1}^{j-2} (2^{j-l-1} - 1)}{2^{(j-1)} d_j} k_j \frac{h^j}{2^j} \\ &\quad + \sum_{i=j+1}^n (-1)^{j-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{j-2} (2^{i-l-1} - 1)}{2^{(j-1)(i-j+1)} d_j} k_i \frac{h^i}{2^i} \\ &\quad + O(h^{n+1}). \end{aligned} \quad (2.15)$$

Agora, subtraímos (2.14) de 2^j vezes (2.15), o que nos fornece

$$\begin{aligned} I &= \left[F_j \left(\frac{h}{2} \right) + \frac{F_j \left(\frac{h}{2} \right) - F_j(h)}{2^j - 1} \right] \\ &\quad + \sum_{i=j+1}^n (-1)^{(j+1)-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{(j+1)-2} (2^{i-l-1} - 1)}{2^{((j+1)-1)(i-(j+1)+1)} 2^{j-1} d_j} k_i h^i \\ &\quad + O(h^{n+1}). \end{aligned} \quad (2.16)$$

□

Corolário 2.1.1. *Seja $F_1(h)$ uma aproximação de I com erro de truncamento da forma*

$$I - F_1(h) = \sum_{i=1}^n k_i h^{2i} + O(h^{2n+2}). \quad (2.17)$$

Então, para $j \geq 2$,

$$F_j(h) := F_{j-1} \left(\frac{h}{2} \right) + \frac{F_{j-1} \left(\frac{h}{2} \right) - F_{j-1}(h)}{4^{j-1} - 1} \quad (2.18)$$

é uma aproximação de I com erro de truncamento da forma

$$\begin{aligned} I - F_j(h) &= \sum_{i=j}^n (-1)^{j-1} \frac{(4^{i-1} - 1) \prod_{l=1}^{j-2} (4^{i-l-1} - 1)}{4^{(j-1)(i-j+1)} d_j} k_i h^{2i} \\ &\quad + O(h^{n+1}), \end{aligned} \quad (2.19)$$

onde d_j é dado recursivamente por $d_{j+1} = 4^{j-1} d_j$, com $d_2 = 1$.

Demonstração. A demonstração é análoga ao do Teorema 2.1.1. \square

Exemplo 2.1.1. Dada uma função $f(x)$, consideremos sua aproximação por diferenças finitas progressiva de ordem h , i.e.

$$\underbrace{f'(x)}_I = \underbrace{\frac{f(x+h) - f(x)}{h}}_{F_1(h)} + \frac{f''(x)}{2}h + \frac{f'''(x)}{6}h^2 + O(h^3). \quad (2.20)$$

Estão, considerando a primeira extrapolação de Richardson, temos

$$F_2(h) = F_1\left(\frac{h}{2}\right) + \left(F_1\left(\frac{h}{2}\right) - F_1(h)\right) \quad (2.21)$$

$$= 4 \frac{f(x+h/2) - f(x)}{h} - \frac{f(x+h) - f(x)}{h} \quad (2.22)$$

$$= \frac{-f(x+h) + 4f(x+h/2) - 3f(x)}{h}, \quad (2.23)$$

a qual é a fórmula de diferenças finitas progressiva de três pontos com passo $h/2$, i.e. $D_{+, (h/2)^2} f(x)$ (veja, Fórmula (1.48)).

Exemplo 2.1.2. Dada uma função $f(x)$, consideremos sua aproximação por diferenças finitas central de ordem h^2 , i.e.

$$\underbrace{f'(x)}_I = \underbrace{\frac{f(x+h) - f(x-h)}{2h}}_{F_1(h)} - \frac{f'''(x)}{6}h^2 - \frac{f^{(5)}(x)}{120}h^4 + O(h^6). \quad (2.24)$$

Estão, considerando a primeira extrapolação de Richardson, temos

$$F_2(h) = F_1\left(\frac{h}{2}\right) + \frac{\left(F_1\left(\frac{h}{2}\right) - F_1(h)\right)}{3} \quad (2.25)$$

$$= \frac{1}{6h} [f(x-h) - 8f(x-h/2) + 8f(x+h/2) - f(x+h)] \quad (2.26)$$

a qual é a fórmula de diferenças finitas central de cinco pontos com passo $h/2$, i.e. $D_{+, (h/2)^4} f(x)$ (veja, Fórmula (1.53)).

2.1.1 Sucessivas extrapolações

[[tag:revisar]]

Sucessivas extrapolações de Richardson podem ser computadas de forma robusta com o auxílio de uma tabela. Seja $F_1(h)$ uma dada aproximação de uma quantidade de interesse I com erro de truncamento da forma

$$I - F_1(h) = k_1 h + k_2 h^2 + k_3 h^3 + \dots + k_n h^n + O(h^{n+1}). \quad (2.27)$$

Então, as sucessivas extrapolações $F_2(h)$, $F_3(h)$, \dots , $F_n(h)$ podem ser organizadas na seguinte forma tabular

$$T = \begin{bmatrix} F_1(h) & & & & \\ F_1(h/2) & F_2(h) & & & \\ F_1(h/2^2) & F_2(h/2) & F_3(h) & & \\ \vdots & \vdots & \vdots & & \\ F_1(h/2^n) & F_2(h/2^{n-1}) & F_3(h/2^{n-2}) & \dots & F_n(h) \end{bmatrix} \quad (2.28)$$

Desta forma, temos que

$$F_j\left(\frac{h}{2^{i-1}}\right) = t_{i,j-1} + \frac{t_{i,j-1} - t_{i-1,j-1}}{2^{j-1} - 1} \quad (2.29)$$

com $j = 2, 3, \dots, n$ e $j \geq i$, onde $t_{i,j}$ é o elemento da i -ésima linha e j -ésima coluna da matriz T .

Exemplo 2.1.3. Consideremos o problema de aproximar a derivada da função $f(x) = \sin(x)$ no ponto $\pi/3$. Usando a fórmula de diferenças finitas progressiva de ordem h obtemos

$$\begin{aligned} f'\left(\frac{\pi}{3}\right) &= \underbrace{\frac{f\left(\frac{\pi}{3} + h\right) - f\left(\frac{\pi}{3}\right)}{h}}_{F_1(h) := D_{+,h}f(\pi/3)} \\ &+ \frac{f''(x)}{2}h + \frac{f'''(x)}{6}h^2 + \dots \end{aligned} \quad (2.30)$$

Na Tabela 2.1 temos os valores das aproximações de $f'(\pi/3)$ computadas via sucessivas extrapolações de Richardson a partir de (2.30) com $h = 0.1$.

Tabela 2.1: Resultados referente ao Exemplo 2.1.3.

$O(h)$	$O(h^2)$	$O(h^3)$	$O(h^4)$
4,55902e-1			
4,78146e-1	5,00389e-1		
4,89123e-1	5,00101e-1	5,00005e-1	
4,94574e-1	5,00026e-1	5,00001e-1	5,00000e-1

Exemplo 2.1.4. Novamente, consideremos o problema de aproximar a derivada da função $f(x) = \sin(x)$ no ponto $\pi/3$. A fórmula de diferenças finitas central de ordem h^2 tem a forma

$$f'\left(\frac{\pi}{3}\right) = \underbrace{\frac{f\left(\frac{\pi}{3} + h\right) - f\left(\frac{\pi}{3} - h\right)}{2h}}_{F_1(h) := D_{0,h^2}f(\pi/3)} - \frac{f'''(x)}{6}h^2 + \frac{f^{(5)}(x)}{120}h^4 - \dots \quad (2.31)$$

Na Tabela 2.2 temos os valores das aproximações de $f'(\pi/3)$ computadas via sucessivas extrapolações de Richardson a partir de (2.31) com $h = 1$.

Tabela 2.2: Resultados referente ao Exemplo 2.1.4.

$O(h^2)$	$O(h^4)$	$O(h^6)$	$O(h^8)$
4,20735e-1			
4,79426e-1	4,98989e-1		
4,94808e-1	4,99935e-1	4,99998e-1	
4,98699e-1	4,99996e-1	5,00000e-1	5,00000e-1

2.1.2 Exercícios

[[tag:revisar]]

E.2.1.1. Mostre que a primeira extrapolação de Richardson de

$$D_{-,h}f(x) = \frac{f(x) - f(x-h)}{h} \quad (2.32)$$

é igual a

$$D_{-(h/2)^2}f(x) = \frac{3f(x) - 4f(x-h) + f(x-2h)}{h}. \quad (2.33)$$

E.2.1.2. Considere o problema de aproximar a derivada de

$$f(x) = \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} + x \quad (2.34)$$

no ponto $x = 2, 5$. Para tanto, use de sucessivas extrapolações de Richardson a partir da aproximação por diferenças finitas:

- a) progressiva de ordem h , com $h = 0, 5$.
- b) regressiva de ordem h , com $h = 0, 5$.
- c) central de ordem h^2 , com $h = 0, 5$.

Nas letras a) e b), obtenha as aproximações de ordem h^3 e, na letra c) obtenha a aproximação de ordem h^6 .

Capítulo 3

Integração

Neste capítulo, estudamos os métodos fundamentais para a **aproximação numérica de integrais definidas de funções** de uma variável real. São chamados de **quadraturas numéricas** e têm a forma

$$\int_a^b f(x) dx \approx \sum_{i=1}^n f(x_i)w_i, \quad (3.1)$$

onde x_i e w_i são, respectivamente, o i -ésimo **nodo** e o i -ésimo **peso da quadratura**, $i = 1, 2, \dots, n$.

3.1 Regras de Newton-Cotes

Buscamos um método para a aproximação numérica da integral de uma dada função $f(x)$ em um dado intervalo $[a, b]$, i.e.

$$I := \int_a^b f(x) dx. \quad (3.2)$$

A ideia das **Regras de Newton-Cotes** é aproximar I pela integral de um polinômio interpolador de $f(x)$ por pontos previamente selecionados.

Seja, $p(x)$ o **polinômio interpolador** de grau n de $f(x)$ pelos dados pontos $\{(x_i, f(x_i))\}_{i=1}^{n+1}$, com $x_1 < x_2 < \dots < x_{n+1}$ e $x_i \in [a, b]$ para todo $i =$

$1, 2, \dots, n+1$. Então, pelo Teorema de Lagrange, temos

$$f(x) = p(x) + R_{n+1}(x), \quad (3.3)$$

onde

$$p(x) = \sum_{i=1}^{n+1} f(x_i) \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{x - x_j}{x_i - x_j} \quad (3.4)$$

e

$$R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=1}^{n+1} (x - x_j), \quad (3.5)$$

para algum $\xi = \xi(x)$ pertencente ao intervalo $[x_1, x_{n+1}]$. Deste modo, temos

$$I := \int_a^b f(x) \quad (3.6a)$$

$$= \int_a^b p(x) dx + \int_a^b R_{n+1}(x) dx \quad (3.6b)$$

$$= \underbrace{\sum_{i=1}^{n+1} f(x_i) \int_a^b \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{(x - x_j)}{x_i - x_j} dx}_{\text{quadratura}} + \underbrace{\int_a^b R_{n+1}(x) dx}_{\text{erro de truncamento}} \quad (3.6c)$$

Ou seja, nas quadraturas (regras) de Newton-Cotes, os nodos são as abscissas dos pontos interpolados e os pesos são as integrais dos polinômios de Lagrange associados.

Na sequência, desenvolvemos as Regras de Newton-Cotes mais usuais e estimamos o erro de truncamento em cada caso¹.

3.1.1 Regras de Newton-Cotes Fechadas

As **Regras de Newton-Cotes Fechadas** são aquelas em que a quadratura inclui os extremos do intervalo de integração.

¹Consulte [3, Cap. 7, Sec. 1.1], para uma abordagem mais geral.

Regra do Trapézio

A Regra do Trapézio é obtida tomando-se os nodos $x_1 = a$ e $x_2 = b$. Então, denotando $h := b - a$ ², os pesos da quadratura são:

$$w_1 = \int_a^b \frac{x - b}{a - b} dx \quad (3.7a)$$

$$= \frac{(b - a)}{2} = \frac{h}{2}. \quad (3.7b)$$

e

$$w_2 = \int_a^b \frac{x - a}{b - a} dx \quad (3.8a)$$

$$= \frac{(b - a)}{2} = \frac{h}{2}. \quad (3.8b)$$

Agora, estimamos o erro de truncamento com

$$R := \int_a^b R_2(x) dx \quad (3.9)$$

$$= \int_a^b \frac{f''(\xi(x))}{2} (x - a)(x - b) dx \quad (3.10)$$

$$\leq C \left| \int_a^b (x - a)(x - b) dx \right| \quad (3.11)$$

$$= C \frac{(b - a)^3}{6} = O(h^3). \quad (3.12)$$

Portanto, a Regra do Trapézio é

$$\int_a^b f(x) dx = \frac{h}{2} [f(a) + f(b)] + O(h^3). \quad (3.13)$$

Exemplo 3.1.1. Consideramos o problema de computar a integral de $f(x) = xe^{-x^2}$ no intervalo $[0, 1/4]$. Analiticamente, temos

$$I = \int_0^{1/4} xe^{-x^2} dx \quad (3.14a)$$

²Neste capítulo, h é escolhido como a distância entre os nodos.

$$= -\frac{e^{-x^2}}{2} \Big|_0^{1/4} \quad (3.14b)$$

$$= \frac{1 - e^{-1/4}}{2} = 3,02935e-2. \quad (3.14c)$$

Agora, usando a Regra do Trapézio, obtemos a seguinte aproximação

$$I \approx \frac{h}{2} [f(0) + f(1/4)] \quad (3.15a)$$

$$= \frac{1/4}{2} \left(0 + \frac{1}{4} e^{-(1/4)^2} \right) \quad (3.15b)$$

$$= 2,93567e-2. \quad (3.15c)$$

```

1 import numpy as np
2
3 # intervalo
4 a = 0.
5 b = 1./4
6 # fun
7 f = lambda x: x*np.exp(-x**2)
8 # quad
9 h = b-a
10 I = h/2*(f(a) + f(b))
11 print(f'I = {I:.5e}')
```

Regra de Simpson

A Regra de Simpson⁶ é obtida escolhendo-se os nodos $x_1 = a$, $x_2 = (a + b)/2$ e $x_3 = b$. Denotando $h := (b - a)/2$, calculamos os pesos

$$w_1 = \int_a^b \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} dx \quad (3.16a)$$

$$= \frac{(b - a)}{6} = \frac{h}{3}, \quad (3.16b)$$

$$w_2 = \int_a^b \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} dx \quad (3.17a)$$

$$= 4 \frac{(b-a)}{6} = 4 \frac{h}{3} \quad (3.17b)$$

e

$$w_3 = \int_a^b \frac{(x-x_1)(x-x_2)}{(x_3-x_1)(x_3-x_2)} dx \quad (3.18)$$

$$= \frac{(b-a)}{6} = \frac{h}{3}. \quad (3.19)$$

Isto nos fornece a quadratura

$$I \approx \frac{h}{3} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (3.20)$$

Para estimar o **erro de truncamento**, consideramos a expansão em polinômio de Taylor⁷ de grau 3 de $f(x)$ em torno do ponto x_2 , i.e.

$$\begin{aligned} f(x) &= f(x_2) + f'(x_2)(x-x_2) \\ &\quad + \frac{f''(x_2)}{2}(x-x_2)^2 \\ &\quad + \frac{f'''(x_2)}{6}(x-x_2)^3 \\ &\quad + \frac{f^{(4)}(\xi_1(x))}{24}(x-x_2)^4, \end{aligned} \quad (3.21)$$

donde

$$\begin{aligned} \int_a^b f(x) dx &= 2hf(x_2) + \frac{h^3}{3}f''(x_2) \\ &\quad + \frac{1}{24} \int_a^b f^{(4)}(\xi_1(x))(x-x_2)^4 dx. \end{aligned} \quad (3.22)$$

Daí, usando da fórmula de diferenças finitas central de ordem h^2 , temos

$$f''(x_2) = \frac{f(x_1) - 2f(x_2) + f(x_3)}{h^2} + O(h^2). \quad (3.23)$$

O último termo de (3.22) pode ser estimado por

$$\left| \frac{1}{24} \int_a^b f^{(4)}(\xi_1(x))(x-x_2)^4 dx \right| \leq C \left| \int_a^b (x-x_2)^4 dx \right| \quad (3.24a)$$

$$= C(b-a)^5 = O(h^5). \quad (3.24b)$$

Então, de (3.22), (3.23) e (3.24), temos a **Regra de Simpson**

$$\int_a^b f(x) dx = \frac{h}{3} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] + O(h^5). \quad (3.25)$$

Exemplo 3.1.2. A aproximação da integral do Exemplo 3.1.1 pela a Regra de Simpson é

$$\int_0^{1/4} f(x) dx \approx \frac{1/8}{3} \left[f(0) + 4f\left(\frac{1}{8}\right) + f\left(\frac{1}{4}\right) \right] \quad (3.26a)$$

$$= \frac{1}{24} \left[\frac{1}{2} e^{-(1/8)^2} + \frac{1}{4} e^{-(1/4)^2} \right] \quad (3.26b)$$

$$= 3,02959e-2. \quad (3.26c)$$

```

1 import numpy as np
2
3 # intervalo
4 a = 0.
5 b = 1./4
6 # fun
7 f = lambda x: x*np.exp(-x**2)
8 # quad
9 h = (b-a)/2
10 I = h/3*(f(a) + 4*f((a+b)/2) + f(b))
11 print(f'I = {I:.5e}')
```

3.1.2 Regras de Newton-Cotes Abertas

As Regras de Newton-Cotes Abertas não incluem os extremos dos intervalos como nodos das quadraturas.

Regra do Ponto Médio

A Regra do Ponto Médio é obtida usando apenas o nodo $x_1 = (a+b)/2$.

Desta forma, temos

$$\begin{aligned} \int_a^b f(x) dx &= \int_a^b f(x_1) dx \\ &+ \int_a^b f'(\xi(x))(x - x_1) dx, \end{aligned} \quad (3.27)$$

donde, denotando $h := (b - a)$, temos³

$$\int_a^b f(x) dx = hf\left(\frac{a+b}{2}\right) + O(h^3). \quad (3.28)$$

Exemplo 3.1.3. Aproximando a integral dada no Exemplo 3.1.1 pela a Regra do Ponto Médio, obtemos

$$\int_0^{1/4} f(x) dx \approx \frac{1}{4} f\left(\frac{1}{8}\right) \quad (3.29a)$$

$$= \frac{1}{32} e^{-(1/8)^2} \quad (3.29b)$$

$$= 3,07655e-2 \quad (3.29c)$$

```
1 import numpy as np
2
3 # intervalo
4 a = 0.
5 b = 1./4
6 # fun
7 f = lambda x: x*np.exp(-x**2)
8 # quad
9 h = b-a
10 I = h*f((a+b)/2)
11 print(f'I = {I:.5e}')
```

3.1.3 Exercício

E.3.1.1. Aproxime

$$I = \int_{\pi/6}^{\pi/4} e^{-x} \cos(x) dx \quad (3.30)$$

³Para a estimativa do erro de truncamento, consulte o Exercício 3.1.5.

pelas seguintes Regras de Newton-Cotes e compute o erro absoluto em relação ao valor exato:

- a) Regra do Trapézio.
- b) Regra de Simpson.
- c) Regra do Ponto Médio.

E.3.1.2. Aproxime

$$\int_{-1}^0 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (3.31)$$

usando a:

- a) Regra do Ponto Médio.
- b) Regra do Trapézio.
- c) Regra de Simpson.

E.3.1.3. Considere a seguinte tabela de pontos

i	x_i	y_i
1	2.0	1.86
2	2.1	1.90
3	2.2	2.01
4	2.3	2.16
5	2.4	2.23
6	2.5	2.31

Assumindo que $y = f(x)$, compute:

- a) $\int_{2,1}^{2,3} f(x) dx$ usando a Regra do Ponto Médio.
- b) $\int_{2,0}^{2,5} f(x) dx$ usando a Regra do Trapézio.

c) $\int_{2,0}^{2,4} f(x) dx$ usando a Regra de Simpson.

E.3.1.4. Considere uma função $y = f(x)$ com valores tabelados como no Exercício 3.1.3. Observando que

$$\underbrace{\int_{2,0}^{2,4} f(x) dx}_{:=I} = \underbrace{\int_{2,0}^{2,2} f(x) dx}_{:=I_1} + \underbrace{\int_{2,2}^{2,4} f(x) dx}_{:=I_2} \quad (3.32)$$

compute, com a Regra de Simpson, as seguintes aproximações:

- a) $\tilde{I} \approx I$.
- b) $\tilde{I}_1 \approx I_1$.
- c) $\tilde{I}_2 \approx I_2$.
- d) $\tilde{\tilde{I}} = \tilde{I}_1 + \tilde{I}_2$.

Por fim, diga qual das aproximações \tilde{I} e $\tilde{\tilde{I}}$ de I tem maior exatidão. Justifique sua proposta.

E.3.1.5. Mostre que o erro de truncamento da regra do ponto médio é da ordem de h^3 , onde h é o tamanho do intervalo de integração.

E.3.1.6. Desenvolva a Regra de Newton-Cotes Aberta de 2 pontos e estime seu erro de truncamento.

3.2 Regras Compostas de Newton-Cotes

Regras de integração numérica compostas (ou **quadraturas compostas**) são aquelas obtidas da composição de quadraturas aplicadas as subintervalos do intervalo de integração. Mais especificamente, a integral de uma dada

função $f(x)$ em um dado intervalo $[a, b]$ pode ser reescrita como uma soma de integrais em sucessivos subintervalos de $[a, b]$, i.e.

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{x_i}^{x_{i+1}} f(x) dx, \quad (3.33)$$

onde $a = x_1 < x_2 < \dots < x_{n+1} = b$. Então, a aplicação de uma quadratura em cada integral em $[x_i, x_{i+1}]$, $i = 1, 2, \dots, n$, nos fornece uma regra composta.

3.2.1 Regra Composta do Ponto Médio

Consideramos uma partição uniforme do intervalo de integração $[a, b]$ da forma $a = \tilde{x}_1 < \tilde{x}_2 < \dots < \tilde{x}_{n+1} = b$, com $h = \tilde{x}_{i+1} - \tilde{x}_i$, $i = 1, 2, \dots, n$. Então, aplicando a regra do ponto médio a cada integral nos subintervalos $[\tilde{x}_i, \tilde{x}_{i+1}]$, temos

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} f(x) dx \quad (3.34)$$

$$= \sum_{i=1}^n \left[hf \left(\frac{\tilde{x}_i + \tilde{x}_{i+1}}{2} \right) + O(h^3) \right]. \quad (3.35)$$

Agora, observando que $h := (b - a)/n$ e escolhendo os nodos

$$x_i = a + (i - 1/2)h, \quad (3.36)$$

$i = 1, 2, \dots, n$, obtemos a **regra composta do ponto médio com n subintervalos**

$$\int_a^b f(x) dx = \sum_{i=1}^n hf(x_i) + O(h^2). \quad (3.37)$$

Exemplo 3.2.1. Consideramos o problema de computar a integral de $f(x) = xe^{-x^2}$ no intervalo $[0, 1]$. Usando a regra composta do ponto médio com n subintervalos, obtemos a aproximação

$$\underbrace{\int_0^1 xe^{-x^2} dx}_I \approx \underbrace{\sum_{i=1}^n hf(x_i)}_S, \quad (3.38)$$

onde $h = 1/n$ e $x_i = (i - 1/2)h$, $i = 1, 2, \dots, n$. Na Tabela 3.1, temos as aproximações computadas com diversos números de subintervalos, bem como, seus erros absolutos.

Tabela 3.1: Resultados referentes ao Exemplo 3.2.1.

n	S	$ S - I $
1	3.89400e-1	7.3e-2
10	3.16631e-1	5.7e-4
100	3.16066e-1	5.7e-6
1000	3.16060e-1	5.7e-8

Código 3.1: pm_comp.py

```

1 import numpy as np
2
3 def pm_comp(f, a, b, n):
4     h = (b-a)/n
5     S = 0.
6     for i in range(n):
7         x = a + (i+0.5)*h
8         S += f(x)
9     S *= h
10    return S
11
12 # intervalo
13 a = 0.
14 b = 1.
15 # integrando
16 def f(x):
17     return x*np.exp(-x**2)
18
19 # quad
20 n = 10
21 S = pm_comp(f, a, b, n)
22 # exata
23 I = 1./2 - np.exp(-1.)/2

```

```

24 # erro abs
25 print(f'{n}: {S:.5e}, {np.fabs(S-I):.1e}')
```

3.2.2 Regra Composta do Trapézio

Para obtermos a **regra composta do trapézio**, consideramos uma partição uniforme do intervalo de integração $[a, b]$ da forma $a = x_1 < x_2 < \dots < x_{n+1} = b$ com $h = x_{i+1} - x_i$, $i = 1, 2, \dots, n$. Então, aplicando a regra do trapézio em cada integração nos subintervalos, obtemos

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{x_i}^{x_{i+1}} f(x) dx \quad (3.39a)$$

$$= \sum_{i=1}^n \left\{ \frac{h}{2} [f(x_i) + f(x_{i+1})] + O(h^3) \right\} \quad (3.39b)$$

$$= \frac{h}{2} f(x_1) + \sum_{i=2}^n h f(x_i) + \frac{h}{2} f(x_{n+1}) + O(h^2). \quad (3.39c)$$

Desta forma, a **regra composta do trapézio com n subintervalos** é

$$\int_a^b f(x) dx = \frac{h}{2} \left[f(x_1) + 2 \sum_{i=2}^n f(x_i) + f(x_{n+1}) \right] + O(h^2), \quad (3.40)$$

onde $h = (b - a)/n$ e $x_i = a + (i - 1)h$, $i = 1, 2, \dots, n$.

Exemplo 3.2.2. Consideramos o problema de computar a integral de $f(x) = xe^{-x^2}$ no intervalo $[0, 1]$. Usando a regra composta do trapézio com n subintervalos, obtemos a aproximação

$$\underbrace{\int_0^1 xe^{-x^2} dx}_I \approx \frac{h}{2} \underbrace{\left[f(x_1) + 2 \sum_{i=2}^n f(x_i) + f(x_{n+1}) \right]}_S, \quad (3.41)$$

onde $h = 1/n$ e $x_i = (i - 1)h$, $i = 1, 2, \dots, n$. Na Tabela 3.2, temos as aproximações computadas com diversos números de subintervalos, bem como, seus erros absolutos.

Tabela 3.2: Resultados referentes ao Exemplo 3.2.2.

n	S	$ S - I $
1	1.83940e-1	1.3e-1
10	3.14919e-1	1.1e-3
100	3.16049e-1	1.1e-5
1000	3.16060e-1	1.1e-7

Código 3.2: trap_comp.py

```

1 import numpy as np
2
3 def trap_comp(f, a, b, n):
4     h = (b-a)/n
5     S = f(a)
6     for i in range(1,n):
7         x = a + i*h
8         S += 2*f(x)
9     S += f(b)
10    S *= h/2
11    return S

```

3.2.3 Regra Composta de Simpson

A fim de obtermos a **regra composta de Simpson**, consideramos uma partição uniforme do intervalo de integração $[a, b]$ da forma $a = \tilde{x}_1 < \tilde{x}_2 < \dots < \tilde{x}_{n+1} = b$, com $h = (\tilde{x}_{i+1} - \tilde{x}_i)/2$, $i = 1, 2, \dots, n$. Então, aplicando a regra de Simpson a cada integral nos subintervalos $[\tilde{x}_i, \tilde{x}_{i+1}]$, temos

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} f(x) dx \quad (3.42a)$$

$$= \sum_{i=1}^n \left\{ \frac{h}{3} \left[f(\tilde{x}_i) + 4f\left(\frac{\tilde{x}_i + \tilde{x}_{i+1}}{2}\right) + f(\tilde{x}_{i+1}) \right] + O(h^5) \right\}. \quad (3.42b)$$

Então, observando que $h = (b - a)/(2n)$ e tomando $x_i = a + (i - 1)h$, $i = 1, 2, \dots, n$, obtemos a **regra composta de Simpson com n subintervalos**

$$\int_a^b f(x) dx = \frac{h}{3} \left[f(x_1) + 2 \sum_{i=2}^n f(x_{2i-1}) + 4 \sum_{i=1}^n f(x_{2i}) + f(x_{n+1}) \right] + O(h^4) \quad (3.43)$$

Exemplo 3.2.3. Consideramos o problema de computar a integral de $f(x) = xe^{-x^2}$ no intervalo $[0, 1]$. Usando a regra composta de Simpson com n subintervalos, obtemos a aproximação

$$\underbrace{\int_0^1 xe^{-x^2} dx}_I \approx \frac{h}{3} \underbrace{\left[f(x_1) + 2 \sum_{i=2}^n f(x_{2i-1}) + 4 \sum_{i=1}^n f(x_{2i}) + f(x_{n+1}) \right]}_S, \quad (3.44)$$

onde $h = 1/(2n)$ e $x_i = (i-1)h$, $i = 1, 2, \dots, n$. Na Tabela 3.3, temos as aproximações computadas com diversos números de subintervalos, bem como, seus erros absolutos.

Tabela 3.3: Resultados referentes ao Exemplo 3.2.3.

n	S	$ I - S $
1	3.20914e-1	4.9e-3
10	3.16061e-1	3.4e-7
100	3.16060e-1	3.4e-11
1000	3.16060e-1	4.2e-15

Código 3.3: simpson_comp.py

```

1 import numpy as np
2
3 def simpson_comp(f, a, b, n):
4     h = (b-a)/(2*n)
5     S = f(a)
6     for i in range(1, n):
7         x = a + (2*i)*h
8         S += 2*f(x)
9     for i in range(0, n):

```

```
10         x = a + (2*i+1)*h
11         S += 4*f(x)
12     S += f(b)
13     S *= h/3
14     return S
```

Exercícios

E.3.2.1. Aproxime

$$\int_{-1}^0 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (3.45)$$

usando a:

- a) regra composta do ponto médio com 10 subintervalos.
- b) regra composta do trapézio com 10 subintervalos.
- c) regra composta de Simpson com 10 subintervalos.

E.3.2.2. Considere

$$I = \int_{\pi/6}^{\pi/4} e^{-x} \cos(x) dx \quad (3.46)$$

Para cada uma das seguintes quadraturas, compute a aproximação de I com 5 dígitos significativos corretos.

- a) regra composta do ponto médio.
- b) regra composta do trapézio.
- c) regra composta de Simpson.

E.3.2.3. Considere a seguinte tabela de pontos

i	x_i	y_i
1	2.0	1.86
2	2.1	1.90
3	2.2	2.01
4	2.3	2.16
5	2.4	2.23
6	2.5	2.31

Assumindo que $y = f(x)$, e usando o máximo de subintervalos possíveis, calcule:

a) $\int_{2,0}^{2,4} f(x) dx$ usando a regra do ponto médio composta.

b) $\int_{2,0}^{2,5} f(x) dx$ usando a regra do trapézio composta.

c) $\int_{2,0}^{2,4} f(x) dx$ usando a regra de Simpson composta.

3.3 Quadratura de Romberg

[[tag:revisar]]

A quadratura de Romberg é construída por sucessivas extrapolações de Richardson da regra do trapézio composta. Sejam $h_k = (b - a)/(2k)$, $x_i = a + (i - 1)h_k$ e

$$R_{k,1} := \frac{h_k}{2} \left[f(a) + 2 \sum_{i=2}^{2k} f(x_i) + f(b) \right] \quad (3.47)$$

a regra do trapézio composta com $2k$ subintervalos de

$$I := \int_a^b f(x) dx. \quad (3.48)$$

Por sorte, o erro de truncamento de aproximar I por $R_{k,1}$ tem a seguinte forma

$$I - R_{k,1} = \sum_{i=1}^{\infty} k_i h_k^{2i}, \quad (3.49)$$

o que nos permite aplicar a extrapolação de Richardson para obter aproximações de mais alta ordem.

Mais precisamente, para obtermos uma aproximação de I com erro de truncamento da ordem h^{2n} , $h = (b - a)$, computamos $R_{k,1}$ para $k = 1, 2, \dots, n$. Então, usamos das sucessivas extrapolações de Richardson

$$R_{k,j} := R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}, \quad (3.50)$$

$j = 2, 3, \dots, n$, de forma a computarmos $R_{n,n}$, a qual fornece a aproximação desejada.

Exemplo 3.3.1. Consideremos o problema de aproximar a integral de $f(x) = xe^{-x^2}$ no intervalo $[0, 1]$. Para obtermos uma quadratura de Romberg de ordem 4, calculamos

$$R_{1,1} := \frac{1}{2}[f(0) + f(1)] = 1,83940e-1 \quad (3.51)$$

$$R_{2,1} := \frac{1}{4}[f(0) + 2f(1/2) + f(1)] = 2,86670e-1. \quad (3.52)$$

Então, calculando

$$R_{2,2} = R_{2,1} + \frac{R_{2,1} - R_{1,1}}{3} = 3,20914e-1, \quad (3.53)$$

a qual é a aproximação desejada.

Tabela 3.4: Resultados referentes ao Exemplo 3.3.1.

k	$R_{k,1}$	$R_{k,2}$	$R_{k,3}$	$R_{k,4}$
1	1,83940e-1			
2	2,86670e-1	3,20914e-1		
3	3,08883e-1	3,16287e-1	3,15978e-1	
4	3,14276e-1	3,16074e-1	3,16059e-1	3,16061e-1

Na Tabela 3.4, temos os valores de aproximações computadas pela quadratura de Romberg até ordem 8.

Exercícios

[[tag:revisar]]

E.3.3.1. Aproxime

$$\int_{-1}^0 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (3.54)$$

usando a quadratura de Romberg de ordem 4.

3.4 Grau de Exatidão

O grau de exatidão é uma medida de exatidão de uma quadratura numérica. Mais precisamente, dizemos que uma dada quadratura numérica de nodos e pesos $\{(x_i, \omega_i)\}_{i=1}^n$ tem grau de exatidão m , quando

$$\int_a^b p(x) dx = \sum_{i=1}^n p(x_i) \omega_i \quad (3.55)$$

para todo polinômio $p(x)$ de grau menor m . Ou ainda, conforme descrito na definição a seguir.

Definição 3.4.1. (**Grau de Exatidão.**) Dizemos que uma dada quadratura numérica de pontos e nodos $\{x_i, \omega_i\}_{i=1}^n$ tem **grau de exatidão** m , quando

$$\int_a^b x^k dx = \sum_{i=1}^n x_i^k \omega_i, \quad \forall k \leq m. \quad (3.56)$$

3.4.1 Regra do Ponto Médio

Vamos determinar o grau de exatidão da regra do ponto médio. Para tanto, verificamos para quais k vale

$$\int_a^b x^k dx = (b-a) \left(\frac{a+b}{2} \right)^k. \quad (3.57)$$

Temos

- $k = 0$:

$$\int_a^b x^0 dx = x|_a^b = b - a, \quad (3.58)$$

$$(b - a) \left(\frac{a + b}{2} \right)^0 = b - a. \quad (3.59)$$

- $k = 1$:

$$\int_a^b x^1 dx = \frac{x^2}{2} \Big|_a^b = \frac{b^2}{2} - \frac{a^2}{2}, \quad (3.60)$$

$$(b - a) \left(\frac{a + b}{2} \right)^1 = (b - a) \frac{(a + b)}{2} = \frac{b^2}{2} - \frac{a^2}{2}. \quad (3.61)$$

- $k = 2$:

$$\int_a^b x^2 dx = \frac{x^3}{3} \Big|_a^b = \frac{b^3}{3} - \frac{a^3}{3}, \quad (3.62)$$

$$(b - a) \left(\frac{a + b}{2} \right)^2 \neq \frac{b^3}{3} - \frac{a^3}{3}. \quad (3.63)$$

Ou seja, a regra do ponto médio tem grau de exatidão 1. Isto quer dizer, que a regra do ponto médio fornece o valor exato para a integral de qualquer polinômio de grau menor ou igual a 1.

Exemplo 3.4.1. A integral

$$I = \int_1^5 1 - 2x dx \quad (3.64a)$$

$$= x - x^2 \Big|_1^5 \quad (3.64b)$$

$$= (5 - 25) - (1 - 1) = -20. \quad (3.64c)$$

Pela regra do ponto médio, temos

$$S = hf \left(\frac{a + b}{2} \right) \quad (3.65a)$$

$$= (5 - 1) \left[1 - 2 \cdot \frac{(1 + 5)}{2} \right] \quad (3.65b)$$

$$= 4(1 - 6) = -20. \quad (3.65c)$$

3.4.2 Regra de Simpson

Vamos determinar o grau de exatidão da regra de Simpson. Para tanto, verificamos para quais k vale

$$\int_a^b x^k dx = \frac{(b-a)}{6} \left(a^k + 4 \left(\frac{a+b}{2} \right)^k + b^k \right). \quad (3.66)$$

Temos

- $k = 0$:

$$\int_a^b x^0 dx = x|_a^b = b - a, \quad (3.67)$$

$$\frac{(b-a)}{6} \left(a^0 + 4 \left(\frac{a+b}{2} \right)^0 + b^0 \right) = b - a. \quad (3.68)$$

- $k = 1$:

$$\int_a^b x^1 dx = \frac{x^2}{2} \Big|_a^b = \frac{b^2}{2} - \frac{a^2}{2}, \quad (3.69)$$

$$\frac{(b-a)}{6} \left(a^1 + 4 \left(\frac{a+b}{2} \right)^1 + b^1 \right) = \frac{(b-a)}{2} (a+b) \quad (3.70)$$

$$= \frac{b^2}{2} - \frac{a^2}{2}. \quad (3.71)$$

- $k = 2$:

$$\int_a^b x^2 dx = \frac{x^3}{3} \Big|_a^b = \frac{b^3}{3} - \frac{a^3}{3}, \quad (3.72)$$

$$\frac{(b-a)}{6} \left(a^2 + 4 \left(\frac{a+b}{2} \right)^2 + b^2 \right) = \frac{(b-a)}{3} (a^2 + ab + b^2) \quad (3.73)$$

$$= \frac{b^3}{3} - \frac{a^3}{3}. \quad (3.74)$$

- $k = 3$:

$$\int_a^b x^3 dx = \frac{x^4}{4} \Big|_a^b = \frac{b^4}{4} - \frac{a^4}{4}, \quad (3.75)$$

$$\frac{(b-a)}{6} \left(a^3 + 4 \left(\frac{a+b}{2} \right)^3 + b^3 \right) \quad (3.76)$$

$$= \frac{(b-a)}{6} \left[\frac{3a^3}{2} + \frac{3b}{2}a^2 + \frac{3a}{2}b^2 + \frac{3b^3}{2} \right] \quad (3.77)$$

$$= \frac{b^4}{4} - \frac{a^4}{4}. \quad (3.78)$$

- $k = 4$:

$$\int_a^b x^4 dx = \frac{x^5}{5} \Big|_a^b = \frac{b^5}{5} - \frac{a^5}{5}, \quad (3.79)$$

$$\frac{(b-a)}{6} \left(a^4 + 4 \left(\frac{a+b}{2} \right)^4 + b^4 \right) \neq \frac{b^5}{5} - \frac{a^5}{5}. \quad (3.80)$$

Ou seja, a regra de Simpson tem grau de exatidão 3. Isto significa que ela fornece o valor exato da integral de qualquer polinômio de grau menor ou igual a 3.

Exemplo 3.4.2. A integral

$$I = \int_{-1}^1 1 - x^2 dx \quad (3.81a)$$

$$= x - \frac{x^3}{3} \Big|_{-1}^1 \quad (3.81b)$$

$$= \left(1 - \frac{1}{3} \right) - \left(-1 + \frac{1}{3} \right) \quad (3.81c)$$

$$= \frac{4}{3}. \quad (3.81d)$$

Pela regra de Simpson, temos

$$S = \frac{h}{3} \left[f(a) + 4f \left(\frac{a+b}{2} \right) + f(b) \right] \quad (3.82a)$$

$$= \frac{1}{3} \left[(1 - (-1)^2) + 4(1 - 0^2) + (1 - (1)^2) \right] \quad (3.82b)$$

$$= \frac{4}{3}. \quad (3.82c)$$

3.4.3 Exercícios

E.3.4.1. Determine o grau de exatidão da regra do trapézio.

E.3.4.2. Calcule

$$\int_{-7\sqrt{3}}^{7\sqrt{3}} \pi x - e \, dx. \quad (3.83)$$

E.3.4.3. Determine o nodo e o peso da quadratura numérica de um único nodo e de grau de exatidão 1 para o intervalo de integração $[-1, 1]$.

E.3.4.4. Considere uma quadratura numérica de dois nodos e pesos

$$S = f(x_1)\omega_1 + f(x_2)\omega_2. \quad (3.84)$$

Determine as possíveis escolhas de pesos e nodos para que ela tenha grau de exatidão 2 no intervalo de integração $[0, 1]$.

E.3.4.5. Mostre que a seguinte quadratura numérica

$$S = f(-1)\frac{1}{3} + f(0)\frac{4}{3} + f(1)\frac{1}{3} \quad (3.88)$$

tem grau de exatidão 3 no intervalo de integração $[-1, 1]$.

3.5 Quadratura Gauss-Legendre

Quadraturas gaussianas são quadraturas numéricas de máximo grau de exatidão. Especificamente, **quadraturas de Gauss-Legendre** são quadraturas gaussianas para integrais da forma

$$I = \int_{-1}^1 f(x) \, dx. \quad (3.89)$$

Vamos começar considerando o problema de determinar a quadratura de Gauss-Legendre de apenas um ponto, i.e.

$$S = f(x_1)\omega_1. \quad (3.90)$$

Começamos por exigir a integração exata de polinômios de grau 0, o que nos leva a

$$\omega_1 x_1^0 = \int_{-1}^1 x^0 dx \quad (3.91a)$$

$$\omega_1 = x \Big|_{-1}^1 = 2. \quad (3.91b)$$

Agora, exigindo a integração exata de polinômios de grau 1, obtemos

$$\omega_1 x_1^1 = \int_{-1}^1 x^1 dx \quad (3.92a)$$

$$2x_1 = \frac{x^2}{2} \Big|_{-1}^1 = 0 \quad (3.92b)$$

$$x_1 = 0. \quad (3.92c)$$

Com isso, concluímos que a quadratura de um nodo de maior grau de exatidão para tais integrais é a de nodo $x_1 = 0$ e peso $\omega_1 = 2$. Observamos que esta é a regra do ponto médio para o intervalo de integração $[-1, 1]$.

Seguindo esse raciocínio, ao buscarmos por uma **quadratura de n pontos** com maior grau de exatidão possível para integrais no intervalo $[-1, 1]$, acabamos tendo que resolver um sistema de equações

$$\sum_{i=1}^n x_i^k \omega_i = \int_{-1}^1 x^k dx, \quad (3.93)$$

para $k = 0, 1, 2, \dots, 2n - 1$. I.e., no que temos $2n$ incógnitas (n nodos e n pesos) a determinar, podemos exigir o **grau de exatidão máximo de $2n - 1$** .

O sistema (3.93) é um sistema não linear para os nodos e a determinação de soluções para n grande não é uma tarefa trivial. Alternativamente, veremos que **os nodos da quadratura de Gauss-Legendre de n nodos são as raízes do** polinômio de Legendre de grau n . Por definição, o **polinômio de Legendre de grau n** , denotado por $P_n(x)$, satisfaz a seguinte propriedade de ortogonalidade

$$\int_{-1}^1 p(x) P_n(x) dx = 0, \quad (3.94)$$

para todo polinômio $p(x)$ de grau menor que n . Com isso, estabelecemos o seguinte resultado.

Teorema 3.5.1. *A quadratura de Gauss-Legendre de n nodos tem as raízes do polinômio de Legendre de grau n como seus nodos e seus pesos são dados por*

$$\omega_i = \int_{-1}^1 \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} dx. \quad (3.95)$$

Demonstração. Sejam x_1, x_2, \dots, x_n as raízes do polinômio de Legendre de grau n . Queremos mostrar que

$$\int_{-1}^1 p(x) dx = \sum_{i=1}^n p(x_i) \omega_i, \quad (3.96)$$

para todo polinômio $p(x)$ de grau menor ou igual $2n - 1$. Primeiramente, suponhamos que $p(x)$ seja um polinômio de grau menor que n . Então, tomando sua representação por polinômio de Lagrange nos nodos x_i , $i = 1, 2, \dots, n$, temos

$$\int_{-1}^1 p(x) dx = \int_{-1}^1 \sum_{i=1}^n p(x_i) \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} dx \quad (3.97)$$

$$= \sum_{i=1}^n p(x_i) \int_{-1}^1 \prod_{\substack{j=1 \\ j \neq i}}^n \frac{x - x_j}{x_i - x_j} dx \quad (3.98)$$

$$= \sum_{i=1}^n p(x_i) \omega_i. \quad (3.99)$$

Isto mostra o resultado para polinômios $p(x)$ de grau menor que n . Agora, suponhamos que $p(x)$ é um polinômio de grau maior ou igual que n e menor ou igual a $2n - 1$. Dividindo $p(x)$ pelo polinômio de Legendre de grau n , $P_n(x)$, obtemos

$$p(x) = q(x)P_n(x) + r(x), \quad (3.100)$$

onde $q(x)$ e $r(x)$ são polinômio de grau menor que n . Ainda, nas raízes x_1, x_2, \dots, x_n temos $p(x_i) = r(x_i)$ e da ortogonalidade dos polinômios de Legendre (veja, equação (3.94)), temos

$$\int_{-1}^1 p(x) dx = \int_{-1}^1 q(x)P_n(x) + r(x) dx \quad (3.101)$$

$$= \int_{-1}^1 r(x) dx. \quad (3.102)$$

Agora, do resultado anterior aplicado a $r(x)$, temos

$$\int_{-1}^1 p(x) dx = \sum_{i=1}^n r(x_i) \omega_i = \sum_{i=1}^n p(x_i) \omega_i. \quad (3.103)$$

Isto complete o resultado para polinômios de grau menor ou igual a $2n-1$. \square

Exemplo 3.5.1. (Gauss-Legendre de 2 pontos.) Considaremos a quadratura de Gauss-Legendre de 2 nodos. Do Teorema 3.5.1, seus nodos são as raízes do polinômio de Legendre de grau 2

$$P_2(x) = \frac{3}{2}x^2 - \frac{1}{2}, \quad (3.104)$$

as quais são

$$x_1 = -\frac{\sqrt{3}}{3}, \quad (3.105a)$$

$$x_2 = \frac{\sqrt{3}}{3}. \quad (3.105b)$$

Os pesos são, então

$$\omega_1 = \int_{-1}^1 \frac{x - x_1}{x_2 - x_1} dx \quad (3.106a)$$

$$= \frac{\sqrt{3}}{2} \left[\frac{x^2}{2} + \frac{\sqrt{3}}{3}x \right]_{-1}^1 \quad (3.106b)$$

$$= 1 \quad (3.106c)$$

e

$$\omega_2 = \int_{-1}^1 \frac{x - x_2}{x_1 - x_2} dx \quad (3.107a)$$

$$= -\frac{\sqrt{3}}{2} \left[\frac{x^2}{2} - \frac{\sqrt{3}}{3}x \right]_{-1}^1 \quad (3.107b)$$

$$= 1 \quad (3.107c)$$

Ou seja, a quadratura de Gauss-Legendre de 2 pontos tem o seguinte conjunto de nodos e pesos $\{(x_1 = -\sqrt{3}/3, \omega_1 = 1), (x_2 = \sqrt{3}/3, \omega_2 = 1)\}$. Esta, por sua vez, é exata para polinômios de grau menor ou igual a 3. De fato, verificando para potência de x^k temos:

- $k = 0$:

$$\int_{-1}^1 x^0 dx = 2 \quad (3.108a)$$

$$x_1^0 \omega_1 + x_2^0 \omega_2 = \left(-\frac{\sqrt{3}}{3}\right)^0 + \left(\frac{\sqrt{3}}{3}\right)^0 = 2. \quad (3.108b)$$

- $k = 1$:

$$\int_{-1}^1 x^1 dx = 0 \quad (3.109a)$$

$$x_1^1 \omega_1 + x_2^1 \omega_2 = \left(-\frac{\sqrt{3}}{3}\right)^1 + \left(\frac{\sqrt{3}}{3}\right)^1 = 0. \quad (3.109b)$$

- $k = 2$:

$$\int_{-1}^1 x^2 dx = \frac{2}{3} \quad (3.110a)$$

$$x_1^2 \omega_1 + x_2^2 \omega_2 = \left(-\frac{\sqrt{3}}{3}\right)^2 + \left(\frac{\sqrt{3}}{3}\right)^2 = \frac{2}{3}. \quad (3.110b)$$

- $k = 3$:

$$\int_{-1}^1 x^3 dx = 0 \quad (3.111a)$$

$$x_1^3 \omega_1 + x_2^3 \omega_2 = \left(-\frac{\sqrt{3}}{3}\right)^3 + \left(\frac{\sqrt{3}}{3}\right)^3 = 0. \quad (3.111b)$$

- $k = 4$:

$$\int_{-1}^1 x^4 dx = \frac{2}{5} \quad (3.112a)$$

$$x_1^4 \omega_1 + x_2^4 \omega_2 = \left(-\frac{\sqrt{3}}{3}\right)^4 + \left(\frac{\sqrt{3}}{3}\right)^4 = \frac{2}{9}. \quad (3.112b)$$

Tabela 3.5: Conjunto de nodos e pesos da quadratura de Gauss-Legendre.
 Fonte: [Wikipedia:Gauss-Legendre Quadrature](#).

n	x_i	ω_i
1	0	2
2	$\pm \frac{\sqrt{3}}{2}$	1
3	0 $\pm \sqrt{\frac{3}{5}}$	$\frac{8}{9}$ $\frac{5}{9}$
4	$\pm \sqrt{\frac{3}{7} - \frac{2}{7}\sqrt{\frac{6}{5}}}$ $\pm \sqrt{\frac{3}{7} + \frac{2}{7}\sqrt{\frac{6}{5}}}$	$\frac{18 + \sqrt{30}}{36}$ $\frac{18 - \sqrt{30}}{36}$
5	0 $\pm \frac{1}{3}\sqrt{5 - 2\sqrt{\frac{10}{7}}}$ $\pm \frac{1}{3}\sqrt{5 + 2\sqrt{\frac{10}{7}}}$	$\frac{128}{225}$ $\frac{322 + 13\sqrt{70}}{900}$ $\frac{322 - 13\sqrt{70}}{900}$

Exemplo 3.5.2. Considere o problema de obter uma aproximação para $I = \int_{-1}^1 \cos(x) dx$ usando a quadratura de Gauss-Legendre. Calculemos algumas aproximações com $n = 1, 2$ e 3 pontos:

- $n = 1$:

$$\int_{-1}^1 \cos(x) dx \approx 2 \cos 0 \quad (3.113a)$$

$$= 2. \quad (3.113b)$$

- $n = 2$:

$$\int_{-1}^1 x e^{-x^2} dx \approx \cos(-\sqrt{3}/3) + \cos(-\sqrt{3}/3) \quad (3.114a)$$

$$= 1,67582. \quad (3.114b)$$

- $n = 3$:

$$\begin{aligned} \int_{-1}^1 x e^{-x^2} dx &\approx \frac{8}{9} \cos 0 + \frac{5}{9} \cos(-\sqrt{3/5}) \\ &+ \frac{5}{9} \cos(\sqrt{3/5}) = 1,68300. \end{aligned} \quad (3.115a)$$

Na Tabela 3.6, temos as aproximações de I com a quadratura de Gauss-Legendre de $n = 1, 2, 3, 4$ e 5 pontos (denotado por \tilde{I} , bem como, o erro absoluto com respeito ao valor analítico da integral.

Tabela 3.6: Resultados referentes ao Exemplo 3.5.2.

n	\tilde{I}	$ I - \tilde{I} $
1	2.00000	3.2e-01
2	1.67582	7.1e-03
3	1.68300	6.2e-05
4	1.68294	2.8e-07
5	1.68294	7.9e-10

```

1 import numpy as np
2 from numpy.polynomial.legendre import leggauss
3
4 # integrando
5 f = lambda x: np.cos(x)
6 # quadratura
7 n = 4
8 x,w = leggauss(n)
9 # aproximação
10 S = np.sum(f(x)*w)
11 print(f'{n}: S = {S:.5e}')
```

3.5.1 Intervalos de integração arbitrários

A quadratura de Gauss-Legendre é desenvolvida para aproximar integrais definidas no intervalo $[-1, 1]$. Por sorte, uma integral definida em um intervalo arbitrário $[a, b]$ pode ser reescrita como uma integral no intervalo $[-1, 1]$ através de uma mudança de variável apropriada.

Assumindo a **mudança de variável**

$$x = \frac{b-a}{2}(u+1) + a \quad (3.116)$$

temos

$$dx = \frac{b-a}{2} du \quad (3.117)$$

e, portanto,

$$\int_a^b f(x) dx = \int_{-1}^1 f\left(\frac{b-a}{2}(u+1) + a\right) \cdot \frac{b-a}{2} du. \quad (3.118)$$

Portanto, para computarmos $\int_a^b f(x) dx$ podemos aplicar a quadratura de Gauss-Legendre na integral definida no $[-1, 1]$ dada conforme acima.

Exemplo 3.5.3. Usemos a quadratura de Gauss-Legendre com 2 pontos para aproximar a integral

$$\int_0^1 x e^{-x^2} dx. \quad (3.119)$$

Fazendo a mudança de variável $x = u/2 + 1/2$, temos

$$\int_0^1 x e^{-x^2} dx = \int_{-1}^1 \left(\frac{u}{2} + \frac{1}{2}\right) e^{-\left(\frac{u}{2} + \frac{1}{2}\right)^2} du. \quad (3.120)$$

Então, aplicando a quadratura temos

$$\begin{aligned} \int_0^1 x e^{-x^2} dx &= \left(-\frac{\sqrt{3}}{6} + \frac{1}{2}\right) e^{-\left(-\frac{\sqrt{3}}{6} + \frac{1}{2}\right)^2} \\ &\quad + \left(\frac{\sqrt{3}}{6} + \frac{1}{2}\right) e^{-\left(\frac{\sqrt{3}}{6} + \frac{1}{2}\right)^2} \end{aligned} \quad (3.121a)$$

$$= 3,12754e-1. \quad (3.121b)$$

```

1 import numpy as np
2 from numpy.polynomial.legendre import leggauss
3
4 # integral
5 a = 0
6 b = 1
7 f = lambda x: x*np.exp(-x**2)
8 # quadratura
9 n = 2
10 x,w = leggauss(n)
11 # mud de var
12 x = (b-a)/2*(x+1)+a
13 w = (b-a)/2*w
14 # aproximação
15 S = np.sum(f(x)*w)
16 print(f'{n}: S = {S:.5e}')

```

3.5.2 Exercícios

E.3.5.1. Aproxime

$$\int_{-1}^1 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (3.122)$$

usando a quadratura de Gauss-Legendre com:

- a) $n = 1$ ponto.
- b) $n = 2$ pontos.
- c) $n = 3$ pontos.
- d) $n = 4$ pontos.
- e) $n = 5$ pontos.

E.3.5.2. Aproxime

$$\int_0^1 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (3.123)$$

usando a quadratura de Gauss-Legendre com:

- a) $n = 1$ ponto.
- b) $n = 2$ pontos.
- c) $n = 3$ pontos.
- d) $n = 4$ pontos.
- e) $n = 5$ pontos.

E.3.5.3. Aproxime

$$\int_{-1}^1 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (3.124)$$

usando a quadratura de Gauss-Legendre com:

- a) $n = 5$ ponto.
- b) $n = 10$ pontos.
- c) $n = 20$ pontos.

E.3.5.4. Use uma quadratura de Gauss-Legendre para computar a integral

$$I = \int_{-1}^2 x \sin(x^3) dx \quad (3.125)$$

com 6 dígitos significativos corretos.

3.6 Quadraturas gaussianas com pesos

[[tag:revisar]]

A quadratura gaussiana estudada na seção anterior (Seção 3.5) é um caso particular de quadraturas de máximo grau de exatidão para integrais da

forma

$$\int_a^b f(x)w(x) dx, \quad (3.126)$$

onde $w(x)$ é positiva e contínua, chamada de função peso. Como anteriormente, os nodos x_i , $i = 1, 2, \dots, n$, da quadratura gaussiana de n pontos são as raízes do polinômio $p_n(x)$ que é ortogonal a todos os polinômios de grau menor que n . Aqui, isto significa

$$\int_a^b q(x)p_n(x)w(x) dx = 0, \quad (3.127)$$

para todo polinômio $q(x)$ de grau menor que n .

3.6.1 Quadratura de Gauss-Chebyshev

[[tag:revisar]]

Quadraturas de Gauss-Chebyshev são quadraturas gaussianas para integrais da forma

$$\int_{-1}^1 f(x)(1-x^2)^{-1/2} dx. \quad (3.128)$$

Neste caso, na quadratura gaussiana de n pontos os nodos x_i são as raízes do n -ésimo polinômio de Chebyshev $T_n(x)$. Pode-se mostrar (veja, por exemplo, [3, Cap. 7, Sec. 4.1]) que o conjunto de pontos desta quadratura são dados por

$$x_i = \cos\left(\frac{2i-1}{2n}\pi\right), \quad (3.129)$$

$$w_i = \frac{\pi}{n}. \quad (3.130)$$

Exemplo 3.6.1. Considere o problema de aproximar a integral

$$\int_{-1}^1 \frac{e^{-x^2}}{\sqrt{1-x^2}} dx. \quad (3.131)$$

Usando a quadratura de Gauss-Chebyshev de n pontos temos:

- $n = 1$:

$$\int_{-1}^1 \frac{e^{-x^2}}{\sqrt{1-x^2}} dx \approx \pi e^{-\cos(\pi/2)^2} = \pi. \quad (3.132)$$

- $n = 2$:

$$\int_{-1}^1 \frac{e^{-x^2}}{\sqrt{1-x^2}} dx \approx \frac{\pi}{2} e^{-\cos(\pi/4)^2} + \frac{\pi}{2} e^{-\cos(3\pi/4)^2} \quad (3.133)$$

$$= 1,90547. \quad (3.134)$$

- $n = 3$:

$$\int_{-1}^1 \frac{e^{-x^2}}{\sqrt{1-x^2}} dx \approx \frac{\pi}{3} e^{-\cos(\pi/6)^2} + \frac{\pi}{3} e^{-\cos(\pi/2)^2} + \frac{\pi}{3} e^{-\cos(5\pi/6)^2} \quad (3.135)$$

$$= 2,03652. \quad (3.136)$$

n	\tilde{I}
1	3,14159
2	1,90547
3	2,03652
4	2,02581
5	2,02647
6	2,02644
10	2,02644

Tabela 3.7: Resultados referentes ao Exemplo 3.6.1.

Na Tabela 3.7, temos as aproximações \tilde{I} da integral computadas com a quadratura de Gauss-Chebyshev com diferentes números de pontos.

3.6.2 Quadratura de Gauss-Laguerre

[[tag:revisar]]

Quadraturas de Gauss-Laguerre são quadraturas gaussianas para integrais da forma

$$\int_0^{\infty} f(x)e^{-x} dx. \quad (3.137)$$

Neste caso, na quadratura gaussiana de n pontos os nodos x_i são as raízes do n -ésimo polinômio de Laguerre $L_n(x)$ e os pesos por

$$w_i = -\frac{1}{n[L'_n(x_i)]^2}, \quad i = 1, 2, \dots, n. \quad (3.138)$$

Na Tabela 3.8, temos os pontos da quadratura de Gauss-Laguerre para diversos valores de n .

Tabela 3.8: Pontos da quadratura de Gauss-Laguerre.

n	x_i	w_i
1	1,0000000e+00	1,0000000e+00
2	3,4142136e+00	1,4644661e-01
	5,8578644e-01	8,5355339e-01
3	6,2899451e+00	1,0389257e-02
	2,2942804e+00	2,7851773e-01
	4,1577456e-01	7,1109301e-01
4	9,3950709e+00	5,3929471e-04
	4,5366203e+00	3,8887909e-02
	1,7457611e+00	3,5741869e-01
	3,2254769e-01	6,0315410e-01
	1,2640801e+01	2,3369972e-05
5	7,0858100e+00	3,6117587e-03
	3,5964258e+00	7,5942450e-02
	1,4134031e+00	3,9866681e-01
	2,6356032e-01	5,2175561e-01

Exemplo 3.6.2. Na Tabela 3.9, temos as aproximações \tilde{I} da integral $I = \int_0^{\infty} \sin(x)e^{-x} dx$ obtidas pela quadratura de Gauss-Laguerre com diferentes pontos n .

3.6.3 Quadratura de Gauss-Hermite

[[tag:revisar]]

n	\tilde{I}
1	8,41471e-01
2	4,32459e-01
3	4,96030e-01
4	5,04879e-01
5	4,98903e-01

Tabela 3.9: Resultados referentes ao Exemplo 3.6.1.

Quadraturas de Gauss-Hermite são quadraturas gaussianas para integrais da forma

$$\int_{-\infty}^{\infty} f(x) e^{-x^2} dx. \quad (3.139)$$

Seus nodos x_i , $i = 1, 2, \dots, n$ são as raízes do n -ésimo polinômio de Hermite e os pesos são dados por

$$w_i = \frac{2^{n+1} n! \sqrt{\pi}}{[H'_n(x_i)]^2}. \quad (3.140)$$

Na Tabela 3.10, temos os pontos da quadratura de Gauss-Hermite para diversos valores de n .

Exemplo 3.6.3. Na Tabela 3.11, temos as aproximações \tilde{I} da integral $I = \int_{-\infty}^{\infty} x \operatorname{sen}(x) e^{-x^2} dx$ obtidas pela quadratura de Gauss-Hermite com diferentes pontos n .

Exercícios

[[tag:revisar]]

E.3.6.1. Aproxime

$$\int_{-1}^1 \frac{\operatorname{sen}(x+2) - e^{-x^2}}{\sqrt{1-x^2}} dx \quad (3.141)$$

usando a quadratura de Gauss-Chebyshev com:

Tabela 3.10: Pontos da quadratura de Gauss-Hermite.

n	x_i	w_i
1	0,0000000e+00	1,7724539e+00
2	-7,0710678e-01	8,8622693e-01
	7,0710678e-01	8,8622693e-01
3	-1,2247449e+00	2,9540898e-01
	1,2247449e+00	2,9540898e-01
	0,0000000e+00	1,1816359e+00
4	-1,6506801e+00	8,1312835e-02
	1,6506801e+00	8,1312835e-02
	-5,2464762e-01	8,0491409e-01
	5,2464762e-01	8,0491409e-01
5	-2,0201829e+00	1,9953242e-02
	2,0201829e+00	1,9953242e-02
	-9,5857246e-01	3,9361932e-01
	9,5857246e-01	3,9361932e-01
	0,0000000e+00	9,4530872e-01

n	\tilde{I}
1	0,00000e+00
2	8,14199e-01
3	6,80706e-01
4	6,90650e-01
5	6,90178e-01

Tabela 3.11: Resultados referentes ao Exemplo 3.6.3.

- a) $n = 1$ ponto.
- b) $n = 2$ pontos.
- c) $n = 3$ pontos.
- d) $n = 4$ pontos.
- e) $n = 5$ pontos.

E.3.6.2. Aproxime

$$\int_0^\infty (\text{sen}(x+2) - e^{-x^2}) e^{-x} dx \quad (3.142)$$

usando a quadratura de Gauss-Laguerre com:

- a) $n = 3$ pontos.
- b) $n = 4$ pontos.
- c) $n = 5$ pontos.

E.3.6.3. Aproxime

$$\int_{-\infty}^\infty \text{sen}(x+2)e^{-x^2} - e^{-2x^2} dx \quad (3.143)$$

usando a quadratura de Gauss-Hermite com:

- a) $n = 3$ pontos.
- b) $n = 4$ pontos.
- c) $n = 5$ pontos.

3.7 Método de Monte Carlo

[[tag:revisar]]

O método de Monte Carlo é uma técnica não determinística para a aproximação de integrais. Mais especificamente, o método compreende a aproximação

$$\int_a^b f(x) dx \approx \frac{(b-a)}{n} \sum_{i=1}^n f(x_i), \quad (3.144)$$

onde x_1, x_2, \dots, x_n são pontos de uma sequência aleatória em $[a, b]$. Aqui, não vamos entrar em detalhes sobre a escolha desta sequência e, sem mais justificativas, assumiremos uma sequência de pontos uniformemente distribuídos no intervalo de integração.

Exemplo 3.7.1. Na tabela 3.12 temos aproximações \tilde{I} computadas para

$$I = \int_0^1 x e^{-x^2} dx \quad (3.145)$$

usando o método de Monte Carlo com diferentes números de pontos n . Aqui, os pontos foram gerados no GNU Octave pela sequência *quasi*-randômica obtida da função `rand` inicializada com `seed=0`.

n	\tilde{I}	$ I - \tilde{I} $
10	2,53304e-01	6,3e-02
100	3,03149e-01	1,3e-02
1000	3,08415e-01	7,6e-03
10000	3,16385e-01	3,2e-04
100000	3,15564e-01	5,0e-04

Tabela 3.12: Resultados referentes ao Exemplo 3.7.1.

Exercícios

[[tag:revisar]]

E.3.7.1. Use o método de Monte Carlo para obter uma aproximação de

$$\int_{-1}^1 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (3.146)$$

com precisão de 10^{-2} .

Capítulo 4

Problema de Valor Inicial

Neste capítulo, discutimos sobre técnicas numéricas para aproximar a solução de Equações Diferenciais Ordinárias com valor inicial (condição inicial), i.e. problemas da forma

$$\mathbf{y}'(t) = \mathbf{f}(t, \mathbf{y}(t)), \quad t > t_0, \quad (4.1a)$$

$$\mathbf{y}(t_0) = \mathbf{y}_0, \quad (4.1b)$$

onde $\mathbf{y} : t \in \mathbb{R} \mapsto \mathbf{y}(t) \in \mathbb{R}^n$ é a função incógnita com dadas $\mathbf{f} : (t, \mathbf{y}) \in \mathbb{R} \times \mathbb{R}^n \mapsto \mathbf{f}(t, \mathbf{y}) \in \mathbb{R}^n$ e $\mathbf{y}_0 \in \mathbb{R}^n$, $n \geq 1$.

4.1 Método de Euler

Dado um Problema de Valor Inicial (PVI)

$$y'(t) = f(t, y(t)), \quad t > t_0, \quad (4.2a)$$

$$y(t_0) = y_0, \quad (4.2b)$$

temos que $f(t, y)$ é a derivada da solução $y(t)$ no tempo t . Então, aproximando a derivada pela **razão fundamental** de passo $h > 0$

$$y'(t) \approx \frac{y(t+h) - y(t)}{h}, \quad (4.3)$$

obtemos

$$\frac{y(t+h) - y(t)}{h} \approx f(t, y) \quad (4.4)$$

$$y(t+h) \approx y(t) + hf(t, y(t)). \quad (4.5)$$

Isto nos motiva a **iteração do Método de Euler**⁸

$$y^{(0)} = y_0, \quad (4.6a)$$

$$y^{(k+1)} = y^{(k)} + hf(t^{(k)}, y^{(k)}), \quad (4.6b)$$

com $k = 0, 1, 2, \dots, n$, $y^{(k)} \approx y(t^{(k)})$, $t^{(k)} = t_0 + kh$ e **passo** $h > 0$.

Exemplo 4.1.1. Consideramos o seguinte problema de valor inicial

$$y' - y = \text{sen}(t), 0 < t < 1, \quad (4.7a)$$

$$y(0) = \frac{1}{2}. \quad (4.7b)$$

Sua solução analítica é

$$y(t) = e^t - \frac{1}{2} \text{sen}(t) - \frac{1}{2} \cos(t). \quad (4.8)$$

Para computarmos a solução pelo Método de Euler, reescrevemos o problema da seguinte forma

$$y' = y + \text{sen}(t), 0 < t < 1, \quad (4.9a)$$

$$y(0) = \frac{1}{2}, \quad (4.9b)$$

donde identificamos $f(t, y) := y + \text{sen}(t)$, $t_0 = 0$ e $y_0 = 1/2$.

Tabela 4.1: Resultados obtidos para o problema do Exemplo 4.1.1 com $h = 1e-1$.

k	$t^{(k)}$	$y^{(k)}$	$y(t^{(k)})$
0	0.0	5.00e-1	5.00e-1
1	0.1	5.50e-1	5.58e-1
2	0.2	6.15e-1	6.32e-1
3	0.3	6.96e-1	7.24e-1
4	0.4	7.96e-1	8.37e-1
5	0.5	9.14e-1	9.70e-1
6	0.6	1.05e+0	1.13e+0
7	0.7	1.22e+0	1.31e+0
8	0.8	1.40e+0	1.52e+0
9	0.9	1.61e+0	1.76e+0
10	1.0	1.85e+0	2.03e+0

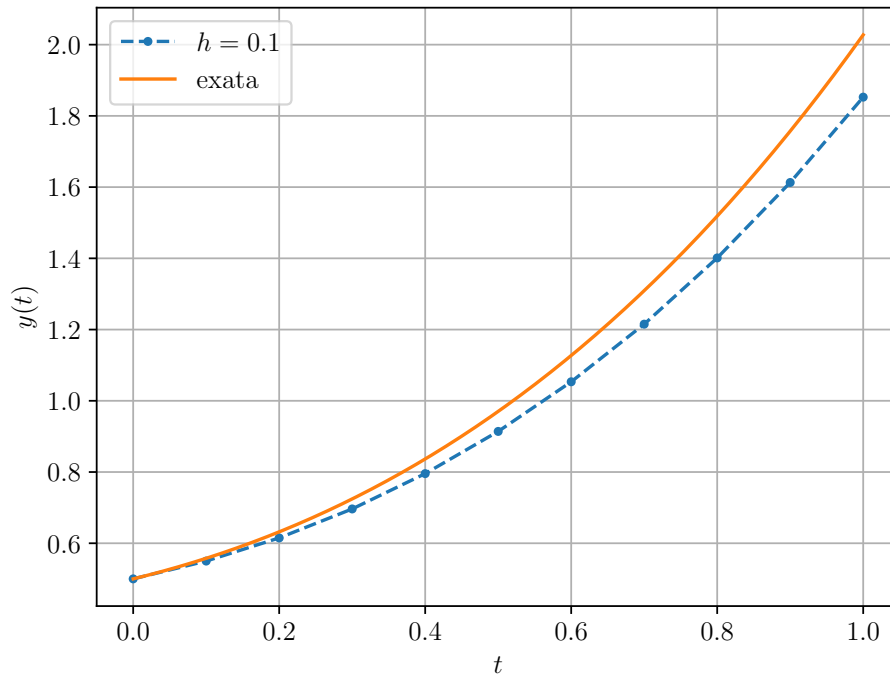


Figura 4.1: Esboço das soluções numérica (pontos) e analítica (linha) para o problema do Exemplo 4.1.1.

Código 4.1: euler.py

```
1 def euler(f, t0, y0, h, n):
2     t = np.empty(n+1)
3     t[0] = t0
4     y = np.empty(n+1)
5     y[0] = y0
6     for k in range(n):
7         t[k+1] = t[k] + h
8         y[k+1] = y[k] + h*f(t[k], y[k])
9     return t, y
```

4.1.1 Análise Numérica

O Método de Euler com passo h aplicado ao problema de valor inicial (4.2), pode ser escrito da seguinte forma

$$\tilde{y}(t^{(0)}; h) = y_0, \quad (4.10a)$$

$$\tilde{y}(t^{(k+1)}; h) = \tilde{y}(t^{(k)}; h) + h\Phi(t^{(k)}, \tilde{y}(t^{(k)}; h), \quad (4.10b)$$

onde $\tilde{y}(t^{(k)})$ representa a aproximação da solução exata y no tempo $t^{(k)} = t_0 + kh$, $k = 0, 1, 2, \dots$. Métodos que podem ser escritos dessa forma, são chamados de **Métodos de Passo Simples** (ou único). No caso específico do Método de Euler, temos

$$\Phi(t, y; h) := f(t, y(t)). \quad (4.11)$$

Consistência

Agora, considerando a solução exata y de (4.2), introduzimos

$$\Delta(t, y; h) := \begin{cases} \frac{y(t+h) - y(t)}{h}, & h \neq 0, \\ f(t, y(t)) & h = 0. \end{cases} \quad (4.12)$$

Com isso, vamos analisar o chamado **erro de discretização local**

$$\tau(t, y; h) := \Delta(t, y; h) - \Phi(t, y; h), \quad (4.13)$$

que estabelece uma medida quantitativa com que a solução exata $y(t)$ no tempo $t+h$ satisfaz a iteração do método de passo simples.

Definição 4.1.1. (Consistência.) Um método de passo simples é dito ser consistente quando

$$\lim_{h \rightarrow 0} \tau(t, y; h) = 0, \quad (4.14)$$

ou, equivalentemente, quando

$$\lim_{h \rightarrow 0} \Phi(t, y; h) = f(t, y). \quad (4.15)$$

Observação 4.1.1. (Consistência do Método de Euler.) Da Definição 4.1.1, temos que o **Método de Euler é consistente**. De fato, temos

$$\lim_{h \rightarrow 0} \tau(t, y; h) = \lim_{h \rightarrow 0} (\Delta(t, y; h) - \Phi(t, y; h)) \quad (4.16)$$

$$= \lim_{h \rightarrow 0} \left(\frac{y(t+h) - y(t)}{h} - f(t, y(t)) \right) \quad (4.17)$$

$$= y'(t) - f(t, y(t)) = 0. \quad (4.18)$$

A **ordem do erro de discretização local** de um método de passo simples é dita ser p , quando

$$\tau(t, y; h) = O(h^p), \quad (4.19)$$

ou seja, quando

$$\lim_{h \rightarrow 0} \frac{\tau(t, y; h)}{h^p} = C, \quad (4.20)$$

para alguma constante C .

Para determinarmos a ordem do Método de Euler, tomamos a **expansão em série de Taylor**⁹ da solução exata $y(t)$ em torno de t , i.e.

$$y(t+h) = y(t) + hy'(t) + \frac{h^2}{2}y''(t) + \frac{h^3}{6}y'''(t + \theta h), \quad (4.21)$$

para algum $0 < \theta < 1$. Como $y'(t) = f(t, y(t))$, temos

$$y''(t) = \frac{d}{dt}f(t, y(t)) \quad (4.22)$$

$$= f_t(t, y) + f_y(t, y)y' \quad (4.23)$$

$$= f_t(t, y) + f_y(t, y)f(t, y). \quad (4.24)$$

Então, rearranjando os termos em (4.21), obtemos

$$\Delta(t, y; h) = f(t, y(t)) + \frac{h}{2}[f_t(t, y) + f_y(t, y)f(t, y)] + O(h^2). \quad (4.25)$$

Portanto, para o Método de Euler temos

$$\tau(t, y; h) := \Delta(t, y; h) - \Phi(t, y; h) \quad (4.26)$$

$$= \Delta(t, y; h) - f(t, y) \quad (4.27)$$

$$= \frac{h}{2} [f_t(t, y) + f_y(t, y)f(t, y)] + O(h^2) \quad (4.28)$$

$$= O(h). \quad (4.29)$$

Isto mostra que o Método de Euler é de ordem 1.

Convergência

A análise acima trata apenas da consistência do Método de Euler. Para analisarmos a convergência de métodos de passo simples, definimos o erro de discretização global

$$e(t; h_n) := \tilde{y}(t; h_n) - y(t), \quad (4.30)$$

onde $\tilde{y}(t; h_n) \approx y(t)$ para $h_n := (t - t_0)/n$. Dizemos que o método é convergente quando

$$\lim_{n \rightarrow \infty} e(t; h_n) = 0. \quad (4.31)$$

Ainda, dizemos que o método tem erro de discretização global de ordem p quando

$$e(t; h_n) = O(h_n^p) \quad (4.32)$$

para todo $t \in [t_0, t_f]$, $t_f > t_0$.

Lema 4.1.1. ([8, Cap. 7, Seção 7.2]) Se a sequência $(\xi^{(k)})_{k \in \mathbb{R}}$ satisfaz a estimativa

$$|\xi^{(k+1)}| \leq (1 + \delta) |\xi^{(k)}| + B, \quad (4.33)$$

para dados $\delta > 0$ e $B \geq 0$, $k = 0, 1, 2, \dots$, então

$$|\xi^{(n)}| \leq e^{n\delta} |\xi^{(0)}| + \frac{e^{n\delta} - 1}{\delta} B. \quad (4.34)$$

Demonstração. De forma iterativa, temos

$$|\xi^{(1)}| \leq (1 + \delta) |\xi^{(0)}| + B \quad (4.35)$$

$$|\xi^{(2)}| \leq (1 + \delta) |\xi^{(1)}| + B \quad (4.36)$$

$$= (1 + \delta)^2 |\xi^{(0)}| + (1 + \delta)B + B \quad (4.37)$$

$$\vdots \quad (4.38)$$

$$|\xi^{(k)}| \leq (1 + \delta)^k |\xi^{(0)}| + B \sum_{k=0}^{k-1} (1 + \delta)^k \quad (4.39)$$

$$= (1 + \delta)^k |\xi^{(0)}| + B \frac{(1 + \delta)^k - 1}{\delta}. \quad (4.40)$$

Observando que $0 < 1 + \delta \leq e^\delta$ para $\delta > -1$, concluímos que

$$|\xi^{(k)}| \leq e^{k\delta} |\xi^{(0)}| + \frac{e^{k\delta} - 1}{\delta} B. \quad (4.41)$$

□

Teorema 4.1.1. (Estimativa do Error Global.) Considere o PVI (4.2), para $t_0 = a$, $y_0 \in \mathbb{R}$. Suponha que f é Lipschitz contínua em y

$$|f(t, y) - f(t, z)| \leq L|y - z|, \quad (4.42)$$

para todo $(t, y) \in [a, b] \times \mathbb{R}$ e que exista $M > 0$ tal que

$$|y''(t)| \leq M, \quad (4.43)$$

para todo $t \in [a, b]$. Então, as iteradas do Método de Euler $y^{(k)} \approx y(t^{(k)})$, $t^{(k)} = t_0 + kh$, $h > (b - a)/n$, $k = 0, 1, 2, \dots, n + 1$, satisfazem a seguinte **estimativa do erro de discretização global**

$$|y^{(k)} - y(t^{(k)})| \leq \frac{hM}{2L} \left[e^{L(t^{(k)} - t_0)} - 1 \right]. \quad (4.44)$$

Demonstração. Para $k = 0$ o resultado é imediato. Agora, usamos o polinômio de Taylor

$$y(t^{(k+1)}) = y(t^{(k)}) + hf(t^{(k)}, y(t^{(k)})) + \frac{h^2}{2} y''(\xi^{(k)}), \quad (4.45)$$

onde $t^{(k)} \leq \xi^{(k)} \leq t^{(k+1)}$, $k = 0, 1, 2, \dots, n$. Já, as iteradas de Euler são

$$y^{(k+1)} = y^{(k)} + hf(t^{(k)}, y^{(k)}). \quad (4.46)$$

Subtraindo essas equações, obtemos

$$\begin{aligned} y^{(k+1)} - y(t^{(k+1)}) &= y^{(k)} - y(t^{(k)}) \\ &\quad + h \left[f(t^{(k)}, y^{(k)}) - f(t^{(k)}, y(t^{(k)})) \right] - \frac{h^2}{2} y''(\xi^{(k)}) \end{aligned} \quad (4.47)$$

Da hipótese de f Lipschitz, temos

$$\begin{aligned} |y^{(k+1)} - y(t^{(k+1)})| &\leq |y^{(k)} - y(t^{(k)})| \\ &\quad + hL |y^{(k)} - y(t^{(k)})| + \frac{h^2}{2} |y''(\xi^{(k)})| \end{aligned} \quad (4.48)$$

Ou, ainda,

$$|y^{(k+1)} - y(t^{(k+1)})| \leq (1 + hL) |y^{(k+1)} - y(t^{(k+1)})| + \frac{h^2 M}{2}. \quad (4.49)$$

Do Lema 4.1.1, temos

$$|y^{(k+1)} - y(t^{(k+1)})| \leq \frac{h^2 M}{2} \frac{e^{khL} - 1}{hL}, \quad (4.50)$$

donde segue a estimativa do erro global (4.44). \square

Observação 4.1.2. (Convergência.) Do Teorema 4.1.1, a ordem do erro de discretização global de um método de passo simples é igual a sua ordem do erro de discretização local. Portanto, o Método de Euler é convergente e é de ordem 1.

Exemplo 4.1.2. Consideramos o seguinte problema de valor inicial

$$y' = y + 1, 0 < t < 1, \quad (4.51a)$$

$$y(0) = 0. \quad (4.51b)$$

Na Tabela 4.2, temos as aproximações $\tilde{y}(1)$ de $y(1)$ computadas pelo Método de Euler com diferentes passos h . A solução analítica deste problema é $y(t) = e^t - 1$.

Tabela 4.2: Resultados referentes ao Exemplo 4.1.2.

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	1.59374	$1.2e-1$
10^{-2}	1.70481	$1.3e-2$
10^{-3}	1.71692	$1.4e-3$
10^{-5}	1.71827	$1.4e-5$
10^{-7}	1.71828	$1.4e-7$
10^{-9}	1.71828	$1.4e-9$

Erros de Arredondamento

O Teorema 4.1.1 não leva em consideração os erros de arredondamento. Levando em conta esses erros, a iteração do Método de Euler tem a forma

$$\tilde{y}^{(0)} = y_0 + \delta^{(k)}, \quad (4.52a)$$

$$\tilde{y}^{(k+1)} = \tilde{y}^{(k)} + hf(t^{(k)}, \tilde{y}^{(k)}) + \delta^{(k+1)}, \quad (4.52b)$$

onde $\delta^{(k)}$ é o erro devido a arredondamentos na k -ésima iterada, $t^{(k)} = t_0 + hk$, $k = 0, 1, 2, \dots, n$. Assumindo as hipóteses do Teorema 4.1.1, podemos mostrar a seguinte estimativa de erro global

$$\begin{aligned} |\tilde{y}^{(k+1)} - y(t^{(k+1)})| &\leq \frac{1}{L} \left(\frac{hM}{2} + \frac{\delta}{h} \right) \left[e^{L(t^{(k)} - t_0)} - 1 \right] \\ &\quad + |\delta_0| e^{L(t^{(k)} - t_0)}, \end{aligned} \quad (4.53)$$

para $\delta^{(k)} < \delta$, $k = 0, 1, 2, \dots, n$.

4.1.2 Sistemas de Equações

Seja um sistema de EDOs¹ com valor iniciais

$$\mathbf{y}' = \mathbf{f}(t, \mathbf{y}), t_0 < t \leq t_f, \quad (4.54a)$$

$$\mathbf{y}(t_0) = \mathbf{y}_0, \quad (4.54b)$$

com dada $\mathbf{f} : (t, \mathbf{y}) \in [t_0, t_f] \times \mathbb{R}^m \mapsto \mathbb{R}^m$, dados valores iniciais $\mathbf{y}_0 \in \mathbb{R}^m$ e incógnita $\mathbf{y} : t \in [t_0, t_f] \mapsto \mathbb{R}^m$, $n \geq 1$.

¹Equações Diferenciais Ordinárias

Do ponto de vista algorítmico, a **iteração do Método de Euler** é diretamente estendida **para sistemas**:

$$\begin{aligned} \mathbf{y}^{(0)} &= \mathbf{y}_0, \\ \mathbf{y}^{(k+1)} &= \mathbf{y}^{(k)} + h \mathbf{f}(t^{(k)}, \mathbf{y}^{(k)}), \end{aligned} \quad (4.55)$$

para $\mathbf{y}^{(k)} \approx \mathbf{y}(t^{(k)})$, $t^{(k)} = t_0 + kh$, $h = (t_f - t_0)/n$, $k = 0, 1, 2, \dots, n$.

Exemplo 4.1.3. Consideramos o sistema de EDOs

$$y_1' = -y_1 + y_2 - e^{-t} - \sin(t) + \cos(t), \quad (4.56a)$$

$$y_2' = 2y_1 + 3y_2 - 6e^t - 2\cos(t), \quad (4.56b)$$

para $0 < t \leq 1$ com condições iniciais

$$y_1(0) = 0, \quad (4.57a)$$

$$y_2(0) = 3. \quad (4.57b)$$

Este sistema tem solução analítica

$$y_1(t) = e^t - 2e^{-t} + \cos(t), \quad (4.58a)$$

$$y_2(t) = 2e^t + e^{-t}. \quad (4.58b)$$

Podemos reescrevê-lo na forma vetorial

$$\underbrace{\begin{bmatrix} y_1' \\ y_2' \end{bmatrix}}_{\mathbf{y}'(t)} = \underbrace{\begin{bmatrix} -y_1 + y_2 + e^{-t} - \sin(t) + \cos(t) \\ 2y_1 + 3y_2 - 6e^t - 2\cos(t) \end{bmatrix}}_{\mathbf{f}(t, \mathbf{y})}, 0 < t \leq t_f \quad (4.59a)$$

$$\underbrace{\begin{bmatrix} y_1(0) \\ y_2(0) \end{bmatrix}}_{\mathbf{y}(0)} = \underbrace{\begin{bmatrix} 0 \\ 3 \end{bmatrix}}_{\mathbf{y}_0} \quad (4.59b)$$

Usando o Método de Euler com $h = 10^{-2}$ obtemos as soluções mostradas na figura abaixo.

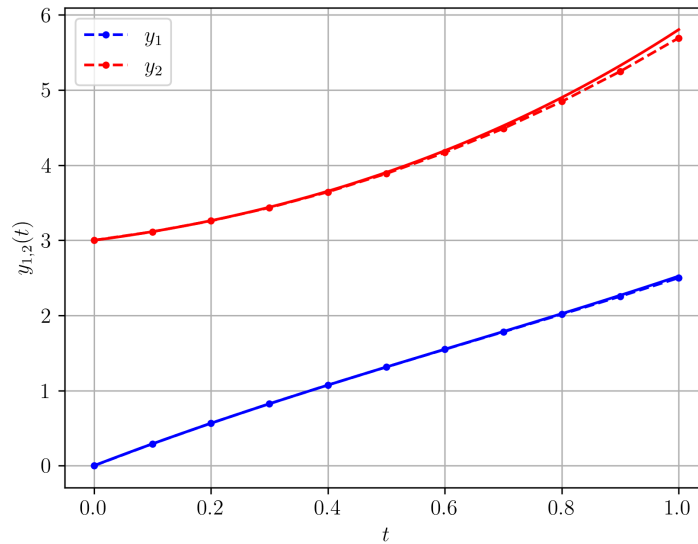


Figura 4.2: Soluções numérica (linha pontilhada) *versus* analítica (linha contínua) para o PVI do Exemplo 4.1.3.

```

1 import numpy as np
2
3 def euler(f, t0, y0, h, n):
4     t = np.empty(n+1)
5     m = y0.size
6     y = np.empty((n+1, m))
7
8     t[0] = t0
9     y[0] = y0
10
11     for k in range(n):
12         t[k+1] = t[k] + h
13         y[k+1] = y[k] + h*f(t[k], y[k])
14     return t, y
15
16 def f(t, y):
17     v = np.array([-y[0] + y[1] \
18                  - np.exp(-t) \

```

```

19         + np.cos(t) \
20         - np.sin(t), \
21         2*y[0] + 3*y[1]
22         - 6*np.exp(t)
23         - 2*np.cos(t)])
24     return v
25
26
27 h = 1e-2
28 n = round(1./h)
29 t0 = 0.
30 y0 = np.array([0., 3.])
31 t, y = euler(f, t0, y0, h, n)

```

4.1.3 Equações de Ordem Superior

Seja dado o PVI de ordem m

$$\frac{d^m y}{dt^m} = f\left(t, y, \frac{dy}{dt}, \dots, \frac{d^{m-1}y}{dt^{m-1}}\right), \quad (4.60a)$$

$$y(t_0) = y_0, \left. \frac{dy}{dt} \right|_{t=0} = y'_0, \dots, \left. \frac{d^{(m-1)}y}{dt^{(m-1)}} \right|_{t=0} = y_0^{(m-1)}, \quad (4.60b)$$

para $t_0 \leq t \leq t_f$.

Para resolvê-lo com o Método de Euler, a ideia é reescrevê-lo como um sistema de EDOs de primeira ordem com condições iniciais. Isso pode ser feito com a mudança de variáveis

$$u_1 = y, \quad (4.61)$$

$$u_2 = \frac{dy}{dt}, \quad (4.62)$$

$$u_3 = \frac{d^2 y}{dt^2}, \quad (4.63)$$

$$\vdots \quad (4.64)$$

$$u_m = \frac{d^{m-1}y}{dt^{m-1}}. \quad (4.65)$$

Com isso e do PVI (4.60), obtemos o sistema de EDOs de primeira ordem

$$u_1' = u_2, \quad (4.66a)$$

$$u_2' = u_3, \quad (4.66b)$$

$$u_3' = u_4, \quad (4.66c)$$

$$\vdots \quad (4.66d)$$

$$u_m' = f(t, u_1, u_2, \dots, u_m), \quad (4.66e)$$

para $t_0 < t \leq t_f$ e com condições iniciais

$$u_1(t_0) = y_0, \quad (4.67a)$$

$$u_2(t_0) = y_0', \quad (4.67b)$$

$$u_3(t_0) = y_0'', \quad (4.67c)$$

$$\vdots \quad (4.67d)$$

$$u_m(t_0) = y_0^{(m-1)}. \quad (4.67e)$$

Exemplo 4.1.4. Consideramos o seguinte PVI de ordem superior

$$y'' - ty' + y = (2 + t)e^{-t} - t \cos(t), 0 < t \leq 1, \quad (4.68a)$$

$$y(0) = 1, y'(0) = 0. \quad (4.68b)$$

Sua solução analítica é

$$y(t) = \sin(t) + e^{-t}. \quad (4.69)$$

Para reescrevê-lo como uma sistema de EDOs de primeira ordem, tomamos as mudanças de variáveis $u_1 = y$ e $u_2 = y'$. Com isso, obtemos

$$u_1' = u_2, \quad (4.70a)$$

$$u_2' = tu_2 - u_1 + (2 + t)e^{-t} - t \cos(t), \quad (4.70b)$$

para $0 < t \leq t_f$ e com condições iniciais

$$u_1(0) = 1, \quad (4.71a)$$

$$u_2(0) = 0. \quad (4.71b)$$

Com passo $h = 10^{-2}$, o Método de Euler aplicado a este sistema fornece a solução do PVI mostrada na figura abaixo.

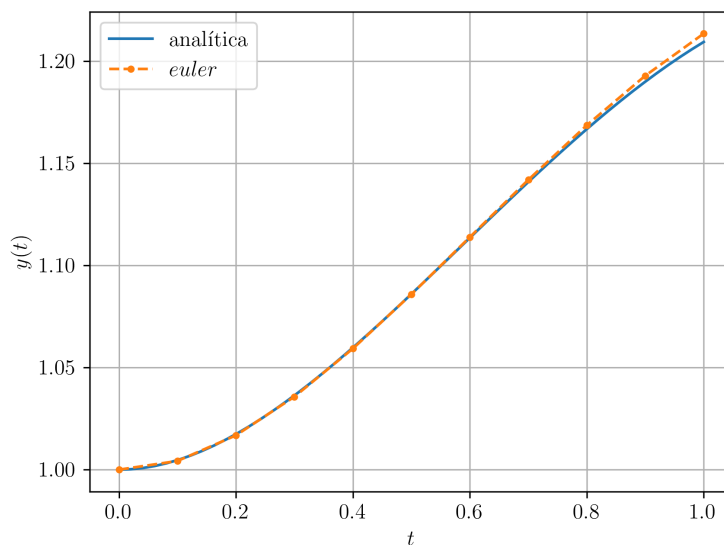


Figura 4.3: Solução numérica *versus* analítica computadas para o PVI do Exemplo 4.1.4.

4.1.4 Exercícios

E.4.1.1. O problema de valor inicial

$$y' = \pi [\cos^2(\pi t) - \sin^2(\pi t)], \quad 0 < t \leq 1.5, \quad (4.72a)$$

$$y(0) = 0. \quad (4.72b)$$

tem solução analítica $y(t) = \sin(\pi t) \cos(\pi t)$. Compute a aproximação $\tilde{y}(1.5; h) \approx y(1.5)$ pelo Método de Euler com passo $h = 10^{-1}$ e forneça o erro $e(1.5; h) := \tilde{y}(1.5; h) - y(1.5)$

E.4.1.2. Use o Método de Euler para computar a solução de

$$y' = e^{2t} - 2y, \quad 0 < t \leq 1, \quad (4.73a)$$

$$y(0) = 0. \quad (4.73b)$$

Escolha um passo h adequado de forma que $y(1)$ seja computado com precisão de 5 dígitos significativos.

E.4.1.3. Considere o seguinte problema de valor inicial

$$y' + e^{-y^2+1} = 2, \quad t > 1, \quad (4.74a)$$

$$y(1) = -1. \quad (4.74b)$$

Use o Método de Euler para computar o valor aproximado de $y(2)$ com precisão de 6 dígitos significativos.

E.4.1.4. Use o Método de Euler para computar a solução de

$$y' = -30y, \quad 0 < t \leq 1, \quad (4.75a)$$

$$y(0) = \frac{1}{3} \quad (4.75b)$$

A solução analítica é $y(t) = \frac{1}{3}e^{-30t}$. Compute a solução aproximação $\tilde{y}(1)$ e o erro $|\tilde{y}(1) - y(1)|$ usando o passo $h = 10^{-1}$. O erro obtido está de acordo com a estimativa (4.44)?

E.4.1.5. Para o sistema de EDOs do Exemplo 4.1.3, verifique a ordem de convergência do Método de Euler computando o erro $\varepsilon = \|\tilde{\mathbf{y}}(1) - \mathbf{y}(1)\|$ com diferentes tamanhos de passos $h = 10^{-1}, 10^{-2}, \dots, 10^{-6}$.

E.4.1.6. Para o PVI de segunda ordem dado no Exemplo 4.1.4, tente computar a solução para tempos finais $t_f = 2, 3, \dots, 5$. Faça uma comparação gráfica entre as soluções numérica e analítica. O que ocorre ao aumentarmos o tempo final? Justifique sua resposta.

Análise Numérica

E.4.1.7. Mostre que se $\delta > -1$, então $0 < 1 + \delta \leq e^\delta$.

E.4.1.8. Seja dado um PVI (4.2), $t_0 \leq t \leq t_f$. Sejam $\tilde{y}^{(k)}$, $k = 0, 1, 2, \dots, n$, as aproximações computadas conforme em (4.52), com $\delta^{(k)} < \delta$. Assumindo as mesmas hipóteses do Teorema 4.1.1, mostre a estimativa de erro global (4.53).

E.4.1.9. Assumindo um erro de arredondamento máximo de $\delta > 0$, use (4.53) para obter uma estimativa para a melhor escolha de h .

4.2 Métodos de Taylor de Alta Ordem

Métodos de Taylor¹⁰ são usados para computar a solução numérica de Problemas de Valor Inicial (PVI) da forma

$$y' = f(t, y), \quad t_0 < t \leq t_f, \quad (4.76a)$$

$$y(t_0) = y_0, \quad (4.76b)$$

onde $y : [t_0, t_f] \mapsto \mathbb{R}$ é a função incógnita, dada $f : [t_0, t_f] \times \mathbb{R} \rightarrow \mathbb{R}$ e dado valor inicial $y_0 \in \mathbb{R}$.

Na Seção 4.1, vimos que a ordem do erro de discretização local do Método de Euler¹¹ é também a do erro de discretização global. Este resultado é generalizado pelo Teorema 4.2.1, para todo o método de passo simples

$$y^{(0)} = y_0, \quad (4.77a)$$

$$y^{(k+1)} = y^{(k)} + h\Phi(t^{(k)}, y^{(k)}), \quad (4.77b)$$

onde $y^{(k)} \approx y(t^{(k)})$, $t^{(k)} = t_0 + kh$, $h = (t_f - t_0)/n$, $k = 0, 1, 2, \dots, n$.

Antes, lembramos que o erro de discretização local é definido por

$$\tau(t, y; h) := \Delta(t, y; h) - \Phi(t, y; h), \quad (4.78)$$

onde

$$\Delta(t, y; h) := \begin{cases} \frac{y(t+h) - y(t)}{h} & , h \neq 0, \\ f(t, y(t)) & , h = 0. \end{cases} \quad (4.79)$$

Já, o **erro de discretização global** é definido por

$$e(t; h_n) := \tilde{y}(t; h_n) - y(t), \quad (4.80)$$

onde $\tilde{y}(t; h_n) \approx y(t)$ dada por (4.77) para $h_n = (t - t_0)/n$.

Com o objetivo de desenvolvermos métodos de alta ordem, podemos usar o polinômio de Taylor de ordem m de $y = y(t)$

$$\begin{aligned} y(t+h) &= y(t) + hy'(t) + \frac{h^2}{2}y''(t) \\ &+ \cdots + \frac{h^m}{m!} \frac{d^m y}{dt^m}(t) + \frac{h^{m+1}}{(m+1)!} \frac{d^{m+1} y}{dt^{m+1}}(\xi), \end{aligned} \quad (4.81)$$

donde

$$\begin{aligned} y(t+h) &= y(t) + hf(t, y) + \frac{h^2}{2}f'(t, y) \\ &+ \cdots + \frac{h^m}{m!} \frac{d^{m-1} f}{dt^{m-1}}(t, y) \\ &+ \frac{h^{m+1}}{(m+1)!} \frac{d^m f}{dt^m}(\xi, y(\xi)) \end{aligned} \quad (4.82)$$

e, portanto

$$\begin{aligned} \Delta(t, y; h) &= f(t, y) + \frac{h}{2}f'(t, y) \\ &+ \cdots + \frac{h^m}{m!} \frac{d^{m-1} f}{dt^{m-1}}(t, y) \\ &+ \frac{h^{m+1}}{(m+1)!} \frac{d^m f}{dt^m}(\xi, y(\xi)) \end{aligned} \quad (4.83)$$

Isto nos motiva a **iteração do Método de Taylor de Ordem m** :

$$y^{(0)} = y_0, \quad (4.84a)$$

$$y^{(k+1)} = y^{(k)} + hT^{(m)}(t^{(k)}, y^{(k)}), \quad (4.84b)$$

onde

$$\begin{aligned} T^{(m)}(t^{(k)}, y^{(k)}) &:= f(t^{(k)}, y^{(k)}) + \frac{h}{2}f'(t^{(k)}, y^{(k)}) \\ &+ \cdots + \frac{h^{m-1}}{m!} \frac{d^{m-1} f}{dt^{m-1}}(t^{(k)}, y^{(k)}) \end{aligned} \quad (4.85)$$

Exemplo 4.2.1. Considere o PVI

$$y' = y + \sin(t), \quad 0 < t \leq 1, \quad (4.86a)$$

$$y(0) = \frac{1}{2}. \quad (4.86b)$$

Vamos usar o Método de Taylor de Ordem 2 para computar sua solução e comparar com a solução analítica

$$y(t) = e^t - \frac{1}{2} \sin(t) - \frac{1}{2} \cos(t). \quad (4.87)$$

h	$ \tilde{y}(1) - y(1) $
10^{-1}	4.9e-3
10^{-2}	5.2e-5
10^{-3}	5.2e-7
10^{-4}	5.2e-9
10^{-5}	5.2e-11

Código 4.2: taylor.py

```

1 import numpy as np
2
3 def taylor(Phi, t0, y0, h, n):
4     t = t0
5     y = y0
6     for k in range(n):
7         y += h*Phi(t, y, h)
8         t += h
9     return t, y
10
11 def f(t, y):
12     return y + np.sin(t)
13
14 def fl(t, y):
15     return f(t, y) + np.cos(t)
16
17 def Phi(t, y, h):
18     return f(t, y) + h/2*fl(t, y)
19

```



```

20 # analítica
21 def exata(t):
22     return np.exp(t) - 0.5*np.sin(t) - 0.5*np.cos(
    t)
23
24 h = 1e-1
25 n = round(1/h)
26 t,y = taylor(Phi, 0., 0.5, h, n)

```

4.2.1 Análise Numérica

Teorema 4.2.1. (Convergência, [8, Cap. 7, Seção 7.2].) Considere o PVI (4.76), para $t_0 \in [a, b]$ e $y_0 \in \mathbb{R}$. Seja Φ contínua em

$$G := \{(t, y, h) : a \leq t \leq b, |y - y(t)| \leq \gamma, 0 \leq |h| \leq h_0\}, \quad (4.88)$$

para $h_0 > 0$ e $\gamma > 0$. Sejam também, M, N constantes tais que

$$|\Phi(t, y; h) - \Phi(t, z; h)| \leq M|y - z|, \quad (4.89)$$

para todas $(t, y; h), (t, z; h) \in G$. Se, ainda, para algum $p > 0$ e para todo $t \in [a, b]$, $|h| \leq h_0$, temos a **estimativa do erro de discretização local**

$$|\tau(t, y(t); h)| \leq N|h|^p, \quad (4.90)$$

então existe \bar{h} , $0 < \bar{h} < h_0$, tal que vale a seguinte **estimativa do erro de discretização global**

$$|e(t; h_n)| \leq |h_n|^p N \frac{e^{M|t-t_0|} - 1}{M}, \quad (4.91)$$

para todo $t \in [a, b]$ e para todo $h_n = (t - t_0)/n$, $n = 1, 2, \dots$, com $|h_n| \leq \bar{h}$.

Demonstração. Seja

$$\tilde{\Phi}(t, y; h) := \begin{cases} \Phi(t, y; h) & , (t, y, h) \in G, \\ \Phi(t, y(t) + \gamma; h) & , t \in [a, b], |h| \leq h_0, y \geq y(t) + \gamma, \\ \Phi(t, y(t) - \gamma; h) & , t \in [a, b], |h| \leq h_0, y \leq y(t) - \gamma, \end{cases} \quad (4.92)$$

A função $\tilde{\Phi}$ é contínua em

$$\tilde{G} := \{(t, y; h) : t \in [a, b], y \in \mathbb{R}, |h| \geq h_0\} \quad (4.93)$$

e satisfaz

$$\left| \tilde{\Phi}(t, y; h) - \tilde{\Phi}(t, z; h) \right| \leq M|y - z|, \quad (4.94)$$

para todas $(t, y; h), (t, z; h) \in \tilde{G}$. Ainda, como $\tilde{\Phi}(t, y(t); h) = \Phi(t, y(t); h)$, também temos que

$$|\Delta(t, y(t); h) - \tilde{\Phi}(t, y(t); h)| \leq N|h|^p, \quad (4.95)$$

para $t \in [a, b]$ e $|h| \leq h_0$.

Sejam, $\tilde{y}^{(k)} := \tilde{y}(t^{(k)}; h)$, $t^{(k)} = t_0 + kh$, $\tilde{y}^{(0)} = y_0$:

$$\tilde{y}^{(k+1)} = \tilde{y}^{(k)} + h\tilde{\Phi}(t^{(k)}, \tilde{y}^{(k)}; h), y(t^{(k+1)}) = y(t^{(k)}) + h\Delta(t^{(k)}, y(t^{(k)}; h)). \quad (4.96)$$

Definindo $\tilde{e}^{(k)} := \tilde{y}^{(k)} - y(t^{(k)})$, obtemos a fórmula de recorrência

$$\tilde{e}^{(k+1)} = \tilde{e}^{(k)} + h \left[\tilde{\Phi}(t^{(k)}, \tilde{y}^{(k)}; h) - \Delta(t^{(k)}, y(t^{(k)}; h)) \right] \quad (4.97)$$

$$= \tilde{e}^{(k)} + h \left[\tilde{\Phi}(t^{(k)}, \tilde{y}^{(k)}; h) - \tilde{\Phi}(t^{(k)}, y(t^{(k)}; h)) \right] \quad (4.98)$$

$$+ h \left[\tilde{\Phi}(t^{(k)}, y(t^{(k)}; h)) - \Delta(t^{(k)}, y(t^{(k)}; h)) \right]. \quad (4.99)$$

Agora, de (4.94) e (4.95), temos

$$\left| \tilde{\Phi}(t^{(k)}, \tilde{y}^{(k)}; h) - \tilde{\Phi}(t^{(k)}, y(t^{(k)}; h)) \right| \leq M|\tilde{e}^{(k)}| \quad (4.100)$$

$$\left| \Delta(t^{(k)}, y(t^{(k)}; h)) - \tilde{\Phi}(t^{(k)}, y(t^{(k)}; h)) \right| \leq N|h|^p \quad (4.101)$$

Portanto, de (4.99), temos

$$\left| \tilde{e}^{(k+1)} \right| \leq (1 + |h|M) \left| \tilde{e}^{(k)} \right| + N|h|^{p+1} \quad (4.102)$$

Então, do Lema 4.1.1, temos

$$\left| \tilde{e}^{(k)} \right| \leq N|h|^p \frac{e^{k|h|M} - 1}{M}. \quad (4.103)$$

Sejam, agora, $t \in [a, b]$, $t \neq t_0$ fixo e $h := h_n = (t - t_0)/n$, $n > 0$. Então, $t^{(n)} = t_0 + nh = t$ e de (4.103) temos

$$\left| \tilde{e}(t, h_n) \right| \leq N|h_n|^p \frac{e^{M|t-t_0|} - 1}{M}, \quad (4.104)$$

para todo $t \in [a, b]$, $|h_n| \leq h_0$. Uma vez que $|t - t_0| \leq |b - a|$ e $\gamma > 0$, existe \bar{h} , $0 < \bar{h} \leq h_0$, tal que $|\tilde{e}(t, h_n)| \leq \gamma$ para todo $t \in [a, b]$ e $|h_n| \leq \bar{h}$. Logo, para o método de passo simples (4.77) gerado por Φ , temos para $|h| \leq \bar{h}$ que

$$\tilde{y}^{(k)} = y^{(k)}, \quad (4.105)$$

$$\tilde{e}^{(k)} = e^{(k)}, \quad (4.106)$$

$$\tilde{\Phi}(t^{(k)}, \tilde{y}^{(k)}; h) = \Phi(t^{(k)}, y^{(k)}; h). \quad (4.107)$$

Concluimos que

$$|e(t, h_n)| \leq N|h_n|^p \frac{e^{M|t-t_0|} - 1}{M}, \quad (4.108)$$

para todo $t \in [a, b]$ e $h_n = (t - t_0)/n$, $n = 1, 2, \dots$, com $|h_n| \leq \bar{h}$. \square

4.2.2 Exercícios

E.4.2.1. Use o Método de Taylor de $O(h^2)$ para computar a solução de

$$y' + \cos(t) = y, \quad 0 < t \leq 1, \quad (4.109a)$$

$$y(0) = \frac{1}{2}. \quad (4.109b)$$

A solução analítica é $y(t) = \frac{1}{2} \cos(t) - \frac{1}{2} \sin(t)$. Faça testes numéricos com $h = 10^{-1}$, 10^{-2} , 10^{-3} e 10^{-4} , observe os resultados obtidos e o erro $\varepsilon := |\tilde{y}(1) - y(1)|$, onde \tilde{y} corresponde a solução numérica. O erro tem o comportamento esperado? Justifique sua resposta.

E.4.2.2. Use o Método de Taylor $O(h^2)$ para computar a solução do PVI (4.109) com $h = 10^{-1}$. Faça um esboço do gráfico do erro $e(t; h = 10^{-1}) = |\tilde{y}(t) - y(t)|$ e verifique se ele tem a forma esperada conforme a estimativa do erro global (4.91).

E.4.2.3. Use o Método de Taylor de $O(h^3)$ para computar a solução do PVI (4.109). Escolha o passo h de forma que a solução numérica tenha precisão de 6 dígitos significativos.

E.4.2.4. Considere o seguinte PVI

$$y' = y^2 - ty, \quad 1 < t \leq 2, \quad (4.110a)$$

$$y(1) = -2. \quad (4.110b)$$

Compute a solução com o Método de Taylor de $O(h^p)$ com passo $h = 10^{-1}$:

a) $p = 2$.

b) $p = 3$.

c) $p = 4$.

E.4.2.5. Considere o seguinte PVI

$$y' - t^2 y = 0, \quad 1 < t \leq 3, \quad (4.111a)$$

$$y(1) = \frac{1}{2}. \quad (4.111b)$$

Compute a solução com o Método de Taylor de $O(h^p)$ com passo $h = 10^{-1}$:

a) $p = 2$.

b) $p = 3$.

c) $p = 4$.

Análise Numérica

E.4.2.6. Considere o PVI (4.109). Verifique que o Método de Taylor de $O(h^2)$ satisfaz as estimativas do erro local (4.90) e do erro global (4.91). Forneça valor estimados para os parâmetros N e M .

4.3 Métodos de Runge-Kutta

Seja um Problema de Valor Inicial (PVI) da forma

$$y' = f(t, y), \quad t_0 < t \leq t_f, \quad (4.112a)$$

$$y(t_0) = y_0, \quad (4.112b)$$

onde $y : [t_0, t_f] \mapsto \mathbb{R}$ é a função incógnita, dada $f : [t_0, t_f] \times \mathbb{R} \rightarrow \mathbb{R}$ e dado valor inicial $y_0 \in \mathbb{R}$. Seguimos usando a notação $y^{(k)} \approx y(t^{(k)})$, $t^{(k)} = t_0 + kh$, $k = 0, 1, 2, \dots, n$, $h = (t_f - t_0)/n$.

Os métodos de Runge¹²-Kutta¹³ de s -estágios são métodos de passo simples da seguinte forma

$$y^{(k+1)} = y^{(k)} + h \underbrace{\sum_{i=1}^s c_i \phi_i(t^{(k)}, y^{(k)})}_{:=\Phi(t^{(k)}, y^{(k)})}, \quad (4.113)$$

onde

$$\phi_1 := f(t^{(k)}, y^{(k)}), \quad (4.114a)$$

$$\phi_2 := f(t^{(k)} + \alpha_2 h, y^{(k)} + h\beta_{2,1}\phi_1), \quad (4.114b)$$

$$\phi_3 := f(t^{(k)} + \alpha_3 h, y^{(k)} + h(\beta_{3,1}\phi_1 + \beta_{3,2}\phi_2)), \quad (4.114c)$$

\vdots

$$\phi_s := f\left(t^{(k)} + \alpha_s h, y^{(k)} + h \sum_{j=1}^{s-1} \beta_{s,j} \phi_j\right), \quad (4.114d)$$

com os coeficientes $c_i, \alpha_i, i = 1, 2, \dots, s$ e $\beta_{i,j}, j = 1, 2, \dots, s-1$, escolhidos de forma a obtermos um método de passo simples com erro local da ordem desejada.

Na sequência, discutimos alguns dos métodos de Runge-Kutta usualmente utilizados. Pode-se encontrar uma lista mais completa em [3, Cap. 8, Seção 3.2].

4.3.1 Métodos de Runge-Kutta de ordem 2

Precisamos apenas de 2 estágios para obtermos métodos de Runge-Kutta de ordem 2. Tomamos a forma

$$y^{(k+1)} = y^{(k)} + h \underbrace{(c_1 \phi_1 + c_2 \phi_2)}_{:=\Phi(t^{(k)}, y^{(k)})} \quad (4.115)$$

com

$$\phi_1(t^{(k)}, y^{(k)}) := f(t^{(k)}, y^{(k)}), \quad (4.116a)$$

$$\phi_2(t^{(k)}, y^{(k)}) := f(t^{(k)} + \alpha_2 h, y^{(k)} + h\beta_{2,1}f(t^{(k)}, y^{(k)})). \quad (4.116b)$$

Nosso objetivo é de determinar os coeficientes $c_1, c_2, \alpha_2, \beta_{2,1}$ tais que o método (4.115) tenha erro de discretização local de $O(h^2)$. Da definição do erro local (4.78)

$$\tau(t, y; h) := \Delta(t, y; h) - \Phi(t, y; h), \quad (4.117)$$

e por polinômio de Taylor de $y(t)$ ²

$$\Delta(t, y; h) = f(t, y(t)) + \frac{h}{2} \frac{d}{dt} f(t, y) + O(h^2) \quad (4.118)$$

$$\begin{aligned} &= f(t, y(t)) + \frac{h}{2} [f_t(t, y) \\ &\quad + f_y(t, y)f(t, y)] + O(h^2). \end{aligned} \quad (4.119)$$

De (4.115), temos

$$\Phi(t, y; h) = c_1 f(t, y) + c_2 f(t + \alpha_2 h, y + h\beta_{2,1}f(t, y)) \quad (4.120)$$

Agora, tomando a expansão por série de Taylor de $\Phi(t, y; h)$, temos

$$\begin{aligned} \Phi(t, y; h) &= (c_1 + c_2)f(t, y) + c_2 h [\alpha_2 f_t(t, y) \\ &\quad + \beta_{2,1} f_y(t, y)f(t, y)] + O(h^2). \end{aligned} \quad (4.121)$$

Então, por comparação de (4.119) e (4.121), temos

$$c_1 + c_2 = 1 \quad (4.122)$$

$$c_2 \alpha_2 = \frac{1}{2} \quad (4.123)$$

$$c_2 \beta_{21} = \frac{1}{2}. \quad (4.124)$$

Este sistema tem mais de uma solução possível.

²Consulte (4.83) para mais detalhes sobre a expansão em polinômio de Taylor de $\Delta(t, y; h)$.

Método do Ponto Médio

O Método do Ponto Médio é um método de Runge-Kutta de ordem 2 proveniente da escolha de coeficientes

$$\begin{aligned} c_1 &= 0, \\ c_2 &= 1, \\ \alpha_2 &= \frac{1}{2}, \\ \beta_{2,1} &= \frac{1}{2}. \end{aligned} \tag{4.125}$$

Logo, a iteração do Método do Ponto Médio é

$$\begin{aligned} y^{(0)} &= y_0 \\ y^{(k+1)} &= y^{(k)} + hf\left(t^{(k)} + \frac{h}{2}, y^{(k)} + \frac{h}{2}f(t^{(k)}, y^{(k)})\right), \end{aligned} \tag{4.126}$$

com $k = 0, 1, 2, \dots, n$.

Exemplo 4.3.1. Consideramos o seguinte PVI

$$y' - y = \sin(t), 0 < t \leq 1, \tag{4.127a}$$

$$y(0) = \frac{1}{2}. \tag{4.127b}$$

Na Tabela 4.3, temos as aproximações $\tilde{y}(1) \approx y(1)$ computadas pelo Método do Ponto Médio com diferentes passos h .

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	2.02175	$5.6\text{e}-03$
10^{-2}	2.02733	$6.0\text{e}-05$
10^{-3}	2.02739	$6.1\text{e}-07$
10^{-4}	2.02740	$6.1\text{e}-09$
10^{-5}	2.02740	$6.1\text{e}-11$
10^{-6}	2.02740	$1.9\text{e}-12$

Tabela 4.3: Resultados referentes ao Exemplo 4.3.1.

Código 4.3: pm.py

```

1 import numpy as np
2
3 def pm(f, t0, y0, h, n):
4     t = t0
5     y = y0
6     for k in range(n):
7         ya = y + h/2*f(t, y)
8         y += h*f(t+h/2, ya)
9         t += h
10    return t, y
11
12 def f(t, y):
13     return y + np.sin(t)
14
15 # analítica
16 def exata(t):
17     return np.exp(t) - 0.5*np.sin(t) - 0.5*np.cos(
18         t)
19
20 h = 1e-1
21 n = round(1./h)
22 t,y = pm(f, 0., 0.5, h, n)
23 print(f'{h:.1e}: {y:.5e} {np.abs(y-exata(1)):.1e}'
24       )

```

Método de Euler Modificado

O Método de Euler Modificado é um método de Runge-Kutta de ordem 2 proveniente da escolha de coeficientes

$$\begin{aligned}
 c_1 &= \frac{1}{2}, \\
 c_2 &= \frac{1}{2}, \\
 \alpha_2 &= 1, \\
 \beta_{21} &= 1.
 \end{aligned}
 \tag{4.128}$$

Logo, a **iteração do Método de Euler Modificado** é

$$\begin{aligned} y^{(0)} &= y_0 \\ y^{(k+1)} &= y^{(k)} + \frac{h}{2} \left[f(t^{(k)}, y^{(k)}) \right. \\ &\quad \left. + f(t^{(k)} + h, y^{(k)} + hf(t^{(k)}, y^{(k)})) \right]. \end{aligned} \quad (4.129)$$

Exemplo 4.3.2. Consideremos o seguinte problema de valor inicial

$$y' - y = \sin(t), t > 0 \quad (4.130a)$$

$$y(0) = \frac{1}{2}. \quad (4.130b)$$

Na Tabela 4.4, temos as aproximações $\tilde{y}(1)$ de $y(1)$ computadas pelo Método de Euler modificado com diferentes passos h .

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	2.02096	$6.4e-03$
10^{-2}	2.02733	$6.9e-05$
10^{-3}	2.02739	$6.9e-07$
10^{-4}	2.02740	$6.9e-09$
10^{-5}	2.02740	$6.9e-11$
10^{-6}	2.02740	$2.0e-12$

Tabela 4.4: Resultados referentes ao Exemplo 4.3.2

Código 4.4: eulerm.py

```

1 import numpy as np
2
3 def eulerm(f, t0, y0, h, n):
4     t = t0
5     y = y0
6     for k in range(n):
7         ya = y + h*f(t, y)
8         y += h/2 * (f(t, y) \
9                     + f(t+h, ya))
10        t += h

```

```

11     return t, y
12
13 def f(t, y):
14     return y + np.sin(t)
15
16 # analítica
17 def exata(t):
18     return np.exp(t) - 0.5*np.sin(t) - 0.5*np.cos(
19         t)
20 h = 1e-1
21 n = round(1./h)
22 t,y = eulerm(f, 0., 0.5, h, n)
23 print(f'{h:.1e}: {y:.5e} {np.abs(y-exata(1)):.1e}'
24       )

```

4.3.2 Método de Runge-Kutta de ordem 4

Um dos métodos de Runge-Kutta mais empregados é o seguinte método de ordem 4:

$$\begin{aligned}
 y^{(0)} &= y_0, \\
 y^{(k+1)} &= y^{(k)} + \frac{h}{6}(\phi_1 + 2\phi_2 + 2\phi_3 + \phi_4),
 \end{aligned}
 \tag{4.131}$$

com

$$\phi_1 := f\left(t^{(k)}, y^{(k)}\right), \tag{4.132a}$$

$$\phi_2 := f\left(t^{(k)} + h/2, y^{(k)} + h\phi_1/2\right), \tag{4.132b}$$

$$\phi_3 := f\left(t^{(k)} + h/2, y^{(k)} + h\phi_2/2\right), \tag{4.132c}$$

$$\phi_4 := f\left(t^{(k)} + h, y^{(k)} + h\phi_3\right), \tag{4.132d}$$

Exemplo 4.3.3. Consideremos o seguinte PVI

$$y' - y = \sin(t), t > 0 \tag{4.133a}$$

$$y(0) = \frac{1}{2}. \tag{4.133b}$$

Na Tabela 4.5, temos as aproximações $\tilde{y}(1) \approx y(1)$ computadas pelo Método de Runge-Kutta de Quarta Ordem com diferentes passos h .

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	2.02739	$2.8\text{e}-06$
10^{-2}	2.02740	$3.1\text{e}-10$
10^{-3}	2.02740	$3.0\text{e}-14$
10^{-4}	2.02740	$4.4\text{e}-14$

Tabela 4.5: Resultados referentes ao Exemplo 4.3.3

4.3.3 Exercícios

E.4.3.1. Considere o seguinte problema de valor inicial

$$y' + e^{-y^2+1} = 2, 1 < t \leq 2, \quad (4.134)$$

$$y(1) = -1. \quad (4.135)$$

Use os seguintes métodos de Runge-Kutta com passo $h = 0,1$ para computar o valor aproximado de $y(2)$:

- a) Método do Ponto Médio.
- b) Método de Euler Modificado.
- c) Método de Runge-Kutta de Quarta Ordem.

E.4.3.2. (4.109) Considere o seguinte problema de valor inicial

$$y' + \cos(t) = y, \quad 0 < t \leq 1, \quad (4.136a)$$

$$y(0) = \frac{1}{2}. \quad (4.136b)$$

A solução analítica é $y(t) = \frac{1}{2} \cos(t) - \frac{1}{2} \sin(t)$. Faça testes numéricos com $h = 10^{-1}, 10^{-2}, 10^{-3}$ e 10^{-4} , observe os resultados obtidos e o erro $\varepsilon := |\tilde{y}(1) - y(1)|$, onde \tilde{y} corresponde a solução numérica. Faça testes para:

- a) Método do Ponto Médio.

- b) Método de Euler Modificado.
- c) Método de Runge-Kutta de Quarta Ordem.

O erro tem o comportamento esperado? Justifique sua resposta.

E.4.3.3. Considere os métodos de Runge-Kutta aplicados para computar a solução do PVI (4.109). Para cada um, faça um esboço do gráfico do erro $e(t; h = 10^{-1}) = |\tilde{y}(t) - y(t)|$ e verifique se ele tem a forma esperada conforme a estimativa do erro global (4.91).

E.4.3.4. Mostre que o **Método de Kutta** é $O(h^3)$. Sua iteração é definida por

$$\begin{aligned} y^{(0)} &= y_0, \\ y^{(k+1)} &= y^{(k)} + \frac{h}{6} (\phi_1 + 4\phi_2 + \phi_3), \end{aligned} \quad (4.137)$$

com $k = 0, 1, 2, \dots, n$, onde

$$\begin{aligned} \phi_1 &= f(t, y) \\ \phi_2 &= f(t + h/2, y + h\phi_1/2) \\ \phi_3 &= f(t + h, y - h\phi_1 + 2h\phi_2). \end{aligned} \quad (4.138)$$

Aplique-o para o PVI dado no Exercício (4.109) e verifique se o erro global satisfaz a ordem esperada.

E.4.3.5. Considere o seguinte PVI

$$y' = y^2 - ty, \quad 1 < t \leq 2, \quad (4.139a)$$

$$y(1) = -2. \quad (4.139b)$$

Use os seguintes métodos de Runge-Kutta com passo $h = 0.1$ para computar o valor aproximado de $y(2)$:

- a) Método do Ponto Médio.
- b) Método de Euler Modificado.

c) Método de Runge-Kutta de Quarta Ordem.

E.4.3.6. Considere o seguinte PVI

$$y' - t^2 y = 0, \quad 1 < t \leq 3, \quad (4.140a)$$

$$y(1) = \frac{1}{2}. \quad (4.140b)$$

Use os seguintes métodos de Runge-Kutta com passo $h = 10^{-2}$ para computar o valor aproximado de $y(3)$:

a) Método do Ponto Médio.

b) Método de Euler Modificado.

c) Método de Runge-Kutta de Quarta Ordem.

4.4 Método de Euler Implícito

Seja o Problema de Valor Inicial (PVI)

$$y' = f(t, y), t_0 < t \leq t_f, \quad (4.141a)$$

$$y(t_0) = y_0, \quad (4.141b)$$

com dados $t_0, t_f \in \mathbb{R}$ e $y_0 \in \mathbb{R}$, sendo a função $y = y(t)$ a incógnita.

Consideramos a discretização no tempo $t^{(k)} := t_0 + hk$, $k = 0, 1, 2, \dots, n$, com passo $h := (t_f - t_0)/n$. De (4.141) e do Teorema Fundamental do Cálculo temos

$$y(t^{(k+1)}) = y(t^{(k)}) + \int_{t^{(k)}}^{t^{(k+1)}} f(t, y) dt. \quad (4.142)$$

A integral pode ser aproximada por

$$\int_{t^{(k)}}^{t^{(k+1)}} f(t, y) dt \approx hf(t^{(k+1)}, y^{(k+1)}) \quad (4.143)$$

donde obtemos

$$y(t^{(k+1)}) \approx y(t^{(k)}) + hf(t^{(k+1)}, y^{(k+1)}). \quad (4.144)$$

Isto nos motiva a **iteração do Método de Euler Implícito**¹⁴

$$\begin{aligned} y^{(0)} &= y_0, \\ y^{(k+1)} &= y^{(k)} + hf(t^{(k+1)}, y^{(k+1)}), \end{aligned} \quad (4.145)$$

sendo $y^{(k)} \approx y(t^{(k)})$, $k = 0, 1, 2, \dots, n$.

Exemplo 4.4.1. Consideremos o seguinte PVI

$$y' - y = \sin(t), t > 0 \quad (4.146a)$$

$$y(0) = \frac{1}{2}. \quad (4.146b)$$

Na Tabela 4.6, temos as aproximações $\tilde{y}(1) \approx y(1)$ computadas pelo Método de Euler Implícito com diferentes passos h .

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	2.23660	$2.1e-1$
10^{-2}	2.04660	$1.9e-2$
10^{-3}	2.02930	$1.9e-3$
10^{-4}	2.02759	$1.9e-4$
10^{-5}	2.02741	$1.9e-5$
10^{-6}	2.02740	$1.9e-6$

Tabela 4.6: Resultados referentes ao Exemplo 4.4.1

Código 4.5: eulerImp.py

```
1 import numpy as np
2 from scipy.optimize import fsolve
3
4 def eulerimp(f, t0, y0, h, n):
5     t = t0
6     y = y0
7     for k in range(n):
8         y = fsolve(lambda x:
9                     x - y - h*f(t+h, x),
10                    x0 = y, xtol=1e-14)[0]
11     t += h
```

```

12     return t, y
13
14 def f(t, y):
15     return y + np.sin(t)
16
17 # analítica
18 def exata(t):
19     return np.exp(t) - 0.5*np.sin(t) - 0.5*np.cos(
20         t)
21
22 h = 1e-1
23 n = round(1./h)
24 t,y = eulerimp(f, 0., 0.5, h, n)
25 print(f'{h:.1e}: {y:.5e} {np.abs(y-exata(1)):.1e}'
26       )

```

4.4.1 Análise Numérica

O Método de Euler Implícito é $O(h)$. De fato, tomando o polinômio de Taylor

$$y(t) = y(t+h) - hf(t+h, y(t+h)) + O(h^2), \quad (4.147)$$

temos

$$\tau(t, y; h) := \Delta(t, y; h) - \Phi(t, y; h) \quad (4.148)$$

$$= \underbrace{\frac{y(t+h) - y(t)}{h}}_{:=\Delta} - \underbrace{f(t+h, y(t+h); h)}_{:=\Phi} \quad (4.149)$$

$$= O(h). \quad (4.150)$$

Estabilidade

Um método é dito ser estável quando pequenas perturbações na condição inicial produzem pequenas alterações nas aproximações subsequentes, i.e. os resultados dependem continuamente dos dados iniciais.

Exemplo 4.4.2. Consideramos o seguinte PVI

$$y' = -40y, 0 < t \leq 1, \quad (4.151a)$$

$$y(0) = \frac{1}{3}. \quad (4.151b)$$

A solução exata é $y(t) = \frac{1}{5}e^{-40t}$. Na tabela abaixo, temos os resultados obtidos por computações com o Método de Euler (Explícito, \tilde{y}_e) e o Método de Euler Implícito (\tilde{y}_i) para $h = 10^{-1}$ e 10^{-2} .

h	$ \tilde{y}_e(1) - y(1) $	$ \tilde{y}_i(1) - y(1) $
10^{-1}	2.0e+04	3.8e-08
10^{-2}	1.4e-18	8.1e-16

Estabilidade do Euler Explícito

Consideramos o PVI

$$y' = \lambda y, t > 0, \quad (4.152a)$$

$$y(0) = y_0, \quad (4.152b)$$

para dados $\lambda < 0$ e $y_0 \in \mathbb{R}$.

A iteração do [Método de Euler Explícito](#) para este PVI consiste em

$$\begin{aligned} y^{(0)} &= y_0, \\ y^{(k+1)} &= y^{(k)} + h\lambda y^{(k)}, \end{aligned} \quad (4.153)$$

donde temos

$$y^{(k+1)} = (1 + h\lambda)^{k+1} y_0. \quad (4.154)$$

Tendo em vista a solução exata $y(t) = y_0 e^{\lambda t}$, temos que o erro global é

$$\left| y(t^{(k)}) - y^{(k)} \right| = \left| y_0 e^{\lambda h k} - y_0 (1 + h\lambda)^k \right| \quad (4.155)$$

$$= \left| (e^{\lambda h})^k - (1 + h\lambda)^k \right| |y_0| \quad (4.156)$$

e, portanto, a exatidão é determinada por quão bem $1 + h\lambda$ aproxima $e^{\lambda h}$. Observamos que, para qualquer $\lambda < 0$, $(e^{\lambda h})^k \rightarrow 0$ quando $t \rightarrow \infty$. Por outro lado, para $y^{(k)} \rightarrow 0$, quando $t \rightarrow 0$, é necessário que $|1 + h\lambda| < 1$, i.e. o passo do Método de Euler fica restrito

$$h < \frac{2}{|\lambda|} \quad (4.157)$$

Supondo um erro de arredondamento δ_0 (apenas) na condição inicial, as aproximações subsequentes do Método de Euler Explícito ficariam

$$y^{(k+1)} = (1 + h\lambda)^{k+1} (y_0 + \delta_0), \quad (4.158)$$

donde temos que

$$\delta^{(k)} = (1 + h\lambda)^k \delta_0 \quad (4.159)$$

é o valor propagado de δ_0 na k -ésima iteração. Ou seja, quando $|1 + h\lambda| > 1$, temos que $\delta^{(k)} \rightarrow \infty$ quando $k \rightarrow \infty$ e o método é instável. Concluimos que o Método de Euler Explícito é estável para

$$h < \frac{2}{|\lambda|}. \quad (4.160)$$

Estabilidade do Euler Implícito

O Método de Euler Implícito é incondicionalmente estável. Para o PVI (4.152), o método produz as aproximações

$$y^{(k)} = (1 - h\lambda)^{-k} y_0. \quad (4.161)$$

Aqui, para qualquer $\lambda < 0$, temos que

$$(1 - h\lambda)^{-k} \rightarrow 0, \quad k \rightarrow \infty, \quad (4.162)$$

para qualquer escolha do passo $h > 0$. Isto mostra a estabilidade incondicional do método. Também, o Exercício 4.4.7 mostra que o método é convergente para o PVI (4.152).

4.4.2 Exercícios

E.4.4.1. Considere o seguinte problema de valor inicial

$$y' = y + 1, \quad 0 < t \leq 1, \quad (4.163a)$$

$$y(0) = 0, \quad (4.163b)$$

com solução exata $y(t) = e^t - 1$. Use o Método de Euler Implícito com $h = 10^{-1}$ para computar uma aproximação de $y(1)$. Então, verifique a ordem de convergência para diferentes passos $h = 10^{-1}, 10^{-2}, 10^{-3}$ e 10^{-4} .

E.4.4.2. Considere o seguinte problema de valor inicial

$$y' = -50y + 50, 0 < t \leq 1, \quad (4.165a)$$

$$y(0) = 2, \quad (4.165b)$$

com solução exata $y(t) = 1 + e^{-50t}$. Com $h = 10^{-1}$, compute a aproximação de $y(1)$ dada pelo

a) Método de Euler Explícito.

b) Método de Euler Implícito.

Por que os resultados são tão diferentes entre os métodos? Escolha um passo h em que ambos produzam resultados satisfatórios e justifique sua escolha.

E.4.4.3. Use o Método de Euler Implícito, com $h = 10^{-1}$, para computar aproximações para a solução do PVI discutido no Exemplo 4.1.1. Compare os resultados com aqueles apresentados com o Método de Euler Explícito.

E.4.4.4. O problema de valor inicial

$$y' = \pi [\cos^2(\pi t) - \sin^2(\pi t)], \quad t > 0, \quad (4.166a)$$

$$y(0) = 0. \quad (4.166b)$$

tem solução analítica $y(t) = \sin(\pi t) \cos(\pi t)$. Compute a aproximação $\tilde{y}(1.5) \approx y(1.5)$ pelo Método de Euler Implícito com passo $h = 10^{-1}$ e forneça o erro $\varepsilon := |\tilde{y}(1.5, h) - y(1.5)|$.

E.4.4.5. Use o Método de Euler Implícito para computar a solução de

$$y' = e^{2t} - 2y, \quad 0 < t \leq 1, \quad (4.167a)$$

$$y(0) = 0. \quad (4.167b)$$

Escolha um passo h adequado de forma que $y(1)$ seja computado com precisão de 5 dígitos significativos.

E.4.4.6. Considere o seguinte problema de valor inicial

$$y' + e^{-y^2+1} = 2, \quad t > 1, \quad (4.168a)$$

$$y(1) = -1. \quad (4.168b)$$

Use o Método de Euler Implícito para computar o valor aproximado de $y(2)$ com precisão de 6 dígitos significativos.

Análise Numérica

E.4.4.7. Mostre que o Método de Euler Implícito é convergente para a solução exata do PVI (4.152) para qualquer $\lambda < 0$.

4.5 Métodos de Passo Múltiplo

Seja um Problema de Valor Inicial (PVI)

$$y'(t) = f(t, y(t)), \quad t_0 < t \leq t_f, \quad (4.173a)$$

$$y(t_0) = y_0. \quad (4.173b)$$

Assumimos uma discretização uniforme no tempo $t^{(k)} = t_0 + kh$, com tamanho de passo $h = (t_f - t_0)/n$. Do Teorema Fundamental do Cálculo, temos

$$y(t^{(k+i)}) = y(t^{(k-j)}) + \int_{t^{(k-j)}}^{t^{(k+i)}} f(s, y(s)) ds. \quad (4.174)$$

A ideia é aproximar a integral por uma quadratura de Newton¹⁵-Cotes¹⁶. Das regras³, temos

$$\int_{t^{(k-j)}}^{t^{(k+i)}} f(s, y(s)) ds \approx \sum_{l=1}^m f(s^{(l)}, y(s^{(l)})) w^{(l)}, \quad (4.175)$$

onde $s^{(l)}$ são os nodos e $w^{(l)}$ os pesos da quadratura, $l = 1, 2, \dots, m$.

³Consulte as [Notas de Aula: Matemática Numérica II: Integração: Regras de Newton-Cotes](#).

4.5.1 Métodos de Adams-Bashforth

Métodos de Adams-Bashforth são métodos explícitos de passo múltiplo obtidos ao escolhermos $j = 0$ e $i = 1$ em (4.175), i.e.

$$y(t^{(k+1)}) = y(t^{(k)}) + \int_{t^{(k)}}^{t^{(k+1)}} f(s, y(s)) ds. \quad (4.176)$$

Aplicando as regras de Newton-Cotes, escolhemos os nodos de quadratura $s_l = t^{(k-l+1)}$, $l = 1, 2, \dots, m$, e, então

$$\int_{t^{(k)}}^{t^{(k+1)}} f(x, y(x)) dx \approx \sum_{l=1}^m f(s_l, y(s_l)) w_l, \quad (4.177)$$

e

$$w_l = \int_{t^{(k)}}^{t^{(k+1)}} \prod_{\substack{p=1 \\ p \neq l}}^m \frac{s - s^{(p)}}{s^{(l)} - s^{(p)}} ds. \quad (4.178)$$

Agora, fazendo a mudança de variável $u = (s - t^{(k)})/h$, obtemos

$$w_l = h \int_0^1 \prod_{\substack{p=1 \\ p \neq l}}^m \frac{u + p - 1}{p - l} du \quad (4.179)$$

Donde, obtemos o seguinte esquema numérico

$$y^{(k+1)} = y^{(k)} + h \sum_{l=1}^m w_l f(t^{(k-l+1)}, y^{(k-l+1)}), \quad (4.180)$$

onde

$$w_l = \int_0^1 \prod_{\substack{p=1 \\ p \neq l}}^m \frac{s + p - 1}{p - l} ds. \quad (4.181)$$

Observação 4.5.1. (Ordem de Truncamento.) A ordem de truncamento de um Método de Adams-Bashforth de m -passos é $O(h^m)$ [2].

Método de Adams-Bashforth de Ordem 2

Tomando $m = 2$ em (4.181), temos

$$c_1 = \int_0^1 s + 1 \, ds = \frac{3}{2} \quad (4.182)$$

e

$$c_2 = \int_0^1 -s \, ds = -\frac{1}{2}. \quad (4.183)$$

Então, de (4.180) temos a **iteração do método de Adams-Bashforth de 2 passos**:

$$\begin{aligned} y^{(0)} &= y_0, \\ y^{(1)} &= \tilde{y}_1, \\ y^{(k+1)} &= y^{(k)} + \frac{h}{2} \left[3f(t^{(k)}, y^{(k)}) - f(t^{(k-1)}, y^{(k-1)}) \right], \end{aligned} \quad (4.184)$$

com $t^{(k)} = t_0 + kh$, $h = (t_f - t_0)/n$, $k = 0, 1, 2, \dots, n$.

Exemplo 4.5.1. Consideramos o seguinte PVI

$$y' - y = \sin(t), t > 0 \quad (4.185a)$$

$$y(0) = \frac{1}{2}. \quad (4.185b)$$

Na Tabela 4.7, temos as aproximações $\tilde{y}(1)$ de $y(1)$ computadas pelo Método de Adams-Bashforth de 2 passos. Como este método é de ordem 2, escolhemos inicializá-lo pelo método do ponto médio, de forma a mantermos a consistência.

Tabela 4.7: Resultados referentes ao Exemplo 4.5.1

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	2.01582	$1.2e-02$
10^{-2}	2.02727	$1.3e-04$
10^{-3}	2.02739	$1.3e-06$
10^{-4}	2.02740	$1.3e-08$
10^{-5}	2.02740	$1.3e-10$

Código 4.6: abs2.py

```
1 import numpy as np
2
3 def ab2(f, t0, y0, h, n):
4
5     # inicialização
6     y1 = y0 + h/2*f(t0, y0)
7     y1 = y0 + h*f(t0+h/2, y1)
8     t1 = t0 + h
9
10    # iterações
11    for k in range(1,n):
12        y = y1 + h/2*(3*f(t1, y1) \
13                    - f(t0, y0))
14        t = t1 + h
15
16        t0 = t1
17        y0 = y1
18
19        t1 = t
20        y1 = y
21
22    return t, y
23
24 def f(t, y):
25     return y + np.sin(t)
26
27 # analítica
28 def exata(t):
29     return np.exp(t) - 0.5*np.sin(t) - 0.5*np.cos(
30         t)
31 h = 1e-1
32 n = round(1./h)
33 t,y = ab2(f, 0., 0.5, h, n)
34 print(f'{h:.1e}: {y:.5e} {np.abs(y-exata(1)):.1e}'
35       )
```

Método de Adams-Bashforth de Ordem 4

Tomando $m = 4$ em (4.181) obtemos, de (4.180), a **iteração do método de Adams-Bashforth de 4 passos**

$$\begin{aligned}
 y^{(0)} &= y_0, \\
 y^{(1)} &= \tilde{y}_1, \\
 y^{(2)} &= \tilde{y}_2, \\
 y^{(k+1)} &= y^{(k)} + \frac{h}{24} \left[55f(t^{(k)}, y^{(k)}) \right. \\
 &\quad \left. - 59f(t^{(k-1)}, y^{(k-1)}) + 37f(t^{(k-2)}, y^{(k-2)}) \right. \\
 &\quad \left. - 9f(t^{(k-3)}, y^{(k-3)}) \right],
 \end{aligned} \tag{4.186}$$

Exemplo 4.5.2. Consideremos o seguinte problema de valor inicial

$$y' - y = \sin(t), t > 0 \tag{4.187}$$

$$y(0) = \frac{1}{2}. \tag{4.188}$$

Na Tabela 4.8, temos as aproximações $\tilde{y}(1)$ de $y(1)$ computadas pelo método de Adams-Bashforth de 4 passos. Como este método é de ordem 3, escolhemos inicializá-lo pelo método de Runge-Kutta de ordem 4, de forma a mantermos a consistência.

Tabela 4.8: Resultados referentes ao Exemplo 4.5.2

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	2.02735	$5.0\text{e}-05$
10^{-2}	2.02740	$7.7\text{e}-09$
10^{-3}	2.02740	$7.9\text{e}-13$

Código 4.7: ab4.py

```

1 import numpy as np
2
3 def ab4(f, t0, y0, h, n):
4
5     t = np.empty(5)

```

```
6     t[0] = t0
7     y = np.empty(5)
8     y[0] = y0
9
10    # inicialização
11    for k in range(3):
12        phi1 = f(t[k], y[k])
13        phi2 = f(t[k]+h/2, y[k] + h*phi1/2)
14        phi3 = f(t[k]+h/2, y[k] + h*phi2/2)
15        phi4 = f(t[k]+h, y[k] + h*phi3)
16
17        y[k+1] = y[k] + h/6 \
18            * (phi1 + 2*phi2 + 2*phi3 + phi4)
19        t[k+1] = t[k] + h
20
21    # iterações
22    for k in range(3,n):
23        y[4] = y[3] + h/24*(55*f(t[3], y[3]) \
24            - 59*f(t[2], y[2]) \
25            + 37*f(t[1], y[1]) \
26            - 9*f(t[0], y[0]))
27        t[4] = t[3] + h
28
29        t[:4] = t[1:]
30        y[:4] = y[1:]
31
32    return t[4], y[4]
33
34 def f(t, y):
35     return y + np.sin(t)
36
37 # analítica
38 def exata(t):
39     return np.exp(t) - 0.5*np.sin(t) - 0.5*np.cos(
40         t)
41
42 h = 1e-3
43 n = round(1./h)
```



```

43 t,y = ab4(f, 0., 0.5, h, n)
44 print(f'{h:.1e}: {y:.5e} {np.abs(y-exata(1)):.1e} '
      )

```

4.5.2 Métodos de Adams-Moulton

Métodos de Adams-Moulton são esquemas implícitos obtidos tomando-se $i = 1, j = 0$ em (4.174) e incluindo-se $t^{(k+1)}$ como nodo da quadratura em (4.175).

Método de Adams-Moulton de 2 Passos

A iteração do de Adams-Moulton de 2 Passos (A-B-2)⁴ é

$$\begin{aligned}
 y^{(0)} &= y_0, \\
 y^{(1)} &= \tilde{y}_1, \\
 y^{(k+1)} &= y^{(k)} + \frac{h}{12} \left[5f(t^{(k+1)}, y(t^{(k+1)})) \right. \\
 &\quad \left. + 8f(t^{(k)}, y(t^{(k)})) \right. \\
 &\quad \left. - f(t^{(k-1)}, y(t^{(k-1)})) \right]
 \end{aligned} \tag{4.189}$$

Observação 4.5.2. (Estimativa do Erro Local.) O método A-B-2 tem erro de truncamento local da $O(h^3)$.

A inicialização do método A-B-2 requer a computação de $y^{(1)}$ por algum método de passo simples. Manter a consistência é um desafio e uma alternativa é a utilização de um esquema preditor-corretor.

Método Preditor-Corretor

Um Método Preditor-Corretor consiste em acoplar um método explícito com um implícito. A cada passo no tempo $t^{(k)}$, o método explícito (**preditor**) é usado para computar uma primeira aproximação $\tilde{y}^{(k)} \approx y(t^{(k)})$ e, o método implícito (**corretor**) é usado para computar $y^{(k)}$, usando $\tilde{y}^{(k)}$ no esquema.

⁴Consulte o Exercício 4.5.6

Exemplo 4.5.3. Consideremos o seguinte PVI

$$y' - y = \sin(t), 0 < t \leq 1, \quad (4.190a)$$

$$y(0) = \frac{1}{2}. \quad (4.190b)$$

Na Tabela 4.9, temos as aproximações $\tilde{y}(1)$ de $y(1)$ computadas pelo Método Preditor-Corretor de Adams de 2 passos⁵. Para a inicialização, usamos o Método do Ponto Médio 4.126, como preditor o Método de Adams-Bashforth de 2 passos (4.184) e como corretor o Método de Adams-Moulton (4.189).

Tabela 4.9: Resultados referentes ao Exemplo 4.5.3.

h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
10^{-1}	2.02638	1.0e-03
10^{-2}	2.02739	1.1e-06
10^{-3}	2.02740	1.2e-09

Código 4.8: pca2.py

```

1 import numpy as np
2
3 def pca2(f, t0, y0, h, n):
4
5     t = np.empty(3)
6     t[0] = t0
7     y = np.empty(3)
8     y[0] = y0
9
10    # inicialização (PM 2)
11    y[1] = y[0] + h/2*f(t[0], y[0])
12    y[1] = y[0] + h*f(t[0]+h/2, y[1])
13    t[1] = t[0] + h
14
15
16    # iterações
17    for k in range(1, n):

```

⁵Com erro de truncamento local de $O(h^2)$.

```

18
19     # preditor (AB 2)
20     y[2] = y[1] + h/2*(3*f(t[1],y[1]) \
21                      - f(t[0], y[0]))
22     t[2] = t[1] + h
23     # corretor (AM 2)
24     y[2] = y[1] + h/12*(5*f(t[2],y[2]) \
25                        + 8*f(t[1], y[1]) \
26                        - f(t[0], y[0]))
27
28     t[:2] = t[1:]
29     y[:2] = y[1:]
30
31     return t[2], y[2]
32
33 def f(t, y):
34     return y + np.sin(t)
35
36 # analítica
37 def exata(t):
38     return np.exp(t) - 0.5*np.sin(t) - 0.5*np.cos(
39         t)
40
41 h = 1e-1
42 n = round(1./h)
43 t,y = pca2(f, 0., 0.5, h, n)
44 print(f'{h:.1e}: {y:.5e} {np.abs(y-exata(1)):.1e}'
45       )

```

Método de Adams-Moulton de 4 Passos

O Método de Adams-Moulton de 4 Passos é um método implícito com erro

de truncamento local de $O(h^5)$. Sua iteração consiste em

$$\begin{aligned}
 y^{(0)} &= y_0, \\
 y^{(1)} &= \tilde{y}_1, \\
 y^{(2)} &= \tilde{y}_2, \\
 y^{(3)} &= \tilde{y}_3, \\
 y^{(k+1)} &= y^{(k)} + \frac{h}{720} \left[251f(t^{(k+1)}, y^{(k+1)}) \right. \\
 &\quad \left. + 646f(t^{(k)}, y^{(k)}) - 264f(t^{(k-1)}, y^{(k-1)}) \right. \\
 &\quad \left. + 106f(t^{(k-2)}, y^{(k-2)}) - 19f(t^{(k-3)}, y^{(k-3)}) \right].
 \end{aligned} \tag{4.191}$$

Exemplo 4.5.4. Consideramos o seguinte PVI

$$y' - y = \sin(t), 0 < t \leq 1, \tag{4.192a}$$

$$y(0) = \frac{1}{2}. \tag{4.192b}$$

Podemos computar uma aproximação para $y(1)$ usando um esquema preditor-corretor com: inicialização pelo Método RK-4 [4.131](#), preditor o Método de Adams-Bashforth de 4 passos ([4.186](#)) e como corretor o Método de Adams-Moulton ([4.191](#)). Isto nos fornece um método com erro de truncamento local mínimo de $O(h^4)$. Consulte o Exercício [4.5.4](#).

4.5.3 Exercícios

E.4.5.1. Considere o seguinte problema de valor inicial

$$y' + e^{-y^2+1} = 2, \quad t > 1, \tag{4.193a}$$

$$y(1) = -1. \tag{4.193b}$$

Inicializando pelo Método de Euler, use os seguintes métodos de passo múltiplo com $h = 0,1$ para computar o valor aproximado de $y(2)$:

- método de Adams-Bashforth de ordem 2.
- método de Adams-Bashforth de ordem 3.

c) método de Adams-Bashforth de ordem 4.

E.4.5.2. (4.109) Considere o PVI

$$y' + \cos(t) = y, \quad 0 < t \leq 1, \quad (4.194a)$$

$$y(0) = \frac{1}{2}. \quad (4.194b)$$

Usando um método de inicialização adequado, aplique os seguintes métodos para computar aproximações para $y(1)$:

a) Método de Adams-Bashforth de 2 Passos.

b) Método de Adams-Bashforth de 4 Passos.

Em cada caso, verifique se seus resultados satisfazem a ordem esperada do erro de truncamento local.

E.4.5.3. Desenvolva o Método de Adams-Bashforth de ordem 3. Para tanto, assuma $m = 3$ em (4.181) para obter as iterações

$$\begin{aligned} y^{(0)} &= y_0, \\ y^{(k+1)} &= y^{(k)} + \frac{h}{12} \left[23f(t^{(k)}, y^{(k)}) \right. \\ &\quad \left. - 16f(t^{(k-1)}, y^{(k-1)}) + 5f(t^{(k-2)}, y^{(k-2)}) \right], \end{aligned} \quad (4.195)$$

Escolha um método adequado para inicializá-lo e implemente-o para computar a solução aproximada de $y(1)$ para o PVI

$$y' - y = \sin(t), 0 < t \leq 1, \quad (4.196a)$$

$$y(0) = \frac{1}{2}. \quad (4.196b)$$

E.4.5.4. Considere o PVI

$$y' - y = \sin(t), 0 < t \leq 1, \quad (4.197a)$$

$$y(0) = \frac{1}{2}. \quad (4.197b)$$

Compute aproximações para $y(1)$ usando um esquema preditor-corretor com: inicialização pelo Método RK-4 4.131, preditor o Método de Adams-Bashforth de 4 passos (4.186) e como corretor o Método de Adams-Moulton (4.191). Verifique que isso nos fornece um método com erro de truncamento local mínimo de $O(h^4)$.

E.4.5.5. O Método de Adams-Moulton de 3 passos (AM-3) é um método implícito com erro de truncamento local de $O(h^4)$. Sua iteração consiste em

$$\begin{aligned} y^{(0)} &= y_0, \\ y^{(1)} &= \tilde{y}_1, \\ y^{(2)} &= \tilde{y}_2, \\ y^{(k+1)} &= y^{(k)} + \frac{h}{24} \left[9f(t^{(k+1)}, y^{(k+1)}) \right. \\ &\quad \left. + 19f(t^{(k)}, y^{(k)}) - 5f(t^{(k-1)}, y^{(k-1)}) \right. \\ &\quad \left. + f(t^{(k-2)}, y^{(k-2)}) \right]. \end{aligned} \quad (4.198)$$

Refaça o Exercício 4.5.4 substituindo o método corretor pelo AM-3. Verifique se suas computações satisfazem o esperado erro de truncamento local.

Análise Numérica

E.4.5.6. Mostre o desenvolvimento do Método de Adams-Moulton de 2 passos (4.189).

4.6 Método adaptativo com controle de erro

[[tag:revisar]]

Consideremos um problema de valor inicial

$$y'(t) = f(t, y(t)), \quad t > t_0, \quad (4.199)$$

$$y(t_0) = y_0. \quad (4.200)$$

e um método de passo simples

$$y^{(1)} = y_0, \quad (4.201)$$

$$y^{(i+1)}(h^{(i+1)}) = y^{(i)} + h^{(i+1)}\Phi(t^{(i)}, y^{(i)}; h^{(i+1)}), \quad (4.202)$$

com $t^{(i)} = t_0 + (i-1)h^{(i)}$. Nesta seção, discutiremos uma estimativa para o maior valor de $h^{(i+1)}$ tal que o erro de discretização global $e(t^{(i+1)}; h^{(i+1)})$ seja controlado por uma dada tolerância TOL , i.e.

$$|e(t^{(i+1)}; h^{(i+1)})| := |y^{(i+1)}(h^{(i+1)}) - y(t^{(i+1)})| \approx TOL. \quad (4.203)$$

Para um método de ordem h^p , pode-se mostrar que (veja, [3, Cap. 7, Seç. 7.2])

$$y^{(i+1)}(h^{(i+1)}) = y(t^{(i+1)}) + e_p(t^{(i+1)})(h^{(i+1)})^p, \quad (4.204)$$

onde $e(t^{(i+1)})$ é uma função apropriada. Então, assumindo que $e(t^{(i)}; h^{(i)}) = 0$, temos

$$e_p(t^{(i+1)}) = h^{(i+1)}e'_p(t^{(i)}) \quad (4.205)$$

e, portanto, para termos (4.203) impomos que

$$|(h^{(i+1)})^{p+1}e'_p(t^{(i)})| = TOL. \quad (4.206)$$

Daí, se obtermos uma aproximação para $e'_p(t^{(i)})$ teremos uma aproximação para o passo $h^{(i+1)}$.

Para estimarmos $e_p(t^{(i+1)})$, observamos que de (4.204) temos

$$y^{(i+1)}\left(\frac{h^{(i+1)}}{2}\right) = y(t^{(i+1)}) + e_p(t^{(i+1)})\frac{(h^{(i+1)})^p}{2^p} \quad (4.207)$$

e, então, subtraindo esta de (4.204) temos

$$y^{(i+1)}(h^{(i+1)}) - y^{(i+1)}\left(\frac{h^{(i+1)}}{2}\right) = e_p(t^{(i+1)})\left(\frac{h^{(i+1)}}{2}\right)^p (2^p - 1), \quad (4.208)$$

donde

$$e_p(t^{(i+1)})\left(\frac{h^{(i+1)}}{2}\right)^p = \frac{y^{(i+1)}(h^{(i+1)}) - y^{(i+1)}\left(\frac{h^{(i+1)}}{2}\right)}{2^p - 1}. \quad (4.209)$$

Daí, de (4.205), obtemos

$$e'_p(t^{(i)})h^{(i+1)}\left(\frac{h^{(i+1)}}{2}\right)^p = \frac{y^{(i+1)}(h^{(i+1)}) - y^{(i+1)}\left(\frac{h^{(i+1)}}{2}\right)}{2^p - 1}, \quad (4.210)$$

o que nos fornece a seguinte aproximação de $e'_p(t^{(i)})$

$$e'_p(t^{(i)}) = \frac{1}{(h^{(i+1)})^{p+1}} \frac{2^p}{2^p - 1} \left[y^{(i+1)}(h^{(i+1)}) - y^{(i+1)}\left(\frac{h^{(i+1)}}{2}\right) \right]. \quad (4.211)$$

Assim sendo, de (4.206) temos que o passo $h^{(i+1)}$ apropriado é tal que

$$\frac{2^p}{2^p - 1} \left| y^{(i+1)}(h^{(i+1)}) - y^{(i+1)}\left(\frac{h^{(i+1)}}{2}\right) \right| \approx TOL. \quad (4.212)$$

Com base nesta estimativa podemos propor o seguinte método de passo adaptativo. Partindo de uma escolha arbitrária de h , computamos $y^{(i+1)}(h)$ e $y^{(i+1)}(h/2)$ de $y^{(i)}$. Então, enquanto

$$\frac{2^p}{2^p - 1} \left| y^{(i+1)}(h) - y^{(i+1)}\left(\frac{h}{2}\right) \right| > TOL, \quad (4.213)$$

tomamos sucessivas divisões de h por 2, até satisfazermos (4.212). Obtido o h que satisfaz (4.212), temos computado $y^{(i+1)}$ com $h^{(i+1)} = h$.

Exemplo 4.6.1. Consideremos o seguinte problema de valor inicial

$$y' - y = \sin(t), t > 0 \quad (4.214)$$

$$y(0) = \frac{1}{2}. \quad (4.215)$$

A Figura 4.4 mostra a comparação entre $y(t)$ e a solução numérica obtida da aplicação do Método de Euler com passo adaptativo. No método, utilizamos o passo inicial $h^{(1)} = 0,1$ e tolerância $TOL = 10^{-4}$. Ao compararmos esta figura com a Figura (4.1) fica evidente o controle do erro.

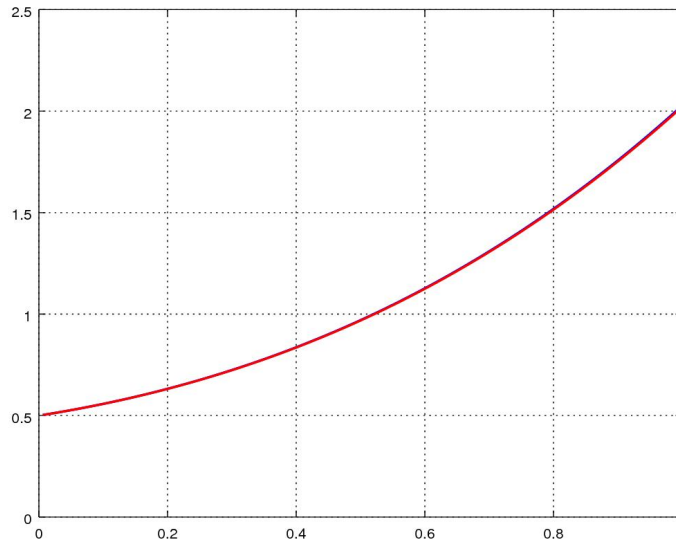


Figura 4.4: Resultados referentes ao Exemplo 4.6.1.

4.6.1 Exercícios

[[tag:revisar]]

E.4.6.1. Considere o seguinte problema de valor inicial

$$y' + e^{-y^2+1} = 2, \quad t > 1, \quad (4.216)$$

$$y(1) = -1. \quad (4.217)$$

Use o Método de Euler com passo adaptativo para computar o valor aproximado de $y(2)$. Para tanto, utilize o passo inicial $h = 0,1$ e a tolerância de $TOL = 10^{-4}$.

Capítulo 5

Problema de Valor de Contorno

Neste capítulo, estudamos métodos numéricos para resolver **Problemas de Valores de Contorno** da forma

$$u'' = f(x, u, u'), \quad a < x < b, \quad (5.1a)$$

$$\eta_1 u'(a) + \theta_1 u(b) = g_1 \quad (5.1b)$$

$$\eta_2 u'(b) + \theta_2 u(b) = g_2 \quad (5.1c)$$

onde a incógnita é $u = u(x)$ com dada $f = f(x, u, u')$ e dados parâmetros η_1, θ_1 (não simultaneamente nulos), η_2, θ_2 (não simultaneamente nulos), g_1 e g_2 .

5.1 Método de Diferenças Finitas

Consideramos o seguinte problema linear de valor de contorno (PVC)

$$u'' + \alpha(x)u' + \beta(x)u = f(x), \quad a < x < b, \quad (5.2a)$$

$$u(a) = g, \quad (5.2b)$$

$$u(b) = h. \quad (5.2c)$$

onde a incógnita é $u = u(x)$ com dada fonte $f = f(x)$ e dados parâmetros g e h .

A aproximação pelo **Método de Diferenças Finitas** (MDF) de (5.2a)-(5.2c) surge da substituição das derivadas por Fórmulas de Diferenças Finitas.

De forma geral, o método pode ser dividido em três etapas: 1. discretização do domínio, 2. discretização das equações, 3. resolução do problema discreto.

1. Discretização do Domínio.

A discretização do domínio é seu particionamento em subintervalos (células computacionais) e pontos (nodos computacionais). Por simplicidade, vamos considerar apenas o caso de um particionamento uniforme. Particionamos o domínio $D = [a, b]$ em n de subintervalos de tamanho de malha

$$h = \frac{b - a}{n}, \quad (5.3)$$

e os nodos da partição podem ser indexados da seguinte forma

$$x_i = a + (i - 1)h, \quad (5.4)$$

com $i = 1, 2, 3, \dots, n + 1$.

2. Discretização das Equações.

Começando por (5.2a), em um nodo $x = x_i, i = 2, 3, \dots, n$, temos

$$u''(x_i) + \alpha(x_i)u'(x_i) + \beta(x_i)u(x_i) = f(x_i). \quad (5.5)$$

Podemos substituir a segunda derivada de u pela fórmula de diferenças finitas central de ordem h^2

$$u''(x_i) = \underbrace{\frac{u(x_i - h) - 2u(x_i) + u(x_i + h))}{h^2}}_{D_{0,h^2}^2 u(x_i)} + O(h^2). \quad (5.6)$$

A primeira derivada de u também pode ser substituída pela fórmula de diferenças finitas central de ordem h^2

$$u'(x_i) = \underbrace{\frac{u(x_i + h) - u(x_i - h))}{2h}}_{D_{0,h^2} u(x_i)} + O(h^2). \quad (5.7)$$

Agora, denotando $u_i \approx u(x_i)$, temos $u_{i-1} \approx u(x_i - h)$ e $u_{i+1} \approx u(x_i + h)$. Substituindo as derivadas pelas fórmulas de diferenças finitas, temos de (5.5)

que

$$\begin{aligned} \left(\frac{u_{i-1} - 2u_i + u_{i+1}}{h^2} \right) + \alpha(x_i) \left(\frac{u_{i+1} - u_{i-1}}{2h} \right) \\ + \beta(x_i)u_i + O(h^2) = f(x_i), \end{aligned} \quad (5.8)$$

Rearranjando os termos e desconsiderando o termo do erro de truncamento, obtemos o seguinte sistema de equações lineares

$$\begin{aligned} \left(\frac{1}{h^2} - \frac{\alpha_i}{2h} \right) u_{i-1} + \left(\beta_i - \frac{2}{h^2} \right) u_i \\ + \left(\frac{1}{h^2} + \frac{\alpha_i}{2h} \right) u_{i+1} = f_i, \end{aligned} \quad (5.9)$$

onde, usamos a notação $\alpha_i = \alpha(x_i)$, $\beta_i = \beta(x_i)$ e $f_i = f(x_i)$.

Observamos que este sistema consiste em $n - 1$ equações envolvendo as $n + 1$ incógnitas u_i , $i = 1, 2, \dots, n + 1$. Para fechá-lo, usamos as condições de contorno. De (5.2b), temos

$$u_1 = g \quad (5.10)$$

e de (5.2c) temos

$$u_{n+1} = h, \quad (5.11)$$

lembrando que $u_0 \approx u(x_0)$ e $u_n \approx u(x_n)$.

Por fim, as equações (5.9)-(5.11) formam o seguinte **problema discretizado**

$$u_1 = g, \quad (5.12a)$$

$$\begin{aligned} \left(\frac{1}{h^2} - \frac{\alpha_i}{2h} \right) u_{i-1} + \left(\beta_i - \frac{2}{h^2} \right) u_i \\ + \left(\frac{1}{h^2} + \frac{\alpha_i}{2h} \right) u_{i+1} = f_i, \end{aligned} \quad (5.12b)$$

$$u_{n+1} = h, \quad (5.12c)$$

para $i = 2, 3, \dots, n$.

3. Resolução do Problema Discreto.

O problema discreto (5.12) consiste em um sistema linear de $n + 1$ equações com $n + 1$ incógnitas. Na forma matricial temos

$$A\mathbf{u} = \mathbf{b} \quad (5.13)$$

onde $\mathbf{u} = (u_1, u_2, \dots, u_{n+1})$ é o vetor das incógnitas, $\mathbf{b} = (g, f_2, f_3, \dots, f_n, h)$. A matriz dos coeficientes é $A = [a_{i,j}]_{i,j=1}^{n+1,n+1}$ e seus elementos não nulos são

$$\begin{aligned} a_{1,1} &= 1, \\ a_{i,i-1} &= \frac{1}{h^2} - \frac{\alpha_i}{2h}, \end{aligned} \quad (5.14)$$

$$a_{i,i} = \beta_i - \frac{2}{h^2}, \quad (5.15)$$

$$a_{i,i+1} = \frac{1}{h^2} + \frac{\alpha_i}{2h}, \quad (5.16)$$

$$a_{n+1,n+1} = 1, \quad (5.17)$$

para $i = 2, 3, \dots, n$.

A resolução do problema discreto se resume, a resolver o sistema $A\mathbf{u} = \mathbf{b}$, o que pode ser feito por qualquer método numérico apropriado.

Exemplo 5.1.1. Consideramos o seguinte PVC

$$-u'' = \pi^2 \sin(\pi x), \quad 0 < x < 1, \quad (5.18)$$

$$u(0) = 0, \quad (5.19)$$

$$u(1) = 0. \quad (5.20)$$

A solução analítica deste problema é $u(x) = \sin(\pi x)$. Usando o MDF como acima, encontramos o problema discreto

$$u_1 = 0, \quad (5.21a)$$

$$-\frac{1}{h^2}u_{i-1} + \frac{2}{h^2}u_i - \frac{1}{h^2}u_{i+1} = \pi^2 \sin(\pi x_i), \quad (5.21b)$$

$$u_{n+1} = 0, \quad (5.21c)$$

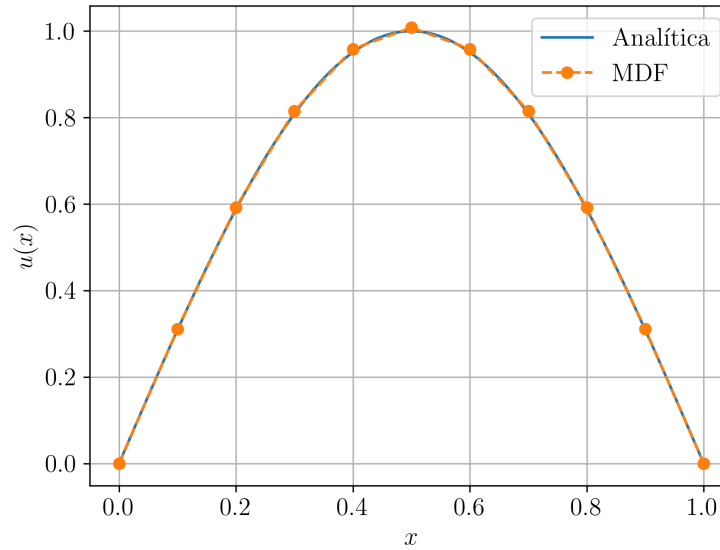


Figura 5.1: Resultado referente ao Exemplo 5.1.1.

Tabela 5.1: Resultados referentes ao Exemplo 5.1.1.

h	$\ \tilde{u} - u\ _{L^2}$
1.0e-1	1.8e-2
5.0e-2	6.5e-3
2.5e-2	2.3e-3
1.0e-3	5.8e-4

com tamanho de malha $h = 1/n$ e nodos $x_i = (i - 1)h$ indexados por $i = 1, 2, \dots, n + 1$.

Resolvendo este sistema com $h = 10^{-1}$ obtemos a solução numérica apresentada na Figura 5.1. Ainda, na Tabela 5.1 temos a comparação na norma L^2 da solução numérica \tilde{u} com a solução analítica $u = (u(x_i))_{i=1}^{n+1}$ para diferentes escolhas de h .

Código 5.1: pvc_mdf.py

```
1 import numpy as np
```

```
2
3 # malha
4 n = 10
5 h = 1./n
6 xx = np.linspace(0., 1., n+1)
7
8 # fonte
9 def f(x):
10     return np.pi**2*np.sin(np.pi*x)
11
12 # prob discreto
13 A = np.zeros((n+1, n+1))
14 b = np.empty(n+1)
15
16 # c.c. x = 0.
17 A[0,0] = 1.
18 b[0] = 0.
19
20 # pts internos
21 for i in range(1,n):
22     A[i,i-1] = -1./h**2
23     A[i,i] = 2./h**2
24     A[i,i+1] = -1./h**2
25     b[i] = f(xx[i])
26
27 # c.c. x = 1.
28 A[n,n] = 1.
29 b[n] = 0.
30
31 # resol
32 u = npla.solve(A, b)
```

5.1.1 Exercícios

E.5.1.1. Considere o PVC

$$-u'' = \pi^2 \cos(\pi x), \quad 0 < x < 1, \quad (5.22)$$

$$u(0) = 1, \quad (5.23)$$

$$u(1) = -1. \quad (5.24)$$

A solução analítica deste problema é $u(x) = \cos(\pi x)$. Use o MDF para computar aproximações numéricas $\tilde{\mathbf{u}}_h$ com tamanhos de malha $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ e verifique o erro absoluto $\varepsilon_{\text{abs}} := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|$.

E.5.1.2. Considere o PVC

$$-u'' = 2, \quad -1 < x < 1, \quad (5.25)$$

$$u(-1) = 0, \quad (5.26)$$

$$u(1) = 0. \quad (5.27)$$

A solução analítica deste problema é $u(x) = 1 - x^2$. Use o MDF com $n = 20$ subintervalos na malha e verifique o erro absoluto $\varepsilon_{\text{abs}} := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|$. Por que o erro está próximo precisão de máquina? Justifique sua resposta.

E.5.1.3. Considere o seguinte PVC

$$-u'' + u' = f(x), \quad -1 < x < 1, \quad (5.28a)$$

$$u(-1) = 0, \quad (5.28b)$$

$$u'(1) = 0, \quad (5.28c)$$

onde

$$f(x) = \begin{cases} 1 & , x \leq 0 \\ 0 & , x > 0 \end{cases} \quad (5.29)$$

Use uma aproximação adequada pelo método de diferenças finitas para obter o valor aproximado de $u(0)$ com precisão de 2 dígitos significativos.

E.5.1.4. Considere o PVC

$$-u'' = \pi^2 \cos(\pi x), \quad 0 < x < 1, \quad (5.30)$$

$$u(0) = 1, \quad (5.31)$$

$$u'(1) = 0. \quad (5.32)$$

A solução analítica deste problema é $u(x) = \cos(\pi x)$. Aplique o MDF para computar aproximações numéricas usando a:

a) fórmula de diferenças finitas $D_{-,h}u(x)$ no contorno $x = 1$.

b) fórmula de diferenças finitas $D_{-,h^2}u(x)$ no contorno $x = 1$.

Quais das duas produz o resultado mais preciso? Justifique sua resposta.

5.2 Método de Elementos Finitos

Consideramos o seguinte problema linear de valor de contorno (PVC)

$$-u'' = f(x), \quad a < x < b, \quad (5.33a)$$

$$u(a) = 0, \quad (5.33b)$$

$$u(b) = 0. \quad (5.33c)$$

onde a incógnita é $u = u(x)$ com dada fonte $f = f(x)$.

A solução pelo **Método de Elementos Finitos** (FEM) de (5.2a)-(5.2c) surge da aproximação do problema em um espaço de dimensão finita de funções. São três passos fundamentais: 1. escrever a formulação fraca do problema¹, 2. escrever a formulação de elementos finitos e 3. resolver o problema de elementos finitos.

1. Formulação Fraca

Para obter a **formulação fraca** do PVC (5.3)-(5.33c), multiplicamos (5.3) por uma arbitrária função teste $v = v(x)$

$$-u''v = fv \quad (5.34)$$

e integramos no domínio $a \leq x \leq b$, i.e.

$$-\int_a^b u''v \, dx = \int_a^b fv \, dx. \quad (5.35)$$

Então, aplicando **integração por partes** no primeiro termo do lado esquerdo, obtemos

$$\int_a^b u'v' \, dx - [u'v]_{x=a}^b = \int_a^b fv \, dx. \quad (5.36)$$

¹Por convenção, (5.2a)-(5.2c) é chamado de formulação forte do problema.

Vamos denotar o produto interno em $L^2([a, b])$ ² por

$$(u, v)_2 := \int_a^b uv \, dx \quad (5.37)$$

e nos contornos

$$\langle u, v \rangle := u(b)v(b) - u(a)v(a). \quad (5.38)$$

Com isso, definimos a **formulação fraca** como o seguinte problema: **encontrar** $u \in V := H_0^1([a, b])$ ³ tal que

$$a(u, v) = l(v), \quad \forall v \in V, \quad (5.39)$$

onde a **forma bilinear** é

$$a(u, v) := (u', v')_2 \quad (5.40)$$

e a **forma linear** é

$$l(v) := (f, v)_2. \quad (5.41)$$

2. Formulação de Elementos Finitos

A **formulação de elementos finitos** do problema (5.3)-(5.33c) é obtida a partir de (5.39) pela substituição do espaço de funções V por um **espaço de dimensão finita** V_h . A ideia é que $V_h \rightarrow V$, bem como a solução de elementos finitos $u_h \rightarrow u \in V$ quando $h \rightarrow 0$.

Para construir o **espaço de elementos finitos** V_h , vamos considerar elementos do tipo

$$P_1(I) := \{v = v(x); v(x) = c_0 + c_1x, \\ x \in I, c_0, c_1 \in \mathbb{R}\}, \quad (5.42)$$

onde I é um intervalo fechado.

Sobre o domínio, assumimos uma malha uniforme

$$M([a, b]) := \{x_1, x_2, \dots, x_{n+1}\} \quad (5.43)$$

² $u \in L^2([a, b]) \Leftrightarrow \int_a^b |u|^2 \, dx < \infty$.

³ $H_0^1([a, b]) := \{u = u(x); u, u' \in L^2([a, b]), u(a) = u(b) = 0\}$.

com $h = (b - a)/n$, $x_i = a + (i - 1)h$, $i = 1, 2, \dots, n + 1$. Nesta, definimos o espaço de funções

$$V_{h,0} := \left\{ v = v(x); v \in C^0[a, b], v(a) = v(b) = 0, \right. \\ \left. v|_{[x_i, x_{i+1}]} \in P_1([x_i, x_{i+1}]), i = 1, 2, \dots, n \right\}. \quad (5.44)$$

Pode-se mostrar que $V_h = \text{span}\{\phi_i\}_{i=1}^{n-1}$, com base nodal

$$\phi_j(x_i) = \begin{cases} 1 & , i = j, \\ 0 & , i \neq j \end{cases} \quad (5.45)$$

para $i, j = 2, \dots, n$ e $\phi_1(a) = 0 = \phi_n(b)$. Podemos verificar que

$$\phi_i(x) = \begin{cases} (x - x_{i-1})/h & , x \in [x_{i-1}, x_i], \\ (x_{i+1} - x)/h & , x \in [x_i, x_{i+1}], \\ 0 & , \text{noutros casos} \end{cases} \quad (5.46)$$

Com isso, definimos a **formulação de elementos finitos** sendo o seguinte problema: encontrar $u_h \in V_{h,0}$ tal que

$$a(u_h, v_h) = l(v_h), \quad \forall v_h \in V_h. \quad (5.47)$$

Tendo em vista que $V_h = \text{span}\{\phi_i\}_{i=1}^{n-1}$, este é equivalente a

$$a(u_h, \phi_j) = l(\phi_j), \quad \forall 1 \leq j \leq n - 1. \quad (5.48)$$

3. Resolução do Problema de Elementos Finitos

O problema de elementos finitos (5.48) consiste em um sistema linear $Au = b$. De fato, a solução $u_h \in V_{h,0}$ pode ser escrita como a seguinte combinação linear

$$u_h = \sum_{j=1}^{n-1} u_j \phi_j. \quad (5.49)$$

Logo, temos que

$$a(u_h, \phi_i) = \left(\sum_{j=1}^{n-1} u_j \phi_j, \phi_i \right)_2, \quad (5.50a)$$

$$= \sum_{j=1}^{n-1} u_j (\phi_j, \phi_i)_2, \quad (5.50b)$$

$$= A\mathbf{u}, \quad (5.50c)$$

onde a **matriz dos coeficientes** é $A = [a_{i,j} := (\phi_j, \phi_i)]_{i,j=1}^{n-1}$ e o **vetor das incógnitas** é $\mathbf{u} = (u_j)_{j=1}^{n-1}$. Doutro lado, temos

$$l(\phi_i) = (f, \phi_i)_2, \quad (5.51)$$

o que nos fornece o **vetor dos termos constantes** $\mathbf{b} = (b_i := (f, \phi_i)_2)_{i=1}^{n-1}$.

O cálculo dos elementos de A fornece

$$a_{i,i} = (\phi'_i, \phi'_i)_2 \quad (5.52a)$$

$$= \int_a^b (\phi'_i)^2 dx \quad (5.52b)$$

$$= \int_{x_{i-1}}^{x_{i+1}} (\phi'_i)^2 dx \quad (5.52c)$$

$$= \int_{x_{i-1}}^{x_i} \left[\left(\frac{x - x_{i-1}}{h} \right)' \right]^2 dx \quad (5.52d)$$

$$+ \int_{x_i}^{x_{i+1}} \left[\left(\frac{x_{i+1} - x}{h} \right)' \right]^2 dx \quad (5.52e)$$

$$= \frac{2}{h}, \quad i = 1, 2, \dots, n-1, \quad (5.52f)$$

$$a_{i,i+1} = (\phi'_{i+1}, \phi'_i)_2 \quad (5.53a)$$

$$= \int_a^b \phi'_{i+1} \phi'_i dx \quad (5.53b)$$

$$= \int_{x_i}^{x_{i+1}} \left(\frac{x_{i+1} - x}{h} \right)' \left(\frac{x - x_{i+1}}{h} \right)' dx \quad (5.53c)$$

$$= -\frac{1}{h}, \quad i = 1, 2, \dots, n-2, \quad (5.53d)$$

$$a_{i-1,i} = (\phi'_{i-1}, \phi'_i)_2 \quad (5.54a)$$

$$= \int_a^b \phi_{i-1} \phi_i dx \quad (5.54b)$$

$$= \int_{x_{i-1}}^{x_i} \left(\frac{x_i - x}{h} \right)' \left(\frac{x - x_i}{h} \right)' dx \quad (5.54c)$$

$$= -\frac{1}{h}, \quad i = 2, \dots, n-1, \quad (5.54d)$$

observando que, noutros casos, $a_{i,j} = 0$.

Um cálculo aproximado dos elementos de \mathbf{b} fornece⁴

$$b_i = (f, \phi_i)_2 \quad (5.55a)$$

$$= \int_a^b f(x) \phi_i(x) dx \quad (5.55b)$$

$$= \int_{x_{i-1}}^{x_i} f(x) \frac{(x - x_{i-1})}{h} dx \quad (5.55c)$$

$$+ \int_{x_i}^{x_{i+1}} f(x) \frac{(x_{i+1} - x)}{h} dx \quad (5.55d)$$

$$\approx \frac{h}{2} f(x_{i-1/2}) + \frac{h}{2} f(x_{i+1/2}). \quad (5.55e)$$

Exemplo 5.2.1. Consideramos o seguinte PVC

$$-u'' = \pi^2 \sin(\pi x), \quad 0 < x < 1, \quad (5.56)$$

$$u(0) = 0, \quad (5.57)$$

$$u(1) = 0. \quad (5.58)$$

A solução analítica deste problema é $u(x) = \sin(\pi x)$.

Resolvendo este sistema com $h = 10^{-1}$ obtemos a solução numérica apresentada na Figura 5.2.

Código 5.2: pvc_mef.py

```
1 import numpy as np
2
3 # malha
4 n = 10
```

⁴Por simplicidade, usando a regra do ponto médio para aproximar as integrais.

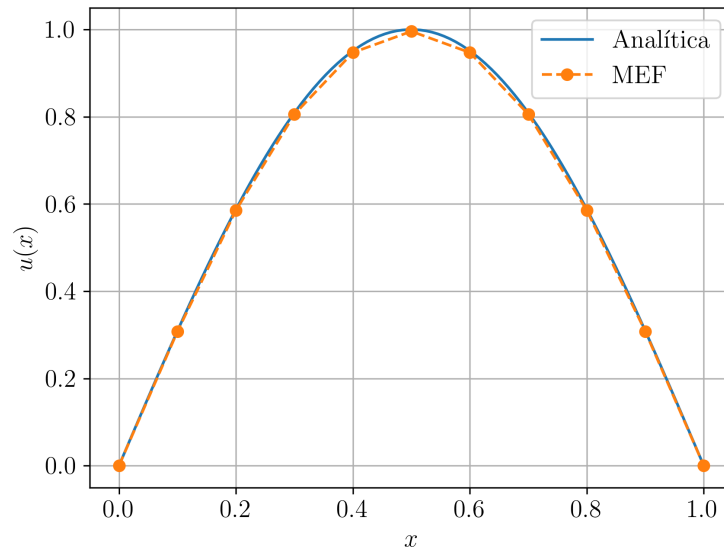


Figura 5.2: Resultado referente ao Exemplo 5.2.1.

```
5 h = 1./n
6 xx = np.linspace(0., 1., n+1)
7
8 # fonte
9 def f(x):
10     return np.pi**2*np.sin(np.pi*x)
11
12 # prob discreto
13 A = np.zeros((n-1, n-1))
14 b = np.empty(n-1)
15
16 # c.c. x = 0.
17 A[0,0] = 2./h
18 A[0,1] = -1./h
19 b[0] = h/2 * (f(xx[1]-0.5*h) + f(xx[1]+0.5*h))
20
21 # pts internos
22 for i in range(1,n-2):
```

```

23     A[i,i-1] = -1./h
24     A[i,i] = 2./h
25     A[i,i+1] = -1./h
26     b[i] = h/2 * (f(xx[i+1]-0.5*h) + f(xx[i
+1]+0.5*h))
27
28 # c.c. x = 1.
29 A[n-2,n-3] = -1./h
30 A[n-2,n-2] = 2./h
31 b[n-2] = h/2 * (f(xx[n-1]-0.5*h) + f(xx[n-1]+0.5*h
))
32
33 # resol
34 u = npla.solve(A, b)
35 ## c.c. (dirichlet)
36 u = np.concatenate(([0.],u,[0.]))

```

5.2.1 Exercícios

E.5.2.1. Considere o PVC

$$-u'' = \pi^2 \cos(\pi x), \quad 0 < x < 1, \quad (5.59)$$

$$u(0) = 1, \quad (5.60)$$

$$u(1) = -1. \quad (5.61)$$

A solução analítica deste problema é $u(x) = \cos(\pi x)$. Use o MEF para computar aproximações numéricas $\tilde{\mathbf{u}}_h$ com tamanhos de malha $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ e verifique o erro absoluto $\varepsilon_{\text{abs}} := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|$.

E.5.2.2. Considere o PVC

$$-u'' = 2, \quad -1 < x < 1, \quad (5.62)$$

$$u(-1) = 0, \quad (5.63)$$

$$u(1) = 0. \quad (5.64)$$

A solução analítica deste problema é $u(x) = 1 - x^2$. Use o MEF com $n = 20$ subintervalos na malha e verifique o erro absoluto $\varepsilon_{\text{abs}} := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|$. Por que o erro está próximo precisão de máquina? Justifique sua resposta.

E.5.2.3. Considere o seguinte PVC

$$-u'' + u' = f(x), \quad -1 < x < 1, \quad (5.65a)$$

$$u(-1) = 0, \quad (5.65b)$$

$$u'(1) = 0, \quad (5.65c)$$

onde

$$f(x) = \begin{cases} 1 & , x \leq 0 \\ 0 & , x > 0 \end{cases} \quad (5.66)$$

Use uma aproximação adequada pelo MEF para obter o valor aproximado de $u(0)$ com precisão de 2 dígitos significativos.

E.5.2.4. Considere o PVC

$$-u'' = \pi^2 \cos(\pi x), \quad 0 < x < 1, \quad (5.67)$$

$$u(0) = 1, \quad (5.68)$$

$$u'(1) = 0. \quad (5.69)$$

A solução analítica deste problema é $u(x) = \cos(\pi x)$. Aplique o MEF para computar uma aproximação numérica com erro absoluto de no máximo 10^{-3} na norma L^2 .

5.3 Método de Volumes Finitos

O Método de Volumes Finitos (MVF) é um método de discretização apropriado para problemas conservativos. Consideramos o seguinte problema linear de valor de contorno (PVC)

$$-u_{xx} = f(x), \quad a < x < b, \quad (5.70a)$$

$$u(a) = 0, \quad (5.70b)$$

$$u(b) = 0. \quad (5.70c)$$

onde a incógnita é $u = u(x)$ com dada fonte $f = f(x)$. A Eq. (5.70) pode ser reescrita na forma conservativa

$$\text{div}(\mathbf{F}) = f, \quad (5.71)$$

onde $\mathbf{F} = -u_x$

1. Discretização Espacial.

Assumimos uma malha do domínio $[a, b]$ da forma

$$a = x_{\frac{1}{2}} < x_1 < x_{\frac{3}{2}} < \cdots < x_{i-\frac{1}{2}} < x_i < x_{i+\frac{1}{2}} < \cdots < x_n < x_{n+\frac{1}{2}} = b, \quad (5.72)$$

onde $h = (b - a)/n$, $x_{i-\frac{1}{2}} = a + (i - 1)h$, $h^- = h^+ = h/2$, $i = 1, 2, \dots, n$. Também denotamos $K_i = (x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ a i -ésima célula da malha.

2. Discretização das Equações.

No MVF, as incógnitas u_i , $i = 1, 2, \dots, n$, são as aproximações para o valor médio de u nas células K_i , i.e.

$$u_i = \frac{1}{|K_i|} \int_a^b u(x) dx. \quad (5.73)$$

O problema discreto para u_i é obtido tomando a média da Eq. na célula K_i , donde temos

$$-\frac{1}{h} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} u_{xx} dx = \frac{1}{h} \int_{K_i} f dx, \quad (5.74a)$$

$$\frac{1}{h} \left[-u_x \left(x_{i+\frac{1}{2}} \right) + u_x \left(x_{i-\frac{1}{2}} \right) \right] = \frac{1}{h} \int_{K_i} f dx \quad (5.74b)$$

Por fórmula de diferenças finitas central, temos

$$u_x \left(x_{i+\frac{1}{2}} \right) = \frac{u_i - u_{i-1}}{h} + O(h) \quad (5.75)$$

e

$$u_x \left(x_{i-\frac{1}{2}} \right) = \frac{u_{i+1} - u_i}{h} + O(h) \quad (5.76)$$

Com isso, obtemos as equações

$$\frac{1}{h} \left(-\frac{u_{i+1} - u_i}{h} + \frac{u_i - u_{i-1}}{h} \right) = \frac{1}{h} \int_{K_i} f dx, \quad (5.77)$$

Rearranjando os termos e aproximando a integral de f pela regra do ponto médio, obtemos

$$-\frac{1}{h^2} u_{i-1} + \frac{2}{h^2} u_i - \frac{1}{h^2} u_{i+1} = f_i, \quad (5.78)$$

onde $f_i := f(x_i)$ e $i = 2, 3, \dots, n-1$.

Na célula K_1 , tomamos a aproximação

$$u_x \left(x_{\frac{1}{2}} \right) = \frac{u_1 - u_{\frac{1}{2}}}{h/2} + O(h), \quad (5.79a)$$

$$== \frac{u_1}{h/2} + O(h). \quad (5.79b)$$

Aplicando na Eq. (5.74b), obtemos

$$\frac{1}{h} \left(-\frac{u_2 - u_1}{h} + \frac{u_1}{h/2} \right) = \frac{1}{h} \int_{K_1} f \, dx, \quad (5.80)$$

Analogamente, integrando na célula K_n de fronteira, obtemos

$$\frac{1}{h} \left(\frac{u_n}{h/2} + \frac{u_n - u_{n-1}}{h} \right) = \frac{1}{h} \int_{K_n} f \, dx. \quad (5.81)$$

Por fim, obtemos o **problema discreto**

$$\frac{3}{h^2} u_1 - \frac{1}{h^2} u_2 = f_1, \quad (5.82a)$$

$$-\frac{1}{h^2} u_{i-1} + \frac{2}{h^2} u_i - \frac{1}{h^2} u_{i+1} = f_i, \quad (5.82b)$$

$$-\frac{1}{h^2} u_{n-1} + \frac{3}{h^2} u_n = f_n, \quad (5.82c)$$

para $i = 2, 3, \dots, n-1$.

3. Resolução do Problema Discreto.

A resolução do problema discreto se resume a computar a solução do sistema linear (5.82). Sua forma matricial é $A\mathbf{u} = \mathbf{b}$, onde a matriz de coeficientes $A = [a_{i,j}]_{i,j=1}^{n,n}$ tem elementos da diagonal são

$$a_{1,1} = \frac{3}{h^2}, \quad (5.83a)$$

$$a_{i,i} = \frac{2}{h^2}, \quad 2 \leq i \leq n-1, \quad (5.83b)$$

$$a_{n,n} = \frac{3}{h^2}, \quad (5.83c)$$

e os demais $a_{i,j}$ para $i \neq j$

$$a_{i,j} = \begin{cases} -\frac{1}{h^2} & , j = i - 1 \text{ ou } j = i + 1, \\ 0 & , \text{noutros casos} \end{cases} \quad (5.84)$$

O vetor dos termos constantes é $\mathbf{b} = (b_i = f_i)_{i=1}^n$, $f_i = f(x_i)$ e o vetor das incógnitas é $u = (u_i)_{i=1}^n$, sendo u_i a aproximação do valor médio de u na célula K_i , $i = 1, 2, \dots, n$.

Exemplo 5.3.1. Consideramos o seguinte PVC

$$-u'' = \pi^2 \sin(\pi x), \quad 0 < x < 1, \quad (5.85)$$

$$u(0) = 0, \quad (5.86)$$

$$u(1) = 0. \quad (5.87)$$

A solução analítica deste problema é $u(x) = \sin(\pi x)$.

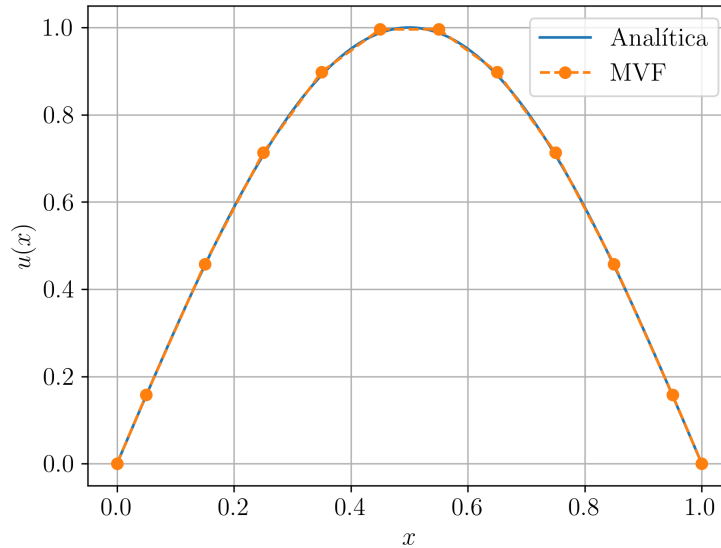


Figura 5.3: Resultado referente ao Exemplo 5.3.1.

Resolvendo este sistema com $h = 10^{-1}$ obtemos a solução numérica apresentada na Figura 5.3.

Código 5.3: pvc_mvf.py

```
1 import numpy as np
2
3 # fonte
4 def f(x):
5     return np.pi**2*np.sin(np.pi*x)
6
7 # malha
8 n = 10
9 h = 1./n
10 xx = np.linspace(h/2, 1.-h/2, n)
11
12 # prob. discreto
13 A = np.zeros((n,n))
14 b = np.empty(n)
15
16 # c.c. x = 0
17 A[0,0] = 3./h**2
18 A[0,1] = -1./h**2
19 b[0] = f(xx[0])
20
21 # pts internos
22 for i in range(1,n-1):
23     A[i,i-1] = -1./h**2
24     A[i,i] = 2./h**2
25     A[i,i+1] = -1./h**2
26     b[i] = f(xx[i])
27
28 # c.c. x = 1
29 A[n-1,n-2] = -1./h**2
30 A[n-1,n-1] = 3./h**2
31 b[n-1] = f(xx[n-1])
32
33 # resol prob disc
34 u = npla.solve(A, b)
35
36 xx = np.concatenate(([0.],xx,[1.])))
```

```
37 u = np.concatenate(([0.],u,[0.]))
```

5.3.1 Exercícios

E.5.3.1. Considere o PVC

$$-u'' = \pi^2 \cos(\pi x), \quad 0 < x < 1, \quad (5.88)$$

$$u(0) = 1, \quad (5.89)$$

$$u(1) = -1. \quad (5.90)$$

A solução analítica deste problema é $u(x) = \cos(\pi x)$. Use o MVF para computar aproximações numéricas $\tilde{\mathbf{u}}_h$ com tamanhos de malha $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$ e verifique o erro absoluto $\varepsilon_{h,\text{abs}} := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|$.

E.5.3.2. Considere o PVC

$$-u'' = 2, \quad -1 < x < 1, \quad (5.91)$$

$$u(-1) = 0, \quad (5.92)$$

$$u(1) = 0. \quad (5.93)$$

A solução analítica deste problema é $u(x) = 1 - x^2$. Use o MVF com $n = 20$ subintervalos na malha e verifique o erro absoluto $\varepsilon_{\text{abs}} := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|$.

E.5.3.3. Considere o seguinte PVC

$$-u'' + u' = f(x), \quad -1 < x < 1, \quad (5.94a)$$

$$u(-1) = 0, \quad (5.94b)$$

$$u'(1) = 0, \quad (5.94c)$$

onde

$$f(x) = \begin{cases} 1 & , x \leq 0 \\ 0 & , x > 0 \end{cases} \quad (5.95)$$

Use uma aproximação adequada pelo MVF para obter o valor aproximado de $u(0)$ com precisão de 2 dígitos significativos.

E.5.3.4. Considere o PVC

$$-u'' = \pi^2 \cos(\pi x), \quad 0 < x < 1, \quad (5.96)$$

$$u(0) = 1, \quad (5.97)$$

$$u'(1) = 0. \quad (5.98)$$

A solução analítica deste problema é $u(x) = \cos(\pi x)$. Aplique o MVF para computar uma aproximação numérica com erro absoluto de no máximo 10^{-3} na norma L^2 .

5.4 Problemas Não-Lineares

Vamos estudar a resolução de **Problemas Não-Lineares de Valores de Contorno** da forma

$$u_{xx} = f(x, u, u_x), \quad a \leq x \leq b, \quad (5.99a)$$

$$u(a) = u_a, \quad (5.99b)$$

$$u(b) = u_b, \quad (5.99c)$$

onde $f = f(x, u, u_x)$ é uma função não linear para u ou u_x .

Empregando o **Método de Diferenças Finitas** (MDF), começamos assumindo uma **malha** uniforme de n -subintervalos com **nodos** $x_i = a + (i-1)h$, **tamanho de malha** $h = (b-a)/n$, $i = 1, 2, \dots, n+1$. Denotando $u_i \approx u(x_i)$ e aplicando fórmulas de diferenças finitas centrais para u_{xx} e u_x , a Eq. (5.99a) fornece

$$\frac{1}{h^2}u_{i-1} - \frac{2}{h^2}u_i + \frac{1}{h^2}u_{i+1} = f\left(x_i, u_i, \frac{1}{2h}u_{i+1} - \frac{1}{2h}u_{i-1}\right), \quad (5.100)$$

para $i = 2, 3, \dots, n$. As condições de contorno Eqs. (5.99b)-(5.99c), fornecem as equações de fechamento

$$u_1 = u_a, \quad (5.101a)$$

$$u_{n+1} = u_b. \quad (5.101b)$$

Com isso, temos que o **problema discreto** associado consiste em: encontrar $\mathbf{u} = (u_i)_{i=1}^{n+1}$ solução do seguinte sistema de equações não-lineares

$$u_1 - u_a = 0, \quad (5.102a)$$

$$-\frac{1}{h^2}u_{i-1} + \frac{2}{h^2}u_i - \frac{1}{h^2}u_{i+1} + f\left(x_i, u_i, \frac{1}{2h}u_{i+1} - \frac{1}{2h}u_{i-1}\right) = 0, \quad (5.102b)$$

$$u_{n+1} - u_b = 0. \quad (5.102c)$$

A **resolução do problema discreto** (5.102) pode ser feito com o **Método de Newton**¹⁷. Para tanto, observamos que o sistema tem a forma vetorial

$$F(\mathbf{u}) = \mathbf{0}, \quad (5.103)$$

onde $F(\mathbf{u}) = (f_i(\mathbf{u}))_{i=1}^{(n+1)}$ é a função vetorial de componentes

$$f_1(\mathbf{u}) = u_1 - u_a, \quad (5.104a)$$

$$f_i(\mathbf{u}) = -\frac{1}{h^2}u_{i-1} + \frac{2}{h^2}u_i - \frac{1}{h^2}u_{i+1} + f\left(x_i, u_i, \frac{1}{2h}u_{i+1} - \frac{1}{2h}u_{i-1}\right), \quad (5.104b)$$

$$f_{n+1}(\mathbf{u}) = u_{n+1} - u_b, \quad (5.104c)$$

com $i = 2, 3, \dots, n$. A **iteração do Método de Newton** consiste em

$$\mathbf{u}^{(0)} = \text{aprox. inicial}, \quad (5.105a)$$

$$\mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \boldsymbol{\delta}^{(k)}, \quad (5.105b)$$

onde $\boldsymbol{\delta}^{(k)}$ é a **atualização de Newton** computada por

$$J_F(\mathbf{u}^{(k)})\boldsymbol{\delta}^{(k)} = -F(\mathbf{u}^{(k)}), \quad (5.106)$$

para $k = 0, 1, 2, \dots$ até que um critério de parada seja satisfeito. A **matriz jacobiana**¹⁸ é denotada por $J_F(\mathbf{u}^{(k)}) = [J_{i,j}]_{i,j=1}^{n+1, n+1}$ e tem elementos não nulos

$$J_{1,1} = 1, \quad (5.107)$$

$$j_{i,i-1} = -\frac{1}{h^2} - \frac{1}{2h} f_{u_x} \left(x_i, u_i, \frac{1}{2h} u_{i+1} - \frac{1}{2h} u_{i-1} \right), \quad (5.108a)$$

$$j_{i,i} = \frac{2}{h^2} + f_u \left(x_i, u_i, \frac{1}{2h} u_{i+1} - \frac{1}{2h} u_{i-1} \right), \quad (5.108b)$$

$$j_{i,i+1} = -\frac{1}{h^2} + \frac{1}{2h} f_{u_x} \left(x_i, u_i, \frac{1}{2h} u_{i+1} - \frac{1}{2h} u_{i-1} \right), \quad (5.108c)$$

para $i = 2, 3, \dots, n$ e

$$j_{n+1,n+1} = 1. \quad (5.109)$$

Exemplo 5.4.1. Vamos considerar o seguinte PVC

$$uu_x - u_{xx} = \pi \sin(\pi x) [\pi + \cos(\pi x)], \quad 0 < x < 1, \quad (5.110a)$$

$$u(0) = u(1) = 0. \quad (5.110b)$$

Rearranjando os termos, podemos escrevê-lo na forma da Eq. (5.99), com $u_a = u_b = 0$ e

$$f(x, u, u_x) = uu_x - \pi \sin(\pi x) [\pi + \cos(\pi x)]. \quad (5.111)$$

Com isso, calculamos

$$f_u(x, u, u_x) = u_x, \quad (5.112a)$$

$$f_{u_x}(x, u, u_x) = u. \quad (5.112b)$$

Então, a aplicação do MDF-Newton com $h = 10^{-1}$ fornece o resultado da Fig. 5.4. A solução exata é $u(x) = \sin(\pi x)$.

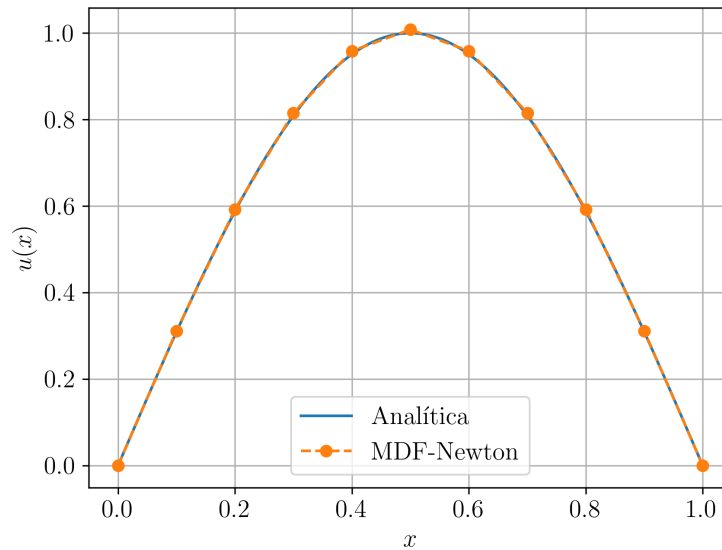


Figura 5.4: Resultado da aplicação do MDF-Newton para o PVC do Ex. 5.4.1.

Código 5.4: mdf-newton.py

```
1 import numpy as np
2 import numpy.linalg as npla
3 from numpy import pi, sin, cos
4
5 # parâmetros
6 n = 10
7 h = 1./n
8 xx = np.linspace(0., 1., n+1)
9
10 # c.c. Dirichlet
11 ua = 0.
12 ub = 0.
13
14 def f(x, u, ux):
15     return u*ux - pi*sin(pi*x)*(pi + cos(pi*x))
16
```

```

17 def fu(x, u, ux):
18     return ux
19
20 def fux(x, u, ux):
21     return u
22
23 # rhs
24 def F(u):
25     y = np.empty(n+1)
26     # f_1
27     y[0] = u[0] - ua
28     # f_i
29     for i in range(1,n):
30         ux = u[i+1]/(2*h) - u[i-1]/(2*h)
31         y[i] = -1./h**2*u[i-1] + 2./h**2*u[i] -
1./h**2*u[i+1] \
32             + f(xx[i], u[i], ux)
33     # f_{n+1}
34     y[n] = u[n] - ub
35
36     return y
37
38 # jacobiana
39 def J(u):
40     J = np.zeros((n+1,n+1))
41     J[0,0] = 1.
42     for i in range(1,n):
43         ux = 1./(2*h)*u[i+1] - 1./(2*h)*u[i-1]
44         J[i,i-1] = -1./h**2 - 1/(2*h)\
45             * fux(xx[i], u[i], ux)
46         J[i,i] = 2/h**2 + fu(xx[i], u[i], ux)
47         J[i,i+1] = -1./h**2 + 1/(2*h)\
48             * fux(xx[i], u[i], ux)
49     J[n,n] = 1.
50
51     return J
52
53 # aprox inicial

```

```

54 u = np.zeros(n+1)
55
56 # iterações de Newton
57 maxiter = 10
58 for k in range(maxiter):
59
60     # passo de Newton
61     dlta = npla.solve(J(u), -F(u))
62
63     # atualização
64     u += dlta
65
66     ndlta = npla.norm(dlta)
67     print(f'{k+1}: norm = {ndlta:.2e}')
68     if (ndlta < 1e-10):
69         print('convergiu.')
70         break

```

5.4.1 Exercícios

E.5.4.1. Considere o PVC

$$u^2 - u_{xx} = \cos^2(\pi x) + \pi^2 \cos(\pi x), \quad 0 < x < 1, \quad (5.113a)$$

$$u(0) = 1, \quad (5.113b)$$

$$u(1) = -1. \quad (5.113c)$$

Este problema tem solução analítica $u(x) = \cos(\pi x)$. Use o MDF-Newton para computar u_h aproximações de u para $h = 10^{-1}, 10^{-2}, 10^{-3}, 10^{-4}$. Então, verifique a convergência com base no erro $\varepsilon_h := \|\tilde{\mathbf{u}} - \mathbf{u}\|_2$. A convergência tem a taxa esperada? Justifique sua resposta.

E.5.4.2. Considere o PVC

$$uu_x - u_{xx} = 2 + x(1-x)(1-2x), \quad 0 < x < 1, \quad (5.114a)$$

$$u_x(0) = 1, \quad (5.114b)$$

$$u(1) = 0. \quad (5.114c)$$

Este problema tem solução analítica $u(x) = x(1 - x)$. Use o MDF-Newton para computar u_h aproximações de u para $h = 10^{-1}, 10^{-2}, 10^{-3}$:

- a) aplicando as diferenças finitas $D_{0,h^2}u(x)$ para $0 < x < 1$ e $D_{+,h}u(x)$ para $x = 0$.
- b) aplicando as diferenças finitas $D_{0,h^2}u(x)$ para $0 < x < 1$ e $D_{+,h^2}u(x)$ para $x = 0$.

Qual dessas formulações tem a melhor taxa de convergência do erro em relação ao passo de malha h ? Justifique e verifique sua resposta.

E.5.4.3. Desenvolva uma versão do método MEF-Newton ([Método de Elementos Finitos](#) com o [Método de Newton](#)) para computar a solução aproximada do PVC dado no Exemplo 5.4.1. Implemente-o e verifique a convergência do método para $h = 10^{-1}, 10^{-2}$ e 10^{-3} .

E.5.4.4. Desenvolva uma versão do método MVF-Newton ([Método de Volumes Finitos](#) com o [Método de Newton](#)) para computar a solução aproximada do PVC dado no Exemplo 5.4.1. Implemente-o e verifique a convergência do método para $h = 10^{-1}, 10^{-2}$ e 10^{-3} .

E.5.4.5. Desenvolva uma versão do método MEF-Newton ([Método de Elementos Finitos](#) com o [Método de Newton](#)) para computar a solução aproximada do PVC dado no Exercício 5.4.2. Implemente-o e verifique a convergência do método para $h = 10^{-1}, 10^{-2}$ e 10^{-3} .

E.5.4.6. Desenvolva uma versão do método MVF-Newton ([Método de Volumes Finitos](#) com o [Método de Newton](#)) para computar a solução aproximada do PVC dado no Exercício 5.4.2. Implemente-o e verifique a convergência do método para $h = 10^{-1}, 10^{-2}$ e 10^{-3} .

Capítulo 6

Equações Diferenciais Parciais

Neste capítulo, estudamos alguns tópicos fundamentais da aplicação do Método de Diferenças Finitas (MDF) para a solução numérica de Equações Diferenciais Parciais (EDPs).

6.1 Equação de Poisson

Consideramos a **equação de Poisson**¹⁹ (ou **equação de Laplace**²⁰ heterogênea) no domínio retangular $D = (a, b) \times (c, d)$ com condições de contorno de Dirichlet homogêneas

$$\Delta u = f(x, y), \quad (x, y) \in D, \quad (6.1a)$$

$$u(x, y) = 0, \quad \partial D, \quad (6.1b)$$

onde $u = u(x, y)$ é a incógnita, $\Delta u := u_{xx} + u_{yy}$ e ∂D é a fronteira do domínio D .

A aplicação do Método de Diferenças Finitas para resolver este problema consiste dos mesmos passos usados para resolver problemas de valores de contorno (consulte Seção 5.1), a saber: 1. discretização do domínio, 2. discretização das equações, 3. resolução do problema discreto.

1. Discretização do Domínio (Malha).

$$y_j := c + (j - 1)h_y, \quad (6.2b)$$

O produto cartesiano das partições em x e y nos fornece uma partição do domínio \overline{D} da forma

$$P(\overline{D}) = \{(x_1, y_1), (x_1, y_2), \dots, (x_i, y_j), \dots, (x_{n_x}, y_{n_y})\}, \quad (6.3)$$

cujos nodos (x_i, y_j) podem ser enumerados (indexados) por $k = i + (j-1)(n_x + 1)$. Por simplicidade, no decorrer do texto, assumiremos $n_x = n_y =: n$ e, por conseguinte, $h_x = h_y = h$ e temos a **enumeração**

$$k = i + (j-1)(n+1). \quad (6.4)$$

Consulte a Figura 6.1.

2. Discretização das Equações.

Usando a **fórmula de diferenças finitas central** de ordem h^2 para a segunda derivada, temos

$$u_{xx}(x, y) = \frac{u(x+h, y) - 2u(x, y) + u(x-h, y)}{h^2} + O(h^2), \quad (6.5)$$

$$u_{yy}(x, y) = \frac{u(x, y+h) - 2u(x, y) + u(x, y-h)}{h^2} + O(h^2). \quad (6.6)$$

Daí, denotando $u_{ij} \approx u(x_i, y_j)$ temos

$$u_{xx}(x_i, y_j) = \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} + O(h^2), \quad (6.7)$$

$$u_{yy}(x_i, y_j) = \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + O(h^2). \quad (6.8)$$

Então, da Eq. 6.1a temos

$$\begin{aligned} & \frac{u_{i+1,j} - 2u_{i,j} + u_{i-1,j}}{h^2} \\ & + \frac{u_{i,j+1} - 2u_{i,j} + u_{i,j-1}}{h^2} + O(h^2) = f(x_i, y_j). \end{aligned} \quad (6.9)$$

Agora, com base na enumeração (6.4) denotamos $u_k := u_{i+(j-1)(n+1)}$, desprezando o erro de truncamento e rearranjando os termos, obtemos

$$\frac{1}{h^2}u_{k-n} + \frac{1}{h^2}u_{k-1} - \frac{4}{h^2}u_k + \frac{1}{h^2}u_{k+1} + \frac{1}{h^2}u_{k+n} = f_k, \quad (6.10)$$

para $k = i + (j+1)(n+1)$ com $i, j = 2, 3, \dots, n$ (nodos internos). Isto é, esta última expressão nos fornece um sistema de $(n-1)^2$ equações para $(n+1)^2$ incógnitas $\mathbf{u} = (u_k)_{k=1}^{(n+1)^2}$. Para fechar o sistema, usamos as condições de contorno (6.1b)

$$u_k = 0 \quad (6.11)$$

para $k = i + (j + 1)(n + 1)$ com $i = 1, n + 1$ e $j = 1, 2, \dots, n + 1$, ou $i = 2, 3, \dots, n$ e $j = 1, n + 1$.

Com isso, o **problema discreto** obtido da aplicação do MDF **consiste no** sistema linear de $(n + 1)^2 \times (n + 1)^2$ (6.10)-(6.11).

3. Resolução do Problema Discreto.

O problema discreto (6.10)-(6.11) pode ser escrito na forma matricial

$$A\mathbf{u} = \mathbf{b}, \quad (6.12)$$

onde o vetor da incógnitas é $\mathbf{u} = (u_k)_{k=1}^{(n+1)^2}$. A matriz dos coeficientes $A = [a_{l,m}]_{l,m=1}^{(n+1)^2, (n+1)^2}$ e o vetor dos termos constantes $\mathbf{b} = (b_k)_{k=1}^{(n+1)^2}$ têm elementos não nulos

$$\begin{aligned} i = 1, n + 1, \quad j = 1, 2, \dots, n + 1 : \\ a_{k,k} = 1, \\ b_k = 0, \end{aligned} \quad (6.13)$$

$$\begin{aligned} i = 1, 2, \dots, n + 1, \quad j = 1, n + 1 : \\ a_{k,k} = 1, \\ b_k = 0, \end{aligned} \quad (6.14)$$

$$\begin{aligned} i, j = 2, 3, \dots, n \quad : \\ a_{k,k-n} = \frac{1}{h^2}, \\ a_{k,k-1} = \frac{1}{h^2}, \\ a_{k,k} = -\frac{4}{h^2}, \\ a_{k,k+1} = \frac{1}{h^2}, \\ a_{k,k+n} = \frac{1}{h^2}, \\ b_k = f(x_i, y_j). \end{aligned} \quad (6.15)$$

Assim sendo, basta empregarmos um método apropriado para resolver o sistema linear (6.12) para obter a solução aproximada de u nos nodos (x_i, y_j) .

Exemplo 6.1.1. Consideramos o seguinte problema

$$\Delta u = -2\pi^2 \sin(\pi x) \sin(\pi y), \quad (x, y) \in (0, 1)^2, \quad (6.16a)$$

$$u = 0, \quad (x, y) \in \partial D. \quad (6.16b)$$

A solução exata é $u(x, y) = \sin(\pi x) \sin(\pi y)$.

A Figura 6.2 mostra o gráfico de superfície da solução aproximada obtida pelo MDF com $h = 10^{-1}$. A Figura 6.3 mostra a comparação entre os gráficos de contorno das soluções numérica e exata (linhas brancas).

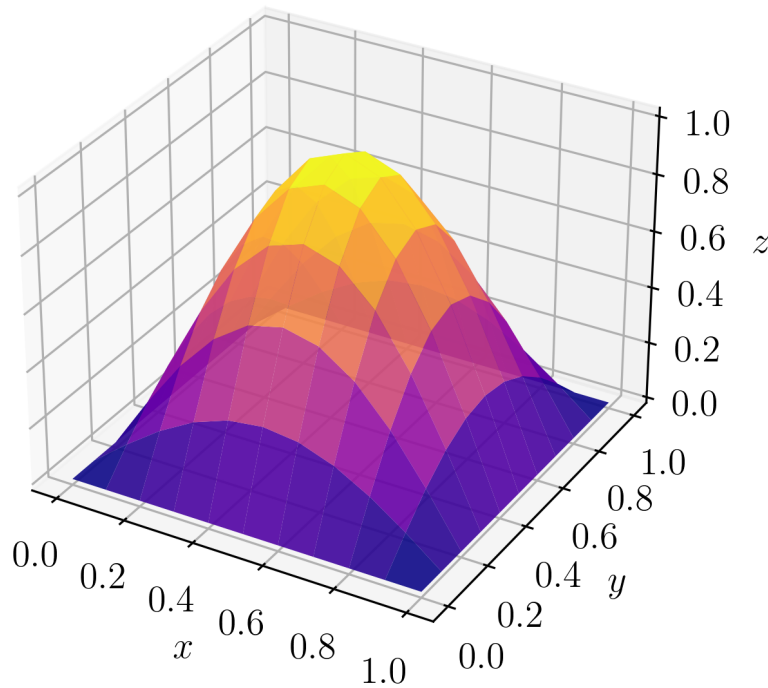


Figura 6.2: Solução aproximada do problema de Poisson do Exemplo 6.1.1.

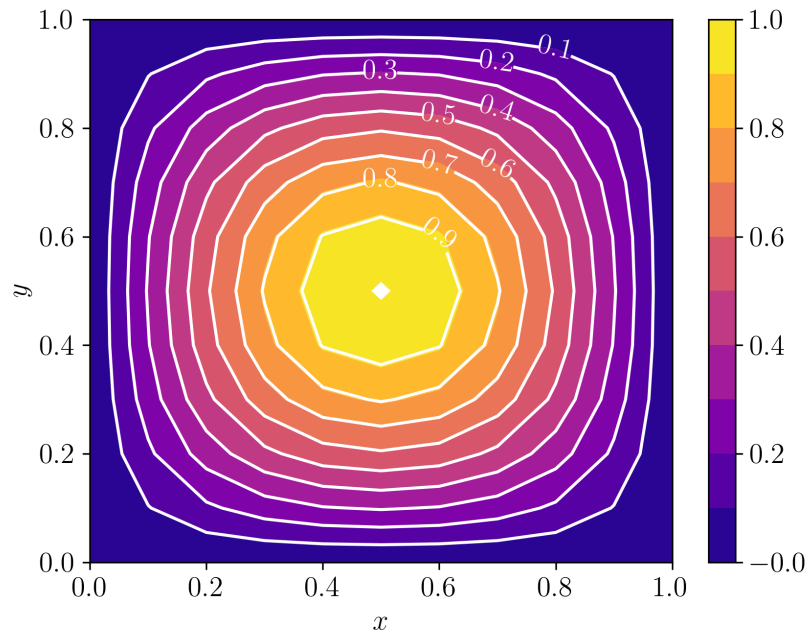


Figura 6.3: Comparação das soluções numérica e exata (isolinhas brancas) do Exemplo 6.1.1.

Código 6.1: mdf_poisson.py

```
1 import numpy as np
2 import numpy.linalg as npla
3
4 # malha
5 n = 10
6 h = 1./n
7 xx = np.linspace(0., 1., n+1)
8 yy = np.linspace(0., 1., n+1)
9
10 # rhs
11 def f(x,y):
12     return -2*np.pi**2*np.sin(np.pi*x)*np.sin(np.
13         pi*y)
14 # problema discreto
```

```
15 A = np.zeros(((n+1)**2, (n+1)**2))
16 b = np.empty((n+1)**2)
17
18 # c.c.
19 for j in range(n+1):
20     # i = 0
21     k = j*(n+1)
22     A[k,k] = 1.
23     b[k] = 0.
24     # i = n
25     k = n + j*(n+1)
26     A[k,k] = 1.
27     b[k] = 0.
28
29 for i in range(1,n):
30     # j = 0
31     k = i
32     A[k,k] = 1.
33     b[k] = 0.
34     # j = n
35     k = i + n*(n+1)
36     A[k,k] = 1.
37     b[k] = 0.
38
39 # pts internos
40 for i in range(1,n):
41     for j in range(1,n):
42         k = i + j*(n+1)
43         A[k,k-n-1] = 1./h**2
44         A[k,k-1] = 1./h**2
45         A[k,k] = -4./h**2
46         A[k,k+1] = 1./h**2
47         A[k,k+n+1] = 1./h**2
48         b[k] = f(xx[i],yy[j])
49
50 # resol p.d.
51 u = npla.solve(A, b)
```

6.1.1 Exercícios

E.6.1.1. Use o MDF para encontrar uma solução aproximada do seguinte problema de Poisson

$$\Delta u = -2\pi^2 \sin(\pi x) \sin(\pi y), \quad (x, y) \in D = (-1, 1)^2, \quad (6.17a)$$

$$u = 0, \quad (x, y) \in \partial D. \quad (6.17b)$$

A solução exata é $u(x, y) = \sin(\pi x) \sin(\pi y)$. Faça uma comparação gráfica entre as soluções numérica e exata no caso de $h = 10^{-1}$ (malha uniforme). Compare o erro $\varepsilon_h := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|_2$ para $n = 10, 20, 40, 80, 160$ (número de subintervalos na malha uniforme). A taxa de convergência é a esperada? Justifique sua resposta.

E.6.1.2. Use o MDF para encontrar uma solução aproximada do seguinte problema de Laplace

$$\Delta u = 0, \quad (x, y) \in (0, 1)^2, \quad (6.18a)$$

$$u(0, y) = u(1, y) = y^2 - y, \quad 0 \leq y \leq 1, \quad (6.18b)$$

$$u(x, 0) = u(x, 1) = x - x^2, \quad 0 \leq x \leq 1. \quad (6.18c)$$

A solução exata é $u(x, y) = x - x^2 + y^2 - y$. Faça uma comparação gráfica entre as soluções numérica e exata no caso de $h = 10^{-1}$. Compare o erro $\varepsilon_h := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|_2$ para $n = 10, 20, 40, 80, 160$ (número de subintervalos na malha uniforme).

E.6.1.3. Considere o problema

$$\Delta u = -2\pi^2 \sin(\pi x) \sin(\pi y), \quad (x, y) \in (0, 1)^2, \quad (6.19a)$$

$$u(0, y) = 0, \quad 0 \leq y \leq 1, \quad (6.19b)$$

$$u_x(1, y) = 0, \quad 0 \leq y \leq 1, \quad (6.19c)$$

$$u(x, 0) = u(x, 1) = 0, \quad 0 \leq x \leq 1. \quad (6.19d)$$

A solução exata é $u(x, y) = \sin(\pi x) \sin(\pi y)$. Com uma malha uniforme, obtenha uma solução aproximada com o MDF empregando, na fronteira com condições de Neumann²¹:

a) $D_{-,h}$ fórmulas diferença regressiva de ordem h .

b) D_{-,h^2} diferença regressiva de ordem h^2 .

Compare a taxa de convergência do erro $\varepsilon_h := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|_2$ entre essas duas formulações.

E.6.1.4. Considere o problema

$$\Delta u = -2\pi^2 \sin(\pi x) \sin(\pi y), \quad (x, y) \in (0, 1)^2, \quad (6.20a)$$

$$u(0, y) = u(1, y) = 0, \quad 0 \leq y \leq 1, \quad (6.20b)$$

$$u_y(x, 0) = u_y(x, 1) = 0, \quad 0 \leq x \leq 1. \quad (6.20c)$$

A solução exata é $u(x, y) = \sin(\pi x) \sin(\pi y)$. Com uma malha uniforme, obtenha uma solução aproximada com o MDF empregando, nas fronteiras com condições de Neumann:

a) fórmulas de diferenças finitas de $O(h)$.

b) fórmulas de diferenças finitas de $O(h^2)$.

Compare a taxa de convergência do erro $\varepsilon_h := \|\tilde{\mathbf{u}}_h - \mathbf{u}\|_2$ entre essas duas formulações.

E.6.1.5. Use o MDF para encontrar uma solução aproximada do seguinte problema de Poisson

$$\Delta u = 1, \quad (x, y) \in D = (-1, 1)^2, \quad (6.21a)$$

$$u = 0, \quad (x, y) \in \partial D. \quad (6.21b)$$

Usando uma malha uniforme, obtenha soluções para $n = 10, 20, 40, 80, 160$ (número de subintervalos). Sua solução está correta? Justifique sua resposta.

6.2 Equação do Calor

Consideramos a **equação do calor** com condição inicial dada e condições de contorno de Dirichlet homogêneas

$$u_t - \alpha u_{xx} = f(t, x), \quad 0 < t \leq t_f, \quad a < x < b, \quad (6.22a)$$

$$u(0, x) = u_0(x), \quad a < x < b, \quad (6.22b)$$

$$u(t, a) = u(t, b) = 0, \quad 0 < t \leq t_f, \quad (6.22c)$$

onde $u = u(t, x)$ é a incógnita.

O problema (6.22) é um problema de valor inicial com condições de contorno. Uma das estratégias numéricas de solução é o chamado **Método das Linhas**, o qual trata separadamente as discretizações espacial e temporal. Aqui, vamos começar pela discretização espacial e, então, trataremos a discretização temporal.

1. Discretização Espacial.

Na discretização espacial, aplicamos o **Método de Diferenças Finitas** (MDF). Começamos considerando uma malha uniforme de nodos $x_i = a + (i-1)h_x$, $i = 1, 2, \dots, n_x + 1$, com tamanho de malha $h_x = (b-a)/n_x$, sendo n_x o número de subintervalos. Denotando $u_i(t) \approx u(t, x_i)$ e empregando a fórmula de diferenças finitas centrais D_{0,h^2}^2 , temos que a Eq. (6.22a) fica aproximada por

$$\frac{du_i}{dt} = \frac{\alpha}{h_x^2} u_{i-1} - \frac{2\alpha}{h_x^2} u_i + \frac{\alpha}{h_x^2} u_{i+1} + f(t, x_i), \quad (6.23)$$

para $i = 2, 3, \dots, n_x$. Agora, das condições de contorno (6.22c), temos $u_1 = 0$ e $u_n = 0$. Com isso, obtemos o seguinte sistema de equações diferenciais ordinárias

$$\frac{du_2}{dt} = -\frac{2\alpha}{h_x^2} u_2 + \frac{\alpha}{h_x^2} u_3 + f(t, x_2), \quad (6.24a)$$

$$\frac{du_i}{dt} = \frac{\alpha}{h_x^2} u_{i-1} - \frac{2\alpha}{h_x^2} u_i + \frac{\alpha}{h_x^2} u_{i+1} + f(t, x_i), \quad (6.24b)$$

$$\frac{du_n}{dt} = \frac{\alpha}{h_x^2} u_{n-2} - \frac{2\alpha}{h_x^2} u_{n-1} + f(t, x_{n-1}), \quad (6.24c)$$

onde $i = 3, 4, \dots, n-1$ e com condições iniciais dadas por (6.22b), i.e.

$$u_i(0) = u_0(x_i), \quad (6.25)$$

para $i = 2, 3, \dots, n$. Este sistema pode ser escrito na seguinte forma matricial

$$\frac{d\tilde{u}}{dt} = A\tilde{u} + \tilde{f}, \quad (6.26)$$

onde $\tilde{\mathbf{u}}(t) = (u_2(t), u_3(t), \dots, u_n(t))$, $\tilde{f}(t) = (f(t, x_2), f(t, x_3), \dots, f(t, x_n))$ e A é uma matriz $(n-1) \times (n-1)$ da forma

$$A = \begin{bmatrix} -\frac{2\alpha}{h^2} & \frac{\alpha}{h^2} & 0 & 0 & 0 & \cdots & 0 & 0 \\ \frac{\alpha}{h^2} & -\frac{2\alpha}{h^2} & \frac{\alpha}{h^2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & \frac{\alpha}{h^2} & -\frac{2\alpha}{h^2} & \frac{\alpha}{h^2} & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & \cdots & 0 & 0 \\ 0 & 0 & & 0 & 0 & \cdots & \frac{\alpha}{h^2} & -\frac{2\alpha}{h^2} \end{bmatrix}. \quad (6.27)$$

2. Discretização Temporal.

Para a discretização temporal vamos usar o **esquema- θ** . Consideramos os tempos discretos $t^{(k)} = kh_t$, com passo no tempo $h_t = t_f/n_t$, para $k = 0, 1, 2, \dots, n_t$. Denotando $u_i^{(k)} \approx u(t^{(k)}, x_i)$, o esquema consiste nas iterações

$$\tilde{\mathbf{u}}^{(0)} = \tilde{\mathbf{u}}_0 \quad (6.28a)$$

$$\begin{aligned} \tilde{\mathbf{u}}^{(k+1)} = \tilde{\mathbf{u}}^{(k)} + (1-\theta)h_t \left(A\tilde{\mathbf{u}}^{(k)} + \tilde{\mathbf{f}}^{(k)} \right) \\ + \theta h_t \left(A\tilde{\mathbf{u}}^{(k+1)} + \tilde{\mathbf{f}}^{(k+1)} \right), \end{aligned} \quad (6.28b)$$

para $k = 0, 1, \dots, n_t - 1$ e para um escolhido $0 \leq \theta \leq 1$. No caso, f não depende de u e a Eq. (6.28b) é equivalente ao sistema linear

$$(I - \theta h_t A) \tilde{\mathbf{u}}^{(k+1)} = [I + (1-\theta)h_t A] \tilde{\mathbf{u}}^{(k)} + h_t \tilde{\mathbf{f}}_\theta^{(k)}, \quad (6.29)$$

com $\tilde{\mathbf{f}}_\theta^{(k)} = (1-\theta)\tilde{\mathbf{f}}^{(k)} + \theta\tilde{\mathbf{f}}^{(k+1)}$.

Observação 6.2.1. (Estabilidade e Erro de Truncamento.) Para $\theta = 0$ (**Método de Euler Explícito**) o esquema numérico **condicionalmente** estável [2, Cap. 12, Sec. 2] para

$$\alpha \frac{h_t}{h^2} \leq \frac{1}{2}. \quad (6.30)$$

Para $\theta = 1$ (**Método de Euler Implícito**) o esquema é incondicionalmente estável. Em ambos estes casos, o erro de truncamento é $O(h_t + h_x^2)$. Escolhendo-se $\theta = \frac{1}{2}$ (**Método de Crank-Nicolson**), o esquema numérico é incondicionalmente estável e com erro de truncamento $O(h_t^2 + h_x^2)$.

Exemplo 6.2.1. Consideramos o seguinte problema de calor

$$u_t - u_{xx} = (\pi^2 - 1)e^{-t} \sin(\pi x), \quad 0 < t \leq 1, \quad 0 \leq x \leq 1, \quad (6.31a)$$

$$u(0, x) = \sin(\pi x), \quad 0 \leq x \leq 1, \quad (6.31b)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 1, \quad (6.31c)$$

Este problema tem solução exata $u(t, x) = e^{-t} \sin(\pi x)$. A Figura 6.4 mostra o gráfico de superfície $u = u(t, x)$ da solução numérica. Na Figura 6.5, temos a comparação entre a solução numérica e a solução exata (isolinhas).

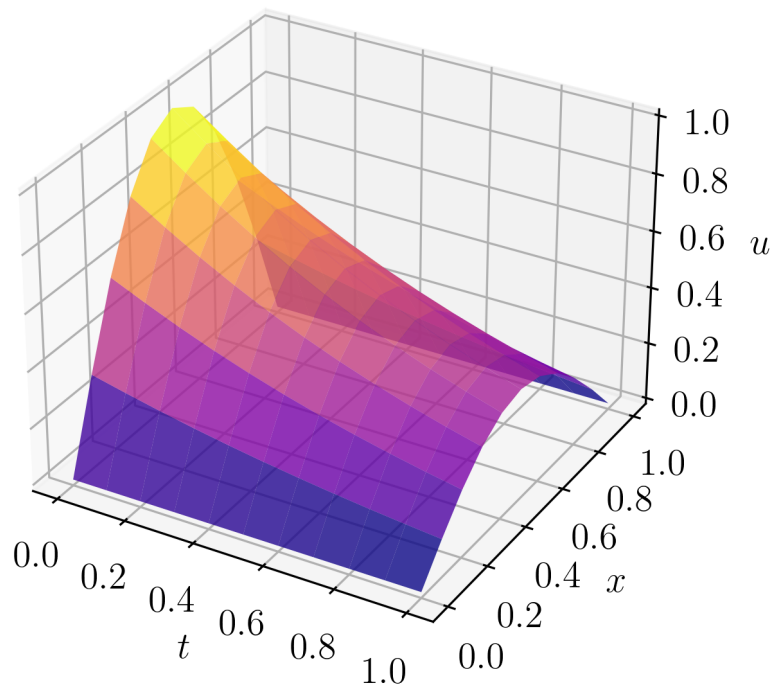


Figura 6.4: Solução aproximada do problema de calor do Exemplo 6.2.1.

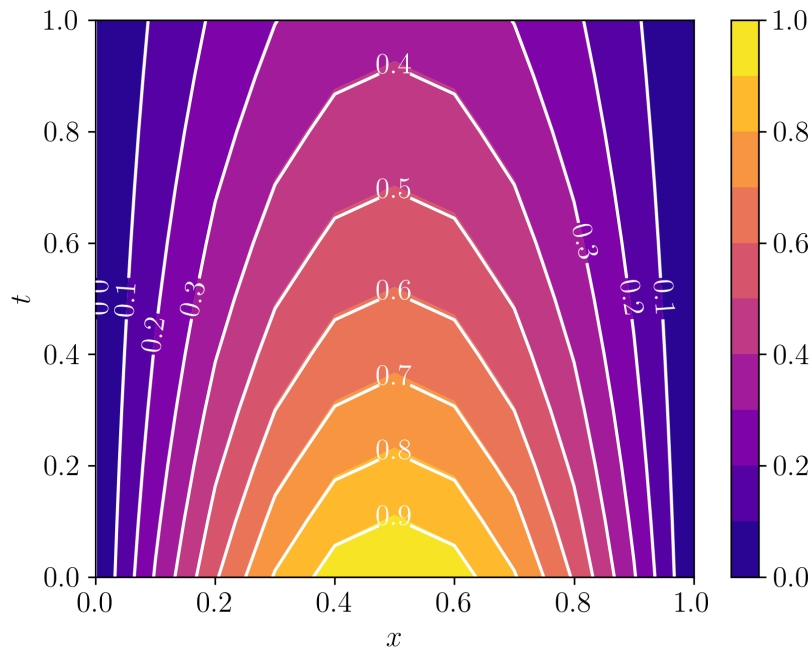


Figura 6.5: Comparação das soluções numérica e exata (isolinhas brancas) do Exemplo 6.2.1.

Código 6.2: calor.py

```

1 import numpy as np
2 import numpy.linalg as npla
3
4 # params
5 alpha = 1.
6 theta = 0.5
7
8 # malha temporal
9 nt = 10
10 ht = 1./nt
11 tt = np.linspace(0., 1., nt+1)
12
13 # malha espacial
14 nx = 10
15 h = 1./n

```

```
16 xx = np.linspace(0., 1., n+1)
17
18 # rhs
19 def f(t, x):
20     return (np.pi**2-1)*np.exp(-t)*np.sin(np.pi*x)
21
22 # auxiliares
23 lbda = alpha/h**2
24
25 # matriz difusão
26 A = np.zeros(((nx-1), (nx-1)))
27 A[0,0] = -2*lbda
28 A[0,1] = lbda
29 for i in range(1,nx-2):
30     A[i,i-1] = lbda
31     A[i,i] = -2*lbda
32     A[i,i+1] = lbda
33 A[nx-2,nx-3] = lbda
34 A[nx-2,nx-2] = -2*lbda
35
36 # matrizes auxiliares
37 Jth = np.identity(A.shape[0]) - theta*ht*A
38 J1th = np.identity(A.shape[0]) + (1-theta)*ht*A
39
40 # c.i.
41 u0 = np.sin(np.pi * xx)
42
43 # laço no tempo
44 u = u0.copy()
45 for k in range(nt):
46     print(f'{k+1}: t = {tt[k+1]:f}')
47     fth = (1-theta)*f(tt[k],xx[1:-1]) + theta*f(tt
48 [k+1],xx[1:-1])
49     u[1:-1] = npla.solve(Jth, J1th@u0[1:-1]+ht*fth
50 )
51     u0 = u.copy()
```

6.2.1 Exercícios

E.6.2.1. Considere o problema

$$u_t - u_{xx} = 0, \quad 0 < t \leq 1, \quad -\pi < x < \pi, \quad (6.32a)$$

$$u(0, x) = \sin(x), \quad -\pi \leq x \leq \pi, \quad (6.32b)$$

$$u(t, -\pi) = u(t, \pi) = 0, \quad 0 \leq t \leq 1. \quad (6.32c)$$

Sua solução exata é $u(t, x) = e^{-t} \sin(x)$. Implemente o MDF com esquema- θ em uma malha uniforme de tamanho espacial h_x e passo no tempo h_t para obter uma solução numérica \mathbf{u}_{h_x, h_t} . Então, verifique a taxa de convergência do erro $\varepsilon_{h_x, h_t} := \|\mathbf{u}_h - \mathbf{u}\|_2$ para os diferentes esquemas:

a) Euler Explícito: $\theta = 0$.

b) Euler Implícito: $\theta = 1$.

c) Crank-Nicolson: $\theta = \frac{1}{2}$.

E.6.2.2. Considere o problema

$$u_t - \alpha u_{xx} = 0, \quad 0 < t \leq 1, \quad -\pi < x < \pi, \quad (6.33a)$$

$$u(0, x) = \sin(\alpha x), \quad -\pi \leq x \leq \pi, \quad (6.33b)$$

$$u(t, -\pi) = u(t, \pi) = 0, \quad 0 \leq t \leq 1. \quad (6.33c)$$

Sua solução exata é $u(t, x) = e^{-\alpha^2 t} \sin(\alpha x)$. Implemente o MDF com esquema- θ em uma malha uniforme. Faça testes numéricos para analisar a validade da condição de estabilidade (6.30) para os seguintes esquemas:

a) Euler Explícito: $\theta = 0$.

b) Euler Implícito: $\theta = 1$.

c) Crank-Nicolson: $\theta = \frac{1}{2}$.

E.6.2.3. Considere o problema

$$u_t - u_{xx} = \left(\frac{\pi^2}{4} - 1 \right) e^{-t} \cos\left(\frac{\pi}{2} x \right), \quad 0 < t \leq 1, \quad 0 < x < 1, \quad (6.34a)$$

$$u(0, x) = \cos\left(\frac{\pi}{2}x\right), \quad 0 \leq x \leq 1, \quad (6.34b)$$

$$u(t, 0) = e^{-t}, \quad 0 \leq t \leq 1, \quad (6.34c)$$

$$u(t, 1) = 0, \quad 0 \leq t \leq 1. \quad (6.34d)$$

Sua solução exata é $u(t, x) = e^{-t} \cos(\pi x/2)$. Implemente o MDF com esquema- θ em uma malha uniforme de tamanho espacial h_x e passo no tempo h_t para obter uma solução numérica \mathbf{u}_{h_x, h_t} . Então, verifique a taxa de convergência do erro $\varepsilon_{h_x, h_t} := \|\mathbf{u}_h - \mathbf{u}\|_2$ para os diferentes esquemas:

a) Euler Explícito: $\theta = 0$.

b) Euler Implícito: $\theta = 1$.

c) Crank-Nicolson: $\theta = \frac{1}{2}$.

E.6.2.4. Considere o problema

$$u_t - u_{xx} = \left(\frac{\pi^2}{4} - 1\right) e^{-t} \cos\left(\frac{\pi}{2}x\right), \quad 0 < t \leq 1, \quad 0 < x < 1, \quad (6.35a)$$

$$u(0, x) = \cos\left(\frac{\pi}{2}x\right), \quad 0 \leq x \leq 1, \quad (6.35b)$$

$$u_x(t, 0) = 0, \quad 0 \leq t \leq 1, \quad (6.35c)$$

$$u(t, 1) = 0, \quad 0 \leq t \leq 1. \quad (6.35d)$$

Sua solução exata é $u(t, x) = e^{-t} \cos(\pi x/2)$. Implemente o MDF com o Método de Crank-Nicolson em uma malha uniforme para obter uma solução numérica \mathbf{u}_{h_x, h_t} . Então, verifique a taxa de convergência do erro $\varepsilon_{h_x, h_t} := \|\mathbf{u}_h - \mathbf{u}\|_2$ para os seguintes diferentes esquemas:

a) empregando a diferença finita D_{+, h_x} na condição de contorno de Neumann.

b) empregando a diferença finita D_{+, h_x^2} na condição de contorno de Neumann.

E.6.2.5. Considere o seguinte problema de calor

$$u_t - u_{xx} = (\pi^2 - 1)e^{-t} \sin(\pi x), \quad 0 < t \leq 1, \quad 0 \leq x \leq 1, \quad (6.36a)$$

$$u(0, x) = \sin(\pi x), \quad 0 \leq x \leq 1, \quad (6.36b)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 1, \quad (6.36c)$$

Sua solução exata $u(t, x) = e^{-t} \sin(\pi x)$. Faça implementações numéricas do Método das Linhas com MDF na discretização espacial e empregando os seguintes [métodos de Runge-Kutta](#) para resolver o sistema de EDOs associado:

a) Método do Ponto Médio.

b) Método de R-K-4.

E.6.2.6. (Equação de Burgers.) Considere o problema

$$u_t + uu_x = \alpha u_{xx}, \quad 0 < t \leq 1, \quad 0 < x < 1, \quad (6.37a)$$

$$u(0, x) = 2\alpha\pi \frac{\sin(\pi x)}{2 + \cos(\pi x)}, \quad 0 \leq x \leq 1, \quad (6.37b)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 1. \quad (6.37c)$$

Sua solução analítica é [9]

$$u(t, x) = 2\alpha\pi \frac{e^{-\alpha\pi^2 t} \sin(\pi x)}{2 + e^{-\alpha\pi^2 t} \cos(\pi x)}. \quad (6.38)$$

Faça uma implementação numérica com MDF e com esquema- θ para resolver este problema. Teste os esquemas para $\alpha = 1, 0.1, 0.01, 0.001$.

6.3 Equação da Onda

Consideramos a [equação da onda](#) com condições iniciais dadas e condições de contorno de Dirichlet homogêneas

$$u_{tt} - \alpha u_{xx} = 0, \quad 0 < t \leq t_f, \quad a < x < b, \quad (6.39a)$$

$$u(0, x) = f(x), \quad a \leq x \leq b, \quad (6.39b)$$

$$u_t(0, x) = g(x), \quad a \leq x \leq b, \quad (6.39c)$$

$$u(t, a) = u(t, b) = 0, \quad 0 \leq t \leq t_f, \quad (6.39d)$$

onde $u = u(t, x)$ é a incógnita com f, g e $\alpha > 0$ dadas.

Para a aplicação do [Método das Diferenças Finitas](#) (MDF), assumimos as discretizações: no tempo, $t^{(k)} = kh_t$, $j = 0, 1, \dots, n_t$, $h_t = t_f/n_t$; no espaço

$x_i = a + (i - 1)h_x$, $i = 1, 2, \dots, n_x + 1$, $h_x = (b - a)/n_x$. Então, assumindo a notação $u_i^{(k)} \approx u(t^{(k)}, x_i)$ usando a fórmula de diferenças finitas central D_{0,h^2}^2 , obtemos a seguinte forma discreta da equação Eq. (6.39a)

$$\frac{u_i^{(k+1)} - 2u_i^{(k)} + u_i^{(k+1)}}{h_t^2} - \alpha \frac{u_i^{(k)} - 2u_i^{(k)} + u_{i+1}^{(k)}}{h_x^2} = 0, \quad (6.40)$$

para $i = 2, 3, \dots, n_x$, $j = 1, 2, \dots, n_t - 1$. Denotando $\lambda := \alpha h_t^2 / h_x^2$, rearranjando os termos e aplicando as condições de contorno, obtemos

$$u_2^{(k+1)} = 2(1 - \lambda)u_2^{(k)} + \lambda u_3^{(k)} - u_2^{(k-1)}, \quad (6.41a)$$

$$u_i^{(k+1)} = \lambda u_i^{(k)} + 2(1 - \lambda)u_i^{(k)} + \lambda u_{i+1}^{(k)} - u_i^{(k-1)}, \quad (6.41b)$$

$$u_{n_x}^{(k+1)} = \lambda u_{n_x-1}^{(k)} + 2(1 - \lambda)u_{n_x}^{(k)} - u_{n_x}^{(k-1)}, \quad (6.41c)$$

para $i = 2, 3, \dots, n_x$, $j = 1, 2, \dots, n_t - 1$. Ou, equivalentemente, na forma matricial

$$\tilde{\mathbf{u}}^{(k+1)} = A\tilde{\mathbf{u}}^{(k)} - \tilde{\mathbf{u}}^{(k-1)}, \quad (6.42)$$

para $k = 1, 2, \dots, n_t - 1$, onde $\tilde{\mathbf{u}}^{(k)} = (u_i^{(k)})_{i=2}^{n_x}$ e $A = [a_{i,j}]_{i,j=1}^{n_x-1, n_x-1}$ é a matriz tridiagonal de elementos

$$a_{i,j} = \begin{cases} a_{i,i-1} = \lambda & , 1 < i \leq n_x - 1, \\ a_{i,i} = 2(1 - \lambda) & , 1 \leq i \leq n_x - 1, \\ a_{i,i+1} = \lambda & , 1 \leq i < n_x - 1. \end{cases} \quad (6.43)$$

Para a inicialização, a Eq. (6.42) requer que conheçamos $\tilde{\mathbf{u}}^{(0)}$ e $\tilde{\mathbf{u}}^{(1)}$. A primeira, vem diretamente da condição inicial Eq. (6.39b), i.e.

$$\tilde{\mathbf{u}}^{(0)} = f(\tilde{\mathbf{x}}), \quad (6.44)$$

onde $\tilde{\mathbf{x}} = (x_i)_{i=2}^{n_x}$. Agora, aplicando a fórmula de diferenças finitas progressiva $D_{+,h}$, temos da condição inicial Eq. (6.39c)

$$\frac{\tilde{\mathbf{u}}^{(1)} - \tilde{\mathbf{u}}^{(0)}}{h_t} = g(\tilde{\mathbf{x}}) \quad (6.45)$$

ou, equivalentemente,

$$\tilde{\mathbf{u}}^{(1)} = \tilde{\mathbf{u}}^{(0)} + h_t g(\tilde{\mathbf{x}}). \quad (6.46)$$

De tudo isso, temos que a solução numérica da equação da onda pode ser computada com a seguinte iteração

$$\tilde{\mathbf{u}}^{(0)} = f(\tilde{\mathbf{x}}), \quad (6.47a)$$

$$\tilde{\mathbf{u}}^{(1)} = \tilde{\mathbf{u}}^{(0)} + h_t g(\tilde{\mathbf{x}}), \quad (6.47b)$$

$$\tilde{\mathbf{u}}^{(k+1)} = A\tilde{\mathbf{u}}^{(k)} - \tilde{\mathbf{u}}^{(k-1)}, \quad (6.47c)$$

para $k = 1, 2, \dots, n_t - 1$, com $\mathbf{u}^{(k)} = (0, \tilde{\mathbf{u}}, 0)$.

Observação 6.3.1. (Condição de Estabilidade e Erro de Truncamento.)

Pode-se mostrar a seguinte condição de estabilidade [3, p. 487]

$$\alpha \frac{h_t}{h_x} \leq 1. \quad (6.48)$$

Com isso e para f e g suficientemente suaves, o esquema numérica (6.47) tem erro de truncamento $O(h_t^2 + h_x^2)$.

Exemplo 6.3.1. Consideramos o seguinte problema

$$u_{tt} - u_{xx} = 0, \quad 0 < t \leq 2, \quad 0 < x < 1, \quad (6.49a)$$

$$u(0, x) = 0, \quad 0 \leq x \leq 1, \quad (6.49b)$$

$$u_t(0, x) = \pi \sin(\pi x), \quad 0 \leq x \leq 1, \quad (6.49c)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 2. \quad (6.49d)$$

Sua solução exata é $u(t, x) = \sin(\pi t) \sin(\pi x)$. A Figura 6.6 contém gráficos de comparação entre as soluções numérica e exata. Para a solução numérica, tomamos $n_t = 40$ ($h_t = 0.05$) e $n_x = 10$ ($h_x = 0.1$).

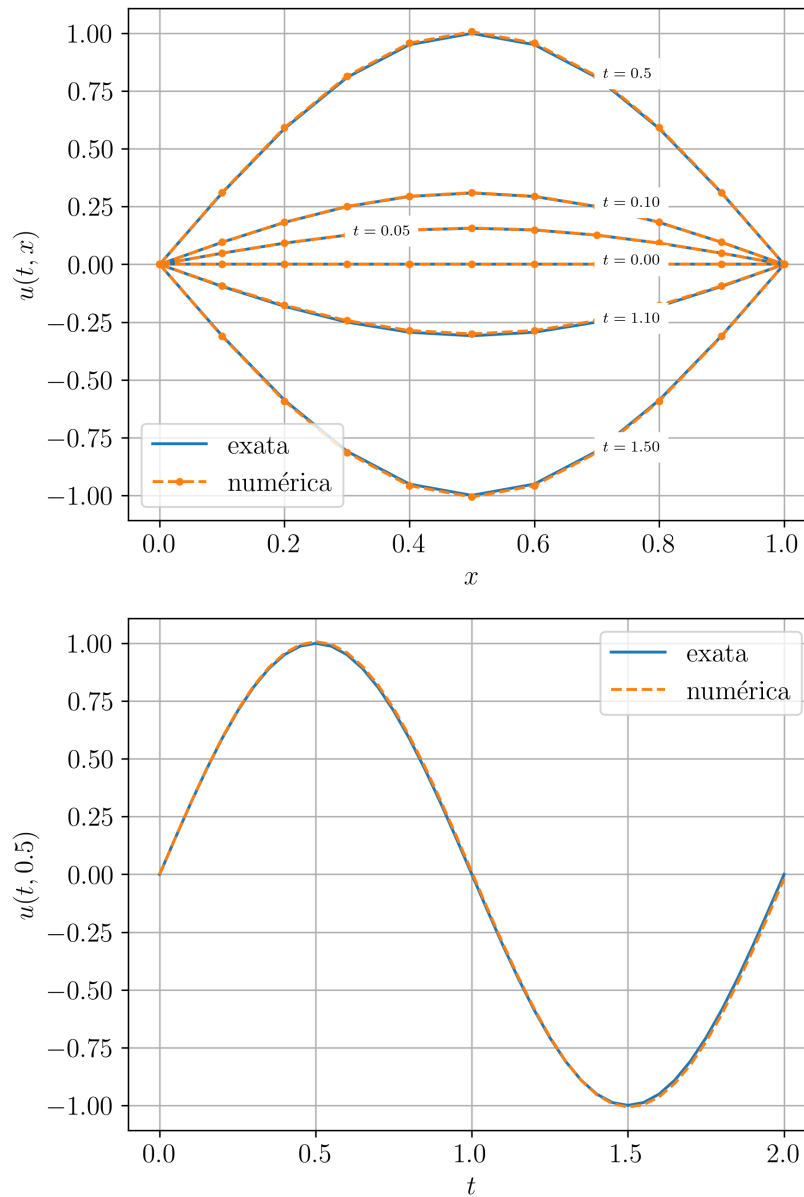


Figura 6.6: Gráficos comparativos das soluções numérica e exata do problema de onda do Exemplo 6.3.1.

```
1 import numpy as np
2 from numpy import pi, sin, cos
```



```
3
4 # params
5 nt = 40
6 ht = 2./nt
7 tt = np.linspace(0., 2., nt+1)
8
9 nx = 10
10 hx = 1./nx
11 xx = np.linspace(0., 1., nx+1)
12
13 # c.i.s
14 def f(x):
15     return np.zeros_like(x)
16
17 def g(x):
18     return pi*sin(pi*x)
19
20 # auxiliares
21 lbda = ht**2/hx**2
22
23 A = np.zeros(((nx-1), (nx-1)))
24 A[0,0] = 2*(1. - lbda)
25 A[0,1] = lbda
26 for i in range(1,nx-2):
27     A[i,i-1] = lbda
28     A[i,i] = 2*(1 - lbda)
29     A[i,i+1] = lbda
30 A[nx-2,nx-3] = lbda
31 A[nx-2,nx-2] = 2*(1 - lbda)
32
33 # laço no tempo
34 ## c.i.s
35 u0 = f(xx)
36
37 u1 = u0.copy()
38 u1[1:-1] = u0[1:-1] + ht*g(xx[1:-1])
39
40 u = u1.copy()
```

```

41 for k in range(1,nt):
42
43     print(f'{k+1}: t = {tt[k+1]:f}')
44
45     u[1:-1] = A@u1[1:-1] - u0[1:-1]
46
47     u0 = u1.copy()
48     u1 = u.copy()

```

6.3.1 Exercício

E.6.3.1. Considere o problema

$$u_{tt} - u_{xx} = 0, \quad 0 < t \leq 1.5, \quad 0 < x < 1, \quad (6.50a)$$

$$u(0, x) = 0, \quad 0 \leq x \leq 1, \quad (6.50b)$$

$$u_t(0, x) = \pi \sin(\pi x), \quad 0 \leq x \leq 1, \quad (6.50c)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 1.5. \quad (6.50d)$$

Sua solução exata é $u(t, x) = \sin(\pi t) \sin(\pi x)$. Faça testes numéricos para determinar os passos h_t e h_x para os quais o esquema numérico (6.47) compute o valor de $u(1.5, 0.5)$ com 5 dígitos significativos corretos.

E.6.3.2. Considere o problema

$$u_{tt} - u_{xx} = e^{-t}(2 + x - x^2), \quad 0 < t \leq 1, \quad 0 < x < 1, \quad (6.51a)$$

$$u(0, x) = x - x^2, \quad 0 \leq x \leq 1, \quad (6.51b)$$

$$u_t(0, x) = x^2 - x, \quad 0 \leq x \leq 1, \quad (6.51c)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 1. \quad (6.51d)$$

Sua solução exata é $u(t, x) = e^{-t}(x - x^2)$. Implemente um esquema numérico semelhante ao (6.47) para computar soluções numéricas desse problema.

E.6.3.3. Considere o problema

$$u_{tt} - u_{xx} = e^{-t}(2 + x - x^2), \quad 0 < t \leq 1, \quad 0 < x < 1, \quad (6.52a)$$

$$u(0, x) = x - x^2, \quad 0 \leq x \leq 1, \quad (6.52b)$$

$$u_t(0, x) = x^2 - x, \quad 0 \leq x \leq 1, \quad (6.52c)$$

$$u_x(t, 0) = e^{-t}, \quad 0 \leq t \leq 1, \quad (6.52d)$$

$$u(t, 1) = 0, \quad 0 \leq t \leq 1. \quad (6.52e)$$

Sua solução exata é $u(t, x) = e^{-t}(x - x^2)$. Implemente um esquema numérico semelhante ao (6.47) para computar soluções numéricas desse problema.

E.6.3.4. Considere o problema

$$u_{tt} - u_{xx} = 0, \quad 0 < t \leq 2, \quad 0 < x < 1, \quad (6.53a)$$

$$u(0, x) = 0, \quad 0 \leq x \leq 1, \quad (6.53b)$$

$$u_t(0, x) = \pi \sin(\pi x), \quad 0 \leq x \leq 1, \quad (6.53c)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 2. \quad (6.53d)$$

Sua solução exata é $u(t, x) = \sin(\pi t) \sin(\pi x)$. Baseado em (6.47), desenvolva um novo esquema numérico substituindo o passo (6.47b) por um esquema numérico de mais alta ordem.

E.6.3.5. Considere o problema

$$u_{tt} - \alpha u_{xx} = 0, \quad 0 < t \leq 1, \quad 0 < x < 1, \quad (6.54a)$$

$$u(0, x) = x(1 - x), \quad 0 \leq x \leq 1, \quad (6.54b)$$

$$u_t(0, x) = 0, \quad 0 \leq x \leq 1, \quad (6.54c)$$

$$u(t, 0) = u(t, 1) = 0, \quad 0 \leq t \leq 2. \quad (6.54d)$$

Use o esquema numérico (6.47) para fazer testes numéricos para $\alpha = 1, 0.5, 0.1, 0.01$. É necessário ajustar os parâmetros h_t e h_x ao variar o parâmetro α ? Justifique sua resposta.

Resposta dos Exercícios

E.1.1.1. $f'(\pi/3) = -0.866025e+0, h = 10^{-7}$

E.1.1.2. $f'(\pi/3) = -0.866025e+0, h = 10^{-6}$

E.1.1.3. $f'(\pi/3) = -0.866025e+0, h = 10^{-3}$

E.1.1.4. a) $D_{+,h}f(2.5) = 1,05949$; b) $D_{-,h}f(2.5) = 1,05877$; c) $D_{0,h^2}f(2.5) = 1,05913$;

E.1.1.5.

i	dy/dx
1	$4.0e-2$
2	$7.5e-1$
3	$1.3e+0$
4	$1.1e+0$
5	$7.5e-1$
6	$8.0e-1$

E.1.1.8.

i	dy/dx
1	5.0e-2
2	7.5e-1
3	1.3e+0
4	1.1e+0
5	7.5e-1
6	8.5e-1

E.1.2.1. a) $7.25162\text{e}-2$; b) $7.24701\text{e}-2$; c) $7.24696\text{e}-2$; d) $7.24696\text{e}-2$; $h = 10^{-2}$;

E.1.2.2. $f''(1) = 3.92288\text{e}+0$, $h = 10^{-3}$.

E.1.2.3. 4.0;

E.1.3.1. 1.05913

E.1.3.2.

- a) $\frac{1}{12h} [3f(x-4h) - 16f(x-3h) + 36f(x-2h) - 48f(x-h) + 25f(x)]$
 b) $\frac{1}{12h} [-f(x-3h) + 6f(x-2h) - 18f(x-h) + 10f(x) + 3f(x+h)]$
 c) $\frac{1}{12h} [f(x-2h) - 8f(x-h) + 8f(x+h) - f(x+2h)]$
 d) $\frac{1}{12h} [-3f(x-h) - 10f(x) + 18f(x+h) - 6f(x+2h) + f(x+3h)]$
 d) $\frac{1}{12h} [-25f(x) + 48f(x+h) - 36f(x+2h) + 16f(x+3h) - 3f(x+4h)]$

E.1.3.3.

i	1	2	3	4	5	6
dy/dx	1,7500e-1	7,2500e-1	1,4250e+0	1,1250e+0	4,2500e-1	1,6750e+0

E.2.1.2. a) 1,05919; b) 1,05916; c) 1,05913

E.3.1.1. $I = 1.08414\text{e}-1$, a) $\tilde{I} = 1.09356\text{e}-01$, $|\tilde{I} - I| = 9.4\text{e}-4$, b) $\tilde{I} =$

1.08413e-01, $|\tilde{I} - I| = 7.1\text{e-}07$, c) $\tilde{I} = 1.07942\text{e-}01$, $|\tilde{I} - I| = 4.7\text{e-}04$

E.3.1.2. a) $3,33647\text{e-}1$; b) $1,71368\text{e-}1$; c) $2,79554\text{e-}1$

E.3.1.3. a) $4,02000\text{e-}1$; b) $1,04250E + 0$; c) $8,08667\text{e-}1$

E.3.1.4. a) $\tilde{I} = 8.08667\text{e-}01$, b) $\tilde{I}_1 = 3.82333\text{e-}01$, c) $\tilde{I}_2 = 4.29333\text{e-}01$, d) $\tilde{\tilde{I}} = 8.11667\text{e-}01$. (mais exata)

E.3.1.5. Use um procedimento semelhante aquele usado para determinar a ordem do erro de truncamento da regra de Simpson.

E.3.1.6. $h := \frac{(b-a)}{3}$,

$$\int_a^b f(x) dx = \frac{3h}{2} \left[f\left(a + \frac{1}{3}(b-a)\right) + f\left(a + \frac{2}{3}(b-a)\right) \right] + O(h^3).$$

E.3.2.1. a) $2.69264\text{e-}1$; b) $2.68282\text{e-}1$; c) $2.68937\text{e-}1$

E.3.2.2. Dica: para cada quadratura, observe a convergência das aproximações com sucessivos refinamentos no número de intervalos.

E.3.2.3. a) $8.12000\text{e-}1$; b) 1.03850 ; c) $8.11667\text{e-}1$

E.3.3.1. $2,68953\text{e-}1$

E.3.4.1. 1

E.3.4.2. Dica: use a regra do ponto médio.

E.3.4.3. $x_1 = 0, \omega_1 = 2$

E.3.4.4.

$$\omega_1 + \omega_2 = 1 \quad (3.85)$$

$$x_1\omega_1 + x_2\omega_2 = \frac{1}{2} \quad (3.86)$$

$$x_1^2\omega_1 + x_2^2\omega_2 = \frac{1}{3} \quad (3.87)$$

E.3.4.5. Dica: consulte o grau de exatidão da regra de Simpson.

E.3.5.1. a) $-2.61712\text{e}-1$; b) $2.55351\text{e}-1$; c) $8.97510\text{e}-2$; d) $1.27411\text{e}-1$; e) $1.21016\text{e}-1$.

E.3.5.2. a) $-1.54617\text{e}-1$; b) $-1.50216\text{e}-1$; c) $-1.47026\text{e}-1$; d) $-1.47190\text{e}-1$; e) $-1.47193\text{e}-1$.

E.3.5.3. a) $1.21016\text{e}-1$; b) $1.21744\text{e}-1$; c) $1.21744\text{e}-1$

E.3.5.4. $5.93738\text{e}-1$

E.3.6.1. a) $-2,84951\text{E}-01$; b) $2,66274\text{e}-01$; c) $1,49496\text{e}-01$; d) $1,60085\text{e}-01$; e) $1,59427\text{e}-01$.

E.3.6.2. a) $-1,03618\text{e}-1$; b) $-5,56446\text{e}-2$; c) $-4,19168\text{e}-2$

E.3.6.3. a) $-1,31347$; b) $-1,23313$; c) $-1,26007$

E.3.7.1. $1,2\text{e}-1$

E.4.1.1. $\tilde{y}(1.5) = 3.14159e-1$, $e(1, h) = 3.1E - 01$

E.4.1.2. $h = 10^{-6}$, $\tilde{y}(1) = 1.8134e+0$

E.4.1.3. $-5.58858e-1$

E.4.1.4. $|\tilde{y}(1) - y(1)| = 3.4e+2$. Dica: verifique as hipóteses do Teorema 4.1.1.

	h	$\tilde{\mathbf{y}}(1)$	$\ \tilde{\mathbf{y}}(1) - \mathbf{y}(1)\ $
E.4.1.5.	10^{-1}	(2.387, 5.077)	$7.4e-1$
	10^{-2}	(2.500, 5.693)	$1.1e-1$
	10^{-3}	(2.520, 5.793)	$1.2e-2$
	10^{-4}	(2.523, 5.803)	$1.2e-3$
	10^{-5}	(2.523, 5.804)	$1.2e-4$
	10^{-6}	(2.523, 5.804)	$1.2e-5$

E.4.1.6. Dica: O PVI do Exemplo 4.1.4 é um problema rígido.

E.4.1.7. Dica: use o polinômio de Taylor de grau 2 de e^δ .

E.4.1.8. Dica: estude a demonstração do Teorema 4.1.1.

E.4.1.9. $h = \sqrt{2\delta/M}$. Dica: Encontre o mínimo de $E(h) := M/2 + \delta/h^2$.

	h	$\tilde{y}(1)$	$ \tilde{y}(1) - y(1) $
E.4.2.1.	$1e-1$	$-1.52293e-1$	$1.7e-3$
	$1e-2$	$-1.50602e-1$	$1.8e-5$
	$1e-3$	$-1.50585e-1$	$1.8e-7$
	$1e-4$	$-1.50584e-1$	$1.8e-9$

E.4.2.2. Dica: o gráfico de $e(t; h = 10^{-1})$ tem a forma de uma função exponencial crescente.

E.4.2.3. $h = 10^{-2}$, $\tilde{y}(1) = -1.50584e-1$

E.4.2.4. Dica: $y(2) = -2.10171e-1$.

E.4.2.5. Dica: $y(2) = 2.90306e+3$.

E.4.3.1. a) $-6.00654e-1$; b) $-6.00703e-1$; c) $-5.99608e-1$

E.4.3.3. Dica: o gráfico de $e(t; h = 10^{-1})$ tem a forma de uma função exponencial crescente para todos os métodos de R-K.

E.4.3.5. Dica: $y(2) = -2.10171e-1$.

E.4.3.6. Dica: $y(3) = 2.90306e+3$.

E.4.4.1. Dica.

$$y^{(k+1)} = y^{(k)} + hf(t^{(k+1)}, y^{(k+1)}) \quad (4.164a)$$

$$y^{(k+1)} = y^{(k)} + h(y^{(k+1)} + 1) \quad (4.164b)$$

$$y^{(k+1)} = \frac{y^{(k)} + h}{1 - h}. \quad (4.164c)$$

E.4.4.2. Dica: consulte a condição de estabilidade (4.160).

E.4.4.4. $\tilde{y}(1.5) = 6.65400e-1$, $\varepsilon = 6.7E - 01$

E.4.4.7.

$$y(t^{(k)}) = y_0 e^{\lambda t^{(k)}} \quad (4.169)$$

$$= y_0 \lim_{h \rightarrow 0^+} (1 - h\lambda)^{-t^{(k)}/h} \quad (4.170)$$

$$= y_0 \lim_{h \rightarrow 0^+} (1 - h\lambda)^{-k} \quad (4.171)$$

$$= \lim_{h \rightarrow 0^+} y^{(k)}. \quad (4.172)$$

E.4.5.1. a) $-6.00696e-1$; b) $-5.96694e-1$; c) $-5.96161e-1$

E.4.5.2. Dica: solução analítica é $y(t) = \frac{1}{2} \cos(t) - \frac{1}{2} \sin(t)$.

a) Inicialização pelo Método do Ponto Médio, $\tau = O(h^2)$; b) Inicialização pelo Método de RK-4, $\tau = O(h^4)$.

E.4.5.3. Dica: use um Método de R-K com $O(h^p)$, $p \geq 3$, como inicializador.

E.4.5.4. Dica: solução analítica é $y(t) = e^t - \frac{1}{2} \sin(t) - \frac{1}{2} \cos(t)$.

E.4.5.5. Dica: $\tau = O(h^4)$.

E.4.6.1. $-5.99240e-1$

	h	$\ \tilde{u} - u\ _{L^2}$
E.5.1.1.	10^{-1}	$3.9e-3$
	10^{-2}	$1.2e-4$
	10^{-3}	$3.9e-6$
	10^{-4}	$1.2e-7$

E.5.1.2. $\varepsilon_{\text{abs}} = 3.1e-14$.

E.5.1.3. $2,7e-1$

E.5.1.4. b) resultado mais preciso.

E.5.2.3. $7,2e-1$

E.5.3.3. $7,2e-1$

E.5.4.1.

h	ε_h
10^{-1}	$4.0e-03$
10^{-2}	$1.3e-04$
10^{-3}	$4.0e-06$
10^{-4}	$1.3e-07$

E.5.4.2. b) tem melhor taxa de convergência.

E.6.3.1. $h_t = 2.5e-3$, $h_x = 1.e-2$

E.6.3.4. Dica: use, por exemplo, um método de [R-K-2](#).

E.6.3.5. Dica: consulte a Observação [6.3.1](#).

Bibliografia

- [1] Björk, A.. Numerical Methods for Least Squares Problems. SIAM, 1996.
- [2] Burden, R.L.; Faires, J.D.; Burden, A.M.. Análise Numérica. 3. ed., Cengage Learning, 2016. ISBN: 978-8522123414. Acesso [SABI+UFRGS](#).
- [3] Isaacson, E.; Keller H.B.. Analysis of Numerical Methods. Dover, 1994.
- [4] Lemire, D.. Number Parsing at a Gigabyte per Second. Software: Practice and Experience, 51(8), 2021, 1700-1727. DOI: [10.1002/spe.2984](#).
- [5] Nocedal, J.; Wright, S.J.. Numerical Optimization. Springer, 2006.
- [6] Press, W.H.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P.. Numerical Recipes. 3. ed., Cambridge University Press, 2007.
- [7] Ralston, A.; Rabinowitz, P.. A First Course in Numerical Analysis. 2. ed., Dover: New York, 2021. ISBN 048641454X.
- [8] Stoer, J.; Bulirsch, R.. Introduction to numerical analysis. 2. ed., Springer-Verlag, 1993.
- [9] Wood, W.L.. An exact solution for Burger's equation. Commun. Numer. Meth. Engng 2006; 22: 797–798. DOI: [10.1002/cnm.850](#).