

Matemática numérica

Pedro H A Konzen

3 de junho de 2018

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

Nestas notas de aula são abordados temas introdutórios de matemática numérica. Como ferramenta computacional de apoio didático, faço uso de códigos em **GNU Octave** (compatíveis com **MATLAB**). Ressalto que os códigos apresentados são implementações ingênuas com intuito de ressaltar os aspectos fundamentais dos métodos numéricos discutidos no texto.

Agradeço aos(às) estudantes que assiduamente ou esporadicamente contribuem com correções, sugestões e críticas em prol do desenvolvimento deste material didático.

Pedro H A Konzen

Sumário

| | |
|--|-----------|
| Capa | i |
| Licença | ii |
| Prefácio | iii |
| Sumário | v |
| 1 Introdução | 1 |
| 2 Aproximação por mínimos quadrados | 2 |
| 2.1 Problemas lineares | 2 |
| 2.1.1 Método das equações normais | 3 |
| 2.2 Problemas não lineares | 10 |
| 2.2.1 Método de Gauss-Newton | 14 |
| 2.2.2 Método de Levenberg-Marquardt | 16 |
| 3 Derivação | 19 |
| 3.1 Derivadas de primeira ordem | 19 |
| 3.1.1 Desenvolvimento por polinômio de Taylor | 21 |
| 3.2 Derivadas de segunda ordem | 26 |
| 3.3 Diferenças finitas por polinômios interpoladores | 29 |
| 3.3.1 Fórmulas de dois pontos | 29 |
| 3.3.2 Fórmulas de cinco pontos | 31 |
| 4 Técnicas de extrapolação | 34 |
| 4.1 Extrapolação de Richardson | 34 |
| 4.1.1 Sucessivas extrapolações | 38 |
| 4.1.2 Exercícios | 41 |

| | | |
|----------|--|-----------|
| 5 | Integração | 43 |
| 5.1 | Regras de Newton-Cotes | 43 |
| 5.1.1 | Regras de Newton-Cotes fechadas | 44 |
| 5.1.2 | Regras de Newton-Cotes abertas | 48 |
| 5.2 | Regras compostas de Newton-Cotes | 50 |
| 5.2.1 | Regra composta do ponto médio | 50 |
| 5.2.2 | Regra composta do trapézio | 52 |
| 5.2.3 | Regra composta de Simpson | 53 |
| 5.3 | Quadratura de Romberg | 56 |
| | Respostas dos Exercícios | 59 |
| | Referências Bibliográficas | 61 |
| | Índice Remissivo | 62 |

Capítulo 1

Introdução

Em construção ...

Capítulo 2

Aproximação por mínimos quadrados

2.1 Problemas lineares

Dado um conjunto de n pontos $\{(x_i, y_i)\}_{i=1}^n$, $x_i \neq x_j$ para $i \neq j$, e uma família de $m \leq n$ funções $\{f_i(x)\}_{i=1}^m$, o problema linear de aproximação por mínimos quadrados consiste em determinar os m coeficientes $\{c_i\}_{i=1}^m$ tal que a função

$$f(x; c) = \sum_{j=1}^m c_j f_j(x) \quad (2.1)$$

$$= c_1 f_1(x) + c_2 f_2(x) + c_3 f_3(x) + \cdots + c_m f_m(x) \quad (2.2)$$

aproxime o conjunto de pontos dados no sentido de mínimos quadrados, i.e. o vetor dos coeficientes $c = (c_1, c_2, \dots, c_m)$ é solução do seguinte problema linear de minimização

$$\min_c \left\{ E := \sum_{i=1}^n (y_i - f(x_i; c))^2 \right\}. \quad (2.3)$$

A fim de trabalharmos com uma notação mais compacta, definimos o resíduo $r(c) = (r_1(c), r_2(c), \dots, r_n(c))$, onde $r_i(c) := y_i - f(x_i)$ e $c = (c_1, c_2, \dots, c_m)$. Com esta notação, o problema de mínimos quadrados se resume a resolver

$$\min_c \{ E := \|r(c)\|_2^2 \}. \quad (2.4)$$

2.1.1 Método das equações normais

A fim de resolver o problema de mínimos quadrados (2.4), observamos que o erro quadrático

$$E = \|r(c)\|_2^2 \quad (2.5)$$

$$= \sum_{i=1}^n r_i(c)^2 \quad (2.6)$$

$$= \sum_{i=1}^n (y_i - f(x_i; c))^2 \quad (2.7)$$

$$= \sum_{i=1}^n \left(y_i - \sum_{j=1}^m c_j f_j(x_i) \right)^2 \quad (2.8)$$

$$= \|y - Ac\|_2^2, \quad (2.9)$$

onde $y = (y_1, y_2, \dots, y_n)$ e

$$A := \begin{bmatrix} f_1(x_1) & f_2(x_1) & \cdots & f_m(x_1) \\ f_1(x_2) & f_2(x_2) & \cdots & f_m(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ f_1(x_n) & f_2(x_n) & \cdots & f_m(x_n) \end{bmatrix}. \quad (2.10)$$

Os parâmetros c_j que minimizam o erro E são solução do seguinte sistema de equações

$$\frac{\partial E}{\partial c_j} = 2 \sum_{i=1}^n r_i(c) \frac{\partial}{\partial c_j} r_i(c) = 0, \quad (2.11)$$

onde $j = 1, 2, \dots, m$. Ou, em uma notação mais apropriada,

$$\nabla_c E = 0 \Leftrightarrow A^T r(c) = 0 \quad (2.12)$$

$$\Leftrightarrow A^T (y - Ac) = 0 \quad (2.13)$$

$$\Leftrightarrow A^T Ac = A^T y. \quad (2.14)$$

Portanto, o problema linear de mínimos quadrados se resume em resolver as chamadas **equações normais**

$$A^T Ac = A^T y. \quad (2.15)$$

Logo, o problema linear de mínimos quadrados (2.4) reduz-se a resolver o sistema linear (2.15) para c . Isto nos leva a questão de verificar se $A^T A$ é invertível. De sorte, da disciplina de álgebra linear temos o seguinte teorema.

Teorema 2.1.1. *A matriz $A^T A$ é positiva definida se, e somente se, as colunas de A são linearmente independentes (i.e. $\text{posto}(A) = n$).*

Demonstração. Se as colunas de A são linearmente independentes, então $x \neq 0$ implica $Ax \neq 0$ e, equivalentemente, $x^T A^T \neq 0$. Portanto, $x \neq 0$ implica $x^T A^T A x = \|Ax\|_2^2 > 0$, o que mostra que $A^T A$ é positiva definida.

Suponhamos, agora, que as colunas de A não são linearmente independentes. Então, existe $x_0 \neq 0$ tal que $Ax_0 = 0$. Mas, então, $x_0^T A^T A x_0 = 0$, o que mostra que $A^T A$ não é positiva definida. \square

Este teorema nos fornece uma condição suficiente para a existência (e unicidade) de solução do problema linear de mínimos quadrados. Mais especificamente, se as colunas da matriz A são linearmente independentes, então os coeficientes da função $f(x)$ que melhor ajustam os pontos dados são

$$c = (A^T A)^{-1} A^T y. \quad (2.16)$$

Exemplo 2.1.1. (Ajuste de polinômios) Considere o problema de ajustar o conjunto de pontos

| i | 1 | 2 | 3 | 4 |
|-------|-----|------|-----|-----|
| x_i | -1 | 0 | 1 | 1,5 |
| y_i | 1,2 | -0,1 | 0,7 | 2,4 |

por um polinômio quadrático da forma

$$p(x) = p_1 x^2 + p_2 x + p_n \quad (2.17)$$

no sentido de mínimos quadrados.

Neste caso, a família de funções do problema de mínimos quadrados é $f_1(x) = x^2$, $f_2(x) = x$ e $f_3(x) = 1$. Assim sendo, os coeficientes $p = (p_1, p_2, p_3)$ são solução do seguinte sistema linear

$$A^T A p = A^T y, \quad (2.18)$$

onde $y = (y_1, y_2, y_3)$ e

$$A := \begin{bmatrix} x_1^2 & x_1 & 1 \\ x_2^2 & x_2 & 1 \\ x_3^2 & x_3 & 1 \\ x_4^2 & x_4 & 1 \end{bmatrix}. \quad (2.19)$$



Figura 2.1: Esboço do polinômio ajustado no Exemplo 2.1.1.

Emfim, resolvendo as equações normais (2.18), obtemos

$$p(x) = 1,25x^2 - 0,188x - 0,203. \quad (2.20)$$

A Figura 2.2 mostra um esboço dos pontos (em vermelho) e do polinômio ajustado (em azul).

Os coeficientes e um esboço do polinômio ajustado podem ser obtidos no GNU Octave com o seguinte código:

```
#pontos
x = [-1 0 1 1.5]';
y = [1.2, -0.1, 0.7, 2.4]';

#resol. as eqs. normais
A = [x.^2 x.^1 x.^0];
```

```

p = inv(A'*A)*A'*y

#esboco do pol. ajustado
xx = linspace(-1.25,1.75);
plot(x,y,'ro',...
      xx,polyval(p,xx));grid

```

Exemplo 2.1.2. (Ajuste de curvas) Consideremos o mesmo conjunto de pontos do exemplo anterior (Exemplo 2.1.1). Aqui, vamos ajustar uma curva da forma

$$f(x) = c_1 \sin(x) + c_2 \cos(x) + c_3 \quad (2.21)$$

no sentido de mínimos quadrados. Para tanto, formamos a matriz

$$A := \begin{bmatrix} \sin(x_1) & \cos(x_1) & 1 \\ \sin(x_2) & \cos(x_2) & 1 \\ \sin(x_3) & \cos(x_3) & 1 \\ \sin(x_4) & \cos(x_4) & 1 \end{bmatrix} \quad (2.22)$$

e, então, resolvemos as equações normais $A^T A c = A^T y$ para o vetor de coeficientes $c = (c_1, c_2)$. Fazendo isso, obtemos $c_1 = -0,198$, $c_2 = -2,906$ e $c_3 = 2,662$. A Figura ?? mostra um esboço da curva ajustada (linha azul) aos pontos dados (círculos vermelhos).

Os coeficientes e um esboço do polinômio ajustado podem ser obtidos no GNU Octave com o seguinte código:

```

#pontos
x = [-1 0 1 1.5]';
y = [1.2, -0.1, 0.7, 2.4]';

#resol. as eqs. normais
A = [sin(x) cos(x) ones(4,1)];
c = inv(A'*A)*A'*y

#curva ajustada
f = @(x) c(1)*sin(x) + c(2)*cos(x) + c(3)

#esboco da fun. ajustada
xx = linspace(-1.25,1.75);
plot(x,y,'ro',...
      xx,f(xx));grid

```



Figura 2.2: Esboço da curva ajustada no Exemplo 2.1.2.

Exemplo 2.1.3. (Um problema não linear) Consideremos o problema de ajustar, no sentido de mínimos quadrados, à função

$$f(x) = c_1 e^{c_2 x} \quad (2.23)$$

ao seguinte conjunto de pontos

| i | 1 | 2 | 3 | 4 |
|-------|-----|-----|-----|-----|
| x_i | -1 | 0 | 1 | 1,5 |
| y_i | 8,0 | 1,5 | 0,2 | 0,1 |

Aqui, temos um problema não linear de mínimos quadrados que pode ser transformado em um problema linear fazendo-se

$$y = c_1 e^{c_2 x} \Rightarrow \ln y = \ln c_1 e^{c_2 x} \quad (2.24)$$

$$\Rightarrow \ln y = \ln c_1 + c_2 x. \quad (2.25)$$

Isto é, denotando $d_1 := \ln c_1$ e $d_2 := c_2$, o problema se resume a ajustar uma reta $r(x) = d_1 + d_2x$ ao conjunto de pontos $\{(x_i, \ln y_i)\}_{i=1}^4$.

Para resolver o problema transformado, formamos a matriz

$$A := \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \quad (2.26)$$

e, então, resolvemos as equações normais $A^T A d = A^T \ln y$, com $\ln y = (\ln y_1, \ln y_2, \ln y_3, \ln y_4)$, donde obtemos $d_1 = 0,315$ e $d_2 = -1,792$. Das definições de d_1 e d_2 , temos $c_2 = d_2 = -1,792$ e $c_1 = e^{d_1} = 1,371$. A Figura 2.3 mostra um esboço da curva $f(x) = c_1 e^{c_2 x}$ ajustada (linha azul) aos pontos dados (círculos vermelhos).

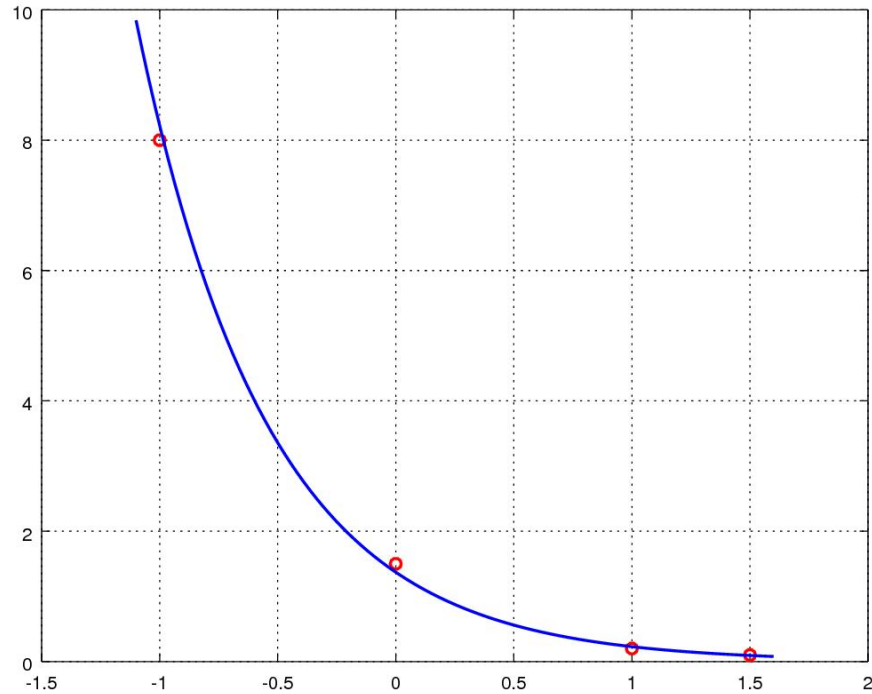


Figura 2.3: Esboço da curva ajustada no Exemplo 2.1.3.

O ajuste e um esboço da função ajustada podem ser feitos no GNU Octave com o seguinte código:

```
#pontos
x = [-1 0 1 1.5]';
y = [8.0 1.5 0.2 0.1]';

#resol. as eqs. normais
A = [ones(4,1) x];
d = inv(A'*A)*A'*log(y)

#fun. ajustada
c = [exp(d(1)); d(2)]
f = @(x) c(1)*exp(c(2)*x);

#esboco da fun. ajustada
xx = linspace(-1.1,1.6);
plot(x,y,'ro','linewidth',1.5,...
      xx,f(xx),'b-','linewidth',1.5);grid
```

Exercícios

E 2.1.1. Determine a reta $y = c_1x + c_2$ que melhor se ajusta, no sentido de mínimos quadrados, aos pontos

| i | 1 | 2 | 3 | 4 | 5 |
|-------|------|------|------|------|------|
| x_i | -2,5 | -1,3 | 0,2 | 1,7 | 2,3 |
| y_i | 3,8 | 1,5 | -0,7 | -1,5 | -3,2 |

Por fim, compute a norma L^2 do resíduo, i.e. $\|r(c)\|_2 = \|y - (c_1x + c_2)\|_2$ para os pontos dados.

E 2.1.2. Determine o polinômio $y = c_1x^3 + c_2x^2 + c_3x + c_4$ que melhor se ajusta, no sentido de mínimos quadrados, aos pontos

| i | 1 | 2 | 3 | 4 | 5 |
|-------|------|------|-----|-----|------|
| x_i | -2,5 | -1,3 | 0,2 | 1,7 | 2,3 |
| y_i | 3,8 | 0,5 | 2,7 | 1,2 | -1,3 |

Por fim, compute a norma L^2 do resíduo, i.e. $\|r(c)\|_2$.

E 2.1.3. Determine a curva $y = c_1 \sin x + c_2 \cos x + c_3$ que melhor se ajusta, no sentido de mínimos quadrados, aos pontos

| i | 1 | 2 | 3 | 4 | 5 |
|-------|------|------|-----|-----|------|
| x_i | -2,5 | -1,3 | 0,2 | 1,7 | 2,3 |
| y_i | 3,8 | 0,5 | 2,7 | 1,2 | -1,3 |

Por fim, compute a norma L^2 do resíduo, i.e. $\|r(c)\|_2$.

E 2.1.4. Use a transformação $z = \ln y$ para ajustar, no sentido de mínimos quadrados, a curva $y = c_1 e^{c_2(x-c_3)^2}$ aos pontos

| i | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|------|-----|-----|-----|-----|-----|
| x_i | -0,5 | 0,5 | 1,3 | 2,1 | 2,7 | 3,1 |
| y_i | 0,1 | 1,2 | 2,7 | 0,9 | 0,2 | 0,1 |

2.2 Problemas não lineares

Um problema não linear de mínimos quadrados consiste em ajustar uma dada função $f(x; c)$ que dependa não linearmente dos parâmetros $c = (c_1, c_2, \dots, c_m)$, $m \geq 1$, a um dado conjunto de $n \geq m$ pontos $\{(x_i, y_i)\}_{i=1}^n$. Mais especificamente, buscamos resolver o seguinte problema de minimização

$$\min_{\{c_1, c_2, \dots, c_m\}} \left[E := \sum_{i=1}^n (y_i - f(x_i; c))^2 \right]. \quad (2.27)$$

Aqui, denotaremos por $r(c) := (r_1(c), r_2(c), \dots, r_n(c))$ o vetor dos resíduos $r_i(c) := y_i - f(x_i, c)$. Com isso, o problema se resume a encontrar o vetor de parâmetros c que minimiza

$$E = \|r(c)\|_2^2. \quad (2.28)$$

Tais parâmetros são solução do seguinte sistema de equações

$$\frac{\partial E}{\partial c_j} = 2 \sum_{i=1}^n r_i(c) \frac{\partial}{\partial c_j} r_i(c) = 0 \quad (2.29)$$

ou, equivalentemente, da equação

$$\nabla E = 0 \Leftrightarrow J_R^T(c) r(c) = 0, \quad (2.30)$$

onde

$$J_R(c) := \begin{bmatrix} \frac{\partial r_1}{\partial c_1} & \frac{\partial r_1}{\partial c_2} & \cdots & \frac{\partial r_1}{\partial c_m} \\ \frac{\partial r_2}{\partial c_1} & \frac{\partial r_2}{\partial c_2} & \cdots & \frac{\partial r_2}{\partial c_m} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial r_n}{\partial c_1} & \frac{\partial r_n}{\partial c_2} & \cdots & \frac{\partial r_n}{\partial c_m} \end{bmatrix} \quad (2.31)$$

é a jacobiana do resíduo r em relação aos parâmetros c .

Podemos usar o método de Newton para resolver (2.30). Para tanto, escolhemos uma aproximação inicial para $c^{(1)} = (c_1^{(1)}, c_2^{(1)}, \dots, c_m^{(1)})$ e iteramos

$$H_R(c^{(k)})\delta^{(k)} = -J_R^T(c)r(c) \quad (2.32)$$

$$c^{(k+1)} = c^{(k)} + \delta^{(k)}, \quad (2.33)$$

onde $\delta^{(k)} = (\delta_1^{(k)}, \delta_2^{(k)}, \delta_m^{(k)})$ é a atualização de Newton (ou direção de busca) e $H_R(c) := [h_{p,q}(c)]_{p,q=1}^{m,m}$ é a matriz hessiana, cujos elementos são

$$h_{p,q} := \sum_{i=1}^n \left\{ \frac{\partial r_i}{\partial c_q} \frac{\partial r_i}{\partial c_p} + r_i \frac{\partial^2 r_i}{\partial c_q \partial c_p} \right\}. \quad (2.34)$$

Exemplo 2.2.1. Consideremos o problema de ajustar, no sentido de mínimos quadrados, a função

$$f(x; c) = c_1 e^{c_2 x} \quad (2.35)$$

ao seguinte conjunto de pontos

| i | 1 | 2 | 3 | 4 |
|-------|-----|-----|-----|-----|
| x_i | -1 | 0 | 1 | 1,5 |
| y_i | 8,0 | 1,5 | 0,2 | 0,1 |

Aqui, vamos utilizar a iteração de Newton para o problema de mínimos quadrados, i.e. a iteração dada em (2.32)-(2.33). Para tanto, para cada $i = 1, 2, 3, 4$, precisamos das seguintes derivadas parciais do resíduo $r_i(c) :=$

$$y_i - c_1 e^{c_2 x_i}:$$

$$\frac{\partial}{\partial c_1} r_i(c) = -e^{c_2 x_i}, \quad (2.36)$$

$$\frac{\partial}{\partial c_2} r_i(c) = -c_1 x_i e^{c_2 x_i}, \quad (2.37)$$

$$\frac{\partial^2}{\partial c_1^2} r_i(c) = 0, \quad (2.38)$$

$$\frac{\partial^2}{\partial c_1 \partial c_2} r_i(c) = \frac{\partial^2}{\partial c_2 \partial c_1} r_i(c) = -x_i e^{c_2 x_i}, \quad (2.39)$$

$$\frac{\partial^2}{\partial c_2^2} r_i(c) = -c_1 x_i^2 e^{c_2 x_i}. \quad (2.40)$$

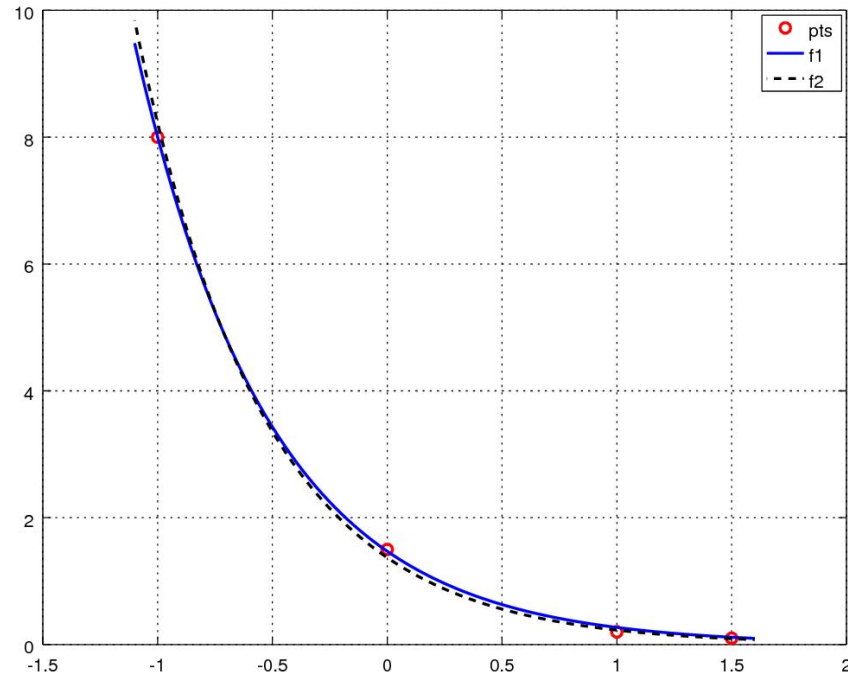


Figura 2.4: Esboço da curva ajustada no Exemplo 2.2.1.

Com isso e tomando $c^{(1)} = (1,4, -1,8)$ (motivado do Exemplo 2.1.3), computamos as iterações de Newton (2.32)-(2.33). Iterando até a precisão de

$TOL = 10^{-4}$, obtemos a solução $c_1 = 1,471$ e $c_2 = -1,6938$. Na Figura 2.4 vemos uma comparação entre a curva aqui ajustada (—) e aquela obtida no Exemplo 2.1.3 (---).

O ajuste discutido neste exemplo pode ser computado no GNU Octave com o seguinte código:

```
#pontos
global x = [-1 0 1 1.5]';
global y = [8.0 1.5 0.2 0.1]';

#fun. objetivo
f = @(x,c) c(1)*exp(c(2)*x);

#residuo
r = @(c) y - f(x,c);

#jacobiana
function A = J(c)
    global x
    A = zeros(4,2);
    A(:,1) = - exp(c(2)*x);
    A(:,2) = - c(1)*x.*exp(c(2)*x);
endfunction

#hessiana
function A = H(c)
    global x
    global y
    A = zeros(2,2);
    A = J(c)'*J(c);
    for i=1:4
        A(1,1) += 0;
        A(1,2) += (y(i) - c(1)*exp(c(2)*x(i))) * ...
            (- x(i)*exp(c(2)*x(i)));
        A(2,1) += (y(i) - c(1)*exp(c(2)*x(i))) * ...
            (- x(i)*exp(c(2)*x(i)));
        A(2,2) += (y(i) - c(1)*exp(c(2)*x(i))) * ...
            (- c(1)*x(i)^2*exp(c(2)*x(i)));
    end
endfunction
```

```

    endfor
endfunction

#aprox. inicial
c = [1.4 -1.8]';

#iteracoes de Newton
k=0;
do
    k+=1;
    delta = - inv(H(c))*J(c)'*r(c);
    c = c + delta;
    [k,c',norm(delta)]
until ((k>10) | (norm(delta)<1e-4))

```

Observamos que a solução obtida no exemplo anterior (Exemplo 2.2.1) difere da previamente encontrada no Exemplo 2.1.3. Naquele exemplo, os parâmetros obtidos nos fornecem $E = 6,8\text{E}-2$, enquanto que a solução do exemplo anterior fornece $E = 6,1\text{E}-3$. Isto é esperado, pois naquele exemplo resolvemos um problema aproximado, enquanto no exemplo anterior resolvemos o problema por si.

O emprego do método de Newton para o problema de mínimos quadrados tem a vantagem da taxa de convergência quadrática, entretanto requer a computação das derivadas parciais de segunda ordem do resíduo. Na sequência discutimos alternativas comumente empregadas.

2.2.1 Método de Gauss-Newton

O método de Gauss-Newton é uma técnica iterativa que aproxima o problema não linear de mínimos quadrados (2.27) por uma sequência de problemas lineares. Para seu desenvolvimento, começamos de uma aproximação inicial $c^{(1)} = (c_1^{(1)}, c_2^{(1)}, \dots, c_m^{(1)})$ dos parâmetros que queremos ajustar. Também, assumindo que a n -ésima iterada $c^{(k)}$ é conhecida, faremos uso da aproximação de primeira ordem de $f(x, c)$ por polinômio de Taylor, i.e.

$$f(x; c^{(k+1)}) \approx f(x; c^{(k)}) + \nabla_c f(x; c^{(k)})(c^{(k+1)} - c^{(k)}), \quad (2.41)$$

onde

$$\nabla_c f(x; c) = \left[\frac{\partial}{\partial c_1} f(x; c) \quad \frac{\partial}{\partial c_2} f(x; c) \quad \cdots \quad \frac{\partial}{\partial c_m} f(x; c) \right]. \quad (2.42)$$

O método consiste em obter a solução do problema não linear (2.27) pelo limite dos seguintes problemas lineares de mínimos quadrados

$$\min_{\delta^{(k)}} \left[\tilde{E} := \sum_{i=1}^n (y_i - f(x_i, c^{(k)}) - \nabla_c f(x_i; c^{(k)}) \delta^{(k)})^2 \right] \quad (2.43)$$

$$c^{(k+1)} = c^{(k)} + \delta^{(k)}. \quad (2.44)$$

Agora, usando a notação de resíduo $r(c) = y - f(x; c)$, observamos que (2.50) consiste no problema linear de mínimos quadrados

$$\min_{\delta^{(k)}} \|r(c^{(k)}) + J_R(c^{(k)}) \delta^{(k)}\|_2^2, \quad (2.45)$$

o qual é equivalente a resolver as equações normais

$$J_R^T(c^{(n)}) J_R(c^{(n)}) \delta^{(n)} = -J_R^T(c) r(c). \quad (2.46)$$

Com isso, dada uma aproximação inicial $c^{(1)}$, a **iteração do método de Gauss-Newton** consiste em

$$J_R^T(c^{(k)}) J_R(c^{(k)}) \delta^{(k)} = -J_R^T(c) r(c) \quad (2.47)$$

$$c^{(k+1)} = c^{(k)} + \delta^{(k)}. \quad (2.48)$$

Exemplo 2.2.2. A aplicação da iteração de Gauss-Newton ao problema de mínimos quadrados discutido no Exemplo 2.2.1 nos fornece a mesma solução obtida naquele exemplo (preservadas a aproximação inicial e a tolerância de precisão).

A implementação do método de Gauss-Newton para este problema no GNU Octave pode ser feita com o seguinte código:

```
#pontos
global x = [-1 0 1 1.5]';
y = [8.0 1.5 0.2 0.1]';

#fun. objetivo
f = @(x,c) c(1)*exp(c(2)*x);

#residuo
r = @(c) y - f(x,c);
```

```

#jacobiana
function A = J(c)
    global x
    A = zeros(4,2);
    A(:,1) = - exp(c(2)*x);
    A(:,2) = - c(1)*x.*exp(c(2)*x);
endfunction

#aprox. inicial
c = [1.4 -1.8]';

#iteracoes de Gauss-Newton
k=0;
do
    k+=1;
    delta = - inv(J(c)'*J(c))*J(c)'\*r(c);
    c = c + delta;
    [k,c',norm(delta)]
until ((k>10) | (norm(delta)<1e-4))

```

O método de Gauss-Newton pode ser lentamente convergente para problemas muito não lineares ou com resíduos grandes. Nesse caso, métodos de Gauss-Newton com amortecimento são alternativas robustas [1, 4]. Na sequência, introduziremos um destes métodos, conhecido como método de Levenberg-Marquardt.

2.2.2 Método de Levenberg-Marquardt

O método de Levenberg-Marquardt é uma variação do método de Gauss-Newton no qual a direção de busca $\delta^{(n)}$ é obtida da solução do seguinte problema regularizado

$$\min_{\delta^{(k)}} \{ \|r(c^{(k)}) + J_R(c^{(k)})\delta^{(k)}\|_2^2 + \mu^{(k)}\|\delta^{(k)}\|_2^2 \} \quad (2.49)$$

ou, equivalentemente,

$$\min_{\delta^{(k)}} \left\| \begin{bmatrix} r(c^{(k)}) \\ 0 \end{bmatrix} + \begin{bmatrix} J_R(c^{(k)}) \\ \mu^{(k)} I \end{bmatrix} \delta^{(k)} \right\|_2^2 \quad (2.50)$$

A taxa de convergência das iterações de Levenberg-Marquardt é sensível a escolha do parâmetro $\mu^{(k)} \geq 0$. Aqui, faremos esta escolha por tentativa e erro. O leitor pode aprofundar-se mais sobre esta questão na literatura especializada (veja, por exemplo, [1, 4]).

Observação 2.2.1. Quando $\mu^{(k)} \equiv 0$ para todo n , o método de Levenberg-Marquardt é equivalente ao método de Gauss-Newton.

Exemplo 2.2.3. Consideremos o problema de mínimos quadrados discutido no Exemplo 2.2.1. O método de Gauss-Newton falha para este problema se escolhermos, por exemplo, $c^{(1)} = (0, 0)$. Isto ocorre pois, para esta escolha de $c^{(1)}$, a jacobiana $J(c^{(1)})$ não tem posto completo. Entretanto, o método de Levenberg-Marquardt com $\mu^{(k)} = 0,1$ é convergente, mesmo para esta escolha de $c^{(1)}$.

A implementação no GNU Octave do método de Levenberg-Marquardt (com $\mu^{(k)} = 0,1$ constante) para este problema pode ser feita com o seguinte código:

```
#pontos
global x = [-1 0 1 1.5]';
y = [8.0 1.5 0.2 0.1]';

#fun. objetivo
f = @(x,c) c(1)*exp(c(2)*x);

#residuo
r = @(c) y - f(x,c);

#jacobiana
function A = JR(c)
    global x;
    A = zeros(4,2);
    A(:,1) = - exp(c(2)*x);
    A(:,2) = - c(1)*x.*exp(c(2)*x);
endfunction

#aprox. inicial
c = [0 0]';

#param. de amortecimento
```

```

mu = 0.1;

#iteracoes de Gauss-Newton
k=0;
do
    k+=1;
    JJ = [JR(c);mu*eye(2,2)];
    delta = - inv(JJ'*JJ)*JJ'*[r(c);zeros(2,1)];
    c = c + delta;
    printf("%d %1.1e %1.3e %1.3e\n", k,norm(delta),c')
until ((k>10) | (norm(delta)<1e-4))

```

Exercícios

E 2.2.1. Use o método de Gauss-Newton para ajustar, no sentido de mínimos quadrados e com precisão de 10^{-4} , a curva $y = c_1 e^{c_2(x-c_3)^2}$ aos pontos

| i | 1 | 2 | 3 | 4 | 5 | 6 |
|-------|------|-----|-----|-----|-----|-----|
| x_i | -0,5 | 0,5 | 1,3 | 2,1 | 2,7 | 3,1 |
| y_i | 0,1 | 1,2 | 2,7 | 0,9 | 0,2 | 0,1 |

Use as condições iniciais:

- a) $c_1 = 2,1$, $c_2 = -1$ e $c_3 = 1,3$.
- b) $c_1 = 1$, $c_2 = -1$ e $c_3 = -1$.

E 2.2.2. Resolva o exercício anterior (Exercício 2.2.1) usando o método de Levenberg-Marquardt com amortecimento constante $\mu = 0,2$.

Capítulo 3

Derivação

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

3.1 Derivadas de primeira ordem

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

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}. \quad (3.1)$$

Assim sendo e assumindo $h > 0$ ¹ próximo de zero, temos que $f'(x)$ pode ser aproximada pela razão fundamental, i.e.

$$f'(x) \approx \underbrace{\frac{f(x+h) - f(x)}{h}}_{D_h f(x)}. \quad (3.2)$$

Analisando a Figura 3.1 vemos que, 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 $(x+h, f(x+h))$.

Exemplo 3.1.1. 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 razão fundamental (3.2),

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



Figura 3.1: Interpretação geométrica da aproximação da derivada pela razão fundamental. Veja no [Geogebra](#).

temos

$$f' \left(\frac{\pi}{3} \right) \approx D_h f(x) = \frac{f \left(\frac{\pi}{3} + h \right) - f \left(\frac{\pi}{3} \right)}{h} \quad (3.3)$$

$$= \frac{\text{sen} \left(\frac{\pi}{3} \right) - \text{sen} \left(\frac{\pi}{3} \right)}{h}. \quad (3.4)$$

Na Tabela 3.1 temos os valores desta aproximação para diferentes escolhas da passo h .

| 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^{-10} | 5.00000E-1 |

Tabela 3.1: Valores aproximados da derivada de $f(x) = \text{sen}(x)$ no ponto $x = \pi/6$ usado a expressão (3.2).

No GNU Octave, podemos fazer estes cálculos com o seguinte código:

```
f = @(x) sin(x);
Df = @(x,h) (f(x+h)-f(x))/h;
x=pi/3;
h=1e-1;
printf('%1.5e\n',Df(x,h))
```

A aproximação (3.2) é uma **fórmula de diferenças finitas**. Existem várias aproximações deste tipo que podem ser derivadas. Além disso, tais derivações nos permitem estimar o erro na utilização de tais fórmulas para a aproximação de derivadas. Na sequência, discutiremos o desenvolvimento de fórmulas de diferenças finitas usando polinômios de Taylor.

3.1.1 Desenvolvimento por polinômio de Taylor

Aqui, discutimos a obtenção de fórmulas de diferenças finitas via polinômio de Taylor.

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

$$f(x+h) = f(x) + hf'(x) + O(h^2). \quad (3.5)$$

Agora, isolando $f'(x)$, obtemos

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h). \quad (3.6)$$

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

$$D_{+,h}f(x) := \frac{f(x+h) - f(x)}{h}. \quad (3.7)$$

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

Exemplo 3.1.2. 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

$$f'\left(\frac{\pi}{3}\right) \approx D_{+,h}f(x) = \frac{f\left(\frac{\pi}{3} + h\right) - f\left(\frac{\pi}{3}\right)}{h} \quad (3.8)$$

$$= \frac{\sin\left(\frac{\pi}{3} + h\right) - \sin\left(\frac{\pi}{3}\right)}{h}. \quad (3.9)$$

Na Tabela 3.2 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 3.2: Resultados referente ao Exemplo 3.1.2.

| 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 |

No GNU Octave, podemos fazer estes cálculos com o seguinte código:

```

f = @(x) sin(x);
Df = @(x,h) (f(x+h)-f(x))/h;
x=pi/3;
h=1e-1;
printf('%1.5e %1.1e\n',Df(x,h),abs(cos(x)-Df(x,h)))

```

Observação 3.1.1. No exemplo acima (Exemplo 3.1.2), 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 (veja, Tabela 3.2). 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 do de truncamento. Isto se deve a dominância dos erros de arredondamento para valores muito pequenos de h .

Para mais informações sobre o comportamento do erro de arredondamento em fórmulas de diferenças finitas, veja, por exemplo, [REAMAT - Cálculo Numérico - Versão GNU Octave - Diferenças Finitas - Erro de arredondamento](#).

Diferenças finitas regressiva de ordem h

Substituindo h por $-h$ na equação (3.5), obtemos

$$f(x - h) = f(x) - hf'(x) + O(h^2), \quad (3.10)$$

donde obtemos a fórmula de diferenças finitas regressiva de ordem h

$$D_{-,h}f(x) = \frac{f(x) - f(x - h)}{h}. \quad (3.11)$$

Exemplo 3.1.3. 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

$$f'\left(\frac{\pi}{3}\right) \approx D_{-,h}f(x) = \frac{f\left(\frac{\pi}{3}\right) - f\left(\frac{\pi}{3} - h\right)}{h} \quad (3.12)$$

$$= \frac{\sin\left(\frac{\pi}{3}\right) - \sin\left(\frac{\pi}{3} - h\right)}{h}. \quad (3.13)$$

Na Tabela 3.3 temos os valores desta aproximação para diferentes escolhas de h , bem como, o erro absoluto da aproximação de $f'(\pi/3)$ por $D_{-,h}f(\pi/3)$. No GNU Octave, podemos fazer estes cálculos com o seguinte código:

Tabela 3.3: Resultados referente ao Exemplo 3.1.3.

| 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 |

```
f = @(x) sin(x);
Df = @(x,h) (f(x)-f(x-h))/h;
x=pi/3;
h=1e-1;
printf('%1.5E %1.1E\n',Df(x,h),abs(cos(x)-Df(x,h)))
```

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 , obtemos

$$f(x+h) = f(x) + hf'(x) + \frac{h}{2}f''(x) + O(h^3) \quad (3.14)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h}{2}f''(x) + O(h^3). \quad (3.15)$$

Então, subtraindo esta segunda equação da primeira, temos

$$f(x+h) - f(x-h) = 2hf'(x) + O(h^3). \quad (3.16)$$

Agora, isolando $f'(x)$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2), \quad (3.17)$$

o que nos fornece a chamada fórmula de diferenças finitas central de ordem h^2

$$D_{0,h^2}f(x) := \frac{f(x+h) - f(x-h)}{2h}. \quad (3.18)$$

Exemplo 3.1.4. 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

$$f' \left(\frac{\pi}{3} \right) \approx D_{0,h^2} f(x) = \frac{f \left(\frac{\pi}{3} + h \right) - f \left(\frac{\pi}{3} - h \right)}{2h} \quad (3.19)$$

$$= \frac{\sin \left(\frac{\pi}{3} + h \right) - \sin \left(\frac{\pi}{3} - h \right)}{2h}. \quad (3.20)$$

Tabela 3.4: Resultados referente ao Exemplo 3.1.4.

| 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 |

Na Tabela 3.4 temos os valores desta aproximação para diferentes escolhas de h , bem como, o erro absoluto da aproximação de $f'(\pi/3)$ por $D_{0,h^2} f(\pi/3)$. No GNU Octave, podemos fazer estes cálculos com o seguinte código:

```
f = @(x) sin(x);
Df = @(x,h) (f(x+h)-f(x-h))/(2*h);
x=pi/3;
h=1e-1;
printf('%1.5E %1.1E\n',Df(x,h),abs(cos(x)-Df(x,h)))
```

Exercícios

E 3.1.1. Calcule aproximações da derivada de

$$f(x) = \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} + x \quad (3.21)$$

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 3.1.2. 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 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.

3.2 Derivadas de segunda ordem

Diferentemente do que é costumeiro 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 de primeira ordem. Na sequência, desenvolveremos e aplicaremos uma fórmula de diferenças finitas central para a aproximação de derivadas de segunda ordem.

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

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(x) + O(h^4), \quad (3.22)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f'''(x) + O(h^4). \quad (3.23)$$

$$(3.24)$$

Somando estas duas equações, obtemos

$$f(x+h) + f(x-h) = 2f(x) + h^2f''(x) + O(h^4). \quad (3.25)$$

Então, isolando $f''(x)$ temos

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + O(h^2). \quad (3.26)$$

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

$$D_{0,h^2}^2 f(x) := \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (3.27)$$

Exemplo 3.2.1. Consideremos 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.

$$f''(x) \approx D_{0,h^2} D_{0,h^2} f(x) = \frac{D_{0,h^2} f(x+h) - D_{0,h^2} f(x-h)}{2h} \quad (3.28)$$

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

$$f''(x) \approx D_{0,h^2}^2 f(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2}. \quad (3.29)$$

Tabela 3.5: Resultados referente ao Exemplo 3.2.1. Notação: $\delta_{DD} := |f''(\pi/6) - D_{0,h^2} D_{0,h^2} f(\pi/6)|$ e $\delta_{D^2} := |f''(\pi/6) - D_{0,h^2}^2 f(\pi/6)|$.

| h | $D_{0,h^2} D_{0,h^2} f(\pi/6)$ | δ_{DD} | $D_{0,h^2}^2 f(\pi/6)$ | 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 |

Na Tabela 3.5 temos os valores computados em ambos os casos e seus respectivos erros absolutos para diversas escolhas de h . Observamos que a aplicação 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 (3.28) mostra que

$$D_{0,h^2} D_{0,h^2} f(x) = \underbrace{\frac{f(x+2h) - 2f(x) + f(x-2h)}{4h^2}}_{D_{0,(2h)^2}^2 f(x)}. \quad (3.30)$$

No GNU Octave, podemos fazer estes cálculos com o seguinte código:


```

f = @(x) sin(x) + x^2;
Df = @(x,h) (f(x+h)-f(x-h))/(2*h);
DDf = @(x,h) (Df(x+h,h)-Df(x-h,h))/(2*h);
D2f = @(x,h) (f(x+h) - 2*f(x) + f(x-h))/(h^2);
x=pi/6;
h=1e-1;
printf('%1.5E %1.1E %1.5E %1.1E\n',...
    DDf(x,h),abs(1.5-DDf(x,h)),...
    D2f(x,h),abs(1.5-D2f(x,h)))

```

Exercícios

E 3.2.1. Use a fórmula de diferenças finitas central de ordem h^2 para computar aproximações da segunda derivada de

$$f(x) = \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} + x \quad (3.31)$$

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 3.2.2. 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 .

3.3 Diferenças finitas por polinômios interpoladores

Aqui, discutimos a obtenção de 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

$$f(x) = p(x) + R_{n+1}(x), \quad (3.32)$$

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

$$R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=1}^{n+1} (x - x_j). \quad (3.33)$$

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 3.3.1. *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*

$$R_{n+1}^{(k)} = \frac{f^{(n+1)}(\eta)}{(n+1-k)!} \prod_{j=1}^{n+1-k} (x - \xi_j), \quad (3.34)$$

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]. □

3.3.1 Fórmulas de dois pontos

Seja $p(x)$ 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.

$$f(x) = p(x) + R_2(x) \quad (3.35)$$

$$= f(x_1) \frac{x - x_2}{x_1 - x_2} + f(x_2) \frac{x - x_1}{x_2 - x_1} + R_2(x). \quad (3.36)$$

Denotando $h = x_2 - x_1$, temos

$$f(x) = f(x_1) \frac{x - x_2}{-h} + f(x_2) \frac{x - x_1}{h} + R_2(x). \quad (3.37)$$

e, derivando com respeito a x

$$f'(x) = \frac{f(x_2) - f(x_1)}{h} + R_2^{(1)}(x), \quad (3.38)$$

onde $R_2^{(1)}(x)$ é dado conforme o teorema 3.3.1.

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

$$f(x) = \underbrace{\frac{f(x+h) - f(x)}{h}}_{D_{+,h}f(x)} + O(h). \quad (3.39)$$

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

$$f(x) = \underbrace{\frac{f(x) - f(x-h)}{h}}_{D_{-,h}f(x)} + O(h). \quad (3.40)$$

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.

$$f(x) = f(x_1) \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} \quad (3.41)$$

$$+ f(x_2) \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} \quad (3.42)$$

$$+ f(x_3) \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} + R_3(x). \quad (3.43)$$

Derivando em relação a x , obtemos

$$f'(x) = f(x_1) \frac{(x_2 - x_3)(2x - x_2 - x_3)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} \quad (3.44)$$

$$+ f(x_2) \frac{(x_1 - x_3)(-2x + x_1 + x_3)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} \quad (3.45)$$

$$+ f(x_3) \frac{(x_1 - x_2)(2x - x_1 - x_2)}{(x_1 - x_2)(x_1 - x_3)(x_2 - x_3)} + R_3^{(1)}(x). \quad (3.46)$$

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

$$D_{+,h_1,h_2}f(x) = \frac{1}{h_1h_2(h_1+h_2)} (-h_2(2h_1+h_2)f(x) \quad (3.47)$$

$$+ (h_1+h_2)^2 f(x+h_1) \quad (3.48)$$

$$- h_1^2 f(x+h_1+h_2)). \quad (3.49)$$

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

$$D_{+,h^2}f(x) = \frac{1}{2h} [-3f(x) + 4f(x+h) - f(x+2h)]. \quad (3.50)$$

Escolhendo $x = x_2$, $x_1 = x - h$ e $x_3 = x + h$ na equação (3.44), obtemos a **fórmula de diferenças finitas central de ordem h^2**

$$D_{0,h^2} = \frac{1}{2h} [f(x+h) - f(x-h)]. \quad (3.51)$$

Por fim, escolhendo $x = x_3$, $x_1 = x - 2h$ e $x_2 = x - h$ na equação (3.44), obtemos a **fórmula de diferenças finitas regressiva de ordem h^2**

$$D_{-,h^2} = \frac{1}{2h} [3f(x) - 4f(x-h) + f(x-2h)]. \quad (3.52)$$

3.3.2 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

$$f(x) = \sum_{i=1}^5 f(x_i) \left(\prod_{j=1, j \neq i}^5 \frac{x - x_j}{x_i - x_j} \right) + R_5(x). \quad (3.53)$$

Calculando a derivada em relação a x , temos

$$f'(x) = \sum_{i=1}^5 f(x_i) \left(\sum_{\substack{j=1 \\ j \neq i}}^5 \prod_{\substack{k=1 \\ k \neq i, k \neq j}}^5 \frac{x - x_k}{x_i - x_k} \right) + R_5^{(1)}(x). \quad (3.54)$$

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

$$D_{+,h^4}f(x) := \frac{1}{12h} [f(x - 2h) - 8f(x - h) + 8f(x + h) - f(x + 2h)]. \quad (3.55)$$

Exercícios

E 3.3.1. Use a fórmula de diferenças finitas central de ordem h^4 para computar a aproximação da derivada de

$$f(x) = \frac{\sin(x + 2) - e^{-x^2}}{x^2 + \ln(x + 2)} + x \quad (3.56)$$

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

E 3.3.2. Obtenha as seguintes fórmulas de diferenças finitas de 5 pontos com passo h constante e com:

- 4 pontos para frente.
- 1 ponto para traz e 3 pontos para frente.
- 2 pontos para traz e 2 pontos para frente.
- 3 pontos para traz e 1 pontos para frente.
- 4 pontos para traz.

E 3.3.3. 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 3.3.2). Para tanto, dê preferência para fórmulas centrais sempre que possível.

Capítulo 4

Técnicas de extrapolação

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

4.1 Extrapolação de Richardson

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

$$I = F_1(h) + \underbrace{k_1h + k_2h^2 + k_3h^3 + O(h^4)}_{\text{erro de truncamento}}. \quad (4.1)$$

Então, dividindo h por 2, obtemos

$$I = F_1\left(\frac{h}{2}\right) + k_1\frac{h}{2} + k_2\frac{h^2}{4} + k_3\frac{h^3}{8} + O(h^4). \quad (4.2)$$

Agora, de forma a eliminarmos o termo de ordem h das expressões acima, subtraímos (4.1) de 2 vezes (4.2), o que nos leva a

$$I = \underbrace{\left[F_1\left(\frac{h}{2}\right) + \left(F_1\left(\frac{h}{2}\right) - F_1(h) \right) \right]}_{F_2(h)} - k_2\frac{h^2}{2} - k_3\frac{3h^3}{4} + O(h^4). \quad (4.3)$$

Ou seja, denotando

$$F_2(h) := F_1\left(\frac{h}{2}\right) + \left(F_1\left(\frac{h}{2}\right) - F_1(h) \right) \quad (4.4)$$

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.*

$$I = F_2\left(\frac{h}{2}\right) - k_2 \frac{h^2}{8} - k_2 \frac{3h^3}{32} + O(h^4) \quad (4.5)$$

Então, subtraindo (4.3) de 4 vezes (4.5) de, obtemos

$$I = \underbrace{\left[3F_2\left(\frac{h}{2}\right) + \left(F_2\left(\frac{h}{2}\right) - F_2(h) \right) \right]}_{F_3(h)} + k_3 \frac{3h^3}{8} + O(h^4). \quad (4.6)$$

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

$$F_3(h) = F_2\left(\frac{h}{2}\right) + \frac{F_2\left(\frac{h}{2}\right) - F_2(h)}{3}, \quad (4.7)$$

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.*

$$I = F_3\left(\frac{h}{2}\right) + k_3 \frac{3h^3}{64} + O(h^4). \quad (4.8)$$

E, então, subtraindo (4.6) de 8 vezes (4.8), temos

$$I = \underbrace{\left[F_3\left(\frac{h}{2}\right) + \left(\frac{F_3\left(\frac{h}{2}\right) - F_3(h)}{7} \right) \right]}_{F_4(h)} + O(h^4). \quad (4.9)$$

Ou seja,

$$F_4(h) = \left[F_3\left(\frac{h}{2}\right) + \frac{F_3\left(\frac{h}{2}\right) - F_3(h)}{7} \right] \quad (4.10)$$

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

Teorema 4.1.1. *Seja $F_1(h)$ uma aproximação de I com erro de truncamento da forma*

$$I - F_1(h) = \sum_{i=1}^n k_1 h^i + O(h^{n+1}). \quad (4.11)$$

Então, para $j \geq 2$,

$$F_j(h) := F_{j-1}\left(\frac{h}{2}\right) + \frac{F_{j-1}\left(\frac{h}{2}\right) - F_{j-1}(h)}{2^{j-1} - 1} \quad (4.12)$$

é uma aproximação de I com erro de truncamento da forma

$$\begin{aligned} I - F_j(h) &= \sum_{i=j}^n (-1)^{j-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{j-2} (2^{i-l-1} - 1)}{2^{(j-1)(i-j+1)} d_j} k_i h^i \\ &\quad + O(h^{n+1}), \end{aligned} \quad (4.13)$$

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 (4.3). Assumimos, agora, que vale

$$\begin{aligned} I - F_j(h) &= (-1)^{j-1} \frac{(2^{j-1} - 1) \prod_{l=1}^{j-2} (2^{j-l-1} - 1)}{2^{(j-1)} d_j} k_j h^j \\ &\quad + \sum_{i=j+1}^n (-1)^{j-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{j-2} (2^{i-l-1} - 1)}{2^{(j-1)(i-j+1)} d_j} k_i h^i \\ &\quad + O(h^{n+1}). \end{aligned} \quad (4.14)$$

para $j \geq 2$. Então, tomamos

$$\begin{aligned} I - F_j\left(\frac{h}{2}\right) &= (-1)^{j-1} \frac{(2^{j-1} - 1) \prod_{l=1}^{j-2} (2^{j-l-1} - 1)}{2^{(j-1)} d_j} k_j \frac{h^j}{2^j} \\ &\quad + \sum_{i=j+1}^n (-1)^{j-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{j-2} (2^{i-l-1} - 1)}{2^{(j-1)(i-j+1)} d_j} k_i \frac{h^i}{2^i} \\ &\quad + O(h^{n+1}). \end{aligned} \quad (4.15)$$

Agora, subtraímos (4.14) de 2^j vezes (4.15), o que nos fornece

$$\begin{aligned} I &= \left[F_j \left(\frac{h}{2} \right) + \frac{F_j \left(\frac{h}{2} \right) - F_j(h)}{2^j - 1} \right] \\ &+ \sum_{i=j+1}^n (-1)^{(j+1)-1} \frac{(2^{i-1} - 1) \prod_{l=1}^{(j+1)-2} (2^{i-l-1} - 1)}{2^{((j+1)-1)(i-(j+1)+1)} 2^{j-1} d_j} k_i h^i \\ &+ O(h^{n+1}). \end{aligned} \quad (4.16)$$

□

Corolário 4.1.1. *Seja $F_1(h)$ uma aproximação de I com erro de truncamento da forma*

$$I - F_1(h) = \sum_{i=1}^n k_i h^{2i} + O(h^{2n+2}). \quad (4.17)$$

Então, para $j \geq 2$,

$$F_j(h) := F_{j-1} \left(\frac{h}{2} \right) + \frac{F_{j-1} \left(\frac{h}{2} \right) - F_{j-1}(h)}{4^{j-1} - 1} \quad (4.18)$$

é uma aproximação de I com erro de truncamento da forma

$$\begin{aligned} I - F_j(h) &= \sum_{i=j}^n (-1)^{j-1} \frac{(4^{i-1} - 1) \prod_{l=1}^{j-2} (4^{i-l-1} - 1)}{4^{(j-1)(i-j+1)} d_j} k_i h^{2i} \\ &+ O(h^{n+1}), \end{aligned} \quad (4.19)$$

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 4.1.1. □

Exemplo 4.1.1. Dada uma função $f(x)$, consideremos sua aproximação por diferenças finitas progressiva de ordem h , i.e.

$$\begin{aligned} \underbrace{f'(x)} I &= \underbrace{\frac{f(x+h) - f(x)}{h}}_{F_1(h)} \\ &+ \frac{f''(x)}{2} h + \frac{f'''(x)}{6} h^2 + O(h^3). \end{aligned} \quad (4.20)$$

Estão, considerando a primeira extrapolação de Richardson, temos

$$F_2(h) = F_1\left(\frac{h}{2}\right) + \left(F_1\left(\frac{h}{2}\right) - F_1(h)\right) \quad (4.21)$$

$$= 4 \frac{f(x + h/2) - f(x)}{h} - \frac{f(x + h) - f(x)}{h} \quad (4.22)$$

$$= \frac{-f(x + h) + 4f(x + h/2) - 3f(x)}{h}, \quad (4.23)$$

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 (3.50)).

Exemplo 4.1.2. Dada uma função $f(x)$, consideremos sua aproximação por diferenças finitas central de ordem h^2 , i.e.

$$\underbrace{f'(x)} I = \underbrace{\frac{f(x + h) - f(x - h)}{2h}}_{F_1(h)} - \frac{f'''(x)}{6} h^2 - \frac{f^{(5)}(x)}{120} h^4 + O(h^6). \quad (4.24)$$

Estão, considerando a primeira extrapolação de Richardson, temos

$$F_2(h) = F_1\left(\frac{h}{2}\right) + \frac{\left(F_1\left(\frac{h}{2}\right) - F_1(h)\right)}{3} \quad (4.25)$$

$$= \frac{1}{6h} [f(x - h) - 8f(x - h/2) + 8f(x + h/2) - f(x + h)] \quad (4.26)$$

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 (3.55)).

4.1.1 Sucessivas extrapolações

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

$$I - F_1(h) = k_1 h + k_2 h^2 + k_3 h^3 + \dots + k_n h^n + O(h^{n+1}). \quad (4.27)$$

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

$$T = \begin{bmatrix} F_1(h) & & & & \\ F_1(h/2) & F_2(h) & & & \\ F_1(h/2^2) & F_2(h/2) & F_3(h) & & \\ \vdots & \vdots & \vdots & & \\ F_1(h/2^n) & F_2(h/2^{n-1}) & F_3(h/2^{n-2}) & \dots & F_n(h) \end{bmatrix} \quad (4.28)$$

Desta forma, temos que

$$F_j\left(\frac{h}{2^{i-1}}\right) = t_{i,j-1} + \frac{t_{i,j-1} - t_{i-1,j-1}}{2^{j-1} - 1} \quad (4.29)$$

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 4.1.3. 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

$$\begin{aligned} f'\left(\frac{\pi}{3}\right) &= \underbrace{\frac{f\left(\frac{\pi}{3} + h\right) - f\left(\frac{\pi}{3}\right)}{h}}_{F_1(h) := D_{+,h}f(\pi/3)} \\ &\quad + \frac{f''(x)}{2}h + \frac{f'''(x)}{6}h^2 + \dots \end{aligned} \quad (4.30)$$

Na Tabela 4.1 temos os valores das aproximações de $f'(\pi/3)$ computadas via sucessivas extrapolações de Richardson a partir de (4.30) com $h = 0.1$.

Tabela 4.1: Resultados referente ao Exemplo 4.1.3.

| $O(h)$ | $O(h^2)$ | $O(h^3)$ | $O(h^4)$ |
|------------|------------|------------|------------|
| 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 |

No GNU Octave, podemos fazer estes cálculos com o seguinte código:

```

#funcao
f = @(x) sin(x);
x=pi/3;

#aproximacao de ordem 1
dfp = @(x,h) (f(x+h)-f(x))/h;
h=0.1;

#tabela c/ sucessivas extrapolacoes
T=zeros(4,4);
for i=1:4
    T(i,1) = dfp(x,h/2^(i-1));
endfor
for j=2:4
    for i=j:4
        T(i,j) = T(i,j-1) ...
            + (T(i,j-1)-T(i-1,j-1))/(2^(j-1)-1);
    endfor
endfor

```

Exemplo 4.1.4. 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

$$\begin{aligned}
 f' \left(\frac{\pi}{3} \right) &= \underbrace{\frac{f \left(\frac{\pi}{3} + h \right) - f \left(\frac{\pi}{3} - h \right)}{2h}}_{F_1(h) := D_{0,h^2} f(\pi/3)} \\
 &\quad - \frac{f'''(x)}{6} h^2 + \frac{f^{(5)}(x)}{120} h^4 - \dots
 \end{aligned} \tag{4.31}$$

Na Tabela 4.2 temos os valores das aproximações de $f'(\pi/3)$ computadas via sucessivas extrapolações de Richardson a partir de (4.31) com $h = 1$. No GNU Octave, podemos fazer estes cálculos com o seguinte código:

```

#funcao obj.
f = @(x) sin(x);
x=pi/3;

#aprox. O(h^2)

```

Tabela 4.2: Resultados referente ao Exemplo 4.1.4.

| $O(h^2)$ | $O(h^4)$ | $O(h^6)$ | $O(h^8)$ |
|------------|------------|------------|------------|
| 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 |

```

h=1;
dfp = @(x,h) (f(x+h)-f(x-h))/(2*h);

#tabela c/ sucessivas extrapolacoes
T=zeros(4,4);
for i=1:4
    T(i,1) = dfp(x,h/2^(i-1));
endfor
for j=2:4
    for i=j:4
        T(i,j) = T(i,j-1) ...
            + (T(i,j-1)-T(i-1,j-1))/(4^(j-1)-1);
    endfor
endfor

```

4.1.2 Exercícios

E 4.1.1. Mostre que a primeira extrapolação de Richardson de

$$D_{-,h}f(x) = \frac{f(x) - f(x-h)}{h} \quad (4.32)$$

é igual a

$$D_{-, (h/2)^2}f(x) = \frac{3f(x) - 4f(x-h) + f(x-2h)}{h}. \quad (4.33)$$

E 4.1.2. Considere o problema de aproximar a derivada de

$$f(x) = \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} + x \quad (4.34)$$

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 5

Integração

Neste capítulo, discutimos os métodos numéricos fundamentais para a aproximação de integrais definidas de funções. Tais métodos são chamados de **quadraturas numéricas** e têm a forma

$$\int_a^b f(x) dx \approx \sum_{i=1}^n f(x_i)w_i, \quad (5.1)$$

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

5.1 Regras de Newton-Cotes

Dada uma função $f(x)$ e um intervalo $[a, b]$, denotamos por

$$I := \int_a^b f(x) dx. \quad (5.2)$$

a integral de $f(x)$ no intervalo $[a, b]$. A ideia das regras de Newton-Cotes é aproximar I pela integral de um polinômio interpolador de $f(x)$ por pontos previamente selecionados.

Seja, então, $p(x)$ 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

$$f(x) = p(x) + R_{n+1}(x), \quad (5.3)$$

onde

$$p(x) = \sum_{i=1}^{n+1} f(x_i) \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{(x - x_j)}{x_i - x_j} \quad (5.4)$$

e

$$R_{n+1}(x) = \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{j=1}^{n+1} (x - x_j), \quad (5.5)$$

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

$$I := \int_a^b f(x) \quad (5.6)$$

$$= \int_a^b p(x) dx + \int_a^b R_{n+1}(x) dx \quad (5.7)$$

$$= \underbrace{\sum_{i=1}^{n+1} f(x_i) \int_a^b \prod_{\substack{j=1 \\ j \neq i}}^{n+1} \frac{(x - x_j)}{x_i - x_j} dx}_{\text{quadratura}} + \underbrace{\int_a^b R_{n+1}(x) dx}_{\text{erro de truncamento}} \quad (5.8)$$

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, abordaremos as regras de Newton-Cotes mais usuais e estimaremos o erro de truncamento caso a caso. Para uma abordagem mais geral, recomenda-se consultar [3, Ch 7., Sec. 1.1].

5.1.1 Regras de Newton-Cotes fechadas

As regras de Newton-Cotes fechadas são aqueles que a quadratura incluem os extremos do intervalo de integração, i.e. os nodos extremos são $x_1 = a$ e $x_{n+1} = b$.

Regra do trapézio

A regra do trapézio é obtida tomando-se os nodos $x_1 = a$ e $x_2 = b$. Então, denotando $h := b - a$ ¹, os pesos da quadratura são:

$$w_1 = \int_a^b \frac{x - b}{a - b} dx \quad (5.9)$$

$$= \frac{(b - a)}{2} = \frac{h}{2} \quad (5.10)$$

e

$$w_2 = \int_a^b \frac{x - a}{b - a} dx \quad (5.11)$$

$$= \frac{(b - a)}{2} = \frac{h}{2}. \quad (5.12)$$

Agora, estimamos o erro de truncamento com

$$E := \int_a^b R_2(x) dx \quad (5.13)$$

$$= \int_a^b \frac{f''(\xi(x))}{2} (x - a)(x - b) dx \quad (5.14)$$

$$\leq C \left| \int_a^b (x - a)(x - b) dx \right| \quad (5.15)$$

$$= C \frac{(b - a)^3}{6} = O(h^3). \quad (5.16)$$

Portanto, a **regra do trapézio** é dada por

$$\int_a^b f(x) dx = \frac{h}{2}(f(a) + f(b)) + O(h^3). \quad (5.17)$$

Exemplo 5.1.1. Consideremos o problema de computar a integral de $f(x) = xe^{-x^2}$ no intervalo $[0, 1/4]$. Analiticamente, temos

$$I = \int_0^{1/4} xe^{-x^2} dx = -\frac{e^{-x^2}}{2} \Big|_0^{1/4} \quad (5.18)$$

$$= \frac{1 - e^{-1/4}}{2} = 3,02935\text{E}-2. \quad (5.19)$$

¹Neste capítulo, h é escolhido como a distância entre os nodos.

Agora, usando a regra do trapézio, obtemos a seguinte aproximação para I

$$I \approx \frac{h}{2}(f(0) + f(1/2)) \quad (5.20)$$

$$= \frac{1/4}{2} \left(0 + \frac{1}{4} e^{-(1/4)^2} \right) = 2,93567\text{E}-2. \quad (5.21)$$

Podemos obter a aproximação dada pela regra do trapézio no GNU Octave com o seguinte código:

```
f = @(x) x*exp(-x^2);
a=0;
b=0.25;
h=b-a;
Itrap = (h/2)*(f(a)+f(b));
printf("%1.5E\n",Itrap)
```

Regra de Simpson

A regra de Simpson é obtida escolhendo-se os nodos $x_1 = a$, $x_2 = (a + b)/2$ e $x_3 = b$. Com isso e denotando $h = (b - a)/2$, calculamos os seguintes pesos:

$$w_1 = \int_a^b \frac{(x - x_2)(x - x_3)}{(x_1 - x_2)(x_1 - x_3)} dx \quad (5.22)$$

$$= \frac{(b - a)}{6} = \frac{h}{6}, \quad (5.23)$$

$$w_2 = \int_a^b \frac{(x - x_1)(x - x_3)}{(x_2 - x_1)(x_2 - x_3)} dx \quad (5.24)$$

$$= 4 \frac{(b - a)}{6} = 4 \frac{h}{6} \quad (5.25)$$

e

$$w_3 = \int_a^b \frac{(x - x_1)(x - x_2)}{(x_3 - x_1)(x_3 - x_2)} dx \quad (5.26)$$

$$= \frac{(b - a)}{6} = \frac{h}{6}. \quad (5.27)$$

Isto nos fornece a chamada **regra de Simpson**

$$I \approx \frac{h}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (5.28)$$

Nos resta estimar o erro de truncamento da regra de Simpson. Para tanto, consideramos a expansão em polinômio de Taylor de grau 3 de $f(x)$ em torno do ponto x_2 , i.e.

$$\begin{aligned} f(x) &= f(x_2) + f'(x_2)(x - x_2) + \frac{f''(x_2)}{2}(x - x_2)^2 \\ &\quad + \frac{f'''(x_2)}{6}(x - x_2)^3 \\ &\quad + \frac{f^{(4)}(\xi_1(x))}{24}(x - x_2)^4, \end{aligned} \quad (5.29)$$

donde

$$\begin{aligned} \int_a^b f(x) dx &= 2hf(x_2) + \frac{h^3}{3}f''(x_2) \\ &\quad + \frac{1}{24} \int_a^b f^{(4)}(\xi_1(x))(x - x_2)^4 dx. \end{aligned} \quad (5.30)$$

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

$$f''(x_2) = \frac{f(x_1) - 2f(x_2) + f(x_3)}{h^2} + O(h^2). \quad (5.31)$$

Ainda, o último termo da equação (5.30) pode ser estimado por

$$\left| \frac{1}{24} \int_a^b f^{(4)}(\xi_1(x))(x - x_2)^4 dx \right| \leq C \left| \int_a^b (x - x_2)^4 dx \right| \quad (5.32)$$

$$= C(b - a)^5 = O(h^5). \quad (5.33)$$

Então, de (5.30), (5.31) e (5.1.1), temos

$$\int_a^b f(x) dx = \frac{h}{3} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] + O(h^5), \quad (5.34)$$

o que mostra que a **regra de Simpson tem erro de truncamento da ordem h^5** .

Exemplo 5.1.2. Aproximando a integral dada no Exemplo 5.1.1 pela a regra de Simpson, temos

$$\int_0^{1/4} f(x) dx \approx \frac{1/8}{3} \left[f(0) + 4f\left(\frac{1}{8}\right) + f\left(\frac{1}{4}\right) \right] \quad (5.35)$$

$$= \frac{1}{24} \left[\frac{1}{2} e^{-(1/8)^2} + \frac{1}{4} e^{-(1/4)^2} \right] \quad (5.36)$$

$$= 3,02959\text{E}-2. \quad (5.37)$$

Podemos computar a aproximação dada pela regra de Simpson no GNU Octave com o seguinte código:

```
f = @(x) x*exp(-x^2);
a=0;
b=1/4;
h=(b-a)/2;
Isimp = (h/3)*(f(a)+4*f((a+b)/2)+f(b));
printf("%1.5E\n",Isimp)
```

5.1.2 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

$$\int_a^b f(x) dx = \int_a^b f(x_1) dx + \int_a^b f'(\xi(x))(x - x_1) dx, \quad (5.38)$$

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

$$\int_a^b f(x), dx = hf\left(\frac{a+b}{2}\right) + O(h^3). \quad (5.39)$$

Deixa-se para o leitor a verificação do erro de truncamento (veja, Exercício 5.1.3).

Exemplo 5.1.3. Aproximando a integral dada no Exemplo 5.1.1 pela a regra do ponto médio, temos

$$\int_0^{1/4} f(x) dx \approx \frac{1}{4} f\left(\frac{1}{8}\right) \quad (5.40)$$

$$= \frac{1}{32} e^{-(1/8)^2} \quad (5.41)$$

$$= 3,07655\text{E}-2 \quad (5.42)$$

Podemos computar a aproximação dada pela regra do ponto médio no GNU Octave com o seguinte código:

```
f = @(x) x*exp(-x^2);
a=0;
b=0.25;
h=b-a;
Ipmd = h*f((a+b)/2);
printf("%1.5E\n",Ipmd)
```

Exercício

E 5.1.1. Aproxime

$$\int_{-1}^0 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (5.43)$$

usando a:

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

E 5.1.2. 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 |

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

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 5.1.3. 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 5.1.4. Obtenha a regra de Newton-Cotes aberta de 2 pontos e estime seu erro de truncamento.

5.2 Regras compostas de Newton-Cotes

Regras de integração numérica compostas (ou quadraturas compostas) são aquelas obtidas da composição de quadraturas aplicadas aos 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.

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{x_i}^{x_{i+1}} f(x) dx, \quad (5.46)$$

onde $a = x_1 < x_2 < \cdots < x_{n+1} = b$. 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.

5.2.1 Regra composta do ponto médio

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

$[\tilde{x}_i, \tilde{x}_{i+1}]$, temos

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} f(x) dx \quad (5.47)$$

$$= \sum_{i=1}^n \left[hf \left(\frac{\tilde{x}_i + \tilde{x}_{i+1}}{2} \right) + O(h^3) \right]. \quad (5.48)$$

Agora, observando que $h := (b - a)/n$ e escolhendo os nodos $x_i = a + (i - 1/2)h$, $i = 1, 2, \dots, n$, obtemos a **regra composta do ponto médio com n subintervalos**

$$\int_a^b f(x) dx = \sum_{i=1}^n hf(x_i) + O(h^2). \quad (5.49)$$

Exemplo 5.2.1. Consideremos 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

$$\underbrace{\int_a^b f(x) dx}_I \approx \underbrace{\sum_{i=1}^n hf(x_i)}_S, \quad (5.50)$$

onde $h = 1/(4n)$ e $x_i = (i - 1/2)h$, $i = 1, 2, \dots, n$. Na Tabela 5.1, temos as aproximações computadas com diversos números de subintervalos, bem como, seus erros absolutos.

Tabela 5.1: Resultados referentes ao Exemplo 5.2.1.

| n | S | $ I - S $ |
|------|------------|-----------|
| 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 |

Podemos fazer estas computações com o auxílio do seguinte código GNU Octave:

```
f = @(x) x*exp(-x^2);
a=0;
b=1;
n=10;
```



```

h=(b-a)/n;
s=0;
for i=1:n
    x=a+(i-1/2)*h;
    s+=h*f(x);
endfor
printf("%1.5E %1.1E\n",s,abs((1-e^(-1))/2-s))

```

5.2.2 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

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{x_i}^{x_{i+1}} f(x) dx \quad (5.51)$$

$$= \sum_{i=1}^n \left\{ \frac{h}{2} [f(x_i) + f(x_{i+1})] + O(h^3) \right\} \quad (5.52)$$

$$= f(x_1) \frac{h}{2} + \sum_{i=2}^n h f(x_i) + f(x_{n+1}) \frac{h}{2} + O(h^2). \quad (5.53)$$

Desta forma, a regra composto do trapézio com n subintervalos é

$$\int_a^b f(x) dx = \frac{h}{2} \left[f(x_1) + \sum_{i=2}^n 2f(x_i) + f(x_{n+1}) \right] + O(h^2), \quad (5.54)$$

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

Exemplo 5.2.2. Consideremos 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

$$\underbrace{\int_a^b f(x) dx}_I \approx \frac{h}{2} \underbrace{\left[f(x_1) + 2 \sum_{i=2}^{n-1} f(x_i) + f(x_{n+1}) \right]}_S, \quad (5.55)$$

onde $h = 1/(4n)$ e $x_i = (i - 1)h$, $i = 1, 2, \dots, n$. Na Tabela 5.2, temos as aproximações computadas com diversos números de subintervalos, bem como, seus erros absolutos.

Podemos fazer estas computações com o auxílio do seguinte código GNU Octave:

Tabela 5.2: Resultados referentes ao Exemplo 5.2.2.

| n | S | $ I - S $ |
|------|------------|-----------|
| 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 |

```
f = @(x) x*exp(-x^2);
a=0;
b=1;
n=1000;
h=(b-a)/n;
s=f(a);
for i=2:n
    x=a+(i-1)*h;
    s+=2*f(x);
endfor
s+=f(b);
s*=h/2;
printf("%1.5E %1.1E\n",s,abs((1-e^(-1))/2-s))
```

5.2.3 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

$$\int_a^b f(x) dx = \sum_{i=1}^n \int_{\tilde{x}_i}^{\tilde{x}_{i+1}} f(x) dx \quad (5.56)$$

$$= \sum_{i=1}^n \left\{ \frac{h}{3} \left[f(\tilde{x}_i) + 4f\left(\frac{\tilde{x}_i + \tilde{x}_{i+1}}{2}\right) + f(\tilde{x}_{i+1}) \right] + O(h^5) \right\}. \quad (5.57)$$

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

$$\int_a^b f(x) dx = \frac{h}{3} \left[f(x_1) + 2 \sum_{i=2}^n f(x_{2i-1}) + 4 \sum_{i=1}^n f(x_{2i}) + f(x_{n+1}) \right] + O(h^4) \quad (5.58)$$

Exemplo 5.2.3. Consideremos 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

$$\underbrace{\int_a^b f(x) dx}_I \approx \frac{h}{3} \underbrace{\left[f(x_1) + 2 \sum_{i=2}^n f(x_{2i-1}) + 4 \sum_{i=1}^n f(x_{2i}) + f(x_{n+1}) \right]}_S, \quad (5.59)$$

onde $h = 1/(8n)$ e $x_i = (i - 1)h$, $i = 1, 2, \dots, n$. Na Tabela 5.3, temos as aproximações computadas com diversos números de subintervalos, bem como, seus erros absolutos.

Tabela 5.3: Resultados referentes ao Exemplo 5.2.3.

| 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 |

Podemos fazer estas computações com o auxílio do seguinte código GNU Octave:

```
f = @(x) x*exp(-x^2);
a=0;
b=1;
n=1000;
h=(b-a)/(2*n);
s=f(a);
for i=2:n
    x=a+(2*i-2)*h;
    s+=2*f(x);
endfor
```

```

for i=1:n
    x=a+(2*i-1)*h;
    s+=4*f(x);
endfor
s+=f(b);
s*=h/3;
printf("%1.5E %1.1E\n",s,abs((1-e^(-1))/2-s))

```

Exercícios

E 5.2.1. Aproxime

$$\int_{-1}^0 \frac{\sin(x+2) - e^{-x^2}}{x^2 + \ln(x+2)} dx \quad (5.60)$$

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 5.2.2. 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 |

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.

5.3 Quadratura de Romberg

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

$$R_{k,1} := \frac{h_k}{2} \left[f(a) + 2 \sum_{i=2}^{2k} f(x_i) + f(b) \right] \quad (5.61)$$

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

$$I := \int_a^b f(x) dx. \quad (5.62)$$

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

$$I - R_{k,1} = \sum_{i=1}^{\infty} k_i h_k^{2i}, \quad (5.63)$$

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

$$R_{k,j} := R_{k,j-1} + \frac{R_{k,j-1} - R_{k-1,j-1}}{4^{j-1} - 1}, \quad (5.64)$$

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

Exemplo 5.3.1. 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

$$R_{1,1} := \frac{1}{2} [f(0) + f(1)] = 1,83940\text{E}-1 \quad (5.65)$$

$$R_{2,1} := \frac{1}{4} [f(0) + 2f(1/2) + f(1)] = 2,86670\text{E}-1. \quad (5.66)$$

Então, calculando

$$R_{2,2} = R_{2,1} + \frac{R_{2,1} - R_{1,1}}{3} = 3,20914\text{E}-1, \quad (5.67)$$

Tabela 5.4: Resultados referentes ao Exemplo 5.3.1.

| 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 |

a qual é a aproximação desejada.

Na Tabela 5.4, temos os valores de aproximações computadas pela quadratura de Romberg até ordem 8.

Podemos fazer estas computações com o auxílio do seguinte código GNU Octave:

```
#integral
f = @(x) x*exp(-x^2);
a=0;
b=1;

#ordem 2n
n=4;

R = zeros(n,n);
#R(k,1)
for k=1:n
    h = (b-a)/(2^(k-1));
    R(k,1) = f(a);
    for i=2:2^(k-1)
        x = a + (i-1)*h;
        R(k,1) += 2*f(x);
    endfor
    R(k,1) += f(b);
    R(k,1) *= h/2;
endfor
#extrapola
for j=2:n
    for k=j:n
        R(k,j) = R(k,j-1) + (R(k,j-1)-R(k-1,j-1))/(4^(j-1)-1);
    endfor
endfor
```

```
endfor
#sol.
for i = 1:n
    printf("%1.5E ",R(i,:))
    printf("\n")
end
```

Exercícios

Em construção ...

Resposta dos Exercícios

E 2.1.1. Código. $c_1 = -1,3259$, $c_2 = 8,66071\text{E}-2$, $\|r(c)\|_2 = 1,01390$.

E 2.1.2. Código. $c_1 = -4,50361\text{E}-1$, $c_2 = -2,78350\text{E}-1$, $c_3 = 1,46291$, $c_4 = 2,09648$, $\|r(c)\|_2 = 5,71346$

E 2.1.3. Código. $c_1 = -2,76842$, $c_2 = -7,17935\text{E}-1$, $c_3 = 1,37014\text{E}-1$, $\|r(c)\|_2 = 2,48880\text{E}+1$

E 2.1.4. Código. $c_1 = 2,10131\text{E}+0$, $c_2 = -9,73859\text{E}-1$, $c_3 = 1,25521\text{E}+0$

E 2.2.1. Código. a) $c_1 = 2,69971\text{E}+0$, $c_2 = -1,44723\text{E}+0$, $c_3 = 1,24333\text{E}+0$;
b) divergente.

E 2.2.2. Código. a) $c_1 = 2,69971\text{E}+0$, $c_2 = -1,44723\text{E}+0$, $c_3 = 1,24333\text{E}+0$;
b) $c_1 = 2,69971\text{E}+0$, $c_2 = -1,44723\text{E}+0$, $c_3 = 1,24333\text{E}+0$

E 3.1.1. Código. 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 3.1.2. Código.

| i | 1 | 2 | 3 | 4 | 5 | 6 |
|---------|--------|--------|--------|--------|--------|--------|
| dy/dx | 4,0E-1 | 7,5E-1 | 1,3E+0 | 1,1E+0 | 7,5E-1 | 8,0E-1 |

E 3.2.1. Código. 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 3.2.2. Código. 4,0;

E 3.3.1. Código. 1,05913

E 3.3.2. Código.

- 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 3.3.3. Código.

| 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 4.1.2. Código. a) 1,05919; b) 1,05916; c) 1,05913

E 5.1.1. Código. a) 3,33647E-1; b) 1,71368E-1; c) 2,79554E-1

E 5.1.2. Código. a) 4,02000E-1; b) 1,04250E+0; c) 8,08667E-1

E 5.1.3. Use um procedimento semelhante aquele usado para determinar a ordem do erro de truncamento da regra de Simpson.

E 5.1.4.

$$\int_a^b f(x) dx = \frac{3h}{2} \left[f\left(a + \frac{1}{3}(b-a)\right) \right. \quad (5.44)$$

$$\left. + f\left(a + \frac{2}{3}(b-a)\right) \right] + O(h^3), \quad h = \frac{(b-a)}{3} \quad (5.45)$$

E 5.2.1. Código. a) 2,69264E-1; b) 2,68282E-1; c) 2,68937E-1

E 5.2.2. Código. a) 8,12000E-1; b) 1,03850; c) 8,11667E-1

Referências Bibliográficas

- [1] A. Björk. *Numerical methods for least squares problems*. SIAM, 1996.
- [2] R.L. Burden, J.D. Faires, and A.M. Burden. *Análise Numérica*. CENGAGE Learning, 10. ed. edition, 2015.
- [3] E. Isaacson and H.B. Keller. *Analysis of numerical methods*. Dover, 1994.
- [4] J. Nocedal and S.J. Wright. *Numerical optimization*. Springer, 2006.

Índice Remissivo

- diferenças finitas, [21](#)
 - central de ordem h^2 , [24](#)
 - derivada segunda, [26](#)
 - progressiva de ordem h , [22](#)
 - regressiva de ordem h , [23](#)
- equações normais, [3](#)
- erro de
 - truncamento, [22](#)
- quadratura composta, [50](#)
- regra composta
 - de Simpson, [54](#)
 - do trapézio, [52](#)
- regra de Simpson, [47](#)
- regra do
 - ponto médio, [48](#)
 - trapézio, [45](#)