

Cálculo Numérico

Elias Salomão Helou Neto

8 de abril de 2022

LISTA DE FIGURAS

1.1.1	Ilustração da Proposição 1.1.2.	12
3.3.1	Polinômio interpolador de $\sin(x)$	62

LISTA DE ALGORITMOS

1.2.1	Bissecção	14
1.5.1	Método de Newton	22
1.6.1	Método da Secante	26
2.2.1	Substituição Reversa	41
2.3.1	Eliminação Gaussiana: Solução de Sistemas Lineares	44
2.3.2	Eliminação Gaussiana com Pivoteamento: Solução de Sistemas Lineares	45
2.5.1	Eliminação Gaussiana com Pivoteamento Parcial: Solução de Sistemas Lineares	53
2.7.1	Fatoração $PA = LU$	54

SUMÁRIO

1	RAÍZES DE FUNÇÕES	11
1.1	Introdução	11
1.2	Algoritmo de Bissecção	13
1.3	Ordem de Convergência	15
1.3.1	Ordem de Convergência da Bissecção	17
1.4	Método de Newton	18
1.5	Método de Newton	19
1.5.1	Convergência Quadrática do Método de Newton	21
1.6	Método da Secante	25
1.6.1	Sequências de Fibonacci	26
1.6.2	Convergência do Método da Secante	28
1.7	Exercícios	33
1.7.1	33
1.7.2	33
1.7.3	34
1.7.4	34
1.7.5	34

	1.7.6	35
2	SISTEMAS DE EQUAÇÕES LINEARES	37
2.1	Introdução	37
2.2	Solução de Sistemas Triangulares	40
2.3	Eliminação Gaussiana	42
2.4	Exercícios	46
	2.4.1	46
	2.4.2	47
	2.4.3	47
2.5	Eliminação Gaussiana com Pivoteamento Parcial	47
2.6	Exercícios	48
	2.6.1	48
	2.6.2	49
2.7	Fatoração $PA = LU$	49
2.8	Exercícios	51
	2.8.1	51
	2.8.2	51
	2.8.3	51
	2.8.4	51
3	INTERPOLAÇÃO POLINOMIAL	55
3.1	Introdução	55
3.2	O Polinômio Interpolador	56
3.3	Forma de Lagrange do Polinômio Interpolador	58
	3.3.1 Construção	58
	3.3.2 Avaliação	62
	3.3.3 Primeira Forma Baricêntrica	63

3.3.4	Segunda Forma Baricêntrica	65
3.3.5	Atualização	66
3.4	Forma de Newton do Polinômio Interpolador .	66
3.4.1	Operador de Diferenças Divididas . . .	66
3.4.2	Construção	68
3.4.3	Avaliação	70
3.4.4	Atualização	70
3.5	Erro na Interpolação Polinomial	70
3.6	Polinômios	71
REFERÊNCIAS BIBLIOGRÁFICAS		73

CAPÍTULO 1

RAÍZES DE FUNÇÕES DE UMA VARIÁVEL REAL

1.1 INTRODUÇÃO

Neste capítulo trataremos do problema geral de aproximar uma solução para a equação $f(x) = 0$ onde $f : \mathbb{R} \rightarrow \mathbb{R}$ é uma função contínua. Todos os métodos que apresentaremos são iterativos e a fim de estudar sua eficiência computacional o conceito de ordem de convergência é apresentado.

DEFINIÇÃO 1.1.1 (Raiz). Dizemos que $x^* \in \mathbb{R}$ é uma raiz de f se $f(x^*) = 0$.

É útil lembrarmo-nos de alguns fatos elementares do cálculo que são relevantes ao considerarmos o problema em questão. Primeiro, é claro que a partir do teorema do valor intermediário teremos a seguinte:

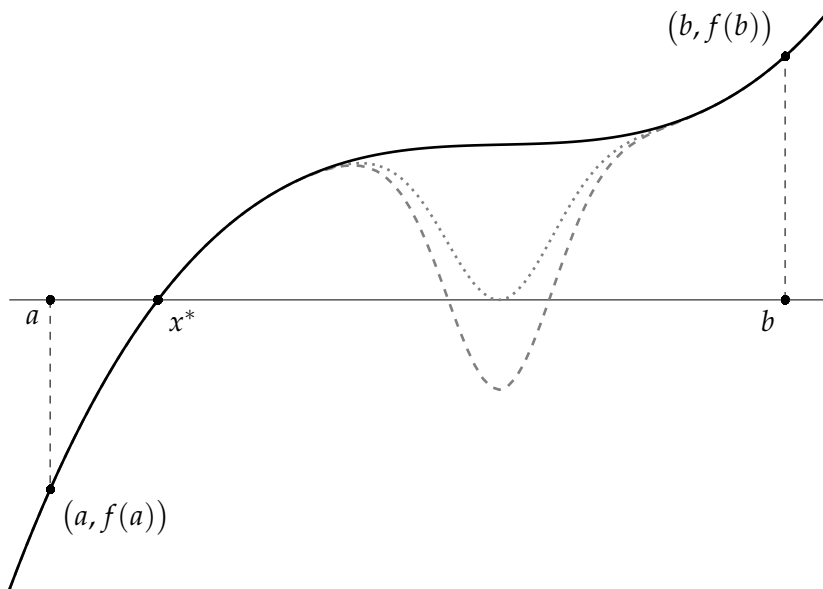


FIGURA 1.1.1 – Ilustração da Proposição 1.1.2. O ponto x^* na figura foi obtido pelo método de bissecção.

PROPOSIÇÃO 1.1.2. *Seja $f : \mathbb{R} \rightarrow \mathbb{R}$ contínua em $[a, b]$ tal que $f(a)f(b) < 0$. Então existe $x^* \in (a, b)$ tal que $f(x^*) = 0$.*

Demonstração. A afirmação decorre imediatamente do teorema do valor intermediário já que $f(a)f(b) < 0$ implica que ou $f(a) < 0$ e $f(b) > 0$ ou $f(a) > 0$ e $f(b) < 0$. \square

Esta proposição tem uma interpretação geométrica imediata, ilustrada na Figura 1.1.1: o teorema do valor intermediário nos diz que o gráfico de uma função contínua deve poder ser dese-

nhado sem que a “caneta seja retirada do papel”. Desta forma, como $f(a)f(b) < 0$ implica que os pontos $(a, f(a))$ e $(b, f(b))$ estão de lados opostos do eixo x , ao desenhar o gráfico da função eventualmente cruzaremos a reta $y = 0$ e neste ponto teremos a raiz. Observe que não há garantias acerca da unicidade. Por exemplo, na Figura 1.1.1 vemos dois traçados alternativos para f onde temos duas ou três soluções. Obviamente um número arbitrário delas é possível.

1.2 ALGORITMO DE BISSECÇÃO

A idéia por trás do método de bissecção é diminuir sucessivamente o tamanho do intervalo onde sabemos existir uma raiz. Com esta finalidade dividimos o intervalo original ao meio e utilizamos a proposição 1.1.2 para decidir em qual das metades podemos garantir a existência de uma solução. A formalização deste procedimento resulta no método de bissecção, Algoritmo 1.2.1.

Vejamos agora que o método de bissecção realmente está correto:

TEOREMA 1.2.1 (Método de Bissecção). *O Algoritmo 1.2.1 para em um número finito de passos cujo número máximo é dado por*

$$k_0 := \left\lceil \log_2 \frac{b-a}{\epsilon} \right\rceil - 1,$$

e a sua saída está correta de acordo com o alegado na seção correspondente.

Demonstração. Primeiro consideramos o comprimento dos intervalos $[a_k, b_k]$, que chamaremos de $\epsilon_k := (b_k - a_k)$. É fácil ver que $\epsilon_{k+1} = \epsilon_k/2$ pois $e_{k+1} = (b_{k+1} - a_{k+1})$ pode

Entrada: $f : \mathbb{R} \rightarrow \mathbb{R}$, a e b tais que $a < b$ e $f(a)f(b) < 0$ e $\epsilon > 0$.

Inicialização:

$a_0 = a$; $b_0 = b$; $x_0 = (a_0 + b_0)/2$; $k = 0$;

Execução:

enquanto $(b_k - a_k)/2 > \epsilon$

se $f(x_k)f(a_k) < 0$

Raiz em (a_k, x_k) .

$a_{k+1} = a_k$; $b_{k+1} = x_k$;

senão se $f(x_k)f(b_k) < 0$

Raiz em (x_k, b_k) .

$a_{k+1} = x_k$; $b_{k+1} = b_k$;

senão

Raiz encontrada!

retorna x_k ;

$++k$; $x_k = (a_k + b_k)/2$;

retorna x_k ;

Saída: Retorna x_k tal que $|x^* - x_k| \leq \epsilon$ onde $f(x^*) = 0$ ou onde f é descontínua.

ALGORITMO 1.2.1 – Bissecção

ser igual a $[b_k - (a_k + b_k)/2] = (b_k - a_k)/2$ ou então valer $[(a_k + b_k)/2 - a_k] = (b_k - a_k)/2$. Desta forma, dado que $\epsilon_0 = (b - a)$, teremos $\epsilon_k = (b - a)/2^k$.

O algoritmo para quando $\epsilon_k/2 \leq \epsilon$ ou se $f(x_k) = 0$. Caso esta última situação seja atingida para algum $k < k_0$ o teorema vale trivialmente. Suponhamos então que isso não acontece. Assim, precisamos garantir que $\epsilon_{k_0}/2 \leq \epsilon$, mas $\epsilon_k/2 = (b - a)/2^{k+1}$ então $\epsilon_k/2 \leq \epsilon$ vale se e somente se

$$\begin{aligned} \frac{b-a}{2^{k+1}} \leq \epsilon &\Leftrightarrow 2^{k+1} \geq \frac{b-a}{\epsilon} \\ &\Leftrightarrow k \geq \log_2 \frac{b-a}{\epsilon} - 1. \end{aligned}$$

Uma vez que k_0 é o primeiro inteiro que satisfaz esta desigualdade a primeira parte do teorema está demonstrada.

Com relação à saída do algoritmo, observamos que o valor retornado pelo método é x_{k_0} , o centro do intervalo $[a_{k_0}, b_{k_0}]$ cujo comprimento é menor ou igual a 2ϵ . Além disso, o algoritmo garante que $f(a_k)f(b_k) < 0$ para todo k . Assim, se f for contínua em $[a_{k_0}, b_{k_0}]$ a proposição 1.1.2 garante a existência de uma raiz $x^* \in (a_{k_0}, b_{k_0})$ e, portanto, $|x_{k_0} - x^*| < \epsilon$. Por outro lado, caso f não seja contínua em todo este intervalo então existe $x^* \in [a_{k_0}, b_{k_0}]$ onde f é descontínua, e para este x^* sabemos que $|x_{k_0} - x^*| \leq \epsilon$, o que finaliza a demonstração. \square

O método de bissecção é muito robusto. Ele converge sempre (em aritmética de precisão infinita); mesmo que a função seja descontínua a convergência está garantida. Mais que isso, se nenhuma descontinuidade salta o eixo x então o método continua convergindo para um zero da função porque toda a sequência de operações depende apenas do sinal de f . Portanto, mesmo quando a convergência é para uma descontinuidade esta é relevante, pois ocorrerá num ponto onde a função dá um salto através do eixo x .

Apesar de sua robustez, o algoritmo de bissecção converge muito lentamente. Sua convergência é apenas linear, como veremos em pouco tempo. Antes, porém, os conceitos de ordem e razão de convergência de uma sequência devem ser apresentados.

1.3 ORDEM DE CONVERGÊNCIA

Ao considerarmos uma sequência convergente $\{x_k\}$ com limite x^* é natural perguntarmos quão rápido esta sequência con-

verge, especialmente se ela tiver sido gerada por um algoritmo e o seu limite for a solução de algum problema que desejamos resolver. Nestes casos um estudo da velocidade de convergência é fundamental ao avaliarmos a complexidade computacional do algoritmo.

A ferramenta que utilizaremos para avaliar a velocidade com que uma sequência converge é a ordem de convergência, de acordo com a definição a seguir.

DEFINIÇÃO 1.3.1 (Ordem e razão de convergência estrita). *Dizemos que a sequência $\{x_k\} \subset \mathbb{R}$ converge para x^* com ordem de convergência $p \geq 1$ se existir uma sequência $m_k \geq |x_k - x^*|$ tal que o seguinte limite exista para $c \in \mathbb{R}_+$:*

$$\lim_{k \rightarrow \infty} \frac{m_{k+1}}{m_k^p} = c.$$

Alguns casos particulares merecem sua própria denominação. Quando $p = 1$ e $c \in (0, 1)$ a convergência é dita *linear* e c é a *razão de convergência*; se a convergência é linear com qualquer razão $c > 0$ dizemos que a convergência é *superlinear*. Quando $p = 2$ a sequência apresenta ordem *quadrática* de convergência.

Devemos fazer alguns comentários acerca da definição acima. O primeiro fato a ser notado é que nossa definição é diferente da adotada por muitos autores. Na maioria das vezes dita-se a existência do limite

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^p} = c. \quad (1.1)$$

Uma outra forma comum de definir ordem de convergência p seria pela seguinte desigualdade, válida para k grande o suficiente:

$$|x_{k+1} - x^*| \leq c|x_k - x^*|^p. \quad (1.2)$$

Esta última abordagem já é mais geral do que (1.1), mais continua insatisfatória pois sequer poderíamos incluir nela a nossa primeira análise. Observe que para $p = 1$ e $c < 1$ a definição acima implicaria na monotonicidade da sequência $|x_k - x^*|$, ao menos para k grande o suficiente, mas não podemos garantir que o algoritmo de bissecção satisfaça sequer 1.2.

De acordo com um exemplo encontrado em [?], se o intervalo inicial for $[0, 1]$ e tivermos uma única raiz em $3/5$ podemos verificar que o algoritmo da bissecção satisfaz $|x_{2l} - 3/5| > |x_{2l-1} - 3/5|$ para todo $l \geq 1$. A razão de tal fato deve-se à representação de $3/5$ na forma *diádica*

$$x = \frac{1}{2} + \sum_{i=2}^{\infty} c_i \frac{1}{2^i}, \quad c_i = \pm 1,$$

que no caso fica:

$$\frac{3}{5} = \frac{1}{2} + \frac{1}{4} - \frac{1}{8} - \frac{1}{16} + \frac{1}{32} + \frac{1}{64} - \dots$$

Não é difícil ver que a sequência x_k gerada pelo algoritmo da bissecção começando com o intervalo $[0, 1]$ consiste exatamente nas somas parciais da representação diádica de x^* . Uma vez que a série diádica deste racional não é monótona, a desigualdade (1.2) não poderia valer.

1.3.1 ORDEM DE CONVERGÊNCIA DA BISSECÇÃO

TEOREMA 1.3.2. *A sequência $\mathbf{x} = \{x_k\}$ gerada pelo algoritmo de bissecção converge com ordem linear e razão igual a $1/2$.*

Demonstração. Seja x^* o limite da sequência gerada pelo algoritmo. Uma vez que x_k é o centro do intervalo $[a_k, b_k]$ e sabemos

que x^* pertence ao intervalo então é óbvio que

$$|x_k - x^*| \leq \frac{|b_k - a_k|}{2} =: m_k.$$

Porém, já vimos na demonstração do teorema 1.2.1 que $|b_k - a_k| = (b - a)/2^k$, logo $m_{k+1}/m_k = 1/2$. Isto mostra que $\mathbf{m} = \{m_k\}$ tem convergência linear com razão $1/2$, o que demonstra a afirmação em vista da definição 1.3.1. \square

1.4 MÉTODO DE NEWTON

Utilizando o método da bissecção dividimos por dois o estimador para o erro de nossa aproximação por iteração. Isto significa que ganhamos um dígito binário de precisão de cada vez. Desta forma para melhorar uma casa decimal na estimativa do erro temos que efetuar $\log_2 10 \approx 3.3$ iterações; em computadores dotados de precisão dupla com 52 bits na mantissa precisaríamos efetuar 52 iterações para encontrar a raiz com precisão completa.

Por sua facilidade de aplicação e extrema robustez, o método da bissecção pode servir para uso eventual, como na produção de uma ilustração (vide a Figura 1.1.1), ou onde pouca precisão for necessária. Porém, quando se trata de incorporar o algoritmo como parte de algum método numérico onde encontrar raízes é uma rotina repetida centenas ou milhares de vezes a lentidão do método de bissecção começa a tornar-se um incômodo.

Esta última afirmação torna-se ainda mais verdadeira quando percebemos que há métodos muito mais rápidos do que o de bissecção e que são apropriados para uma ampla gama de problemas. O primeiro destes métodos que apresentaremos é o

método de Newton¹, que baseia-se na observação de que o problema $f(x) = 0$ é igual a

$$T_n^{f,\bar{x}}(x) + \frac{f^{(n+1)}(\xi)}{(n+1)!}(x - \bar{x})^{n+1} = 0,$$

onde $T_n^{f,\bar{x}}(x)$ é o polinômio de Taylor de grau n em torno de \bar{x} de f . Desta forma, se a raiz procurada x^* estiver próxima de \bar{x} o segundo termo do lado esquerdo da igualdade será pequeno em $x = x^*$ e podemos tentar encontrar uma solução para $T_n^{\bar{x}}(x)$ como uma aproximação para o problema original.

Cada iteração do método de Newton é definida pela fórmula recursiva

$$T_1^{f,x_k}(x_{k+1}) = 0.$$

Esta equação pode ser escrita explicitamente:

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0$$

e resolvida de forma simples para x_{k+1} :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

1.5 MÉTODO DE NEWTON

Utilizando o método da bissecção dividimos por dois o estimador para o erro de nossa aproximação por iteração. Isto significa que ganhamos um dígito binário de precisão de cada vez. Desta

¹Sir Isaac Newton praticamente dispensa apresentações. Um dos pais do cálculo, foi uma das mentes que mais influenciou o pensamento da humanidade em todos os tempos com suas teorias sobre ótica e mecânica.

forma para melhorar uma casa decimal na estimativa do erro temos que efetuar $\log_2 10 \approx 3.3$ iterações; em computadores dotados de precisão dupla com 52 bits na mantissa precisaríamos efetuar 52 iterações para encontrar a raiz com precisão completa.

Por sua facilidade de aplicação e extrema robustez, o método da bissecção pode servir para uso eventual, como na produção de uma ilustração (vide a figura 1.1.1), ou onde pouca precisão for necessária. Porém, quando se trata de incorporar o algoritmo como parte de algum método numérico onde encontrar raízes é uma rotina repetida centenas ou milhares de vezes a lentidão do método de bissecção começa a tornar-se um incômodo.

Esta última afirmação torna-se ainda mais verdadeira quando percebemos que há métodos muito mais rápidos do que o de bissecção e que são apropriados para uma ampla gama de problemas. O primeiro destes métodos que apresentaremos é o método de Newton², que baseia-se na observação de que o problema $f(x) = 0$ é igual a

$$T_n^{f, \bar{x}}(x) + \frac{f^{(n+1)}(\tilde{\zeta})}{(n+1)!} (x - \bar{x})^{n+1} = 0,$$

onde $T_n^{f, \bar{x}}(x)$ é o polinômio de Taylor de grau n em torno de \bar{x} de f . Desta forma, se a raiz procurada x^* estiver próxima de \bar{x} o segundo termo do lado esquerdo da igualdade será pequeno em $x = x^*$ e podemos tentar encontrar uma solução para $T_n^{\bar{x}}(x)$ como uma aproximação para o problema original.

Cada iteração do método de Newton é definida pela fórmula recursiva

$$T_1^{f, x_k}(x_{k+1}) = 0.$$

²Sir Isaac Newton praticamente dispensa apresentações. Um dos pais do cálculo, foi uma das mentes que mais influenciou o pensamento da humanidade em todos os tempos com suas teorias sobre ótica e mecânica.

Esta equação pode ser escrita explicitamente:

$$f(x_k) + f'(x_k)(x_{k+1} - x_k) = 0$$

e resolvida de forma simples para x_{k+1} :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}.$$

Há uma interpretação geométrica interessante para o método de Newton. Dado um ponto x_k , a próxima iteração fornecida pelo algoritmo será a intersecção da reta tangente à função f no ponto $(x_k, f(x_k))$ com o eixo x . Se a função em questão for suave a aproximação linear tornar-se-á cada vez melhor conforme nos aproximamos da raiz e o processo fica cumulativamente mais preciso.

Desta forma, não é uma surpresa que as propriedades do método de Newton sejam muito superiores às do método da bissecção. De fato, é o método mais rápido que apresentaremos para a solução de equações com relação ao critério de convergência assintótica. Isto não significa que seja o método a ser escolhido cegamente. Há outros fatores a serem considerados, dos quais o mais importante é a necessidade do cálculo da derivada, que fazem com que o método da secante, a ser apresentado a seguir, torne-se um competidor à altura. Antes dessas considerações práticas dediquemo-nos ao estudo preciso das propriedades assintóticas do método.

1.5.1 CONVERGÊNCIA QUADRÁTICA DO MÉTODO DE NEWTON

TEOREMA 1.5.1. *Se x_0 estiver próximo o suficiente de x^* onde $f(x^*) = 0$, f tiver duas derivadas contínuas em um intervalo em torno da raiz e $f'(x^*) \neq 0$ então x_k converge quadraticamente para x^* .*

Entrada: $f : \mathbb{R} \rightarrow \mathbb{R}$, $x_0 \in \mathbb{R}$ e $\epsilon > 0$.

Inicialização:

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}; \quad k = 1;$$

Execução:

enquanto $|x_k - x_{k-1}| > \epsilon$
 $\quad \left| \begin{array}{l} x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}; \\ ++k; \end{array} \right.$

retorna x_k ;

Saída: Retorna um valor x_k tal que $|x^* - x_k| \leq \epsilon$ onde $f(x^*) = 0$.

ALGORITMO 1.5.1 – Método de Newton

Demonstração. Consideremos o que ocorre, de uma iteração para outra, com o erro no método de Newton. Denotemos $\epsilon_k = x_k - x^*$

$$\begin{aligned} \epsilon_{k+1} &= \epsilon_k - \frac{f(x_k)}{f'(x_k)} \\ &= \frac{\epsilon_k f'(x_k) - f(x_k)}{f'(x_k)}. \end{aligned}$$

Agora suponhamos que x_k pertença ao intervalo em torno de x^* onde as hipóteses de continuidade valem e esteja próximo o suficiente de x^* para que $f(x_k) \neq 0$. Construímos então os seguintes polinômios de Taylor em torno de x^* :

$$\begin{aligned} f(x_k) &= f(x^*) + \epsilon_k f'(x^*) + \frac{\epsilon_k^2}{2} f''(\xi_k) \\ &= \epsilon_k f'(x^*) + \frac{\epsilon_k^2}{2} f''(x^*) + \epsilon_k^2 \delta_k, \end{aligned}$$

onde $\delta_k = [f''(\xi_k) - f''(x^*)]/2$ e

$$\begin{aligned} f'(x_k) &= f'(x^*) + \epsilon_k f''(\eta_k) \\ &= f'(x^*) + \epsilon_k f''(x^*) + \epsilon_k \delta'_k, \end{aligned}$$

com $\delta_k = f''(\eta_k) - f''(x^*)$ e onde tanto ξ_k quanto η_k pertencem ao intervalo (x_k, x^*) (ou (x^*, x_k)). Desta forma, denotando $\Delta_k = \delta'_k - \delta_k$, temos

$$\begin{aligned} \epsilon_k f'(x_k) - f(x_k) &= \frac{\epsilon_k^2}{2} f''(x^*) + \epsilon_k^2 (\delta'_k - \delta_k) \\ &= \epsilon_k^2 \left(\frac{1}{2} f''(x^*) + \Delta_k \right). \end{aligned} \quad (1.3)$$

E daí

$$\epsilon_{k+1} = \epsilon_k^2 \frac{f''(x^*) + 2\Delta_k}{2f'(x_k)}. \quad (1.4)$$

Observemos então que como f'' e f' são contínuas em torno de x^* e $f'(x^*) \neq 0$, existe $\sigma > 0$ tal que o conjunto $B_\sigma(x^*) := \{x \mid |x - x^*| \leq \sigma\}$ esteja contido no intervalo em torno de x^* onde f é duas vezes continuamente diferenciável e que para qualquer $x_k \in B_\sigma(x^*)$ valham as seguintes desigualdades: $|f'(x_k)| \geq 1/2|f'(x^*)|$ e $|\Delta_k| \leq 1/2$. Para este σ , se $x_k \in B_\sigma(x^*)$ teremos:

$$|\epsilon_{k+1}| \leq |\epsilon_k|^2 \frac{|f''(x^*)| + 1}{|f'(x^*)|}.$$

Portanto, é claro que se escolhermos σ tal que, além das condições acima, tenhamos

$$\sigma < \frac{|f'(x^*)|}{|f''(x^*)| + 1}$$

então

$$|\epsilon_{k+1}| \leq |\epsilon_k|C$$

onde $C < 1$ e, assim, $x_{k+1} \in B_\sigma(x^*)$. Isto permite que utilizemos (1.4) repetidamente e prova que $\{\epsilon_k\} \rightarrow 0$, ou seja, $\{x_k\} \rightarrow x^*$ se $x_0 \in B_\sigma(x^*)$.

Para demonstrar que a convergência é quadrática utilizamos (1.4, pg. 23) para obter

$$\begin{aligned} \lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^2} &= \lim_{k \rightarrow \infty} \frac{|\epsilon_{k+1}|}{|\epsilon_k|^2} \\ &= \lim_{k \rightarrow \infty} \left| \frac{f''(x^*) + 2\Delta_k}{2f'(x_k)} \right| \\ &= \frac{|f''(x^*)|}{2|f'(x^*)|}, \end{aligned}$$

igualdade que valerá se $\epsilon_k \neq 0$ para todo k . Se este não for o caso então existe k_0 tal que $\epsilon_{k_0} = 0$, ou seja, a convergência é atingida num número finito de passos. \square

NOTA 1.5.2. Se utilizarmos diretamente o polinômio de Taylor em torno de x_k podemos encontrar uma demonstração mais simples para a convergência do método de Newton da seguinte forma:

$$0 = f(x^*) = f(x_k) - \epsilon_k f'(x_k) + \epsilon_k^2 \frac{f''(\xi_k)}{2}.$$

A partir desta igualdade obtemos

$$\epsilon_k^2 \frac{f''(\xi_k)}{2f'(x_k)} = \epsilon_k - \frac{f(x_k)}{f'(x_k)} = \epsilon_{k+1}.$$

Utilizando esta estratégia, porém, o cancelamento dos termos de ordem inferior na diferença $\epsilon_k f'(x_k) - f(x_k)$, que aparece claramente no início da demonstração fornecida, passaria sem ser notado. Este

cancelamento é útil na demonstração da convergência de outros algoritmos e por isso esta prova mais longa, mas que segue um padrão mais interessante, foi preferida.

1.6 MÉTODO DA SECANTE

O método de Newton requer o cálculo da derivada da função em cada iteração. Em um sistema automatizado para a solução de equações isto requer a manutenção de um subsistema de manipulação simbólica para o cálculo de uma expressão para esta derivada. Outra opção é exigir esta expressão do usuário. Em ambos os casos há problemas. Muitas vezes o cálculo da função será executado por uma “caixa preta” e uma expressão simbólica para a função pode simplesmente não estar disponível para ser diferenciada. Por outro lado, a exigência de que o usuário forneça a derivada pode ser infactível quando a função cuja raiz é desejada varia com muita frequência.

A solução padrão nesses casos é voltarmos para o método da secante. A idéia aqui é utilizar a reta secante à função passando pelos pontos $(x_k, f(x_k))$ e $(x_{k-1}, f(x_{k-1}))$ como uma aproximação para a tangente em x_k e proceder como no método de Newton. A equação resultante para a iteração fica:

$$x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}.$$

Desta forma fica claro o papel da razão

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

como aproximação para $f'(x_k)$ e a relação com o método de Newton. A formalização do método pode ser vista no algoritmo 1.6.1.

Entrada: $f : \mathbb{R} \rightarrow \mathbb{R}$, $x_0 \in \mathbb{R}$, $x_1 \in \mathbb{R}$ e $\epsilon > 0$.

Inicialização:

$$x_2 = x_1 - f(x_1) \frac{x_1 - x_0}{f(x_1) - f(x_0)}; \quad k = 2;$$

Execução:

enquanto $|x_k - x_{k-1}| > \epsilon$
 $\left[\begin{array}{l} x_{k+1} = x_k - f(x_k) \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}; \\ ++k; \end{array} \right.$

retorna x_k ;

Saída: Retorna um valor x_k tal que $|x^* - x_k| \leq \epsilon$ onde $f(x^*) = 0$.

ALGORITMO 1.6.1 – Método da Secante

Antes de proceder com a análise do algoritmo da secante nós precisamos fazer uma breve digressão sobre sequências de Fibonacci.

1.6.1 SEQUÊNCIAS DE FIBONACCI

DEFINIÇÃO 1.6.1 (Sequência de Fibonacci). *Uma sequência $\{x_k\}$ é uma sequência de Fibonacci quando seus termos satisfazem:*

$$x_{k+2} = x_{k+1} + x_k.$$

Não é difícil ver que quaisquer dois termos consecutivos determinam completamente a sequência pois a partir deles podemos calcular os termos seguintes e deduzir os anteriores. Portanto, não há ambiguidade em denotar por $F_{a,b}$ a sequência de Fibonacci com $x_0 = a$ e $x_1 = b$. Utilizando este fato é fácil mostrar que quaisquer dois termos determinam unicamente a sequência.

Lançando mão de um truque bastante utilizado para analisar fórmulas de recorrência podemos encontrar uma fórmula geral para o termo destas sequências, conforme a proposição a seguir.

PROPOSIÇÃO 1.6.2. *Os termos de uma sequência de Fibonacci $\{x_k\}$ são dados por*

$$\alpha \left(\frac{1 + \sqrt{5}}{2} \right)^k + \beta \left(\frac{1 - \sqrt{5}}{2} \right)^k$$

onde α e β podem ser determinados a partir de quaisquer dois termos x_{k_1} e x_{k_2} ($k_1 \neq k_2$) da sequência pelo seguinte sistema linear de equações:

$$\begin{pmatrix} \left(\frac{1+\sqrt{5}}{2} \right)^{k_1} & \left(\frac{1-\sqrt{5}}{2} \right)^{k_1} \\ \left(\frac{1+\sqrt{5}}{2} \right)^{k_2} & \left(\frac{1-\sqrt{5}}{2} \right)^{k_2} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} x_{k_1} \\ x_{k_2} \end{pmatrix}.$$

Demonstração. Primeiro imaginemos que exista um número real p tal que $\{p^n\}$ seja uma sequência de Fibonacci. Isto ocorre se e somente se $p^{n+1} = p^n + p^{n-1}$. Se excluirmos a solução trivial $p = 0$ podemos dividir ambos os lados por p^{n-1} e obter a equação do segundo grau

$$p^2 - p - 1 = 0,$$

cujas soluções são

$$\phi = \frac{1 + \sqrt{5}}{2} \quad \text{e} \quad \varphi = \frac{1 - \sqrt{5}}{2}.$$

Assim conseguimos duas sequências de Fibonacci $\mathbf{y} = \{y_k\} = \{\phi^k\}$ e $\mathbf{z} = \{z_k\} = \{\varphi^k\}$. Da definição, é fácil ver que se \mathbf{y} e \mathbf{z}

são sequências de Fibonacci então $\alpha\mathbf{y} + \beta\mathbf{z}$ também é pois

$$\begin{aligned}\alpha y_k + \beta z_k &= \alpha(y_{k-1} + y_{k-2}) + \beta(z_{k-1} + z_{k-2}) \\ &= \alpha y_{k-1} + \beta z_{k-1} + \alpha y_{k-2} + \beta z_{k-2}.\end{aligned}$$

Dados dois termos x_{k_1} e x_{k_2} basta então resolver o sistema

$$\begin{pmatrix} \phi^{k_1} & \varphi^{k_1} \\ \phi^{k_2} & \varphi^{k_2} \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \begin{pmatrix} x_{k_1} \\ x_{k_2} \end{pmatrix}$$

para que $\mathbf{x} = \alpha\mathbf{y} + \beta\mathbf{z}$ seja uma sequência de Fibonacci com os termos desejados. Este é o sistema da proposição, resta mostrar que ele não é singular, mas isto é fácil dado que o determinante da matriz é

$$\phi^{k_1} \varphi^{k_2} - \phi^{k_2} \varphi^{k_1} = (\phi\varphi)^{k_1} (\varphi^{k_2-k_1} - \phi^{k_2-k_1}) \neq 0. \quad \square$$

Após estas preliminares podemos proceder para a análise do método da secante.

1.6.2 CONVERGÊNCIA DO MÉTODO DA SECANTE

TEOREMA 1.6.3. *Se x_0 e x_1 estiverem próximos o suficiente de x^* onde $f(x^*) = 0$, f tiver duas derivadas contínuas em um intervalo em torno da raiz e $f'(x^*) \neq 0$ então x_k converge com ordem $(1 + \sqrt{5})/2 \approx 1.62$ para x^* .*

Demonstração. Denotemos $\epsilon_k = x_k - x^*$. Começamos estimando $f(x_{k-1})$ pelo polinômio de Taylor em torno de x_k :

$$\begin{aligned}f(x_{k-1}) &= f(x_k) + f'(x_k)(\epsilon_{k-1} - \epsilon_k) \\ &\quad + \frac{f''(v_k)}{2}(\epsilon_{k-1} - \epsilon_k)^2,\end{aligned}$$

onde $v_k \in (x_{k-1}, x_k)$ (ou (x_k, x_{k-1})). Desta forma temos

$$\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} = \frac{\epsilon_k - \epsilon_{k-1}}{f'(x_k)(\epsilon_k - \epsilon_{k-1}) - \frac{f''(v_k)}{2}(\epsilon_k - \epsilon_{k-1})^2}$$

e, portanto,

$$\frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} = \frac{1}{f'(x_k) + \frac{1}{2}f''(v_k)(\epsilon_{k-1} - \epsilon_k)}.$$

A iteração agora pode ser escrita como:

$$\epsilon_{k+1} = \epsilon_k - \frac{f(x_k)}{f'(x_k) + \frac{1}{2}f''(v_k)(\epsilon_{k-1} - \epsilon_k)},$$

que claramente remete a uma iteração de Newton aproximada.

Procedendo como na análise do método de Newton reescrevemos esta equação:

$$\epsilon_{k+1} = \frac{\epsilon_k [f'(x_k) + \frac{1}{2}f''(v_k)(\epsilon_{k-1} - \epsilon_k)] - f(x_k)}{f'(x_k) + \frac{1}{2}f''(v_k)(\epsilon_{k-1} - \epsilon_k)}.$$

Em (1.3, pg. 23) avaliamos a diferença $\epsilon_k f'(x_k) - f(x_k)$; utilizando aquele resultado e as mesmas notações teremos

$$\epsilon_{k+1} = \epsilon_k \frac{\epsilon_k [f''(x^*) - f''(v_k) + 2\Delta_k] + \epsilon_{k-1} f''(v_k)}{2f'(x_k) + f''(v_k)(\epsilon_{k-1} - \epsilon_k)}.$$

Denotando³ $\Delta'_k = f''(x^*) - f''(v_k) + 2\Delta_k$ temos a forma mais compacta

$$\epsilon_{k+1} = \epsilon_k \frac{\epsilon_k \Delta'_k + \epsilon_{k-1} f''(v_k)}{2f'(x_k) + f''(v_k)(\epsilon_{k-1} - \epsilon_k)}. \quad (1.5)$$

³Reveja a fórmula (1.3, pg. 23) à página 23 e a discussão precedente para a definição de Δ_k .

Já assumimos implicitamente, ao utilizar as expansões de Taylor acima, que x_k e x_{k-1} pertencem ao intervalo em torno da raiz onde as derivadas são assumidas contínuas. Agora notamos que tomando eles próximos o suficiente de x^* podemos tornar a fração no lado direito da igualdade acima tão pequena, em módulo, quando queiramos e, portanto, teremos

$$|\epsilon_{k+1}| = |\epsilon_k|C$$

onde $0 \leq C < 1$ e, portanto, $\{\epsilon_k\} \rightarrow 0$ e $|\epsilon_{k+1}| < |\epsilon_k|$, logo $|\epsilon_{k+1}/\epsilon_k| < 1$.

Retomando (1.5, pg. 29) temos:

$$\epsilon_{k+1} = \epsilon_k \epsilon_{k-1} \frac{\frac{\epsilon_k}{\epsilon_{k-1}} \Delta'_k + f''(v_k)}{2f'(x_k) + f''(v_k)(\epsilon_{k-1} - \epsilon_k)}.$$

Denotemos

$$c_k = \frac{\frac{\epsilon_k}{\epsilon_{k-1}} \Delta'_k + f''(v_k)}{2f'(x_k) + f''(v_k)(\epsilon_{k-1} - \epsilon_k)},$$

assim, uma vez que $|\epsilon_{k+1}/\epsilon_k| < 1$ e $\{\epsilon_k\} \rightarrow 0$ implica que $\{\Delta'_k\} \rightarrow 0$, $\{f''(v_k)\} \rightarrow f''(x^*)$ e $\{f'(x_k)\} \rightarrow f'(x^*)$ temos $\lim_{k \rightarrow \infty} c_k = f''(x^*)/(2f'(x^*))$, ou seja:

$$\lim_{k \rightarrow \infty} \frac{\epsilon_{k+1}}{\epsilon_k \epsilon_{k-1}} = \frac{f''(x^*)}{2f'(x^*)}.$$

Como $\{c_k\}$ é convergente, existe um limitante $c \geq c_k$ para todo k . Desta forma teremos

$$|\epsilon_{k+1}| \leq c|\epsilon_k||\epsilon_{k-1}|,$$

ou seja,

$$c|\epsilon_{k+1}| \leq c|\epsilon_k|c|\epsilon_{k-1}|.$$

Então denotamos $\varepsilon_k = c|\epsilon_k|$ que, portanto, satisfaz

$$\varepsilon_{k+1} \leq \varepsilon_k \varepsilon_{k-1}.$$

Agora escolhemos uma base β e escrevemos os termos iniciais como $\varepsilon_0 = \beta^{e_0}$ e $\varepsilon_1 = \beta^{e_1}$. Então $\varepsilon_2 \leq \beta^{e_0+e_1}$ e em geral $\varepsilon_{k+1} \leq \beta^{e_k+e_{k-1}}$, logo os expoentes seguem uma sequência de Fibonacci. Denotemos por z_k o k -ésimo termo desta sequência:

$$c|\epsilon_k| \leq \beta^{z_k} \Rightarrow |\epsilon_k| \leq \frac{\beta^{z_k}}{c} =: m_k.$$

Basta então estudarmos a sequência $\{m_k\}$ para verificar seu comportamento assintótico.

Sejam $\phi = (1 + \sqrt{5})/2$ e $\varphi = (1 - \sqrt{5})/2$. Então teremos

$$\lim_{k \rightarrow \infty} \frac{m_k}{m_{k-1}^\phi} = \lim_{k \rightarrow \infty} \frac{c^\phi \beta^{z_k}}{c \beta^{\phi z_{k-1}}} = c^{\phi-1} \lim_{k \rightarrow \infty} \beta^{z_k - \phi z_{k-1}}.$$

Avaliemos o limite no expoente:

$$\lim_{k \rightarrow \infty} (z_k - \phi z_{k-1}) = \lim_{k \rightarrow \infty} (\alpha \phi^k + \beta \varphi^k - \phi(\alpha \phi^{k-1} + \beta \varphi^{k-1})),$$

portanto

$$\lim_{k \rightarrow \infty} (z_k - \phi z_{k-1}) = \lim_{k \rightarrow \infty} \beta \varphi^{k-1} (\varphi - \phi) = 0,$$

porque $|\varphi| < 1$. Notando que $\phi - 1 = -\varphi$ temos:

$$\lim_{k \rightarrow \infty} \frac{m_k}{m_{k-1}^\phi} = c^{-\varphi}.$$

□

NOTA 1.6.4. Há uma fórmula interessante para a iteração do método da secante em termos de diferenças divididas. Para obtê-la simplesmente manipulamos a equação da iteração do algoritmo:

$$\begin{aligned}
 \epsilon_{k+1} &= \epsilon_k - f(x_k) \frac{\epsilon_k - \epsilon_{k-1}}{f(x_k) - f(x_{k-1})} \\
 &= \frac{f(x_k)\epsilon_k - f(x_{k-1})\epsilon_k - f(x_k)\epsilon_{k-1} + f(x_k)\epsilon_k}{f(x_k) - f(x_{k-1})} \\
 &= \epsilon_k \epsilon_{k-1} \frac{f(x_k)/\epsilon_k - f(x_{k-1})/\epsilon_{k-1}}{f(x_k) - f(x_{k-1})} \\
 &= \epsilon_k \epsilon_{k-1} \frac{f[x_{k-1}, x^*, x_k]}{f[x_{k-1}, x_k]}.
 \end{aligned}$$

Onde usamos o operador de *diferenças divididas* definido pela recursão

$$f[x_1, \dots, x_n] = \frac{f[x_2, \dots, x_n] - f[x_1, \dots, x_{n-1}]}{x_n - x_1}$$

com a base recursiva dada por

$$f[x] = f(x).$$

Esta fórmula está em pleno acordo com os nossos resultados, o que podemos confirmar utilizando o *teorema do valor médio para diferenças divididas* que, se a função em questão for suficientemente continuamente diferenciável, afirma que teremos

$$f[x_0, \dots, x_n] = \frac{f^{(n)}(\xi)}{n!}$$

com $\xi \in (\min\{x_1, \dots, x_n\}, \max\{x_1, \dots, x_n\})$.

NOTA 1.6.5. Caso desejemos estudar a sequência $\{|\epsilon_k|\}$ diretamente, ao invés de $\{m_k\}$ ainda é possível mostrar a convergência de $\{y_k\} = \{|\epsilon_k|/|\epsilon_{k-1}|^\phi\}$ (veja [?]). Porém, mesmo sem demonstrar a existência do limite, se apenas supusermos que ele existe, podemos calcular o seu valor y^* .

Lembrando que $1/\phi = \phi - 1$ teremos

$$y_{k+1} = \frac{|\epsilon_{k+1}|}{|\epsilon_k|^\phi} = c_k \frac{|\epsilon_{k-1}|}{|\epsilon_k|^{\phi-1}} = c_k \frac{|\epsilon_{k-1}|}{|\epsilon_k|^{1/\phi}} = c_k y_k^{-1/\phi}.$$

Então concluímos que

$$y_{k+1} y_k^{1/\phi} = c_k.$$

Portanto, aplicando limites teremos

$$y^* y^{*1/\phi} = \frac{|f''(x^*)|}{2|f'(x^*)|} \Leftrightarrow y^{*1+1/\phi} = \frac{|f''(x^*)|}{2|f'(x^*)|} \Leftrightarrow y^{*\phi} = \frac{|f''(x^*)|}{2|f'(x^*)|}.$$

Logo, se o limite existir teremos

$$\lim_{k \rightarrow \infty} \frac{|\epsilon_{k+1}|}{|\epsilon_k|^\phi} = \left| \frac{f''(x^*)}{2f'(x^*)} \right|^{-\phi}.$$

Para todos os efeitos práticos nossos resultados são satisfatórios pois o limitante c na demonstração do teorema de convergência do método da secante pode ser feito tão próximo de $|f''(x^*)/[2f'(x^*)]|$ quanto desejarmos, desde que seja maior. Na prática isto significa que a razão assintótica observada será realmente no mínimo $|f''(x^*)/[2f'(x^*)]|^{1/\phi}$.

1.7 EXERCÍCIOS

1.7.1

Explique como utilizar um método que encontra solução de equações da forma $f(x) = 0$ para calcular $\sqrt[n]{a}$ onde n é um inteiro positivo e a um real não negativo;

1.7.2

Como ficam as iterações do método de Newton para o problema abordado no exercício anterior?

1.7.3

Explique a origem da relação

$$q \approx \frac{\log \frac{|x_{k+1} - x^*|}{|x_k - x^*|}}{\log \frac{|x_k - x^*|}{|x_{k-1} - x^*|}}$$

onde $\{x_k\}$ é uma sequência que converge para x^* tal que

$$\lim_{k \rightarrow \infty} \frac{|x_{k+1} - x^*|}{|x_k - x^*|^q} = c.$$

1.7.4

Utilize a fórmula aproximada para q dada no exercício anterior a fim de estimar a ordem de convergência dos algoritmos de Newton e secante quando aplicados ao problema do exercício 1.7.1. Os valores obtidos pela implementação computacional correspondem à teoria?

1.7.5

Implemente o método de Newton e o utilize para encontrar o valor de π com 15 casas decimais e estime a ordem de convergência q utilizando as seguintes equações e partindo de $x_0 = 3$:

- $1 + \cos(x) = 0$;
- $\sin(x) = 0$.

Explique os resultados obtidos à luz da teoria de convergência do método.

1.7.6

Repita o exercício anterior com o método da secante.

CAPÍTULO 2

SOLUÇÃO DE SISTEMAS DE EQUAÇÕES LINEARES

2.1 INTRODUÇÃO

Um sistemas de equações lineares é uma coleção de equações da forma

$$\begin{array}{ccccccc}
 a_{1,1}x_1 & + & a_{1,2}x_2 & + & \cdots & + & a_{1,n}x_n & = & b_1 \\
 a_{2,1}x_1 & + & a_{2,2}x_2 & + & \cdots & + & a_{2,n}x_n & = & b_2 \\
 & & & & & & \vdots & & \\
 a_{m,1}x_1 & + & a_{m,2}x_2 & + & \cdots & + & a_{m,n}x_n & = & b_m
 \end{array} \tag{2.1}$$

onde os termos a_{ij} e b_j com $i = 1, \dots, m$ e $j = 1, \dots, n$ são constates do problema. Por solução entendemos um conjunto de números x_1, \dots, x_n tal que todas as equações do sistema sejam satisfeitas.

Podemos denotar o sistema de equações de forma mais compacta se utilizarmos da álgebra usual de matrizes da seguinte

forma:

$$Ax = b. \quad (2.2)$$

Nesta equação, a matriz $A \in \mathbb{R}^{m \times n}$, que denominaremos *matriz associada* ou *matriz de coeficientes* do sistema, seria então dada¹ por

$$A = [a_{ij}] = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix}$$

e os vetores $x \in \mathbb{R}^m$ e $b \in \mathbb{R}^n$ seriam

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \quad \text{e} \quad b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}.$$

O leitor deve se acostumar com a notação compacta da forma (2.2) o quanto antes pois utilizaremos a mesma sempre que possível. Note que as operações usuais entre matrizes e vetores implicam que $Ax = b$ seja exatamente equivalente ao sistema original (2.1, pg. 37). A equação matricial, quando descrita através de suas componentes, exhibe as variáveis de pé ao lado da matriz de coeficientes:

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix}$$

¹Por simplicidade, manteremos a discussão no campo dos reais, mas todos os algoritmos que proporemos podem ser utilizados sem modificações com coeficientes e incógnitas complexas.

A primeira pergunta que surge à mente é sobre a existência de soluções para um sistema linear. Logo em seguida, desejamos saber qual seria o possível número dessas soluções. Esta segunda pergunta é, na realidade, mais fácil de ser respondida do que a primeira, conforme a proposição a seguir mostra.

PROPOSIÇÃO 2.1.1. *Um sistema de equações lineares possui uma, nenhuma ou infinitas soluções.*

Demonstração. Se existir um número finito maior do que um de soluções podemos escolher duas delas quaisquer, que denotaremos por \mathbf{x} e \mathbf{y} . Se ambas resolvem o sistema podemos mostrar que $\mathbf{z}_\alpha := (1 - \alpha)\mathbf{x} + \alpha\mathbf{y}$ também é solução para qualquer $\alpha \in \mathbb{R}$ pois

$$\begin{aligned} A\mathbf{z}_\alpha &= A[(1 - \alpha)\mathbf{x} + \alpha\mathbf{y}] = (1 - \alpha)A\mathbf{x} + \alpha A\mathbf{y} \\ &= (1 - \alpha)\mathbf{b} + \alpha\mathbf{b} = \mathbf{b}. \end{aligned}$$

Agora escolhamos $\alpha_1 \neq \alpha_2$ quaisquer e notamos que $\mathbf{z}_{\alpha_1} \neq \mathbf{z}_{\alpha_2}$ pois

$$\begin{aligned} \mathbf{z}_{\alpha_1} - \mathbf{z}_{\alpha_2} &= (1 - \alpha_1)\mathbf{x} + \alpha_1\mathbf{y} - (1 - \alpha_2)\mathbf{x} - \alpha_2\mathbf{y} \\ &= \alpha_1(\mathbf{y} - \mathbf{x}) - \alpha_2(\mathbf{y} - \mathbf{x}) \\ &= (\alpha_1 - \alpha_2)(\mathbf{y} - \mathbf{x}). \end{aligned}$$

Esta quantidade é diferente de zero pois $\mathbf{x} \neq \mathbf{y}$ e $\alpha_1 \neq \alpha_2$. Assim, para cada $\alpha \in \mathbb{R}$ temos uma solução \mathbf{z}_α diferente, ou seja, infinitas delas. \square

A discussão sobre as condições para a existência de uma única solução será postergada para um momento mais adequado, pois trataremos dela construtivamente, partindo de métodos para a obtenção da mesma.

Exercício 2.1.1. Qual é a menor dimensão para a qual podemos encontrar as três situações descritas na Proposição 2.1.1?

2.2 SOLUÇÃO DE SISTEMAS TRIANGULARES

Por enquanto fixaremos nossa atenção em sistemas quadrados, ou seja, em sistemas cujas matrizes associadas possuem um número igual de linhas e colunas. Além disso, na presente seção trataremos de um tipo específico de sistema lineares cujas soluções são fáceis de encontrar.

Uma matriz $U \in \mathbb{R}^{n \times n}$ é dita triangular superior quando é da forma

$$U = \begin{bmatrix} u_{1,1} & u_{1,2} & u_{1,3} & \cdots & u_{1,n} \\ & u_{2,2} & u_{2,3} & \cdots & u_{2,n} \\ & & u_{3,3} & \cdots & u_{3,n} \\ & & & \ddots & \vdots \\ & & & & u_{n,n} \end{bmatrix}.$$

Nesta notação, espaços em branco significam que os coeficientes correspondentes da matriz valem 0.

DEFINIÇÃO 2.2.1 (Matrizes e Sistemas Triangulares Superiores e Inferiores). *Uma matriz $\mathbb{R}^{m \times n} \ni U = [u_{i,j}]$ é triangular superior quando $j < i \Rightarrow u_{i,j} = 0$. Se um sistema linear tem uma matriz associada triangular superior, tal sistema também é dito triangular superior.*

De forma semelhante, um sistema é triangular inferior quando sua matriz $\mathbb{R}^{m \times n} \ni L = [l_{i,j}]$ for triangular inferior, ou seja, se $j > i$ implicar em $l_{i,j} = 0$.

Estudaremos aqui o caso triangular superior e o leitor será instado a encontrar o algoritmo para o caso triangular inferior

Entrada: $[u_{i,j}] = U \in \mathbb{R}^{n \times n}$ triang. sup.; $b \in \mathbb{R}^n$.

Execução:

para $i = n, \dots, 1$

$\bar{b} = b_i - \sum_{j=i+1}^n u_{i,j}x_j$;

se $u_{i,i} \neq 0$

$x_i = \bar{b}/u_{i,i}$;

senão se $\bar{b} == 0$

aviso: "Infinitas soluções detectadas!";

$x_i = 0$;

senão

erro: "Sistema sem solução!";

Saída: $x \in \mathbb{R}^n$ solução do problema ou um erro alertando para a inexistência de soluções.

ALGORITMO 2.2.1 – Substituição Reversa

nos exercícios. Em sistemas triangulares, podemos ver que sempre temos uma das equações que pode ser resolvida de forma trivial.

No caso superior, a última equação nos fornece de imediato o valor para x_n , que seria dado por (desconsideremos, no momento, a possibilidade de $u_{n,n}$ ser nulo) $x_n = b_n/u_{n,n}$. Em geral, a linha i de um sistema triangular superior permite resolver para a variável x_i desde que x_{i+1}, \dots, x_n sejam conhecidas pois teremos, dado que $u_{i,j} = 0$ para $j < i$:

$$\begin{aligned} \sum_{j=1}^n u_{i,j}x_j = b_i &\Leftrightarrow \sum_{j=i}^n u_{i,j}x_j = b_i \\ &\Leftrightarrow u_{i,i}x_i = b_i - \sum_{j=i+1}^n u_{i,j}x_j. \end{aligned}$$

A solução desta última equação para $x_{i,i}$ é trivial quando $u_{i,i} \neq 0$. Se $u_{i,i} = 0$ temos duas opções: se o lado esquerdo da igualdade for não nulo, não será possível resolvê-la. Caso o lado direito seja igual a zero, temos infinitas soluções pois qualquer escolha para x_i serve. Formalizando temos o Algoritmo 2.2.1.

2.3 ELIMINAÇÃO GAUSSIANA

O algoritmo de eliminação Gaussiana resolve um sistema linear mais geral transformando-o em um sistema triangular superior equivalente. Um exemplo expõe melhor o princípio básico do método. Imaginemos que temos o seguinte sistema:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & 3 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

Note que se subtrairmos a primeira equação de todas as subseqüentes, eliminaremos a dependência das equações na variável x_1 . Dessa forma, ficaremos com o sistema equivalente:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 2 & 3 & 2 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 2 \\ 3 \end{bmatrix}.$$

O sistema inicial foi escolhido para tornar as coisas mais fáceis, mas a ideia deve estar se tornando clara. Agora, tratamos de eliminar a dependência de x_2 nas equações subseqüentes à segunda linha. Como a última equação já não depende de x_2 ,

basta subtrair o dobro da primeira linha da segunda:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & -1 & -4 \\ 0 & 0 & -1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 3 \end{bmatrix}.$$

Por fim subtraímos a penúltima igualdade da última equação:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 \\ 0 & 0 & -1 & -4 \\ 0 & 0 & 0 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 3 \end{bmatrix}.$$

Desta forma, podemos aplicar o Algoritmo 2.2.1 e encontrar a solução. Primeiro, $x_4 = 3/3 = 1$; na sequência temos $x_3 = [0 - (-4 \times 1)]/-1 = -4$; logo depois $x_2 = [1 - (2 \times -4 + 3 \times 1)]/1 = 6$; por fim, $x_1 = [1 - (1 \times 6 + 1 \times -4 + 1 \times 1)]/1 = -2$. Ou seja:

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 3 & 4 \\ 1 & 3 & 4 & 3 \\ 1 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} -2 \\ 6 \\ -4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}.$$

Portanto, a ideia é anular entradas da matriz de coeficientes sem alterar a solução do sistema linear de forma a obter no fim um sistema triangular superior equivalente ao sistema original. Isto é feito subtraindo múltiplos da linha i de A das linhas $\{i+1, i+2, \dots, n\}$ de A com o intuito de anular os elementos $\{a_{i+1,i}, a_{i+2,i}, \dots, a_{n,i}\}$ da matriz de coeficientes do sistema. Caso os coeficientes das colunas $\{1, 2, \dots, i-1\}$ das linhas i sejam nulos, isso não alterará zeros eventualmente existentes nessas colunas ao fim de todas operações. Observe que o valor correto

Entrada: $[a_{i,j}] = A \in \mathbb{R}^{n \times n}$, $\mathbf{b} \in \mathbb{R}^n$.

Execução:

```

para  $i = 1, 2, \dots, n$ 
  para  $k = i + 1, i + 2, \dots, n$ 
    se  $a_{i,i} == 0$ 
      erro: "Erro na eliminação!";
     $\alpha \leftarrow a_{k,i} / a_{i,i}$ ;
    para  $j = i, i + 1, \dots, n$ 
       $a_{k,j} \leftarrow a_{k,j} - \alpha a_{i,j}$ ;
     $b_k \leftarrow b_k - \alpha b_i$ ;

```

Utilize o Algoritmo 2.2.1 para encontrar a solução de $A\mathbf{x} = \mathbf{b}$;

Saída: $\mathbf{x} \in \mathbb{R}^n$ solução do problema ou um erro alertando para problemas na eliminação.

ALGORITMO 2.3.1 – Eliminação Gaussiana: Solução de Sistemas Lineares

pelo qual multiplicar a linha i antes de subtraí-la da linha k a fim de zerar o elemento $a_{k,i}$ é $a_{k,i} / a_{i,i}$. A formalização da sequência de passos utilizadas para resolver um sistema linear utilizando eliminação gaussiana é dada no Algoritmo 2.3.1.

Para melhor ilustrar essa sequência de operações, utilizaremos um esquema para visualização das entradas da matriz que são alteradas a cada passo. Um + representará coeficientes que não foram alterados desde a etapa anterior da eliminação, um \times coeficientes que foram alterados, um 0 irá representar coeficientes que acabaram de se tornar nulos e um espaço em branco representará elementos que já eram nulos anteriormente. Visualizada dessa forma, a eliminação gaussiana para uma matriz 5×5 produz a seguinte sequência de operações (incluindo a matriz original):

Entrada: $[a_{i,j}] = A \in \mathbb{R}^{n \times n}$; $b \in \mathbb{R}^n$.

Execução:

para $i = 1, 2, \dots, n$
 para $k = i + 1, i + 2, \dots, n$
 $I =_{k \in \{i, i+1, \dots, n\}} |a_{k,i}|$
 para $j = i, i + 1, \dots, n$
 $a_{k,j} \leftarrow a_{k,j} - \alpha a_{i,j}$;
 $b_k = b_k - \alpha b_i$;

Utilize o Algoritmo 2.2.1 para encontrar a solução de $Ax = b$;

Saída: $x \in \mathbb{R}^n$ solução do problema ou um erro alertando para problemas na eliminação.

ALGORITMO 2.3.2 – Eliminação Gaussiana com Pivoteamento: Solução de Sistemas Lineares

$$\begin{bmatrix} + & + & + & + & + \\ + & + & + & + & + \\ + & + & + & + & + \\ + & + & + & + & + \\ + & + & + & + & + \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} + \\ + \\ + \\ + \\ + \end{bmatrix}.$$

$$\begin{bmatrix} + & + & + & + & + \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} + \\ \times \\ \times \\ \times \\ \times \end{bmatrix}.$$

$$\begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} + \\ + \\ \times \\ \times \\ \times \end{bmatrix}.$$

$$\begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & 0 & \times & \times \\ & & 0 & \times & \times \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} + \\ + \\ + \\ \times \\ \times \end{bmatrix}.$$

$$\begin{bmatrix} + & + & + & + & + \\ & + & + & + & + \\ & & + & + & + \\ & & & + & + \\ & & & 0 & \times \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix} = \begin{bmatrix} + \\ + \\ + \\ + \\ \times \end{bmatrix}.$$

2.4 EXERCÍCIOS

2.4.1

Aplique a eliminação gaussiana ao sistema

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 3 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 3 \\ 2 \end{bmatrix}.$$

Qual a solução desse sistema? Por que o método falhou?

2.4.2

Defina apropriadamente um sistema triangular inferior e descreva um método para a solução desse tipo de sistema linear.

2.4.3

Um sistema por bandas satisfaz, para algum b muito menor do que n , $|j - i| \geq b \Rightarrow a_{ij} = 0$. Descreva um método de solução eficiente para esse tipo de sistema. Note que tanto a eliminação como a substituição reversa precisam ser especialmente pensadas para essa situação.

2.5 ELIMINAÇÃO GAUSSIANA COM PIVOTEAMENTO PARCIAL

A essa altura já deve ter sido possível notar que o método da eliminação gaussiana apresentado na seção anterior não é capaz de resolver todo sistema linear que possui uma única solução, o que é um problema grave, ao menos em tese. Por exemplo, o sistema simples

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

faz com que o método de eliminação gaussiana falhe já nos primeiro passo pois $a_{1,1} = 0$. Em geral esse tipo de falha ocorre sempre que $a_{i,i}$ se anula no passo i do laço mais externo do método.

Uma solução para esse tipo de problema é fazer um pivoteamento, ou seja, mudar a ordem das equações no sistema para que o “pivô”, isto é, o elemento $a_{i,i}$ não seja nulo no princípio de

cada passo do laço exterior do algoritmo. Há diversas formas de se conseguir esse efeito, uma das quais resulta no algoritmo conhecido como eliminação gaussiana com pivoteamento parcial. Neste método, no passo i do laço externo do algoritmo a linha I com $I \in \{i, i+1, \dots, n\}$ tal que $|a_{I,i}| = \max_{l \in \{i, i+1, \dots, n\}} |a_{l,i}|$ é permutada com a linha i . Dessa forma, o método somente falha se todos os elementos $a_{i,i}, a_{i+1,i}, \dots, a_{n,i}$ forem nulos, o que significa que colunas $1, 2, \dots, i$ são linearmente dependentes (pois são i vetores com apenas as primeiras $i-1$ componentes diferentes de zero).

Talvez a primeira ideia que venha à mente seja simplesmente, no caso em que $a_{i,i}$ seja nulo, procurar pelo primeiro $a_{k,i} \neq 0$ e fazer a permutação entre a linha onde isso ocorre e a linha i . Entretanto, o custo computacional do pivoteamento parcial descrito no parágrafo acima é mínimo comparado ao restante das operações executadas pelo algoritmo e ele possui a propriedade adicional de tornar o método menos susceptível a erros de arredondamento mesmo nos casos em que o o algoritmo sem pivoteamento não falharia.

2.6 EXERCÍCIOS

2.6.1

Aplique a eliminação gaussiana com pivoteamento parcial ao sistema

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 3 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 4 \\ 3 \\ 2 \end{bmatrix}.$$

2.6.2

Implemente o método da substituição reversa e os métodos de eliminação gaussiana e eliminação gaussiana com pivoteamento parcial. Teste suas implementações para sistemas aleatórios de tamanho 300×300 ou maior e

- meça a diferença relativa entre o tempo de execução dos algoritmos
- calcule para os dois métodos o valor de $\|Ax - b\|$

repita esse experimento algumas vezes e analise os resultados. Quantas vezes a eliminação gaussiana sem pivoteamento falhou?

2.7 FATORAÇÃO $PA = LU$

Caso determinados coeficientes obtidos durante eliminação e a ordem das permutações de linhas forem armazenados apropriadamente, o algoritmo de eliminação gaussiana com pivoteamento parcial é capaz de fornecer uma fatoração da forma $PA = LU$ onde P é uma matriz obtida através de permutação das linhas da matriz identidade, A é a matriz original, L é uma matriz triangular inferior com os elementos da diagonal igual a 1 e U é uma matriz triangular superior.

Há diversas utilidades para uma fatoração desse tipo. Por exemplo, suponhamos que seja necessário resolver diversos sistemas lineares com a mesma matriz de coeficientes mas com distintos lados direito. De posse de uma fatoração $PA = LU$, um sistema na forma $Ax = b$ pode ser substituído pelo equivalente $PAx = Pb$. Note que o efeito de pré-multiplicar ambos os lados por P é o de trocar linhas do sistema. Desta forma, vemos que o sistema equivale a $LUx = b$ e esse sistema pode ser resolvido

em dois passos que envolvem somente a solução de sistemas triangulares:

$$Ly = Pb$$

$$Ux = y.$$

Outra aplicação é no cálculo de determinantes. Uma vez que $\det(LU) = \det(L) \det(U)$, temos

$$\det(P) \det(A) = \det(PA) = \det(LU) = \det(L) \det(U).$$

Além disso, sabemos que $\det(P) = \pm 1$, e o valor é -1 se houve um número ímpar de permutações a partir da identidade para chegar em P e 1 caso contrário. Também é conhecido o fato que o determinante de uma matriz triangular é o produto dos elementos de sua diagonal. Como os elementos da diagonal de L são todos 1 , temos

$$\det(P) \det(A) = \det(U).$$

Ou seja, $\det(A) = \det(P) \prod_{i=1}^n u_{i,i}$.

O algoritmo que computa a fatoração $PA = LU$ é a eliminação gaussiana com pivoteamento parcial, desde que troquemos as linhas de P , partindo de $P = I$, conforme os pivoteamentos são efetuados e que armazenemos os fatores α calculados pelo método na posição correspondente de L . O Algoritmo 2.7.1 detalha as operações necessárias.

2.8 EXERCÍCIOS

2.8.1

Calcule a fatora  o $PA = LU$ de

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 3 & 4 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}.$$

2.8.2

Implemente um algoritmo para c  culo da fatora  o $PA = LU$. Nesta implementa  o, represente a fatora  o na forma compacta em que, ao longo da execu  o do algoritmo, as entradas de L s  o armazenadas na parte triangular inferior da matriz A enquanto a matriz U    armazenada na parte triangular de A . Fa  a tamb  m com que a matriz P seja representada por um vetor p inicializado como $p = (1, 2, \dots, n)^T$ que ter   suas entradas permutadas em vez de permutarmos as linhas de uma matriz.

2.8.3

Aplique o algoritmo do exerc  cio anterior para criar um programa que calcule a inversa de uma matriz n  o-singular.

2.8.4

Um sistema por bandas satisfaz, para algum b muito menor do que n , $|j - i| \geq b \Rightarrow a_{i,j} = 0$. Descreva um m  todo de eficiente para c  culo da fatora  o $A = LU$ para esse caso (note

a ausência da matriz de permutação, ou seja, o algoritmo é sem pivoteamento – por quê?).

Entrada: $[a_{i,j}] = A \in \mathbb{R}^{n \times n}$; $\mathbf{b} \in \mathbb{R}^n$.

Execução:

para $i = 1, 2, \dots, n$

 Encontre $I \in \{i, i+1, \dots, n\}$ tal que:

$$|a_{I,i}| = \max_{l \in \{i, i+1, \dots, n\}} |a_{l,i}|$$

 Troque a linha I de A com a linha i de A

 Troque os valores de b_I e b_i

para $k = i+1, i+2, \dots, n$

se $a_{i,i} == 0$

 └ **erro:** "Erro na eliminação!";

$\alpha \leftarrow a_{k,i} / a_{i,i};$

para $j = i+1, \dots, n$

 └ $a_{k,j} \leftarrow a_{k,j} - \alpha a_{i,j};$

 └ $b_k = b_k - \alpha b_i;$

Utilize o Algoritmo 2.2.1 para encontrar a solução de $A\mathbf{x} = \mathbf{b}$;

Saída: $\mathbf{x} \in \mathbb{R}^n$ solução do problema ou um erro alertando para problemas na eliminação.

ALGORITMO 2.5.1 – Eliminação Gaussiana com Pivoteamento Parcial:
Solução de Sistemas Lineares

Entrada: $[a_{i,j}] = A \in \mathbb{R}^{n \times n}$.

Faça $L \leftarrow 0$

Faça $U \leftarrow A$

Faça $P \leftarrow I$

Execução:

para $i = 1, 2, \dots, n$

 Encontre $I \in \{i, i+1, \dots, n\}$ tal que:

$$|u_{I,i}| = \max_{l \in \{i, i+1, \dots, n\}} |u_{l,i}|$$

 Troque a linha I de U com a linha i de U

 Troque a linha I de L com a linha i de L

 Troque a linha I de P com a linha i de P

para $k = i+1, i+2, \dots, n$

se $u_{i,i} == 0$

erro: “Erro na fatoração: matriz singular!”;

$l_{k,i} \leftarrow u_{k,i}/u_{i,i}$;

para $j = i+1, \dots, n$

$u_{k,j} \leftarrow u_{k,j} - l_{k,i}a_{i,j}$;

Faça $l_{i,i} = 1$ para $i \in \{1, 2, \dots, n\}$.

Saída: P, L, U tais que $PA = LU$ ou mensagem de erro alertando que a fatoração é impossível.

CAPÍTULO 3

INTERPOLAÇÃO POLINOMIAL

3.1 INTRODUÇÃO

Por interpolação entendemos a estimação de uma função a partir de amostras pontuais de forma que o erro se anule nos pontos amostrados. Por exemplo, dada uma tabela de logaritmos, como estimar o valor de $\log x$ quando x não é um dos pontos tabelados? Mais geralmente, dado o valor de $f : \mathbb{R} \rightarrow \mathbb{R}$ no conjunto $X = \{x_0, \dots, x_n\}$, procuramos $p(x)$ que forneça uma estimativa para o valor de $f(x)$. Como sabemos o valor de f nos pontos, é natural exigirmos $p(x_i) = f(x_i)$ para $i = 0, \dots, n$. Se uma função se iguala a f em todo $x \in X$ dizemos que ela interpola f nos pontos de X .

No restante deste capítulo estudaremos polinômios interpoladores. Veremos que, se definidos com o grau apropriado, eles sempre existem e são únicos. Fazemos isso e partimos simultaneamente para o estudo métodos práticos para a obtenção e uso do polinômio interpolador.

Basicamente, o problema de interpolação polinomial será resolvido em duas etapas, mais uma terceira opcional. Primeiro desejamos construir o polinômio interpolador a partir do conjunto de dados; depois desejaremos avaliar o polinômio obtido em um dado ponto x ; por fim podemos precisar incluir mais um ponto na interpolação, ou seja, atualizar o polinômio interpolador. Apresentaremos os métodos de Lagrange, especialmente em suas encarnações baricêntricas, e o de Newton. As três etapas (construção, avaliação e atualização) de ambos serão discutidas.

Após apresentar as metodologias discutiremos o erro em que incorremos ao aproximar uma função por um polinômio interpolador, o fenômeno de Runge, os pontos de Chebyshev e a forma que assume o polinômio interpolador nesses pontos sob a representação baricêntrica.

Por fim serão apresentadas notas bibliográficas para o leitor interessado em se aprofundar no assunto, como por exemplo onde encontrar uma discussão acerca da estabilidade dos métodos.

3.2 O POLINÔMIO INTERPOLADOR

DEFINIÇÃO 3.2.1. *Dado um conjunto de $n + 1$ pontos distintos $X = \{x_0, \dots, x_n\}$ dizemos que $p(x)$ é o polinômio interpolador de f em X se $p(x_i) = f(x_i)$ para $i = 0, \dots, n$ e p tem um grau menor ou igual a n .*

É mais fácil mostrar que o polinômio interpolador existe construindo o mesmo. Para tanto, seja $X = \{x_0, \dots, x_n\}$ e considere os seguinte polinômios, com $i = 0, \dots, n$:

$$\ell_i(x) := \frac{(x - x_0) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)}. \quad (3.1)$$

Então vemos que ℓ_i possui grau n e também que

$$\ell_i(x_j) = \begin{cases} 0 & \text{se } i \neq j \\ 1 & \text{se } i = j \end{cases}.$$

Uma vez que para todo i o termo $\ell_i(x)$ é um polinômio de grau igual a n o seguinte polinômio possui grau menor ou igual a n :

$$p(x) := f(x_0)\ell_0(x) + f(x_1)\ell_1(x) + \cdots + f(x_n)\ell_n(x).$$

Pela forma como foi construído, é bastante claro que $p(x_i) = f(x_i)$, ou seja, $p(x)$ é o polinômio interpolador. Em geral podemos estabelecer o seguinte resultado:

PROPOSIÇÃO 3.2.2. *Dados $n + 1$ pontos distintos x_0, \dots, x_n e $n + 1$ valores quaisquer f_0, \dots, f_n , existe um polinômio $p(x)$ de grau menor ou igual a n tal que $p(x_i) = f_i$ para $i = 0, \dots, n$.*

Demonstração. Simplesmente tome $p(x) = \sum_{i=0}^n f_i \ell_i(x)$. □

Para que a Definição 3.2.1 esteja correta (note que ela define “o polinômio interpolador”, não “um polinômio interpolador”), é preciso mostrar que este polinômio é único, conforme a seguir:

PROPOSIÇÃO 3.2.3. *O polinômio da Proposição 3.2.2 é único.*

Demonstração. Sejam dois polinômios quaisquer $p(x)$ e $q(x)$ de grau menor ou igual a n que satisfaçam $p(x_i) = q(x_i) = f_i$ para $i = 0, \dots, n$. Então $r(x) := p(x) - q(x)$ também é um polinômio de grau menor ou igual a n que satisfaz $r(x_i) = 0$, $i = 0, \dots, n$. Entretanto, um polinômio com $n + 1$ raízes ou é identicamente nulo ou possui grau maior ou igual a $n + 1$, de onde concluímos que $r(x) \equiv 0$, ou seja, $p(x) = q(x)$. □

As próximas seções dedicam-se a estabelecer metodologias para o cálculo do polinômio interpolador. Algumas delas possuem interesse principalmente teórico enquanto outras suscitam imediatamente algoritmos práticos.

3.3 FORMA DE LAGRANGE DO POLINÔMIO INTERPOLADOR

3.3.1 CONSTRUÇÃO

Dizemos que o polinômio interpolador, quando escrito como

$$p(x) = \sum_{i=0}^n f_i \ell_i(x), \quad (3.2)$$

está na forma de Lagrange.

Dada a definição (3.1, pg. 56), para determinar esse polinômio precisamos calcular todos os coeficientes abaixo para $i = 0, \dots, n$:

$$w_i := \frac{1}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}. \quad (3.3)$$

O cálculo de cada um desses valores exige o cômputo de n adições, $n - 1$ multiplicações e uma inversão, ao todo $2n$ operações, como são $n + 1$ pesos temos $2n(n + 1)$ operações de ponto flutuante. Uma característica importante dessa construção é que a parte custosa dela é independente dos valores f_i , o que tanto é econômico computacionalmente em muitas circunstâncias como pode facilitar construções interessantes, como no Exemplo 3.3.2.

Exemplo 3.3.1. Suponha que desejamos passar uma reta entre dois pontos (x_1, y_1) e (x_2, y_2) no plano. Se $x_1 = x_2$, a reta é

dada por $x = x_1$, senão temos um problema de interpolação polinomial cuja solução na forma de Lagrange é dada por:

$$y = y_1 \frac{x - x_2}{x_1 - x_2} + y_2 \frac{x - x_1}{x_2 - x_1}.$$

Podemos rearranjar esta expressão para a mais familiar, da trigonometria:

$$\frac{y - y_1}{x - x_1} = \frac{y_2 - y_1}{x_2 - x_1}.$$

Exemplo 3.3.2. Queremos aproximar o valor de

$$\frac{2}{\sqrt{\pi}} \int_0^{0.1} e^{-x^2} dx$$

integrando um polinômio que interpole e^{-x^2} nos pontos 0, 0.05 e 0.1. Em geral procuraremos a aproximação

$$\int_a^b f(x) dx \approx \int_a^b p(x) dx,$$

onde $p(x)$ interpola f nos pontos a , b e $(a + b)/2$. Tornamos o problema mais familiar se introduzirmos as variáveis $x_0 = a$ e h

$= (b - a)/2$ e denotarmos $x_i = x_0 + ih$:

$$\begin{aligned}
 \int_a^b p(x)dx &= \int_{x_0}^{x_0+2h} p(x)dx \\
 &= \int_{x_0}^{x_0+2h} f(x_0) \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} dx \\
 &\quad + \int_{x_0}^{x_0+2h} f(x_1) \frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} dx \\
 &\quad + \int_{x_0}^{x_0+2h} f(x_2) \frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} dx \\
 &= \frac{f(x_1)}{2h^2} \int_{-h}^h t(t-h)dt \\
 &\quad - \frac{f(x_0)}{h^2} \int_0^{2h} t(t-2h)dt \\
 &\quad + \frac{f(x_2)}{2h^2} \int_{-h}^h t(t+h)dt \\
 &= \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)].
 \end{aligned}$$

Tal expressão pode ser escrita como

$$\int_a^b f(x)dx \approx \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right].$$

A aproximação que obtivemos acima é conhecida como a regra de Simpson. Podemos testar sua validade numérica para o caso em questão utilizando a função `erf()` do Octave:

```

octave:1> f = inline( 'exp(-x.^2)' );
octave:2> est = 0.1/6 * ( f(0) + 4*f(0.05) + f(0.1) );
octave:3> abs( 2/sqrt( pi )*est - erf( 0.1 ) )
ans = 4.6375e-08

```

Caso esteja curioso sobre `erf()`, peça ajuda:

```
octave:4> help erf
' erf ' is a built-in function
```

```
-- Mapping Function: erf (Z)
    Computes the error function,
```

$$\text{erf}(z) = \frac{2}{\sqrt{\pi}} \int_{t=0}^z e^{-t^2} dt$$

See also: erfc, erfinv

Exemplo 3.3.3. Consideremos a tarefa de encontrar o polinômio interpolador da função $\sin(x)$ nos pontos $\{-\pi, -\pi/2, 0, \pi/2, \pi\}$. Simplificamos o problema notando que $f(x_0) = f(x_2) = f(x_4) = 0$. Por isso precisamos calcular apenas:

$$\begin{aligned} w_1 &= \frac{1}{\left(-\frac{\pi}{2} + \pi\right) \left(-\frac{\pi}{2} - 0\right) \left(-\frac{\pi}{2} - \frac{\pi}{2}\right) \left(-\frac{\pi}{2} - \pi\right)} \\ &= \frac{1}{\frac{\pi}{2} \cdot -\frac{\pi}{2} \cdot \pi \cdot \frac{3\pi}{2}} = -\frac{8}{3\pi^4}; \\ w_3 &= \frac{1}{\left(\frac{\pi}{2} + \pi\right) \left(\frac{\pi}{2} + \frac{\pi}{2}\right) \left(\frac{\pi}{2} - 0\right) \left(\frac{\pi}{2} - \pi\right)} \\ &= \frac{1}{\frac{3\pi}{2} \cdot \pi \cdot \frac{\pi}{2} \cdot -\frac{\pi}{2}} = -\frac{8}{3\pi^4}. \end{aligned}$$

Lembrando que $f(x_1) = -1$ e $f(x_3) = 1$ vemos que o polinômio interpolador é dado por:

$$\begin{aligned} p(x) &= \frac{8}{3\pi^4} (x + \pi) x \left(x - \frac{\pi}{2}\right) (x - \pi) \\ &\quad - \frac{8}{3\pi^4} (x + \pi) \left(x + \frac{\pi}{2}\right) x (x - \pi). \end{aligned}$$

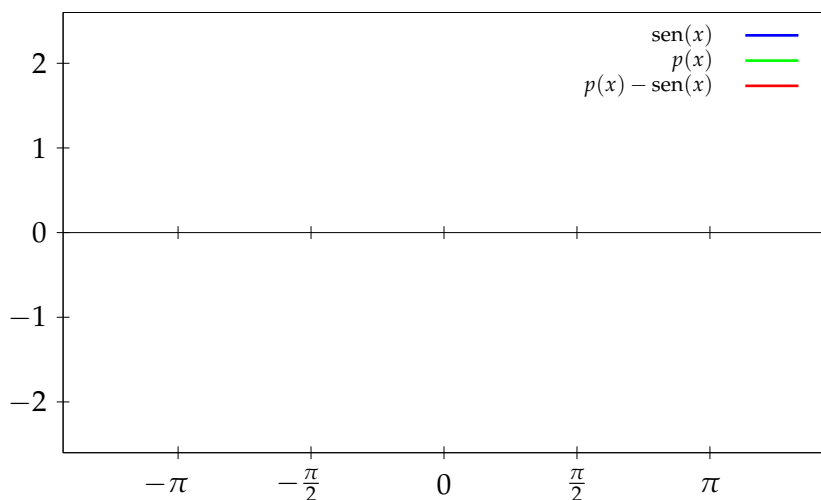


FIGURA 3.3.1 – Polinômio interpolador de $\text{sen}(x)$ nos pontos $\{-\pi, -\pi/2, 0, \pi/2, \pi\}$. Também são exibidos gráficos de $\text{sen}(x)$ e da diferença entre o polinômio interpolador e a função.

3.3.2 AVALIAÇÃO

Após encontrado o polinômio interpolador do exemplo acima, queremos compará-lo graficamente à função $\text{sen}(x)$. Podemos utilizar o Octave:

```
octave:1> x = -4.5:0.1:4.5;
octave:2> y = (x + pi).* x .* (x - pi/2).*(x - pi);
octave:3> y = y - (x + pi).*(x + pi/2).* x .* (x - pi);
octave:4> y = y * 8/(3*pi^4);
octave:5> plot( x, sin(x), x, y, x, y-sin(x) );
```

Isto resultaria em um gráfico essencialmente igual ao apresentado na Figura 3.3.1.

Ainda que tenhamos resolvido o problema da interpolação polinomial, a forma (3.2, pg. 58) é bastante inconveniente quando precisamos avaliar o polinômio em um ponto x qualquer. Casos como o Exemplo 3.3.3 são exceções e normalmente não podemos confiar em coincidências do tipo $f(x_i) = 0$. Em geral, para o cálculo de $p(x)$ a partir de (3.2, pg. 58) precisamos calcular $\ell_i(x)$ para $i = 0, \dots, n$. Cada $\ell_i(x)$ utiliza $2n$ operações de ponto flutuante (n subtrações, $n - 1$ multiplicações e uma divisão), totalizando $2n(n + 1)$ operações apenas para o cálculo dos polinômios de Lagrange. Este gargalo de $O(n^2)$ seria desastroso em aplicações práticas, mas felizmente existem formas alternativas de representar o polinômio interpolador utilizando os pesos w_i que não apresentam esta dificuldade e possuem ainda outras vantagens. Estas são as formas baricêntricas que veremos a seguir.

Exercício 3.3.1. Encontre o polinômio $p(x)$ que interpola a função $f(x) = e^{-x}$ nos pontos $\{-0.1, 0, 0.1, 0.2, 0.3\}$. Avalie $p(-0.05)$ e $p(0.15)$.

3.3.3 PRIMEIRA FORMA BARICÊNTRICA

Na listagem do começo da subseção anterior notamos que as linhas 2 e 3 são muito semelhantes. De fato, a melhor forma de digitar a terceira linha é pressionar a seta para cima no teclado e reeditar a segunda linha (que reaparecerá após digitarmos a seta para cima). Apenas dois termos serão alterados, isso é uma pequena consequência prática do seguinte fato matemático:

$$\frac{\ell_i(x)}{w_i}(x - x_i) = \frac{\ell_j(x)}{w_j}(x - x_j). \quad (3.4)$$

Estávamos calculando duas frações como as acima, portanto:

$$\frac{\ell_i(x)}{w_i} = \frac{\ell_j(x)}{w_j} \frac{x - x_j}{x - x_i}.$$

A razão $\frac{x-x_j}{x-x_i}$ explica porque apenas dois fatores são diferentes: o fator $x - x_i$ “sai” do produto no numerador¹ de $\ell_j(x)$ em (3.1, pg. 56) e $x - x_j$ “entra”.

Inspirados em (3.4, pg. 63) definimos

$$\ell(x) = (x - x_0) \cdots (x - x_n) = \frac{\ell_i(x)}{w_i} (x - x_i).$$

Então escrevemos

$$p(x) = \sum_{i=0}^n f_i \ell_i(x) = \sum_{i=0}^n f_i \frac{w_i}{x - x_i} \ell(x).$$

Finalmente temos então a primeira forma baricêntrica:

$$p(x) = \ell(x) \sum_{i=0}^n \frac{w_i}{x - x_i} f_i. \quad (3.5)$$

Qual é a grande vantagem que (3.5) possui sobre (3.2, pg. 58)? Contando o número de operações necessárias para avaliar a equação acima temos: $4n + 3$ para avaliar a soma e $2n + 1$ para avaliar $\ell(x)$, totalizando $6n + 5$ operações ao incluirmos o produto final. Contrastando com o $O(n^2)$ da fórmula (3.2, pg. 58) podemos entender a vantagem da forma baricêntrica.

Exercício 3.3.2. Repita o Exercício 3.3.1 utilizando a primeira forma baricêntrica.

¹Note que este numerador é exatamente $\ell_j(x)/w_j$.

Exercício 3.3.3. Mostre que se um polinômio p de grau menor ou igual a n satisfaz $p(x_i) = c$ para $n + 1$ valores distintos $\{x_0, \dots, x_n\}$ então p é constante: $p(x) \equiv c$. Dica: $p(x) - c$ tem $n + 1$ raízes.

3.3.4 SEGUNDA FORMA BARICÊNTRICA

A ideia da segunda forma baricêntrica é eliminar fatores comum $\ell(x)$ em (3.5) através da representação do número 1 pela mesma fórmula. Este expediente leva a representações mais eficientes para algumas distribuições úteis para os pontos de interpolação porque fatores comuns a todos w_i podem ser eliminados. A ideia resulta também numa função que é robusta com relação ao cálculo dos pesos w_i no sentido que ela interpola os pontos desejados mesmo que estes coeficientes tenham sido calculados erroneamente.

Se um polinômio de grau n vale 1 em $n + 1$ pontos então esse polinômio é constante (veja Exercício 3.3.3), ou seja, utilizando (3.5) temos:

$$1 = \ell(x) \sum_{i=0}^n \frac{w_i}{x - x_i}.$$

Portanto

$$p(x) = \frac{\ell(x) \sum_{i=0}^n \frac{w_i}{x - x_i} f_i}{\ell(x) \sum_{i=0}^n \frac{w_i}{x - x_i}}.$$

Logo:

$$p(x) = \frac{\sum_{i=0}^n \frac{w_i}{x - x_i} f_i}{\sum_{i=0}^n \frac{w_i}{x - x_i}}, \quad (3.6)$$

que é a segunda forma baricêntrica.

3.3.5 ATUALIZAÇÃO

3.4 FORMA DE NEWTON DO POLINÔMIO INTERPOLADOR

3.4.1 OPERADOR DE DIFERENÇAS DIVIDIDAS

Antes de exibir a forma de Newton do polinômio interpolador nos permitimos uma digressão a fim de apresentar o principal ingrediente da construção. Introduzimos o operador de diferenças divididas de ordem $n + 1$ da seguinte forma:

DEFINIÇÃO 3.4.1. *Definimos a diferença dividida de f nos pontos x_0, \dots, x_n como*

$$f[x_0, \dots, x_n] = \begin{cases} \frac{f[x_0, \dots, x_{n-1}] - f[x_1, \dots, x_n]}{x_0 - x_n} & \text{se } n > 0 \\ f(x_0) & \text{caso contrário.} \end{cases}$$

Para calcular a diferença dividida $f[x_0, \dots, x_n]$ precisaremos de $f[x_0, \dots, x_{n-1}]$ e de $f[x_1, \dots, x_n]$. O cálculo destas exige o conhecimento de $f[x_0, \dots, x_{n-2}]$, $f[x_1, \dots, x_{n-1}]$ e $f[x_2, \dots, x_n]$. Seguindo por esse caminho logo vemos que será necessário o

cálculo de todas as diferenças divididas da seguinte tabela:

$$\begin{array}{ccccccc}
 f[x_0] & & & & & & \\
 & f[x_0, x_1] & & & & & \\
 f[x_1] & & & & & & \\
 & f[x_1, x_2] & & \ddots & & & \\
 f[x_2] & & & & f[x_0, \dots, x_{n-1}] & & \\
 & f[x_2, x_3] & & \cdots & & f[x_0, \dots, x_n] & \\
 f[x_3] & & & & f[x_1, \dots, x_n] & & \\
 & \vdots & & \ddots & & & \\
 \vdots & & & & & & \\
 & f[x_{n-1}, x_n] & & & & & \\
 f[x_n] & & & & & &
 \end{array}$$

Por exemplo,

PROPOSIÇÃO 3.4.2. *Podemos escrever a diferença dividida como*

$$f[x_0, \dots, x_n] = \sum_{i=0}^n \frac{f(x_i)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_n)}.$$

Demonstração. Procedemos por indução. A equação vale trivialmente para $n = 0$, suponhamos como hipótese indutiva que vale para algum n , então incluímos mais um ponto a partir da

definição:

$$\begin{aligned}
 f[x_0, \dots, x_{n+1}] &= \frac{f[x_0, \dots, x_n] - f[x_1, \dots, x_{n+1}]}{x_0 - x_{n+1}} \\
 &= \sum_{i=0}^n \frac{f(x_i)}{(x_0 - x_{n+1}) \prod_{j=0}^{i-1} (x_i - x_j) \prod_{j=i+1}^n (x_i - x_j)} \\
 &\quad - \sum_{i=1}^{n+1} \frac{f(x_i)}{(x_0 - x_{n+1}) \prod_{j=1}^{i-1} (x_i - x_j) \prod_{j=i+1}^{n+1} (x_i - x_j)} \\
 &= \sum_{i=0}^{n+1} \frac{(x_i - x_{n+1})f(x_i) - (x_i - x_0)f(x_i)}{(x_0 - x_{n+1}) \prod_{j=0}^{i-1} (x_i - x_j) \prod_{j=i+1}^{n+1} (x_i - x_j)} \\
 &= \sum_{i=0}^{n+1} \frac{(x_0 - x_{n+1})f(x_i)}{(x_0 - x_{n+1}) \prod_{j=0}^{i-1} (x_i - x_j) \prod_{j=i+1}^{n+1} (x_i - x_j)} \\
 &= \sum_{i=0}^{n+1} \frac{f(x_i)}{(x_i - x_0) \cdots (x_i - x_{i-1})(x_i - x_{i+1}) \cdots (x_i - x_{n+1})}. \square
 \end{aligned}$$

Uma consequência imediata do resultado acima é a seguinte proposição:

PROPOSIÇÃO 3.4.3. *Seja (i_0, \dots, i_n) uma permutação qualquer de $(0, \dots, n)$, então*

$$f[x_0, \dots, x_n] = f[x_{i_0}, \dots, x_{i_n}].$$

3.4.2 CONSTRUÇÃO

PROPOSIÇÃO 3.4.4 (Forma de Newton do Polinômio Interpolador). *O polinômio que interpola f nos pontos distintos x_0, \dots, x_n é*

dado por:

$$\begin{aligned} p_n(x) &= f(x_0) + (x - x_0)f[x_0, x_1] \\ &\quad + (x - x_0)(x - x_1)f[x_0, x_1, x_2] + \cdots \\ &\quad + (x - x_0) \cdots (x - x_{n-1})f[x_0, \dots, x_n]. \end{aligned} \quad (3.7)$$

Mais que isso, temos

$$f(x) = p_n(x) + (x - x_0) \cdots (x - x_n)f[x_0, \dots, x_n, x].$$

Demonstração. Primeiro notamos que a segunda fórmula implica que o polinômio definido acima interpola a função nos pontos x_0, \dots, x_n pois o segundo termo no lado direito da referida igualdade se anula nesses pontos. Dessa forma tudo que precisamos mostrar é a segunda igualdade.

Podemos ver que ela vale trivialmente para $n = 0$. Partimos então para uma demonstração por indução supondo que a afirmação valha para n e mostrando que isso implica a sua veracidade para $n + 1$. Pela Proposição 3.4.4 temos

$$\begin{aligned} f[x_0, \dots, x_{n+1}, x] &= f[x, x_0, \dots, x_{n+1}] \\ &= \frac{f[x, x_0, \dots, x_n] - f[x_0, \dots, x_{n+1}]}{x - x_{n+1}} \\ &= \frac{f[x_0, \dots, x_n, x] - f[x_0, \dots, x_{n+1}]}{x - x_{n+1}}. \end{aligned}$$

Agora aplicamos a hipótese indutiva rearranjada da forma

$$f[x_0, \dots, x_n, x] = \frac{f(x) - p_n(x)}{(x - x_0) \cdots (x - x_n)}$$

e substituímos essa expressão na igualdade anterior:

$$f[x_0, \dots, x_{n+1}, x] = \frac{\frac{f(x) - p_n(x)}{(x - x_0) \cdots (x - x_n)} - f[x_0, \dots, x_{n+1}]}{x - x_{n+1}}.$$

Isolando $f(x)$ obtemos

$$f(x) = p_n(x) + (x - x_0) \cdots (x - x_n) f[x_0, \dots, x_{n+1}] \\ + (x - x_0) \cdots (x - x_{n+1}) f[x_0, \dots, x_{n+1}, x].$$

Porém, observando (3.7, pg. 69) vemos que essa igualdade equivale a

$$f(x) = p_{n+1}(x) + (x - x_0) \cdots (x - x_{n+1}) f[x_0, \dots, x_{n+1}, x]. \quad \square$$

3.4.3 AVALIAÇÃO

3.4.4 ATUALIZAÇÃO

3.5 ERRO NA INTERPOLAÇÃO POLINOMIAL

TEOREMA 3.5.1. *Seja f contínua em $[a, b]$ com $f^{(n+1)}$ contínua em (a, b) então, se $a \leq x_0 \leq x_1 \leq \cdots \leq x_n \leq b$ e $P(x)$ é o polinômio que interpola f nos pontos x_0, \dots, x_n temos*

$$f(x) - P(x) = (x - x_0)(x - x_1) \cdots (x - x_n) \frac{f^{(n+1)}(\xi_x)}{(n+1)!},$$

onde $\xi_x \in (\min\{x, x_0\}, \max\{x, x_n\})$.

Demonstração. Como $P(x)$ interpola f , vemos que $R(x) := f(x) - P(x)$ se anula nos pontos $\{x_0, \dots, x_n\}$. Definamos a função

$$F(t) := f(t) - P(t) - (t - x_0) \cdots (t - x_n) \frac{f(x) - P(x)}{(x - x_0) \cdots (x - x_n)}.$$

Não é difícil ver que $F(t)$ se anula em $\{x, x_0, \dots, x_n\}$. Assim, aplicando o teorema de Rolle vemos que $F'(t)$ se anula em $n+1$

pontos distintos. Repetindo o argumento, $F''(t)$ vale zero em n pontos também distintos. Continuando o processo podemos ver que $F^{(n+1)}$ anula-se em um ponto que denotamos por ξ_x . Porém, diferenciando sucessivamente teremos:

$$F^{(n+1)}(t) = f^{(n+1)}(t) - (n+1)! \frac{f(x) - P(x)}{(x-x_0) \cdots (x-x_n)}.$$

Avaliando a função em $t = \xi_x$ obtém-se

$$0 = f^{(n+1)}(\xi_x) - (n+1)! \frac{f(x) - P(x)}{(x-x_0) \cdots (x-x_n)}.$$

Agora basta isolar $f(x) - P(x)$ nesta igualdade. □

3.6 POLINÔMIOS

Precisamos definir precisamente os objetos matemáticos com que trabalhamos durante o restante do capítulo. Sem dúvida o leitor está familiarizado com polinômios, portanto a presente seção será breve, apenas apresentando o conceito e os resultados básicos que utilizamos no decorrer da discussão acima.

DEFINIÇÃO 3.6.1. *Definimos um polinômio como uma função $p : \mathbb{R} \mapsto \mathbb{R}$ que pode ser escrita como:*

$$p(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad (3.8)$$

para algum inteiro positivo n e $n+1$ reais a_0, \dots, a_n .

A mais comumente citada característica distintiva de um polinômio talvez seja o seu grau:

DEFINIÇÃO 3.6.2. *O grau de um polinômio $p(x)$ é o maior n para o qual $a_n \neq 0$. O grau do polinômio nulo $p(x) \equiv 0$ é, por definição, -1 .*

Por exemplo, a posição de um corpo sob aceleração constante pode ser descrita como $p(t) = x_0 + v_0t + \frac{a}{2}t^2$, que é um polinômio de grau dois em t se $a \neq 0$.

REFERÊNCIAS BIBLIOGRÁFICAS