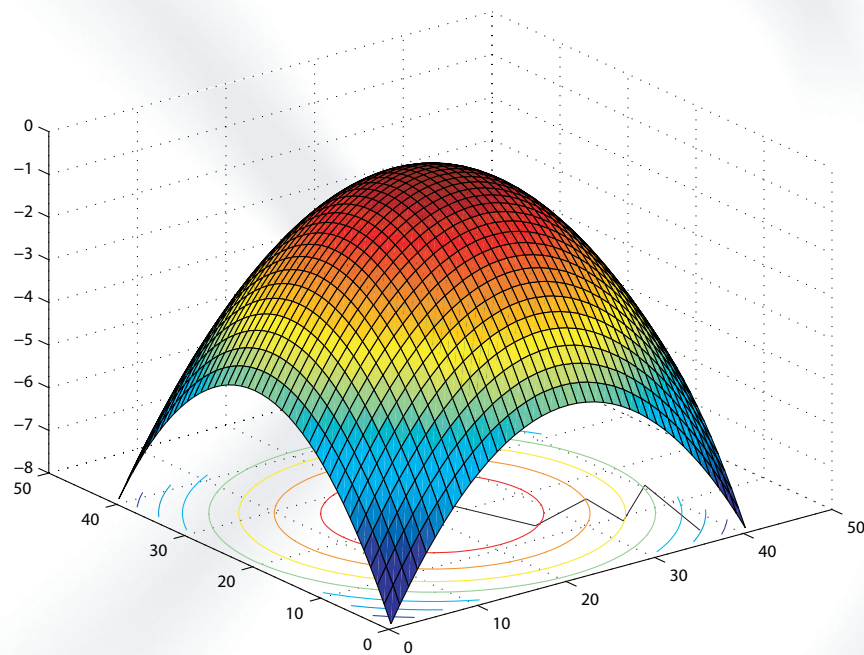


CÁLCULO NUMÉRICO

COM



MATLAB

Flaulles B.Bergamaschi

PARA ELIANE...

Sumário

1	Sistemas Lineares	1
1.1	Introdução	1
1.1.1	Solução de um sistema $n \times n$	3
1.2	Métodos Diretos	4
1.2.1	Método de Gauss	4
1.2.2	Decomposição LU	7
1.3	Métodos Iterativos	11
1.3.1	Método Iterativo de Jacobi	11
1.3.2	Critério de Parada	14
1.3.3	Método Iterativo de Gauss-Seidel	15
1.3.4	Método do Refinamento Iterativo	17
1.3.5	Número Condicional	17
1.3.6	Convergência do M. Iterativo de Jacobi e Gauss-Seidel	21
1.4	Sistemas Lineares Complexos	22
1.5	Exercícios	24
2	Zeros de função	29
2.1	Introdução	29
2.2	Método de Localização de Zeros	31
2.3	Método do Meio Intervalo - MMI	33
2.4	Método da Secante	35
2.4.1	Convergência no Método da Secante	37
2.5	Método de Newton	37
2.5.1	Convergência no Método de Newton	39
2.6	Método da Iteração Linear	40

2.6.1	A Função de Iteração	43
2.7	Comentários Finais Sobre os Métodos	45
2.7.1	Localização de Zeros	45
2.7.2	Método do Meio Intervalo - MMI	45
2.7.3	Método da Secante	45
2.7.4	Método de Newton	45
2.7.5	Método da Iteração Linear	46
2.8	Zeros de um Polinômio	47
2.8.1	Multiplicidade de um zero	54
2.9	Exercícios	55
3	Interpolação	59
3.1	Introdução	59
3.2	Interpolação de Lagrange	60
3.3	Interpolação com Diferenças Divididas Finitas - DDF	63
3.3.1	Propriedades de uma DDF	63
3.3.2	Obtenção da Fórmula	64
3.4	Erro na Interpolação	65
3.5	Método de Briot-Ruffini	69
3.6	Considerações Finais	70
3.7	Exercícios	71
4	Integração Numérica	75
4.1	Introdução	75
4.2	Regra dos Trapézios	76
4.2.1	Erro na Integral por Trapézios	77
4.3	1ª Regra de Simpson	79
4.4	Quadratura Gaussiana	82
4.5	Exercícios	87
5	Mínimos e Máximos	89
5.1	Introdução	89
5.2	Método da Seção Áurea	92
5.2.1	Convergência no Método da Seção Áurea	93
5.3	Superfícies em \mathbb{R}^3	93
5.4	Método do Gradiente	95
5.5	Bacias de atração	96
5.6	Método de direções aleatórias	97

5.7	Exercícios	98
6	Introdução ao Matlab	101
6.1	Introdução	101
6.2	Comandos	103
6.2.1	Comando de leitura	103
6.2.2	Comando de impressão	103
6.2.3	Comando de atribuição	104
6.2.4	Estrutura de decisão	105
6.2.5	Estruturas de repetição	106
6.3	Itens Básicos do Matlab	109
6.3.1	Operadores relacionais	109
6.3.2	Conectivos lógicos	110
6.3.3	Funções Pré-definidas	110
6.3.4	Script	111
6.4	Vetores e Matrizes	114
6.5	Funções em Matlab	117
6.6	Gráficos Bidimensionais	119
6.7	Gráficos Tridimensionais	121
6.8	Exercícios	122
7	Implementação dos Métodos	125
7.1	Sistemas Lineares	125
7.2	Zeros de Função	136
7.3	Interpolação	142
7.4	Integração	144
7.5	Otimização	145
7.6	Exercícios	151
8	Respostas dos exercícios	153
A	Erros	159
A.1	Números em ponto flutuante	159
	Bibliografia	167

Lista de Figuras

1.1	Solução geométrica de um sistema 2×2	3
1.2	Retas paralelas	4
1.3	Seqüência x^n	14
2.1	Zeros de uma função	30
2.2	$f'(x) > 0$ e $f'(x) < 0$	30
2.3	Zeros de $f(x)$	31
2.4	Zeros de $f(x)$	32
2.5	Criando uma partição \mathbb{P}	33
2.6	(x_n) convergindo para o zero ϵ	34
2.7	Seqüência (x_n) no método da secante	36
2.8	Seqüência (x_n) no método de Newton	38
2.9	Seqüência (x_n) no método da Iteração Linear	41
2.10	Seqüência (x_n) no Método da Iteração Linear	41
2.11	Método gráfico	47
2.12	Limite dos zeros	50
2.13	Isolando zeros	51
2.14	Limite de zeros complexos	53
3.1	Erro	69
4.1	Área do trapézio	77
4.2	Cálculo da área por trapézios	78
4.3	Cálculo da área pela 1ª regra de Simpson	80
4.4	Cálculo da área por trapézios	80
5.1	Mínimos local e global	90
5.2	Parabolóide	94

5.3	Seqüência (p_n) se aproximando do mínimo de $f(x)$. .	96
5.4	Bacias de atração	97
6.1	Gráfico de $f(x) = x^2$	120
6.2	Gráfico de $f(x, y) = x^2 + y^2$	121
6.3	Gráfico+(curva de nível) de $f(x, y) = x^2 + y^2$	121

Prefácio

Este livro de cálculo numérico foi escrito com base nas notas de aula, ministradas na Universidade Federal do Espírito Santo e na Universidade Estadual do Sudoeste da Bahia. A importância do cálculo numérico para os estudantes de engenharia, física e matemática se faz na conexão dos métodos numéricos matemáticos com a computação, cada vez mais presente no dia-dia acadêmico. Lembremos que os métodos numéricos começaram a ser criados mesmo antes de Cristo; agora com o advento do computador esses métodos puderam ser implementados e assim uma quantidade de problemas puderam ser resolvidos. O que antes se conhecia apenas como existência e unicidade, agora pode ser determinado por aproximações tão precisas quanto se queira. Sempre respeitando as limitações de máquina.

Tentei escrever um livro para ser usado em um semestre letivo nos cursos de graduação, por isso, no que tange o conteúdo de cada capítulo, existe uma preocupação em expor os principais métodos numéricos. O leitor interessado pode procurar na bibliografia para aprofundar mais. Temos como objetivo que o estudante tenha uma visão profunda em temas relevantes e uma visão geral em outros, montando assim seu conhecimento e metodologia de estudo em relação aos métodos numéricos.

Em cada capítulo, focamos assuntos relevantes em cálculo numérico e que tenham importância teórica e prática. Muitas demonstrações não são feitas devido a complexidade e o fato de fugirem do tema principal, mas podem ser obtidas nas referências. Dedicamos um capítulo, a introdução do software Matlab, que será suficiente para implementar os métodos apresentados aqui. Claro que os métodos também podem ser implementados em uma outra linguagem. Um comentário importante: grande parte dos métodos numéricos apresentados aqui já estão implementados no Matlab em forma de funções pré-definidas. Mesmo assim, é de suma importância o estudo desses métodos, uma vez que os problemas práticos exigem mudanças.

Dei especial atenção ao capítulo de zeros de funções, apresentando métodos numéricos e analíticos para encontrar zeros de uma função. Uma seção dedicada exclusivamente aos zeros de um po-

linômio, faz o diferencial deste livro. Fornecendo um potente teorema que limita os zeros de um polinômio.

Em geral não é comum o tema Mínimos e Máximos em textos de cálculo numérico. Por isso, o professor desejoso, pode omitir esse capítulo. Esse tema foi colocado neste texto devido a importância prática cada vez maior, e o fato de ser uma ótima aplicação dos métodos numéricos.

Como pré-requisito, o leitor deve ter em mente os cursos de cálculo, álgebra linear e geometria analítica. No capítulo 6 fizemos uma introdução aos algoritmos em Matlab, o que elimina um curso básico de programação como pré-requisito.

Gostaria de agradecer aos alunos da UESB e em especial ao professor Júlio César pelas correções e sugestões efetuadas no decorrer do trabalho.

Vitória da Conquista, inverno de 2009.
Flaules Boone Bergamaschi

Capítulo 1

Sistemas Lineares

1.1 Introdução

Neste capítulo vamos desenvolver técnicas e métodos numéricos para resolver sistemas lineares $n \times n$. Esses sistemas aparecem com frequência em problemas da engenharia, física, química, etc... Com o advento do computador tais métodos ganharam mais atenção, ficando assim evidente a importância de um estudo mais aprofundado. Lembramos o leitor que alguns tópicos básicos de álgebra linear são necessários. Por isso, aconselhamos o uso de algum livro sobre o assunto para acompanhamento.

Um sistema de equações lineares com m equações e n incógnitas é dado na forma:

$$(*) \begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & & \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n & = & b_m \end{cases}$$

Com a_{ij} ($i = 1, \dots, m$, $j = 1 \dots n$) números reais ou complexos.

A solução do sistema (*) é um conjunto (x_1, x_2, \dots, x_n) que satisfaça todas as m equações.

O sistema (*) também pode ser escrito na forma matricial:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix},$$

ou seja, $AX = B$ com A sendo a matriz dos coeficientes, X o vetor de incógnitas e B o vetor de termos independentes.

Chamamos de matriz ampliada do sistema (*) a matriz formada pela junção do vetor de termos independentes e a matriz dos coeficientes.

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} & b_m \end{bmatrix}$$

É importante notar que a matriz ampliada do sistema é o ponto de partida para encontrar-mos a solução do sistema via métodos numéricos, o que desenvolveremos mais adiante.

Exemplo 1.1.1. *Consideremos o seguinte sistema 2×2*

$$\begin{cases} 2x_1 + x_2 = 3 \\ x_1 + 4x_2 = 5 \end{cases}$$

$$A = \begin{bmatrix} 2 & 1 \\ 1 & 4 \end{bmatrix}, \quad B = \begin{bmatrix} 3 \\ 5 \end{bmatrix}, \quad X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$\text{Matriz ampliada do sistema: } \begin{bmatrix} 2 & 1 & 3 \\ 1 & 4 & 5 \end{bmatrix}$$

Por motivos técnicos e computacionais trataremos apenas o caso de sistemas onde o número de equações e incógnitas são iguais, ou seja, $m = n$. Matricialmente isso quer dizer que, a matriz dos coeficientes é uma matriz quadrada. Também vamos assumir que a matriz dos coeficientes é uma matriz real.

Geometricamente a solução de uma sistema linear $(m \times n)$ é a interseção de m hiperplanos em \mathbb{R}^n . Veja no exemplo:

Exemplo 1.1.2. Considere o sistema do exemplo anterior onde a solução é dada por $P = (1, 1)^T$, que é exatamente a interseção das retas $2x_1 + x_2 = 3$ e $x_1 + 4x_2 = 5$ conforme Figura 1.1

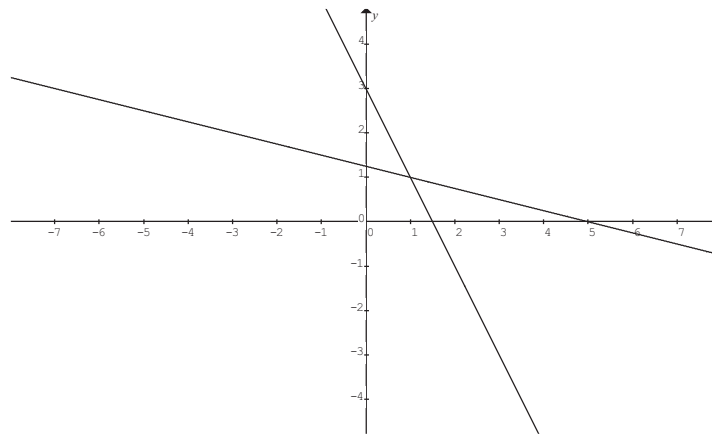


Figura 1.1: Solução geométrica de um sistema 2×2

1.1.1 Solução de um sistema $n \times n$

Vamos relembrar alguns resultados da álgebra linear:

- i. Um sistema linear ($n \times n$) possui solução única se o determinante da matriz dos coeficientes é diferente de zero.*
- ii. Caso o determinante seja zero, o sistema não possui solução ou possui infinitas soluções.*

A demonstração dos itens acima pode ser encontrada em [2]

Exemplo 1.1.3. Considere o sistema

$$\begin{cases} 2x_1 + x_2 = 3 \\ 4x_1 + 2x_2 = -6 \end{cases}, \text{ onde } \det \begin{bmatrix} 2 & 1 \\ 4 & 2 \end{bmatrix} = 0$$

Neste caso as retas $2x_1 + x_2 = 3$ e $4x_1 + 2x_2 = -6$ (veja Figura 1.2) são paralelas. Logo esse sistema não possui solução.

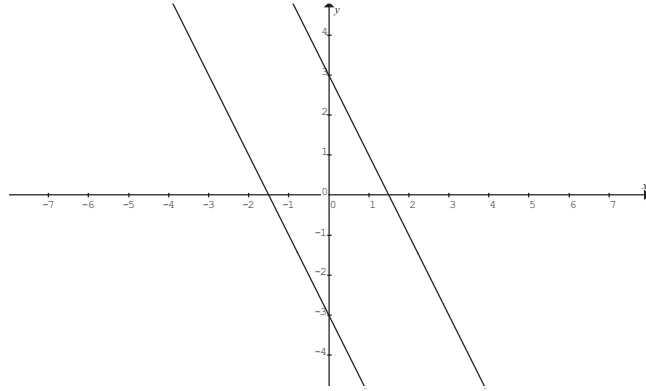


Figura 1.2: Retas paralelas

Em problemas práticos é comum encontrar sistemas lineares de grande porte, por exemplo $n > 1000$. Por isso é necessário desenvolvermos métodos numéricos para encontrar a solução de tais sistemas de tal forma que, seja sempre possível implementar algoritmos computacionais.

Começamos com os métodos numéricos diretos que fornecem a solução exata ¹ através de um número finito de passos.

1.2 Métodos Diretos

1.2.1 Método de Gauss

Este método trabalha com a equivalência de sistemas através de operações elementares na matriz ampliada. Para começar, vamos definir as operações elementares sobre as linhas de uma matriz.

Operações elementares

- i. trocar linhas, $L_i \longleftrightarrow L_j$.*
- ii. Multiplicar uma linha por um escalar $k \neq 0$, $L_i \longrightarrow kL_i$.*
- iii. Substituir uma linha por sua soma com um múltiplo escalar de outra linha. $k \neq 0$, $L_i \longrightarrow L_i + kL_j$.*

¹Quando não existem erros de truncamento e arredondamento

Definição 1.1. Dizemos que as matrizes $A_{m \times n}$ e $B_{m \times n}$ são linha equivalentes, se $B_{m \times n}$ pode ser obtida através de operações elementares em $A_{m \times n}$.

Na definição acima podemos usar a notação $A \sim B$ (A é linha equivalente a B).

Exemplo 1.2.1. A matriz $A = \begin{bmatrix} 0 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix}$ é linha equivalente a $B =$

$$\begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}.$$

Aplicando operações elementares em A temos:

$$\underbrace{\begin{bmatrix} 0 & 2 & 2 \\ 1 & 2 & 3 \end{bmatrix}}_A \xrightarrow{L_1 \leftrightarrow L_2} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 2 & 2 \end{bmatrix} \xrightarrow{L_2 \rightarrow \frac{1}{2}L_2} \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \end{bmatrix} \xrightarrow{L_1 \rightarrow L_1 + (-1)L_2} \underbrace{\begin{bmatrix} 1 & 1 & 2 \\ 0 & 1 & 1 \end{bmatrix}}_B$$

Teorema 1.1. Dois sistemas que possuem matrizes ampliadas equivalentes são equivalentes, ou seja, tem mesma solução.

A demonstração desse teorema pode ser encontrada em [2].

Exemplo 1.2.2. O sistema $\begin{cases} 2x_1 + x_2 = 3 \\ x_1 + x_2 = 2 \end{cases}$ e $\begin{cases} 4x_1 + 2x_2 = 6 \\ \frac{1}{2}x_1 + \frac{1}{2}x_2 = 1 \end{cases}$ são equivalentes.

Basta observar que:

$$\begin{bmatrix} 2 & 1 & 3 \\ 1 & 1 & 2 \end{bmatrix} \xrightarrow{L_1 \rightarrow 2L_1} \begin{bmatrix} 4 & 2 & 6 \\ 1 & 1 & 2 \end{bmatrix} \xrightarrow{L_2 \rightarrow \frac{1}{2}L_2} \begin{bmatrix} 4 & 2 & 6 \\ \frac{1}{2} & \frac{1}{2} & 1 \end{bmatrix}$$

Com o Teorema 1.1 estamos prontos para iniciar o método de Gauss, que consiste em transformar a matriz ampliada de um sistema através de operações elementares em uma matriz da forma:

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} & b_1 \\ 0 & a_{22} & a_{23} & \cdots & a_{2n} & b_2 \\ 0 & 0 & a_{33} & \cdots & a_{3n} & b_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{mn} & b_m \end{bmatrix}, \quad (1.1)$$

ou seja, a matriz dos coeficientes é uma matriz triangular superior.

Exemplo 1.2.3. Considere o sistema $\begin{cases} x_1 + x_2 = 2 \\ 2x_1 + x_2 = 3 \end{cases}$. Efetuando operações elementares na matriz ampliada temos:

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \end{bmatrix} \xrightarrow{L_2 \rightarrow L_2 + (-2)L_1} \begin{bmatrix} 1 & 1 & 2 \\ 0 & -1 & -1 \end{bmatrix}$$

Portanto este sistema é equivalente a $\begin{cases} x_1 + x_2 = 2 \\ -x_2 = -1 \end{cases}$, que é facilmente resolvido por substituição retroativa, ou seja, encontramos x_2 na segunda equação e substituímos na primeira equação, encontrando x_1 .

Daremos agora os passos para obter a matriz equivalente no caso de um sistema 3×3 .

Considere então o sistema $\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{cases}$ e sua matriz ampliada:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & b_1 \\ a_{21} & a_{22} & a_{23} & b_2 \\ a_{31} & a_{32} & a_{33} & b_3 \end{bmatrix}$$

Com os passos abaixo é possível transformar a matriz A em uma matriz na forma dada em (1.1), através de operações elementares.

1º passo

Definimos o elemento chamado de *pivo* como a_{11} e calculamos:

$$m_{21} = -\frac{a_{21}}{pivo} \text{ e } m_{31} = -\frac{a_{31}}{pivo}$$

2º passo

Operamos na matriz ampliada A :

$$L_2 \longrightarrow L_2 + m_{21}L_1$$

$$L_3 \longrightarrow L_3 + m_{31}L_1$$

3º passo

O elemento *pivo* passe a ser a_{22} e calculamos:

$$m_{32} = -\frac{a_{32}}{pivo}$$

4º passo

$$L_3 \longrightarrow L_3 + m_{32}L_2$$

Veja que o elemento *pivo* toma sempre os elementos na diagonal da matriz dos coeficientes.

O caso $n \times n$ é análogo ao dado acima. Observamos que nesse algoritmo matemático, o elemento *pivo* deve ser diferente de zero, em outras palavras, todos os elementos na diagonal da matriz dos coeficientes deve ser diferente de zero. Mas nem tudo está perdido! Caso algum elemento a_{kk} seja igual a zero, deve-se usar a operação elementar de troca de linha, ou seja, troca-se a linha k por uma linha r tal que $k < r$.

1.2.2 Decomposição LU

Seja $AX = B$ um sistema $n \times n$ e $\det(A) \neq 0$. Suponha que A possa se decompor no produto de uma matriz triangular inferior L , e uma matriz triangular superior U , tal que $A = LU$, assim $AX = B$ equivale a $(LU)X = B$. Dessa forma podemos obter dois sistemas, $LY = B$ e $UX = Y$. Como L e U são triangulares, o sistema $LY = B$ é rapidamente resolvido por substituição retroativa, e logo após $UX = Y$.

Assim o problema agora é decompor a matriz A no produto de L e U . Recorremos então a álgebra linear onde esse problema é conhecido como decomposição LU . Começamos com o

Teorema 1.2. *Seja $A_{n \times n}$ uma matriz qualquer e A_{kk} uma submatriz de $A_{n \times n}$ formada pela interseção das primeiras k linhas e k colunas. Se $\det(A_{kk}) \neq 0$ para $k = 1, \dots, n-1$ então existem e são únicas as matrizes L e U tal que $A = LU$.*

A demonstração pode ser encontrada em [8]

Para obter L e U o processo vem da eliminação Gaussiana, aquela feita na seção anterior, onde L é uma matriz triangular inferior com diagonal igual a 1 e multiplicadores $-m_{ij}$, e U uma matriz triangular superior formada pelos elementos da forma final de A . Veja no caso 3×3 ,

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ -m_{31} & -m_{32} & 1 \end{bmatrix}}_L \cdot \underbrace{\begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}}_U$$

Exemplo 1.2.4. *Considere o sistema*
$$\begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ 4x_1 + 4x_2 - 3x_3 = 3 \\ 2x_1 - 3x_2 + x_3 = -1 \end{cases},$$

onde a matriz dos coeficientes é dada por $A = \begin{bmatrix} 2 & 3 & -1 \\ 4 & 4 & -3 \\ 2 & -3 & 1 \end{bmatrix}$.

Tomando pivo = a_{11} , calculando $m_{21} = -\frac{a_{21}}{\text{pivo}} = -2$, $m_{31} = -\frac{a_{31}}{\text{pivo}} = -1$ e fazendo $L_2 \rightarrow L_2 + m_{21}L_1$ e $L_3 \rightarrow L_3 + m_{31}L_1$ obtemos a matriz:

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

Tomando agora, pivo = a_{22} , calculando $m_{32} = -\frac{a_{32}}{\text{pivo}} = -3$ e fazendo $L_3 \rightarrow L_3 + m_{32}L_2$ obtemos a matriz:

$$\begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & 5 \end{bmatrix}$$

Dessa forma obtemos a decomposição de A em:

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \quad e \quad U = \begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & 5 \end{bmatrix}$$

Assim o sistema $LY = B$ equivale a:

$$\begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 3 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ -1 \end{bmatrix} \Rightarrow \begin{cases} y_1 = 5 \\ 2y_1 + y_2 = 3 \\ y_1 + 3y_2 + y_3 = -1 \end{cases}$$

Resolvendo por substituição retroativa temos a solução $y_1 = 5$, $y_2 = -7$ e $y_3 = 15$, onde podemos agora montar o sistema $UX = Y$:

$$\begin{bmatrix} 2 & 3 & -1 \\ 0 & -2 & -1 \\ 0 & 0 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ -7 \\ 15 \end{bmatrix} \Rightarrow \begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ -2x_2 - x_3 = -7 \\ 5x_3 = 15 \end{cases}$$

Novamente por substituição retroativa obtemos $x_1 = 1$, $x_2 = 2$ e $x_3 = 3$.

O leitor já deve ter observado que usamos o método de eliminação de Gauss para fazer a decomposição LU . Isso nos leva a pensar que o método de Gauss da seção anterior é equivalente a decomposição LU . Veremos essa diferença no exemplo abaixo.

Exemplo 1.2.5. *Inversão de Matrizes.*

Dada uma matriz $A_{n \times n}$ tal que $\det(A) \neq 0$. Então A possui inversa A^{-1} . Para encontrar A^{-1} devemos resolver n sistemas lineares, veja:

Fazendo $A^{-1} = X$ temos $AX = I$, ou seja,

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \underbrace{\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nn} \end{bmatrix}}_{A^{-1}} = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 \end{bmatrix}$$

onde os sistemas são:

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_{11} \\ x_{21} \\ \vdots \\ x_{n1} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad (\text{sistema 1})$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_{12} \\ x_{22} \\ \vdots \\ x_{n2} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix} \quad (\text{sistema 2})$$

\vdots

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \cdot \begin{bmatrix} x_{1n} \\ x_{2n} \\ \vdots \\ x_{nn} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \quad (\text{sistema } n)$$

Veja que os n sistemas lineares tem a mesma matriz de coeficientes A . Para usar o método de Gauss deveríamos aplicá-lo n vezes. Por outro lado, uma vez aplicado o método de Gauss em A obtemos a decomposição LU . Agora resolvemos os sistemas por substituição retroativa. Isso reduz consideravelmente o número de operações.

Existem outros métodos para se obter as matrizes L e U , o leitor interessado pode consultar o método de Doolittle e Crout em [11].

1.3 Métodos Iterativos

Os métodos iterativos são caracterizados por uma função chamada de função de iteração. Essa função deve ser obtida de tal forma que possamos garantir que a sequência produzida por ela convirja para solução do sistema, em outras palavras, dado o sistema $AX = B$, com $A_{n \times n}$, devemos obter $\phi(x)$ tal que, dado x^0 construímos a sequência $x^1 = \phi(x^0)$, $x^2 = \phi(x^1)$, \dots , $x^n = \phi(x^{n-1})$, \dots , e $\lim_{n \rightarrow \infty} x^n = \bar{x}$, onde \bar{x} é a solução exata do sistema $AX = B$.

1.3.1 Método Iterativo de Jacobi

Considere um sistema 3×3 :

$$\begin{array}{l} (1) \\ (2) \\ (3) \end{array} \left\{ \begin{array}{l} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = b_3 \end{array} \right.$$

$$\text{De (1) temos que } x_1 = \frac{b_1}{a_{11}} - \frac{a_{12}}{a_{11}}x_2 - \frac{a_{13}}{a_{11}}x_3 \quad (1.2)$$

$$\text{De (2) temos que } x_2 = \frac{b_2}{a_{22}} - \frac{a_{21}}{a_{22}}x_1 - \frac{a_{23}}{a_{22}}x_3 \quad (1.3)$$

$$\text{De (3) temos que } x_3 = \frac{b_3}{a_{33}} - \frac{a_{31}}{a_{33}}x_1 - \frac{a_{32}}{a_{33}}x_2 \quad (1.4)$$

ou

$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \underbrace{\begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \frac{b_3}{a_{33}} \end{bmatrix}}_d + \underbrace{\begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & \frac{a_{13}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 \end{bmatrix}}_F \cdot \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

Agora podemos montar a função de iteração de Jacobi:

$$\phi(x) = d + Fx$$

O caso geral é análogo, ou seja, a matriz F é dada por

$$\begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & -\frac{a_{13}}{a_{11}} & \dots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & -\frac{a_{23}}{a_{22}} & \dots & -\frac{a_{2n}}{a_{22}} \\ -\frac{a_{31}}{a_{33}} & -\frac{a_{32}}{a_{33}} & 0 & \dots & -\frac{a_{3n}}{a_{33}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & -\frac{a_{n3}}{a_{nn}} & \dots & 0 \end{bmatrix}$$

e d é dado por

$$\begin{bmatrix} \frac{b_1}{a_{11}} \\ \frac{b_2}{a_{22}} \\ \frac{b_3}{a_{33}} \\ \vdots \\ \frac{b_n}{a_{nn}} \end{bmatrix}$$

Exemplo 1.3.1. Vamos resolver pelo método de Jacobi o sistema

$$\begin{cases} 2x_1 - x_2 = 1 \\ x_1 + 2x_2 = 3 \end{cases}$$

$$F = \begin{bmatrix} 0 & \frac{1}{2} \\ -\frac{1}{2} & 0 \end{bmatrix} \quad e \quad d = \begin{bmatrix} \frac{1}{2} \\ \frac{3}{2} \end{bmatrix}$$

Começamos com a solução inicial $x^0 = (0, 0)$ e iteramos:

$$x^1 = \phi(x^0) = d + Fx^0 = \begin{bmatrix} \frac{1}{2} \\ \frac{3}{2} \end{bmatrix}$$

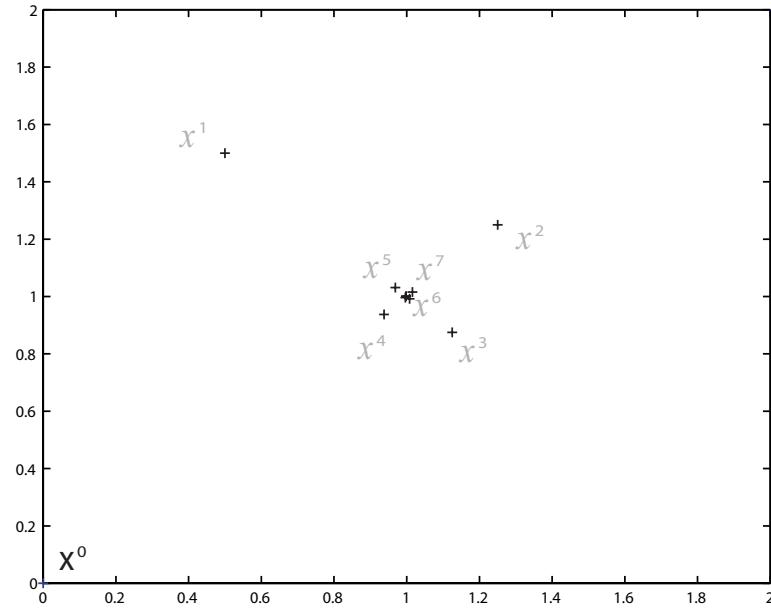
$$x^2 = \phi(x^1) = d + Fx^1 = \begin{bmatrix} \frac{5}{4} \\ \frac{5}{4} \end{bmatrix}$$

$$x^3 = \phi(x^2) = d + Fx^2 = \begin{bmatrix} \frac{9}{8} \\ \frac{7}{8} \end{bmatrix}$$

$$x^4 = \phi(x^3) = d + Fx^3 = \begin{bmatrix} \frac{15}{16} \\ \frac{15}{16} \end{bmatrix}$$

Continuando teremos $x^9 \cong \begin{bmatrix} 0.998 \\ 1.002 \end{bmatrix}$ que é uma boa aproximação da solução exata $\bar{x} = (1, 1)^T$. Poderíamos continuar a seqüência com x^{10}, x^{11}, \dots , onde surge a pergunta: Quando parar? Isso será respondido na próxima seção.

Podemos também observar a aproximação da seqüência x^n para a solução exata \bar{x} por um gráfico. Veja figura 1.3.

Figura 1.3: Seqüência x^n

1.3.2 Critério de Parada

Devemos parar o método iterativo de Jacobi quando, para um dado δ temos que $\|x^n - x^{n-1}\|_\infty < \delta$, onde $\|x^n\|_\infty = \text{Max}\{|x_i|; 1 \leq i \leq n\}$.

Podemos utilizar outras normas para o critério de parada, por exemplo:

$$\|x^n\|_p = \sqrt[p]{\sum_{i=1}^n \|x_i\|^p}, \quad p \in \mathbb{N}$$

Você pode utilizar essa norma, mas por convenção neste texto vamos trabalhar sempre com a norma $\|\cdot\|_\infty$.

Um outro critério de parada é o número de iterações, ou seja, fixado um k produzimos a seqüência (x^n) até o termo x^k . Isso nem sempre resulta em uma boa aproximação.

Exemplo 1.3.2. Considere o Exemplo 1.3.1. Suponha que seja dado

como critério de parada $\delta = 0.6$ assim devemos fazer:

$$\|x^1 - x^0\| = \|x^1\| = \frac{3}{2} > \delta$$

$$\|x^2 - x^1\| = \left\| \begin{pmatrix} \frac{5}{4} \\ \frac{5}{4} \end{pmatrix} - \begin{pmatrix} \frac{1}{2} \\ \frac{3}{2} \end{pmatrix} \right\| = \left\| \begin{pmatrix} \frac{3}{4} \\ -\frac{1}{4} \end{pmatrix} \right\| = \frac{3}{4} > \delta$$

$$\|x^3 - x^2\| = \frac{1}{2} < \delta \text{ parar o método.}$$

Observe que se definimos δ menor, então devemos produzir mais termos da sequência x^n para atingir a precisão desejada.

1.3.3 Método Iterativo de Gauss-Seidel

Este método é muito parecido com o método de Jacobi, na verdade é uma pequena alteração no método de Jacobi que produz o método de Gauss-Seidel.

Relembramos que uma solução aproximada x^k para um sistema $n \times n$ é dada pelo vetor $x^k = (x_1^k, x_2^k, \dots, x_n^k)^T$. Recorde das equações (1.2), (1.3), (1.4) no método de Jacobi e definimos agora a equação geral para uma solução $x^k = (x_1^k, x_2^k, \dots, x_n^k)^T$ e $x^{k-1} = (x_1^{k-1}, x_2^{k-1}, \dots, x_n^{k-1})^T$:

$$x_i^k = \frac{b_i}{a_{ii}} - \frac{1}{a_{ii}} \sum_{j=1, j \neq i}^n a_{ij} x_j$$

Cada elemento da solução x^k depende exclusivamente dos elementos da solução anterior. No método de Gauss-Seidel isso muda um pouco veja:

$$x_1^k = \frac{1}{a_{11}} (b_1 - a_{12}x_2^{k-1} - a_{13}x_3^{k-1} - \dots - a_{1n}x_n^{k-1})$$

$$x_2^k = \frac{1}{a_{22}} (b_2 - a_{21}x_1^k - a_{23}x_3^{k-1} - \dots - a_{2n}x_n^{k-1})$$

\vdots

$$x_n^k = \frac{1}{a_{nn}}(b_n - a_{n1}x_1^k - a_{n2}x_2^k - \cdots - a_{nn-1}x_{n-1}^k)$$

ou

$$x_i^{k+1} = \frac{1}{a_{ii}} \left(b_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i+1}^n a_{ij}x_j^k \right)$$

ou

$$x_i^k = d_i + \left(\sum_{j=1}^{i-1} f_{ij}x_j^k + \sum_{j=i+1}^n f_{ij}x_j^{k-1} \right), \text{ com } i = 1, 2, \dots, n, f_{ij} \text{ e}$$

d_i entradas da matriz F e d dadas no método de Jacobi.

Não abordaremos aqui, mas é possível criar uma função de iteração $\phi(x)$ como a que foi feita no método de Jacobi. Para isso, veja [10].

Exemplo 1.3.3. *Resolva pelo método de Gauss-Seidel o sistema*

$$\begin{cases} 2x_1 - x_2 &= 1 \\ x_1 + 2x_2 &= 3 \end{cases}.$$

Começando com $x^0 = (0, 0)^T$ temos:

$$\text{Equações iterativas de Gauss-Seidel} \begin{cases} x_1^{k+1} = \frac{1}{2}(1 + x_2^k) \\ x_2^{k+1} = \frac{1}{2}(3 - x_1^{k+1}) \end{cases}$$

fazendo $k = 0$ temos:

$$\begin{cases} x_1^1 = \frac{1}{2}(1 + x_2^0) = \frac{1}{2}(1 + 0) = 0.5 \\ x_2^1 = \frac{1}{2}(3 - x_1^1) = \frac{1}{2}(3 - 0.5) = 1.25 \end{cases}$$

fazendo $k = 1$ temos:

$$\begin{cases} x_1^2 = \frac{1}{2}(1 + x_2^1) = \frac{1}{2}(1 + 1.25) = 1.125 \\ x_2^2 = \frac{1}{2}(3 - x_1^2) = \frac{1}{2}(3 - 1.125) = 0.9375 \end{cases}$$

Continuando podemos observar que o método de Gauss-Seidel converge mais rápido que o método de Jacobi.

1.3.4 Método do Refinamento Iterativo

Considere um sistema $n \times n$, $AX = B$ com solução exata \bar{x} e x^0 uma solução aproximada. Assim $\bar{x} = e^0 + x^0$, onde e^0 é o erro cometido. Como $A\bar{x} = b$ então:

$$A(e^0 + x^0) = B \implies Ae^0 + Ax^0 = B \implies Ae^0 = \underbrace{B - Ax^0}_{r_0} \implies Ae^0 = r^0$$

Resolvendo o sistema $Ae^0 = r^0$ teremos uma aproximação de e^0 . Como $\bar{x} = e^0 + x^0$, então e^0 melhora a solução x^0 .

Este processo pode ser repetido até que se obtenha uma precisão desejada.

1.3.5 Número Condicional

Para esta seção, recomendamos antes a leitura do apêndice. Começamos considerando o sistema abaixo:

$$\begin{cases} 2x_1 + 3x_2 = 8 \\ 2x_1 + 3.0005x_2 = 6.001 \end{cases}$$

Se usarmos aritmética com cinco casas decimais, a solução por método de Gauss será $x = (1, 2)^T$. Suponha que somos forçados a usar apenas 3 casas decimais para resolver o sistema abaixo:

$$\begin{cases} 2x_1 + 3x_2 = 8 \\ 2x_1 + 3.001x_2 = 6.001 \end{cases}$$

A solução agora será $x = (2.5, 1)^T$. Observe que os sistemas diferem apenas no coeficiente a_{22} . O erro relativo nesse coeficiente é:

$$\frac{a'_{22} - a_{22}}{a_{22}} = \frac{3.001 - 3.0005}{3.0005} \cong 0.00017$$

no entanto, o erro relativo nas coordenadas das soluções é:

$$\frac{x'_1 - x_1}{x_1} = \frac{2.5 - 1}{1} = 1.5, \quad \frac{x'_2 - x_2}{x_2} = \frac{1 - 2}{2} = -0.5$$

Portanto, uma pequena perturbação(variação) na matriz dos coeficientes gera uma grande mudança na solução. Assim, temos um problema quando vamos resolver um sistema via computador, pois o mesmo efetua truncamentos/arredondamentos; estes geram pequenas mudanças na matriz dos coeficientes o que pode ocasionar em uma solução totalmente equivocada.

Definição 1.2. *Uma matriz A é **mal condicionada** se mudanças relativamente pequenas em seus elementos podem causar mudanças relativamente grandes nas soluções de $Ax = b$. A é **bem condicionada** se mudanças relativamente pequenas em seus elementos resultam em mudanças relativamente pequenas nas soluções de $Ax = b$.*

Dessa forma se A for mal condicionada, a solução do sistema $Ax = b$ não será muito precisa. Por outro lado se A for bem condicionada podemos calcular a solução do sistema com bastante precisão. Dizemos que o sistema é bem condicionado(mal condicionado) se sua matriz dos coeficientes é bem condicionada(mal condicionada).

Visto a necessidade de saber sobre o condicionamento de uma matriz, vamos agora tentar criar uma medida para dizer se A é bem ou mal condicionada.

Seja então, A uma matriz invertível $n \times n$ e considere um sistema $Ax = b$. Se x é a solução exata do sistema e x' uma solução calculada(via algum método), o erro pode ser representado pelo vetor $e = x - x'$. Veja que $\|e\|$ é o erro absoluto e $\frac{\|e\|}{\|x\|}$ o erro relativo. Em geral não temos como calcular esses erros, uma vez que não conhecemos a solução exata do sistema. Assim para testar a precisão da solução x' devemos colocar x' no sistema original e observar o quão próximo está $b' = Ax'$ de b , ou seja, $r = b - b' = b - Ax'$, onde r é chamado de resíduo e $\frac{\|r\|}{\|b\|}$ o resíduo relativo.

A pergunta agora é a seguinte: Será que o resíduo relativo é uma boa estimativa para o erro relativo? A resposta depende do condicionamento da matriz A . No exemplo anterior temos:

$$r = b - Ax' = (0, 0.0005)^T$$

o resíduo relativo

$$\frac{\|r\|}{\|b\|} = \frac{0.0005}{8.001} \cong 0.000062$$

o erro relativo

$$\frac{\|e\|}{\|x\|} = \frac{1.5}{2} = 0.75$$

Veja que o erro relativo é quase 12096 vezes o resíduo relativo.

Vamos mostrar que em geral para matrizes mal condicionadas o resíduo relativo é muito menor que o erro relativo, ou seja, se o resíduo relativo está próximo de zero não podemos dizer que a solução x' é boa ou não. Por outro lado, para matrizes bem condicionadas o resíduo relativo e o erro relativo estão bastante próximos. Para mostrar isso vamos usar a norma matricial. Seja A uma matriz $m \times n$ a norma de A será:

$$\|A\| = \max_{1 \leq i \leq m} \left\{ \sum_{j=1}^n |a_{ij}| \right\}$$

Observe que $\|v\| = |v|$ se v é um vetor coluna. E nessa norma vale $\|AB\| \leq \|A\| \cdot \|B\|$.

Assim temos:

$$r = b - Ax' = Ax - Ax' = Ae$$

e portanto

$$e = A^{-1}r$$

mais ainda,

$$\|e\| \leq \|A^{-1}\| \cdot \|r\| \quad e \quad \|r\| \leq \|A\| \cdot \|e\|$$

o que implica em

$$\frac{\|r\|}{\|A\|} \leq \|e\| \leq \|A^{-1}\| \cdot \|r\| \quad (1.5)$$

do fato de que $Ax = b$ e $x = A^{-1}b$ temos:

$$\|b\| \leq \|A\| \cdot \|x\| \quad e \quad \|x\| \leq \|A^{-1}\| \cdot \|b\|$$

o que implica em

$$\frac{\|b\|}{\|A\|} \leq \|x\| \leq \|A^{-1}\| \cdot \|b\|. \quad (1.6)$$

Dividindo a inequação 1.5 pela inequação 1.6 temos:

$$\frac{\|b\|}{\|r\|} \leq \frac{\|e\|}{\|x\|} \leq \frac{\|b\|}{\|r\|}. \quad (1.7)$$

Veja que $1 = \|I\| = \|AA^{-1}\| \leq \|A\| \cdot \|A^{-1}\|$, logo a desigualdade 1.7 pode ser escrita como:

$$\frac{1}{\|A\| \cdot \|A^{-1}\|} \frac{\|b\|}{\|r\|} \leq \frac{\|e\|}{\|x\|} \leq \frac{\|b\|}{\|r\|} \|A\| \cdot \|A^{-1}\|,$$

o número $\|A\| \cdot \|A^{-1}\|$ é chamado de **número condicional** de A , denotado por $\text{cond}(A)$, assim,

$$\frac{1}{\text{cond}(A)} \frac{\|b\|}{\|r\|} \leq \frac{\|e\|}{\|x\|} \leq \frac{\|b\|}{\|r\|} \text{cond}(A).$$

Essa última desigualdade mostra que; se $\text{cond}(A)$ está próximo de 1 temos o resíduo relativo próximo do erro relativo. Mas se $\text{cond}(A)$ for grande temos uma diferença entre o resíduo relativo e o erro relativo. Dessa forma quanto mais próximo de 1 o número condicional de uma matriz estiver melhores serão os resultados via método numéricos.

Exemplo 1.3.4. Seja $A = \begin{bmatrix} 3 & 3 \\ 4 & 5 \end{bmatrix}$ e $A^{-1} = \begin{bmatrix} 5/3 & -1 \\ -4/3 & 1 \end{bmatrix}$.

Temos $\|A\| = 9$ e $\|A^{-1}\| = \frac{8}{3}$, logo $\text{cond}(A) = 9 \cdot \frac{8}{3} = 24$. Teoricamente o erro relativo na solução calculada para um sistema $Ax = b$ via métodos numéricos pode ser 24 vezes maior que o resíduo relativo.

Exemplo 1.3.5. Suponha que $x' = (2, 0.1)$ seja a solução calculada para o sistema $\begin{cases} 2x_1 + 3x_2 = 8 \\ 2x_1 + 3.0005x_2 = 6.001 \end{cases}$. Vamos determinar o resíduo r e o resíduo relativo $\frac{\|r\|}{\|b\|}$:

$$r = \begin{bmatrix} 6 \\ 9 \end{bmatrix} - \begin{bmatrix} 3 & 3 \\ 4 & 5 \end{bmatrix} \cdot \begin{bmatrix} 2 \\ 0.1 \end{bmatrix} = \begin{bmatrix} -0.3 \\ 0.5 \end{bmatrix}$$

assim,

$$\frac{\|r\|}{\|b\|} = \frac{0.5}{8} = \frac{1}{18};$$

observe que, se não soubéssemos que a solução exata deste sistema é $(1, 1)^T$ e $\text{cond}(A) = 24$ seríamos levados a pensar que a solução $(2, 0.1)^T$ é uma boa solução, uma vez que o resíduo relativo $\frac{\|r\|}{\|b\|}$ está próximo de zero.

Fica claro que o número condicional de uma matriz invertível nos dá uma informação preciosa a respeito da propagação de erros durante a execução dos métodos numéricos.

1.3.6 Convergência do M. Iterativo de Jacobi e Gauss-Seidel

A primeira vista o método de Jacobi e Gauss-Seidel seriam ideais para resolver sistemas lineares $n \times n$ (que possuem solução). A má notícia é que, nem todos os sistemas $n \times n$ podem ser resolvidos com esses métodos. Existe uma condição de convergência que é crucial. Ela será dada nos teoremas abaixo.

Teorema 1.3. *Se para cada i fixo, $i = 1, \dots, n$ temos que $\sum_{j=1}^n |f_{ij}| \leq L < 1$, então o método iterativo de Jacobi e Gauss-Seidel convergem para a solução exata do sistema.*

Em outras palavras, o teorema diz que: Se a soma em módulo de cada linha da matriz F for menor que 1, então o método de Jacobi e Gauss-Seidel convergem para a solução exata do sistema, seja qual for a solução inicial x^0 .

Esse critério de linhas também pode ser estendido para colunas. Veja o

Teorema 1.4. *Se para cada j fixo, $j = 1, \dots, n$ temos que $\sum_{i=1}^n |f_{ij}| \leq L < 1$, então o método iterativo de Jacobi e Gauss-Seidel convergem para a solução exata do sistema.*

Corolário 1.1. A condição $\sum_{j=1}^n |f_{ij}| \leq L < 1$ no Teorema 1.3 é equivalente a $|a_{ij}| > \sum_{j=1, j \neq i}^n |a_{ij}|$ para cada $i = 1, \dots, n$ fixo.

A demonstração desses teoremas pode ser encontrada em [1].

Se F satisfaz um dos teoremas acima, então podemos aplicar tanto o método de Jacobi quanto o método de Gauss-Seidel, pois é garantida a convergência.

1.4 Sistemas Lineares Complexos

Consideremos uma sistema $AX = B$ onde A, X e B são matrizes complexas. Então são escritas na forma:

$$\begin{aligned} A &= M + Ni \\ B &= c + di \\ X &= s + ti \end{aligned} \tag{1.8}$$

onde M, N, c, d, s, t são matrizes reais. Substituindo a equação (1.8) em $AX = B$ teremos:

$$(M + Ni)(s + ti) = c + di \implies Ms - Nt + (Ns + Mt)i = c + di,$$

ou seja,

$$\begin{cases} Ms - Nt = c \\ Ns + Mt = d \end{cases}$$

O sistema acima pode ser visto como:

$$\left[\begin{array}{cc|cc} M & & & -N \\ & & & \\ \hline & & & \\ & & & \\ N & & & M \end{array} \right] \cdot \begin{bmatrix} s \\ t \end{bmatrix} = \begin{bmatrix} c \\ d \end{bmatrix} \tag{1.9}$$

que é um sistema real, e pode ser resolvido com os métodos apresentados anteriormente.

Exemplo 1.4.1. *Resolva o sistema*
$$\begin{cases} (1+2i)x_1 + 3x_2 = -5+4i \\ -x_1 + x_2 = -1 \end{cases}$$

Vamos decompor a matriz dos coeficientes:

$$A = \begin{bmatrix} 1+2i & 3+0i \\ -1+0i & 1+0i \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 3 \\ -1 & 1 \end{bmatrix}}_M + \underbrace{\begin{bmatrix} 2 & 0 \\ 0 & 0 \end{bmatrix}}_N i$$

$$B = \begin{bmatrix} -5+4i \\ -1+0i \end{bmatrix} = \underbrace{\begin{bmatrix} -5 \\ -1 \end{bmatrix}}_c + \underbrace{\begin{bmatrix} 4 \\ 0 \end{bmatrix}}_d i$$

$$X = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \underbrace{\begin{bmatrix} s_1 \\ s_2 \end{bmatrix}}_s + \underbrace{\begin{bmatrix} t_1 \\ t_2 \end{bmatrix}}_t i$$

Escrevendo o sistema na forma (1.9) temos:

$$\begin{bmatrix} 1 & 3 & -2 & 0 \\ -1 & 1 & 0 & 0 \\ 2 & 0 & 1 & 3 \\ 0 & 0 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} s_1 \\ s_2 \\ t_1 \\ t_2 \end{bmatrix} = \begin{bmatrix} -5 \\ -1 \\ 4 \\ 0 \end{bmatrix}$$

Resolvendo o sistema por métodos anteriores (por exemplo pelo método de Gauss) obtemos: $s_1 = 0, s_2 = -1, t_1 = 1, t_2 = 1$, dessa forma a solução do sistema é dada por $x_1 = i, x_2 = -1 + i$.

1.5 Exercícios

1.5.1. *Através de operações elementares mostre que a matriz*
$$\begin{bmatrix} 3 & 5 & 0 \\ 2 & 0 & 1 \\ 5 & 1 & -1 \end{bmatrix}$$
é equivalente à
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

1.5.2. *Mostre geometricamente que o sistema abaixo não possui solução real:*

$$\begin{cases} 2x_1 + 6x_2 = 8 \\ 3x_1 + 9x_2 = 15 \\ x_1 + 3x_2 = 6 \end{cases}$$

1.5.3. *Resolva os sistemas abaixo por retro substituição:*

$$a) \begin{cases} x_1 - 3x_2 + x_3 = 6 \\ 4x_2 - x_3 = 5 \\ x_3 = 4 \end{cases} \quad b) \begin{cases} 2x_1 = 2 \\ x_1 + x_2 = 3 \\ x_1 + x_2 + x_3 = 4 \end{cases}$$

1.5.4. *Resolva pelo método de Gauss os sistemas:*

$$a) \begin{cases} x_1 + x_2 + x_3 + x_4 = 0 \\ x_1 + x_2 + x_3 - x_4 = 4 \\ x_1 + x_2 - x_3 + x_4 = -4 \\ x_1 - x_2 + x_3 + x_4 = 2 \end{cases} \quad b) \begin{cases} x_1 + 2x_2 + x_3 = 0 \\ 2x_1 + x_2 + 3x_3 = 0 \\ 3x_1 + 2x_2 + x_3 = 0 \end{cases}$$

1.5.5. *Mostre que:*

$$\begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ 4x_1 + 4x_2 - 3x_3 = 3 \\ 2x_1 - 3x_2 + x_3 = -1 \end{cases} \text{ é equivalente a } \begin{cases} 2x_1 + 3x_2 - x_3 = 5 \\ -2x_2 - x_3 = -7 \\ -6x_2 + 2x_3 = -6 \end{cases}$$

1.5.6. *Através de decomposição LU obtenha a solução dos sistemas do exercício 1.5.4.*

1.5.7. *Calcule a matriz inversa A^{-1} da matriz abaixo. Para isso, use decomposição LU.*

$$\begin{bmatrix} 1 & 2 & -1 \\ 2 & 3 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

1.5.8. *Resolva através do método iterativo de Jacobi os sistemas abaixo com $\delta = 0.7$:*

$$a) \begin{cases} 10x_1 + x_2 + x_3 &= 12 \\ 2x_1 + 4x_2 + x_3 &= 7 \\ x_1 - x_2 + 3x_3 &= 3 \end{cases} \quad b) \begin{cases} 2x_1 + x_2 &= 2 \\ x_1 + 3x_2 &= 1 \end{cases}$$

1.5.9. *Resolva os sistemas do exercício 1.5.8 pelo método de Gauss-Seidel com $\delta = 0.7$.*

1.5.10. *Explique por que o método de Gauss-Seidel não garante convergência para solução exata do sistema abaixo;*

$$\begin{cases} x_1 + x_2 + x_3 &= 3 \\ 2x_1 + x_2 + x_3 &= 4 \\ 3x_1 + 2x_2 + x_3 &= 6 \end{cases}$$

1.5.11. *Aplique o método do Refinamento Iterativo na solução do exercício 1.5.8 com uma iteração. Verifique se a nova solução melhora a anterior.*

1.5.12. *Dê exemplos de sistemas mal condicionados.*

1.5.13. Sejam $A = \begin{bmatrix} 3 & 2 \\ 1 & 1 \end{bmatrix}$ e $b = \begin{bmatrix} 5 \\ 2 \end{bmatrix}$. A solução do sistema $Ax = b$ calculada com duas casas decimais é $x = (1.1, 0.88)^T$.

a) Determine o vetor resíduo r e o valor do resíduo relativo $\frac{\|r\|}{\|b\|}$.

b) Encontre o valor de $\text{cond}(A)$.

c) Sem calcular a solução exata, use os resultados encontrados em a) e b) para obter cotas para o erro relativo na solução calculada.

d) Calcule a solução exata x e determine o erro relativo. Compare esse valor com as cotas encontradas no item c).

1.5.14. Sejam A e B matrizes invertíveis $n \times n$. Mostre que

$$\text{cond}(AB) \leq \text{cond}(A)\text{cond}(B).$$

1.5.15. Mostre que, se A é uma matriz invertível $n \times n$ e $\text{adj}(A)$ é a matriz adjunta de A temos:

$$\|\text{adj}(A)\| = \max_{1 \leq i \leq n} \left\{ \prod_{j=1, j \neq i}^n |a_{jj}| \right\}.$$

1.5.16. Seja A uma matriz invertível diagonal $n \times n$. Mostre que

$$\text{cond}(A) = \frac{\max_{1 \leq j \leq n} |a_{jj}|}{\min_{1 \leq j \leq n} |a_{jj}|}$$

1.5.17. Resolva o sistema:

$$\begin{cases} -2ix_1 + 3x_2 &= 2 + 5i \\ (1+i)x_1 + ix_2 &= -3 \end{cases}$$

1.5.18. Considere a tabela de valores nutricionais² dos alimentos abaixo:

Alimento	Vitamina A	Vitamina B	Vitamina C
1-Pera	1g	3g	4g
1-Uva	2g	3g	5g
1-Maça	3g	2g	3g

Deseja-se saber quanto de cada alimento deve-se ingerir para obter 11g de vitamina A, 13g de vitamina B e 20g de vitamina C em uma dieta diária.

Monte um sistema e resolva pelo método de Gauss esse problema.

1.5.19. Descreva o método de resolução de sistemas lineares complexos $n \times n$ através de redução à sistemas lineares reais. Verifique com isto, que um sistema $n \times n$ complexo é transformado em um sistema real $(2n) \times (2n)$.

1.5.20. Demonstre o Corolário 1.1.

1.5.21. Monte um sistema que tenha como solução $x_1 = 1$, $x_2 = 2$, $x_3 = 0$, $x_4 = 1$.

²valores fictícios

Capítulo 2

Zeros de função

2.1 Introdução

Neste capítulo estamos interessados em obter os zeros de uma função real através de métodos numéricos. Em outras palavras, dada uma certa função $f(x)$, gostaríamos de encontrar ϵ tal que, $f(\epsilon) = 0$. Por exemplo, a função $f(x) = x^2 - 3x + 2$ tem dois zeros, $\epsilon_1 = 2$ e $\epsilon_2 = 1$. Vale lembrar que uma função pode ter zeros reais ou complexos. Neste texto não vamos tratar o caso complexo, o leitor interessado pode encontrar em [11].

Para começar, vamos enunciar algumas definições e teoremas. Para o nosso estudo não será necessário demonstrá-los.

Definição 2.1. Dizemos que uma função $f : I \rightarrow \mathbb{R}$ é contínua em um ponto c , se para toda seqüência (x_n) em I , tivermos

$$\lim_{n \rightarrow \infty} x_n = c \implies \lim_{n \rightarrow \infty} f(x_n) = f(c)$$

A grosso modo podemos entender essa definição da seguinte forma:

Dizemos que $f(x)$ é contínua, se ao traçar seu gráfico não levantamos o lápis do papel.

Teorema 2.1. Seja $f(x)$ uma função contínua definida no intervalo $[a, b]$ tal que $f(a)f(b) < 0$, então $f(x)$ possui pelo menos um zero em $[a, b]$.

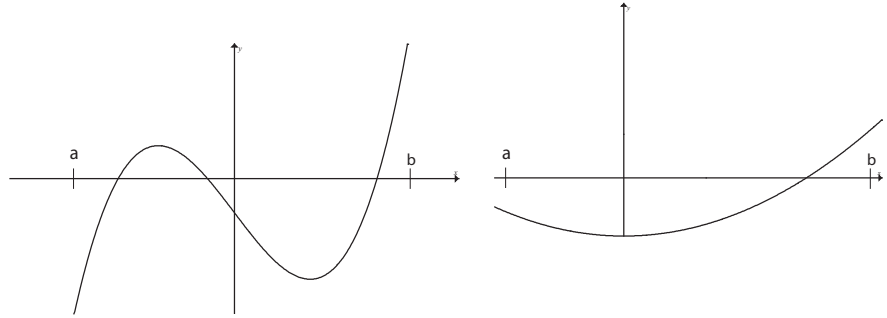
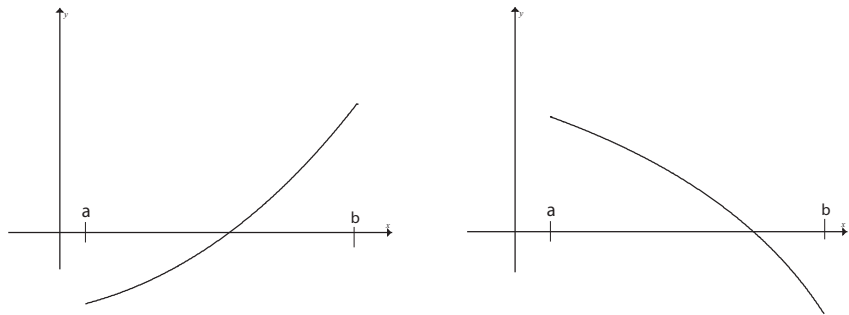


Figura 2.1: Zeros de uma função

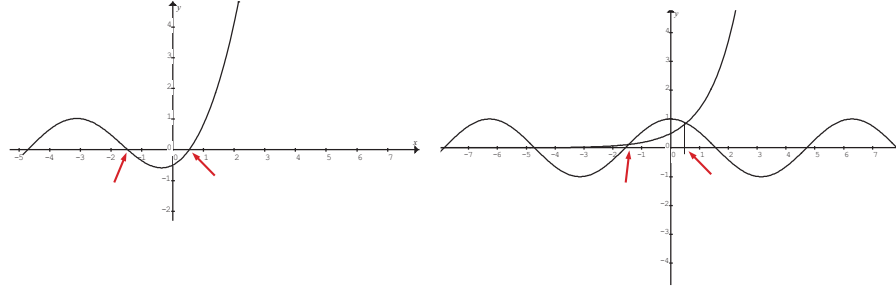
Figura 2.2: $f'(x) > 0$ e $f'(x) < 0$

Esse teorema nos diz que, se $f(x)$ é contínua e $f(a)$ tem sinal diferente de $f(b)$, então $f(x)$ corta o eixo x pelo menos uma vez, ou seja, existe $\epsilon \in [a, b]$ tal que $f(\epsilon) = 0$. Veja a figura 2.1. Este teorema é consequência direta do teorema do valor intermediário estudado nos cursos de Cálculo.

Teorema 2.2. *Seja $f(x)$ uma função definida no intervalo $[a, b]$ tal que $f(a)f(b) < 0$ e $f'(x) > 0$ para todo $x \in (a, b)$. Então $f(x)$ possui um único zero em $[a, b]$.*

O Teorema 2.1 garante apenas a existência mas não a unicidade. Acrescentando a hipótese da derivada ser positiva em todo intervalo (o que implica em $f(x)$ ser crescente) ¹(veja figura 2.2) obtemos o

¹também vale para negativa

Figura 2.3: Zeros de $f(x)$

Teorema 2.2 que garante a unicidade desse zero.

Teorema 2.3. *Se $f(x)$ pode ser escrita como diferença de duas funções, digamos $g(x)$ e $h(x)$, então os zeros de $f(x)$ são exatamente os pontos de interseção de $g(x)$ e $h(x)$.*

Veja que se $f(x) = g(x) - h(x)$ e ϵ é um zero de $f(x)$, então $0 = f(\epsilon) = g(\epsilon) - h(\epsilon) \implies g(\epsilon) = h(\epsilon)$

Exemplo 2.1.1. *Considere $f(x) = \frac{1}{2}e^x - \cos(x)$ com $g(x) = \frac{1}{2}e^x$ e $h(x) = \cos(x)$. Observe a figura 2.3. Veja que $g(x)$ e $h(x)$ possuem varias interseções. Cada interseção é um zero de $f(x)$.*

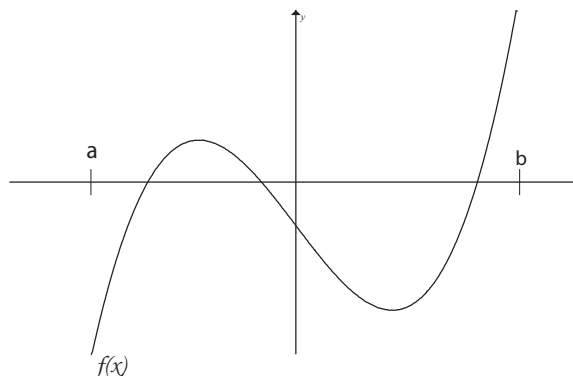
Definição 2.2. *Diz-se que uma seqüência (x_n) é de Cauchy quando, para todo $\epsilon > 0$ dado, existe $n_0 \in \mathbb{N}$ tal que $m, n > n_0 \implies |x_m - x_n| < \epsilon$*

Em outras palavras, uma seqüência é de *Cauchy* se seus termos estão cada vez mais próximos uns dos outros.

Teorema 2.4. *Uma seqüência (x_n) real converge se, e somente se, é de Cauchy.*

2.2 Método de Localização de Zeros

Nesta seção vamos desenvolver um método para isolar os zeros de uma função em intervalos. Em outras palavras, obter intervalos onde existe um único zero.

Figura 2.4: Zeros de $f(x)$

Consideremos uma função $f(x)$ cujo gráfico seja dado pela figura 2.4. Gostaríamos de encontrar uma partição do intervalo $[a, b]$ digamos $\mathbb{P} = \{p_1, p_2, p_3, \dots, p_n\}$, onde $p_j = p_{j-1} + \gamma$ para algum γ escolhido. Tal que, cada zero esteja em um subintervalo (p_{k-1}, p_k) criado pela partição. Veja exemplo abaixo:

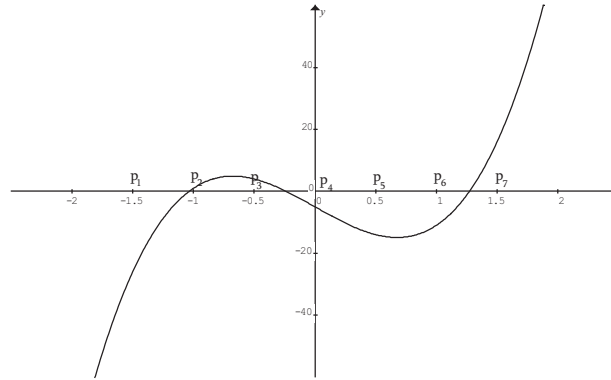
Exemplo 2.2.1. Considere $f(x) = 16x^3 - 22x - 5$ (figura 2.5) no intervalo $[-2, 2]$. Tomamos uma partição com $\gamma = \frac{1}{2}$ e obtemos a tabela:

\mathbb{P}	-2	$-\frac{3}{2}$	-1	$-\frac{1}{2}$	0	$\frac{1}{2}$	1	$\frac{3}{2}$	2
$f(\mathbb{P})$	< 0	< 0	> 0	> 0	< 0	< 0	< 0	> 0	> 0

Portanto, aplicando o Teorema 2.1 existem zeros em $[-\frac{3}{2}, -1]$, $[-\frac{1}{2}, 0]$ e $[1, \frac{3}{2}]$.

É bom lembrar que quanto mais fina a partição, ou seja, quanto menor o γ melhores são as chances de isolar todos os zeros da função.

Em geral não sabemos onde estão os zeros da função $f(x)$, simplesmente criamos uma partição e observamos a mudança de sinal de acordo com o Teorema 2.1 para encontrar os intervalos onde $f(x)$ possui zeros. O próximo passo é encontrar uma aproximação para esses zeros. Isso será feito nas próximas seções.

Figura 2.5: Criando uma partição \mathbb{P}

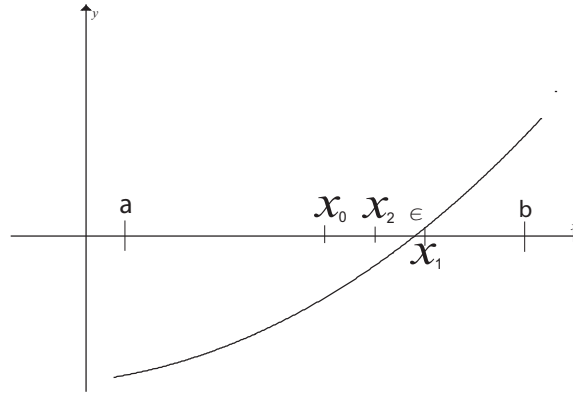
2.3 Método do Meio Intervalo - MMI

Considere uma função $f(x)$ contínua no intervalo $[a, b]$ tal que $f(a)f(b) < 0$ e $f(x)$ possua um único zero nesse intervalo. O MMI consiste em efetuar sucessivas divisões no intervalo $[a, b]$ de forma que o zero fique dentro algum subintervalo (bem pequeno) criado pelas divisões. Ao contrário do Método de Localização de Zeros, agora estamos interessados na aproximação do zero, não do intervalo. Acompanhe a explicação abaixo com a figura 2.6.

Inicialmente dividimos o intervalo $[a, b]$ ao meio em $x_0 = \frac{a+b}{2}$ e verificamos se $f(x_0) = 0$, caso contrário analisamos o sinal de $f(a)f(x_0)$ e $f(x_0)f(b)$.

Suponhamos que $f(a)f(x_0) > 0$ e $f(x_0)f(b) < 0$. Assim pelo Teorema 2.1 existe um zero ϵ em (x_0, b) . Descartamos o intervalo $[a, x_0]$. Dividimos o intervalo (x_0, b) ao meio em $x_1 = \frac{x_0+b}{2}$ e verificamos se $f(x_1) = 0$, caso contrário, repetimos o processo verificando o sinal de $f(x_0)f(x_1)$ e $f(x_1)f(b)$. Suponhamos que $f(x_0)f(x_1) < 0$ e $f(x_1)f(b) > 0$. Descartamos o intervalo (x_1, b) e dividimos $[x_0, x_1]$ em x_2 . O processo segue até que $|x_n - x_{n-1}|$ seja menor ou igual que um certo δ fixado.

Na figura 2.6 verificamos que se continuarmos o método a sequência (x_n) converge para o zero ϵ . Mostraremos agora uma prova rigorosa de que a sequência (x_n) realmente converge para ϵ , ou seja,

Figura 2.6: (x_n) convergindo para o zero ϵ

$$\lim_{n \rightarrow \infty} x_n = \epsilon.$$

Para isso, tome somente os intervalos aproveitados:

$$[a, b], [x_0, b], [x_0, x_1], \dots, [x_n, x_k], \dots$$

renomeando para

$$[a_0, b_0], [a_1, b_1], [a_2, b_2], \dots, [a_n, b_n], \dots$$

Veja que $[a_0, b_0] \supset [a_1, b_1] \supset [a_2, b_2] \supset \dots \supset [a_n, b_n] \supset \dots$, com $f(a_n)f(b_n) < 0$ e ambas as seqüências (a_n) e (b_n) limitadas. Outro fato é que $a_0 \leq a_1 \leq a_2 \leq \dots \leq a_n \leq \dots$ e $b_0 \geq b_1 \geq b_2 \geq \dots \geq b_n \geq \dots$, onde concluímos que essas seqüências são monótonas e limitadas. Pelo Teorema de Bolzano-Weierstrass² temos que (a_n) e (b_n) convergem para um certo L , ou seja,

$$\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = L.$$

Ainda resta mostrar que L é exatamente o zero ϵ . Para isso calcule o limite na desigualdade $f(a_n)f(b_n) < 0$ e use o fato de $f(x)$ ser continua. Temos então:

$$\lim_{n \rightarrow \infty} (f(a_n)f(b_n)) < 0$$

²Veja livro de cálculo ou análise matemática

$$\lim_{n \rightarrow \infty} f(a_n) \lim_{n \rightarrow \infty} f(b_n) < 0$$

$$f\left(\lim_{n \rightarrow \infty} a_n\right) f\left(\lim_{n \rightarrow \infty} b_n\right) < 0$$

$$f(L)f(L) < 0$$

$$f(L)^2 < 0 \implies f(L) = 0$$

Como ϵ é o único zero em $[a \ b]$ então $\epsilon = L$.

Exemplo 2.3.1. Encontre o zero de $f(x) = x^2 - 2$ no intervalo $[0 \ 2]$ com $\delta = 0.07$.

n	a_n	b_n	x_n	$ x_n - x_{n-1} $	
0	0	2	1	$1 > \delta$	
1	1	2	1.5	$0.5 > \delta$	
2	1	1.5	1.25	$0.25 > \delta$	
3	1.25	1.5	1.37	$0.12 > \delta$	
4	1.37	1.5	1.43	$0.06 \leq \delta$	← parar o método

Assim nossa aproximação para o zero de $f(x)$ seria $x_4 = 1.43$

2.4 Método da Secante

O método da Secante³ é parecido com o MMI no sentido de criar uma seqüência que se aproxima do zero procurado.

Antes de aplicar o método devemos observar como a função $f(x)$ se comporta no intervalo. Pois como veremos a concavidade muda o modo como iremos criar a seqüência (x_n) .

Vamos então as hipóteses do método. Seja $f(x)$ uma função duas vezes diferenciável em $[a \ b]$ e $f(a)f(b) < 0$. Suponhamos como primeiro caso que:

$$f'(x) > 0 \text{ e } f''(x) > 0, \text{ para todo } x \in [a \ b]$$

³também chamado de método das cordas

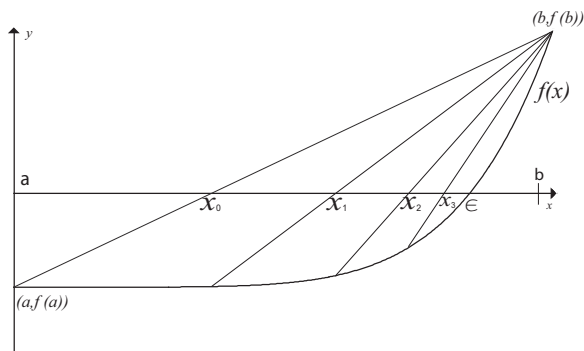


Figura 2.7: Seqüência (x_n) no método da secante

Com essas hipóteses temos que o gráfico de $f(x)$ em $[a, b]$ tem a forma da figura 2.6. O método da secante consiste em tomar a reta (digamos L_0) que passa pelos pontos $(a, f(a)), (b, f(b))$. O primeiro elemento da seqüência (x_n) , ou seja, o elemento x_0 será a interseção dessa reta com o eixo x , veja figura 2.7.

A segunda reta (digamos L_1) passa pelos pontos $(b, f(b)), (x_0, f(x_0))$. A interseção dessa reta com o eixo x gera o elemento x_1 . O processo segue até que $|x_n - x_{n-1}|$ seja menor ou igual que um certo δ fixado.

Ainda não conhecemos bem os elementos da seqüência x_n , por isso vamos começar determinando x_0 . Lembramos que a equação de uma reta que passa por dois pontos quaisquer $(x, y), (s_0, t_0)$ e dada por:

$$y - t_0 = m(x - s_0),$$

onde m é o coeficiente angular da reta.

O coeficiente angular da reta L_0 é dado por $m_{L_0} = \frac{f(b)-f(a)}{b-a}$. Assim a equação da reta L_0 é:

$$y - f(b) = \frac{f(b) - f(a)}{b - a}(x - b) \implies y = f(b) + \frac{f(b) - f(a)}{b - a}(x - b)$$

x_0 é o ponto onde essa reta corta o eixo x , ou seja, $y = 0$, então

$$0 = f(b) + \frac{f(b) - f(a)}{b - a}(x_0 - b) \implies x_0 = b - \frac{f(b)}{f(b) - f(a)}(b - a)$$

Como o mesmo procedimento aplicado na reta L_1 obtemos o ponto x_1 :

$$x_1 = b - \frac{f(b)}{f(b) - f(x_0)}(b - x_0)$$

Portanto a seqüência x_n será dada por:

$$\boxed{x_n = b - \frac{f(b)}{f(b) - f(x_{n-1})}(b - x_{n-1})} \quad (2.1)$$

Veja exercício 2.9.5 para outros casos.

2.4.1 Convergência no Método da Secante

Podemos afirmar que a seqüência (x_n) é monótona e limitada. Dessa forma:

$$\lim_{n \rightarrow \infty} x_n = L.$$

Passando o limite na equação (2.1) obtemos:

$$\lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} \left(b - \frac{f(b)}{f(b) - f(x_{n-1})}(b - x_{n-1}) \right)$$

assim

$$L = b - \frac{f(b)}{f(b) - f(L)}(b - L) \implies f(L) = 0$$

Como ϵ é o único zero em $[a, b]$ então $\epsilon = L$.

2.5 Método de Newton

Considere para o método de Newton as mesmas hipóteses para $f(x)$ como no método da secante, ou seja, $f(x)$ uma função duas vezes diferenciável em $[a, b]$ e $f(a)f(b) < 0$. Suponhamos como primeiro caso que:

$$f'(x) > 0 \text{ e } f''(x) > 0, \text{ para todo } x \in [a, b]$$

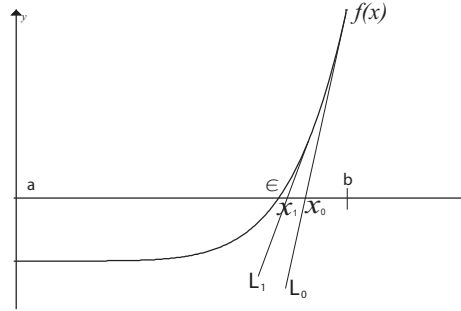


Figura 2.8: Seqüência (x_n) no método de Newton

Com essas hipóteses temos que o gráfico de $f(x)$ em $[a, b]$ tem a forma da figura 2.6. O método de Newton difere pouco do método da secante. Agora em vez da reta secante, tomamos a reta (digamos L_0) tangente no ponto $(b, f(b))$, ou seja, essa reta tem coeficiente angular $f'(b)$. Veja figura 2.8.

Observe que x_0 é a interseção de L_0 com o eixo x e x_1 é a interseção da reta L_1 com o eixo x . Como no método da secante vamos determinar o ponto x_0 . Para isso usaremos o fato do coeficiente angular m_{L_0} ser igual a $f'(b)$. Dessa forma a equação da reta L_0 fica sendo:

$$y - f(b) = f'(b)(x - b) \implies y = f(b) + f'(b)(x - b)$$

Como x_0 é o ponto onde essa reta corta o eixo x , ou seja, $y = 0$ então:

$$x_0 = b - \frac{f(b)}{f'(b)}$$

O mesmo procedimento é aplicado em L_1 e obtemos o ponto

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

o que implica no caso geral

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \quad (2.2)$$

2.5.1 Convergência no Método de Newton

Podemos afirmar que a sequência (x_n) é monótona e limitada. Dessa forma:

$$\lim_{n \rightarrow \infty} x_n = L.$$

Passando o limite na equação (2.2) obtemos:

$$\lim_{n \rightarrow \infty} x_n = \lim_{n \rightarrow \infty} \left(x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})} \right)$$

assim

$$L = L - \frac{f(L)}{f'(L)} \implies f(L) = 0$$

Como ϵ é o único zero em $[a, b]$ então $\epsilon = L$.

Exemplo 2.5.1. *Vamos aplicar o método de Newton no Exemplo 2.3.1.*

n	x_n	$ x_n - x_{n-1} $	
0	1.5	$1.5 > \delta$	
1	1.416	$0.08 > \delta$	
2	1.414	$0.002 \leq \delta$	← parar o método

Assim nossa aproximação para o zero de $f(x)$ seria $x_2 = 1.414$.

Uma observação muito importante é o fato do método de Newton atingir δ com $n = 2$, ao passo que no MMI, $n = 4$. Isso não acontece por acaso, o método de Newton realmente converge mais rápido. Nunca é demais lembrar que o M. de Newton exige que mais hipóteses sobre a função do que o MMI.

2.6 Método da Iteração Linear

Começamos esse método obtendo a função de iteração $\phi(x)$. Para isso isolamos a variável x na equação $f(x) = 0$. Tome como exemplo $f(x) = x^2 - 5x + 6$:

$$\begin{aligned} x^2 - 5x + 6 &= 0 \\ x^2 - x - 4x + 6 &= 0 \\ x^2 - 4x + 6 &= x \end{aligned}$$

Assim $\phi(x) = x^2 - 4x + 6$.

Veja agora que o ponto fixo (veja definição adiante) de $\phi(x)$ é justamente o zero de $f(x)$, ou seja, k tal que $\phi(k) = k$ implica em $f(k) = 0$.

Na verdade, o que fizemos foi transformar o problema de encontrar zeros, para o problema de encontrar ponto fixo.

Definição 2.3. Diz-se que k é ponto fixo de $\phi(x)$, se $\phi(k) = k$.

A seqüência (x_n) será criada a partir de uma aproximação inicial x_0 (mais a frente comentaremos essa escolha) da seguinte forma:

$$x_1 = \phi(x_0), x_2 = \phi(x_1), \dots, x_n = \phi(x_{n-1}), \dots$$

Observe nas figuras (2.9) e (2.10) a seqüência (x_n) .

Como nem tudo são flores! O método da Iteração Linear nem sempre pode ser usado, conforme o teorema abaixo.

Teorema 2.5. Se $|\phi'(x)| \leq \lambda < 1$ para todo $x \in [a, b]$, então para qualquer valor inicial $x_0 \in [a, b]$ a seqüência $x_1 = \phi(x_0), x_2 = \phi(x_1), \dots, x_n = \phi(x_{n-1}), \dots$ converge para um certo L que é o ponto fixo de $\phi(x)$ em $[a, b]$.

demonstração

Consideremos o primeiro caso (figura 2.9) com $x_{n-1} < x_n$. Pelo teorema do valor médio⁴ existe $\omega_n \in (x_{n-1}, x_n)$ tal que:

$$|\phi(x_n) - \phi(x_{n-1})| = |\phi'(\omega_n)| |x_n - x_{n-1}|,$$

⁴Veja qualquer livro de cálculo

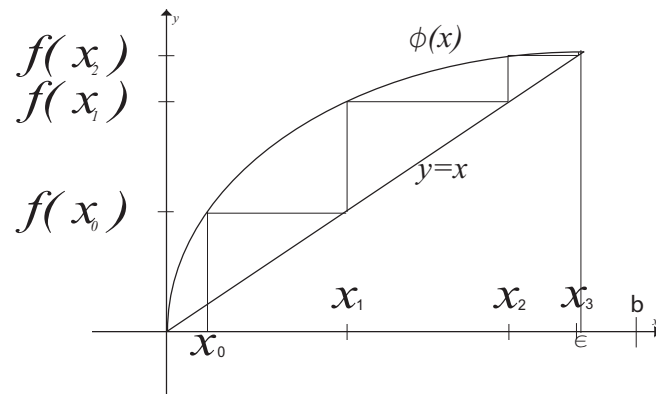


Figura 2.9: Seqüência (x_n) no método da Iteração Linear

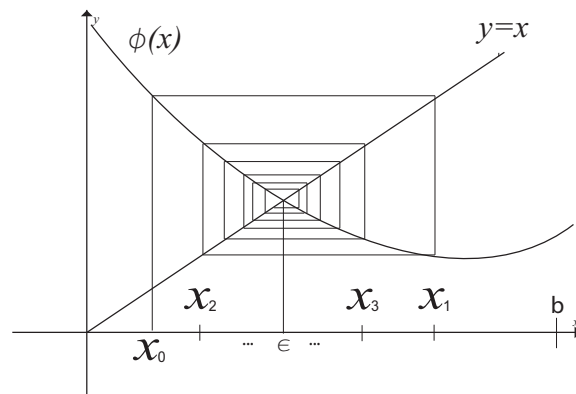


Figura 2.10: Seqüência (x_n) no Método da Iteração Linear

segue por hipótese que $|\phi'(\omega_n)| \leq \lambda$, então

$$|\phi(x_n) - \phi(x_{n-1})| \leq \lambda |x_n - x_{n-1}|$$

implicando em

$$|x_{n+1} - x_n| \leq \lambda |x_n - x_{n-1}|$$

por analogia teremos

$$|x_{n+1} - x_n| \leq \lambda |x_n - x_{n-1}| \leq \lambda^2 |x_{n-1} - x_{n-2}| \leq \dots \leq \lambda^n |x_1 - x_0|$$

ou seja,

$$|x_{n+1} - x_n| \leq \lambda^n |x_1 - x_0|$$

Passando o limite nesta última desigualdade teremos:

$$\begin{aligned} \lim_{n \rightarrow \infty} |x_{n+1} - x_n| &\leq \lim_{n \rightarrow \infty} \left(\underbrace{\lambda^n |x_1 - x_0|}_{\text{constante}} \right) \\ &\leq |x_1 - x_0| \underbrace{\lim_{n \rightarrow \infty} \lambda^n}_0 \\ &\leq 0 \end{aligned}$$

Logo (x_n) é uma seqüência de *Cauchy*, pelo Teorema 2.4 a seqüência (x_n) converge para um certo L em (a, b) , ou seja, $\lim_{n \rightarrow \infty} x_n = L$. Como $\phi(x)$ é diferenciável, logo contínua, temos que:

$$\phi(L) = \phi\left(\lim_{n \rightarrow \infty} x_n\right) = \lim_{n \rightarrow \infty} \phi(x_n) = \lim_{n \rightarrow \infty} x_{n+1} = L$$

Portanto L é o ponto fixo de $\phi(x)$. ■

Corolário 2.1. *O ponto fixo L dado no Teorema 2.5 é o único ponto fixo de $\phi(x)$ em $[a, b]$.*

demonstração

A prova será feita por redução ao absurdo. Para isso, suponha que exista um outro ponto fixo M em $[a, b]$. Pelo teorema do valor médio existe $c \in (L, M)$ tal que:

$$\begin{aligned} |\phi(L) - \phi(M)| &= |\phi'(c)| |L - M| \\ &\leq \lambda |L - M| \end{aligned}$$

mais ainda,

$$\begin{aligned} |L - M| &\leq \lambda |L - M| \\ (1 - \lambda) |L - M| &\leq 0 \end{aligned}$$

Como $\lambda < 1$ então $(1 - \lambda) > 0$, ou seja,

$$0 \leq (1 - \lambda) |L - M| \leq 0$$

$$(1 - \lambda) |L - M| = 0$$

Portanto $L = M$. ■

Mais detalhes sobre ponto fixo veja [7].

2.6.1 A Função de Iteração

Pelo Teorema 2.5 observamos que a escolha de $\phi(x)$ é muito importante para a convergência do Método da Iteração Linear. Por isso, para aplicar esse método devemos procurar uma função de iteração $\phi(x)$ que satisfaça as hipóteses desse teorema. O exemplo abaixo mostra como podemos obter varias funções de iteração. Bastando para isso, escolher e isolar x na equação.

Exemplo 2.6.1. Encontre o zero de $f(x) = 2x - \cos(x)$ em $[\frac{1}{5}, 2]$, com $\delta = 0.01$

1ª função de iteração

$$\begin{aligned} 2x - \cos(x) &= 0 \\ x + x - \cos(x) &= 0 \\ \cos(x) - x &= x \end{aligned}$$

assim $\phi_1(x) = \cos(x) - x$

2ª função de iteração

$$2x - \cos(x) = 0 \implies x = \frac{\cos(x)}{2}$$

$$\text{assim } \phi_2(x) = \frac{\cos(x)}{2}$$

3ª função de iteração

$$\begin{aligned} 2x - \cos(x) &= 0 && \text{somando } x \text{ em ambos os lados} \\ 2x + x - \cos(x) &= x \\ 3x - \cos(x) &= x \end{aligned}$$

$$\text{assim } \phi_3(x) = 3x - \cos(x)$$

Vamos agora calcular derivada de cada função obtida, observando seu comportamento no intervalo $[\frac{1}{5} \ 2]$.

$$|\phi_1'(x)| = |- \operatorname{sen}(x) - 1| > 1, \text{ para algum } x \in [\frac{1}{5} \ 2]$$

$$|\phi_2'(x)| = |-\frac{\operatorname{sen}(x)}{2}| < 1, \text{ para todo } x \in [\frac{1}{5} \ 2]$$

$$|\phi_3'(x)| = |3 + \operatorname{sen}(x)| > 1, \text{ para todo } x \in [\frac{1}{5} \ 2]$$

Portanto concluímos que $\phi_2(x)$ deve ser a função de iteração. Uma vez que $\phi_2(x)$ satisfaz o Teorema 2.5. Assim começamos nossa seqüência (x_n) com o elemento $x_0 = \frac{1}{5}$ e iteramos (a calculadora deve estar em radianos):

$$\begin{aligned} x_1 &= \phi_2(x_0) = \phi_2\left(\frac{1}{5}\right) = 0.49003 \\ x_2 &= \phi_2(x_1) = \phi_2(0.49003) = 0.44116 \\ x_3 &= \phi_2(x_2) = \phi_2(0.44116) = 0.45213 \\ x_4 &= \phi_2(x_3) = \phi_2(0.45213) = 0.44976 \end{aligned}$$

Paramos o método em x_4 , porque $|x_4 - x_3| = 0.00237 \leq \delta$.

A escolha de x_0

Caso a função $f(x)$ tenha a forma da figura 2.9, x_0 deve ser escolhido de tal forma que x_0 seja menor que o zero ϵ , pois caso contrário o método pode não convergir. Uma sugestão seria começar com o extremo do intervalo (nesse caso o ponto a).

Caso $f(x)$ tenha forma da figura 2.10 o elemento x_0 pode ser arbitrário dentro do intervalo.

2.7 Comentários Finais Sobre os Métodos

2.7.1 Localização de Zeros

É um bom método para localizar os possíveis intervalos onde se encontram os zeros de uma função. Porém, se o intervalo de pesquisa for muito grande ou se a função possuir muitos zeros, o método pode se tornar computacionalmente inviável.

Se o γ escolhido não for suficientemente pequeno podemos ter intervalos onde existem dois ou mais zeros. Se γ for pequeno o método pode não ser viável.

2.7.2 Método do Meio Intervalo - MMI

A grande vantagem deste método consiste no fato que só exigimos que a função $f(x)$ seja contínua. Mas infelizmente a convergência é lenta.

2.7.3 Método da Secante

Exige que o sinal de $f'(x)$ e $f''(x)$ sejam constantes no intervalo. Nem sempre a função tem derivadas, o que inviabiliza o método. Se o intervalo de procura for muito grande o método pode se tornar lento.

2.7.4 Método de Newton

Exige que o sinal de $f'(x)$ e $f''(x)$ sejam constantes no intervalo. Mas tem convergência extraordinária.

2.7.5 Método da Iteração Linear

A função de iteração $\phi(x)$ deve ser obtida através de um processo manual, ou seja, analítico. Satisfazendo a hipótese do Teorema 2.5. Nem sempre é fácil encontrar tal função.

Para finalizar, relacionamos o número de iterações que cada método gasta para resolver um problema. Também verificamos os zeros de uma função onde não temos o intervalo de procura. Isto fica claro nos exemplos abaixo.

Exemplo 2.7.1. Encontre um zero de $f(x) = e^{-\frac{1}{10}x} + x^2 - 10$ em $[\frac{5}{2}, \frac{7}{2}]$ com $\delta = 10^{-5}$

Neste exemplo obtemos os resultados:

	MMI	M. Secante	M. Newton	M. Iteração Linear
número de iterações	16	6	3	4

Exemplo 2.7.2. Calcule pelo menos um zero de $f(x) = \log(x) + x$.

Observe que não foi dado o intervalo. Uma saída é aplicar o Teorema 2.3 com $h(x) = -x$ e $g(x) = \log(x)$. O gráfico dessas funções é dado na figura 2.11. Nele podemos observar que a interseção de $h(x)$ e $g(x)$ encontra-se no intervalo $[0, 1]$. Através do gráfico também observamos que essa interseção é única. Logo $f(x)$ possui somente um único zero. Para esse exemplo aplicaremos o método do meio intervalo - MMI, com $\delta = 0.009$. Lembrando que em vez do intervalo $[0, 1]$ trabalharemos com o intervalo $[\frac{1}{100}, 1]$ já que $f(x)$ não está definida em 0.

n	a_n	b_n	x_n	$ x_n - x_{n-1} $	
0	1/100	1	0.505		
1	1/100	0.505	0.257	0.248 > δ	
2	0.257	0.505	0.381	0.124 > δ	
3	0.381	0.505	0.443	0.062 > δ	
4	0.381	0.443	0.412	0.031 > δ	
5	0.381	0.412	0.396	0.016 > δ	
6	0.396	0.412	0.404	0.008 ≤ δ	← parar o método

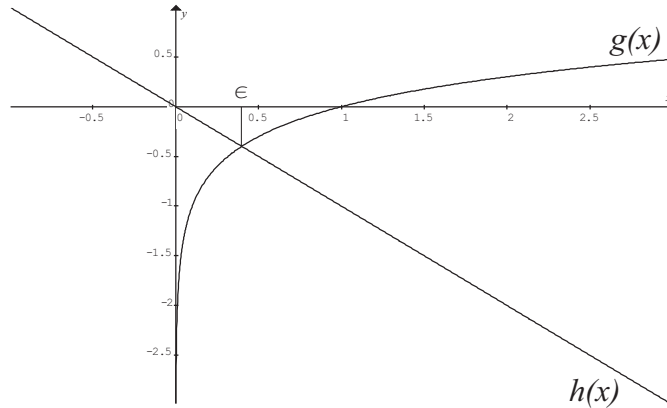


Figura 2.11: Método gráfico

Logo nossa aproximação com $\delta = 0.09$ é $x_6 = 0.404$.

2.8 Zeros de um Polinômio

Nas seções anteriores desenvolvemos métodos para encontrar zeros de uma função. Inclusive de polinômios. Nesta seção vamos aprofundar um pouco mais nosso conhecimento sobre os zeros de um polinômio. Lembramos que neste texto, o zero de um polinômio é o mesmo que a raiz de um polinômio. Nosso principal objetivo é chegar no teorema que limita os zeros de um polinômio qualquer.

Da Álgebra Linear, sabemos que dois espaços vetoriais são isomorfos se, existe um isomorfismo entre eles. Nesta seção vamos utilizar o isomorfismo dos números complexos (\mathbb{C}) em \mathbb{R}^2 , denotado por $\mathbb{C} \approx \mathbb{R}^2$, que associa a cada número complexo $a + bi$ o par (a, b) em \mathbb{R}^2 .

Também vamos utilizar a norma euclidiana em \mathbb{R}^2 dada por:

$$\|(a, b)\| = \sqrt{a^2 + b^2},$$

assim $\|a + bi\| = \|(a, b)\|$.

Nas proposições e teoremas abaixo, os polinômios podem ter coeficientes reais ou complexos. Consideramos apenas polinômios mônicos, ou seja, o coeficiente do termo de maior grau é igual a 1.

Proposição 2.1. *Seja $p(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ um polinômio de grau n e $a_0 \neq 0$. Se $\sum_{i=0}^{n-1} \frac{\|a_i\|}{\|x\|^{n-i}} < 1$ e $x \neq 0$, então $p(x) \neq 0$.*

demonstração

Como $x \neq 0$ podemos escrever

$$p(x) = x^n \left(1 + \sum_{i=0}^{n-1} \frac{a_i}{x^{n-i}} \right) \quad (2.3)$$

e usar a desigualdade triangular⁵.

$$\left\| \sum_{i=0}^{n-1} \frac{a_i}{x^{n-i}} \right\| \leq \sum_{i=0}^{n-1} \frac{\|a_i\|}{\|x\|^{n-i}} < 1 \quad (2.4)$$

De (2.3) e (2.4) concluímos que $p(x) \neq 0$. ■

Corolário 2.2. *Se $\frac{\|a_i\|^{1/(n-i)}}{\|x\|} < \frac{1}{2}$ para $i = 0, \dots, n-1$ e $x \neq 0$. Então $p(x) \neq 0$.*

demonstração

Veja que

$$\begin{aligned} \sum_{i=0}^{n-1} \frac{\|a_i\|}{\|x\|^{n-i}} &= \sum_{i=0}^{n-1} \left(\frac{\|a_i\|^{1/(n-i)}}{\|x\|} \right)^{n-i} \\ &\leq \frac{1}{2^n} + \frac{1}{2^{n-1}} + \dots + \frac{1}{2^2} + \frac{1}{2} \\ &\leq 1 - \frac{1}{2^n} \\ &< 1 \end{aligned}$$

⁵ $\|a + b\| \leq \|a\| + \|b\|$

De acordo com a Proposição 2.1 $p(x) \neq 0$.

■

Estamos prontos para enunciar e demonstrar o

Teorema 2.6. *Seja $p(x) = x^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0$ um polinômio de grau n e $a_0 \neq 0$. Se x é um zero de p , então*

$$\|x\| \leq L$$

$$\text{onde } L = 2 \max_{0 \leq i \leq n-1} \{\|a_i\|^{1/(n-i)}\}$$

demonstração

Negando o Corolário 2.2 temos que; se $p(x) = 0$ então existe i_0 tal

que $\frac{\|a_{i_0}\|^{1/(n-i_0)}}{\|x\|} \geq \frac{1}{2}$. Do fato de

$$\frac{\max_{0 \leq i \leq n-1} \{\|a_i\|^{1/(n-i)}\}}{\|x\|} \geq \frac{\|a_{i_0}\|^{1/(n-i_0)}}{\|x\|}$$

concluimos que

$$\frac{\max_{0 \leq i \leq n-1} \{\|a_i\|^{1/(n-i)}\}}{\|x\|} \geq \frac{1}{2} \implies \|x\| \leq 2 \max_{0 \leq i \leq n-1} \{\|a_i\|^{1/(n-i)}\}$$

■

Exemplo 2.8.1. *Ache o limite superior e inferior dos zeros reais do polinômio*

$$p(x) = x^3 + 3x^2 - 10x + 24.$$

Veja que

$$\begin{aligned} L &= 2 \max_{0 \leq i \leq n-1} \{\|a_i\|^{1/(n-i)}\} \\ &= 2 \max\{\|a_0\|^{1/3}, \|a_1\|^{1/2}, \|a_2\|\} \\ &= 2 \max\{24^{1/3}, 10^{1/2}, 3\} \\ &= 2\sqrt{10} \end{aligned}$$

assim todos os zeros reais de $p(x)$ estão no intervalo $[-L, L]$.

Exemplo 2.8.2. Determine a bola onde todos os zeros reais e complexos de $p(x) = x^3 + 3x^2 - 10x + 24$ estão.

Pelo exemplo anterior temos que $L = 2\sqrt{10}$. Assim se x é um zero de $p(x)$ então $\|x\| \leq L$. Usando a correspondência $\mathbb{C} \approx \mathbb{R}^2$ temos a bola da figura 2.12:

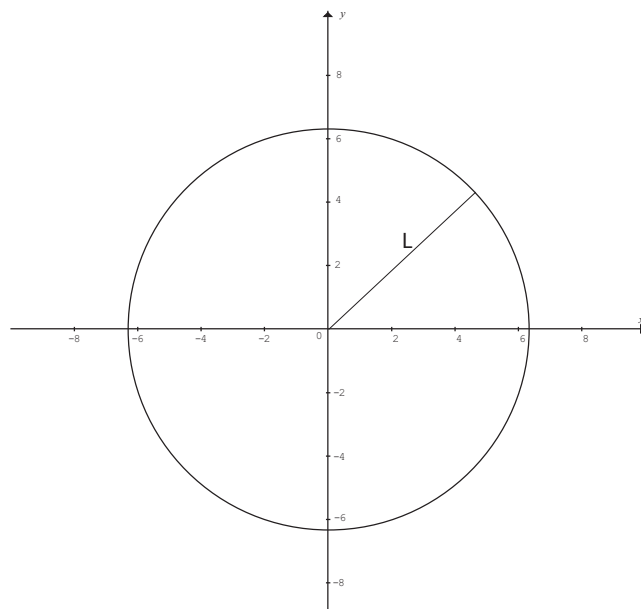


Figura 2.12: Limite dos zeros

Exemplo 2.8.3. Ache o limite superior e inferior dos zeros do polinômio

$$p(x) = 2x^3 - 3x^2 - 2x + 3.$$

Observe que neste caso $p(x)$ não está na forma do Teorema 2.6. Para resolver isso, basta tomar o polinômio

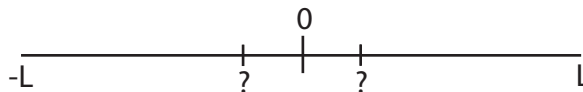


Figura 2.13: Isolando zeros

$$g(x) = \frac{p(x)}{a_3} = \frac{2x^3 - 3x^2 - 2x + 3}{2} = x^3 - \frac{3}{2}x^2 - x + \frac{3}{2}$$

cujo zeros são os mesmos de $p(x)$.

Aplicando o Teorema 2.6 em $g(x)$ temos:

$$L = 2 \max\{(3/2)^{\frac{1}{3}}, 1, 3/2\} = 3$$

assim todos os zeros reais de $p(x)$ estão no intervalo $[-L \quad L] = [-3 \quad 3]$.

Ainda podemos avançar um pouco mais. Queremos agora o limite inferior dos zeros positivos e o limite superior dos zeros negativos, veja (?) na figura 2.13.

Para encontrar estes limites, precisamos de um resultado importante de Álgebra:⁶

Se $p(x) = x^n + a_{n-1}x^{n-1} + \dots + a_1x + a_0$ é um polinômio de grau n , então $p(x)$ tem no máximo n zeros reais ou complexos.

Decorre desse resultado que todo polinômio de grau ímpar tem pelo menos um zero real.

Com esses resultados, seja $(\xi_0, \xi_1, \dots, \xi_{n-1})$ os zeros de $p(x)$, então podemos escrever $p(x)$ na forma (também é dado pela Álgebra):

$$p(x) = (x - \xi_0)(x - \xi_1) \cdots (x - \xi_{n-1}).$$

Consideremos o polinômio

$$P_1(x) = x^n p\left(\frac{1}{x}\right).$$

⁶Veja em [5]

Pelo que acabamos de ver,

$$\begin{aligned}
 P_1(x) &= x^n \left(\frac{1}{x} - \xi_0 \right) \left(\frac{1}{x} - \xi_1 \right) \cdots \left(\frac{1}{x} - \xi_{n-1} \right) \\
 &= x^n \left(\frac{1 - x\xi_0}{x} \right) \left(\frac{1 - x\xi_1}{x} \right) \cdots \left(\frac{1 - x\xi_{n-1}}{x} \right) \\
 &= (1 - x\xi_0)(1 - x\xi_1) \cdots (1 - x\xi_{n-1}).
 \end{aligned}$$

Observe que os zeros de $P_1(x)$ são $\left(\frac{1}{\xi_0}, \frac{1}{\xi_1}, \dots, \frac{1}{\xi_{n-1}} \right)$. Aplicando o Teorema 2.6 em $P_1(x)$ temos a existência de L_1 tal que

$$\left\| \frac{1}{xi_i} \right\| \leq L_1$$

o que implica em $\frac{1}{L_1} \leq \|\xi_i\|$. Portanto, se x é um zero de um polinômio qualquer (mônico), então $L_1 \leq \|x\| \leq L$.

No caso de zeros reais podemos dizer que:

$$-L_1 \leq \frac{1}{\xi_i} \leq L_1 \text{ para } i = 0, \dots, n-1$$

onde

$$\frac{1}{\xi_i} \leq L_1 \implies \frac{1}{L_1} \leq \xi_i$$

e

$$-L_1 \leq \frac{1}{\xi_i} \implies \xi_i \leq -\frac{1}{L_1}$$

assim, os zeros reais negativos de $p(x)$ estão no intervalo $\left[-L \quad -\frac{1}{L_1} \right]$

e os zeros positivos estão em $\left[\frac{1}{L_1} \quad L \right]$.

Exemplo 2.8.4. Ache os limites L, L_1 de $p(x) = 2x^3 - 3x^2 - 2x + 3$. No Exemplo 2.8.3 calculamos $L = 3$. Conforme feito, vamos utilizar o polinômio $g(x) = x^3 - \frac{3}{2}x^2 - x + \frac{3}{2}$. Calculando $P_1(x)$ temos:

$$P_1(x) = x^3 p\left(\frac{1}{x}\right) = x^3 \left(\frac{1}{x^3} - \frac{3}{2x^2} - \frac{1}{x} + \frac{3}{2} \right) = 1 - \frac{3}{2}x - x^2 + \frac{3}{2}x^3.$$

Como $P_1(x)$ não está nas condições do Teorema 2.6, mudamos para

$$g_1(x) = \frac{P_1(x)}{3/2} = x^3 - \frac{2}{3}x^2 - x + \frac{2}{3}$$

Aplicando o Teorema 2.6 em $g_1(x)$ temos:

$$L_1 = 2 \max\{(2/3)^{\frac{1}{3}}, 1, (2/3)\} = 2(2/3)^{\frac{1}{3}} \cong 1.747 \quad e \quad \frac{1}{L_1} = 0.572 .$$

Assim todos os zeros complexos de $p(x)$ estão no disco da figura 2.14.

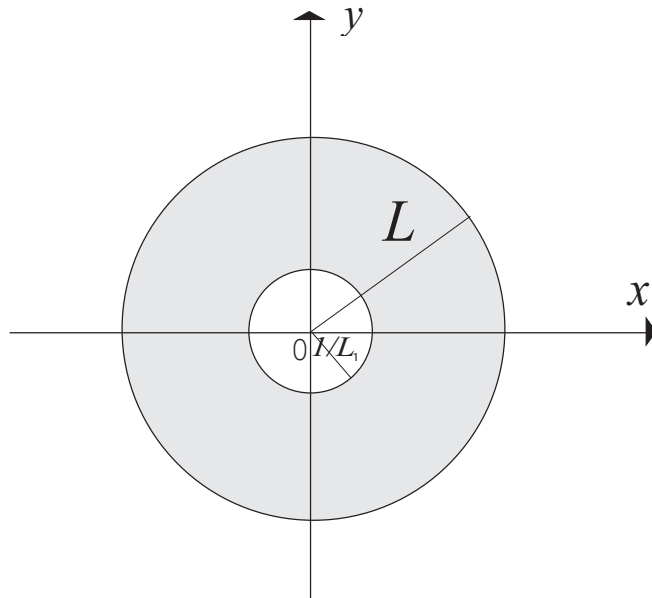


Figura 2.14: Limite de zeros complexos

Portanto todos os zeros reais (se existirem) negativos de $p(x)$ estão em $[-3 \quad -0.572]$ e todos os zeros positivos em $[0.572 \quad 3]$.

Uma vez dados os intervalos $[-L \quad -\frac{1}{L_1}]$ e $[\frac{1}{L_1} \quad L]$ podemos aplicar o método de localização de zeros associado com qualquer método de busca (MMI, Newton, etc) para encontrar os zeros reais do polinômio.

2.8.1 Multiplicidade de um zero

Consideremos o seguinte teorema cuja demonstração pode ser encontrada em [5].

Teorema 2.7. *Todo polinômio de grau n tem exatamente n zeros reais ou complexos.*

Considere o polinômio $p(x) = x^3 + 4x + 5x + 2$. Aplicando o 2.7 temos a existência de 3 zeros. Mas os únicos zeros são $\{-1, -2\}$. Isto parece contraditório. O fato se dá pela multiplicidade do zero -1 , ou seja, -1 é contado duas vezes. Dizemos que o zero -1 tem multiplicidade igual a 2.

Para saber a multiplicidade de um zero basta olhar para a derivada do polinômio. Assim se ϵ é um zero de $p(x)$ então a multiplicidade de ϵ é dado por m , onde

$$\begin{aligned} p'(x) &= 0 \\ p''(x) &= 0 \end{aligned}$$

$$\vdots$$

$$\begin{aligned} p^{m-1}(x) &= 0 \\ p^m(x) &\neq 0 \end{aligned}$$

Exemplo 2.8.5. *No caso de $p(x) = x^3 + 4x + 5x + 2$ temos que:*

$$\begin{aligned} p(-1) &= 0 & p(-2) &= 0 \\ p'(-1) &= 0 & p'(-2) &\neq 0 \\ p''(-1) &\neq 0 \end{aligned}$$

Logo a multiplicidade de -1 é 2 e a multiplicidade de -2 é 1.

Observe que a soma da multiplicidade de todos os zeros é igual ao grau do polinômio. Outro fato, se o polinômio $p(x)$ tem grau ímpar então possui pelo menos um zero real.

Concluimos que para procurar por zeros reais de um polinômio devemos seguir um roteiro:

i Encontrar os limites L, L_1

ii- Aplicar método de localização de zeros.

iii- Contar os zeros.

iv- Aplicar algum método de aproximação(MMI, Newton,Secante,etc)

2.9 Exercícios

2.9.1. *Seja $f(x) = x^4 - x - 10$ definida em $[-2 \ 2]$. Aplique o método de localização de zeros com $\gamma = \frac{1}{4}$ para encontrar os subintervalos onde $f(x)$ possui zeros.*

2.9.2. *Aplique o MMI para encontrar os zeros da função do Exercício 2.9.1.*

2.9.3. *Encontre através do MMI com $\delta = 0.01$ pelo menos um zero de:*

a) $f(x) = x^3 - x + 1$

b) $g(x) = 2e^{-x} - \text{sen}(x)$

c) $h(x) = x \ln(x) - 0.8$

2.9.4. *Aplique o Método da Secante para encontrar uma aproximação para $\sqrt{2}$ e compare com o Método de Newton.*

2.9.5. *Mostre a equação geral para a sequência (x_n) no Método da Secante quando:*

a) $f'(x) < 0$ e $f''(x) > 0$ para todo $x \in [a \ b]$

b) $f'(x) < 0$ e $f''(x) < 0$ para todo $x \in [a \ b]$

c) $f'(x) > 0$ e $f''(x) < 0$ para todo $x \in [a, b]$

Conclua que para qualquer caso, o método da secante é dado por:

$$x_n = c - \frac{f(c)}{f(c) - f(x_{n-1})}(c - x_{n-1}),$$

onde

$$c = \begin{cases} a & \text{se } f(a)f''(a) > 0 \\ b & \text{se } f(b)f''(b) > 0 \end{cases} \quad \text{e} \quad x_0 = \begin{cases} a & \text{se } c \neq a \\ b & \text{se } c \neq b \end{cases}$$

2.9.6. Ache um zero de $f(x) = e^x - \sin(x) - 2$ com $\delta = 0.001$ pelo Método da Secante e pelo Método de Newton. Compare os resultados e diga qual Método converge mais rápido.

2.9.7. Ache um zero de $f(x) = x + \ln(x)$ e $g(x) = 2x^3 + \ln(x) - 5$ com $\delta = 0.001$ pelo Método da Iteração Linear.

2.9.8. Usando o Método da Iteração Linear ache um zero de $f(x) = \cos(x) + \ln(x) + x$ com $\delta = 10^{-2}$.

2.9.9. Calcule uma aproximação para $\sqrt[3]{9}$

2.9.10. Seja $f(x) = e^x + x^3 - 1$, ache x tal que $f(x) = 2$.

2.9.11. Ache os pontos de máximo e mínimo de $f(x) = x^5 + 10x^2 + x$ em $[-2, 1]$.

2.9.12. Determine o mínimo global de $f(x) = 2x^4 - 2x^3 - x^2 - x - 3$

2.9.13. O que ocorre quando aplicamos o MMI em uma função que possui 3 zeros em um intervalo $[a, b]$.

2.9.14. Para cada item abaixo dê um exemplo onde não podemos aplicar o:

a) Método do Meio Intervalo - MMI

b) Método da Secante

c) Método de Newton

2.9.15. De acordo com o texto calcule L, L_1 para $p(x) = x^4 - 5x^3 - 7x^2 + 29x + 30$

2.9.16. Encontre os três zeros reais de $p(x) = x^3 - 2x^2 - 5x + 6$ com $\delta = 10^{-2}$.

Capítulo 3

Interpolação

3.1 Introdução

Abordaremos neste capítulo os aspectos básicos da teoria de interpolação.

Começamos apresentando dois problemas:

i- Considere uma função $f(x)$ conhecida apenas nos pontos (x_0, x_1, \dots, x_n) . Se não temos a forma analítica de $f(x)$ como podemos determinar o valor $f(c)$ para um $c \in (x_i, x_j)$?

ii- Seja $f(x)$ uma função de forma analítica complicada ou de difícil avaliação. Existe uma outra função $g(x)$ tal que $g(x) \cong f(x)$, onde $g(x)$ é uma função de fácil avaliação?

Esses problemas serão resolvidos através da construção de um polinômio chamado polinômio interpolador. Em outros textos é possível encontrar interpolantes trigonométricas e exponenciais, para isso veja [11].

Tudo começa com o teorema abaixo:

Teorema 3.1. (Weirstrass) Se $f(x)$ é uma função contínua em um intervalo fechado $[a, b]$, então existe um polinômio $p(x)$ tal que $p(x) \cong f(x)$ para todo $x \in [a, b]$, ou seja, $|f(x) - p(x)| < \varepsilon$ para qualquer ε dado.

Como muitos teoremas na matemática, o teorema acima só garante a existência do polinômio interpolador, outro fato é que o grau do polinômio $p(x)$ depende do ε escolhido. A demonstração encontra-se em [9].

3.2 Interpolação de Lagrange

Garantida a existência do polinômio interpolador, vamos agora desenvolver um método para encontrá-lo.

Teorema 3.2. (*Lagrange*) *Sejam (x_0, x_1, \dots, x_n) os pontos distintos onde $f(x)$ é conhecida. Então o polinômio interpolador $p(x)$ tem grau n e é dado pela fórmula:*

$$p(x) = \sum_{i=0}^n \left[f(x_i) \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)} \right]$$

Antes de demonstrar o Teorema 3.2 façamos um exemplo:

Exemplo 3.2.1. *Seja $f(x)$ conhecida em :*

$$\begin{aligned} (x_0, f(x_0)) &= (-1, 1) \\ (x_1, f(x_1)) &= (1, 3) \\ (x_2, f(x_2)) &= (2, -1) \\ (x_3, f(x_3)) &= (3, -4) \end{aligned}$$

Desejamos saber o valor de $f\left(\frac{1}{2}\right)$. Usando o Teorema 3.2 temos:

$$\begin{aligned}
p(x) &= \sum_{i=0}^3 \left[f(x_i) \prod_{j=0, j \neq i}^3 \frac{(x - x_j)}{(x_i - x_j)} \right] \\
&= f(x_0) \frac{(x - x_1)(x - x_2)(x - x_3)}{(x_0 - x_1)(x_0 - x_2)(x_0 - x_3)} + f(x_1) \frac{(x - x_0)(x - x_2)(x - x_3)}{(x_1 - x_0)(x_1 - x_2)(x_1 - x_3)} \\
&\quad + f(x_2) \frac{(x - x_0)(x - x_1)(x - x_3)}{(x_2 - x_0)(x_2 - x_1)(x_2 - x_3)} + f(x_3) \frac{(x - x_0)(x - x_1)(x - x_2)}{(x_3 - x_0)(x_3 - x_1)(x_3 - x_2)} \\
&\quad \vdots \\
&= \frac{13}{24}x^3 - \frac{11}{4}x^2 + \frac{11}{24}x + \frac{27}{4}
\end{aligned}$$

Observe que $p(x_i) = f(x_i)$ para $i = 0, 1, 2, 3$ e o grau de $p(x)$ é 3.

Agora, para saber o valor de $f\left(\frac{1}{2}\right)$ basta calcular $p\left(\frac{1}{2}\right) \cong 5.671$

demonstração do Teorema 3.2

Consideremos os polinômios de Lagrange:

$$\begin{aligned}
P_0(x) &= (x - x_1)(x - x_2) \cdots (x - x_n) \\
P_1(x) &= (x - x_0)(x - x_2) \cdots (x - x_n) \\
&\vdots \\
P_n(x) &= (x - x_0)(x - x_1) \cdots (x - x_{n-1})
\end{aligned}$$

Em geral $P_i(x) = \prod_{j=0, j \neq i}^n (x - x_j)$, para $i = 0, \dots, n$.

Esses polinômios tem a seguinte propriedade:

$$P_i(x) \neq 0 \text{ e } P_i(x_j) = 0 \quad (3.1)$$

Pelo Teorema 3.1 temos que $p(x)$ existe e seu grau é igual a n .

Dessa forma podemos escrever $p(x)$ como combinação linear dos polinômios de Lagrange, ou seja, existem escalares (b_0, b_1, \dots, b_n) tais que:

$$p(x) = b_0 P_0(x) + b_1 P_1(x) + \cdots + b_n P_n(x) \quad (3.2)$$

Pela equação (3.1) podemos afirmar que:

$$\begin{aligned} p(x_k) &= \underbrace{b_0 P_0(x_k)}_0 + \underbrace{b_1 P_1(x_k)}_0 + \cdots + b_k P_k(x_k) + \cdots + \underbrace{b_n P_n(x_k)}_0 \\ &= b_k P_k(x_k) \end{aligned}$$

o que implica em

$$b_k = \frac{p(x_k)}{P_k(x_k)} \text{ para } k = 0, 1, \dots, n \quad (3.3)$$

Substituindo a equação (3.3) na equação (3.2) temos:

$$p(x) = \frac{p(x_0)}{P_0(x_0)} P_0(x) + \frac{p(x_1)}{P_1(x_1)} P_1(x) + \cdots + \frac{p(x_n)}{P_n(x_n)} P_n(x)$$

como $p(x_i) = f(x_i)$ então

$$p(x) = \frac{f(x_0)}{P_0(x_0)} P_0(x) + \frac{f(x_1)}{P_1(x_1)} P_1(x) + \cdots + \frac{f(x_n)}{P_n(x_n)} P_n(x)$$

logo

$$p(x) = \sum_{i=0}^n f(x_i) \frac{P_i(x)}{P_i(x_i)} = \sum_{i=0}^n f(x_i) \prod_{j=0, j \neq i}^n \frac{(x - x_j)}{(x_i - x_j)}$$

■

Corolário 3.1. *O polinômio dado no Teorema 3.2 é único.*

demonstração

3.3. INTERPOLAÇÃO COM DIFERENÇAS DIVIDIDAS FINITAS - DDF63

Suponhamos que exista um outro polinômio $s(x)$ tal que $s(x_i) = f(x_i)$ para $i = 0, 1, \dots, n$. Considere o polinômio:

$$T(x) = s(x) - p(x)$$

Veja que $T(x_i) = s(x_i) - p(x_i) = 0$ para $i = 0, 1, \dots, n$. Como o grau de $s(x)$ e $p(x)$ é n , então o grau de $T(x)$ também é n . Mas $T(x)$ tem $n + 1$ zeros o que é um absurdo de acordo com o Teorema 2.7. Logo $s(x) = p(x)$. ■

3.3 Interpolação com Diferenças Divididas Finitas - DDF

Consideremos uma função $f(x)$ contínua em $[a, b]$ e diferenciável em (a, b) . Uma diferença dividida finita - DDF de primeira ordem de $f(x)$ em relação a x_0, x_1 é dada por:

$$f[x_1, x_0] = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

observe que $f[x_1, x_0]$ é uma aproximação para $f'(x_0)$.

A DDF de segunda ordem será dada por:

$$f[x_2, x_1, x_0] = \frac{f[x_2, x_1] - f[x_1, x_0]}{x_2 - x_0}$$

veja que isso é uma aproximação para $f''(x_1)$.

Assim a DDF de n -ésima ordem será dada por:

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0}$$

3.3.1 Propriedades de uma DDF

i- $f[x_n, x_{n-1}, \dots, x_1, x_0] = f[x_{\alpha_0}, x_{\alpha_1}, \dots, x_{\alpha_{n-1}}, x_{\alpha_n}]$ onde $\alpha_0, \alpha_1, \dots, \alpha_{n-1}, \alpha_n$ é qualquer permutação dos inteiros $\{n, n-1, \dots, 1, 0\}$. Por exemplo:

$$f[x_2, x_1, x_0] = f[x_1, x_2, x_0] = f[x_2, x_0, x_1] = f[x_0, x_1, x_2] = f[x_1, x_0, x_2] = f[x_0, x_2, x_1]$$

$$f[x_0, x_1] = \frac{f(x_1)}{x_1 - x_0} + \frac{f(x_0)}{x_0 - x_1}$$

3.3.2 Obtenção da Fórmula

Pelo Teorema 3.1 temos a existência do polinômio $p(x)$. Assim considere (x_0, x_1, \dots, x_n) os $n + 1$ pontos conhecidos de $f(x)$. Pela definição de DDF temos:

$$p[x, x_0] = \frac{p(x) - p(x_0)}{x - x_0}$$

ou

$$p(x) = p(x_0) + (x - x_0)p[x, x_0] \quad (3.4)$$

mais ainda,

$$\begin{aligned} p[x, x_0, x_1] &= \frac{p[x, x_0] - p[x_0, x_1]}{x - x_1} \\ p[x, x_0] &= p[x_0, x_1] + (x - x_1)p[x, x_0, x_1] \end{aligned} \quad (3.5)$$

Substituindo (3.5) em (3.4) temos:

$$p(x) = p(x_0) + (x - x_0)p[x_0, x_1] + (x - x_0)(x - x_1)p[x, x_0, x_1]$$

como

$$p[x, x_0, x_1] = (x - x_2)p[x, x_0, x_1, x_2] + p[x_0, x_1, x_2]$$

então

$$\begin{aligned} p(x) &= p(x_0) + (x - x_0)p[x_0, x_1] + (x - x_0)(x - x_1)p[x_0, x_1, x_2] \\ &\quad + (x - x_0)(x - x_1)(x - x_2)p[x, x_0, x_1, x_2]. \end{aligned}$$

Continuando a substituir $p[x, x_0, x_1, x_2]$ teremos:

$$\begin{aligned} p(x) &= p(x_0) + (x - x_0)p[x_0, x_1] + (x - x_0)(x - x_1)p[x_0, x_1, x_2] \\ &\quad + (x - x_0)(x - x_1)(x - x_2)p[x_0, x_1, x_2, x_3] + \dots + \\ &\quad (x - x_0) \dots (x - x_{n-1})p[x_0, \dots, x_n] + (x - x_0) \dots (x - \\ &\quad x_n)p[x, x_0, \dots, x_n] \end{aligned}$$

Como $p(x)$ é um polinômio de grau n então a $(n+1)$ -ésima derivada é igual a zero. Logo $p[x, x_0, \dots, x_n] = 0$. Dessa forma o polinômio $p(x)$ pode ser escrito como:

$$p(x) = f(x_0) + (x - x_0)p[x_0, x_1] + (x - x_0)(x - x_1)p[x_0, x_1, x_2] + \\ + (x - x_0)(x - x_1)(x - x_2)p[x_0, x_1, x_2, x_3] + \dots + \\ + (x - x_0) \cdots (x - x_{n-1})p[x_0, \dots, x_n]$$

Exemplo 3.3.1. *Vamos construir o polinômio interpolador para função $f(x) = \frac{\sin(x)}{\sqrt{x}}$ via DDF. Para essa interpolação considere os pontos:*

$$(x_0, f(x_0)) = \left(\frac{3\pi}{2}, -0.46\right)$$

$$(x_1, f(x_1)) = \left(\frac{\pi}{2}, 0.797\right)$$

Veja que

$$p[x_0, x_1] = \frac{p(x_0) - p(x_1)}{x_0 - x_1} = \frac{f(x_0) - f(x_1)}{x_0 - x_1} = \frac{-1.257}{\pi}$$

Assim

$$p(x) = f(x_0) + (x - x_0)p[x_0, x_1] \\ = -0.46 + \left(x - \frac{3\pi}{2}\right) \left(\frac{-1.257}{\pi}\right) \\ = -\frac{1.257}{\pi}x + 1,425$$

3.4 Erro na Interpolação

Considere o problema (ii) na Introdução do Capítulo. Nesse caso conhecemos a forma analítica de $f(x)$. Veremos que se $f(x)$ for suficientemente diferenciável, então podemos calcular o erro na interpolação.

Seja $p(x)$ o polinômio interpolador de grau n criado com base em (x_0, x_1, \dots, x_n) . Definimos o erro de como:

$$E_t(x) = f(x) - p(x)$$

A proposição abaixo caracteriza $E_t(x)$. Mas sua demonstração requer um famoso teorema, chamado de Teorema de Rolle. Apenas

enunciaremos esse teorema. Sua demonstração pode ser encontrada em livros de Cálculo ou em [3].

Teorema 3.3. (*Rolle*) Seja $f : [a, b] \rightarrow \mathbb{R}$ uma função contínua definida em um intervalo fechado $[a, b]$. Se $f(x)$ é diferenciável no intervalo aberto (a, b) e $f(a) = f(b) = 0$. Então existe $\xi \in (a, b)$ tal que $f'(\xi) = 0$.

Proposição 3.1. Seja $f(x) = E_t(x) + p(x)$, onde $p(x)$ é o polinômio interpolador de $f(x)$ relativamente aos pontos (x_0, x_1, \dots, x_n) de $[a, b]$ e $f(x)$ seja $(n+1)$ vezes diferenciável em $[a, b]$. Então existe $\xi \in (a, b)$ tal que

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

demonstração

Começamos construindo uma função auxiliar

$$\varphi(x) = f(x) - p(x) - (x - x_0)(x - x_1) \cdots (x - x_n)A,$$

observe que $\varphi(x_0) = \varphi(x_1) = \cdots = \varphi(x_n) = 0$, ou seja, $\varphi(x)$ se anula em $n+1$ pontos. Tome $z \in (a, b)$ distinto de (x_0, x_1, \dots, x_n) e escolhamos A tal que $\varphi(z) = 0$.

$\varphi(x)$ é $(n+1)$ vezes diferenciável, já que isso ocorre com $f(x)$ e $p(x)$. Assim podemos aplicar o Teorema de Rolle repetidas vezes e garantir a existência de $\xi \in (a, b)$ tal que:

$$0 = \varphi^{n+1}(\xi) = f^{n+1}(\xi) - (n+1)!A$$

donde

$$A = \frac{f^{n+1}(\xi)}{(n+1)!}$$

ou

$$E_t(z) = (z - x_0)(z - x_1) \cdots (z - x_n) \frac{f^{n+1}(\xi)}{(n+1)!}$$

Como z foi escolhido arbitrário temos o resultado. ■

A Proposição 3.1 garante a existência de ξ . Na prática não é possível encontrá-lo. Para resolver isso, tomamos $f^{n+1}(\xi)$ tal que $|f^{n+1}(\xi)| = \max\{|f^{n+1}(x)| \mid x \in (a, b)\}$. Assim podemos deixar uma questão em aberto. Se pudéssemos encontrar uma aproximação computacionalmente rápida e confiável para encontrar ξ teremos um polinômio interpolador mais preciso.

Uma observação que deve ser feita, vem do fato de que o polinômio interpolador é único de acordo com o Teorema 3.1, então o erro também é o mesmo, tanto para o polinômio de Lagrange, quanto para o polinômio obtido via DDF. Neste caso o polinômio interpolador obtido por DDF (por exemplo com três pontos $\{x_0, x_1, x_2\}$) é escrito como:

$$p(x) = f(x_0) + (x - x_0)p[x_0, x_1] + (x - x_0)(x - x_1)p[x_0, x_1, x_2]$$

com isto,

$$\begin{aligned} E_t(x) &= f(x) - p(x) \\ &= f(x) - (f(x_0) + (x - x_0)p[x_0, x_1] + (x - x_0)(x - x_1)p[x_0, x_1, x_2]) \\ &= (x - x_0)\left\{f[x, x_0] - \underbrace{p[x_0, x_1]}_{f[x_0, x_1]} - (x - x_1)\underbrace{p[x_0, x_1, x_2]}_{f[x_0, x_1, x_2]}\right\} \\ &= (x - x_0)(x - x_1)(x - x_2)f[x, x_0, x_1, x_2] \end{aligned}$$

Podemos generalizar para

$$E_t(x) = (x - x_0)(x - x_1) \cdots (x - x_n)f[x, x_0, x_1, \dots, x_n]$$

Comparando essa última equação com a Proposição 3.1 concluímos que:

$$f[x, x_0, x_1, \dots, x_n] = \frac{f^{n+1}(\xi)}{(n+1)!}$$

para algum ξ em (a, b) .

Exemplo 3.4.1. Considere a função $f(x) = \sin(x)$ no intervalo $[0, \frac{\pi}{2}]$. Vamos obter o polinômio interpolador via método de Lagrange nos pontos

$$\begin{aligned}(x_0, f(x_0)) &= (0, 0) \\ (x_1, f(x_1)) &= \left(\frac{\pi}{4}, \frac{\sqrt{2}}{2}\right) \\ (x_2, f(x_2)) &= \left(\frac{\pi}{2}, 1\right)\end{aligned}$$

assim

$$p(x) = \sum_{i=0}^2 \left[f(x_i) \prod_{j=0, j \neq i}^2 \frac{(x - x_j)}{(x_i - x_j)} \right] = \left(\frac{8 - 8\sqrt{2}}{\pi^2} \right) x^2 + \left(\frac{4\sqrt{2} - 2}{\pi} \right) x$$

Verificamos facilmente que $|f^3(x)|$ assume máximo igual a 1 em $[0, \frac{\pi}{2}]$ com $x = 0$. Dessa forma tomamos $f^3(\xi) = f^3(0) = -1$ e podemos calcular o erro máximo:

$$E_t(x) = (x - x_0)(x - x_1)(x - x_2) \frac{-1}{3!} = -x \left(x - \frac{\pi}{4} \right) \left(x - \frac{\pi}{2} \right) \frac{1}{6}$$

Como aplicação vejamos o erro no ponto $\frac{3\pi}{8}$.

$$E_t \left(\frac{3\pi}{8} \right) = 0.0303$$

Por comparação vamos calcular os valores exatos:

$$E_t \left(\frac{3\pi}{8} \right) = f \left(\frac{3\pi}{8} \right) - p \left(\frac{3\pi}{8} \right) = 0.018$$

Certamente o leitor deve estar se perguntando; não teria que resultar o mesmo valor? A resposta é não. Quando tomamos um valor aproximado para $f^3(\xi)$ teremos um valor aproximado para o erro. Portanto o erro calculado dessa forma será um valor maior que o erro exato cometido.

O erro pode ser visto geometricamente na figura 3.1.

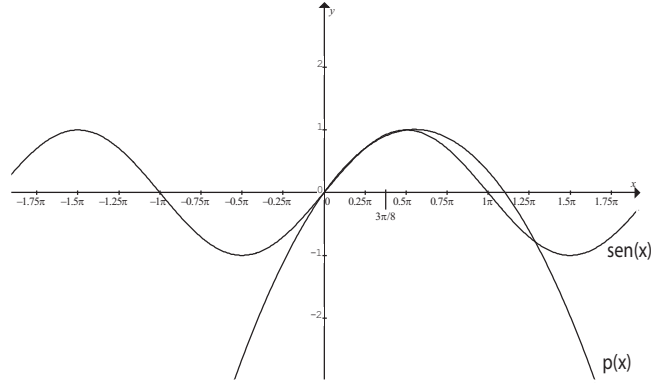


Figura 3.1: Erro

3.5 Método de Briot-Ruffini

Consideremos um polinômio de grau n , $p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ em \mathbb{R} . Veja que para calcular o valor $p(c)$ para um $c \in \mathbb{R}$ efetuamos $\frac{n(n+1)}{2}$ multiplicações e n adições. O método de Briot-Ruffini propõe reduzir o número de multiplicações.

Pela divisão euclidiana de $p(x)$ por $(x - c)$ temos:

$$p(x) = (x - c)Q(x) + R \quad (3.6)$$

onde R é uma constante, já que o grau de $p(x)$ é n e o grau de $Q(x)$ é $n - 1$.

Assim $p(c) = R$, ou seja, basta encontrar R . Para isso começamos escrevendo $Q(x)$ na forma geral:

$$Q(x) = b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \dots + b_1x + b_0 \quad (3.7)$$

e substituindo (3.7) em (3.6)

$$\begin{aligned}
p(x) &= (x - c)(b_{n-1}x^{n-1} + b_{n-2}x^{n-2} + \cdots + b_1x + b_0) + R \\
&= b_{n-1}x^n + (b_{n-2} - cb_{n-1})x^{n-1} + (b_{n-3} - cb_{n-2})x^{n-2} + \cdots + \\
&\quad + (b_1 - cb_2)x^2 + (b_0 - cb_1)x + (cb_0 + R)
\end{aligned}$$

Expandindo $p(x)$

$$a_nx^n + a_{n-1}x^{n-1} + \cdots + a_1x + a_0$$

é igual a

$$b_{n-1}x^n + (b_{n-2} - cb_{n-1})x^{n-1} + (b_{n-3} - cb_{n-2})x^{n-2} + \cdots + (b_1 - cb_2)x^2 + (b_0 - cb_1)x + (R - cb_0)$$

Portanto:

$$\begin{aligned}
b_{n-1} &= a_n \\
b_{n-2} &= a_{n-1} + cb_{n-1} \\
b_{n-3} &= a_{n-2} + cb_{n-2} \\
&\vdots \\
b_0 &= a_1 + cb_1 \\
R &= a_0 + cb_0
\end{aligned}$$

Dessa forma, R é facilmente encontrado e o número de multiplicações é igual a n .

Exemplo 3.5.1. Use o método de Briot-Ruffini para avaliar $p(x) = x^2 + 3x + 1$ em $c = 2$

$$\begin{aligned}
b_1 &= a_2 &= 1 \\
b_0 &= a_1 + cb_1 &= 5 \\
R &= a_0 + cb_0 &= 11
\end{aligned}$$

Assim $p(2) = 11$.

3.6 Considerações Finais

Uma observação importante que devemos fazer em relação a interpolação é a seguinte; quanto maior o número de pontos melhor é a

aproximação fornecida pelo polinômio interpolador. É bom lembrar que quanto mais pontos, maior é o grau do polinômio e assim mais lenta é sua avaliação, por isso é necessário a utilização de métodos de avaliação polinomial tal como o método de Briot-Ruffini¹.

3.7 Exercícios

3.7.1. Use interpolação de Lagrange para determinar uma aproximação de $f\left(\frac{\pi}{5}\right)$ sabendo que:

$$f(0) = 1, f\left(\frac{\pi}{3}\right) = \frac{1}{2}, f\left(\frac{\pi}{2}\right) = 0, f\left(\frac{\pi}{6}\right) = \frac{\sqrt{3}}{2}$$

3.7.2. Seja $f(x) = \frac{\cos(x)}{\sqrt[3]{x^2}}$. Calcule uma aproximação para $f\left(\frac{\pi}{3}\right)$ através do polinômio interpolador de Lagrange de grau 2.

3.7.3. Usando interpolação por DDF obtenha o valor aproximado de $f\left(\frac{1}{2}\right)$ sabendo que:

$$f(1) = 1, f\left(\frac{3}{2}\right) = \frac{2}{3}, f(2) = \frac{1}{2}$$

3.7.4. Os problemas vistos até agora são da forma:

Dado os pontos (x_0, x_1, \dots, x_n) , onde $f(x)$ é conhecida, pede-se para encontrar um valor $f(c)$ para algum $c \in [x_j, x_i]$ conhecido.

Pense agora no problema inverso:

Dado os pontos (x_0, x_1, \dots, x_n) , onde $f(x)$ é conhecida, pede-se para encontrar um valor c onde conhecemos o valor $f(c)$. Chamamos de interpolação inversa.

Determine o valor aproximado de c para $f(c) = 0.95$ usando os valores da função $f(x) = \sin(x)$ (x em radianos) dados abaixo:

i	0	1	2	3
x_i	1.75	1.80	1.85	1.90
$f(x_i)$	0.984	0.9738	0.9613	0.9463

¹Não é o único.

3.7.5. Verifique que, se $f(x)$ for um polinômio de grau n , então o polinômio interpolador $p(x)$ relativo a (x_0, x_1, \dots, x_n) é igual a $f(x)$.

3.7.6. Usando interpolação inversa, determine uma aproximação para um zero de:

a) $f(x) = x^3 - 9x + 10$

b) $g(x) = \ln(x) + 4x - 3$

3.7.7. Interpole a função $f(x) = e^x$ no intervalo $[0, 1]$ com 4 pontos. Calcule o erro de truncamento.

3.7.8. Mostre que $f[x_0, x_1, x_2] = f[x_2, x_1, x_0]$.

3.7.9. Seja $p(x)$ um polinômio de grau n . Mostre que é necessário $\frac{n(n+1)}{2}$ multiplicações e n adições para avaliar $p(c)$ para $c \in \mathbb{R}$ no modelo tradicional de substituição.

3.7.10. Calcule o número de adições e multiplicações para gerar o polinômio interpolador de Lagrange em $(n+1)$ pontos. obs: Conte a divisão como uma multiplicação.

3.7.11. Através de interpolação, encontre c tal que $f(c)$ seja o menor valor que $f(x) = x \ln(x) - x + 2x^2$ assume em $(0, \infty)$.

3.7.12. Determina-se empiricamente o alongamento de uma mola em milímetros, em função da carga P kgf que sobre ela atua, obtendo-se

x	5	10	15	20	25	30	35	40	mm
P	49	105	172	253	352	473	619	793	Kgf

Interpolando adequadamente por meio de polinômios de terceiro grau, encontre as cargas que produzem os seguintes alongamentos na mola:

a) 12mm

b) 22mm

c) 31mm

3.7.13. Numa rodovia federal, medindo-se a posição de um ônibus que partiu do marco zero da rodovia, obtiveram-se as seguintes marcações:

$T(\text{min})$	60	80	100	120	140	160	180
$D(\text{km})$	76	95	112	138	151	170	192

Pede-se o posicionamento do ônibus para os tempos de:

a) 95 min

b) 130 min

c) 170 min

3.7.14. A velocidade do som na água varia com a temperatura. Usando os valores da tabela abaixo, determine o valor aproximado da velocidade do som na água a 100°C

Temperatura ($^\circ\text{C}$)	86.0	93.3	98.9	104.4	110.0
Velocidade (m/s)	1.552	1.548	1.544	1.538	1.532

Capítulo 4

Integração Numérica

4.1 Introdução

A mais de 2000 anos, o cálculo da área e volume de figuras geométricas intriga a curiosidade humana. Os gregos, foram talvez os primeiros a tentar calcular a área de figuras complicadas por aproximações numéricas, um exemplo disso é o cálculo da área e comprimento do círculo pelo método da exaustão de Eudócio (408-355 a.C.) Com a invenção do Cálculo por Newton(1642-1727) e Leibnitz(1646-1716), vários problemas foram resolvidos, mas também surgiram outros. Para integrais, temos o Teorema Fundamental do Cálculo(T.F.C.), ou seja, se $f(x)$ é uma função contínua no intervalo $[a, b]$, então a área sob o gráfico de $f(x)$ é dado por:

$$\int_a^b f(x) dx = F(b) - F(a), \text{ onde } F \text{ é a antiderivada de } f(x).$$

Uma pergunta surge: onde o Cálculo Numérico pode contribuir, uma vez que o T.F.C. parece ter resolvido tudo? Ocorre que, nem sempre podemos aplicar o T.F.C. conforme os problemas abaixo:

i- De acordo com resultados do Cálculo, toda função contínua em um intervalo fechado possui antiderivada¹. Mas o cálculo desta nem sempre é simples. Por exemplo:

¹Isto é, f é integrável

$$\int_a^b e^{-x^2} dx$$

ii- Como calcular a integral de uma função conhecendo-a apenas em um número finito de pontos?.

Desta forma recorreremos a integração numérica² para resolver estes e outros problemas.

Na Interpolação polinômial aprendemos como gerar o polinômio interpolador, ou seja, $p(x) \cong f(x)$. Portanto $\int_a^b p(x) dx \cong \int_a^b f(x) dx$. Nesta linha, vamos desenvolver os métodos numéricos de integração.

4.2 Regra dos Trapézios

Começamos considerando a Interpolação de Lagrange (veja cap. anterior) em uma função $f(x)$ conhecida apenas em $(x_0, f(x_0)), (x_1, f(x_1))$. Seja então

$$p(x) = f(x_0) \frac{x - x_1}{x_0 - x_1} + f(x_1) \frac{x - x_0}{x_1 - x_0}$$

integrando

$$\begin{aligned} \int_{x_0}^{x_1} p(x) dx &= \frac{f(x_0)}{x_0 - x_1} \int_{x_0}^{x_1} (x - x_1) dx + \frac{f(x_1)}{x_1 - x_0} \int_{x_0}^{x_1} (x - x_0) dx \\ &= \frac{x_1 - x_0}{2} [f(x_1) + f(x_0)] \end{aligned}$$

Donde concluímos que $\int_{x_0}^{x_1} p(x) dx$ é exatamente a área do trapézio conforme figura 4.1.

Se tivermos $(n + 1)$ pontos, digamos (x_0, x_1, \dots, x_n) igualmente espaçados, ou seja, $x_k = x_{k-1} + h, (k = 1, \dots, n)$, então podemos generalizar (acompanhe com a figura 4.2) para:

²Também chamado de quadratura, devido a Arquimedes(387-212 a.C.) e o problema da quadratura do círculo

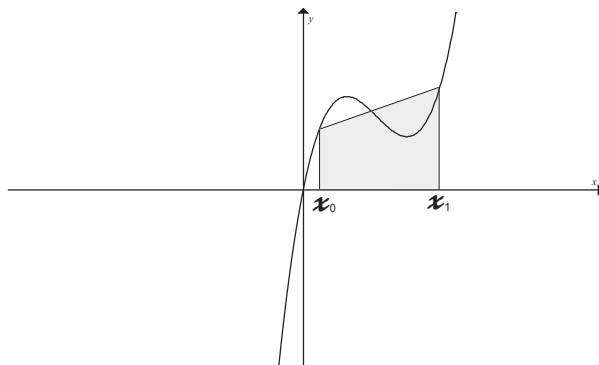


Figura 4.1: Área do trapézio

$$\begin{aligned}
 \int_{x_0}^{x_n} p(x) \, dx &= \int_{x_0}^{x_1} p_1(x) \, dx + \int_{x_1}^{x_2} p_2(x) \, dx + \cdots + \int_{x_{n-1}}^{x_n} p_n(x) \, dx \\
 &= \frac{x_1 - x_0}{2} [f(x_1) + f(x_0)] + \frac{x_2 - x_1}{2} [f(x_2) + f(x_1)] + \cdots + \\
 &\quad + \frac{x_n - x_{n-1}}{2} [f(x_n) + f(x_{n-1})] \\
 &= \frac{h}{2} [f(x_0) + 2f(x_1) + \cdots + 2f(x_{n-1}) + f(x_n)]
 \end{aligned}$$

Com $p_k(x)$ o polinômio interpolador nos pontos $(x_{k-1}, x_k), (k = 1, \dots, n)$.

4.2.1 Erro na Integral por Trapézios

De acordo com a seção 3.4 o erro na interpolação é dado por $E_t(x) = (x - x_0)(x - x_1) \frac{f''(\xi)}{2}$ para $\xi \in (x_0, x_1)$. Integrando teremos

$$\begin{aligned}
 \int_{x_0}^{x_1} E_t(x) \, dx &= \int_{x_0}^{x_1} (x - x_0)(x - x_1) \frac{f''(\xi)}{2} \, dx \\
 &= -\frac{f''(\xi)(x_1 - x_0)^3}{12}
 \end{aligned}$$

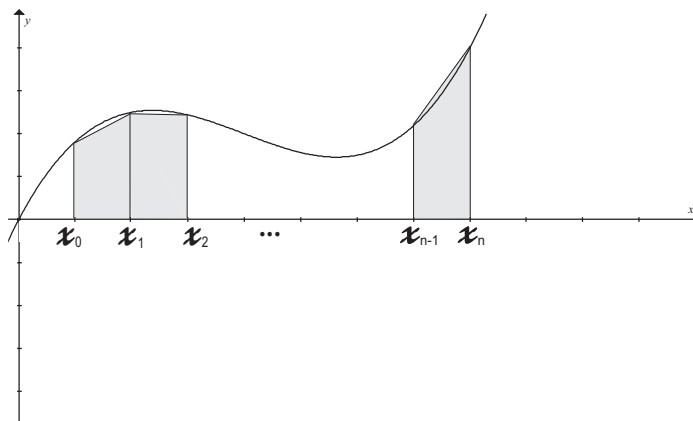


Figura 4.2: Cálculo da área por trapézios

Assumindo $x_1 - x_0 = h$, então o erro de integração é dado por

$$E_I = -\frac{f''(\xi)h^3}{12}$$

Lembre-se que, no caso geral (x_0, x_1, \dots, x_n) , temos que somar todos os erros, ou seja, o erro total será dado por:

$$-\frac{h^3}{12} \sum_{i=1}^{n-1} f''(\xi_i), \text{ onde } \xi_i \in (x_{i-1}, x_i) \text{ para } i = 1, \dots, n.$$

Exemplo 4.2.1. Considere $f(x)$ conhecida nos pontos $\left(\frac{\pi}{3}, \frac{\sqrt{3}}{2}\right), \left(\frac{5\pi}{6}, \frac{1}{2}\right)$.

Pela regra dos trapézios teremos:

$$\int_{x_0}^{x_1} f(x) dx = (x_1 - x_0) \frac{[f(x_1) + f(x_0)]}{2} = \frac{\pi(1 + \sqrt{3})}{8} \cong 1.07287$$

Sabendo que $f(x) = \sin(x)$ e $|f''(\xi)| = \max\{|f''(x)| ; x \in (x_0, x_1)\} = |f''(\pi/2)| = 1$ logo tomamos $f''(\xi) = f''(\pi/2) = -1$ calculamos o erro de integração

$$E_I = -\frac{f''(\xi)(x_1 - x_0)^3}{12} \cong 0.4880$$

Portanto

$$\int_{x_0}^{x_1} f(x) dx + E_I = 1.5409$$

Comparando este resultado com o valor fornecido pelo T.F.C. que é 1.3660 podemos observar que mesmo calculando o erro e o acrescentando a solução, ainda temos uma pequena diferença em relação ao valor exato fornecido pelo T.F.C. Como foi dito no capítulo anterior essa diferença decorre da aproximação para $f''(\xi)$.

4.3 1ª Regra de Simpson

A regra dos trapézios utiliza polinômios interpolantes de grau 1, já que trabalha de dois em dois pontos. No caso de três pontos, podemos ter uma interpolante de grau 2, ou seja, nosso erro de interpolação tende a ser menor. Dessa forma, se $f(x)$ é conhecida em três pontos (x_0, x_1, x_2) , então o polinômio interpolador de Lagrange será dado por:

$$p(x) = f(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + f(x_2) \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}$$

assim

$$\int_{x_0}^{x_2} p(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + f(x_2)].$$

O caso geral em $(n + 1)$ pontos é dado por:

$$\int_{x_0}^{x_2} p(x) dx = \frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

A comparação da 1ª regra de Simpson com a regra dos trapézios pode ser observada nas figuras 4.3 e 4.4.

O erro de integração pode ser rapidamente calculado:

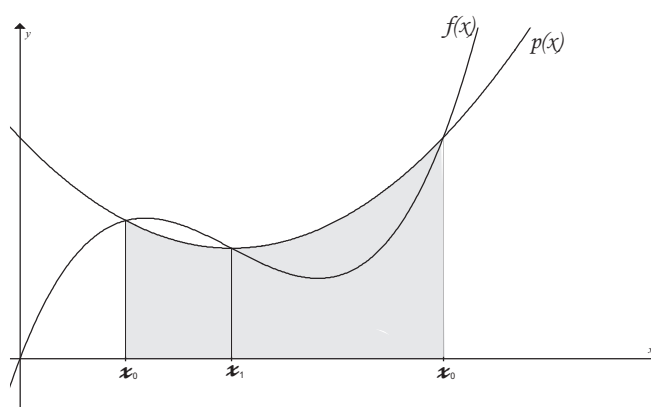


Figura 4.3: Cálculo da área pela 1ª regra de Simpson

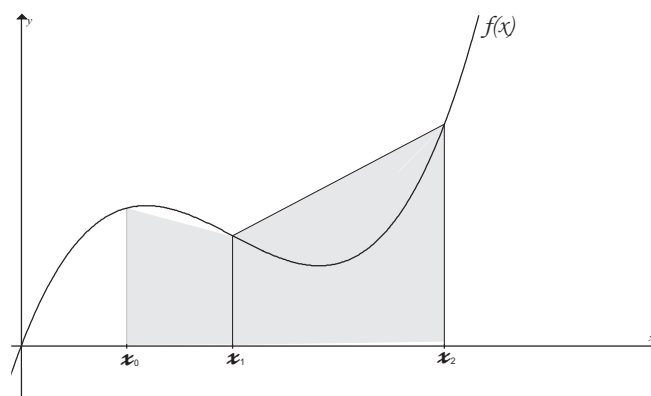


Figura 4.4: Cálculo da área por trapézios

$$\begin{aligned}\int_{x_0}^{x_2} E_t(x) \, dx &= \int_{x_0}^{x_2} (x - x_0)(x - x_1)(x - x_2) \frac{f^3(\xi)}{3!} \, dx \\ &= -\frac{f^3(\xi)h^5}{36}\end{aligned}$$

Portanto o erro na 1ª regra de Simpson será dado por:

$$E_I = -\frac{f^3(\xi)h^5}{36}$$

Lembrando que, no caso geral (x_0, x_1, \dots, x_n) onde n é par, temos que somar todos os erros, ou seja, o erro total será dado por:

$$-\frac{h^5}{36} \sum_{i=1}^{n-1} f^3(\xi_i), \text{ onde } \xi_i \in (x_{i-1}, x_i) \text{ para } i = 1, \dots, n.$$

Exemplo 4.3.1. Considere $f(x)$ conhecida apenas em

$$\begin{aligned}(x_0, f(x_0)) &= (\pi/3, \sqrt{3}/2) \\ (x_1, f(x_1)) &= (2\pi/3, \sqrt{3}/2) \\ (x_2, f(x_2)) &= (\pi, 0)\end{aligned}$$

Pela regra dos trapézios (observe que $h = \pi/3$) temos:

$$\int_{x_0}^{x_2} f(x) \, dx \cong \frac{h}{2}[f(x_0) + 2f(x_1) + f(x_2)] = 1.3603$$

Pela 1ª regra de Simpson temos:

$$\int_{x_0}^{x_2} f(x) \, dx \cong \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] = 1.5114$$

Qual resultado é o melhor? Certamente o valor fornecido pela 1ª regra de Simpson. Uma vez que o erro tende a ser menor.

Para convencer o leitor revelamos que $f(x) = \text{sen}(x)$, e calculamos o valor exato via T.F.C.

$$\int_{\pi/3}^{\pi} \text{sen}(x) \, dx = 1.50$$

Podemos obter a 2ª regra de Simpson com um processo análogo no polinômio interpolador de grau 3 gerado por 4 pontos. Mais detalhes veja exercício 4.5.7.

4.4 Quadratura Gaussiana

Nos métodos anteriores de Integração, usamos intervalos igualmente espaçados, ou seja, $x_k = x_{k-1} + h$. No método de Gauss estes pontos não são mais escolhidos, vamos definir um critério para essa escolha.

Começamos considerando o teorema de mudança de variável:

Teorema 4.1. *(Mudança de variável) Sejam $f : [a \ b] \longrightarrow \mathbb{R}$ contínua,*

$g : [c \ d] \longrightarrow \mathbb{R}$ com derivada integrável e $g([c \ d]) \subset [a \ b]$. Então

$$\int_{g(c)}^{g(d)} f(x) \, dx = \int_c^d f(g(t)) \cdot g'(t) \, dt.$$

A demonstração pode ser encontrada em [6]. Esse teorema será muito útil como veremos.

Para o teorema acima também podemos escrever $u = g(t)$ e $du = g'(t)dt$.

Nosso problema continua sendo tentar calcular $\int_a^b f(x) \, dx$. Para facilitar as contas vamos efetuar a seguinte mudança de variável.

Seja $g : [-1 \ 1] \longrightarrow \mathbb{R}$ tal que, $g(t) = \left(\frac{b-a}{2}\right)t + \frac{b+a}{2}$, observe que $g(-1) = a$ e $g(1) = b$. Aplicando o Teorema 4.1 temos:

$$\int_a^b f(x) \, dx = \int_{-1}^1 F(t) \, dt$$

onde

$$F(t) = f(g(t)).g'(t) = f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) \frac{b-a}{2}.$$

Com isto concluímos que para saber o valor de $\int_a^b f(x) dx$ basta calcular $\int_{-1}^1 F(t) dt$.

O método da Quadratura Gaussiana (não demonstraremos) afirma que:

$$\int_{-1}^1 F(t) dt = \sum_{i=0}^n w_i F(t_i)$$

onde w_k são chamados de pesos e t_k pontos do intervalo $[-1, 1]$.

Nosso objetivo é identificar estes pesos e pontos. Para simplificar começamos com $n = 1$, ou seja, dois pontos apenas. Dessa forma:

$$\int_{-1}^1 F(t) dt = w_0 F(t_0) + w_1 F(t_1) \quad (4.1)$$

Para descobrir estas quatro incógnitas necessitamos de um sistema com quatro equações. Uma vez que estas incógnitas não dependem de $F(t)$ podemos assumir $F(t) = t^k, k = 0, 1, 2, 3$. Ou seja,

$$\int_{-1}^1 t^k dt = w_0 F(t_0) + w_1 F(t_1)$$

$$k = 0 \implies 2 = \int_{-1}^1 t^0 dt = w_0 t_0^0 + w_1 t_1^0$$

$$k = 1 \implies 0 = \int_{-1}^1 t^1 dt = w_0 t_0 + w_1 t_1$$

$$k = 2 \implies \frac{2}{3} = \int_{-1}^1 t^2 dt = w_0 t_0^2 + w_1 t_1^2$$

$$k = 3 \implies 0 = \int_{-1}^1 t^3 dt = w_0 t_0^3 + w_1 t_1^3$$

com estas equações geramos o sistema:

$$\begin{cases} 2 &= w_0 + w_1 \\ 0 &= w_0 t_0 + w_1 t_1 \\ 2/3 &= w_0 t_0^2 + w_1 t_1^2 \\ 0 &= w_0 t_0^3 + w_1 t_1^3 \end{cases}$$

Resolvendo...

$$w_0 = w_1 = 1 \quad e \quad t_0 = -t_1 = 1/\sqrt{3}.$$

Substituindo estes valores na equação (4.1) temos:

$$\int_{-1}^1 F(t) dt = F(-1/\sqrt{3}) + F(1/\sqrt{3})$$

Com um processo análogo na forma geral

$$\int_{-1}^1 F(t) dt = \sum_{i=0}^n w_i F(t_i)$$

w_k e t_k podem ser calculados por polinômios de Legendre(veja [11]). Não faremos essa demonstração, mas os valores podem ser vistos na tabela 4.1.

Observação 4.1. *Se F é um polinômio de até grau 3, então esta fórmula fornece o valor exato da integral. Caso contrário o erro pode ser calculado pela fórmula abaixo:*

$$E = \frac{2^{(2n+3)} [(n+1)!]^4}{(2n+3) [(2n+2)!]^3} F^{(2n+2)}(\xi)$$

com $\xi \in [-1, 1]$.

É bom lembrar que $F^{(2n+2)}(\xi)$ deve ser tomado como o maior valor possível, como feito em seções anteriores.

n	i	t_i	w_i
1	0;1	± 0.57735027	1
2	0;1 2	± 0.77459667 0	$\frac{5}{9}$ $\frac{8}{9}$
3	0;1 2;3	± 0.86113631 ± 0.33998104	0.34785484 0.65214516
4	0;1 2;3 4	± 0.90617985 ± 0.53846931 0	0.23692688 0.47862868 0.53888889
5	0;1 2;3 4;5	± 0.93246951 ± 0.66120939 ± 0.23861919	0.17132450 0.36076158 0.46791394
6	0;1 2;3 4;5 6	± 0.94910791 ± 0.74153119 ± 0.40584515 0	0.12948496 0.27970540 0.38183006 0.41795918
7	0;1 2;3 4;5 6;7	± 0.96028986 ± 0.79666648 ± 0.52553242 ± 0.18343464	0.10122854 0.22238104 0.31370664 0.36268378

Tabela 4.1: Valores de w_k e t_k para $\int_{-1}^1 F(t) dt = \sum_{i=0}^n w_i F(t_i)$

Exemplo 4.4.1. Calcular $\int_0^{\pi/2} \text{sen}(x) \, dx$ por quadratura gaussiana com $n = 1$.

Começamos fazendo a mudança de variável

$$\begin{aligned} F(t) &= f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right) \frac{b-a}{2} \\ &= \text{sen}\left(\frac{\pi}{4}t + \frac{\pi}{4}\right) \frac{\pi}{4} \end{aligned}$$

dessa forma

$$\int_0^{\pi/2} \text{sen}(x) \, dx = \int_{-1}^1 F(t) \, dt = F(-1/\sqrt{3}) + F(1/\sqrt{3}) = 0.9984$$

Compare com o valor fornecido pelo T.F.C.

Exemplo 4.4.2. Calcular $\int_0^{\pi/2} \text{sen}(x) \, dx$ por quadratura gaussiana com $n = 2$.

De acordo com a tabela 4.1 temos

$$\begin{aligned} \int_{-1}^1 F(t) \, dt &= w_0 F(t_0) + w_1 F(t_1) + w_2 F(t_2) \\ &= \frac{5}{9} F(0.77459667) + \frac{5}{9} F(-0.77459667) + \frac{8}{9} F(0) \\ &= 1 \end{aligned}$$

Compare esse resultado com o valor fornecido pelo T.F.C.

Finalizamos lembrando o leitor, que este texto é apenas introdutório e abrange alguns métodos de integração numérica. Em [11] é possível encontrar outros métodos de integração.

4.5 Exercícios

4.5.1. Calcule o valor da integral via regra dos trapézios $\int_0^1 f(x) dx$ apenas sabendo que:

$$\begin{aligned}(x_0, f(x_0)) &= (0, 0) \\ (x_1, f(x_1)) &= (1/2, 1/4) \\ (x_2, f(x_2)) &= (1, 1)\end{aligned}$$

4.5.2. Calcule o valor da integral $\int_0^1 x^2 dx$ pela regra dos trapézios com 3 pontos.

4.5.3. Calcule pela regra dos trapézios com 3 pontos a integral $\int_2^3 \frac{1}{x^3} dx$ e o erro cometido.

4.5.4. Calcule o valor da integral $\int_0^\pi 2x \sin(x^2) dx$ com $n = 3$ pela regra dos trapézios e pela 1ª regra Simpson. Calcule o erro cometido na regra dos trapézios. Compare com o T.F.C.

4.5.5. Calcule o valor da integral $\int_0^3 \sin^{\pi/2}(x+1) \cos(x^2) dx$ com $n = 2$ pela regra dos trapézios e pela 1ª regra Simpson.

4.5.6. Mostre que

$$\int_{x_0}^{x_n} p(x) dx \cong$$

$$\frac{h}{3} [f(x_0) + 4f(x_1) + 2f(x_2) + 4f(x_3) + 2f(x_4) + \cdots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)]$$

para n par, na 1ª regra de Simpson.

4.5.7. Dados os pontos (x_0, x_1, x_2, x_3) mostre que a 2ª regra de Simpson é dada por:

$$\int_{x_0}^{x_3} f(x) \cong \frac{3}{8} h [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)]$$

e o caso geral

$$\int_{x_0}^{x_n} f(x) \cong$$

$$\frac{3}{8}h[f(x_0) + 3f(x_1) + 3f(x_2) + 2f(x_3) + 3f(x_4) + 3f(x_5) + 2f(x_6) + \cdots + 3f(x_{n-2}) + 3f(x_{n-1}) + f(x_n)]$$

4.5.8. Através da 2ª regra de Simpson, com $n = 3$ calcule a integral do exercício 4.5.4.

4.5.9. Através da 2ª regra de Simpson, com $n = 3$ calcule a integral $\int_2^3 \ln(x+2) - 1 \, dx$.

4.5.10. Calcule por quadratura gaussiana a integral $\int_{-2}^3 e^{-x^2} \, dx$ com $n = 1$ e $n = 2$. Calcule também o erro.

4.5.11. Calcule por quadratura gaussiana a integral do exercício 4.5.4.

Capítulo 5

Mínimos e Máximos

5.1 Introdução

Neste capítulo faremos uma breve introdução ao problema de encontrar mínimos e máximos de uma função. Na prática desejamos encontrar o valor mínimo ou máximo que a função assume num determinado intervalo. Esse problema tem aplicações imediatas nas engenharias, física e na própria matemática. No final, daremos um exemplo prático.

Começamos analisando funções de \mathbb{R} em \mathbb{R} .

Definição 5.1. *Seja $f : [a, b] \rightarrow \mathbb{R}$. Dizemos que k é mínimo local se, dado $\delta > 0$, para todo $x \in [a, b] \cap (k - \delta, k + \delta)$ temos $f(x) \geq f(k)$.*

A definição de máximo local segue análogo.

Definição 5.2. *Seja $f : [a, b] \rightarrow \mathbb{R}$. Dizemos que k é máximo local se, dado $\delta > 0$, para todo $x \in [a, b] \cap (k - \delta, k + \delta)$ temos $f(x) \leq f(k)$.*

Observe que o δ define o caráter local do mínimo ou máximo. Definimos agora o mínimo e máximo global, veja:

Definição 5.3. *Seja $f : [a, b] \rightarrow \mathbb{R}$. Dizemos que k é mínimo global se, para todo $x \in [a, b]$ temos $f(x) \geq f(k)$.*

Definição 5.4. *Seja $f : [a, b] \rightarrow \mathbb{R}$. Dizemos que k é máximo global se, para todo $x \in [a, b]$ temos $f(x) \leq f(k)$.*

Exemplo 5.1.1. Considere $f : [a, b] \rightarrow \mathbb{R}$ cujo gráfico é exibido na figura 5.1. Veja que k_0 e k_1 são mínimos locais com k_1 sendo o mínimo global.

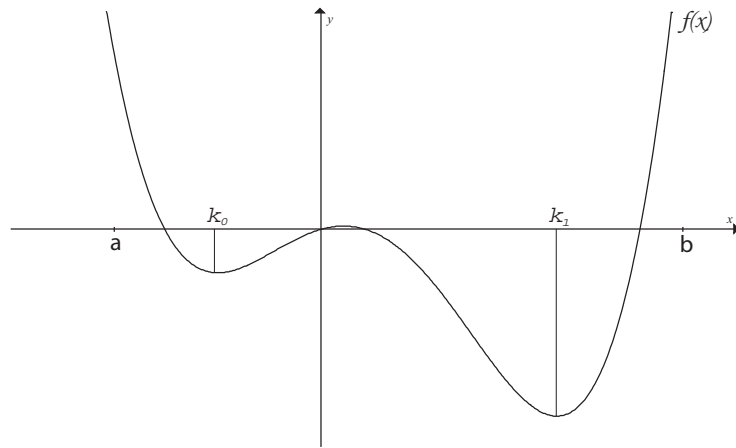


Figura 5.1: Mínimos local e global

Exemplo 5.1.2. Considere $f(x) = x^4 - 2x^3 - 4x^2 + 2x$ no intervalo $[-2, 2]$, cuja derivada será dada por $f'(x) = 4x^3 - 6x^2 - 8x + 2$. Aplicando os métodos para encontrar zeros de função podemos encontrar os zeros de $f'(x)$ que são: -1 , 0.2192 e 2.2808 . Avaliando $f(x)$ nesses pontos temos:

$$f(-1) = -3$$

$$f(0.2192) = 0.2274$$

$$f(2.2808) = -12.9149$$

Assim fica claro que 2.2808 é um ponto de mínimo global no intervalo $[-2, 2]$.

Lembramos o leitor que, se $k \in (a, b)$ é um ponto de máximo ou mínimo local, a derivada se anula em k . Mas se a derivada se anula em k não quer dizer que seja mínimo ou máximo local. Veja o exemplo abaixo.

Exemplo 5.1.3. Considere $f(x) = x^3$ com $f'(x) = 3x^2$. Veja que para $k = 0$, $f'(k) = 0$ mas k não é ponto de mínimo nem máximo. Dizemos que k é um ponto de inflexão, veja definição abaixo.

Definição 5.5. Seja $f(x)$ uma função diferenciável em (a, b) . Dizemos que k é ponto de inflexão horizontal se, existe $\delta > 0$ tal que uma das duas situações ocorre:

i) $f(x) < f(k)$ para todo $k - \delta < x < k$ e $f(x) > f(k)$ para todo $k < x < k + \delta$

ii) $f(x) > f(k)$ para todo $k - \delta < x < k$ e $f(x) < f(k)$ para todo $k < x < k + \delta$.

Observação 5.1. Caso $f(x)$ seja duas vezes diferenciável podemos visualizar o ponto de inflexão como o ponto onde $f''(x)$ muda de sinal.

Com estas definições não poderíamos deixar de enunciar o

Teorema 5.1. Seja $f : (a, b) \rightarrow \mathbb{R}$ uma função n vezes diferenciável e cujas derivadas, f', f'', \dots, f^n sejam contínuas em (a, b) . Seja $k \in (a, b)$ tal que $f'(k) = f''(k) = \dots = f^{n-1}(k) = 0$ e $f^n(k) \neq 0$. Então, se n é par,

$f^n(k) < 0$ implica que f tem máximo em k $f^n(k) > 0$ implica

que f tem mínimo em k .

Se n é ímpar, k é um ponto de inflexão horizontal.

A demonstração pode ser encontrada em [3].

Exemplo 5.1.4. Como aplicação do Teorema 5.1 vamos mostrar que $k = 1$ é ponto de mínimo de $f(x) = x^3 - 3x + 1$. Tome $f'(x) = 3x^2 - 3$, $f''(x) = 6x$, $f^3(x) = 6$, $f^4(x) = 0$, ..., $f^n(x) = 0$. Veja que $f'(k) = 0$ e $f''(k) = 6 > 0$, assim pelo Teorema temos que $k = 1$ é ponto de mínimo.

Devemos lembrar que o Teorema 5.1 tem suas limitações. Veja que uma vez dado k podemos aplicar o Teorema. O grande problema é encontrar k . Por isso, vamos desenvolver um método para encontrar tais pontos.

Para finalizar considere também o

Teorema 5.2. *Toda função contínua em um intervalo fechado possui mínimo/máximo global.*

A demonstração pode ser encontrada em [7].

Deste ponto em diante trataremos apenas o caso de encontrar mínimos de uma função. Uma vez que o mínimo de $f(x)$ é o máximo de $-f(x)$.

5.2 Método da Seção Áurea

Seja $f(x)$ uma função contínua com um único mínimo no intervalo $[a, b]$. O método da seção áurea consiste em criar uma sequência (x_n) que converge para o mínimo da função. A sequência (x_n) será dada por $x_n = \frac{a_n + b_n}{2}$ onde $[a_0, b_0] \supset [a_1, b_1] \supset \dots [a_n, b_n] \supset \dots$.

Assim seja $a_0 = a$ e $b_0 = b$.

$$a_1 = b_0 - 0.618(b_0 - a_0) \text{ e } b_1 = a_0 + 0.618(b_0 - a_0)$$

Se $f(a_1) < f(b_1)$, então $a_2 = a_1$ e $b_2 = a_1 + 0.618(b_1 - a_1)$.

Se $f(a_1) \geq f(b_1)$, então $a_2 = b_1 - 0.618(b_1 - a_1)$ e $b_2 = b_1$

Novamente

Se $f(a_2) < f(b_2)$, então $a_3 = a_2$ e $b_3 = a_2 + 0.618(b_2 - a_2)$.

Se $f(a_2) \geq f(b_2)$, então $a_3 = b_2 - 0.618(b_2 - a_2)$ e $b_3 = b_2$.

E o processo segue até que $|x_n - x_{n-1}|$ seja menor que um certo δ fixado como critério de parada.

Exemplo 5.2.1. *Seja $f(x) = x^2 - 2x + 3$. Vamos encontrar o mínimo de $f(x)$ em $[-2, 3]$ pelo método da seção áurea com $\delta = 0.06$.*

i	a_i	b_i	x_i	$ x_i - x_{i-1} $	
1	-0.09	1.09	0.5		
2	0.3608	1.09	0.7254	$0.2254 > \delta$	
3	0.6393	1.09	0.8647	$0.1393 > \delta$	
4	0.8115	1.09	0.9507	$0.0860 > \delta$	
5	0.9179	1.09	1.0030	$0.0523 < \delta$	\leftarrow parar o método

5.2.1 Convergência no Método da Seção Áurea

Observe que no método, utilizamos os intervalos $[a_0 \ b_0] \supset [a_1 \ b_1] \supset \cdots [a_n \ b_n] \supset \cdots$ com $a_0 \leq a_1 \leq \cdots \leq a_n \leq \cdots$ e $b_0 \geq b_1 \geq \cdots \geq b_n \geq \cdots$. Logo (a_n) e (b_n) são seqüências monótonas e limitadas. Pelo Teorema de Bolzano (veja em [3]) temos:

$$\lim_{n \rightarrow \infty} a_n = \lim_{n \rightarrow \infty} b_n = L$$

o que implica em

$$\lim_{n \rightarrow \infty} x_n = L.$$

Vamos mostrar que L é o mínimo que $f(x)$ assume em $[a \ b]$. Para isso, seja k o ponto mínimo em $[a \ b]$ garantido pelo Teorema 5.2. Assim $f(k) \leq f(a_i)$ e $f(k) \leq f(b_i)$ para $i = 0, \dots, \infty$, o que implica em $k \in [a_i \ b_i]$ para $i = 0, \dots, \infty$, ou seja, $a_i \leq k \leq b_i$ para $i = 0, \dots, \infty$. Aplicando o limite nesta última desigualdade teremos:

$$\lim_{n \rightarrow \infty} a_n \leq k \leq \lim_{n \rightarrow \infty} b_n,$$

ou seja,

$$L \leq k \leq L$$

o que implica em

$$L = k.$$

Observação 5.2. O número 0.618 é chamado de razão áurea. Que era considerado sagrado para os estudantes da escola grega pitagórica (fundada por Pitagors (586-500 a.C.)). Procure conhecer mais sobre esse famoso número e surpreenda-se¹.

5.3 Superfícies em \mathbb{R}^3

Para nosso estudo vamos considerar $f : D \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$, onde D é um retângulo em \mathbb{R}^2 . Por exemplo $f(x, y) = x^2 + y^2$ onde $D = [0 \ 1] \times [1 \ 2]$. Veja que o gráfico de $f(x)$ é o parabolóide dado na figura 5.2.

¹Veja mais em [4].

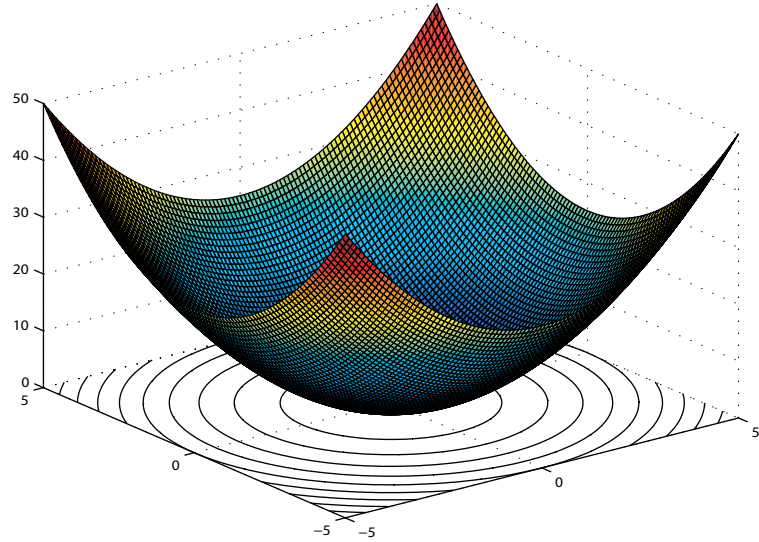


Figura 5.2: Parabolóide

Nosso problema agora é encontrar o mínimo global $k \in D$. Podemos também enunciar da seguinte forma: minimizar f na caixa $D = [a \ b] \times [c \ d]$.

Definição 5.6. *Seja $f : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função diferenciável. O vetor gradiente de f no ponto a será dado por:*

$$\nabla f(a) = \left(\frac{\partial f}{\partial x}(a), \frac{\partial f}{\partial y}(a) \right).$$

Exemplo 5.3.1. *Seja $f(x, y) = x^2 + y^2$. Então $\nabla f(x, y) = (2x, 2y)$. Dessa forma o gradiente de f no ponto $a = (2, 1)$ será dado por $\nabla f(2, 1) = (4, 2)$.*

Definição 5.7. *Uma curva de nível será dada pelo conjunto:*

$$S_c = \{(x, y) \in \mathbb{R}^2 / f(x, y) = c\}$$

Podemos visualizar S_c como os pontos em D com imagem igual a c , o que geometricamente corresponde a um corte na superfície na altura de c . Veja na figura 5.2 os círculos no plano XY. Cada círculo corresponde a superfície de nível.

Proposição 5.1. *Seja $f : D \subset \mathbb{R}^n \rightarrow \mathbb{R}$ uma função diferenciável. Então podemos afirmar que:*

i) o vetor gradiente $\nabla f(x)$ aponta para uma direção onde $f(x)$ é crescente.

ii) dentre todas as direções ao longo das quais $f(x)$ cresce, a direção do gradiente $\nabla f(x)$ é a de crescimento mais rápido.

iii) o gradiente $\nabla f(x)$ é perpendicular a superfície de nível que passa por x .

A demonstração dessas propriedades pode ser encontrada em [7].

Lembramos que, se estamos interessados no mínimo de $f(x)$, então devemos tomar a direção contrária a do gradiente, ou seja, $-\nabla f(x)$ para encontrar a direção onde $f(x)$ mais decresce.

5.4 Método do Gradiente

Tendo em mente a Proposição 5.1, vamos desenvolver um método numérico para localizar o mínimo de uma função em uma caixa D , utilizando vetor gradiente. Para isso, considere um ponto p_0 qualquer em D (acompanhe com a figura 5.3). Pelo item *ii* a direção $-\nabla f(p_0)$ tem maior decrescimento. Nosso objetivo é caminhar nesta direção para encontrar o mínimo de $f(x)$. Para nos auxiliar vamos definir a função $g_0(t) = f(p_0 - t\nabla f(p_0))$.

Observe que a função $g_0(t)$ é de uma variável. Podemos então, aplicar o método da seção áurea para encontrar seu mínimo, digamos p_1 .

Repetimos o processo para p_1 , ou seja, definimos $g_1(t) = f(p_1 - t\nabla f(p_1))$ e encontramos seu mínimo p_2 .

O processo segue até que a distância entre p_n e p_{n-1} seja menor que um certo δ fixado como critério de parada, nesse caso será dado por $\|p_i - p_{i-1}\|$. Também podemos utilizar como critério de parada o próprio vetor gradiente. Nesse caso, paramos o método se a norma do vetor gradiente em p_n for menor que um certo δ .

Essa distância pode ser calculada como a norma do vetor $p_n - p_{n-1}$ de acordo com a seção 1.3.2.

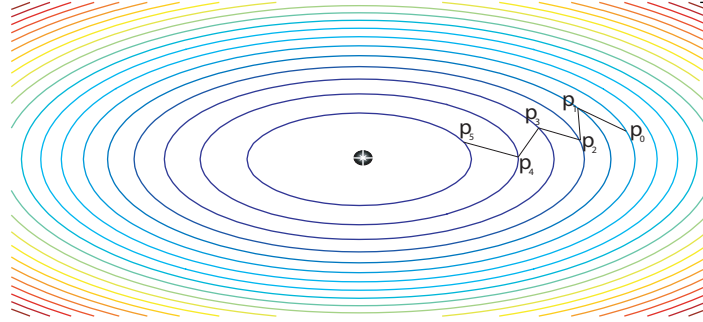


Figura 5.3: Seqüência (p_n) se aproximando do mínimo de $f(x)$

Observação 5.3. Lembramos que, ao aplicar o método da seção áurea na função $g_n(t)$, devemos ter cuidado de restringir t , de tal forma que, o vetor $p_n - t\nabla f(p_n)$ permaneça em D .

Exemplo 5.4.1. Considere $f : [-1 \ 1] \times [-1 \ 1] \rightarrow \mathbb{R}$ dada por $f(x, y) = x^2 + y^2$ (veja o gráfico na figura 5.2). Escolhemos $p_0 = (1, 1)$, logo $\nabla f(p_0) = (2, 2)$. Continuando temos $g(t) = f(p_0 - t\nabla f(p_0)) = f((1, 1) - t(2, 2)) = (1 - 2t)^2 + (1 - 2t)^2 = 8t^2 - 8t + 2$. Minimizando $g(t)$ pelo método da seção áurea temos $t_0 = \frac{1}{2}$. Calculamos agora $p_1 = p_0 - t_0\nabla f(p_0) = (1, 1) - \frac{1}{2}(2, 2) = (0, 0)$. Observe que o método convergiu em apenas uma iteração.

5.5 Bacias de atração

Podemos dizer que cada mínimo/máximo local de uma função diferenciável possui uma bacia de atração. Veja na figura 5.4 as várias bacias de atração de f . Cada bacia possui um mínimo ou um máximo local.

Para sintetizar considere a

Proposição 5.2. Se o método do gradiente for iniciado em um ponto p_0 não situado na bacia de atração do mínimo global; então podem ocorrer duas situações:

i) o método gradiente converge para o mínimo local associado a bacia de atração em que estiver o ponto p_0 .

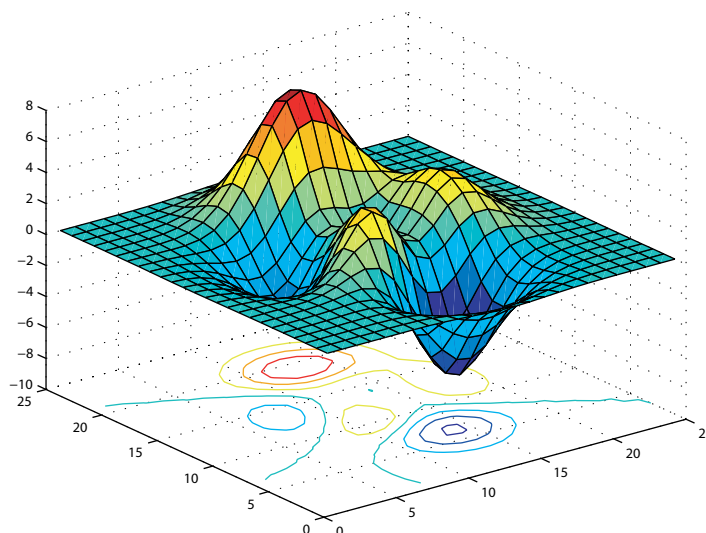


Figura 5.4: Bacias de atração

ii) Caso p_0 não esteja associado a nenhuma bacia de atração, então o método do gradiente não converge.

Assim temos um problema: se a função possui vários mínimos/máximos locais, como encontrar o mínimo/máximo global? A resposta é; iniciar o método do gradiente em vários pontos distintos. Dessa forma temos uma probabilidade de iniciar o método justamente na bacia de atração do mínimo/máximo global. E bom lembrar que esse método não é muito confiável.

5.6 Método de direções aleatórias

Seja $f : D \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ uma função apenas contínua em D . No método do gradiente escolhemos a direção do gradiente para encontrar máximos, e a direção contrária do gradiente para encontrar mínimos. Já no método de direções aleatórias, a direção de busca é aleatória. Para construir o método basta voltar no método do gradiente e substituir $\nabla f(p_n)$ por um vetor qualquer, veja:

Tome p_0 qualquer em D , e v_0 um vetor qualquer. Considere

$g_0(t) = f(p_0 + tv_0)$. Aplicando o método da seção áurea em g_0 encontramos seu mínimo em t_0 . Dessa forma $p_1 = p_0 + t_0v_0$. Novamente escolhemos outro vetor aleatório v_1 , tal que, $v_1 \neq v_0$ e consideramos $g_1(t) = f(p_1 + tv_1)$ cujo mínimo será t_1 .

O processo segue até que $\|p_n - p_{n-1}\|$ seja menor que um certo δ fixado como critério de parada.

Observação 5.4. Lembramos que o método do gradiente converge mais rápido que o método de direções aleatórias. Porém o método do gradiente exige que a função seja diferenciável em D .

Outro fato, no método do gradiente, se o ponto inicial p_0 não estiver em nenhuma bacia de atração, o método pode não convergir. No método de direções aleatórias isso pode não ocorrer, uma vez que as direções são escolhidas aleatoriamente.

Exemplo 5.6.1. Considere $f : [-1 \ 1] \times [-1 \ 1] \rightarrow \mathbb{R}$ dada por $f(x, y) = x^2 + y^2$ (veja o gráfico na figura 5.2). Definimos como critério de parada $\delta = 0.6$.

Escolhemos $p_0 = (1, 1)$ e $v_0 = (1, \frac{1}{2})$. Dessa forma $g_0(t) = f(p_0 + tv_0) = f(1+t, 1+\frac{1}{2}t) = (1+t)^2 + (1+\frac{1}{2}t)^2$. Aplicando o método da seção áurea, g_0 tem mínimo em $t_0 = -\frac{5}{6}$. Assim $p_1 = p_0 + t_0v_0 = (\frac{1}{6}, \frac{7}{12})$. Escolhemos o vetor aleatório $v_1 = (\frac{1}{4}, 1)$, e $g_1(t) = \frac{17}{16}t^2 + \frac{5}{4}t + \frac{53}{144}$. Onde g_1 tem mínimo em $t_1 = -\frac{10}{17}$, continuando $p_2 = (\frac{1}{51}, -\frac{1}{204})$. Paramos o método, pois já atingimos o delta desejado.

i	p_i	$\ p_i - p_{i-1}\ $	
0	(1, 1)	—	
1	(0.166, 0.583)	0.933 > δ	
2	(0.019, 0.0049)	0.596 < δ	← parar o método.

Observe que a sequência (p_n) converge para o ponto $(0, 0)$.

5.7 Exercícios

5.7.1. Apenas usando a derivada, calcule os pontos de máximo, mínimo e inflexão, da função $f(x) = 16x^3 - 22x - 5$ no intervalo $[-2 \ 2]$.

5.7.2. De acordo com o Teorema 5.2, ache os pontos máximo e mínimo de $f(x) = x^3 - 3x + 1$.

5.7.3. Aplique o método da seção áurea para encontrar o ponto de mínimo de $f(x) = x^2 - 5x + 1$ no intervalo $[1, 3]$. Compare o resultado com o fornecido pela derivada.

5.7.4. Apenas fazendo $\nabla f = 0$, encontre o ponto de mínimo de $f(x, y) = 5x^2 + 3y + 3x - 3y$.

Capítulo 6

Introdução ao Matlab

Faremos uma introdução ao algoritmo estruturado na linguagem português, e em paralelo o Matlab.

6.1 Introdução

Toda linguagem de programação é constituída essencialmente por COMANDOS e ESTRUTURAS DE CONTROLE que o computador é capaz de interpretar e executar. Os COMANDOS são ordens bastante primitivas como: Ler um valor; Imprimir um valor; Somar dois valores; Multiplicar dois valores; etc...etc...etc. As ESTRUTURAS DE CONTROLE são uma espécie de comandos de nível mais elevado, porque elas são utilizadas para gerenciar a execução dos comandos mais primitivos. Por exemplo: admitamos que eu queira bater o número 1 no teclado, e queira que o computador exiba na tela:

- a) Todos os números inteiros de 1 a 9.; b) Os pares de 10 a 20;
- c) Os ímpares de 20 a 30.

Para tal, deve-se ter a seguinte seqüência de comandos e estruturas de controle:

- 1 - Ler teclado (Comando: ordena o computador a armazenar na memória o número que eu digitar no teclado.

2 - Repetir 9 vezes (Estrutura de controle: gerencia a repetição dos comandos 2.1 e 2.2)

inicio

2.1 - Exibir teclado 2.2 - teclado + 1

fim

3 - Repetir 6 vezes

inicio 3.1 Exibir teclado 3.2 teclado + 2 fim

4 - teclado -1

5 - Repetir 5 vezes

inicio 5.1 - Exibir teclado 5.2 - teclado +2

fim

Os comandos e estruturas de controle que utilizamos no exemplo acima são apenas ilustrativos. Eles não existem de verdade nas linguagens. Mais adiante daremos esses comandos e estruturas de controle na linguagem Matlab.

ALGORITMO

Receita de bolo é uma expressão mais apropriada à arte culinária e à gastronomia. A receita de bolo é uma seqüência de comandos escritos para um cozinheiro executar. A seqüência de comandos escritos para um computador executar é chamada na verdade de ALGORITMO. Ou seja, a seqüência de comandos e estruturas de controle que escrevemos anteriormente é chamada de algoritmo. Um algoritmo é uma solução que você encontra para um determinado problema fazendo uso exclusivamente dos comandos e estruturas de controle que a linguagem de programação lhe oferece. Para resolver o mesmo problema você pode escrever inúmeros algoritmos diferentes. Tudo vai depender do seu estado de alma no momento. Se você

estiver inspirado, cheio de criatividade, você poderá resolver o problema como um algoritmo bem pequeno com poucas linhas. Se não for o seu dia, o algoritmo pode ficar longo e talvez nem funcionar.

Nosso objetivo central é aprender a escrever algoritmos; ou seja, aprender a encontrar solução para os problemas propostos fazendo o uso exclusivamente dos comandos e estruturas de controle.

6.2 Comandos

6.2.1 Comando de leitura

Ao executar um comando de Leitura o computador faz um pequeno cursor ficar piscando na tela, indicando que ele, o computador, está esperando que você digite um número (ou um caracter) no teclado. Assim que você digita o número, pressionando em seguida a tecla enter, o computador apanha (lê) o número que você digitou e o armazena em uma posição na sua memória.

Em Matlab

input

exemplo:

```
>>A=input('tecle um valor para A')
```

6.2.2 Comando de impressão

Este comando é utilizado quando se deseja que o computador exiba na tela os valores que tem armazenado em diversas posições na memória. Em português seria Imprima(A) e o computador exibe o valor da variável A.

Em Matlab

disp

exemplo:

>>disp(A) *exibe o valor da variável A*

>>disp('exibe este texto') *exibe uma mensagem*

6.2.3 Comando de atribuição

O comando de atribuição é também chamado de COMANDO ARITMÉTICO porque é nele que definimos as expressões aritméticas necessárias à resolução de um problema qualquer. Ao executar um comando de atribuição, o computador armazena na variável cujo nome é mencionado à esquerda do sinal de (=) o valor da expressão aritmética especificada à direita do sinal de (=).

Em Matlab

=

exemplo:

>>a=5 *atribui o valor 5 a variável a*

>>b=a+10 *atribui a variável b o valor de a mais 10*

>>b=b+1 *atribui a variável b o valor de b mais 1*

>>c='o mundo é tão lindo' *atribui a variável c a mensagem o mundo é tão lindo*

Observação 6.1. Lembramos que o Matlab faz distinção de maiúsculas e minúsculas. Assim a variável *c* difere da variável *C*. Outro fato; não é necessário declarar a variável antes de atribuir qualquer valor a ela.

Definição 6.1. Uma estrutura de controle é uma espécie de comando especial, de patente mais elevada, cujo objetivo é controlar a maneira como os comandos são executados.

6.2.4 Estrutura de decisão

Esta estrutura força o computador a tomar uma decisão quanto a executar ou não uma determinada seqüência de comandos. Existem na verdade duas estruturas distintas: a estrutura SE de um único ramo, ou, de uma única alternativa; e a estrutura SE de dois ramos, ou, de duas alternativas.

A estrutura SE de um único ramo (ou de uma única alternativa).

SE <COMPARAÇÃO>

Comandos e outras estruturas

FIM SE

Podemos entender a estrutura SE da seguinte forma: Se <COMPARAÇÃO> for verdadeira então Comandos e outras estruturas é executado. Caso contrário, ou seja, <COMPARAÇÃO> é falso, o computador dá um salto para logo após o FIM SE.

A estrutura SE de dois ramos (ou de duas alternativas).

SE <COMPARAÇÃO>

Comandos 1

SENÃO

Comandos 2

FIM SE

Nessa estrutura, se a <COMPARAÇÃO> for verdadeira então Comandos 1 é executado e Comandos 2 não é executado. Caso <COMPARAÇÃO> for falso então Comandos 2 é executado e Comandos 1 não é executado. Sempre um dos dois é executado.

Em Matlab

SE de um ramo	SE de dois ramos
<pre> if < > Comandos end </pre>	<pre> if < > Comandos 1 else Comandos 2 end </pre>

exemplo:

```

A=input('digite um valor para A')
B=input('digite um valor para B')

if A < B

    disp('A variável A é menor que a variável B')

else

    disp('A variável A é maior ou igual que a variável B')

end

```

6.2.5 Estruturas de repetição

Estrutura PARA

Esta estrutura serve para se repetir uma determinada quantidade de vezes a execução de um ou mais comandos. O formato geral da estrutura de repetição PARA é dado por:

PARA i ;Passo k; DE j ATÉ M

Comandos e outras estruturas.

FIM PARA

A variável i começa valendo j , e adicionando k ela cresce/decrece até M . Cada acréscimo da variável i Comandos e outras estruturas é executado. Caso $k=1$ ao final teremos $(M-j+1)$ repetição de Comandos e outras estruturas. Por exemplo:

```
PARA i;Passo 1; DE 1 ATÈ 10
```

Comandos e outras estruturas.

FIM PARA

Comandos e outras estruturas. É executado $(10-1+1)=10$ vezes.

Observação 6.2. *A variável i, j, k e M devem necessariamente ser inteiras.*

Em Matlab

```
for i=j:k:M
```

Comandos e outras estruturas

```
end
```

exemplo:

```
for i=3:1:5
```

```
disp('mundo')
```

```
end
```

A palavra mundo é repetida 3 vezes.

Estrutura ENQUANTO

O ENQUANTO é uma estrutura de repetição que funciona mais ou menos nos mesmos moldes da estrutura PARA. Ou seja, ENQUANTO e PARA são estruturas de repetição que diferem entre si basicamente nos detalhes. Ambas foram projetadas para fazer o computador repetir uma determinada quantidade de vezes a execução de um bloco de comandos. O formato geral da estrutura de repetição ENQUANTO é dado por:

```
ENQUANTO <COMPARAÇÃO>
```

```
    Comandos e outras estruturas
```

```
FIMENQUANTO
```

Enquanto a <COMPARAÇÃO> for verdadeira Comandos e outras estruturas é executado. O fim da repetição acontece quando <COMPARAÇÃO> se torna falsa. Na fase inicial o estudante comete um erro muito comum, ou seja, criar uma estrutura de repetição sem fim. Por exemplo:

```
A=5
```

```
ENQUANTO A < 10
```

```
    Imprima('mundo')
```

```
FIMENQUANTO
```

Veja que a condição ($A < 10$) é sempre verdadeira. O computador começa a imprimir a palavra mundo e nunca para. Por isso para a estrutura ENQUANTO terminar deve ocorrer algo dentro do Laço que torne a <COMPARAÇÃO> falsa. No exemplo acima façamos uma pequena mudança.

```
A=5
```

```
ENQUANTO A < 10
```

```
    Imprima('mundo')
```

```
A=A+1
```

```
FIMENQUANTO
```

Agora a palavra mundo é repetida apenas 5 vezes.

Em Matlab

```
While < >
```

```
Comandos e outras estruturas
```

```
end
```

exemplo:

```
A=5
```

```
while A < 10
```

```
disp('mundo')
```

```
A=A+1
```

```
end
```

6.3 Itens Básicos do Matlab

6.3.1 Operadores relacionais

<	Menor que
<=	Menor ou igual que
>	Maior que
>=	Maior ou igual que
==	Igual a
~=	Diferente de

6.3.2 Conectivos lógicos

$\&$	E
$ $	Ou
\sim	Não
Xor	Ou excludente

6.3.3 Funções Pré-definidas

$\cos(x)$	Retorna ao cosseno de x
$\sin(x)$	Retorna ao seno de x
$\tan(x)$	Retorna a tangente de x
$\csc(x)$	Retorna a co-secante de x
$\cot(x)$	Retorna a co-tangente de x
\wedge	Potência, exemplo: $5 \wedge 2 = 5^2$
$\exp(x)$	Retorna a e^x
$\log(x)$	Retorna ao $\ln(x)$
$\log_{10}(x)$	Retorna a $\log_{10}(x)$
$\text{sqrt}(x)$	Retorna a \sqrt{x}
$\text{abs}(x)$	Valor absoluto de x ou $ x $
$\text{isreal}(x)$	Verdadeiro para valores reais
$\text{fix}(x)$	Arredondamento na direção do zero
$\text{floor}(x)$	Arredondamento na direção de $-\infty$
$\text{ceil}(x)$	Arredondamento na direção de $+\infty$
$\text{round}(x)$	Arredondamento para o inteiro mais próximo
$\text{rem}(x, y)$	Resto da divisão de x por y
$\text{mod}(x, y)$	Resto com sinal.
$\det(A)$	Determinante da matriz A
$\text{inv}(A)$	Inversa da matriz A
$\text{size}(A)$	Retorna a dimensão da matriz A
$\text{isprime}(x)$	1 se x primo, 0 se x não é primo.
$\text{primes}(x)$	Seqüência de primos menores que x
$\text{gcd}(x, y)$	Máximo divisor comum de x e y
$\text{lcm}(x, y)$	Mínimo múltiplo comum de x e y
$\text{cross}(v, w)$	Produto vetorial $v \times w$
$\text{sum}(v. * w)$	Produto interno $\langle v, w \rangle$

6.3.4 Script

Rapidamente você vai notar que a janela do MatLab não é suficiente para escrever programas mais extensos. Por isso; será necessário criar um Script (Roteiro). Um Script nada mais é do que uma lista de comandos e estruturas que serão executados sequencialmente. Para criar um Script basta clicar em arquivo/novo.

Exemplo 6.3.1. *Vamos criar um Script para somar o primeiros 100 números.*

```
soma=0;

for i=1:1:100

    soma=soma+i;

end

disp(soma)
```

Para executar o Script basta escrever na linha de comando >> o nome do mesmo.

Lembramos que o ';' no final de cada atribuição faz com que os valores não sejam exibidos na tela.

Exemplo 6.3.2. *Façamos um Script para encontrar e imprimir o menor valor de 3 números.*

```
a=input('digite o primeiro número')

b=input('digite o segundo número')

c=input('digite o terceiro número')

if a < b

    if a < c
        disp(a)
        disp(' é o menor valor')
```

```

else
    disp(c)
    disp(' é o menor valor')
end

else

    if b < c
        disp(b)
        disp(' é o menor valor')
    else
        disp(c)
        disp(' é o menor valor')
    end

end

```

Exemplo 6.3.3. *Façamos um Script para ler um número inteiro positivo N e calcular S , onde:*

$$S = \frac{1}{1} + \frac{3}{2} + \frac{5}{3} + \frac{7}{4} + \cdots + \frac{I}{N}$$

$N = \text{input}(\text{'digite o número inteiro positivo } N\text{'})$

$S=0; I=1;$

$\text{for } j=1:1:N$

$S=S+(I/j);$

$I=I+2;$

end

$\text{disp}(\text{'o valor de } S \text{ é: '})$

$\text{disp}(S)$

Exemplo 6.3.4. *Façamos um Script para ler um número inteiro positivo $N > 1$ e imprimir os N primeiros termos da seqüência de Fibonacci.*

```
fibo = 0 1 1 2 3 5 8 13 21 34 55 ... X Y (X+Y) ...

N=input('digite o número inteiro positivo N > 1')

Disp('A seqüência será produzida abaixo')

X=0; Y=1

for i=2:1:N

    Z=X+Y
    X=Y;
    Y=Z;

end
```

Veja que nesse programa não utilizamos os comando *DISP* para exibir o número Z . Simplesmente retiramos o ponto e vírgula (;).

Exemplo 6.3.5. *Façamos um Script para ler um número inteiro positivo $N > 0$ e responder, se N é primo ou não. O Script deve terminar quando o N lido for negativo.*

```
N=input('digite o número inteiro positivo N > 0')

while (N > 0)
    X=2;
    Sinal=0;
    while (X <= (fix(sqrt(N))))
        if (rem(N,X))==0
            Sinal=1;
        end
        X=X+1;
    end
    if Sinal==0
```

```

    disp('Esse número é primo')
else
    disp('Esse número não é primo')
end
N=input('digite o número inteiro positivo N > 0')
end

```

6.4 Vetores e Matrizes

Um vetor pode ser considerado como uma variável múltipla, ou seja, uma variável que possui outras variáveis. Assim o vetor $V=[1 \ 0 \ 0 \ 5 \ 6]$ é considerado um vetor de 5 posições. Cada elemento do vetor V pode ser obtido por referência a posição. Assim $V(1)=1$, $V(2)=0$, $V(3)=0$, $V(4)=5$ e $V(5)=6$.

O mesmo vale para matrizes. Se $M=[1 \ 3 \ 0; 3 \ 3 \ 3; 2 \ 1 \ 8]$ temos a matriz $\begin{pmatrix} 1 & 3 & 0 \\ 3 & 3 & 3 \\ 2 & 1 & 8 \end{pmatrix}$. Podemos fazer referência da seguinte forma: $V(i,j)$ onde i é a linha e j a coluna. Assim $V(3,2)=1$, $V(1,3)=0$ etc...

No Matlab

```

>> V=[1 1 0]
V =
    1    1    0

```

```

>> 5*V
ans =
    5    5    0

```

```

>> K=[2 3 5]
K =
    2    3    5

```

```

>> V+K
ans =
    3    4    5

```

```

>> M=[1 3 0;3 3 3;2 1 8]
M =
     1     3     0
     3     3     3
     2     1     8

>> M(1,3)
ans = 0

>> M(2,3)
ans = 3

>> M(2,2)
ans = 3

>> c=7*M(2,2)
c = 21

>> 2*M
ans =
     2     6     0
     6     6     6
     4     2    16

```

Nota: Não poderíamos deixar de comentar o comando `size` que retorna as dimensões de uma matriz. Se `M` é uma matriz qualquer então o comando `[L,C]=size(M)` grava na variável `L` o número de linhas de `M` e na variável `C` o número de colunas. Caso `M` seja um vetor `L` é sempre igual a 1.

Exemplo 6.4.1. *Vamos escrever um Script para preencher um vetor com os números pares de 1 a 100.*

```

posi=1; num=2;

while posi < 100

    v(posi)=num;
    posi=posi+1;

```



```

num=num+2;

end

disp(V)

```

Exemplo 6.4.2. *Escreva um Script para preencher uma matriz 4×4 da seguinte forma:*

$$\begin{pmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{pmatrix}$$

```

for i=1:1:4
    for j=1:1:4
        M(i,j)=i-1;
    end
end

disp(M)

```

Exemplo 6.4.3. *3) Escreva um Script para colocar em ordem crescente o vetor abaixo:*

$$V=[0 \quad 1 \quad 1 \quad 5 \quad 5 \quad 2 \quad 3 \quad 8 \quad 1]$$

```

for i=1:1:9
    for j=1:1:9
        if v(j)>v(j+1)
            aux=v(j);
            v(j)=v(j+1);
            v(j+1)=aux;
        endif
    endfor
endfor

disp(v)

```

Nota: Em outras linguagens teríamos que criar um pequeno programa para multiplicar duas matrizes A e B . No Matlab essa rotina já está pronta, desde que seja possível a multiplicação. Basta efetuar $A*B$.

6.5 Funções em Matlab

Você certamente já utilizou algumas funções pré-definidas, por exemplo a função $\text{sqrt}(x)$ que retorna a raiz quadrada de x , a função $\text{cos}(x)$ que retorna o co-seno de x e muitas outras já descritas. Nesta seção vamos aprender a criar estas funções que são tão importantes para os problemas que queremos resolver.

Uma função em Matlab é bem parecida com uma função matemática, ou seja, dado um valor x a função associa a um valor y . No computador o valor x é chamado de entrada, y de saída e a associação de processamento.

Assim a função $F(x) = x^2$ pode ser construída da seguinte forma:

```
function [F] = F(x)
```

$$F = x^2$$

```
end
```

Quando retornamos para a linha de comando ($>>$) basta aplicar a função $F(x)$ em um número qualquer que teremos o quadrado desse número. Por exemplo:

```
>> F(2)  
ans=4
```

```
>> F(3)  
ans=9
```

Nota: Cada função deve ser salva com o mesmo nome do cabeçalho. Outro fato; cada função deve ter um arquivo diferente.

Exemplo 6.5.1. Vamos escrever uma função para receber como entrada, um número inteiro positivo N e tenha como saída:

i) 1 se o número for primo.

ii) 0 se o número não for primo.

```
function [eprimo]=eprimo(N)

    X=2;
    Sinal=1;
    while (X <= (fix(sqrt(N))))
        if (rem(N,X))==0
            Sinal=0;
        end
        X=X+1;
    end

    eprimo=Sinal

end
```

Exemplo 6.5.2. Vamos escrever uma função para receber dois valores x, y inteiros positivos e retornar ao quociente inteiro da divisão de x por y .

```
function [quociente]=quociente(x,y)

    quociente= $\frac{X - \text{rem}(x,y)}{y}$ 

end
```

Exemplo 6.5.3. Vamos escrever uma função para receber um vetor V qualquer e retornar o vetor em ordem crescente.

```
function [cresc]=cresc(V)

    [L,C]=size(V)
    for i=1:1:C-1
```

```

    for j=1:1:C-1
        if V(j)>V(j+1)
            aux=V(j);
            V(j)=V(j+1);
            V(j+1)=aux;
        end
    end
end

cresc=V

end

```

6.6 Gráficos Bidimensionais

O processo para criar um gráfico de uma função $f(x)$ é bem simples e parecido com aquele dado ensino fundamental, ou seja, criamos uma tabela de pontos com valores (x, y) e simplesmente marcamos estes pontos no plano. Veja passo a passo a construção do gráfico da função $f(x) = x^2$ no intervalo $[-3 \ 3]$.

1- Criamos a função $f(x)$ (como feito acima em $F(x) = x^2$)

2 - Criamos o vetor X com espaçamento de 1/100 no intervalo $[-3 \ 3]$.

$$x = -3 : 0.01 : 3$$

3 - Não sabemos o tamanho do vetor X por isso usaremos o função pré-definida `size()` para descobrir o tamanho do vetor X . e assim calcular o vetor $Y = f(X)$.

```

[L,C]=size(X)
for i=1:C
    Y = f(X(i));
end

```

4- Para traçar o gráfico basta usar a função pré-definida `plot()`. Essa função simplesmente marca no plano os pontos $(X(i), Y(i))$ dos vetores X e Y .

`plot(X, Y)` veja o resultado a figura 6.1.

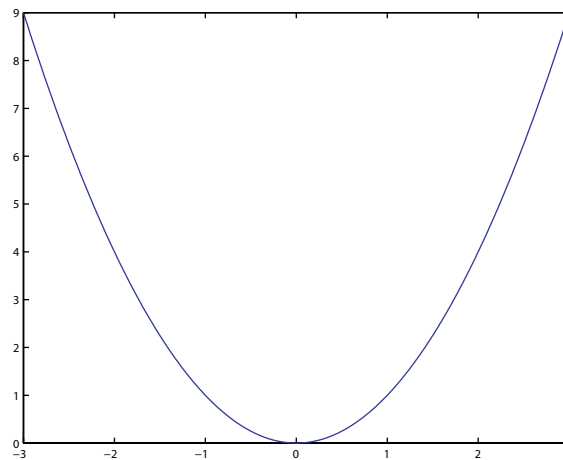


Figura 6.1: Gráfico de $f(x) = x^2$

Modo rápido: O modo com que criamos o gráfico acima, permite mais controle sobre o mesmo. Mas existe uma outra forma de construir mais rapidamente o gráfico de $f(x)$. Veja:

```
>> x = -3 : 0.01 : 3  
>> y = x.^2  
>> plot(x, y)
```

Observação 6.3. Para saber mais sobre a função `plot()` basta digitar `>> help plot`¹ na linha de comando. Para construir dois gráficos ao mesmo tempo basta utilizar o comando `hold on` depois de cada gráfico.

¹Isso pode ser feito com qualquer função pré-definida

6.7 Gráficos Tridimensionais

O processo para construir gráficos tridimensionais é bem parecido com o bidimensional. Vamos construir o gráfico de $f(x, y) = x^2 + y^2$ na caixa $D = [-1 \ 1] \times [-2 \ 3]$.

```
>> [X,Y] = meshgrid(-1 : 0.1 : 1, -2 : 0.1 : 3);
>> Z = X.^2 + Y.^2  Z é uma matriz, por isso, usamos X. e
Y. no cálculo.
>> surf(Z)  Veja figura 6.2 As curvas de nível podem ser
```

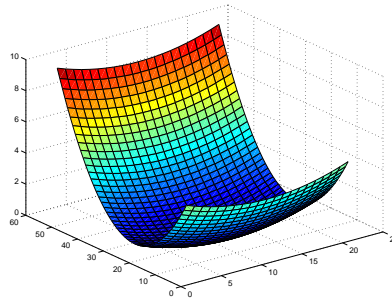


Figura 6.2: Gráfico de $f(x, y) = x^2 + y^2$

construídas com a função *contour*(*z*). Para desenhar gráfico + curva de nível use *surf**c*(*z*) (veja figura 6.3).

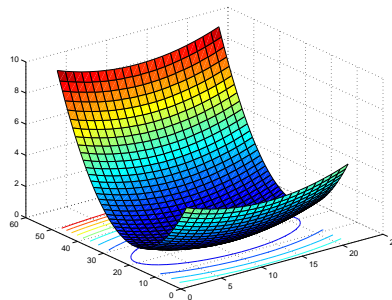


Figura 6.3: Gráfico+(curva de nível) de $f(x, y) = x^2 + y^2$

6.8 Exercícios

6.8.1. Escrever um *SCRIPT* que imprima em ordem decrescente os ímpares entre 500 e 100.

6.8.2. Escrever um *SCRIPT* para ler um inteiro positivo N e dizer se N é ou não múltiplo de 7. Terminar o *SCRIPT* quando N for negativo.

6.8.3. Escreva um *SCRIPT* para imprimir a seqüência:

1 2 3 4 -5 -6 -7 -8 9 10 11 12 -13 -14 -15 -16 17 18 19 20 ...
<1500

6.8.4. Escreva um *SCRIPT* para imprimir a seqüência:

1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 ... <1500

6.8.5. Escreva um *SCRIPT* para imprimir a seqüência de primos menores que 1000.

6.8.6. Escreva um *SCRIPT* para ler um inteiro positivo N e imprimir o valor de S onde:

$$S = \underbrace{\frac{0}{1} + \frac{1}{2} + \frac{1}{3} + \frac{2}{5} + \frac{3}{7} + \frac{5}{11} + \frac{8}{13} + \dots + \frac{F}{P}}_{N \text{ termos}}$$

F é a seqüência de Fibonacci

P é a seqüência de Primos

6.8.7. Escrever uma função em Octave para calcular a norma de um vetor de tamanho qualquer, onde

$$\|V\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2}$$

6.8.8. Escrever uma função em Octave para calcular a norma de uma Matriz A_{ij} onde,

$$\|A_{ij}\| = \sum_{ij} |a_{ij}|$$

6.8.9. Crie uma função para dizer se um "N" pertence ou não a seqüência abaixo:

1 2 4 7 11 16 17 19 22 26 31 32 34 37 41 46 ... 10000

6.8.10. Escreva uma função para calcular o módulo de uma matriz $n \times n$, onde

$$|A| = \text{Max}\{|a_{ij}|\}$$

6.8.11. Criar o gráfico da função $f(x) = 2x^2 + \text{sen}(x)$ no intervalo $[-2 \ 5]$.

6.8.12. Construa o gráfico da função $f(x, y) = \text{sen}(xy)$ na caixa $D = [-2 \ 2, -2 \ 2]$

Capítulo 7

Implementação dos Métodos

Neste capítulo vamos implementar os métodos numéricos que foram estudados anteriormente. Lembramos que o pré-requisito será apenas o conhecimento básico de algoritmos. Caso o leitor não tenha familiaridade com o Matlab, recomendamos a leitura do Capítulo de Introdução ao Matlab. Nossa ordem será:

- i)* Algoritmo em português.
- ii)* Algoritmo em Matlab (em forma de função).
- iii)* Quando possível, função pré-definida do Matlab.

7.1 Sistemas Lineares

Algoritmo 7.1.1. *Método da substituição retroativa, ou retro-substituição. Sistema triangular superior.*

Entrada : $a_{ij}, i \geq j, b_i, j \leq n$ (a_{ij} elementos da M. dos Coeficientes).

```

1 :       $x_n = b_n/a(k, k)$ 
2 :      para  $k = n - 1 : 1$ 
3 :           $soma = b_k$ 
4 :          para  $j = k + 1 : n$ 
5 :               $soma = soma - a_{kj}x_j$ 
6 :          fim para
7 :           $x_k = \frac{soma}{a_{kk}}$ 
8 :      fim para

```

Saída : x

NO MATLAB

```

function [retro]=retro(A)
[L,C]=size(A);
n=L;
m=C;
x(n)=A(n,m)/A(n,n);
for k=n-1:-1:1
    soma=A(k,m);
    for j=k+1:n
        soma=soma-A(k,j)*x(j);
    end
    x(k)=soma/A(k,k);
end
retro=x

```

Nota: Veja que no Matlab, usamos a matriz ampliada do sistema (matriz dos coeficientes + vetor de termos independentes), assim trocamos os elementos b_i por $A(i, m)$.

Exemplo:

Vamos aplicar a função `retro()` no sistema
$$\begin{cases} 2x_1 + x_2 - x_3 = 0 \\ x_2 - x_1 = -2 \\ 1/2x_3 = 2 \end{cases}$$

```
>> A = [2 1 -1 0; 0 1 -1 -2; 0 0 1/2 1]
>> retro(A)
ans =
    1    0    2
```

Algoritmo 7.1.2. *Método da eliminação Gaussiana.*

Entrada: $A = [a_{ij}]$ Matriz $n \times m$

```
1 :      para  $k = 1 : n$ 
2 :          se  $a_{kk} = 0$ 
3 :              para  $s = k + 1 : n$ 
4 :                  se  $a_{sk} \neq 0$ 
5 :                      troque a linha  $s$  com linha  $k$ 
6 :                  fim se
7 :              fim para
8 :          fim se
9 :          para  $i = k + 1 : n$ 
10 :               $m_{ik} = a_{ik}/a_{kk}$ 
11 :              para  $j = k : m$ 
12 :                   $a_{ij} = a_{ij} - m_{ik}a_{kj}$ 
13 :              fim para
14 :          fim para
15 :      fim para
```

Saída: matriz triangular $A = [a_{ij}]$

NO MATLAB

function [egauss]=egauss(A)

[L,C]=size(A);

n=L;

m=C;

for k=1:n

if A(k,k)==0

for s=k+1:n

if A(s,k) ==0

auxiliar=A(k,:);

```

    A(k,:)=A(s,:);
    A(s,:)=auxiliar;
end
end
end
for i=k+1:n
    m(i,k)=A(i,k)/A(k,k);
    for j=k:m
        A(i,j)=A(i,j)-m(i,k)*A(k,j);
    end
end
end
egauss=A

```

Nota: A função `egauss()` pode ser composta com a função `retro()` para resolver uma sistema $n \times n$ qualquer, ou seja, `retro(egauss(A))`, onde A é a matriz ampliada do sistema.

Exemplo:

$$\text{Considere o sistema } \begin{cases} 2x_1 + x_2 - x_3 & = & 0 \\ x_1 + x_2 - x_1 & = & -1 \\ 4x_1 - x_2 - 1/2x_3 & = & 3 \end{cases}$$

```
>> A = [2 1 -1 0; 1 1 -1 -1; 4 -1 -1/2 3]
```

```
>> retro(egauss(A))
```

```
ans =
```

```
1 0 2
```

Algoritmo 7.1.3. Método da eliminação Gaussiana com pivotea-

mento.

Entrada : $A = [a_{ij}]$ Matriz $n \times m$

```

1 :      para  $k = 1 : n$ 
2 :           $w = |a_{kk}|$ 
3 :      para  $s = k + 1 : n$ 
4 :          se  $|a_{sk}| > w$ 
5 :               $w = |a_{sk}|, r = s$ 
6 :          fim se
7 :      fim para
8 :      troque a linha  $s$  com linha  $k$ 
9 :      para  $i = k + 1 : n$ 
10 :           $m_{ik} = a_{ik}/a_{kk}$ 
11 :          para  $j = k : m$ 
12 :               $a_{ij} = a_{ij} - m_{ik}a_{kj}$ 
13 :          fim para
14 :      fim para
15 :  fim para

```

Saída : matriz triangular $A = [a_{ij}]$

NO MATLAB

```

function [egausspiv]=egausspiv(A)
[L,C]=size(A);
n=L;
m=C;
for k=1:n
    w=abs(A(k,k));
    for s=k+1:n
        if abs(A(s,k))>w
            w=abs(A(s,k));
            r=s;
        end
    end
    for i=k+1:n
        m(i,k)=A(i,k)/A(k,k);

```

```

for j=k:m
    A(i,j)=A(i,j)-m(i,k)*A(k,j);
end
end
end
egausspiv=A;

```

Algoritmo 7.1.4. *Decomposição LU.*

Entrada : $A_{n \times n}$

```

1 :      para i = 1 : n
2 :          para j = i : n
3 :               $u_{ij} = a_{ij} - \sum_{k=1}^{i-1} m_{ik}u_{kj}$ 
4 :              para j = i + 1 : n
5 :                   $m_{ji} = \left( a_{ji} - \sum_{k=1}^{i-1} m_{jk}u_{kj} \right) / u_{ii}$ 
6 :              fim para
7 :          fim para
8 :      fim para

```

Saída : matrizes L e U

NO MATLAB

```

function [fatlu]=fatlu(A)
[L,C]=size(A); n=L;
for i=1:n
    for j=i:n
        u(i,j)=A(i,j);
        for k=1:i-1
            u(i,j)=u(i,j)-m(i,k)*u(k,j);
        end
    end
end
for j=i+1:n

```

```

    m(j,i)=A(j,i);
    for k=1:i-1
        m(j,i)=m(j,i)-m(j,k)*u(k,i);
    end
    u(i,j)=u(i,j)/u(i,i);
end
end
L=m U=u

```

Nota: Não fizemos nesse algoritmo, mas é necessário completar a diagonal principal da matriz L com 1.

Exemplo: Considere a matriz $A = \begin{pmatrix} 1 & 1 & 2 \\ 1 & 2 & 3 \\ 0 & 1 & 5 \end{pmatrix}$.

```
>> A = [1 1 2; 1 2 3; 0 1 5]
```

```
>> fatlu(A)
```

```
L =
```

```
0 0
```

```
1 0
```

```
0 1
```

```
U =
```

```
1 1 2
```

```
0 1 1
```

```
0 0 4
```

Algoritmo 7.1.5. Método Iterativo de Jacobi.

Entrada : $A_{n \times n}, b_n, x^0, \delta$

```

1 :       $k = 1$ 
2 :      enquanto  $\|x^k - x^{k-1}\| > \delta$ 
3 :          para  $i = 1 : n$ 
4 :               $x_i^{k+1} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1, j \neq i}^n a_{ij} x_j^k \right)$ 
5 :          fim para
6 :      fim enquanto

```

Saída : solução aproximada x^k

NO MATLAB

```

function [jacobi]=jacobi(A,x,delta)
[L,C]=size(A);
n=L;
m=C;
k=2;
x(:,2)=1;
cont=1;
while norm(x(:,k)-x(:,k-1))>delta
    k=k+1;
    if cont==1
        k=k-1;
        cont=0;
    end
    for i=1:n
        x(i,k)=A(i,m);
        for j=1:n
            if j == i
                x(i,k)=x(i,k)-A(i,j)*x(j,k-1);
            end
        end
        x(i,k)=x(i,k)/A(i,i)
    end
end

```

end
jacobi=*x*;

Nota: A saída da função *jacobi()* será dada pelo vetor *x* na seguinte formatação:

$$\begin{array}{cccc}
 \text{solução} \rightarrow & x^1 & x^2 & \dots & x^k \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & x_1^1 & x_1^2 & \dots & x_1^k \\
 & x_2^1 & x_2^2 & \dots & x_2^k \\
 & \vdots & \vdots & \vdots & \vdots \\
 & x_n^1 & x_n^2 & \dots & x_n^k
 \end{array}$$

Algoritmo 7.1.6. *Método Iterativo de Gauss-Seidel*

Entrada : $A_{n \times n}, b_n, x^0, \delta$

```

1 :       $k = 1$ 
2 :      enquanto  $\|x^k - x^{k-1}\| > \delta$ 
3 :      para  $i = 1 : n$ 
4 :           $x_i^{k+1} = \frac{1}{a_{ii}} \left( b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^n a_{ij} x_j^k \right)$ 
5 :      fim para
6 :      fim enquanto

```

Saída : solução aproximada x^k

NO MATLAB

function [gausseidel]=gausseidel(A,x,delta)

```

[L,C]=size(A);
n=L;
m=C;
k=2;
x(:,2)=1;
cont=1;
while norm(x(:,k)-x(:,k-1))>delta
    k=k+1;
    if cont==1
        k=k-1;
        cont=0;
    end
    for i=1:n
        x(i,k)=A(i,m);
        for j=1:i-1
            x(i,k)=x(i,k)-A(i,j)*x(j,k);
        end
        for j=i+1:n
            x(i,k)=x(i,k)-A(i,j)*x(j,k-1);
        end
        x(i,k)=x(i,k)/A(i,i);
    end
end
gausseidel=x;

```

Nota: A saída da função `gausseidel()` será dada pelo vetor x na

seguinte formatação:

$$\begin{array}{cccc}
 \text{solução} \rightarrow & x^1 & x^2 & \dots & x^k \\
 & \downarrow & \downarrow & \downarrow & \downarrow \\
 & x_1^1 & x_1^2 & \dots & x_1^k \\
 & x_2^1 & x_2^2 & \dots & x_2^k \\
 & \vdots & \vdots & \vdots & \vdots \\
 & x_n^1 & x_n^2 & \dots & x_n^k
 \end{array}$$

Exemplo:

Vamos achar a solução do sistema $\begin{cases} 3x_1 + x_2 = 4 \\ 2x_1 + 6x_2 = 8 \end{cases}$ pelo método de Jacobi e em seguida pelo método de Gauss-Seidel com $\delta = 0.0001$

NO MATLAB

```
>> A = [3 1 4; 2 6 8]
```

```
A =
     3     1     4
     2     6     8
```

```
>> jacobi(A,[0 0]',0.0001)
```

Columns 1 through 7

```

0    1.3333    0.8889    1.0370    0.9877    1.0041    0.9986
0    1.3333    0.8889    1.0370    0.9877    1.0041    0.9986
```

Columns 8 through 11

```

1.0005    0.9998    1.0001    1.0000
1.0005    0.9998    1.0001    1.0000
```

```
>> gausseidel(A,[0 0]',0.0001)
```

ans

0	1.3333	1.0370	1.0041	1.0005	1.0001	1.0000
0	0.8889	0.9877	0.9986	0.9998	1.0000	1.0000

Observe que o método de Jacobi retornou a solução exata (1,1) com $\delta = 0.0001$ em 11 iterações, incluindo a solução inicial, enquanto o método de Gauss-Seidel retornou a solução exata em 7 iterações. Essa diferença tende a aumentar com o tamanho do sistema.

7.2 Zeros de Função

Algoritmo 7.2.1. Método de Localização de Zeros.

Entrada: $f(x), a, b, n$ n é o número de subintervalos

```

1 :       $\gamma = \frac{|b-a|}{n}$ 
2 :       $p_0 = a$ 
3 :      para  $i = 1 : n$ 
4 :           $p_i = p_{i-1} + \gamma$ 
5 :          se  $f(p_i)f(p_{i-1}) \leq 0$ 
6 :              armazenar o intervalo  $[p_{i-1} \ p_i]$  em um vetor  $V$ 
7 :          fim se
8 :      fimpara
```

Saída: Vetor V

NO MATLAB

```

function [MLZ]=MLZ(a,b,n)
gama=abs(b-a)/n;
p(1)=a; %trocamos p(0) por p(1)
j=1;
for i=2:n+1
    p(i)=p(i-1)+gama;
    if(f(p(i))*f(p(i-1)))<=0
```

```

    V(j)=p(i-1);
    V(j+1)=p(i);
    j=j+2;
end
end
MLZ=V;

```

Nota: O retorno da função $MLZ()$ é dado pelos intervalos $[v_1 \ v_2], [v_3 \ v_4], \dots, [v_{k-1} \ v_k]$ onde estão os possíveis zeros de $f(x)$. O vetor V é dado na seguinte formatação:

$$MLZ = V = \begin{bmatrix} v_1 & v_2 & v_3 & v_4 & \dots & v_{k-1} & v_k \end{bmatrix}$$

Exemplo:

Seja $f(x) = x^3 - 2x^2 + 1$ cujos zeros são $-0.6180, 1, 1.6180$. Vamos aplicar a função $MLZ()$ no intervalo $[-1 \ 2]$ com $n = 10$ depois com $n = 500$.

```
>> MLZ(-1,2,10)
```

```
ans =
    -0.7000    -0.4000     0.8000     1.1000     1.4000     1.7000
```

```
>> MLZ(-1,2,500)
```

```
ans =
    -0.6220    -0.6160     0.9980     1.0040     1.6160     1.6220
```

Veja que os zeros de $f(x)$ estão nos intervalos $[v_i \ v_{i+1}]$.

Algoritmo 7.2.2. Método do Meio Intervalo - MMI.

Entrada : $f(x), a, b, \delta$

```

1 :      enquanto  $|b - a| > \delta$ 
2 :       $x = \frac{a + b}{2}$ 
3 :      se  $f(x) = 0$ 
4 :       $x$  é zero, encerrar
5 :      senão
6 :      se  $f(a)f(x) < 0$ 
7 :       $b = x$ 
8 :      senão
9 :       $a = x$ 
10 :     fim se
11 :     fim se
12 :     fim enquanto

```

Saída : x aproximação para o zero de f em $[a \ b]$

NO MATLAB

```

function [MMI]=MMI(a,b,delta)
while abs(b-a)>delta
    x=(a+b)/2;
    if f(x)==0
        MMI=x;
        return %termina a função
    else
        if (f(a)*f(x))<0
            b=x;
        else
            a=x;
        end
    end
end
MMI=x;

```

De acordo com o exemplo do Algoritmo 7.2.1, temos que $f(x)$ possui zeros em $[-0.7 \quad -0.4]$, $[0.8 \quad 1.1]$, $[1.4 \quad 1.7]$. Aplicando a função $MMI()$ em cada intervalo com $\delta = 0.01$ temos:

```
>> MMI(-0.7, -0.4, 0.01)
ans =
    -0.6156
```

```
>> MMI(0.8, 1.1, 0.01)
ans =
    0.9969
```

```
>> MMI(1.4, 1.7, 0.01)
ans =
    1.6156
```

Tente fazer com $\delta = 0.0001$, veja que os resultados são melhores.

Algoritmo 7.2.3. Método da Secante.

Entrada : $f(x), a, b, \delta$

```
1 :      se  $f(a)f''(a) > 0$ 
2 :           $c = a, x_0 = b$ 
3 :      senão
4 :           $c = b, x_0 = a$ 
5 :      fim se
6 :       $x_1 = x_0 - \frac{f(x_0)}{f(x_0) - f(c)}(x_0 - c)$ 
7 :       $n = 1$ 
8 :      enquanto  $|x_n - x_{n-1}| > \delta$ 
9 :           $n = n + 1$ 
10 :       $x_n = x_{n-1} - \frac{f(x_{n-1})}{f(x_{n-1}) - f(c)}(x_{n-1} - c)$ 
11 :      fim enquanto
```

Saída : x_n

NO MATLAB

```

function[secante]=secante(a,b,delta)
if (f(a)*ddf(a))>0    %ddf() é a derivada segunda de f(x)
    c=a;
    x(1)=b;
else
    c=b;
    x(1)=a;
end
x(2)=x(1)-(f(x(1))*(x(1)-c))/(f(x(1))-f(c));
n=2;
while abs(x(n)-x(n-1))>delta
    n=n+1;
    x(n)=x(n-1)-(f(x(n-1))*(x(n-1)-c))/(f(x(n-1))-f(c));
end
secante=x(n);

```

De acordo com o exemplo do Algoritmo 7.2.1, temos que $f(x)$ possui zeros em $[-0.7 \quad -0.4]$, $[0.8 \quad 1.1]$, $[1.4 \quad 1.7]$. Aplicando a função `secante()` em cada intervalo com $\text{delta} = 0.01$ temos:

```

>> secante(-0.7,-0.4,0.01)
ans =
    -0.6163

```

```

>> secante(0.8,1.1,0.01)
ans =
    1.0003

```

```

>> secante(1.4,1.7,0.01)
ans =
    1.6169

```

Tente fazer com $\text{delta} = 0.0001$, veja que os resultados são melhores. Compare com o `MMI()`.

Nota: A função `ddf()` deve ser criada como a função `f()`.

Algoritmo 7.2.4. *Método de Newton.*

Entrada : $f(x), f'(x), a, b, \delta$

1 : $\text{se } f(a)f''(a) > 0$
 2 : $x_0 = a$
 3 : *senão*
 4 : $x_0 = b$
 5 : *fim se*

6 : $x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$
 7 : $n = 1$
 8 : *enquanto* $|x_n - x_{n-1}| > \delta$
 9 : $n = n + 1$
 10 : $x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$
 11 : *fim enquanto*

Saída : x_n

NO MATLAB

```
function[newton]=newton(a,b,delta)
if (f(a)*ddf(a))>0    %ddf() é a derivada segunda de f(x)
    c=a;
    x(1)=b;
else
    c=b;
    x(1)=a;
end
x(2)=x(1)-f(x(1))/ddf(x(1));
n=2;
while abs(x(n)-x(n-1))>delta
    n=n+1;
    x(n)=x(n-1)-f(x(n-1))/ddf(x(n-1));
end
newton=x(n);
```

Nota: A função $ddf()$ deve ser criada como a função $f()$.

Algoritmo 7.2.5. *Método da Iteração Linear.*

Entrada : $\phi(x), x_0, \delta$

```

1 :       $x_1 = \phi(x_0)$ 
2 :       $n = 1$ 
3 :      enquanto  $|x_n - x_{n-1}| > \delta$ 
4 :           $n = n + 1$ 
5 :           $x_n = \phi(x_{n-1})$ 
6 :      fim enquanto

```

Saída : x_n

No Matlab podemos encontrar os zeros, utilizando a função pré-definida $fzero('funcao', [a \ b])$. Por exemplo:

Desejamos encontrar o zero de $f(x) = \sin(x)$ no intervalo $[1 \ 4]$

```

>> x = fzero('sin(x)', [1 4])
x = 3.1416

```

No caso polinomial, como exemplo, considere $f(x) = x^2 - 3x + 2$.

```

>> p = [1 -3 2] vetor que define o polinômio.

```

$r = roots(p)$ calcula as raízes(zeros) do polinômio.

```

r =
     2
     1

```

7.3 Interpolação

Algoritmo 7.3.1. *Avaliação do polinômio interpolador de Lagrange.*

Entrada : $x, X = [x_0, x_1, \dots, x_n], f(X) = [f(x_0), f(x_1), \dots, f(x_n)]$

```

1 :      soma = 0
2 :      para i = 0 : n
3 :          P = 1
4 :          para j = 0 : n
5 :              se j ≠ i
6 :                  P = P * (x - x_j) / (x_i - x_j)
7 :              fim se
8 :          fim para
9 :          soma = soma + f(x_i) * P
10 :      fim para

```

Saída : soma

No Matlab podemos utilizar a função *interp1*. Veja exemplo:

```
>> X = [1 2 3] %vetor de pontos X.
```

```
>> fX = [1 4 9] %vetor imagem f(X)
```

```
>> interp1(X, fX, 2.5) %calcula o polinômio interpolador e
avalia em 2.5
ans = 6.5
```

```
>> p = polyfit(X, fX, 2) %exibe os coeficientes do polinômio
interpolador de grau 2.
```

```
p =
    1.0000    0.0000    0.0000
```

Logo nosso polinômio interpolador será $p(x) = 1x^2 + 0x + 0 = x^2$.

7.4 Integração

Algoritmo 7.4.1. *Integração por trapézios.*

Entrada : x_0, h

```

1 :      soma = 0
2 :      para  $i = 1 : n - 1$ 
3 :           $x_i = x_{i-1} + h$ 
4 :           $soma = soma + f(x_i)$ 
5 :      fim para
6 :       $I_{Tr} = \frac{h}{2}[f(x_0) + 2soma + f(x_n)]$ 

```

Saída : I_{Tr}

No Matlab integramos com o comando *trapz()* que calcula a integral por trapézios. Por exemplo:

Desejamos calcular $\int_0^2 (x^2 + 1) dx$.

`>> x = 0 : 0.01 : 2` faz uma partição do intervalo $[0 \ 2]$ com $h = 0.01$

`>> y = x.^2 + 1` avalia a função no vetor x .

`ITR = trapz(x, y)` calcula o valor da integral.

`ITR = 4.6667`

Algoritmo 7.4.2. *Integração pela 1ª Regra de Simpson.*

Entrada : x_0, h

```

1 :      soma = 0
2 :      para i = 1 : n - 1
3 :           $x_i = x_{i-1} + h$ 
4 :          se (i par)
5 :              soma = soma + 2f( $x_i$ )
6 :          senão
7 :              soma = soma + 4f( $x_i$ )
8 :          fim se
9 :      fim para
10 :       $I_S = \frac{h}{3}[f(x_0) + soma + f(x_n)]$ 

```

Saída : I_S

No Matlab utilizamos o comando $quad('f', a, b)$, onde f deve ser a função a ser integrada no intervalo $[a \ b]$. Lembramos que nesse caso a função f deve ser escrita com arquivo $.m$. Por exemplo:

Desejamos calcular a $\int_0^2 (x^2 + 1) \, dx$.

```

Arquivo f.m
function [f] = f(x)
f = x.^2 + 1

```

```

>> quad('f', 0, 2)
ans = 4.6667

```

7.5 Otimização

Algoritmo 7.5.1. *Mínimo de uma função (de uma variável) pelo método da seção áurea.*

Entrada : $f(x), a, b, \delta$

```

1 :      enquanto  $|b - a| > \delta$ 
2 :           $x_a = b - 0.618(b - a)$ 
3 :           $x_b = a + 0.618(b - a)$ 
4 :          se  $f(x_a) < f(x_b)$ 
5 :               $b = x_b$ 
6 :               $x_b = a + 0.618(b - a)$ 
7 :          senão
8 :               $a = x_a$ 
8 :               $x_a = b - 0.618(b - a)$ 
8 :          fim se
9 :      fim enquanto
10 :       $I = \frac{x_a + x_b}{2}$ 

```

Saída : I

NO MATLAB

```

function [aurea]=aurea(a,b,delta)
while abs(b-a)>delta
    xa=b-0.618*(b-a);
    xb=a+0.618*(b-a);
    if f(xa)<f(xb)
        b=xb;
        xb=a+0.618*(b-a);
    else
        a=xa;
        xa=b-0.618*(b-a);
    end
end
I=(xa+xb)/2;
aurea=I;

```

Considerando $f(x) = \sin(x)$, vamos encontrar o zero de $f(x)$ em $[0 \ 2\pi]$ com $\delta = 0.01$, aplicando a função que acabamos de criar

aurea(). Em seguida vamos comparar o resultado com a função pré-definida do Matlab *fmin()*.

```
>> aurea(0,2*pi,0.01)
```

```
ans =
```

```
4.7128
```

```
>> fmin('sin(x)',0,2*pi)
```

```
ans =
```

```
4.7124
```

Experimente $\delta = 0.0001$.

Algoritmo 7.5.2. *Mínimo de uma função (duas variáveis) pelo método do gradiente.*

Entrada : $f(x), x_0, \delta$

```
1 :       $k = 0$ 
2 :      enquanto  $\|x_k - x_{k-1}\| > \delta$ 
3 :           $g_k = \nabla f(x_k)$ 
4 :           $d_k = -g_k$ 
5 :           $\alpha_k = \text{mínimo por seção áurea de } g_k(t) = f(x_k + td_k)$ 
6 :           $x_{k+1} = x_k + \alpha_k d_k$ 
7 :           $k = k + 1$ 
8 :      fim enquanto
```

Saída : x_k

NO MATLAB

```
function [Mgrad]=Mgrad(x,delta)
k=2;
x=x'; %transforma x em um vetor coluna
x(:,k)=1;
cont=1;
while norm(x(:,k)-x(:,k-1))>delta
    if cont ==1
        k=k+1;
```



```

end
gk=gradiente(x(:,k-1)',0.0001);
dk=-gk;
ak=aurea(0,1,x(:,k-1)',dk',0.0001);
x(:,k)=x(:,k-1)+ak*dk;
cont=cont+1;
end
Mgrad=x(:,k)';

```

A função `Mgrad()` faz uso das funções `gradiente()` e `aurea()`. Vamos conhecer essas funções;

`gradiente()`: Essa função calcula o gradiente da função f no ponto x . O δ serve como grau de precisão. Lembrando que, a função $f(x)$ deve ser diferenciável.

Entrada: x, δ
 Saída: $\nabla f(x)$

```

function[gradiente]=gradiente(x,delta)
[L,C]=size(x);
for i=1:C
    k=x;
    k(i)=x(i)+delta;
    s(i)=(f(k)-f(x))/delta;
end
gradiente=s';

```

`aurea()`: Fizemos uma pequena modificação(veja abaixo) no Algoritmo ?? para adaptar ao método gradiente. Essa modificação inclui na entrada do mesmo o ponto xk e a direção dk . Lembrando que, estamos interessados em encontrar o mínimo da função $gk(t) = f(xk + tdk)$, e nesse caso, t varia em $[a \ b]$. Dessa forma podemos definir manualmente a, b de tal forma que $xk + tdk$ não saia da caixa onde queremos o mínimo da função.

```

function [aurea]=aurea(a,b,xk,dk,delta)
while abs(b-a)>delta

```

```

    xa=b-0.618*(b-a);
    xb=a+0.618*(b-a);
    if f(xk+xa*dk)<f(xk+xb*dk)
        b=xb;
        xb=a+0.618*(b-a);
    else
        a=xa;
        xa=b-0.618*(b-a);
    end
end
I=(xa+xb)/2;
aureo=I;

```

Algoritmo 7.5.3. *Mínimo de uma função (duas variáveis) por direção aleatória.*

Entrada : $f(x), x_0, \delta$

```

1 :      k = 0
2 :      enquanto  $\|x_k - x_{k-1}\| > \delta$ 
4 :           $d_k = \text{rand}(n, 1)$ 
5 :           $\alpha_k = \text{mínimo por seção áurea de } g_k(t) = f(x_k + td_k)$ 
6 :           $x_{k+1} = x_k + \alpha_k d_k$ 
7 :          k = k + 1
8 :      fim enquanto

```

Saída : x_k

O algoritmo em Matlab para direções aleatórias é análogo ao método do gradiente. Apenas modificamos a direção dk com a função pré-definida do Matlab $\text{rand}(2, 1)$ e aumentamos o intervalo da seção áurea. Veja:

NO MATLAB

```

function [aleatorio]=aleatorio(x,delta)
k=2;
x=x';

```

```

x(:,k)=1;
cont=1;
while norm(x(:,k)-x(:,k-1))>delta
    if cont ==1
        k=k+1;
    end
    dk=rand(2,1);
    ak=aurea(-1,1,x(:,k-1)',dk',0.0001);
    x(:,k)=x(:,k-1)+ak*dk;
    cont=cont+1;
end
aleatorio=x(:,k)'
```

Exemplo de aplicação: Considere a função $f(x_1, x_2) = (x_1 + 1)^2 + (x_2 + 2)^2$. Vamos aplicar as funções *Mgrad()* e *aleatorio()* com o ponto inicial $(4, 5)$ e $\delta = 0.001$. O valor k representa a número de iterações em cada função:

```

>> Mgrad([4 5],0.001)
k = 3
Mgrad =
    -1.0000    -2.0000

>> aleatorio([4 5],0.001)
k = 60
aleatorio =
    -1.0027    -1.9969
```

Veja que o método do gradiente resolve com apenas 3 iterações, enquanto o método aleatório em 60. Assim fica claro que, para funções diferenciáveis o método do gradiente é muito melhor. Mas nunca é demais lembrar que, o mesmo só pode ser aplicado em funções diferenciáveis e com ponto inicial na bacia de atração do mínimo.

Também podemos utilizar as funções pré-definidas do Matlab. Assim para encontrar mínimo de uma função de \mathbb{R}^n em \mathbb{R} , utilizamos a função *fmins('f',x0)*. Onde f deve ser uma função que recebe um vetor em \mathbb{R}^n e $x0$ deve ser o valor inicial. Veja:

Desejamos encontrar o mínimo da função $f : \mathbb{R}^2 \rightarrow \mathbb{R}$ dada por $f(x_1, x_2) = (x_1 + 1)^2 + (x_2 + 2)^2$. Tomamos como ponto inicial $x_0 = (4, 5)$.

```
>> x = fmins(' (x(1) + 1) ^ 2 + (x(2) + 2) ^ 2', [4 5]
x = -1.0000    -2.00000
```

7.6 Exercícios

7.6.1. *Teste os algoritmos nos exercícios e exemplos, dos capítulos anteriores.*

7.6.2. *Implemente em Matlab o método da Iteração Linear.*

7.6.3. *Utilizando o método de localização de zeros - MLZ associado com o método da Iteração Linear encontre os 3 zeros reais de $f(x) = x^5 - 4x^3 + x - 1$ no intervalo $[-3 \ 3]$. Faça o mesmo para o método do meio intervalo MMI, Secante e Newton.*

7.6.4. *Implemente em Matlab o método de interpolação de lagrange, cujo algoritmo foi dado. Faça uma função que retorne aos coeficientes do polinômio interpolador, compare sua função com a função pré-definida do Matlab. Dica: use sistemas lineares.*

7.6.5. *Implemente em Matlab os algoritmos de Integração. Compare os resultados com as funções pré-definidas do Matlab.*

7.6.6. *Na função `Mgrad()` utilizamos a função `aurea()` com o intervalo fixo em $[0 \ 1]$. Explique por que na função `aleatorio()` utilizamos a função `aurea()` com o intervalo $[-1 \ 1]$.*

Capítulo 8

Respostas dos exercícios

A solução do exercício é apresentada de forma compacta, devendo o leitor efetuar os cálculos. O leitor também deve estar ciente que varias respostas são aproximações, por exemplo, suponhamos que o leitor tenha encontrado como resposta o valor 1,77 e a resposta do exercício esteja com o valor 1,7659, isso não quer dizer que o leitor errou seus cálculos, ocorre muitas vezes um erro de truncamento/arredondamento quando utilizamos calculadoras manuais. Lembramos que o principal objetivo são os métodos numéricos. Erros de truncamento/arredondamento são comentados no apêndice.

Capítulo 1

Exercício 1.5.2 Veja que cada equação linear é uma reta, desenhando cada reta podemos observar que todas as 3 são paralelas, logo não possuem um ponto em comum. Assim o sistema não possui solução.

Exercício 1.5.4

a) $x_1 = 1, x_2 = -1, x_3 = 2, x_4 = -2$

b) $x_1 = 0, x_2 = 0, x_3 = 0$

1.5.5 Basta mostrar que a matriz ampliada do primeiro sistema é linha equivalente a do segundo sistema. Use $L_2 = L_2 - 2L_1$ e $L_3 = L_3 - L_1$.

Exercício 1.5.7 $L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 4 & 1 \end{bmatrix}$ $U = \begin{bmatrix} 1 & 2 & -1 \\ 0 & -1 & 0 \\ 0 & 0 & 2 \end{bmatrix}$

Faça $Ly = B_1$ e $Ux = y$ para encontrar a primeira coluna da matriz A^{-1} . Faça novamente $Ly = B_2$ e $Ux = y$ para encontrar a segunda coluna da matriz A^{-1} . Por fim faça $Ly = B_3$ e $Ux = y$ para encontrar a terceira coluna de A^{-1} . Onde $B_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $B_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$ e $B_3 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

Exercício 1.5.8

a) $X^0 \cong \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, $X^1 \cong \begin{bmatrix} 1.2 \\ 1.75 \\ 1 \end{bmatrix}$, $X^2 \cong \begin{bmatrix} 0.925 \\ 0.9 \\ 1.1833 \end{bmatrix}$
e $X^3 \cong \begin{bmatrix} 0.9917 \\ 0.9917 \\ 0.9917 \end{bmatrix}$

b) $X^0 \cong \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $X^1 \cong \begin{bmatrix} 1 \\ 0.3333 \end{bmatrix}$ e $X^2 \cong \begin{bmatrix} 0.8333 \\ 0 \end{bmatrix}$

Exercício 1.5.9

a) $X^0 \cong \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$, $X^1 \cong \begin{bmatrix} 1.2 \\ 1.15 \\ 0.9833 \end{bmatrix}$ e $X^2 \cong \begin{bmatrix} 0.9867 \\ 1.0108 \\ 1.0081 \end{bmatrix}$

b) $X^0 \cong \begin{bmatrix} 0 \\ 0 \end{bmatrix}$, $X^1 \cong \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ e $X^2 \cong \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

Exercício 1.5.13 b) Use $A^{-1} = \begin{bmatrix} 1 & -2 \\ -1 & 3 \end{bmatrix}$

Exercício 1.5.14 Use $\|AB\| \leq \|A\| \cdot \|B\|$ e $(AB)^{-1} = B^{-1}A^{-1}$.

Exercício 1.5.15 Observe que a matriz $\text{adj}(A)$ é formada pela transposta da matriz dos cofatores Δ e cada cofator

$$\Delta_{ij} = (-1)^{i+j} \det M_{ij} = \begin{cases} \prod_{k=1, k \neq i}^n a_{kk} & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases},$$

ou seja, a $\text{adj}(A)$ será uma matriz diagonal com cofatores Δ_{ii} . Dessa forma segue o resultado.

Exercício 1.5.16 Seja i_0 tal que $\|\text{adj}(A)\| = \prod_{j=1, j \neq i_0}^n |a_{jj}|$. Veja que para $k \neq i_0$ com $k = 1, \dots, n$ temos:

$$\prod_{j=1, j \neq k}^n |a_{jj}| \leq \prod_{j=1, j \neq i_0}^n |a_{jj}|$$

que implica em

$$|a_{i_0 i_0}| \leq |a_{kk}|.$$

portanto $|a_{i_0 i_0}| = \min_{1 \leq j \leq n} |a_{jj}|$.

Como $A^{-1} = \frac{1}{\det(A)} \text{adj}(A)$ temos:

$$\begin{aligned} \|A^{-1}\| &= \frac{1}{\prod_{j=1}^n |a_{jj}|} \|\text{adj}(A)\| = \frac{1}{\prod_{j=1}^n |a_{jj}|} \max_{1 \leq i \leq n} \left\{ \prod_{j=1, j \neq i}^n |a_{jj}| \right\} \\ &\quad \underbrace{\hspace{10em}}_{\text{para } i_0} \\ \|A^{-1}\| &= \frac{1}{\prod_{j=1}^n |a_{jj}|} \prod_{j=1, j \neq i_0}^n |a_{jj}| = \frac{1}{|a_{i_0 i_0}|} \end{aligned}$$

Dessa forma segue o resultado.

Exercício 1.5.17 $x_1 = -2 + 4i$ e $x_2 = -2 - 3i$

$$\textbf{Exercício 1.5.18} \quad \begin{cases} x_1 + 2x_2 + 3x_3 &= 11 \\ 3x_1 + 3x_2 + 2x_3 &= 13 \\ 4x_1 + 5x_2 + 3x_3 &= 20 \end{cases}$$

Onde x_1, x_2, x_3 representam a quantidade em unidades de pera, uva e maçã. Resolvendo o sistema teremos $x_1 = 1, x_2 = 2, x_3 = 2$. Assim deve-se ingerir 1 pera, 2 uvas e 2 maçãs por dia.

Exercício 1.5.19 Observe que o sistema real é montado a partir de $S = \begin{bmatrix} M & -N \\ N & M \end{bmatrix}$ onde M, N são matrizes $n \times n$. Assim S , que é uma matriz real, tem dimensão $2n \times 2n$.

Exercício 1.5.20 Observe que $f_{ij} = -\frac{a_{ij}}{a_{ii}}$. Substitua no teorema e conclua o resultado.

$$\textbf{Exercício 1.5.21} \quad \begin{cases} 2x_1 + x_2 + x_3 &= 4 \\ x_1 + x_2 - x_3 &= 3 \\ -2x_1 + 2x_2 + x_3 &= 2 \end{cases}$$

Este é apenas um sistema que possui solução $x_1 = 1, x_2 = 2, x_3 = 0$. Mas existem infinitos.

Capítulo 2

Exercício 2.9.1 Aplicando o MLZ com $\gamma = \frac{1}{4}$ temos que, apenas os intervalos $[-\frac{7}{4} \quad -\frac{3}{2}]$ e $[\frac{7}{4} \quad 2]$ possuem zeros.

Obs: Esta função só possui 2 zeros reais.

Exercício 2.9.2 $\varepsilon_1 \cong -1.6975$ e $\varepsilon_2 \cong 1.8556$.

Exercício 2.9.3

a) $\varepsilon \cong -1.3247$

b) $\varepsilon \cong 0.9210$

c) $\varepsilon \cong 1.6324$

Exercício 2.9.7 $\varepsilon_1 \cong 0.5671$ e $\varepsilon_2 \cong 1.3308$.

Exercício 2.9.8 $\varepsilon \cong 0.2875$.

Exercício 2.9.10 $\varepsilon \cong 0.5867$.

Exercício 2.9.11 $Max \cong -1.5704$ e $Min \cong -0.05$.

Exercício 2.9.12 $MinGlobal \cong 1.0861$.

Exercício 2.9.16 $\varepsilon_1 = 3, \varepsilon_2 = -2, \varepsilon_3 = 1$.

Capítulo 3

Exercício 3.7.4 Use a tabela invertida:

i	0	1	2	3
$f(x_i)$	1.75	1.80	1.85	1.90
x_i	0.984	0.9738	0.9613	0.9463

para calcular $p(x)$ e calcule $p(0.95)$.

Exercício 3.7.5 Interpole a função $f(x) = x^2$ em $(0, 0)$ e $(1, 1)$ e veja que $p(x) = x^2$. Isto ocorre porque o polinômio interpolador é único.

Exercício 3.7.6 Use os teoremas do capítulo 2 para encontrar os intervalos onde f possui zeros. Use interpolação inversa em pelo menos dois pontos de forma que o zero de f esteja entre esses pontos.

Exercício 3.7.8 Primeiro mostre que $f[x_i, x_j] = f[x_j, x_i]$ depois use isso para demonstrar o exercício.

Exercício 4.5.7 Adições: $2n^2 + 3n$, Multiplicações: $2n^2 + 2n$

Exercício 3.7.11 Faça $f'(x) = 0$ e encontre os zeros de f' .

Capítulo 4

Exercício 4.5.3 Tomando $h = 1/2$ temos: Integral por trapézios $\cong 0.0725$ com erro $\cong -0.0052$.

Exercício 4.5.4 Trapézios $\cong 3.0806$ e Simpson $\cong 4.1075$ erro $\cong 4.7395$.

Exercício 4.5.6 Calcule a soma:

$$\frac{h}{3} \sum_{i=0}^{n-3} [f(x_i) + 4f(x_{i+1}) + f(x_{i+2})]$$

Exercício 4.5.7 Use o mesmo procedimento adotado para a 1ª regra de Simpson.

Apêndice A

Erros

Quando consideramos métodos computacionais para resolver problemas envolvendo aritmética, estamos sujeitos aos possíveis erros. Para compreender esses erros, devemos entender o tipo de sistema numérico usado em computadores.

Quando colocamos dados em um computador ele utiliza um sistema numérico próprio finito. Essa transformação normalmente envolve erros de arredondamento. Também podem ocorrer erros de arredondamento nos processos algébricos contidos nos algoritmos. Dessa forma, não podemos esperar obter a solução exata do problema original. O que podemos esperar é uma boa aproximação para uma pequena perturbação do problema original. Por exemplo, resolver um sistema $Ax = b$. Quando inserimos a matriz A e o vetor b cometemos erros. Por isso estaremos tentando resolver o sistema $(A + E)x = \hat{b}$, onde E é o erro de arredondamento.

Geralmente os algoritmos numéricos realizam um quantidade muito grande de operações aritméticas, por isso os erros podem crescer como um câncer e as soluções obtidas podem estar completamente erradas. Nessa linha vamos estudar a estabilidade dos métodos numéricos em relação a propagação de erros.

A.1 Números em ponto flutuante

Para entender o tipo de números usado pelo computador definimos o número em ponto flutuante:

Definição A.1. *Um número em ponto flutuante na base b é um número da forma:*

$$\pm \left(\frac{d_1}{b} + \frac{d_2}{b^2} + \cdots + \frac{d_t}{b^t} \right) \times b^e$$

onde $t, d_1, d_2, \dots, d_t, b, e$ são todos inteiros e
 $0 \leq d_i \leq b - 1, i = 1, \dots, t$.

O inteiro t é o número de dígitos e depende da capacidade do computador, assim como o expoente e que fica entre $L \leq e \leq U$, onde L, U também depende da capacidade do computador. A maioria dos computadores usa a base 2.

Exemplo A.1.1. *Os números a seguir são números decimais (base 10) em ponto flutuante com cinco dígitos:*

$$0.53216 \times 10^{-4}$$

$$-0.81724 \times 10^{21}$$

$$0.00112 \times 10^8$$

$$0.11200 \times 10^6$$

Note que os números 0.00112×10^8 e 0.11200×10^6 são iguais. Assim a representação em ponto flutuante não é necessariamente única. Os números em ponto flutuante escritos sem zeros logo após a vírgula são ditos normalizados. Por exemplo:

0.53216×10^{-4} é normalizado.

0.00112×10^8 não é normalizado.

Para mostrar a limitação de um computador hipotético vamos supor que $t = 1, -1 \leq e \leq 1$ e $b = 10$. Teremos ao todo apenas 55 números representados, que são:

$$\begin{aligned} &0, \pm 0.1 \times 10^{-1}, \pm 0.2 \times 10^{-1}, \dots, \pm 0.9 \times 10^{-1} \\ &\pm 0.1 \times 10^0, \pm 0.2 \times 10^0, \dots, \pm 0.9 \times 10^0 \\ &\pm 0.1 \times 10^1, \pm 0.2 \times 10^1, \dots, \pm 0.9 \times 10^1 \end{aligned}$$

Veja que todos estes números estão no intervalo $[-9 \ 9]$. Com este sistema ficamos limitados a representar apenas estes números.

Veremos que na prática o que ocorre é exatamente isto, ou seja, quando inserimos um número no computador, o mesmo busca um número em seu sistema que seja o mais próximo possível. Para continuarmos precisamos da

Definição A.2. *Se x é um número real e x' é sua aproximação em ponto flutuante, então a diferença $x' - x$ é chamada de erro absoluto, e o quociente $\frac{x' - x}{x}$ é chamado de erro relativo.*

Exemplo A.1.2. *Consideramos um computador hipotético com o número de dígitos igual a 4 ($t = 4$) e base $b = 10$.*

O número $x = 62133$ tem representação em ponto flutuante igual a $x' = 0.6213 \times 10^5$. Observe que o erro absoluto $x' - x = -3$ e o erro relativo $\frac{x' - x}{x} = \frac{-3}{62133} \approx -4.8 \times 10^{-5}$.

Observamos com isto que o erro absoluto é exatamente o que se perde na transição para o computador hipotético, e o erro relativo, quando multiplicado por 100, nos dá o percentual do erro cometido. No exemplo acima temos um erro relativo de -4.8×10^{-5} o que multiplicado por 100 corresponde a 0.0048%.

Os erros de arredondamento não ocorrem somente quando inserimos um valor, mas também quando utilizamos as operações aritméticas veja:

Exemplo A.1.3. *Sejam $a' = 0.263 \times 10^4$ e $b' = 0.466 \times 10^1$ números decimais em ponto flutuante com três dígitos. Se esses números forem somados, a soma exata vai ser:*

$$a' + b' = 0.263446 \times 10^4$$

No entanto, a representação em ponto flutuante desse número é 0.263×10^4 . Essa deve ser a soma calculada. Vamos chamar de soma em ponto flutuante $fl(a' + b')$. Assim o erro absoluto é

$$fl(a' + b') - (a' + b') = -4.46$$

e o erro relativo

$$\frac{-4.46}{0.26344 \times 10^4} = -0.17 \times 10^{-2}$$

O erro absoluto na multiplicação é -29.8 e o erro relativo -0.25×10^{-2} o que corresponde a 25%.

Em geral, o erro relativo na aproximação de um número x por sua representação em ponto flutuante x' é denotado por δ . Então

$$\delta = \frac{x' - x}{x} \quad \text{ou} \quad x' = x(1 + \delta). \quad (\text{A.1})$$

O $|\delta|$ pode ser limitado por uma constante positiva ϵ , chamada de precisão da máquina ou épsilon da máquina. O épsilon da máquina é definido como o menor número em ponto flutuante tal que

$$fl(1 + \epsilon) > 1.$$

Por exemplo, se o computador usa números decimais com três dígitos, então

$$fl(1 + 0.499 \times 10^{-2}) = 1$$

enquanto

$$fl(1 + 0.500 \times 10^{-2}) = 1.01$$

Nesse caso, o épsilon da máquina seria 0.500×10^{-2} . Podemos fazer o mesmo para a subtração, multiplicação e divisão. Assim da Equação A.1 temos:

$$fl(a' + b') = (a' + b')(1 + \delta_1)$$

$$fl(a'b') = (a'b')(1 + \delta_2)$$

$$fl(a' - b') = (a' - b')(1 + \delta_3)$$

$$fl\left(\frac{a'}{b'}\right) = \left(\frac{a'}{b'}\right)(1 + \delta_4)$$

Onde todos os δ_i são erros relativos e todos vão ter valor absoluto menor do que ϵ . Note que no exemplo anterior $\delta_1 \approx -0.17 \times 10^{-2}$, $\delta_2 \approx -0.25 \times 10^{-2}$ e $\epsilon = 0.5 \times 10^{-2}$.

O algoritmo abaixo pode ser utilizado para saber com quantos dígitos a máquina trabalha:

```
 $\alpha = 1$   
 $j = 0$   
 $sair = f$   
  enquanto  $sair = f$  faça  
     $\alpha = \frac{\alpha}{2}$   
    se  $1 + \alpha \leq 1$  então  
       $sair = v$   
    fim se  
     $j = j + 1$   
  fim enquanto  
A precisão da máquina será dada por  $j$ .
```


Índice Remissivo

- Áurea, método, 92
- Algoritmo, definição, 102
- Ampliada, matriz, 2
- Atratoras, bacias, 96
- Bacias de atração, 96
- Bolzano, teorema de, 34
- Briot-Ruffini, método de, 69
- Cauchy, 42
- Cauchy, seqüência de, 31
- Comentários finais Zeros de função, 45
- Complexos, sistemas lineares, 22
- Condicional, número, 17
- Contínua, função, 29
- Convergência no método da secante, 37
- Convergência no método de Newton, 39
- Convergência, método de Jacobi e Gauss-Seidel, 21
- Critério de linhas, 21
- Critério de parada, 14
- DDF, 63
- DDF, fórmula geral interpolação, 64
- Decomposição LU, 7
- Direção aleatória, 98
- Direto Método, 4
- Doolittle método, 10
- Elementares, operações, 4
- Eliminação Gaussiana, 8
- Erro, 17
- Erro de truncamento, 65
- Função contínua, 29
- Função de Iteração, 43
- Função de iteração, 11
- Função, zeros de uma, 29
- Funções pré-definidas, 110
- Gauss método, 4
- Gauss, método iterativo de, 15
- Gaussiana, eliminação, 8
- Geométrica, solução, 2
- Global, máximo, 89
- Global, mínimo, 89
- Gradiente de uma função, 94
- Grout método, 10
- Infinitas soluções, 3
- Inflexão, ponto, 91
- Interpolação, 59
- Interpolação de Lagrange, 60
- Interpolação por DDF, 63
- Inversão de matriz, 9

- Inversão, matriz, 9
- Iteração linear, método da, 40
- Iteração, função de, 11
- Iterativo, método do refinamento, 17
- Iterativos métodos, 11
- Jacobi, método, 11
- Linear, Sistema, 1
- Local, máximo, 89
- Local, mínimo, 89
- Localização de zeros, 45
- Localização de zeros, método de, 31
- LU decomposição, 7
- Máximo local, 89
- Método Briot-Ruffini, 69
- Método da iteração linear, 40, 46
- Método da Seção Áurea, 92
- Método da secante, 35, 45
- Método de Doolittle, 10
- Método de Gauss, 4
- Método de Grout, 10
- Método de Localização de Zeros, 31
- Método de Newton, 37, 45
- Método do meio intervalo, 33, 45
- Método do Refinamento Iterativo, 17
- Método iterativo de Jacobi, 11
- Método iterativo Gauss-Seidel, 15
- Métodos Diretos, 4
- Métodos Iterativos, 11
- Mínimo global, 89
- Mínimo local, 89
- Matriz ampliada, 2, 4, 5
- Matriz dos coeficientes, 2
- Matriz quadrada, 2
- Matriz triangular inferior, 7
- Matriz triangular superior, 7
- Meio intervalo, método do, 33
- Multiplicidade de um zero, 54
- Número condicional, 17
- Newton, convergência no método de, 39
- Newton, método, 37
- Operações elementares, 4
- Otimização, 89
- Parada, Critério de, 14
- Partição, 32
- Pivo, 7
- Ponto de inflexão, 91
- Ponto fixo, 40
- Quadrada, matriz, 2
- Refinamento iterativo, método do, 17
- Script, 111
- Seção áurea, 92
- Secante, convergência no método da, 37
- Secante, método da, 35
- Seidel, método iterativo de, 15
- Seqüência de Cauchy, 31
- Seqüência monótona, 34
- Sistemas Complexos, 22
- Sistemas Lineares, 1
- Solução de um sistema $n \times n$, 3
- Solução Geométrica de um sistema, 2

Substituição Retroativa, 9

Substituição retroativa, 6

Teorema de Bolzano, 34

Teorema de Rolle, 66

Teorema de Weirstrass, 59

Teorema do ponto de inflexão, 91

Teorema Interpolação de Lagrange,
60

triangular matriz, 7

Vetor incógnitas, 2

Vetor termos independentes, 2

Zeros de função, 29

Zeros de um polinômio, 47

Referências Bibliográficas

- [1] Leônidas C. Barroso. *Cálculo Numérico com aplicações*. Harbra, 1987.
- [2] José Luiz Boldrini. *Álgebra Linear*. Ed. Harbra, 1980.
- [3] Djairo Guedes de Figueiredo. *Análise I*. Ed. LTC, 1996.
- [4] Howard. Eves. *Introdução à História da Matemática*. Ed. Polígono, 1989.
- [5] I.N. Hernstein. *Tópicos de Álgebra*. Ed. Polígono, 1970.
- [6] Elon Lajes Lima. *Curso de Análise, vol.1*. Ed. SBM, 1993.
- [7] Elon Lajes Lima. *Curso de Análise, vol.2*. Ed. SBM, 1999.
- [8] Elon Lajes Lima. *Álgebra Linear*. Ed. SBM, 1980.
- [9] Walter Rudin. *Princípios de Análise Matemática*. Ed. Ao Livro Técnico, 1971.
- [10] Francis Scheid. *Análise Numérica*. McGraw-Hill, 1991.
- [11] Dercio Sperandio. *Cálculo Numérico: características matemáticas e computacionais dos métodos numéricos*. Ed. Prentice Hall, 2003.