vetor próprio, 95
vetores, decomposicao, 83
vetorial, espaço, 79
vizinhança tubular, 178
volante, media, 162

Weirstrass, aproximacao, 24
wirth,N., v

sistema sub- determinado, 82
sistema.pas, programa, 105, 106
sistemas equivalentes, 90, 91, 96
sistemas indeterminados, 82
sistemas lineares, 76, 95, 106
sol. de equ. lineares, 96, 108
sol. de equac. lineares, 77
sol. equ. linear, 103
solução de eq. dif., 126
solução de equ. difer., 119
som e série de Fourier, 47
somas de riemann, 2
spliexp, 20
spline, 141, 153
spline- p.u., 152
spline-espaco, 153
splineK, 20
Splines, 124
splines, 9, 13, 33, 53, 151
splines por convolucao, 156
splines, espaço de, 154
splines, núcleo, 173
sub-determinado, sistema, 82
superior, triangularização, 96, 101
suporte compacto, 142
suporte compacto, função, 142
suporte de função, 142

Taylor, polinômio, 5
taylor, polinomio, 5
teor. das distrib., 170
teorema da função implícita, 118, 119
teorema de Green, 74
teorema de peano, 138
teorema do ponto fixo, 136
TeX, xi
TexTelmexTel, xi
traço, 74
transformacao linear, 71, 72
translação, 147
transposição, 77
transposta, 77
trapesio, regra, 2, 4
triang. fatoracao, 108, 109
triangular, matriz, 96, 99, 101, 102, 108
triangularização de matrizes, 95, 96
triangularização inferior, 96, 108
triangularização superior, 96, 101, 102
trigonom. polin., 41
trigonom. séries, 41
trigonométrico, pol., 43, 44
trigonométrico, polinômio, 55
tubular, vizinhança, 178
Tutorial, 43

unidade aproximada, 170
unidade, partição, 142, 147
unidade, partição, 148

valor médio
  $\phi$, 162
  viciado, 162
valor próprio, 95
vetor, 57, 79

pivot, 109, 110
Polinômio de Taylor, 63
polinômio de Taylor, 5
polinômio trigon., 41
polinômio trigonométrico, 55, 56
polinômios trigon., 43, 44
polinom. interpolação, 141
polinomio de taylor, 5
ponto fixo, teorema, 136
próprio, valor, 95
próprio, vetor, 95
precisão, conjunto, 10, 25, 28
problema, 120
problema, e.d.o., 119
proces. paralelo, 107
processamento paralelo, 107
processo de converg., 153
produto escalar, 42, 57
produto, determinante, 103
programa FatorT, 115
programa fourier, 43
programa fourier.pas, 45
programa sistema.pas, 105, 106
programa, otimizacao, 4
projetor e função, 151
projetores, 151
projetores e part. da unid., 148

q-spline ← dados discr., 15, 20
quasi-spline, matriz, 18
quasi-splines, 9, 13, 14

recursiva, linguagem, 105
recursivo, programa, 105
redundância, 80
refinamento de partição, 52
regra de Simpson, 5
regra de simpson, 2, 5
regra do trapézio, 10
regra do trapesio, 2, 10
regul. por convol., 172
regularição por convolução, 189, 190
regularização por convolução, 124
regularizacao p/ convol., 156
Runge-Kutta, 123
Runge-Kutta, método, 128, 130

Série de Fourier e som, 47
séries de fourier, 45
séries trigonom., 41
series de fourier, 46
silva, sebastiao, 170
simpson, 6
simpson, regra, 2, 5
Simpson, regra de, 5
sinal, 143, 157, 164, 181
sinal diferenciável, 144
sinal e convol., 159
sinal parabólico, 145
sinal triangular, 143, 182
singular, parte, 111
sistema de equações , 78
sistema homogêneo, 82
sistema ortogonal, 85
sistema ortonormal, 85
sistema redundante, 85

linguagem recursiva, 105
linguagens modulares, v
Lipschiz, constante, 27, 40
lisp, v
little theorem of Picard, 139

método das isóclinas, 134
Méde Euler, 180
método de Euler, 125
método de Gauss, 98
método de Runge-Kutta, 128, 130
matriz, 64
matriz triangular, 99, 101, 102, 108
matriz de custos, 61
matriz de dados, 78
matriz de um quasi-spline, 18
matriz de variação dos custos, 61
matriz diagonal, 75
matriz escalar, 74, 75
matriz jacobina, 59
matriz singular, 105
matriz triangular, 96
matriz, anel, 110
matrizes esparsas, 96, 97
matrizes, algebra, 66
matrizes, fatoração, 109, 112
media volante, 162
medida de dirac, 170
metodo da elimin., 97
metodo de Gauss, 96
metodo de gauss-jordan, 108
metodo iterativo p/ edo, 135
modulares, linguagens, v
multi-número, 64

núcleo, 173
nível, curva, 69
núcleo, 157
núcleo spline, 173
núcleo, operador integral, 157
número generalizado, 64
numero generalizado, 76

operador, 137
operador diferencial e convolucao, 168
operador integral, 137
ortogonal, sistema, 85
ortonormal, sistema, 85
Oscilação , derivada, 175
oscilação e erro, 51, 52
otimização -depen. linear, 85
otimizacao, 65
otimizacao de programa, 4

p.u. - spline, 152
parabólico sinal, 145
paralelo, proces., 107
paralelo, processamento, 107
parciais, e.d., 124
part. da unid. , projetor, 148
parte singular, 111
partição da unidade, 142, 147, 148
partição, refinamento, 52
passo a passo, 124
peano, teorema, 138
Picard, little theorem, 139
Picard, método de, 123

fatoração de matrizes, 109, 112
fatoracao triangular, 108, 109
fatort, programa, 115
fechado, disco, 142
formatação de dados, 20
Fourier, 43
fourier, 41
fourier, séries, 45
fourier, series, 41, 46
fourier,programa, 45
função a suporte compacto, 142
função e projetor, 151
função implícita, teorema, 118, 119
função, extremo, 70
função, suporte, 142
funcao de heaviside, 169
funcional linear, 73, 74
funcoes, espacos, 42

Gauss, método, 98
Gauss, metodo, 96
gauss-jordan, metodo, 108
gerar um espaço, 80
gradiente, 69
grafunXX, 11
grau de liberdade, 36, 38
Green, teorema de, 74

heaviside, funcao, 169
hipótese de alta diferenciabiliade, 23
homogêneas, equações, 79
homogêneo, sistema, 82
homotetia, 147

indepen. linear, 83
independência linear, 62
indeterminados, sistemas, 82
inferior, triangularização, 96
informação , estrutura, vi
injecao de energia, 164
integral estocástica, 8
integral, operador, 137
intensidade de calor, 142
internet, endereço, xi
interpolação polinômial, 49, 50, 54
interpolação polinom., 141
interpolacao polinom., 13
intervalo compacto, 143
intestos, 8
isóclinas, método, 134
Isóclinas, método das, 123

jacobiana, 69
jacobiana de $f$, 59

k-quasi-spline, 13
k-spline, 33

laboratório musical, 47
LaTex, xi
liberdade, grau, 36, 38
linear transformacao, 71, 72
linear, depência, 80
linear, dependência, 85
linear, dependencia, 80
linear, funcional, 73, 74
linear, indepen., 83
lineares, sistemas, 95

189, 190
convolucao e operador diferencial, 168
crítica do Cálculo, 15
ctan, xi
curva de nível, 69
custos, matriz, 61
custos, variação, 61

dados amostrais, 20, 32
dados discr. → q-spline, 15, 20
dados, aquisição, 16, 18
dados, formatação, 20
Debian Foundation, xi
decomposicao de vetores, 83
depência linear, 80
depen. linear. e otimizacao, 85
dependência linear, 62, 80, 85, 86
dependencia linear, 65, 80
derivada da convol., 167
Derivada, oscilação , 175
determinante do produto, 103
determinantes, 105
diferencial, 63
diferencial total, 59
diferencial, aprox., 64
dimensão, 86
dirac, medida, 170
disco fechado, 142, 143
discretisação, 175
distribuicoes, teor., 170
divergente, 74
divisor de zero, 77, 78
Donald Knuth, xi

e.d.o.- problema, 119
edo, 117
edo, metodo iterativo, 135
edp, 118, 124
eliminacao, metodo, 97
emtex, xi
energia, injecao, 164
equ. dif. ordinaria, 117
equ. dif. parcial, 118
equ. linear, sol., 103
equ. lineares, sol., 77
equ. lineares, solução, 96
equ. lineares, solucao, 108
equação característica, 95
equações homogêneas, 79
equações lineares, sistema, 78
equivalencia de sistemas, 90, 91, 96
erro, 24
erro e oscilação, 51, 52
escalar, matriz, 74
escalar, produto, 42, 57
escalonamento de sistemas, 95, 96
espaço de splines, 154
espaço vetorial, 79
espaco de splines, 153
espacos de funcoes, 42
esparsas, matrizes, 96, 97
estocástica, integral, 8
estrutura da informação , vi
Euler, método, 180
Euler, método de, 123, 125
extremos de funções, 70

# Índice remissivo


$\phi$-valor médio, 162
1-spline, 33

algebra de matrizes, 66
algoritmo, complexidade, 103
alta diferenciabilidade, hipótese., 23
amostrais, dados, 20, 32
análise espectral, 46
anel de matrizes, 110
angulo, 57
aprox. diferencial, 63
aprox. e convol. , 163
aprox. polinomial, 13
aproximação diferencial, 64
aproximação por convolução, 172
aproximação linear p.p., 12
aproximacao de funcoes, 9
aproximacao linear p.p., 10
aproximacao polinomial, 9
aproximacao polinomial p.p., 9, 33
aproximacao, Weirstrass, 24
aproximada, unidade, 170
aquisição de dados, 16, 18

backup, viii
bm2font, xi

c, v
Cauchy-Riemann, 122
compactação de arquivos, 5
compacto, conjunto, 143
compacto, intervalo, 143
comparacao: rieamnn-simpson, 7
complexidade do algoritmo, 103
comutativ. da convol., 159
conjunto compacto, 143
conjunto de precisão, 10, 25, 28
constante de Lipschiz, 27, 40
contin. e convol., 165
converg. ,processo, 153
convol. , regular., 156, 172
convol. comutativa, 159
convol. e aprox., 163
convol. e contin., 165
convol. e derivada, 167
convol. e sinal, 159
convolução, 157
convolução, aproximação, 172
convolução, 170
convolução, regularização, 124,

[22] Schumaker, L. *Spline Functions* - 1980
John Willey and Sons - New York.

[23] Shapiro, H.S.
*Smoothing and approximation of functions.*
- Van Nostrand Reinhold Mathematical Studies Nr. 24 - 1970.

[24] SIAM NEWS - *The news journal of the Society of Industrial and Applied Mathematics* - Philadelpphia, PA - USA

[25] Simmons, G.F.
*Introduction to Topology and Modern Analysis.*
McGraw-Hill - Book Company - 1968

[26] Simmons, G.F.
*Differential Equations with App. and Hist. Notes.*

McGraw-Hill - Book Company - 1978

[27] Whaba, Grace *Spline Models for Observational data*
CBMS-NSF Regional Conference Series in App. Mathematics
- SIAM - 1990.

[10] Lanczos, C. *Linear Differential Operators* - Classics in Applied Mathematics - 18 - SIAM - 1996

[11] Lang, S. *Analysis II.*
- Addison-Wesley-Reading Ma

[12] Linz, P. *A critique of numerical Analysis.*
- Bull. of AMS vol 19 no. 2 1989 (407,416)

[13] Meyer, Y Wavelets *Algorithms and Applications* - SIAM - 1994

[14] Monagan e Neuenschwander
*GRADIENT. Algortithmic Differentiation in Maple.*
Pre-Print - Symbolic Computation Group
Institut für Wissenschftliches Rechnen
- ETH - Zürich - Switzerland

[15] Marcelo P. Nunes *Tutorial para o Ensino do Cálculo* - programa tutorial ftp.cpd.furg.br/pub/calculo - 1996-1997

[16] Praciano-Pereira, T. *Cálculo numérico Computacional - Introdução ao Pascal*
- Editora da Universidade Estadual do Vale do Acaraú - 2000

[17] Praciano-Pereira, T. Convolution Splines - 1995 submetido para publicação

[18] Praciano-Pereira, T e Gerônimo, J.R. *Cálculo Diferencial e Integral com apoio computacional*
- Notas mimeografadas - BCE - UEM - 1991

[19] Rossum,
Guido van *A tutorial on Python* - guido@cnri.reston.va.us - ftp.python.org.

[20] Rudin, W. *Real and Complex Variables* - Second Edition - McGraw-Hill - 1974

[21] Scilab Group - *SciLab - programa para simulações numéricas* INRIA - Unité de recherche de Rocquencourt - Projet Meta2 - scilab@inria.fr - 1996

# Bibliografia


[1] D'Azevedo, E.F. Romine, C. H. e Walker, D. W. *Shared-Memory Emulation is key to Billion-Atom Molecular Dynamics Simulation*
- SIAM-News Vol 28 (5) - 1995

[2] Beazley, D. M. e Lomdahl, P. S. *Large-Scale Molecular Dynamics on MPPs: Part I*
- SIAM-News Vol 28 (2) - 1995

[3] Borland *Reference guide to TurboPascal- 1990*
- Borland - 1992

[4] Cláudio, D. M. e Marins, J.M.
*Cálculo Númerico Computacional* - 1994
editora atlas - 2ª Edição

[5] Daubechies, Ingrid, *Ten lectures on wavelets* - SIAM - Philadelphia, PA - 1992

[6] Dieudoné *Calcul Infinitesimal* - 1968
- Collection Méthodes - Herman - Paris.

[7] Lang, Serge *Algebra* - Addison-Wesley - Reading, Mass. 1970

[8] Eaton, John W. *Octave* - A high-level interactive language for numerical computations - 1996 ftp.che.utexas.edu/pub/octave/octave-M.N.tar.gz

[9] Knuth, D.E. *The Art of Computer Programming, Vols 1,2.*
- Addison-Wesley-Reading Ma 1968

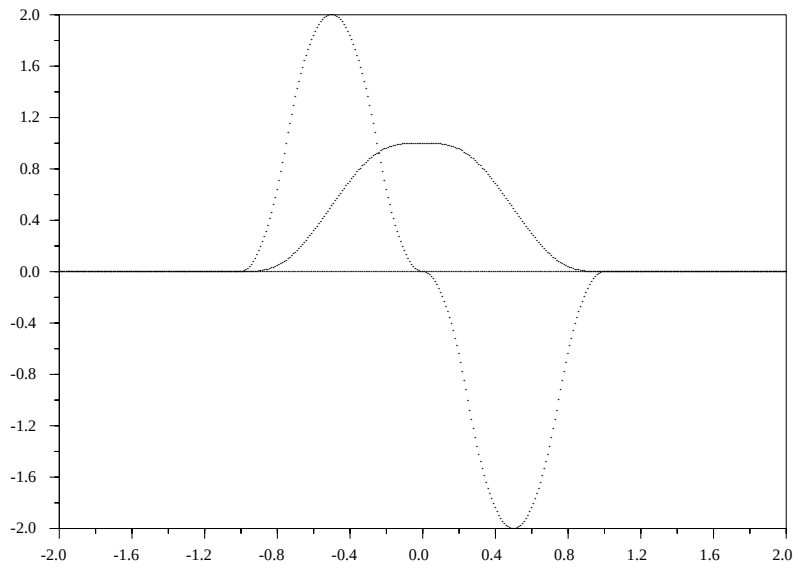



Figura 3.12: Um (n+1)-spline kernel construido indutivamente de um n-spline kernel..

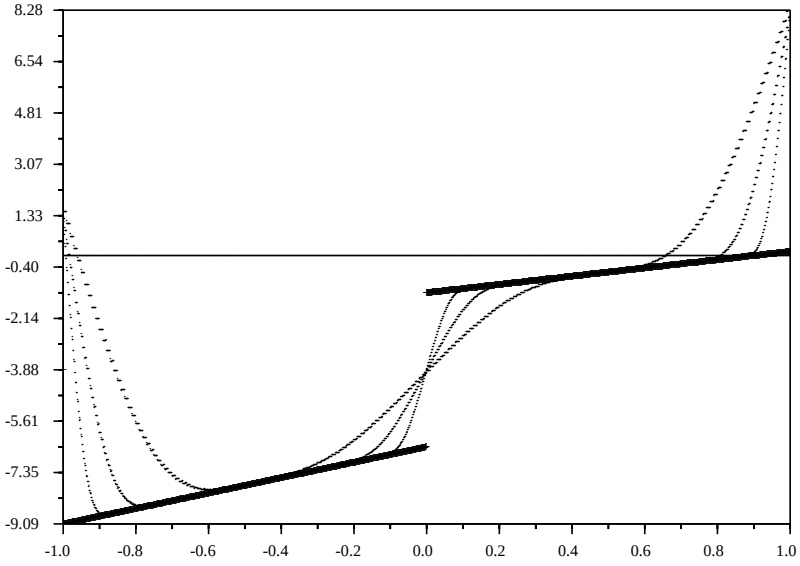



Figura 3.11: Sucessão de regularizações de uma função linear por pedações. Uso de uma sucessão de núcleos com suportes diminuindo de medida.

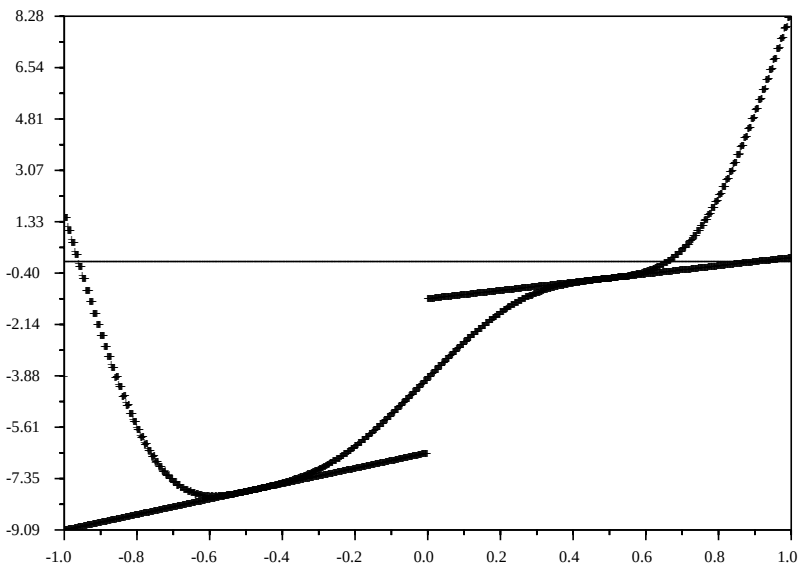



Figura 3.10: Uma função linear por pedações e sua regularização por convolução.

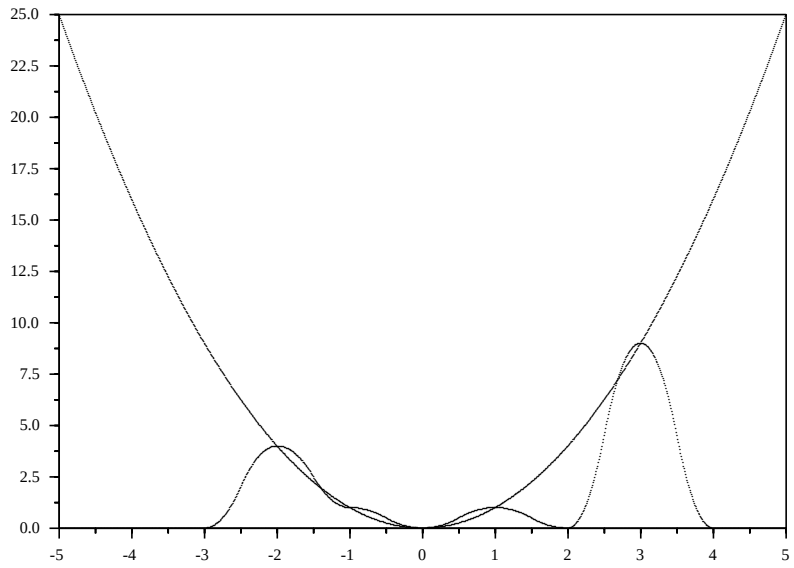



Figura 3.9: A soma alterada com os valores de $f$ em alguns nós, produz uma aproximação $P(f)$ de $f$ .

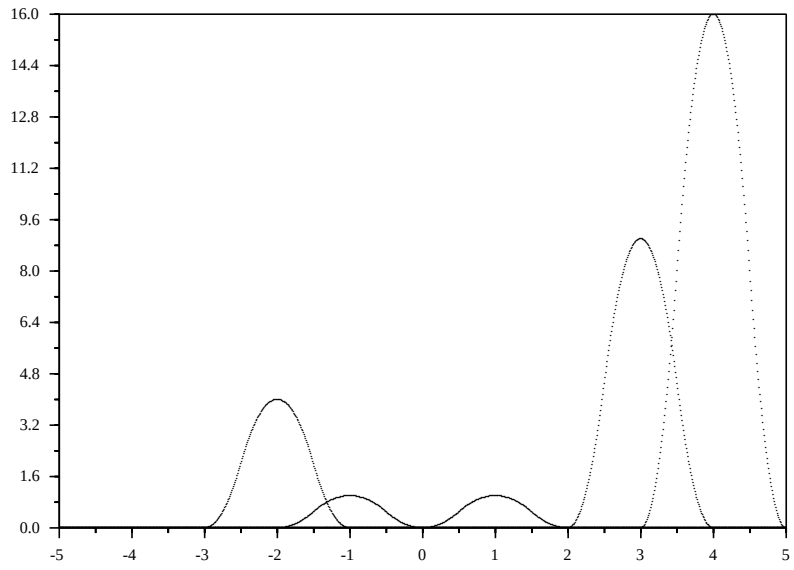



Figura 3.8: Alterando a soma de funções usando como coeficientes alguns valores de uma função dada.

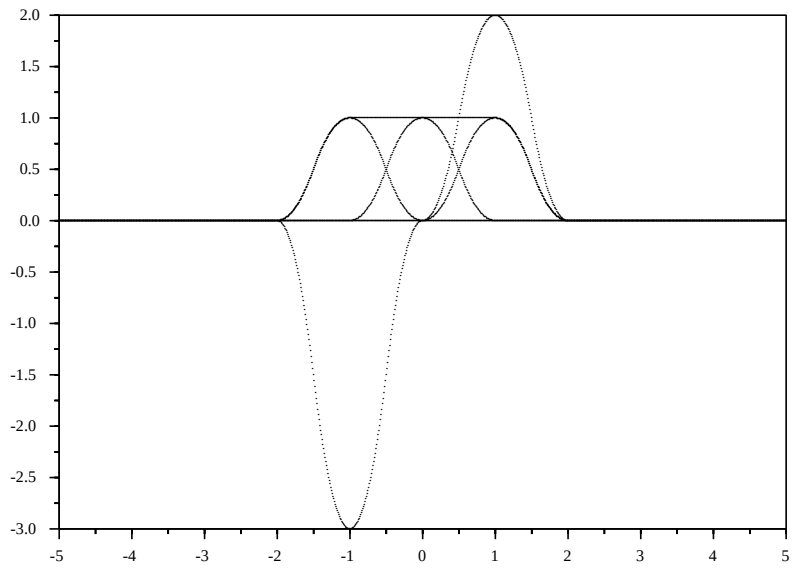



Figura 3.7: Uma combinação linear de uma família de funções que compõe uma *partição da unidade.*

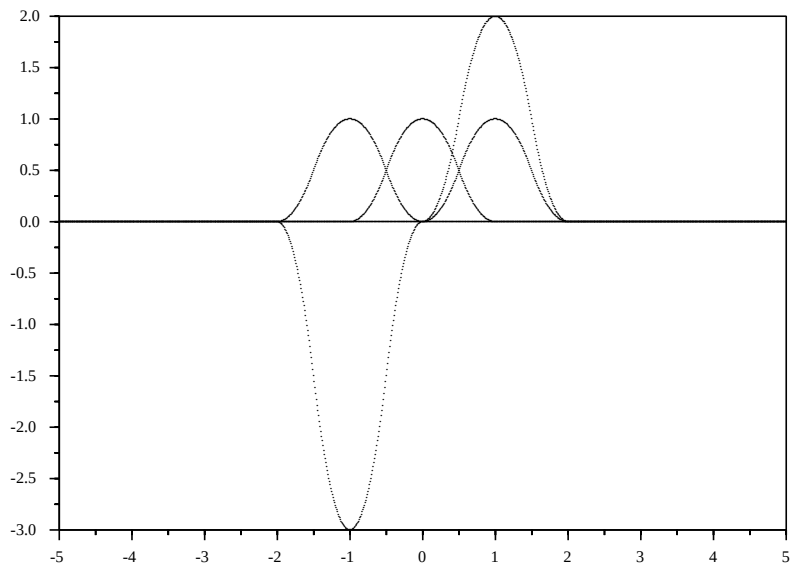



Figura 3.6: Uma família de funções que compõe uma *partição da unidade* .

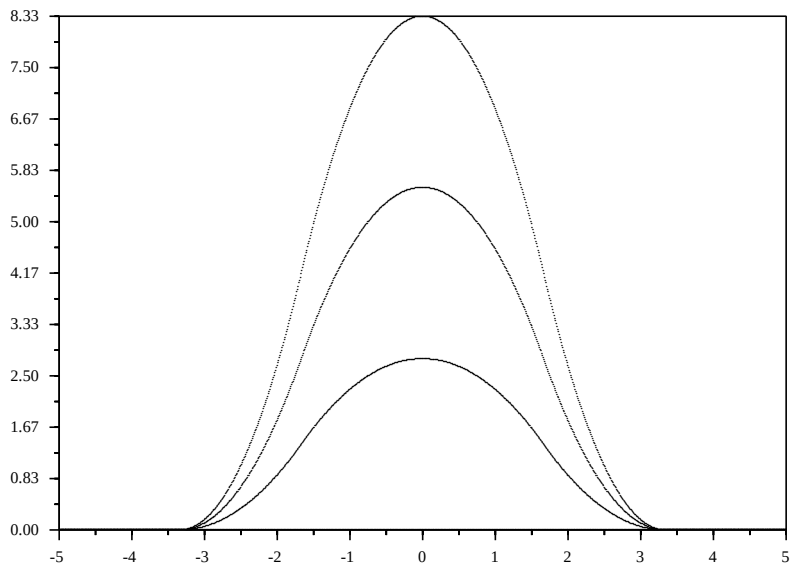



Figura 3.5: sinais parabólicos, bolhas formadas de parábolas por pedaços.

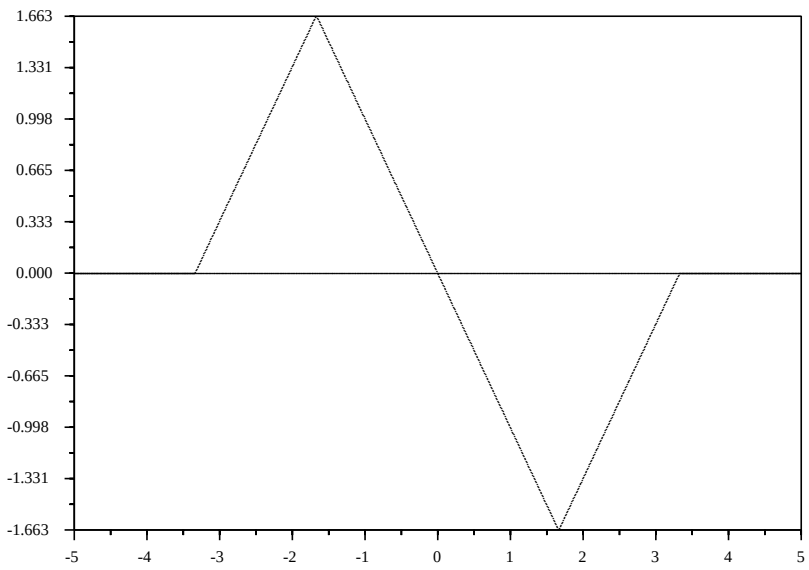



Figura 3.4: função com duas bolhas triangulares.

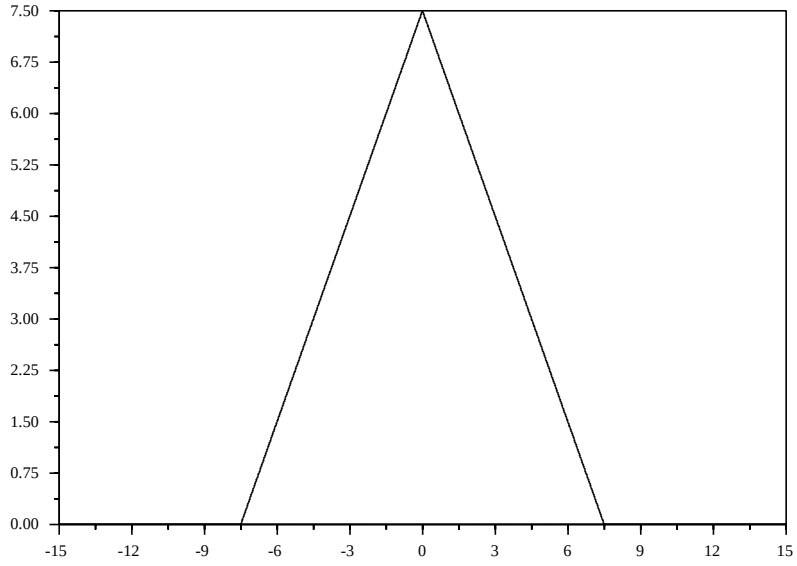



Figura 3.3: Um sinal formado de retas por pedaços com suporte no intervalo $[-7.5, 7.5]$.

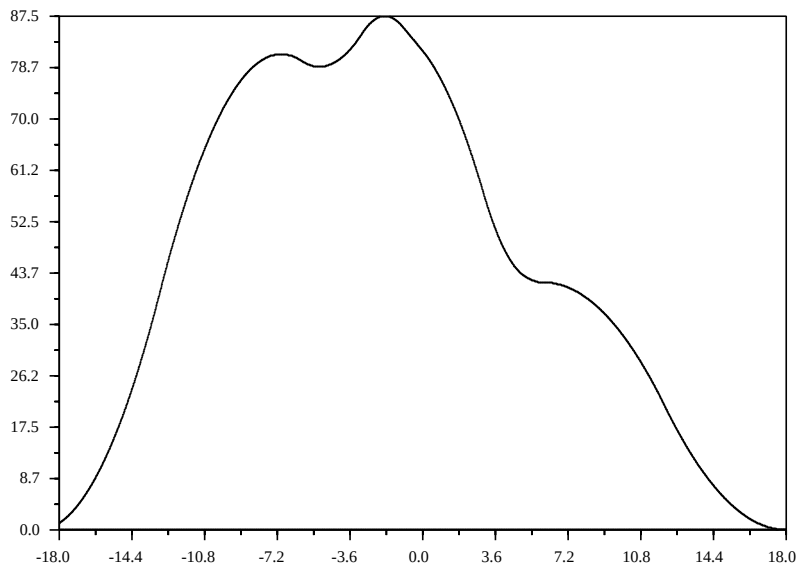



Figura 3.2: um exemplo de sinal .

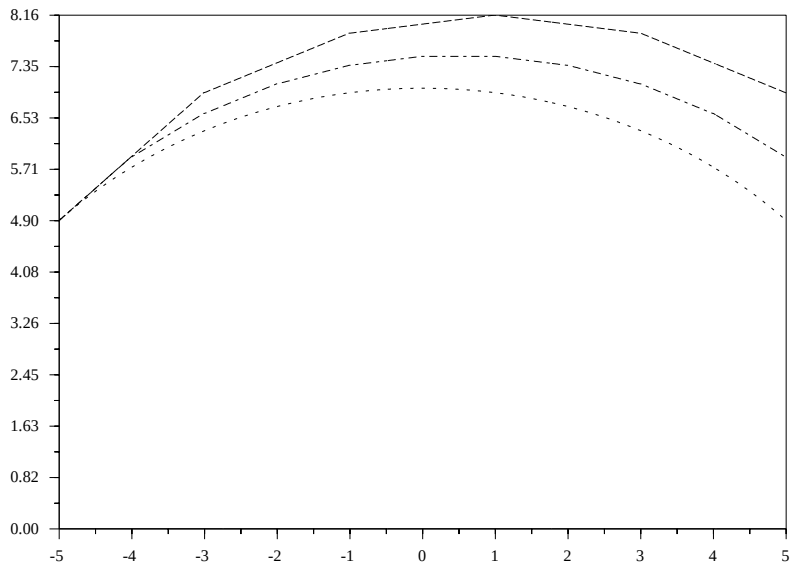



Figura 3.1: Solução aproximada, pelo método de Euler. *uma poligonal que aproxima uma solução particular*

2. *Resolva o problema* $y'' = 0; y'(0) = y(0) = 0$

3. *Resolva o problema:*

$$y'' + 3y' + y = 0 \quad ; y(0) = y'(0) = 1$$

*no intervalo* $[-5, 5]$.

**Dem**: Como $P(D)$ é uma combinação linear de de derivações de ordens sucessivas, estamos considerando portanto uma soma de derivadas

$$\frac{d^k \eta * g}{dx^k}$$

e nos basta demonstrar então que

$$\frac{d^k \eta * g}{dx^k} = \frac{d^k eta}{dx^k} * g \tag{3.79}$$

porque depois somaremos as parcelas para obter o resultado final se este for válido. Mais uma redução na demonstração, se valer a equação (eq. 20, valerá qualquer iterada da mesma. Basta-nos, portanto, demontrar a primeira derivada, que é (teorema 20) já demonstrado. q.e.d.

Vamos discutir a precisão da convolução:

**Teorema: 25** *Aproximação uniforme por convolução .* $|(P(D)\eta * g)(x) - P(D)g(x)|$ *é uniformente pequeno num intervalo fechado* $[a, b]$ *sendo* $\eta$ *um núcleo com suporte* $[-\epsilon, \epsilon]$. **Dem**: :

Como $g$ por hipótese é uma função que tem todas as derivadas indicadas no operador diferencial $L = P(D)$ então podemos simplificar a notação discutindo:

$$|\phi * g - g|$$

é uniformemente pequeno em que $g$ é uma função contínua. Por definição temos:

$$\begin{aligned} &|\phi * g(x) - g(x)| = \\ &= |\int_{-\infty}^{\infty} \phi(t) g(x-t) dt - \int_{-\infty}^{\infty} g(x)\phi(t) dt| = \\ &= |\int_{-\epsilon}^{\epsilon} \phi(t)(g(x-t) - g(x)) dt| \leq \\ &= \int_{-\epsilon}^{\epsilon} \phi(t)|g(x-t) - g(x)| dt \leq \\ &\leq \delta \end{aligned} \tag{3.80}$$

em que para qualquer erro $\delta$ podemos escolher $\epsilon$ tal que a desigualdade acima valha pela continuidade uniforme de $g$ no intervalo $[a, b]$. q.e.d.

O significado geométrico do teorema anterior se pode expressar dizendo que $\phi * g$ se encontra numa *vizinhança tubular* de $g$ sobre o intervalo $[a, b]$.

**Exercício: 32** *Equações lineares.*

1. *Para a equação* $y' = x$ *encontre uma matriz 50 x 50 que discretize* $P(D)\phi * y = x$ *e encontre uma aproximação da parábola que resolve esta equação no intervalo* $[-2, 2]$

da partição considerada do intervalo $[a, b]$ :

$$\begin{array}{l} \sum_{k=0}^{n-1} \eta(x_i - x_k)\Delta x X_k \approx h(x_i) \\ \sum_{k=0}^{n-1} a_{ik} X_k \approx h(x_i) \\ i = 0, ..., (n-1) \end{array} \tag{3.78}$$

cuja solução $\{X_0, ..., X_{n-1}\}$ é um conjunto de $n-1$ pontos que aproximam $f$ em cima dos nós $\{a = x_0, ..., x_{n-1} = b\}$ do intervalo $[a, b]$. A matriz $(a_{ik})_{i,k}$ que obtivemos depende da escolha do núcle e da partição $\{a = x_0, \cdots, x_i, \cdots, x_{n-1} = b\}$ uniforme de $[a, b]$. Esta *dependência* é *fraca* no sentido de que se considerarmos duas sucessões de núcleos convergindo para *Delta de Dirac* os erros entre as respectivas soluções diminue a medida que o índice da sucessão cresce. Isto ainda significa que a real dependência é do suporte do núcleo, porque se a medida do suporte tender a zero, a solução aproximada tende à solução exata.

**Exercício: 31**

*Use um programa de computação algébrica para fazer estas experiências.*

1. *Considere alguns sinais $\phi$ simétricos em relação à origem. Represente graficamente $\phi$ e $P(D)\phi$ para alguns operadores diferenciais lineares.*
2. *Considere $P(x) = x^2 + 3x + 1$ determine a matriz $(a_{ik})_{ik}$ correspondente a discretização de $P(D)y = sen$ em $[-5, 5]$ com uma particao uniforme de passo = 0.1.*

### 3.7.2 As demonstrações .

Começaremos por demontrar a iteração do (teorema 22), página 171.

**Teorema: 24** *Memorização de $P(D)$ via convolução.*

$$P(D)(\eta * g) = P(D)(\eta) * g.$$

$$
\begin{array}{l}
\int\limits_{-\infty}^{\infty} \phi(x-t)f(t)dt \approx h(x) \\
\int\limits_{a}^{b} \phi(x-t)f(t)dt \approx h(x) \\
\sum\limits_{k=0}^{n-1} \phi(x-a-k\Delta x)f(a+k\Delta x)\Delta x \approx h(x) \\
\sum\limits_{k=0}^{n-1} \phi(x-x_k)f(x_k)\Delta x \approx h(x) \\
\sum\limits_{k=0}^{n-1} \phi(x-x_k)X_k \approx h(x)
\end{array}
\tag{3.77}
$$

com as seguintes justificativas:

1. como $\eta$ é um núcleo então

$$\phi * f = P(D)\eta * f \approx h$$

   memoriza aproximadamente o operador diferencial;

2. a suposição de que a $f$ tenha suporte $[a, b]$ justifica a segunda integral;

3. escrevemos com uma soma de Riemann, associada a uma partição uniforme do intervalo $[a, b]$, uma aproximação da integral anterior;

4. chamamos $x_k = a + k\Delta x$ a coordenada de ordem $k$ de um vetor que tem $n$ coordenadas. São os nós da partição uniforme do intervalo $[a, b]$;

5. chamamos de $(X_k)_k = (f(x_k))_k$ a imagem por $f$ do vetor $(x_0, \cdots, x_k, \cdots, x_{n-1})$. Este vetor representa uma aproximação da igónita do problema que estamos resolvendo.

O resultado destas transformações é uma equação linear se discretizarmos a função $h$ que é um *dado* do problema, considerando seus valores em cada um dos nós

$$\{x_0, \cdots, x_i, \cdots, x_{n-1}\}$$

que desejamos resolver:

$$P(D)\phi * f \approx h \tag{3.75}$$

**Observação: 40** Tipos de aproximação. A escolha de sinais positivos elimina um problema sério de aproximação : as oscilações das funções aproximantes e de suas derivadas. Veja o gráfico (fig. 3.10) que se encontra na página 189. Observe que o gráfico da função contínua tangencia o primeiro segmento de reta *por cima* e a seguir tangencia o segmento de reta seguinte, *por baixo* de modo a representar uma média entre os dois segmentos de reta. Isto se deve às duas propriedades do núcleo usado na convolução:

1. positivo;
2. integral valendo 1.

Se você ainda comparar com a aproximação obtido com os polinômios trigonométricos, veja (fig. 1.8), na página 55, os polinômios trigonométricos oscilam em torno da função que eles aproximam. Podemos mostrar mostrar que os polinômios trigonométricos são convolução com o núcleo de Dirichlet que tem apenas a propriedade (2) das enumeradas acima. Observe que *oscilar* não chega a ser um defeito, é devido a esta oscilação que a aproximação, do ponto de vista da integral, dos polinômios trigonométricos é muito boa.

O próximo exercício representa um laboratório para que você compreenda de que oscilação falamos na observação anterior:

**Exercício: 30** *Oscilação da derivada.*

1. *Calcule "geometricamente" a derivada de um núcleo, use alguma definida no exercício 29.* Faça o gráfico do núcleo e desenhe sua derivada num sistemas de eixos sincronizado com o primeiro.

2. *Calcule as derivadas dos núcleos* $(\phi_a)$ *definidos no exercício 29 para* $a \in \{1, 2, 4, 8, 16\}$ .

Para resolver o problema (eq. 3.75) vamos fazer uma discretização de

$$P(D)\phi * f.$$

Para simplificar a notação vamos designar: $\phi = P(D)\eta$ e portanto

$$\eta * f = P(D)\phi * f \tag{3.76}$$

Se $\phi$ for um $n-$spline, a primitiva considerada acima será um polinômio por pedaços de grau $n+1$ com classe diferenciabilidade superior a de $\phi$ de uma unidade, logo um $(n+1)-$spline

Finalmente temos que ajustar o suporte com a *mudança de variável*:

$$\eta := x \mapsto 2\eta(2x)$$

é uma transformação que preserva a integral e diminue a medida do suporte porporcionalmente ao coeficiente 2 portanto $supp(\eta) = [-\epsilon, \epsilon]$. $\boxed{\text{q.e.d.}}$

O exemplo 3.36, 144 é uma construção deste tipo em que $\phi_0$ é 1-Spline, um núcleo triangular e portanto $\eta = \phi_1$ é um 2-Spline. No gráfico (fig. 3.12) o processo do (exemplo 3.36) se repete agora com um núcleo-n-spline em lugar de um núcleo-1-spline, ilustrando o conteúdo teorema.

**Observação: 39** Splines não polinomiais.

Alguns autores fazem referências a um tipo de splines *não polinomiais*. A construção do teorema 23 se serve naturalmente para a construção de tais splines, basta que $\phi$ seja um núcleo obtido por um processo não polinomial e de classe $\mathcal{C}^n$ para que o núcleo resultante do teorema seja de classe $\mathcal{C}^{n+1}$. Tais objetos seriam *splines* não polinomiais que podem ser usados na construção dos projetores que definimos anteriormente.

O algoritmo do teorema 23 é indutivo e podemos assim construir sinais de diferenciabilidade arbitraria, finita. Podemos, com uma outra construção alternativa produzir sinais de classe $\mathcal{C}^\infty$ usando regularização por convolução , uma vez que a regularidade de um núcleo se transfere para a função com ela convoluida.

No exercício 29 você encontra ainda outro método alternativo para construir sinais ou *núcleos*.

Continuaremos do fato de que podemos construir sinais com grau arbitrário de diferenciabilidade com suporte $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$ , e $\epsilon$ arbitrariamente pequeno. A construção acima além do mais mostra que eles podem ser até mesmo (n+1)-Spline a suporte compacto $[-\frac{\epsilon}{2}, \frac{\epsilon}{2}]$ simétricos relativamente a origem, portanto sinais de classe $\mathcal{C}^{n+1}$.

Passaremos agora a uma aproximação da equação diferencial linear

$$P(D)f = h \tag{3.74}$$

em que $\phi$ é uma *núcleo*. Como a aproximação independe, em certa forma, da escolha da núcleo[9], desde que o suporte tenha medida pequena, vamos supor, para tornar as coisas mais simples, que $supp(\phi)$ seja o intervalo $[-\epsilon, \epsilon]$, com $\epsilon$ pequeno e positivo e $\phi$ um (n+1)-spline simétrico relativamente à origem. Um tal núcleo pode ser indutivamente construido da seguinte maneira:

$$\begin{array}{l}
1)\text{Suponha que } \phi_0 \text{ seja um núcleo n-Spline} \\
\quad \text{com suporte } [-\epsilon, \epsilon] \\
3)\text{Defina:} \\
\quad 1)\ \eta_0(x) = \phi_0(x+\epsilon) - \phi_0(x-\epsilon) \\
\quad 2)\ A = \int\limits_{-\infty}^{\infty} \eta_0(t)dt \\
\quad 3)\ \eta_1(x) = \frac{1}{A} \int\limits_{-\infty}^{x} \eta_0(t)dt \\
4)\phi_1(x) = 2\eta_1(2x) \text{ é um núcleo(n+1)-Spline.} \\
\quad \text{cujo suporte é o intervalo } [-\epsilon, \epsilon]
\end{array} \tag{3.73}$$

**Teorema:** **23** *Existência de $(n+1)-$Spline.*

*O algoritmo 3.73 constroi indutivamente núcleos-splines de ordem arbitrária finita.*

**Dem**:

Se $\phi$ for um núcleo centrado em 0 e com suporte igual a $[-\epsilon, \epsilon]$ então $\phi_1 = \phi(x+\epsilon) - \phi(x-\epsilon)$ é uma função com a mesma classe de diferenciabilidade de $\phi$, diferença de duas funções com suportes disjuntos e positivas, ou uma soma de funções de suportes disjuntos e anti-simétricas em torno da origem, assim $supp(\phi_1) = [-2\epsilon, 2\epsilon]$.

Chamemos de $\eta$ uma primitiva de $\phi_1 = \phi(x+\epsilon) - \phi(x-\epsilon)$ com condição inicial menor ou igual a $2\epsilon$. $\eta$ terá derivada nula para $|x| > 2\epsilon$ e como para estes mesmos valores de $x$ ela será a soma das duas integrais com sinais contrários, então vai se anular para todo $x; |x| > 2\epsilon$.

Assim $\eta$ é a suporte compacto.

Se a dividirmos pelo valor da integral teremos um núcleo:

$$\eta := \frac{\eta}{\int\limits_{-\infty}^{\infty} \eta(t)dt}.$$

[9] a aproximação independe até mesmo de uma unidade aproximada, que é uma sucessão de núcleos que "converge" para *Delta de* Dirac, mas isto precisa ser demonstrado... também é preciso explicar o que significa "converge".

**Observação: 38** *Coeficientes variáveis de um Operador Diferencial.*

*Veja que não nenhuma condição especificamente necessária sobre a função $h$ na demonstração do teorema anterior. Tudo que é necessário éque $h(x)$ exista para todo $x$ pelo menos em valor médio, portanto $h$ pode ser até mesmo uma função não -contínua.*

*Isto resolve o exercício 29.5, se $h$ for uma função a suporte compacto $P(D)\phi$* pode *ser a suporte compacto mesmo que $\phi$ ou alguma derivada sua não o seja.* Pode...

Usaremos estes métodos no próximo parágrafo na solução aproximada de equações diferenciais.

## 3.7 Solução de equações lineares.

Vamos tratar esta questão em duas etapas. Primeiro vamos descrever o método em linhas gerais e chegar a um sistema de equações que discretiza a equação diferencial linear que desejamos resolver. Depois vamos verificar a precisão obtida com a aproximação.

### 3.7.1 O método.

Nesta seção vamos usar a *regularização por convolução* para resolver o seguinte problema:

$$L(y) = g(x) \quad ; \quad y(a) = b \tag{3.71}$$

em que $L = P(D)$ é um operador diferencial linear de ordem $n$. O método consiste em discretizar o problema (eq. 3.71) e assim transformá-lo num *sistema de equações lineares*. Este método é usado por outos autores, ver Lanczos, [10, cap. 4, pag 172], por exemplo. Chegaremos ao mesmo resultado de Lanczos, mas com outra metodologia.

O ponto central é a equação

$$\phi * g \approx g \tag{3.72}$$

Vamos terminar esta seção com um teorema que mostra que o método que temos a nossa disposição tem um alcance bastante amplo na solução aproximada de equações diferenciais lineares.

Se $\phi$ for um sinal e $f$ for localmente integrável, então $\phi * f$ é uma função do parâmetro x:

$$(\phi * f)_x = \int_{-\infty}^{\infty} \phi(t) f(x-t) dt$$

e se multiplicarmos a integral pela constante $x$ temos:

$$x(\phi * f)_x = x \int_{-\infty}^{\infty} \phi(t) f(x-t) dt = \int_{-\infty}^{\infty} x\phi(t) f(x-t) dt = [(x\phi) * f]_x$$

porque $x$ é apenas uma constante multiplicando a integral. Valem os mesmo calculos para $h(x)$ em lugar de $x$

$$h(x)(\phi * f)_x = h(x) \int_{-\infty}^{\infty} \phi(t) f(x-t) dt = \int_{-\infty}^{\infty} h(x)\phi(t) f(x-t) dt = [(h(x)\phi) * f]_x$$

Podemos assim generalizar o teorema 21 em que usamos apenas coeficientes constantes no polinômio que definia o operador diferencial:

**Teorema: 22** *Convolução e operadores diferenciais II.*

*Se $\phi$ for uma função de classe $\mathcal{C}^n$ e se $P(D)$ for um operador diferencial de ordem $n$ a coeficientes variáveis então*

$$P(D)(\phi * g) = P(D)(\phi) * g$$

*se a convolução existir.*

Estamos enunciando o teorema com uma amplitude que ultrapassa as condições em que ele foi enunciado anteriormente. Veja as observações que fizemos em 21. Aqui sempre haverá uma função a suporte compacto envolvida na convolução .

*. No início do século o físico inglês Dirac inventou a derivada de $H_0$ que chamou de $\delta_0$ criando uma acirrada polêmica com matemáticos por que além de tudo ele dizia que a derivada $\delta_0$ herdava as propriedades dos núcleos que representavam uma sua aproximação :*

$$\begin{array}{l} \int_{-\infty}^{\infty} \delta_0 dt = 1 \\ \delta_0 \geq 0 \\ \delta_0(0) = \infty \\ \delta_0(x) = 0 \quad \Leftarrow \quad x \neq 0 \end{array} \tag{3.70}$$

*criando assim o* objeto *inteiramente inaceitável para os padrões matemá-ticos. Somente por volta de 1945 os matemáticos conseguiram explicar a* aberração *criada por Dirac. Vários matemáticos chegaram mais ou menos simultâneamente a uma teoria que generaliza o conceito de função , inclusive um matemático português,* Sebastião Silva. *A teoria que é mais conhecida pertence a um francês, Laurent Schwarz, se chama* Teoria das Distribuições *, na qual ele criando um novo objeto, as* distribuições *, classifica a* delta *de Dirac, $\delta_0$, como um exemplo deste novo objeto.*

*Uma das propriedades da* delta *de Dirac é*

$$\delta_0 * f = f.$$

*de modo que a* delta de Dirac *é a unidade relativamente a operação de convolução e um núcleo $\phi$ representa uma aproximação da* delta *de Dirac inclusive neste sentido:*

$$\phi * f \approx f \quad :$$

núcleos *são* quase-*unidades relativamente à convolução. O gráfico (fig. 3.11) na página 190, apresenta o resultado da convolução de uma função com tres núcleos cujos suportes diminuem em medida. Uma sucessão de núcleos com esta propriedade se chama uma* unidade aproximada, *no sentido de que é uma sucessão que se aproxima da* delta de Dirac.

4. *Se $\phi$ for a função definida no primeiro exercício, prove que se $P(D)$ for um operador diferencial de segunda ordem a coeficientes constantes, então $P(D)\phi$ não é a suporte compacto. Calcule o suporte de $P(D)\phi$.*

5. *Retirando-se a hipótese* a coeficientes constantes, *prove com um exemplo que $P(D)\phi$ pode ser a suporte compacto.*

6. *Se*
$$\phi(x) = \begin{cases} 0 \Leftarrow & x \notin [-1,1] \\ (1-x)^3(x+1)^3 & x \in [-1,1] \end{cases}$$
*então* $\sum_{k=-5}^{5} \phi_k \equiv 1 \Leftarrow x \in [-5,5]$. *Assim* $(\phi_k)_{k=1..5}$ *é uma partição da unidade de classe* $\mathcal{C}^2$. *Verifique que não é de classe* $\mathcal{C}^3$.

7. *Se*
$$\phi(x) = \begin{cases} 0 \Leftarrow & x \notin [-1,1] \\ (x-1)^4(x+1)^4 & x \in [-1,1] \end{cases}$$
*verifique que* $(\phi_k)_{k=1..5}$ *é uma partição da unidade e determine sua classe de diferenciabilidade.*

8. *Em cada caso acima, encontre $A$ tal que $\frac{1}{A}\phi$ seja um núcleo.*

9. *Considere um núcleo $\phi$ com suporte $[-1,1]$. Prove que $\phi_a(x) = a\phi(ax)$ é também um núcleo com suporte $[-\frac{1}{a}, \frac{1}{a}]$ desde que $a > 0$. Faça os gráficos de $\phi_a$ com $a \in \{1,2,4,8,16\}$.*

10. *Obtenha graficamente $H_0 * \phi_a$ em que $\phi_a$ está definida no exercício anterior, sendo um núcleo diferenciável. Calcule $(H_0 * \phi_a)'$ e conclua que se tem assim uma aproximação para a derivada de $H_0$ que não existe no* <u>*sentido usual*</u>.

**Observação: 37**

*O último exercício tem um sentido histórico que devemos ressaltar aqui. Como as funções $(H_0 * \phi_a)$ são diferenciáveis no sentido usual e o resultado da diferenciação é $\phi_a$ e, como $\phi * f \approx f$, então a conclusão natural é que $(H_0 * \phi_a)' \approx H_0'$*

compacto a demonstração e o teorema representam o que precisamos.

Com indução finita poderiamos generalizar o teorema anterior:

**Teorema: 21** *Convolução e operadores diferenciais.*

*Se $\phi$ for uma função de classe $\mathcal{C}^n$ e se $P(D)$ for um operador diferencial de ordem $n$ a coeficientes constantes então*

$$P(D)(\phi * g) = P(D)(\phi) * g$$

**Exercício: 29** *Núcleos, convolução e aproximação .*

1. *O programa* **compara.pas** *compara as derivadas primeira e segunda de uma função definida no próprio programa,* formalmente, *por* quociente de diferenças, *e por* convolução *,*
   (a) *Troque o* Núcleo *alí definido por outro que seja de classe $\mathcal{C}^4$ e rode o programa para analisar os valores obtidos.*
   (b) *Mantendo o* núcleo *parabólico, inclua na listagem de dados os valores dos dois núcleos para efeito de comparação de dados da segunda derivada.*
   (c) *Produza um processo estatístico com uma coletânea de funções para verificar se uso de um* núcleo *4-spline garante melhor resultado no cálculo da segunda derivada.*
2. *Se*
$$\phi(x) = \begin{cases} 0 \Leftarrow & x \notin [-1,1] \\ (x-1)^2(x+1)^2 & x \in [-1,1] \end{cases}$$
   *então* $\sum_{k=-5}^{5} \phi_k \equiv 1 \Leftarrow x \in [-5,5]$. *Assim* $(\phi_k)_{k=1..5}$ *é uma partição da unidade contínua e diferenciável.*
3. *Verifique que a função $\phi$ definida no item anterior não é de classe $\mathcal{C}^2$, (duas vezes derivável continuamente).*

$$\begin{aligned}
&\lim_{h=0} \tfrac{\phi * g(x+h) - \phi * g(x)}{h} = \\
&= \lim_{h=0} \int_{-\infty}^{\infty} g(t) \tfrac{\phi(x+h-t) - \phi(x-t)}{h} dt = \\
&= \int_{-\infty}^{\infty} g(t) \lim_{h=0} \tfrac{\phi(x+h-t) - \phi(x-t)}{h} dt = \\
&= \int_{-\infty}^{\infty} g(t) \phi'(x-t) dt
\end{aligned} \tag{3.69}$$

mostramos assim:

**Teorema: 20** *Derivada da convolução .*

*Se a convolução $\phi * g$ existir e se $\phi$ for diferenciável então*

$$(\phi * g)' = \phi' * g$$

A justificativa mais simples para a derivação se permutar com a integral é de que, como as funções integrandas são integráveis, e o intervalo é fechado e limitado, então as somas de Riemann convergem uniformemente, todas, cada uma delas dominada pela norma da partição, logo estas convergências preservam o limite da diferença de quocientes representado pela derivada: *a derivada do limite é o limite da derivada.*

Como a convolução é comutativa, nós poderiamos ter chegado à conclusão de que

$$(\phi * g)' = \phi * g'$$

desde que tivessemos suposto que $g$ fosse diferenciável e com uma pequena modificação na demonstração , (por usamos o suporte compacto de $\phi$). Portanto o teorema não usa o fato de que um dos termos da convolução é um sinal e assim diz mais: se a convolução existir e uma das funções for diferenciável, então a convolução é diferenciável. Entretanto esta nova *formulação* exige uma demonstração diferente da que fizemos acima, como aqui sempre teremos um dos termos da convolução com suporte

*é um núcleo continuamente diferenciável.*

2. *Determine a classe de diferenciabilidade do núcleo definida no exercício anterior.*

3. *Verifique que a função* $\phi_a(x) = a\phi(ax)$ *é um núcleo de classe* $\mathcal{C}^1$ *para qualquer valor de* $a > 0$. *Faça os gráficos desses núcleos para* $a \in \{2, 3, 4, 5\}$.

4. *(***) Calcule* $\phi_5 * H_0$

Vamos continuar analisando resultados que justificam o título desta seção .

Suponha que $\phi$ seja um sinal diferenciável. No bloco anterior de exercícios você tem uma família deles e inclusive um método para construir quantidades indefinidas de tais sinais. Pelo teorema dos acréscimos finitos, (*Teorema do diferencial*), se tem:

$$\phi(x+h) - \phi(x) = \phi'(x)h + o_x(h) \tag{3.67}$$

em que a função $o$ é uma função de duas *variáveis* que tem o mesmo suporte que $\phi$ na *variável* $x$ e

$$\lim_{h=0} \frac{o(h)}{h} = \lim_{h=0} \mathcal{O}(h) = 0,$$

por definição da *derivabilidade* de $\phi$, assim

$$\begin{aligned} &\tfrac{\phi * g(x+h) - \phi * g(x)}{h} = \int_{-\infty}^{\infty} g(t) \tfrac{\phi(x+h-t) - \phi(x-t)}{h} dt = \\ &= \int_{-\infty}^{\infty} g(t)(\phi'(x-t) + \mathcal{O}(h))dt = \\ &= \int_{-\infty}^{\infty} g(t)\phi'(x-t)dt + \int_{-\infty}^{\infty} g(t)\mathcal{O}_x(h)dt. \end{aligned} \tag{3.68}$$

Como a função $O$ tem limite zero uniformemente relativamente a variável x: $\sup_x \mathcal{O}$ depende de $h$ e converge para zero com $h$, e além disto tem mesmo suporte que $f$ então podemos intercambiar a integral com o limite:

$x \mapsto \frac{1}{x}$ não tem esta propriedade no ponto 0. A razão desta última função não ter esta propriedade no ponto 0 reside no fato de que ela não integrável em nenhuma vizinhança de zero.

Daí tiramos a solução da questão : *se uma função g for localmente integrável então ela terá média em todos os seus pontos.* Como $\phi * g(x) = VM_\phi(g)(x)$ é o valor médio de g relativamente a $\phi$ no ponto $x$ se tomarmos o limite

$$\lim_{h=0} \frac{1}{h} \int_{x-h}^{x+h} g(t)dt \ .$$

teremos o valor *instâneo* de $\phi * g$ no ponto $x$ que nada mais é que $\phi * g(x)$, isto é $\phi * g$ tem como limite no ponto $x$ o seu valor no ponto $x$: é uma função contínua. Demonstramos assim:

**Teorema: 19** *Continuidade de $\phi * g$.*

*Se g for localmente integrável, então $\phi * g$ é contínua para todo* núcleo $\phi$.

Vamos ver mais alguns exemplos e resultados que salientam o fato de que com convoluções , *i.e. com médias volantes*, podemos construir funções mais regulares produzindo aproximação para outras funções . Os exercícios do bloco acima fornecem vários exemplos. Mesmo assim vamos construir uma família de exemplos semelhantes e numa direção que depois será de importância para os nossos objetivos. Para que você nos acompanhe com mais atenção lhe propomos que efetue os cálculos seguintes sob nossa cuidadosa assistencia...

**Exercício: 28** *Construção de núcleos cujos suportes decrescem em medida.*

*O exercício marcado com (***) não é dificil, mas certamente é trabalhoso...*

1. *Verifique que a função*

$$\begin{array}{l} x < -1 \ \Rightarrow \ \phi(x) = 0 \\ x \in [-1, 1] \ \Rightarrow \ \phi(x) = \frac{15}{16}(x+1)^2(x-1)^2 \\ x > 1 \ \Rightarrow \ \phi(x) = 0 \end{array}$$

tem a medida do suporte cada vez menor enquanto que a amplitude dos mesmas é cada vez maior. Isto se faz obviamente necessário para que se mantenha a relação de grandeza necessária à constância de

$$\int_{-\infty}^{\infty} \phi_i(t)dt = 1 \quad .$$

Sinais servem para "injetar" uma pequena quantidade energia num *sistema*, funcionando como um pequeno acréscimo feito. Isto pode ser usado, entre outras coisas, para medir "taxas de variação ". Aqui estamos usando sinais, *núcleos*, de suporte pequeno para regularizar funções ou para aproximá-las.

**Exemplo: 27** *Regularização de funções por convolução .*

*No último bloco de exercícios vimos que $\phi * H_0$ é uma função contínua que se aproxima da função de Heaviside $H_0$. Isto é:*

$$\phi * H_0$$

*é uma função continua que foi obtida a partir da função* <u>*descontinua*</u> *$H_0$ em convolução com um núcleo* <u>*também descontínuo*</u> *resultando numa função contínua que aproxima $H_0$.*

Podemos tomar carona com este exemplo para analisar mais a fundo quando é que a convolução produz uma função contínua de uma função descontínua. Usando a linguagem pitoresca de [23], quando fizermos uma média volante com um núcleo $\phi$ em torno dos ponto de uma função $g$ poderemos ter uma média apenas se a função $g$ tiver valor médio em todos os seus pontos:

$$\frac{1}{h}\int_{x-h}^{x+h} g(t)dt \ .$$

As funções contínuas tem esta propriedade. Há funções descontínuas que não têm esta propriedade: a função

Isto demonstra:

**Teorema: 18** *Convolução e aproximação .*

*Se $\phi$ for um núcleo com suporte pequeno, então $\phi * g \approx g$ no sentido de que $\phi * g(x)$ é o $\phi$-valor médio de $g$ numa vizinhança de $x$.*

A seguinte lista de exercícios mostra que na proporção em que o suporte de um núcleo $\phi$ perde *medida*, a convolução $\phi * f$ produz funções mais próximas de $f$:

**Exercício: 27** *Convolução com núcleos, aproximação .*

1. *Considere a função de Heaviside*

$$H_0 = 1 - \chi_{(-\infty,0]}$$

   *calcule $\phi * H_0$ em que $\phi = \chi_{[-\frac{1}{2},\frac{1}{2}]}$*

2. *calcule $\phi * H_0$ em que $\phi = (\frac{5}{2})\chi_{[-\frac{1}{5},\frac{1}{5}]}$*
3. *calcule $\phi * H_0$ em que $\phi = (\frac{10}{2})\chi_{[-\frac{1}{10},\frac{1}{10}]}$*
4. *calcule $\phi * H_0$ em que $\phi = 10\chi_{[-\frac{1}{20},\frac{1}{20}]}$*
5. *calcule $\phi * \phi$ em que $\phi = \chi_{[-\frac{1}{2},\frac{1}{2}]}$*
6. *calcule $\phi * g$ em que $\phi = (\frac{5}{2}) * \chi_{[-\frac{1}{5},\frac{1}{5}]}$ e $g = \chi_{[-\frac{1}{2},\frac{1}{2}]}$*

Os experimentos que fizemos acima nos conduziram à conclusão de que $\phi * f$ é uma aproximação de $f$ tanto melhor quanto $supp(\phi)$ seja pequeno. Na última sequência de exercícios se pode tam-bém concluir que núcleos (que tenham integral 1 e suporte pequeno) têm que, necessáriamente, assumir valores muito grandes, por exemplo, a sequência de núcleos

$$\phi_1 = \chi_{[-\frac{1}{2},\frac{1}{2}]} \tag{3.63}$$

$$\phi_2 = 2.5\chi_{[-\frac{1}{5},\frac{1}{5}]} \tag{3.64}$$

$$\phi_3 = 5 * \chi_{[-\frac{1}{10},\frac{1}{10}]} \tag{3.65}$$

$$\phi_4 = 10 * \chi_{[-\frac{1}{20},\frac{1}{20}]} \tag{3.66}$$

*e assim a integral*

$$x \overset{g}{\mapsto} (\phi * f)(x) = \int_{-\rho}^{\rho} \phi(t) g(x-t) dt$$

*é uma a média ponderada de $N$ valores de $f$ tomados numa $\rho$-vizinhança de $x$ em que $N$ e $\rho$ são arbitrários. Isto é escolhendo um núcleo $\phi$ podemos calcular com esta integral o "valor médio ponderado de $f$" subordinado a um núcleo $\phi$ uma vez que a vizinhança de $x$ depende do $supp(\phi)$. É algumas vezes interessante ter uma notação precisa para este valor médio. Costumamos chamar este valor médio que depende de um determinado peso de* valor médio viciado *pelo peso $\phi$ em que, naturalmente, $\phi$ é um núcleo.*

**Definição : 17** *$\phi$-valor médio* viciado *de uma função .*

*Seja $\phi$ um núcleo com suporte $[-\rho, \rho]$, então a integral*

$$VM_{\phi}(g)(x) = (\phi * g)(x) = \int_{-\rho}^{\rho} \phi(t) g(x-t) dt$$

*é o $\phi$-valor médio de $g$ no ponto $x$.*

Há autores, ver [23], que descrevem a convolução , *geometricamente*, como uma *média volante*, porque é este o resultado da convolução com um núcleo, uma média volante em cada ponto.

A principal consequência destes fatos é que $\phi * f$ é uma função que aproxima $f$ "à maneira de $\phi$" sendo esta aproximação tanto melhor quanto o suporte de $\phi$ seja pequeno porque a média fica melhor localizada em volta de cada ponto x[8].

No exercício acima a aproximação é muito precária porque o suporte da unidade aproximada é muito grande, tem aplitude 1. Veja (fig. 3.11).

[8] Os polinômios trigonométricos são também uma convolução, com um núcleo que oscila levemente em torno de zero, um núcleo que não é positivo.

*Encontre $f*g$ analitica e graficamente e analise a diferença*

$$f*g(x)-f(x).$$

*Faça com um programa em Pascal, por exemplo, o gráfico de $f(x), f*g(x), f*g(x)-f(x)$*

3. *Verifique que $\chi_{[-\frac{1}{2},\frac{1}{2}]}$ é um núcleo e mostre que se $f$ for contínua, então , se $g=\chi_{[0,1]}*f$, $g(x)$ é o valor médio de $f$ numa vizinhança de $x$.*

**Exemplo: 26** *"Solução " do último exercício.*

*Consideremos $\phi*f$ em que $\phi$ é um núcleo, (integral é 1 e o suporte é pequeno), $\chi_{[-\frac{1}{2},\frac{1}{2}]}$ é um exemplo de núcleo.*

*Vamos acrescentar mais uma hipótese sobre o suporte de $\phi$: $supp(\phi)=[-\rho,\rho]$ e mostrar que se $g=\phi*f$ então $g(x)$ é o valor médio de $f$ numa $\rho$-vizinhança de $x$.*

$$\begin{aligned}
g(x)=(\phi*f)(x)&=\int_{-\rho}^{\rho}\phi(t)f(x-t)dt\approx\\
\approx&\sum_{k=0}^{N-1}\phi(-\rho+k\Delta x)\cdot f(x+\rho-k\Delta x)\cdot\Delta x\\
=&\sum_{k=0}^{N-1}\phi(-\rho+k\Delta x)\cdot\Delta x\cdot f(x+\rho-k\Delta x)
\end{aligned}$$

*Como a integral de $\phi$ é 1 então a soma*

$$=\sum_{k=0}^{N-1}\phi(-\rho+k\Delta x)\cdot\Delta x\cdot f(x+\rho-k\Delta x)$$

*representa uma média aritmética ponderada dos $N$ números*

$$\underbrace{f(x+\rho),f(x+\rho-\Delta x),\cdots,f(x+\rho-2\rho+\Delta x)}_{N}$$

*que são os valores de $f$ nos pontos:*

$$\underbrace{x+\rho,x+\rho-\Delta x,\cdots,x-\rho+\Delta x}_{N}\in[x-\rho,x+\rho]$$

O teorema 16 não estabelece claramente um domínio para a operação binária *convolução* como seria correto fazer. O caminho para se conseguir isto seria um pouco longo e sugerimos ao leitor interessado a leitura dum livro de análise, por exemplo [11] onde se pode encontrar a operação de convolução inserida propriamente num contexto algébrico. Para nossas necessidades seria um exagero construir aqui todo este aparato algébrico. Com o Teorema 16 sabemos que quando fizermos convolução com um sinal então a outra função pode pertencer a uma classe bastante ampla para que a operação ainda tenha um signicado preciso. Isto é tudo que precisaremos aqui.

A lista de exercícios seguinte tem a finalidade de deixar o leitor habituado com a nova operação . É preciso lembrar, entretanto, que a convolução ficou durante anos relegada a circunstância de uma *operação de natureza teórica* porque é muito trabalhosa para ser efetuada a mão. A convolução entre duas funções é sempre feita dentro de programas onde ela é rapidamente efetuada. Use um programa para efetuar as contas abaixo, e possivelmente fazer uma ou duas a mão para *sentir o problema*.

**Exercício: 26** *Convolução .*

1. *Calcule as convoluções abaixo:*
   - *(a)* $\chi_{[0,1]} * \chi_{[0,1]}$
   - *(b)* $\chi_{[0,1]} * \chi_{[0,0.1]}$
   - *(c)* $10 \cdot \chi_{[0,1]} * \chi_{[0,0.1]}$
   - *(d)* $\chi_{[0,1]} * \chi_{[-0.1,0.1]}$
   - *(e)* $5 \cdot \chi_{[0,1]} * \chi_{[-0.1,0.1]}$
   - *(f)* $(\chi_{[0,1]} * \chi_{[0,1]}) * \chi_{[0,1]}$
2. *Seja* $f(x) = x^2 + 3x + 1$ *e*

$$\begin{array}{l} g(x) = x + 1 \;\Leftarrow\; x \in [-1, 0) \\ g(x) = 1 - x \;\Leftarrow\; x \in [0, 1] \\ g(x) = 0 \;\Leftarrow\; x \notin [-1, 1] \end{array}$$

sinal,

$$f * g(x) = \int_{-\infty}^{\infty} f(t)g(x-t)dt = \tag{3.60}$$

$$= \int_{a}^{b} f(t)g(x-t)dt = \tag{3.61}$$

$$= -\int_{x-a}^{x-b} f(x-y)g(y)dy = \int_{x-b}^{x-a} f(x-y)g(y)dy \tag{3.62}$$

em que $[a, b]$ contém o suporte de $f$. As duas últimas equações, na sequência acima, são as expressões *computacionais* da convolução, estão no ponto de serem colocadas num programa de computação. Basta que $g$ seja localmente integrável para que estas integrais existam, para cada valor do parâmetro x, o que se pode ver claramente das duas últimas.

**Teorema: 16** *Convolução com um sinal.*
*Se $\phi$ for um sinal e $g$ for uma função localmente integrável, então $\phi * g$ está bem definida.*

O próximo teorema é de uso prático frequente:

**Teorema: 17** *Comutatividade da Convolução .*
*Se $f * g$ existir, então $f * g = g * f$*
*Dem:*

$$(f * g)(x) = \int_{-\infty}^{\infty} f(t)g(x-t)dt = \lim_{\substack{a=-\infty \\ b=\infty}} \int_{a}^{b} f(t)g(x-t)dt =$$

*se $t := x - t$ temos:*

$$= \lim_{\substack{a=-\infty \\ b=\infty}} -\int_{x-a}^{x-b} f(x-t)g(t)dt =$$

*observando que $x$ é uma "constante", passando ao limite temos finalmente*

$$\lim_{\substack{a=-\infty \\ b=\infty}} -\int_{x-a}^{x-b} f(x-t)g(t)dt = \int_{-\infty}^{\infty} f(x-t)g(t)dt = (g * f)(x) \ \square$$

O gráfico (fig. 3.10) ilustra o uso das *núcleos*. Nele você pode ver uma função descontínua, linear por pedaços, e sua convolução com um núcleo. O resultado é uma função contínua que é uma *aproximação* da função linear por pedaços. É isto que se chama de *regularização por convolução*: a função $g$ é descontínua, $G = g * f$ é contínua, e $G$ é uma aproximação de $g$. Na (fig. 3.10), página 189 você pode ver tres gráficos obtidos usando-se tres "valores" distinitos para *núcleo*, $\phi_1, \phi_2, \phi_3$ definidos assim ($f$ é um núcleo) :

$$\phi_1(x) = 2f(2x) \tag{3.57}$$
$$\phi_2(x) = 4f(4x) \tag{3.58}$$
$$\phi_3(x) = 8f(8x) \tag{3.59}$$

**Observação: 36** Mudança de variável.
A operação

$$f \mapsto_a f :_a f(x) = af(ax)$$

produz uma nova função $a-$acelerada em relação a função $f$ e ao mesmo tempo $a-$dilatada, relativamente a mesma função inicial. Conseqüencia disto: ${}_a f$ tem a mesma integral que $f$.

As funções que formam as partições da unidade no pará-grafo anterior são exemplos de sinais, mas nem sempre serão *núcleos*. Sinais ou núcleos não precisam ser contínuos. Há exemplos de núcleos que têm suporte não limitado, mas isto é uma outra história que não vai nos interessar aqui. Fica apenas o registro para que o leitor saiba da possibilidade: a propriedade central dos núcleos é que tenham integral 1 como se verá em seguida, o resto é contornável.

Um outro exemplo de sinal é a função característica de um intervalo limitado. Se a medida do intervalo for 1 este sinal será um núcleo, e neste caso, um núcleo descontínuo.

Se na convolução uma das funções for um **sinal** então a integral que define a convolução se restringe a um intervalo limitado, para cada valor de $x$: suponhamos que $f$ seja um

$$f * g(x) = \int_{-\infty}^{\infty} f(t)g(x-t)dt \tag{3.54}$$

que define uma nova função , *se a integral na* (eq. 3.54) *for finita para todos os valores de* $x$. A convolução entre duas funções $f * g$ nem sempre está definida, por exemplo se $f = g = sin$ então $f * g$ não existe porque a integral *não converge* para nenhum valor de $x$. Precisamos de algumas condições sobre as funções para garantir esta convergência, uma condição *suficiente* mas não *necessária* é que uma das funções que esteja sendo *convoluida* tenha suporte compacto.

**Observação: 34** Domínio da convolução.

Uma questão bastante difícil é procurar o *domínio da convolução*, quer dizer, um espaço $\mathcal{D}$ de funções tal que se $f, g \in \mathcal{D}$ então $f * g$ está definida. Em geral não nos interessa isto a não ser que desejemos estudar as propriedades da *operação* convolução. Não é este o nosso objetivo aqui, com frequência se usa a convolução de forma mais amena:

$$g \overset{\mathcal{F}}{\to} G = \mathcal{F}(g) \tag{3.55}$$

$$\mathcal{F}(g)(x) = G(x) = \int_{-\infty}^{\infty} f(t-x)g(t)dt = f * g(x) = G(x) \tag{3.56}$$

em que a convolução aparece como um processo intermediário para definir uma transformação sobre outras funções. Neste caso o centro da questão é a função $f$ que representa uma espécie de *matriz* de transformação *via* convolução. No sistema de equações acima a função $g$ foi transformada na função $G = f * g$.

É sob esta ótica que estaremos usando a convolução aqui.

Um tipo particular de *sinal* para o qual teremos uso frequente é:

**Definição : 16** *Núcleo.*

*Chamaremos de* **núcleo** *a um sinal cuja integral seja 1 e cujo suporte tenha* <u>*medida* pequena</u>.

**Observação: 35** Multi uso da palavra núcleo.

A palavra *núcleo* é usada com significados diferentes em vários contextos. Algumas vezes faz referência à função que atua como matriz na definição de um operador integral.

Não confundir com *núcleo de um operador integral* que ainda é outra coisa diferente: é o subespaço das soluções nulas de uma equação linear. Neste capítulo estaremos usando a palavra núcleo exclusivamente com o sentido da definição (def. 16).

Obtivemos assim uma aproximação para $y'$ a qual vamos aplicar o método integral de Picard para obter uma solução aproximada:

$$y_1 = b + \int_a^t f_n(x)dx \tag{3.53}$$

$$y_1 = b + \int_a^t f_n(x)dx =$$

$$y_1 = b + \int_a^t \sum_{x_k} g(x_k, y_k)\phi_{x_k}(x)dx = b + \sum_{x_k} g(x_k, y_k) \int_a^t \phi_{x_k}(x)dx$$

$$y_2 = b + \int_a^x (b + \sum_{x_k} g(x_k, y_k) \int_a^t \phi_{x_k}(x)dx)dt$$

$$= b + b(x-a) + \sum_{x_k} g(x_k, y_k) \int_a^x (\int_a^t \phi_{x_k}(x)dx)dt$$

Repetidas integrações levam a expressão :

$$\sum_{k=0}^{n-1} b(x-a)^k + \sum_{x_k} g(x_k, y_k) \int_a^x \cdots \int_a^t \phi_{x_k}(x)dx \ldots dt$$

## 3.6 Regularização por convolução .

Vamos desenvolver neste parágrafo uma técnica que nos auxiliará no próximo ao discretizarmos um operador diferencial linear e portanto na solução aproximadas de equações diferenciais lineares. Ela também representa uma alternativa diferente na construção de splines, ver, [17].

Dadas duas funções integráveis $f, g$ em $\mathbf{R}$, definimos a operação

é uma solução aproximada do (problema 3.48), página 154. A precisão da solução se encontra associada ao refinamento da partição do intervalo em que está definida a "variável livre". Observe que se a partição da unidade for polinomial então a solução é do tipo *splines* com todas as facilidades que isto representa.

**Exercício: 25** *Resolva os problemas abaixo usando uma partição uniforme de malha* $h := 0.1$ *no intervalo indicado e com a condição inicial dada.*

1. $y' = e^{-x^2}$ ; $y(0) = 1$ ; $x \in [0, 10]$
2. $y'' = 0; y'(0) = 0$ ; $y(0) = 0$ ; $x \in [0, 10]$
3. $y' = \frac{x^3}{x^6+1}$ ; $y(0) = 1$ ; $x \in [0, 10]$

### 3.5.5 Solucao periódica de $y' = g(t, y)$.

Vamos extender o método usado na seção anterior para resolver a equação

$$y' = g(t, y); (t_k, y_k)_{t_k \in \Pi[a,b]} \tag{3.51}$$

em que ao invés de ter uma única condição inicial, temos uma família de condições iniciais. Um exemplo de problema deste tipo é quando se procura uma solução periódica para uma equação .

A ideia consiste em aplicar o método de Picard a uma aproximação da derivada obtida por uma interpolação -Spline seme-lhante a que já usamos antes.

Consideremos a partição da unidade $(\phi_{t_k})_{t_k \in \Pi[a,b]}$ e vamos memorizar em $\phi_{t_k}$ as informações da equação diferencial discretizada:

$$y'(t) \approx \sum_{t_k} g(t_k, y_k)\phi_{t_k}(t) = f_n(t) \tag{3.52}$$

Na literatura científica, os Splines cúbicos são de longe os mais utilizados. A razão disto se podem enumerar:

1. O grau é baixo, o que torna estas funções computacionalmente confiáveis. Elas dificilmente saem da faixa de precisão dos computadores a não ser, obviamente para grandes números.

2. O fato de serem cúbicos lhes junta uma variação de concavidade que permite excelente aproximação.

### 3.5.4 Solução de $y' = g$ com 2-Splines.

Vamos aplicar o método desenvolvido na última seção para resolver equações diferenciais. Como primeiro passo vamos resolver uma equação bem simples, mas que tem sua importância específica uma vez que nem sempre podemos encontrar formalmente uma primitiva.

Para resolver o problema:

$$\frac{dy}{dx} = g(x) \ ; \ y(a) = b \ ; \ x \in [A, B] \tag{3.48}$$

vamos considerar uma partição uniforme de $[A, B]$ com passo $h$ sendo $[A, B]$ um intervalo em que o problema tem solução , isto é $g$ está definida em todos os pontos deste intervalo. Se expressarmos g com auxílio da equação 3.41 teremos:

$$g(x) \approx \sum_{x_k \in \Pi_n[a,b]} \phi_{x_k}(x) g(x_k) = P_n(x) \tag{3.49}$$

em que $n$ representa o número de elementos de $\Pi_n[a, b]$ então a integral

$$b + \int_a^x P_n(t)dt \tag{3.50}$$

necessáriamente todos os coeficientes tem que ser nulos, (porque todos os somandos são iguais a 1 sobre os nós da partição ), então concluimos que a família geradora é *linearmente independente* e portanto uma base para o espaço vetorial em questão que vamos designar, muito sugestivamente de

$\boxed{\text{Espaço de k+1-Splines.}}$

$$\mathcal{S}_{\mathcal{P}}. \tag{3.47}$$

A dimensão de $\mathcal{S}_{\mathcal{P}}$ é número de elementos de $\mathcal{P}$ que por sua vez, em consequência da construção que fizemos no parágrafo anterior é o número de nós da partição considerada no intervalo $[a, b]$.

Um "processo de convergência" [7] fica gerado quando sucessivamente refinamos as partições de $[a, b]$ e introduzimos assim mais elementos na base $\mathcal{S}_{\mathcal{P}}$.

Como o objetivo deste parágrafo era o de escrever uma definição para Spline vamos agora fazê-lo. Um Spline é um elemento de um espaço de Splines, que por sua vez é a imagem de um projetor associado a uma partição da unidade construida com $k + 1 - Splines$.

**Definição : 15** *Espaço de Splines sobre* $[a, b]$.

- *Uma função polinomial por pedaços de grau* $k+1$ *a suporte compacto contido em* $[a, b]$ *e de classe* $\mathcal{C}^k$ *é um* $k+1$*-Spline sobre* $[a, b]$.
- *Uma partição da unidade subordinada a uma partição de* $[a, b]$ *formada de* $k + 1$*-Splines gera um espaço vetorial de dimensão finita* $\mathcal{S}_{\mathcal{P}}$ *cujos elementos são* $k + 1$*-Splines.*
- *A dimensão de* $\mathcal{S}_{\mathcal{P}}$ *é o número de nós da partição considerada sobre* $[a, b]$.

[7] a convergência que se tem aqui é semelhante a das somas de Riemann, não se pode falar diretamente de convergência como das sucessões, porque há uma quantidade não enumerável de sucessões convergentes, se houver convergência.

Como estas construções foram feitas *abundamentemente* no capítulo anterior, não precisamos incluir aqui nenhum exemplo, e você sabe não somente que eles existem em qualquer quantidade e forma e que além disto há uma pluralidade de métodos para construí-los, satisfazendo a todos os gostos. Na próxima seção você ainda verá um outro método de construção destes objetos.

Uma partição da unidade feita com $k+1$-Splines é um objeto muito especial: tem $m$ elementos:

$$\mathcal{P} = \{P_1, \cdots, \cdots, \cdots, P_m\} \tag{3.43}$$

e com eles podemos construir combinações lineares:

$$g = \alpha_1 P_1 + \cdots, \cdots, \cdots + \alpha_m P_m \tag{3.44}$$

que voltam a ser polinômiais por pedaços do mesmo grau comum $k+1$ de cada elemento $P_i$ da soma (eq. 3.44)[6]. A classe de continuidade sendo também a mesma de cada elemento da soma, $g$ é uma função polinômial por pedaços, de grau $k+1$ e de classe $\mathcal{C}^k$.

Observe que a equação 3.44 é uma equação funcional.

Se considerarmos todos os coeficientes $\alpha_i = 0$ em 3.44 o resultado será a função identicamente zero. Se tomarmos uma coleção qualquer com $m$ coeficientes, construindo assim a função $g$, ao trocarmos todos os sinais dos coeficientes, resulta na função $-g$ que somada a anterior reproduz 0.

Vemos assim, que o leitor termine os passos, que a equação 3.44 define um espaço vetorial de funções gerado por

$$\mathcal{P} = \{P_1, \cdots, \cdots, \cdots, P_m\}. \tag{3.45}$$

Esta família se chama *geradora*. Como, se impusermos a condição

$$g = \alpha_1 P_1 + \cdots, \cdots, \cdots + \alpha_m P_m = 0 \tag{3.46}$$

[6] Aqui é preciso alterar a definição de grau, caso contrário a afirmação anterior fica falsa. Vamos entender grau como *menor ou igual* ao grau comum aos elementos da família. Assim a função identicamente nula tem grau $k+1$... esta é uma modificação temporânea, para efeito local.

dos gráficos estão as parcelas individualmente representadas, já multiplicadas pelos coeficientes $f(\{-2,-1,0,1,2,3,4\ )$ aplicados aos átomos que produzem o projetor. No outro estão os gráficos conjuntos de $f$ e de $P(f)$.

Neste caso particular, *partições da unidade* polinomiais, a *projeção* é um Spline que aproxima $f$. No caso acima

$$\sum_{x_k \in \Pi[a,b]} \phi_{x_k}(x) f(x_k) \approx f(x)$$

é um 2-Spline. Se quisermos um 3-Spline basta construir um sinal de classe $C^2$ $\phi$ com pedaços de polinômio de grau 3.

Na próxima seção construiremos os Splines, afinal.

**Observação: 33** Projetores.

A sombra de um objeto tridimensional sobre o solo é uma *projeção*. Esta palavra é usada muitas vezes com o sentido de uma redução da dimensão. No caso da sombra, o objeto é tridimensional e sua sombra é bidimensional.

Ao projetarmos perdemos parte das informações, este é o caso típico de uma aproximação. As aproximações são descrições de objetos em que algumas informações *são perdidas* ou simplesmente *não se encontram disponíveis*.

Este é sentido dos *projetores* em Matemática: buscam um espaço de objetos sobre os quais tenhamos domínio, por exemplo sejam *computacionais*, e criam nele uma imagem de um objeto que desejamos estudar. É o que fazem os polinômios trigonométricos, os splines, os polinômios de Taylor.

Os polinômios trigonométricos captam aspectos ondulatórios das funções e as projetam num espaço de ondas. Quanto maior for a quantidade termos, mas precisa será a informação.

Os splines captam informações sobre o comportamento diferenciável das funções, com frequência apenas o valor no ponto, e as projetam num espaço de splines. Quanto maior for a informação de valores ponto a ponto das funções, mais precisa será a projeção.

### 3.5.3 Splines.

Estamos em condições de definir de maneira simples, (depois de tanto material teórico...), os Splines.

Como dissemos no capítulo 3, um Spline é uma função polinômial por pedaços de um determinado grau $k+1$ e de classe de continuidade $\mathcal{C}^k$. Consequentemente um sinal construido por pedaços de polinômio de grau $k+1$ que tenha todas as derivadas até a ordem $k$ contínuas é um exemplo de $k+1$-Spline.

unidade. Observe que a equação 3.41 que foi obtida da equação numérica 3.40 volta a ser uma equação funcional que pode ser ainda escrita asssim:

$$g = \sum_{x_k \in \Pi[a,b]} \phi_{x_k} f(x_k) \approx f \tag{3.42}$$

e voltamos a incentivá-lo a fazer um esforço para <u>ver</u> a diferença entre as duas formas de escrever: a equação 3.42 é uma equação funcional, a equacao 3.41 é uma equação numérica.

Veja o gráficos (fig. 3.8) e (fig. 3.9 em que usamos os coeficientes:

$$f(-2), f(-1), f(0), f(1), f(2), f(3), f(4) \ ; \ f(x) = x^2.$$

O efeito da soma, (equação funcional):

$$\sum_{x_k \in \Pi[a,b]} f(x_k)\phi_{x_k} \approx f$$

é uma aproximação da função $f(x) = x^2$ que tem precisão total sobre os nós $\{-2, 0, 2, 4\}$. Se quisermos ter maior precisão bastará reconstruir a partição da unidade com uma rede mais fina de nós.

Nos dois gráficos (fig. 3.8) e (fig. 3.9) se pode ver comparativamente os gráficos de $f$ e de sua aproximação com este projetor-partição da unidade. Observe que os gráficos, são semelhantes apesar de haver uma deformação que torna difícil de ver esta semelhança. Esta deformação, já dissemos isto antes, se deve a uma otimização que *Scilab* faz da tela gráfica para ocupá-la toda com o gráfico. Esta otimização é muito comum em todos os programas de computação algébrica ou numérica e representa a melhor saida dentre as poucos que se tem para fazer uma boa representação de funções graficamente. Em um

4. Nestas condições a identidade seguinte se verifica sobre $[a, b]$:

$$f \cdot \sum_{x_k \in \Pi[a,b]} \phi_{x_k} = \sum_{x_k \in \Pi[a,b]} \phi_{x_k} f = f. \tag{3.39}$$

porque na primeira expressão da identide acima, temos o produto da função $f$ por uma função que é identicamente 1 sobre o intervalo $[a, b]$. A segunda expressão consiste de aplicação da propriedade distributiva da multiplicação relativamente à adição. As duas expressões coincidem, portanto com a função $f$ sobre o intervalo $[a, b]$. Uma outra forma de expressar a identidade acima é:

$$f(x) \cdot \sum_{x_k \in \Pi[a,b]} \phi_{x_k} = \sum_{x_k \in \Pi[a,b]} \phi_{x_k} f(x) = f(x). \tag{3.40}$$

particularizando o valor da identidade para cada ponto $x \in [a, b]$. A diferença entre as duas equações é: *a primeira é identidade entre funções , a segunda é uma identidade entre números*, que o leitor se esforce para <u>ver</u> esta diferença.

Na figura (fig. 3.6) vemos o gráfico de algumas das funções duma tal família, neste caso elas são simples translações do sinal parabólico, ver exemplo 3.38, definido antes:

$$x \mapsto \phi(x-4) \ , \ x \mapsto \phi(x-2) \ , \ x \mapsto \phi(x) \ , \ x \mapsto \phi(x+2)$$

A soma destes sinais translatados se encontra no gráfico, (fig. 3.7), página 186.

Se fizermos uma pequena modificação na equação (eq. 3.40), página 149 teremos um resultado inteiramente diferente:

$$\sum_{x_k \in \Pi[a,b]} \phi_{x_k}(x) f(x_k) \approx f(x). \tag{3.41}$$

Agora, em vez de somar as funções multiplicadas ponto a ponto, estamos escolhendo alguns valores de $f$ para servir de coeficientes das multiplicações dos sinais que compõem a partição da

neste gráfico há uma função que assume valores positivos e negativos para evitar que *Scilab* nos impeça de idenficar o que acontece com a partição da unidade[5].

Estas são propriedades particulares de uma classe mais geral de famílias de funções chamadas de *Partições da Unidade*. A última condição é específica das aplicações que nos interessam aqui e em geral é substituida por uma condição mais fraca. Nós vamos trabalhar com estas *partições da unidade finitas* que serão suficientes para os nossos objetivos. Consultando [11] você encontrará uma descrição mais detalhada das Partições da Unidade se o seu interesse for excitado. Uma partição da unidade não precisa ser uma família finita de funções mas em cima de compactos estas famílias sempre se reduzem a um número finito satisfazendo a penúltima propriedade lista na (observação, 32), página 147. Somas finitas são derivaveis, integráveis etc... se as parcelas destas somas o forem, este é o uso das partições da unidade, junto com o fato de que os seus elementos *localizam os problemas e os valores*. Na próxima seção usaremos as partições da unidade na aproximação de funções .

### 3.5.2 Projetores definidos com Partições da Unidade.

Vamos mostrar uma aplicação das partições da unidade.

1. Considere o conjunto das funções reais $f$ contínuas definidas no intervalo: $[a, b]$ e um intervalo aberto $\Omega$ que contenha $[a, b]$:

$$f \in \mathcal{C}([a, b]) \ , \ [a, b] \subset \Omega$$

2. Seja uma partição $\{a = x_0, x_1, \ldots, x_n = b\}$ de $[a, b]$.

3. Considere uma coleção de funções $(\phi_{x_k})_{x_k})$ indexada em $\Pi[a, b]$ satisfazendo as condições 32.

[5] Scilab otimiza a janela gráfica de modo que não conseguiriamos ver a soma das funções.

precisam siquer ser semelhantes, basta que tenham áreas algébricas de sinais contrários.

**Exercício: 24** *Translações e homotetias de sinais.*

1. *Chame* $\phi_o$ *a função definida no exemplo 3.36, página 144, e* $\phi_2(x) = \phi(x-2)$ *uma translação para direita de* $\phi$*. Verifique que* $\phi_2 + \phi_o$ *é uma função a suporte compacto, que seu valor é exatamente 1 sobre o intervalo* $[0,2]$*, sempre positiva, se anula fora do intervalo* $[-2,4]$*.*

2. *Chame* $\phi_k(x) = \phi(x-k)_{k\in\{0,1,\ldots,n}$ *em que* $\phi$ *está definida na página 144. Verifique que a função* $\phi_0 + \phi_2 + \phi_4 + \phi_6$ *é identicamente 1 sobre o intervalo* $[0,6]$*, é positiva, diferenciável, e se anula fora do intervalo compacto* $[-2,8]$*.*

3. *Qual é o suporte de* $\phi_{-2} + \phi_0 + \phi_2$ *?*

No último exercício você foi levado a construir uma família $(\phi_\alpha)_\alpha$ de funções que satisfazem à s seguintes condições :

**Propriedades da família** $(\phi_\alpha)_\alpha$**.**

**Observação: 32** *Propriedades dum partição da unidade.*

1. *É uma família de funções positivas e a suporte compacto.*

2. *Todas as funções da família ficam entre zero e 1.*

3. *A soma das funções é a função constante* 1 *sobre um intervalo compacto da reta.*

4. *Cada uma das funções da família assume o valor 1 em exatamente um ponto.*

O gráfico (graf. 3.6) mostra tres funções que formam uma família do tipo descrito. No gráfico há uma função que assume valores positivos e negativos apenas para efeito comparativo. No (graf. 3.7) se pode novamente ver elementos da *partição da unidade* junto com a soma das mesmas. Também

5. *Considere o sinal* $\phi(x) = \int_{-\infty}^{x} f(t)dt$ *em que* $f$ *é a função definida no (exemplo, 3.38). Mostre que a translação* $\phi_a(x) = \phi(x-a)$ *é também um sinal.*

6. *Mostre a soma de duas translações* $\phi_a + \phi_b$ *é um sinal.*

7. *Mostre que* $\psi(x) = \phi(ax)$ *é um sinal se* $\phi$ *o for e que se* $a > 1$ *o suporte de* $\psi$ *terá* medida *menor do que o suporte de* $\phi$*. Verifique que se* $a < 1$ *a* medida*(supporte(*$\psi$*) será maior do que* medida*(supporte(*$\phi$*). Ver a definição de suporte, (def. 13) abaixo.*

8. *Use um programa de computação algébrica, (ou faça um programa em Pascal) para calcular somas de translações. Verifique que o sinal da (fig. , 3.2) éa soma das tres translações* $\phi_{-a} + \phi_0 + \phi_a$ *do sinal definido no exemplo (exemplo, 3.36).*

9. ** Você pode usar os parâmetros* $a, b$ *como* botões *no formato* $\phi(ax+b)$ *para obter sinais com suporte do tamanho desejado e centrado no ponto* $\frac{-b}{a}$ *e inclusive somas destes sinais com* botões *diferentes. Construa um programa em Pascal que faça isto*[4]*.*

10. **** Você pode fazer que um programa altere os valores dos botões mencionados no exercício anterior, ao apertar as teclas:* $\rightarrow, \uparrow, \downarrow, \leftarrow$ *A função do TurboPascal para isto seria* **ReadKey** *e uma comparação com o código ASCII das setas do teclado.*

Vamos explorar a construção feita no exemplo 3.36 para a construção da família de funções que nos interessa.

Os tres exemplos na figura anterior são de funções a suporte compacto. Qualquer função construida com triângulos cujas áreas se anulem mutuamente terá como primitiva uma função diferenciável e a suporte compacto. Os triângulos não

[4] este método se encontra por traz da generalização que as *wavelets* representam para as séries de Fourier, ou para as transformadas de Fourier. Os "botões" recebem os nomes de dilatação e translação.

- *e os dois ramos externos encontram o eixo OX também tangencialmente.*

*Sua equações são :*

$$\boxed{\textit{Um sinal parabólico, } \phi \textit{ .}}$$

$$\begin{cases} x \in [-2,-1] & \Rightarrow \quad \phi(x) = \frac{1}{2}(x+2)^2 \\ x \in [-1,1] & \Rightarrow \quad \phi(x) = -\frac{x^2}{2} + 1 \\ x \in [1,2] & \Rightarrow \quad \phi(x) = \frac{1}{2}(x-2)^2 \\ x \notin [-2,2] & \Rightarrow \quad \phi(x) = 0 \end{cases} \tag{3.38}$$

*No gráfico (fig. 3.5) abaixo você pode ver tres sinais parabólicos obtidos com coeficientes angulares diferentes impostos aos "sinais" triangulares.*

O exercício seguinte propõe uma outra maneira de construir um sinal diferenciável. Na parte final do exercício você será conduzido a fazer algumas experiências "algébricas" com sinais.

**Exercício: 23** *Construção de um sinal diferenciável.*

1. *Encontre os valores dos parâmetros $\alpha, \beta, \gamma$ de modo que as parábolas*

$$\alpha(x+2)^2, \quad -\beta x^2 + \gamma, \quad \alpha(x-2)^2$$

   *se cortem tangencialmente. Encontre os pontos de tangência e defina uma função diferenciável que se anule fora do intervalo $[-2,2]$.*

2. *Verifique na construção anterior que $\alpha$ não é único.*

3. *Verifique que a função $\phi$ construida no exemplo 3.36 é igual a construida acima com $\alpha = 1$.*

4. *Melhore o método de construção no exemplo 3.36 substituindo os triângulos por dois sinais parabólicos, um deles negativo, de modo a obter uma função positiva, de classe $C^2$, que se anule fora de um intervalo fechado.*

que seu suporte é limitado e portanto "*estará centrado em algum ponto*" do espaço...

A função $f$ construida no exemplo anterior, não é diferenciável, mais interessante seria construir uma função que se anulasse fora de um intervalo fechado e o fizesse diferenciávelmente, como parece ser o caso da (fig. , 3.2). Há duas maneiras simples de se o fazer.

**Exemplo: 25** *Construção de um sinal diferenciável I.*

*Construiremos abaixo uma função $f$ definida por dois triângulos simétricos em torno da origem, como na (fig.,3.35), um deles* negativo, *entretanto:*

$$\begin{cases} x \in [-2,-1] & \Rightarrow \quad f(x) = x+2 \\ x \in [-1,1] & \Rightarrow \quad f(x) = -x \\ x \in [1,2] & \Rightarrow \quad f(x) = x-2 \\ x \notin [-2,2] & \Rightarrow \quad f(x) = 0 \end{cases} \tag{3.36}$$

*O gráfico você pode ver na (fig. 3.4). Se calcularmos a integral $\int_a^t f(x)dx$ com condição inicial $a < -2$, o resultado será uma integral nula para valores de $t > 2$ mas positiv ou nula para qualquer outro valor de $t$:*

*Defina agora:*

$$\phi(x) = \int_{-\infty}^{x} f(x)dx \tag{3.37}$$

*É fácil verificar que*

- *$\phi$ é diferenciável, sua derivada sendo $f$;*
- *que se anula fora do intervalo [-2,2].*
- *Seu gráfico é formado por tres ramos de parábolas que coincidem tangencialmente nos pontos $(-1, \frac{1}{2})$, $(1, \frac{1}{2})$ duas a duas;*

*Mas as funções que se originem de algum fenômeno físico, químico ou biológico podem ter suporte limitado. É preciso não confundir o modelo* algébrico *que representa uma aproximação do fenômeno, com o próprio fenômeno... a intensidade do calor se comporta como* $f(r) = \frac{1}{r}$ *em que* $r$ *é a distância da fonte para distâncias não muito pequenas. Entretanto para distâncias muito grandes se pode supor que o calor que atinge um objeto a partir da fonte é nulo e assim* $f$ *é a* suporte compacto. *Melhor dizendo, os instrumentos que temos para medir, já não são mais sensíveis a quantidade calor que emana da fonte. Aqui entraria em uso uma função como* $\chi_{\mathcal{D}(0,r)}$ *multiplicando aquela que modela algebricamente a distribuição de calor em termos da distância,* $f$*, e cujo efeito será de anular os valores do modelo fora da área de sensibilidade do aparelho medidor. O novo modelo será* $\chi_{\mathcal{D}(0,r)} f$.

Um intervalo fechado e limitado é um conjunto compacto. *Entre outros conjuntos, que podemos definir como compactos, se encontra um disco fechado,* $\mathcal{D}[0,r]$*, de centro* $0$ *e raio* $r$.

Precisamos uma família de funções com as seguintes propriedades: *contínuas, positivas, se anulando fora de um intervalo fechado.* Vamos começar com a construção de tais funções , que chamaremos *sinais*.

**Definição : 14** *Sinal*

*Chamamos de sinal uma função positiva, limitada, integrável e a suporte compacto.*

No gráfico (fig. 3.2) você tem um exemplo sinal. Vamos a seguir mostrar-lhe como construir sinais, algebricamente.

**Exemplo: 24** *Construção de um sinal triangular.*

*Considere a função* $f$ *definida pelas condições :*

$$\begin{cases} x \in [-2,-1] & \Rightarrow \quad f(x) = x+2 \\ x \in [-1,0] & \Rightarrow \quad f(x) = -x \\ x \notin [-2,0] & \Rightarrow \quad f(x) = 0 \end{cases} \tag{3.35}$$

$f$ *é contínua e se anula fora do intervalo* $[-2,0]$. *Seu gráfico é "triângulo" sobre o intervalo* $[-2,2]$

Os exemplos de sinais das figuras (fig.,3.3) e (fig.,3.2) podem induzí-lo num erro de intuição, ambos estão centrados na origem. A única coisa que é verdade a respeito de um *sinal* é

**Definição : 13** *Suporte de uma função.*

*Suporte de uma função* $[a,b] \xrightarrow{f} \mathbf{R}$ *é o menor conjunto fechado que contenha* $\{x; f(x) \neq 0\}$.

**Exemplo: 23** *Funções e seus suportes.*

1. $f(x) = x^2; \mathbf{R} \xrightarrow{f} \mathbf{R}$, *o suporte de* $f = supp(f) = \mathbf{R}$ *porque, como* $f$ *é um polinômio, o conjunto* $\{f(x) \neq 0\}$ *é um conjunto aberto e o único conjunto fechado que o contém é* $\mathbf{R}$. *Qualquer polinômio tem como suporte o intervalo em que estiver definido. Num certo sentido isto significa que o conjunto dos zeros de um polinômio é insignificativo.*

2. *Se* $f = \chi_{(a.b]}$ *for a função característica do intervalo* $(a,b]$ *definida por*

$$\chi_{(a,b]}(x) = \begin{cases} x \in (a,b] & 1 \\ x \notin (a,b] & 0 \end{cases}$$

   *então* $supp(f) = [a,b]$.

3. *Se* $f = \chi_{(a,b)}$ *a função característica do intervalo aberto* $(a,b)$, *então* $supp(f) = [a,b]$

4. *Se* $f$ *representa a intensidade de calor que emana de uma fonte em função da distância, relativamente a um determinado instrumento,* $supp(f) = \mathcal{D}[0,r] =$ *o* disco fechado *de centro* $0$ *e raio* $r$, *em que* $r$ *representa a distância a partir da qual o instrumento mediria como* $0$ *a quantidade de calor detectada e* $0$ *é o ponto onde se localiza a fonte de calor.*

**Observação: 31** *Funções a suporte compacto.*

*Os exemplos acima, exceto o último, deixam uma clara impressão de que funções que tiverem suporte diferente de* $\mathbf{R}$ *são funções construidas para ter um suporte especial. De fato é assim, a quase totalidade das funções definidas por algum processo* algébrico, *e aí incluindo funções como*

$$f(x) = \frac{1}{x} \quad , \quad f(x) = \ln|x|,$$

*tem como suporte* $\mathbf{R}$.

*Isto denuncia uma vantagem para o método de Picard se não levarmos em conta que foram feitas 12 iterações ... ou que os nosso programas são muito ruins.*

A óbvia conclusão dos exemplos que temos acima é ainda a mesma que já tiramos anteriormente: os métodos de aproximação deixam a desejar. Entretanto o método de Picard pode permitir que descubramos a solução exata como o exemplo anterior o mostra. O segundo exemplo indica que isto pode ser algumas vezes difícil de se conseguir.

## 3.5 Interpolação polinomial.

Neste parágrafo vamos discutir um outro método para solução de equações diferenciais ordinárias bastante diferente dos já estudados antes. Ele se baseia em Interpolação polinomial cujo estudo se iniciou no Cápitulo 1 e que será aprofundado com a introdução dos Splines que alí apenas foram anunciados.

Queremos resolver o problema:

$$\frac{dy}{dx} = g(t, y) \ ; \ y(a) = b; \tag{3.34}$$

e para isto vamos *memorizar* os valores da derivada numa familia de funções que gozam de algumas propriedades especiais que tornam isto possivel. Esta família de funções se chama *partição da unidade* e vamos iniciar este parágrafo com a sua construção .

### 3.5.1 Partição da unidade.

Vamos definir o conceito de suporte que intuitivamente significa o conjunto que contem os valores não nulos de uma função. Alguns exemplos abaixo lhe mostrarão que esta conceituação intuitiva é falha.

*Assim*

$$y_2(t) = 2 + \int_0^t g(\tau, y_1(\tau))d\tau =$$

$$y_2(t) = 2 + \int_0^t (2 + 2\tau - \tau)d\tau =$$

$$y_2(t) = 2 + \int_0^t (2 + \tau)d\tau \ = \ 2 + 2t + \frac{t^2}{2}$$

*Continuando teremos:*

$$y_3(t) = 2 + \int_0^t g(\tau, y_2(\tau))d\tau =$$

$$y_3(t) = 2 + \int_0^t (2 + 2\tau + \frac{\tau^2}{2} - \tau)d\tau = 2 + 2t + \frac{t^2}{2} + \frac{t^3}{6}$$

$$= 1 + t + \frac{t^2}{2} + \frac{t^3}{6} + t + 1 \approx e^t + t + 1$$

*onde já vemos aparecer uma aproximação da solução exata*

$$y = e^x + x + 1.$$

**Exemplo: 22** *Uma outra comparação .*

*Podemos aplicar o método de Picard para resolver a mesma equação com a condição inicial* $y(-5) = 5$ *para estabelecer comparação com os resultados do parágrafo anterior, em que resolvemos este problema.*

*As primeiras contas já se revelam complicadas e nada de nítido surge como no exemplo anterior, mesmo sabendo qual é a solução exata...*

*Fazendo uso de um programa de computação algébrica, calculamos 12 iterações do método de Picard e assim obtivemos:*

$$y_{12}(-4.77) = 6.2924$$

*enquanto que o valor obtido com a solução exata,* mas obviamente com aproximação calculada pelo próprio programa com 4 casas decimais, *é* 3.9272. *O valor neste mesmo ponto usando-se qualquer dos métodos anteriores é* 7.4467.

**Teorema: 15** *Teorema de Picard (the little theorem of Picard)*

*Seja $g$ uma função definida e contínua assim como $\frac{\partial g}{\partial x}$ num retângulo $R$ do plano com lados paralelos aos eixos. Dada uma condição inicial $(a,b) \in R^o$ existe então um intervalo $(a-h, a+h)$ contido na projeção de $R$ sobre $O\vec{X}$ tal que $y' = g(t,y)$ ; $y(a) = b$ sendo esta solução única.*

A demonstração do teorema Picard pode ser encontrada em [26],páginas (424-426). A referida demonstra-ção claramente constroi a seqência $(y_k)$ acima e mostra sua convergência. É preciso um pouco mais de Matemática para mostrar que o modelo comum para este teorema e o de Newton-Raphson é o mesmo garantindo a existência de um ponto fixo: teriamos que discutir mais aprofundadamente a *topologia* do *espaço* $\mathcal{C}([a,b])$. O leitor interessado pode consultar [25],páginas (337-339), onde se encontra o *teorema do ponto fixo* que justifica tanto o Teorema de Piccard como o Teorema de Newton-Raphson.

Observe também que no Teorema de Peano, (que é anterior ao teorema de Picard), não se tem unicidade da solução .

Vamos resolver a equação

$$\frac{dy}{dt} = g(t,y) = y - t \ ; \ y(0) = 2$$

que já resolvemos anteriormente, para comparar os resultados.

**Exemplo: 21** *Solucão de $\frac{dy}{dx} = g(x,y)$ pelo Teorema de Picard.*

*Como o método sugere temos que usar um* elemento *inicial ao qual aplicaremos o* operador. *Este* elemento *inicial é uma função e pode ser a função linear que passa pelo ponto $(0,2)$ com derivada $g(0,2)$ que é a aproximação mais simples e elementar para resolver o nosso problema.*

e portanto nos encontramos literalmente dentro do espirito do Método de Newton-Raphson[3]: será que existe um ponto fixo $y_0$ tal que $K(y_0) = y_0$? Se houver um tal *ponto fixo* podemos escrever:

$$y_0(t) = b + \int_a^t g(\tau, y_0(\tau))d\tau, \tag{3.32}$$

cuja derivada nos leva de volta ao problema 3.6:

$$\frac{dy_0}{dt} = g(t, y_0(t)) \ ; \ y_0(a) = b \tag{3.33}$$

do qual $y_0$ é a solução .

**Observação: 30** *Constantes.*

Procure não se confundir com esta observação, mas quando resolvemos uma equação, procuramos um objeto, uma *constante*, que satisfaça determinadas condições. Nas equações elementares se se costuma dizer que esta constante resolveu o problema. Da mesma forma uma solução de uma equação diferencial, (ou de uma equação funcional), é uma função determinada, uma constante. Não confundir com função constante.

A resposta para esta questão é *sim*, dada pelo teorema:

**Teorema: 14** *Teorema de Peano*

*Seja $g$ uma função definida e contínua num retângulo $R$ do plano com lados paralelos aos eixos. Dada uma condição inicial $(a, b) \in R^o$ = interior de $R$, existe então um intervalo $(a - h, a + h)$ contido na projeção de $R$ sobre $O\vec{X}$ tal que $y' = g(t, y)$ ; $y(a) = b$.*

Observe-se que o teorema não fala da existência de um ponto fixo para nenhum operador. Mas este teorema é uma generalização do que vamos agora enunciar, tendo demonstrações praticamente idênticas:

[3] não é nada evidente, mas a situação se assemelha a de uma *progressão geométrica* em que o operador $K$ funciona como a razão multiplicativa.

$$f \overset{K}{\mapsto} h \quad ; \quad K(f)(t) = h(t) = b + \int_a^t g(\tau, f(\tau))d\tau, \tag{3.28}$$

definida num espaço de funções e que aplicada a uma função $f$ resulta n'outra função $h$, em outras palavras temos:

$$K : \mathcal{C}([a,b]) \overset{K}{\mapsto} \mathcal{C}([a,b]) \quad ; \quad h(t) = K(f)(t) = b + \int_a^t g(\tau, f(\tau))d\tau, \tag{3.29}$$

que, para ser verdade, basta que a função $g$ seja contínua no intervalo $[a,b]$ como os cálculos seguintes demonstram:

$$|h(x') - h(x'')| = |\int_{x'}^{x''} g(\tau, f(\tau))d\tau| < C|x' - x''| \quad \Rightarrow$$

$$\Rightarrow \quad \{|x' - x''| < \epsilon \quad \Rightarrow \quad |h(x') - h(x'')| < C\epsilon$$

porque a constante $C$ depende apenas de $f, g, x', x''$

**Observação: 29** *Operadores.*

*Por várias razões, até mesmo de preconceito nos primeiros dias em que funções foram definidas* sobre *outras funções, quando uma função se encontra definida num* espaço de funções *, é costume chamá-la de* operador.

Posto nos termos da equação (eq. 3.29) a equação primitiva (eq. 3.26) pode se escrever assim:

$$y_2 = K(y_1); y_3 = K(y_2); \ldots ; y_{k+1} = K(y_k); \tag{3.30}$$

ou ainda podemos observar que o operador $K$ está sendo *iterado*:

$$y_2 = K(y_1); y_3 = K(y_2) = K(K(y_1)); \ldots ; y_{k+1} = K^k(y_1); \tag{3.31}$$

$$F'(x) = g(t, y) \quad \Rightarrow \quad F(t) = b + \int_a^t g(\tau, y) d\tau. \tag{3.25}$$

Esta expressão é, no entanto, incômoda. Como interpretar $y$ no integrando? Quem resolve isto são os *métodos iterativos*: eles consistem da ideia de *começar alguma coisa de algum lugar*, a partir de um ponto inicial, construindo uma sucessão e verificar *depois* se podemos descobrir condições que garantam a *convergência* da mesma, é nisto que se baseia o método de Newton-Raphson para determinação de raizes de uma equação . A fundamentação teórica final de tais métodos é o teorema do ponto do fixo, ver [25, Apêndice, teorema do ponto fixo].

Vamos re-escrever a equação anterior:

$$y_2(t) = b + \int_a^t g(\tau, y_1(\tau)) d\tau, \tag{3.26}$$

com a suposição de que temos alguma *sugestão* $y_1$ que é a primeira função de uma sucessão

$$y_1, y_2, y_3, \ldots, y_k, y_{k+1}, \ldots$$

que convergeria para a solução do problema (eq. 3.6), página 119. A verificação de convergência de que falamos acima pode consistir apenas na demonstração de que a sucessão construida por este método é de Cauchy. Assim

$$y_{k+1}(t) = b + \int_a^t g(\tau, y_k(\tau)) d\tau. \tag{3.27}$$

Vamos *generalizar mais um pouco*, observando que a equação (eq. 3.26) representa uma função $K$:

O resultado gráfico do programa é uma descrição do *fluxo* das *órbitas* da equação diferencial definida no programa que se encontrem dentro de um retângulo, também definido no programa.

Uma *alta* densidade de segmentos reta pode impedir a visualização do fluxo, assim como uma *baixa* densidade poderá ter o mesmo efeito. Existe uma faixa de densidades *certas*, a ser encontrada experimentalmente, que torna nítido este fluxo para uma *determinada região*.

Com frequência se encontra este problema de seleção de uma *densidade adequada* na aquisição de dados para interpretar um fenômeno. Você irá compreender o que estamos dizendo se rodar o programa algumas vezes.

O nome do método se justifica porque podemos identificar atravez dele os pontos em que há mesmo, *iso*, coeficiente angular. Ele tem uma outra importância adicional: permite algumas vezes se ter uma visão do *fluxo das soluções* .

O método das Isóclinas é um método global, *o que contradiz chamá-lo de método passo a passo...*, porque ele nos fornece o fluxo das soluções numa região . Observe que acima escrevemos apenas a equação diferencial, sem nenhuma condição inicial.

Se fizermos um programa traçar pequenas retas com o coeficiente angular $g(x_k, y_k)$ estaremos implementando *graficamente* o método de Euler. No disco você encontra este programa feito: é a procedure **Isoclinas.pas** que se encontra na biblioteca **curvas**.

O leitor deve observar a ausência de um gráfico nesta seção, ela se deve ao fato de que não conseguimos resolver esta questão com SciLab, [21], ficariamos gratos se alguém nos apontasse esta solução. A solução em Pascal é o programa *isoclinas.pas*.

## 3.4 Um método iterativo.

Uma outra alternativa para resolver o *problema* 3.6 consiste numa aplicação quase direta do Teorema Fundamental do Cálculo:

*com*

$$K_1 = hg_k$$
$$K_2 = hg(x_k + h, y_k + K_1)$$

## 3.3 O método das Isóclinas.

Em geral este método é apresentado assim:

1. Seja $\Omega$ uma região do plano em que está definida a equação diferencial
$$\frac{dy}{dt} = g(t, y).$$

2. *Consideremos uma discretização* de $\Omega$ representada por uma partição qualquer da mesma, para fixar ideias: uma malha retangular cobrindo $\Omega$ da qual tomamos apenas os nós $(x_k, y_k)_k$ que se encontrem dentro de $\Omega$.

3. A solução aproximada pelo *Método das Isóclinas* é um conjunto de segmentos de reta $r_k$ centrado em $(x_k, y_k)$ de comprimento inferior à *norma da malha* e com coeficiente angular $g(x_k, y_k)$.

4. A condição de que os segmentos de reta sejam de comprimento inferior à norma da malha é estética. *Estética* nem sempre é futilidade, muitas vezes pode representar uma classificação essencial. Com esta condição o conjunto destes pequenos segmentos de reta dão uma visão de conjunto das soluções da equação diferencial mostrando o *fluxo das soluções* . Rode o programa *isoclinas.pas* referido abaixo e nele altere o tamanho dos segmentos de reta para entender melhor de que *estética* estamos falando.

**Observação: 28** Interpretação dos dados.

Um problema básico na modelagem de fenômenos pode ser facilmente exemplificado com o programa *isoclinas.pas*. Rode este programa alterando o tamanho dos segmentos de reta $r_k$ e possivelmente também alterando a equação diferencial dada.

Uma outra experiência ilustrativa pode ser feita com a mesma equação e diferente condição inicial:

**Exemplo: 20** *Solução aproximada de* $y' - y = x; y(-5) = 5$.
*A solução exata desta equação é* $y = \frac{6}{exp(5)exp(x)} + x + 4$. *Abaixo você pode ver alguns pares de dados comparativos feitos com* Euler1*:*

```
Dominio do programa: x -> [-37.9 , 37.9]  y -> [-19 , 19]
Este programa resolve graficamente e aprox. a equacao diferencial
 dy/dx = f(x,y)
com a precisao escolhida e condicoes iniciais dadas
precisao = 0.1  condicao inicial (-5,5).

               x = -5.0000 y = 5.0000
Sol exata:     x = -5.0000 y = 5.0000
               x = -4.9900 y = 5.1000
Sol exata:     x = -4.9900 y = 4.9503
               x = -4.8000 y = 7.0900
Sol exata:     x = -4.8000 y = 4.1124
```

*em que se pode ver um erro muito grande. Não melhora nada tentar* Euler2 *ou* Euler3. *Se tentar usar os programas não se esqueça de substituir na função* Teste *a expressão da solução exata para a condição inicial escolhida. Você poderá encontrar no disco o programa* **RungeK** *que é ligeira variante de* **Euler2**.

Numa rápida conclusão , você tem aqui uma única equa-ção para a qual, dependendo da condição inicial, os métodos descritos colapsam. Seria incorreto sugerir que temos métodos numéricos de alta precisão para resolver qualquer equação ...

Entretanto se pode melhorar a aproximação diminuindo o passo e consequentemente consumindo mais tempo.

**Exercício: 22** *Modifique o programa RungeK usando o próximo ponto no formato*

$$y_{k+1} = y_k + \frac{K_1 + K_2}{2}$$

```
               x = 0.5000 y = 3.1474
Sol exata:     x = 0.5000 y = 3.1487
               x = 0.7000 y = 3.7116
Sol exata:     x = 0.7000 y = 3.7138
               x = 0.9000 y = 4.3562
Sol exata:     x = 0.9000 y = 4.3596
               x = 1.0000 y = 4.7141
Sol exata:     x = 1.0000 y = 4.7183
               x = 2.0000 y = 10.3662
Sol exata:     x = 2.0000 y = 10.3891
               x = 3.0000 y = 23.9926
Sol exata:     x = 3.0000 y = 24.0855       erro <  0.1
```

*O programa ConvRot que se encontra no disco chama a seguinte rotina que produz os dados acima, entretanto será necessário fazer alterações nos programas* **Eulerxx** *e* **RungeKxx** *que se encontram dentro da biblioteca* **Curvas**. *Você pode encontrar novas versões de* **RungeKXX** *no disco, fora da biblioteca* **Curvas**. *O programa usa algumas funções definidas na biblioteca* **Curvas** *que se encontra no disco, como por exemplo* **Dfx, Dfy** *que calculam aproximadamente as derivadas parciais em relação a $x$ ou a $y$.*

### 3.2.3 Precisão da solução .

O leitor pode ser tentado a *aprofundar* o método tentando calcular derivadas de maior ordem. Antes fazê-lo use o programa **ConvRot** substituindo *Euler* por *Euler2, Euler3*. Verá que não há grande significado em tentar derivações de maior ordem. Há uma propagação de erro que torna relativamente inútil tentar aproximações polinomiais de ordem muito alta. Em suma, as aproximações polinomiais de grau 1 e 2 são muito efetivas além de simples. A este respeito ver [14], na introdução se discute a diferença entre as técnicas de *diferenciação algorítmica* contra a diferenciação por *diferenças divididas* que é o método usado nos nossos programas em Pascal.

Vamos a um exemplo numérico, mas observe que que programas como **Eulerxx.pas** ou **RungeKxx** podem oferecer vários exemplos a gosto do estudante que neles pode variar tanto as equações como a precisão e assim observar melhor os *efeitos* destes parâmetros. Vamos a um cálculo manual, depois dele incluiremos uma listagem de dados feita por computador.

**Exemplo: 19** *A equação* $\frac{dy}{dx} = y - x$ ; $y(0) = 2$ .

*Aqui* $g(x, y) = y - x$. *A condição inicial diz-nos que procuramos uma curva que passe no ponto* $(0, 2)$.

*Então*

$$\frac{\partial g(t,y)}{\partial t} + \frac{\partial g(t,y)}{\partial y} g(t,y) =$$

$$= -1 + y - x = y - (x + 1).$$

*Assim*

$$f(a+h) \approx f(a) + g(a,b)h + \frac{g_1(a,b)}{2}h^2 = 2 + 2h + \frac{h^2}{2}$$

*Se* $h = 0.1$ *então* $f(0.1) \approx 2+0.2+0.005 = 2.205$.
*Os dados abaixo foram obtidos com o programa Euler2.pas em Pascal que se encontra listado mais adiante:*

```
Dominio do programa: x <- [-37.9 , 37.9]  y <- [-19 , 19]
Este programa resolve aprox. a equacao diferencial
 dy/dx = f(x,y)
com a precisao escolhida e condicoes iniciais dadas
precisao = 0.1 ; condicao inicial (0,2).
               x = 0.0000 y = 2.0000
Sol exata:     x = 0.0000 y = 2.0000
               x = 0.1000 y = 2.2050
Sol exata:     x = 0.1000 y = 2.2052
               x = 0.2000 y = 2.4210
Sol exata:     x = 0.2000 y = 2.4214
               x = 0.3000 y = 2.6492
Sol exata:     x = 0.3000 y = 2.6499
```

equação, em que aparecem as derivadas parciais de $g$ seja igual a (eq. 3.17), o que nos leva a solução seguinte nas variáveis $\lambda_1, \lambda_2, \mu_1, \mu_2$ :

$$\lambda_1 + \lambda_2 = 1 \tag{3.21}$$

$$\lambda_2 = \frac{1}{2} \tag{3.22}$$

$$\mu_1 = \mu_2 = 1 \tag{3.23}$$

portanto:

1. existem estes coeficientes,
2. a nova fórmula que depende apenas do valor inicial de $f$ e de $g$
3. temos uma aproximação com a mesma precisão que a fórmula de Taylor de segunda ordem:

$$y_{k+1} \approx y_k + \frac{1}{2}g(a,b)h + g(t_k + h, y_k + g(a.b)h)\frac{h^2}{2} \tag{3.24}$$

usando apenas os valores de $g$ nos pontos da malha.

Demonstramos assim a validade do método de Runge-Kutta, em uma de suas variantes:

**Teorema: 13** *Método de Runge-Kutta.*

*Dada a equação diferencial $\frac{dy}{dt} = g(t,y)$ e uma malha de pontos em que os valores de $g$ são conhecidos e mais uma condição inicial $(a,b) = (a, f(a))$, podemos encontrar o valor aproximado de uma solução $f$ deste problema no ponto $a+h$, em que $h$ é o passo da malha, com a mesma precisão da fórmula de Taylor de segunda ordem com a fórmula:*

$$f(a+h) = y_{k+1} \approx y_k + \frac{1}{2}g(a,b)h + g(t_k + h, y_k + g(a.b)h)\frac{h^2}{2}$$

grau e portanto podemos utilizar este método em substituição ao anterior para obter uma solução gráfica, *ou uma listagem*, mais precisa da solução aproximada da equação diferencial dada.

Por se baseiar num polinômio de Taylor de segundo grau é que a (eq. 3.17) oferece mais precisão na aproximação do valor de $f(a+h)$. Entretanto um problema de ordem prática se impõe, (a razão por traz do método de Runge-Kutta), os dados desta equação diferencial podem ser resultados experimentais. Os valores de $g$, um levantamento de taxas de variação, podem ser valores tomados em alguns pontos de uma malha, e aí não temos uma equação algébrica para calcular derivadas parciais.

Recordando, queremos encontrar valores aproximados pa-ra uma solução $f$ da equação diferencial $\frac{dy}{dt} = g(t, y)$. Isto ainda quer dizer que desejamos encontrar $f(a + h)$ em que $h$ é um *passo* de uma malha de aproximação.

Retomando a (eq. 3.17), como queremos evitar de calcular derivadas parciais[2], O Teorema do valor médio do cálculo diferencial nos traz uma sugestão para substituir as derivadas parciais de $g$ por um quociente com valores desta função nos extremos de um intervalo. O método de Runge-Kutta altera esta sugestão substituindo a diferença pelo valor da função calculada em um único ponto obtido obtido pela introdução de novos coeficientes e depois a imposição de uma igualdade que leve a uma equação linear, por comparação com (eq. 3.17). Vejamos como isto se faz:

$$f(a+h) \approx f(a) + \lambda_1 g(a,b)h + \lambda_2 h g(a+\mu_1 h, b + g(a,b)\mu_2 h) \quad (3.18)$$

$$\approx f(a) + \lambda_1 g(a,b)h + \lambda_2 h[g(a,b) + \frac{\partial g}{\partial t}\mu_1 h + \frac{\partial g}{\partial y} g(a,b)\mu_2 h] \quad (3.19)$$

$$\approx f(a) + (\lambda_1 + \lambda_2) g(a,b)h + (\lambda_2 \mu_1 \frac{\partial g}{\partial t} + \lambda_2 \mu_2 \frac{\partial g}{\partial y} g(a,b))h^2 \quad (3.20)$$

Vamos agora impor a condição de que esta última

[2] Se os dados forem experimentais, não poderemos mesmo calcular derivadas parciais, se a equação diferencial for dada por uma equação, algébrica, então a (eq. 3.17) pode ser usada.

$x(-5) = 49$ e duas poligonais obtidas com passos distintos:2,1. O método de Euler consiste, assim da fórmula de Taylor de primeiro grau.

Usamos apenas polinômios de Taylor de primeiro grau. Nisto se constitue o método de Euler. A garantia de aproximação que se tem neste método vem da fórmula de Taylor. O erro cometido é medido com o resto desta fórmula.

### 3.2.2 Métodos de Runge-Kutta.

Mas podemos insistir calculando agora $f''(a)$ que significa calcular a derivada de $g(t, y)$ em relação a $t$:

$$\frac{d^2y}{dt^2} = \frac{\partial g(x, y)}{\partial t} + \frac{\partial g(t, y)}{\partial y}\frac{dy}{dt} \tag{3.14}$$

que se explica pela regra da cadeia em que derivamos em relação a $t$ e $y$. Agora $\frac{dy}{dt}$ aparece na equação inicial o que nos permite de re-escrever a expressão como:

$$\frac{d^2y}{dt^2} = \frac{\partial g(t, y)}{\partial t} + \frac{\partial g(t, y)}{\partial y}g(t, y) \tag{3.15}$$

Temos o valor da segunda derivada em termos das derivadas pariciais da equção $g(t, y)$ *dada* e podemos retonar a fórmula de Taylor para escrevermos um polinômio do segundo grau em $h$:

$$f(a + h) \approx f(a) + f'(a)h + f''(a)\frac{h^2}{2} \tag{3.16}$$

$$f(a + h) \approx f(a) + g(a, b)h + (\frac{\partial g(t, y)}{\partial t} + \frac{\partial g(t, y)}{\partial y}g(t, y))\frac{h^2}{2} \tag{3.17}$$

Para valores muito pequenos de $h$ o erro desta fórmula do segundo grau é ainda menor do que a do primeiro

Vamos novamente *interpretar* $f(a+h)$. Significa o valor de uma função *desconhecida* no ponto $a+h$. Dependo dos objetivos que se tenha a *solução* do problema pode parar aqui: se o objetivo for:, obter um valor aproximado da função $f$ no ponto $a+h$; está feito!

**Exercício: 21** *1. Use o método de Euler para encontrar (a+h, f(a+h))* aproximadamente *em cada uma das equações abaixo, com h := 0.01:*

*(a)* $y' = sen(xy)$ ; $y(0) = 1$

*(b)* $y' = x^2 + y^2$ ; $y(0) = 1$

*(c)* $y' = -\frac{x}{y}$ ; $y(0) = 1$

*2. Considere uma sucessão* $x_0 = a, \ldots, x_n$ *formando uma p.a. de razão h:= 0.1 e construa uma poligonal que resolva aproximada- mente cada uma das equações anteriores com o método de Euler.*

O objetivo pode ser mais amplo, entretanto, do que apenas encontrar um ponto próximo da condição inicial: se pode desejar um gráfico *entenda-se: uma listagem fina de valores de f nas proximidades da condição inicial dada*, por que um *gráfico* no computador nada mais é que uma listagem de pontos com um passo muito pequeno que pode ser alternadamente apresentada como uma listagem de pares de números. Neste caso desejamos ter o conjunto:

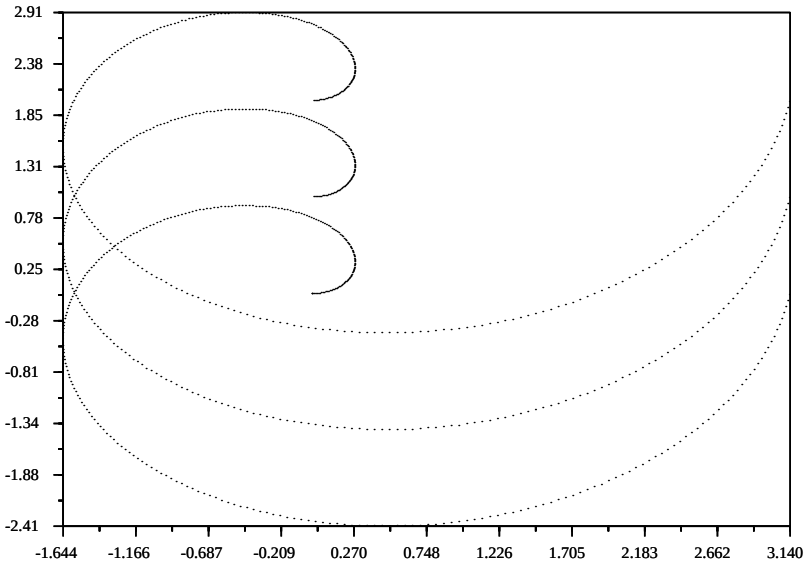
$$\{(a,b) = (x_0, y_0), (x_1, y_1), \ldots, (x_k, y_k), (x_{k+1}, y_{k+1})\} \tag{3.13}$$

Basta *iterar* o processo considerando

$$y_1 = f(a+h) = b + g(a,b)h = b_1$$

como uma nova condição inicial e assim calcular $y_2 = f(a+2h)$.

O próximo gráfico, (fig. 3.1) contém a solução exata da equação diferencial $\frac{dx}{dt} = -\frac{x}{t}$ com a condição inicial




$$f(a+h) = f(a) + f'(a)h = b + g(a,b)h \;\; \Rightarrow \;\; f(a+h) = b + g(a,b)h \tag{3.12}$$

Não se perca com a notação: queremos encontrar uma função $f$ que é solução da equação diferencial e sabemos que $f'(a) = g(a,b)$ ; $b = f(a)$. Estaremos sempre falando a função $f$ que é solução procurada[1].

[1] nem sempre a solução de uma equação diferencial é uma *função*, a maneira mais adequada de falar é chamá-la de *curva*, aqui há um erro suportável...

Nas expressões acima, $h$ representa um erro, o que em programação se chama as vezes *passo*. Podemos assim interpretar a fórmula (eq. 3.11), vendo à direita uma aproximação do valor de $f$ num ponto próximo de $\underline{a}$ sendo $\underline{h}$ o passo escolhido.

A expressão $g(t, y)$ na equacao 3.4 representa uma fórmu-la "algébrica" associando a um ponto do plano a derivada de uma curva que passa por ele: as equações diferenciais representam curvas ou planos definidos através de uma expressão em que as derivadas destes objetos geométri-cos entram envolvidas.

O gráfico (fig. 3.2) abaixo traz tres curvas, cada uma éuma solução particular de uma equação diferencial. Uma "*curva*" é a solução de um *problema*, é determinada por um ponto que caracteriza a condição inicial, observe que esta condição inicial tem o mesmo significado numa integral em que apenas um dos limites de integração está definido enquanto que o outro fica *variável*.

**Exercício: 20** *1. Resolva as seguintes equações diferenciais* exatamente *encontrando a solução identificada pela condição inicial dada:*

*(a)* $y' = 3x$ ; $y(0) = 0$ ;

*(b)* $y'' = 0$ ; $y(0) = 0$ ; $y'(0) = 0$ ;

*(c)* $y' = sen(x)$; $y(0) = 0$ ;

*2. Encontre* todas *as soluções das equações diferenciais do item anterior, (graficamente).*

### 3.2.1 O método de Euler.

Como usar 3.11 em conexão com a equação diferencial $\frac{dy}{dt} = g(t, y)$ ; $y(a) = b$ ? Vamos interpretar $f(a)$ como a condição inicial dada. Assim $x = a \Rightarrow y = b$ e portanto temos

em seu estudo as equações lineares.

O quinto é o mais interessante e tomará mais o nosso tempo porque ele é um método usado também para as *equações diferenciais parciais* e representa uma formulação diferente de um mesmo princípio de aproximação: *elementos finitos*. Para desenvolvê-lo começa-remos por discutir um método diferente de aproximação de funções que é o quadro geral em que se colocam os Splines: *a regularização por convolução* .

Embora citadas diversas vezes acima, as equações diferenciais parciais não serão estudadas neste texto.

A parte final destes capítulo vai se dedicar a um tipo particular de equações diferenciais: as lineares. Nestas aplicaremos o último método.

## 3.2 Métodos *passo a passo*.

Vamos começar estudando tres métodos, *na verdade tres variantes de um mesmo método porque estão todos baseados na fórmula de Taylor*. Nos desagrada a *etiqueta: passo a passo* colocada nestes métodos, porque, como toda etiqueta é restritiva e algumas vezes envelhece mais rápido que as técnicas envolvidas. Você verá algumas razões para esta crítica no texto. Como todos os métodos usam a Fórmula de Taylor, esta deveria ser a *etiqueta*.

Os polinómios de Taylor oferecem um método de apro-ximação de *objetos* diferenciáveis. Relembrando a fórmula:

$$f(a+h) = f(a) + f'(a)h + \cdots + \frac{f^{(n)}(a)}{n!}h^n + R_n(x) \tag{3.10}$$

em que $f$ é uma função suficientemente derivável. Se despresarmos o erro, o que significa substituir $f$ por uma aproximação sua, escreveremos:

$$f(a+h) \approx f(a) + f'(a)h + \cdots + \frac{f^{(n)}(a)}{n!}h^n \tag{3.11}$$

uma sonda, ou um satélite em órbita significa encontrar o *nível energético* preciso em que uma *partícula* vai ficar em torno de um *núcleo*, consideradas as respectivas massas e velocidade angular do satélite. Nestes termos se trata de encontrar uma curva que tangencie outra: solução de uma equação diferencial ordinária.

Este mesmo problema envolve um outro: controlar a rotação do satélite em torno do seu próprio eixo para garantir uma exposição maximal de suas células foto-elétricas à luz solar, por exemplo: *solução de uma equação diferencial parcial por que o satélite deixa de ser uma partícula e o problema deixa de ser univariado passando a multivariado.*

Temos assim que resolver equações diferenciais coisa que nem sempre conseguimos fazer formalmente, isto justifica folgadamente nos preocuparmos com métodos de solução aproximada, numérica, por que em muitos casos é tudo o que podemos fazer.

Vamos estudar cinco métodos de aproximação :

- Os tres primeiros se baseiam na fórmula de Taylor:
  - *Método de Euler*;
  - *Método das Isóclinas*;
  - *Método de Runge-Kutta.*

  O métode de Euler e o de Runge-Kutta são chamados de métodos passo porque podemos obter a solução da equação diferencial, aproximadamente no ponto $(x_k, y_k)$ com auxílio dos valores já obtidos anteriormente. O *método das Isóclinas* é um método global, porque, por exemplo, ignorando condições iniciais, procura descrever o comportamento das soluções no domínio dado. Ele permite algumas vezes um visualização do *fluxo* das curvas-solução.

- O quarto parte de uma aplicação iterativa do Teorema Fundamental do Cálculo é o método de Picard.

- O quinto se baseia nos Splines e se confina na classficação *método de interpolação polinomial.* Aqui nos restringiremos

$$\frac{\partial u}{\partial x} - \frac{\partial u}{\partial t} - 3t + 4x = 0.$$

Resolver equações diferenciais parciais é mais difícil que resolver equações diferenciais ordinárias, em princípio... aliás, *resolver equações diferenciais* de qualquer natureza, ordinárias ou parciais, em princípio não é fácil e nem siquer possivel. Há alguns tipos de equação sobre as quais se sabe tudo, mesmo que resolvé-las explicitamente nem sempre seja algo trivial. Algumas delas foram resolvidas ao longo de anos e somente aos poucos se ficou claro para os matemáticos eles ti-nham a sua solução como é o caso da equação de Laplace:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial t^2} = 0 \tag{3.8}$$

cujas soluções se chamam funções harmônicas e são a parte real ou a parte imaginária de uma função analítica. Estas últimas, por sua vez são as soluções de uma outra equação diferencial parcial que também levou dois séculos para que os matemáticos compreendensem que a havia resolvido indiretamente:

$$\text{Equações de Cauchy-Riemann.} \frac{\partial u}{\partial x} = \frac{\partial v}{\partial y} \frac{\partial u}{\partial y} = -\frac{\partial v}{\partial x} \tag{3.9}$$

que ainda podemos chamar de um sistema de equações diferenciais parciais de primeira ordem.

Podemos facilmente demonstrar as relações acima entre funções analíticas e funções harmônicas, mas representa um curso com duração de um semestre, dependendo, obviamente dos prerequesitos... em que se descrevem 200 anos de história da Matemática.

Seria interessante que fizessemos algumas referências con-cretas a situações em que se usam equações diferenciais. Uma área muito momentosa de aplicações é a espacial. Colocar

*trata-se de uma equação diferencial de segunda ordem, estas precisam de uma condição mútipla para sua solução, e neste caso se fala de condições de fronteira: o início e o fim do intervalo no tempo. O problema é então:*

$$F'' = f$$
$$F(0) = 0$$
$$F(1) = -1$$

*Um outro problema alternativo poderia ser:*

$$F'' = f$$
$$F(0) = 0$$
$$F'(0) = -1$$

*em que as duas condições são dadas no ponto* $t = 0$.

3. *Numa equação diferencial podemos ter expressões combinando operadores aritméticos e funções compondo uma expressão desde que aí apareçam as derivadas de uma função que se desconhece:*

$$3ty'' - 5t^2y' = 0$$

4. *No exemplo anterior os termos "*$3t$*" e "*$5t^2$*" são coeficientes da equa-ção , isto é* os dados que se combinam linearmente *com as variáveis. Aquela equação é portanto uma* equação a coeficientes variáveis *em oposição a equação :*

$$3y'' - 5y' = 0$$

*que é equação a* coeficientes constantes.

5. *Se na expressão aparecerem* derivadas parciais *se trata de uma equação a derivadas parciais, ou* equação diferencial parcial. *Em oposição , as outras em que não aparecem derivadas parciais, são chamadas* equações diferenciais ordinárias. *A equação seguinte é uma equação diferencial parcial:*

Este conjunto formado pela equação mais a *condição inicial* $y(a) = b$ recebe comumente o nome de *problema*. Faremos algumas vezes referências ao problema 3.6 para com isto indicar uma equação mais uma condição inicial. No gráfico acima temos tres soluções de uma equação diferencial, naturalmente associadas a tres condições iniciais distintas. A semi-reta que aparece no gráfico coincide com o ponto inicial $(0, 0)$ de uma dessas curvas, sendo-lhe tangente neste ponto, portanto o coeficiente angular da reta é $g(0, 0)$.

Esta maneira de falar tem um sentido geométrico fácil de explicar: a solução de uma equação diferencial é um *objeto geométrico*, aqui será sempre uma *curva*, que passa por um determinado ponto: $(a, b)$. Isto é o mesmo que a determinação duma primitiva, no Cálculo univariado:

$$F(t) = b \ + \int_a^t f(x)dx \tag{3.7}$$

em que o gráfico de $F$ passa no ponto $(a, b)$: Teorema Fundamental do Cálculo.

**Exemplo: 18** *Alguns exemplos de equação diferencial.*

1. *A equação diferencial acima se expressa como $F' = f$. Tais equações diferenciais são comuns num curso de Cálculo quando se diz que se deseja calcular a primitiva duma função .*

2. *Um pouco mais difícil seria a equação diferencial*

$$F'' = f$$

*que dizemos ser uma equação diferencial de segunda ordem por que em sua expressão se tem derivadas de segunda ordem. Para resolver explicitamente esta equação diferencial, precisamos de* duas condições iniciais*, poderia ser uma condição inicial,* o ponto inicial do intervalo do tempo *e uma condição final,* o ponto final do intervalo de tempo*:*

**Exercício: 19** *Teorema da função implicita.*

1. *Enuncie o Teorema da funcção implicita para funções* $\mathbf{R}^3 \mapsto \mathbf{R}$.

2. *Mostre que a hipótese acima enunciada é suficiente para que uma equação diferencial como foram definidas se possa escrever na forma explicita* $\frac{dy}{dt} = g(t, y)$.

Ao escolhermos escrever a "variável" independente com a letra $t$ estamos também tomando uma posição de indicar que nossas equações diferenciais serão *dependentes do tempo*. É uma atitude natural que vai de encontro a grande maioria das aplicações em que se expressa com uma equação diferencial uma relação de variação ao longo do tempo de algum fenômeno.

### 3.1.2 A solução de uma equação diferencial.

Resolver uma *equação diferencial ordinária* significa encontrar uma curva, algumas vezes o gráfico uma função , que satisfaça equação 3.6, no sentido de que a tangente, ou o coeficiente angular do vetor tangente à curva, ou ao gráfico, é dado pela expressão $g(t, y)$ em qualquer ponto $(t, y(t))$ da curva.

Isto é muito "vago" entretanto, para se chamar de uma solução . Precisamos de um ponto a partir do qual traçar a curva, de modo que o que se estuda são *problemas*. Um *problema* é um par de dados da forma:

Um problema.

$$\frac{dy}{dt} = g(t, y) \tag{3.5}$$

$$y(a) = b \tag{3.6}$$

em que $y(a) = b$ representa um ponto $(a, b)$ por onde passa a curva-solução . Este ponto é chamado apropriadamente de *condição inicial*.

*$cos(xy)$* seria uma expressão algébrica...*)* . *Ou seja, uma equação diferencial ordinária é uma equação em que se encontre envolvida,* como incognita*, uma função $t \mapsto y(t)$ assim como suas derivadas até uma certa ordem $n$, e dizemos então que se trata de uma* **equação diferencial ordinária de ordem** $n$ .

*De modo análogo se definiria uma* equação diferencial parcial *como*

$$F(t, x, u, \frac{\partial u}{\partial t}, \frac{\partial u}{\partial x}, \ldots, \frac{\partial^n u}{\partial t^p x^q}) = 0 \tag{3.2}$$

*que portanto é uma equação em que se encontra envolvida uma função multivariada assim como suas derivadas parciais até uma certa ordem.*

*Neste livro estaremos nos restringindo à discussão das equações diferenciais ordinárias de primeira ordem e segunda ordem. Entretanto o objetivo principal do livro não é o estudo das* equações diferenciais *e sim as técnicas de aproximação polinômial. As equações diferenciais entram aqui apenas como um desafio fascinante de aproximação polinomial. No final do capítulo nos estenderemos a solução aproximada de uma equação diferencial linear de ordem $n$.*

## 3.1 Equações Diferenciais.

### 3.1.1 Exemplos e notação.

$$F(t, y, y') = 0, \tag{3.3}$$

que pelo Teorema da Função Implícita podem ser escritas na forma:

$$\frac{dy}{dt} = g(t, y) \tag{3.4}$$

desde que $\frac{\partial F}{\partial y'} \neq 0$ o que suporemos verdadeiro, em uma certa região definida pelos parâmetros $t, y$.

# Capítulo 3

# Equações Diferenciais Ordinárias.

Introdução .

*Este livro não se dedica a equações diferenciais e portanto este capítulo apenas usa conhecimentos que você deve ter de equações diferenciais obtidos em Cálculo. Nosso objetivo é o método polinomial de aproximação de funções que discutimos no capítulo 1. As equações diferenciais são um exemplo de problema ao qual tentamos aplicar o método polinomial. Na primeira parte do capítulo discutiremos métodos clássicos de aproximação polinomial da solução de uma e.d.o. como* método de Euler, *e* os métodos do tipo Runge-Kuta *que se classificam como* métodos passo a passo. *Também discutiremos o* método das isóclinas *que é um* método global. *Depois nos dedicaremos aos* splines *e a solução aproximada de equações lineares de ordem* maior *em que se utiliza a teoria das matrizes estudadas no capítulo anterior.*

*Uma* equação diferencial ordinária *é uma relação do tipo*

$$F(t, y, y', \ldots, y^{(n)}) = 0 \tag{3.1}$$

*em que $F$ é uma expressão algébrica, (*algébrica está aqui tomado em seu sentido mais amplo, do século 19, em que $G(x, y) =$

```
Teste := False
                      Else inf[i][j] := aMat[i][j];
           j := j+1;
         End;
       inf[i][j] := 1;
       While j <= dim Do
         Begin
           sup[i][j] := aMat[i][j];
           For k := 1 to i-1 Do
sup[i][j] := sup[i][j] - inf[i][k]*sup[k][j];
           j := j+1;
           inf[i][j] := 0;
         End;
     End;
     Apeteco;ClrScr;
     WriteLn('Matriz triangular inferior:  ');
     PrintMatriz(inf,'matriz inferior');
     Apeteco; ClrScr;
     WriteLn('Matriz triangular superior:  ');
     PrintMatriz(sup,'matriz superior');
     Apeteco; ClrScr;
     Repeteco; Resposta := UpCase(Resposta);
 Until (Resposta = 'N');
End.
```

6. *Mostre que se* $\mathcal{A}$ *e* $\mathcal{B}$ *forem duas matrizes* n x n *então podemos decompô-las em sub-matrizes de ordem* jxj, jxk, kxj, kxk ; j+k=n *de modo que a multiplicação* $\mathcal{A} \cdot \mathcal{B}$ *pode ser efetuada usando as sub-matrizes como se fossem números. (Faça algumas experiências para encontrar a lei de formação do produto, introduza uma notação adequada para descrever esta questão ).*

O seguinte trecho de programa em Pascal representa o algoritmo para determinação das duas matrizes triangulares do Teorema 2.72:

```
{** programa para fatoracao de matrizes num produto
    de matrizes  triangulares.
    por Tarcisio Praciano Pereira   -  DMa - UEM  -
    Projeto: Programas em Pascal para Analise Numerica
-  1994     ***}

program FatorT;

Uses Tipos, Matrizes,Graph, Crt;

Var
     sup,inf : Matriz;

Begin
 Repeat
  projeto := ' Fatora'+cedi+atil+'  de Matrizes ';
  RestoreCrtMode; Mask;EntradaMatriz;
  WriteLn(' primeira linha da matriz inferior');
  For  i := 1 to dim   Do
    Begin
      sup[1][i] := aMat[1][i];
      inf[1][i] := 0;
    End;
  inf[1][1] := 1;
  WriteLn(' matrizes inferior e superior');
  For i:= 2 to dim   Do
     Begin
       WriteLn('Escrevendo a linha ', i);
       j := 1;
       While  j < i  Do
         Begin
           sup[i][j] := 0; inf[i][j] := aMat[i][j];
           For k := 1 to  j-1 Do
             inf[i][j] := inf[i][j] - inf[i][k]*sup[k][j];
           If sup[j][j] <> 0  Then
inf[i][j] := inf[i][j]/sup[j][j]
                  Else
                     If inf[i][j] <> 0  Then
```

A condição de inversibilidade de $\mathcal{A}$ é evidentemente neces-sária para a solução descrita acima que exige que $s_{ii} \neq 0$. Entretanto o sistema é*super-determinado: tem mais equações que incognitas* portanto, *se tiver solução* terá um infinidade delas. A *indeterminação* reside na possibilidade de $s_{ii}$ ser igual a zero o que implica em que $i_{ij} \neq 0$ qualquer. Isto mostra que o Teorema 2.72 vale mesmo no caso de matrizes singulares como está enunciado.

As equações tem que ser calculadas nesta ordem dentro dum programa, o programa abaixo faz isto.

**Exercício: 18** *1. Traduza as equações 2.72 para um programa em Pascal (ou C ou outra linguagem), veja abaixo parte do algoritmo em Pascal.*

2. *Faça um programa que teste uma matriz quanto a sua invertibilidade e caso seja inversível a triangularize.*

3. *Mostre que se uma matriz for obtida por permutações das linhas (ou colunas) da matriz identidade então :*

   (a) *seu determinante é* $\pm 1$.

   (b) *seus elementos serão apenas a unidade aparecendo uma única vez numa linha ou coluna mas em toda linha ou coluna há uma entrada não nula igual a 1.*

4. *Mostre que, recíprocamente, se em toda linha ou coluna houver uma entrada igual a unidade sendo as demais entradas nulas então esta matriz é obtida por permutação de linhas (ou colunas) da identidade.*

5. *Considere a sequência de números* $\alpha = (a_1, ...a_n)$ *todos diferentes e a matriz* $\mathcal{A}$ *cuja colunas sejam as potências*

$$\alpha^0, \alpha^1, ...\alpha^{n-1}, \alpha^n$$

*(entenda* $\alpha^k$ *como sendo a sequência das potências* $k$ *dos elementos de* $\alpha$ *). Calcule o determinante de* $\mathcal{A}$ *. (Estabeleça uma hipótese a partir de algumas experiências).*

superior $\mathcal{S}$ em que os elementos da diagonal: $s_{ii}$ memorizam o determinante de uma matriz $\mathcal{A}$ dada:

$$\mathcal{I} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ i_{21} & 1 & 0 & 0 \\ i_{31} & i_{32} & 1 & 0 \\ i_{41} & i_{42} & i_{43} & 1 \end{vmatrix} \quad \mathcal{S} = \begin{vmatrix} s_{11} & s_{12} & s_{13} & s_{14} \\ 0 & s_{22} & s_{23} & s_{24} \\ 0 & 0 & s_{33} & s_{34} \\ 0 & 0 & 0 & s_{44} \end{vmatrix} \quad ; \quad \mathcal{A} = \mathcal{I} \cdot \mathcal{S} \tag{2.70}$$

A equação do Teorema 12 gera um sistema de equações que nos permitem encontrar as matrizes $\mathcal{I}, \mathcal{S}$ uma vez que $\mathcal{A}$ é dada:

$$\begin{array}{ccccccccc}
s_{11} & = & a_{11} & & & & & & \\
s_{12} & = & a_{12} & & & & & & \\
s_{13} & = & a_{13} & & & & & & \\
s_{14} & = & a_{14} & & & & & & \\
i_{21}s_{11} & = & a_{21} & & & & & & \\
i_{21}s_{12} & + & s_{22} & = & a_{22} & & & & \\
i_{21}s_{13} & + & s_{23} & = & a_{23} & & & & \\
i_{21}s_{14} & + & s_{24} & = & a_{24} & & & & \\
i_{31}s_{11} & = & a_{31} & & & & & & \\
i_{31}s_{12} & + & i_{32}s_{22} & = & a_{32} & & & & \\
i_{31}s_{13} & + & i_{32}s_{23} & + & s_{33} & = & a_{33} & & \\
i_{31}s_{14} & + & i_{32}s_{24} & + & s_{34} & = & a_{34} & & \\
i_{41}s_{11} & = & a_{41} & & & & & & \\
i_{41}s_{12} & + & i_{42}s_{22} & = & a_{42} & & & & \\
i_{41}s_{13} & + & i_{42}s_{23} & + & s_{33} & = & a_{43} & & \\
i_{41}s_{14} & + & i_{42}s_{24} & + & i_{43} & + & s_{44} & = & a_{44}
\end{array} \tag{2.71}$$

Se a matriz $\mathcal{A}$ for inversível, a solução deste sistema, (agora pensando num sistema n x n), é da forma:

$$\begin{array}{ll}
\forall j = 1..n & s_{1j} = a_{1j} \\
j < i & i_{ij} = \dfrac{a_{ij} - \sum\limits_{k<j} i_{ik}s_{kj}}{s_{ii}} \\
j = i & i_{ij} = 1 \\
j > i & i_{ij} = 0 \\
j < i & s_{ij} = 0 \\
j \geq i & s_{ij} = a_{ij} - \sum\limits_{k<i} i_{ik}s_{kj}
\end{array} \tag{2.72}$$

$$\mathcal{E} \cdot \mathcal{A} \cdot \mathcal{F} \approx \begin{vmatrix} \mathcal{D} & 0 \\ 0 & 0 \end{vmatrix}$$

*em que $\mathcal{E}, \mathcal{F}$ sao matrizes triangulares tendo apenas a unidade sobre a diagonal principal e $\mathcal{D}$ é uma matriz diagonal de ordem $k$ semelhante do ponto de vista de sistema de equacões a matriz menor de ordem $k$ inversível de $\mathcal{A}$, em que $k$ é o posto de $\mathcal{A}$.*

Podemos escrever este resultado num outro formato. Verifique que nas igualdades abaixo se tem a fatoração de $\mathcal{A}$ como matrizes triangulares:

$$\mathcal{A} = \mathcal{E}' \cdot (\mathcal{D} \cdot \mathcal{F}') = (\mathcal{E}' \cdot \mathcal{D}) \cdot \mathcal{F}'$$

em que à esquerda temos a matriz triangular superior $\mathcal{E}'$ apenas com a unidade na diagonal multiplicada pela matriz triangular inferior $\mathcal{D} \cdot \mathcal{F}'$.

Podemos daí enunciar dois teoremas semelhantes um dos quais é:

**Teorema: 12** *Uma matriz quadrada $\mathcal{A}$ pode ser fatorada num produto de uma matriz triangular inferior por uma matriz triangular superior:*

$$\mathcal{A} = \mathcal{I} \cdot \mathcal{S}$$

*de modo que os elementos na diagonal de $\mathcal{I}$ sejam apenas a unidade e a diagonal de $\mathcal{S}$ memoriza* $\det \mathcal{A}$.

Vamos representar matricialmente o Teorema 12 para deduzir um método de resolução de sistema de equações lineares. Vamos nos fixar em matrizes 4x4 para facilitar a representação esquemática das matrizes. Estaremos ao mesmo tempo estabelecendo o formalismo necessário ao programa FatorT que executa a tarefa expressa no Teorema 12.

Queremos encontrar uma matriz triangular inferior $\mathcal{I}$ apenas com a unidade na diagonal, uma matriz triangular

Exatamente como se anulam elementos numa matriz, podemos anular os blocos $B', C'$ obtendo a matriz-blocos:

$$\mathcal{A} \approx \begin{vmatrix} \mathcal{D} & 0 \\ 0 & G" \end{vmatrix}$$

Vemos assim que tudo se reduz a discutir o que acontece com uma matriz não inversível $G"$ cuja ordem é o complementar relativamente a $n$ da ordem de $\mathcal{D}$ .

**Observação: 27** *Parte singular de uma matriz.*

$G"$ representa a *parte singular* da matriz $\mathcal{A}$. Assim um sistema pode ser *fatorado* de modo a evidenciar a sua *parte singular* e a sua *parte não singular*.

A parte não singular representa a informação únivoca do sistema: cada variável <u>tem</u> uma e uma única solução . Na parte singular, se houver soluções elas não são únicas.

A parte singular de um sistema corresponde a uma equação numérica da forma $0x = b$ que tem solução apenas se $b = 0$ e neste caso tem um número infinito de soluções : qualquer número real é uma solução .

Como a matriz $G"$ pode ser triangularizada superiormente, vemos que ela não pode ter entradas não nulas sobre a diagonal, (do contrário por trocas de linhas e colunas poderiamos deixar uma tal entrada no canto superior esquerdo de $G"$ e depois agregar mais uma linha e uma coluna à matriz $\mathcal{D}$ o que contradiz a hipótese de esta era uma matriz menor de maior ordem inversível. Por razão semelhante $G"$ não pode ter nenhuma entrada não nula acima diagonal tão pouco, isto é $G" \equiv 0$, assim:

$$\mathcal{A} \approx \begin{vmatrix} \mathcal{D} & 0 \\ 0 & 0 \end{vmatrix}$$

As matrizes triangulares superior e inferior na fatoração de $\mathcal{A}$ têm apenas 1 sobre a diagonal. Demonstramos assim o

**Teorema: 11** *(da fatoração triangular)*

*Seja $\mathcal{A}$ uma matriz de ordem $n$ tendo uma matriz menor de ordem $k$ inversível. Então*

**Observação: 26** O anel das matriz triangulares superiores.

O conjunto das matrizes triangulares superiores é estável sob as operações usuais com matrizes: soma de matrizes, multiplicação de uma matriz por um escalar ou produto de matrizes. Este conjunto tem assim uma estrutura algébrica, é um anel.

Como tais matrizes podem ter elementos nulos na diagonal, nem todas são inversíveis.

Podemos mostrar, é o conteúdo do Teorema 7, que o conjunto de todas as matrizes se encontra representado neste conjunto das matrizes triangulares superiores.

O mesmo se pode dizer do conjunto das matrizes triangualres inferiores.

**Definição : 12** *Pivots da matriz $\mathcal{A}$.*

*Suponha que uma matriz $\mathcal{A}$ tem uma fatoração*

$$\mathcal{A} = \mathcal{E}' \cdot \mathcal{D} \cdot \mathcal{F}'$$

*triangular como descrito no teorema 10. Então os elementos da diagonal de $\mathcal{D}$ se chamam* pivots *de $\mathcal{A}$.*

Vamos agora extender o Teorema 10. Antes de fazê-lo observemos que ele representa uma forma *fraca* do Teorema sobre inversão de matrizes, uma vez que não obtivemos a identidade, mas sim uma matriz diagonal. Para extender o teorema vamos retirar a hipótese de inversibilidade de $\mathcal{A}$.

Se $\mathcal{A}$ não for inversível mesmo assim poderá ter alguma *matriz menor* que o seja, basta que não seja nula e haverá pelo menos uma menor de ordem 1 x 1 que seja inversível. Vamos considerar a maior, (de maior ordem), das *menores não inversíveis* de $\mathcal{A}$. Por troca de linhas e colunas esta matriz pode ser colocada como um bloco no canto superior esquerdo de $\mathcal{A}$. Pelo Teorema 10 podemos escrever:

$$\mathcal{A} = \begin{vmatrix} \mathcal{EDF} & C \\ B & G \end{vmatrix}$$

em que $\mathcal{EDF}$ é um produto de blocos de acordo com o Teorema 10 por que alí temos uma matriz menor inversível. Ou seja temos a seguinte *equivalência de matrizes de sistemas de equação* :

$$\mathcal{A} \approx \begin{vmatrix} \mathcal{D} & C' \\ B' & G' \end{vmatrix}$$

em que o bloco $\mathcal{D}$ é o bloco diagonal escrito anteriormente com auxílio do Teorema 10.

Chamemos $\mathcal{F}$ a matriz triangular superior que *memoriza* estas operações , compondo com o que já obtivemos acima temos agora:

$$\mathcal{E} \cdot \mathcal{A} = \mathcal{S} \quad \Rightarrow \quad (\mathcal{E} \cdot \mathcal{A}) \cdot \mathcal{F} = \mathcal{S} \cdot \mathcal{F} = \mathcal{D} \tag{2.69}$$

resultando na matriz diagonal $\mathcal{D}$, pelo lema. $\boxed{\text{q.e.d.}}$

Observe ainda que os elementos da diagonal da matriz $\mathcal{D}$ no teorema 9 são únicos a menos de uma constante: um sistema de equação pode ter uma de suas equações multiplicada por um mesmo número sem que isto altere o resultado da equação , (o sistema de equação assim modificado é equivalente ao anterior). Podemos escolher a constante que multiplica cada equação de tal modo que as entradas nas diagonais de $\mathcal{E}, \mathcal{F}$ sejam apenas 1. Desta forma a matriz $\mathcal{D}$ fica bem determinada. Mais do que isto, pelo teorema 8, página 103, temos:

$$det(\mathcal{D}) = \det((\mathcal{E} \cdot \mathcal{A}) \cdot \mathcal{F}) = 1 \cdot \det(\mathcal{A}) \cdot 1 = det(\mathcal{A})$$

Os números que aparecem na diagonal de $\mathcal{D}$ se chamam *pivots* , e podemos garantir que são todos diferentes de zero devido a hipótese de inversibilidade de $\mathcal{A}$, além do mais o produto deles é $det(\mathcal{A})$.

Todas as matrizes envolvidas na expressão

$$(\mathcal{E} \cdot \mathcal{A}) \cdot \mathcal{F} = \mathcal{D} = \mathcal{E}(\mathcal{A}\mathcal{F}) = \mathcal{E}\mathcal{A}\mathcal{F}$$

são inversíveis, *(porque o determinante do produto é o produto dos determinantes)*. A inversa de uma matriz triangular superior (*inferior*) é também uma matriz triangular superior (*inferior*).

Demonstramos assim um caso particular de fatoração triangular de matrizes:

$\boxed{\textbf{Teorema:}}$ **10** *(Fatoração triangular de uma matriz.)*

*Se $\mathcal{A}$ for inversível então existe uma matriz triangular inferior, $\mathcal{E}$ e uma matriz triangular superior $\mathcal{F}$ tal que*

$$\mathcal{E} \cdot \mathcal{A} \cdot \mathcal{F} = \mathcal{D} \quad \Rightarrow \quad \mathcal{A} = \mathcal{E}' \cdot \mathcal{D}\mathcal{F}'$$

*em que $\mathcal{D}$ é uma matriz diagonal. Os elementos nas diagonais de $\mathcal{E}, \mathcal{F}$ sendo todos 1.*

## 2.7 O método de Gauss-Jordan.

Vamos discutir um exemplo que veremos não ser nada particular assim. Consideremos uma matriz $\mathcal{A}$ . Pelo Teorema 7 podemos encontrar uma matriz $\mathcal{E}$ inversível tal que $\mathcal{E} \cdot \mathcal{A} = \mathcal{S}$ uma matriz triangular superior.

Se supusermos além do mais que $\mathcal{A}$ seja inversível todos os elementos na diagonal de $\mathcal{S}$ serão diferentes de zero: o suficiente para anularmos todos os elementos da mesma linha que lhe estejam a direita repetindo o algorítimo contido no Teorema 7 adaptado a anulação dos elementos das linhas, em vez de anular os elementos das colunas. Estas novas operações serão as *duais* relativamente aquelas que fizemos na demontração do Teorema 7, quer dizer, o que fizemos com as linhas, agora faremos com as colunas. Em particular resulta disto que a matriz que irá *memorizar* as operações feitas, $\mathcal{F}$, será uma matriz triangular superior. Vamos registrar sob a forma de teorema este resultado dual:

**Teorema:** **9** *Triangularização inferior de matrizes. Toda matriz pode ser triangularizada inferiormente e a matriz de transformação $\mathcal{F}$ é uma matriz triangular superior inversível.*

**Dem**:

*Lema:* **1** *A triangularização inferior de uma matriz triangular superior é uma matriz diagonal.*

**Dem**: Temos uma matriz $\mathcal{S}$ que é triangular superior, quer dizer que tem apenas zero abaixo da diagonal principal. A primeira operação do *processo de triangularização inferior* consiste em combinar linearmente a primeira coluna com todas as demais para anular-lhes os elementos da primeira linha o que não altera os elementos abaixo da diagonal da primeira coluna que são todos zero. Mas este primeiro elemento da primeira coluna pode ser zero. Se for, passaremos à primeira coluna que tenha o primeiro elemento não nulo para com ela anular todos os demais elementos da primeira linha e consequentemente a operação de anulação dos elementos da primeira linha não alteram os elementos nulos abaixo da diagonal. O processo se repetirá sucessivamente para anular todos os elementos da segunda linha à direita da diagonal principal, que analogamente não irão alterar os elementos abaixo da diagonal principal. Ao final de $\frac{n(n+1)}{2}$ operações, no máximo, teremos anulado todos os elementos à direita da diagonal principal, sem ter alterado os elementos abaixo desta diagonal. Teremos portanto uma matriz triangular superior que é também é triangular inferior, logo diagonal. q.e.d.

*use o programa anterior para levar este elemento não nulo para a diagonal.*

3. *Melhore o programa anterior para que só leve o elemento para diagonal se ele estiver abaixo da diagonal*

4. *Transforme os programas acima em procedures e crie um gerenciador para as mesmas.*

5. *Inclua no seu "pacote" uma procedure AnulaSubDiagonal que anule os elementos de cada coluna que se encontrem abaixo da diagonal, se na diagonal o elemento não for nulo.*

6. *Construa uma procedure* DeterminanteDiagonal *que calcule o determinante de uma matriz Triangular.*

O programa *sistemas* contém uma solução para os exercícios anteriores. Nele você pode encontrar as procedures *EntradaMatriz, SupTriangulariza, DeterminanteTriangSup, etc....*

**Observação: 25** Processamento paralelo.

Aparentemente o que se pode fazer para tornar mais eficiente a solução de equações é dividí-las em blocos e aplicar programação paralela, entretanto a entrada de dados continuará sendo o gargalo do processo... Na última seção você verá que uma matriz pode ser fatorada e divida em blocos o que inclue blocos nulos. A presença de blocos nulos representa independência de entre conjunto de variáveis. Na linguagem de espaços vetoriais, significa que o sistema se refere a *sub-espaços linearmente independentes*. Esta *independência linear* é que torna possivel o uso de processamento paralelo.

Isto marca o gargalo referido acima. Sópodemos aplicar processamento paralelo a um sistema depois de modificá-lo: bloco-diagonalizá-lo para que surjam os sub-espaços linearmente independentes relativamente ao sistema considerado.

Em programas que gerem sistemas de equações ou que façam leitura automática de dados que produzam matrizes, evidentemente que não existe o *gargalo* da entrada de dados.

A próxima seção contem algumas ferramentas para aplicar estas ideias, entretanto não apresentaremos nada diretamente envolvido com processamento paralelo por as linguagens de processamento que usamos não possuem este conceito. Processamento paralelo, neste livro, é apenas uma curiosidade.

O programa **sistema** não vai tão longe. Ele apenas analisa se há uma solução única, e a calcula ou sendo o Determinante zero verifica se há impossibilidade de solução e o diz. Finalmente se o sistema tiver muitas soluções encontra uma que tenha algumas coordenadas nulas.

O planejamento de um tal programa pode ser o seguinte:

**Exemplo: 17**

```
program sistema;
{**** corpo do programa  *******}
Begin
  EntradaMatriz;
  SupTriangulariza;
  Writeln('Determinante: ',DeterminanteTriangSup(dim,aMat));
  Write('Matriz triangular superior:',Chr(25));
  Write('Matriz modificada de dados:   ',Chr(25));
  PrintMatriz;
  If Teste then
    Begin
     WriteLn(' a solucao eh unica.');
     Resolve;
    End
     Else
       SolucaoPossivel;
  Read;
End.
```

Este é o corpo do programa **sistema** que se encontra no disco. Rode o programa para ver como ele funciona, *na qualidade de usuário...* e depois tente resolver os exercícios abaixo. O programa *Sistemas* se encontra no disco. Se precisar consulte-o. Depois complete o projeto exposto acima, o resultado pode ser incluido no seu pacote GaPlana... Os exercícios abaixo podem lhe servir como roteiro.

**Exercício: 17** *Solução de sistemas lineares.*

*O programa* Sistema *resolve todas estas questões.*

1. *Faça o programa* PermutaLinha *que leia uma matriz e depois troque duas linhas da mesma e volte a imprí-la.*

2. *Faça um programa* ProcuraNaoNulo *que leia uma matriz, identifique em cada linha um elemento não nulo e depois*

A rotina *DeterminanteTriangSup* tem como função verificar se o determinante da matriz triangularizada é nulo ou não para decidir se chama *Resolve* ou *SolucaoPossivel*.

A pergunta: toda matriz pode ser triangularizada? tem uma resposta positiva no Teorema 7.

A primeira tentativa na produção de um programa para resolver equações lineares pode recair na produção de uma rotina que calcule determinantes. Com uma linguagem recursiva isto se torna muito fácil, Pascal não é uma linguagem *verdadeiramente recursiva*... como é o caso de LISP.

Mas calcular determinantes, com linguagens recursivas ou não é sempre muito demorado. Usando-se recursão os programas ficam mais bonitos entretanto a memória da máquina ficará muito mais densamente ocupada o que pode até tornar inviáveis os cálculos.

Calcular determinantes em toda sua generaliada não é essencial e nem siquer decisivo. Na próxima seção você verá como eles podem ser calculados indiretamente.

Um dos casos mais interessantes, a respeito de determinantes de sistemas é aquele em que o determinante se anula... e neste caso o melhor caminho é o da triangulação *superior* ou *inferior* da matriz. Para isto se opera sucessivamente sobre as linhas ou colunas.

Se a matriz for *singular*, isto é , não tiver inversa, portanto com determinante nulo, se vai ter pelo menos uma ultima linha totalmente nula, e o programa pode reorganizar a matriz para que esta linha seja a ultima linha. Se a alguma linha nula corresponder uma linha não nula na matriz de dados, a conclusão é de que o sistema não tem solução.

O programa deve portanto ter uma procedure que troque linhas e analise quando os elementos de uma coluna são zeros

Se o determinante for zero, mas houver solução , elas serão os elementos de uma reta, de um plano etc... Então o que se deve fazer é determinar a dimensão da solução.

1. *Aplique o programa* Sistema *abaixo em uma grande variedade de sistemas de equações .*

2. *Modifique o programa* Sistema *para que as matrizes obtidas em cada operção-linha seja impressa. Este foi o método usado para construir o exemplo desenvolvido acima.*

3. *Considere uma matriz diagonal,* todos seus elementos são nulos exceto os que se encontram sobre a diagonal principal, eventualmente....

   - *Verifique que num sistema de equações com uma tal matriz, se tem apenas:*

     $$a_{ii}x_i = b_i.$$

   - *Encontre a solução sob a hipóse:* $\forall i \;\; a_{ii} \neq 0$.
   - *Pode haver solução sem a hipótese anterior?*

### 2.6.1 O programa *Sistema*.

A solução de um sistema de equações pelo método de substituição encontra um meio natural numa linguagem de programação , em outras palavras ele é essencialmente algoritmico o que justifica seus 150 anos impávidos de existência.

O programa *sistema* representa, em Pascal, o método de eliminação de Gauss. Este programa se constitue de um conjunto de procedures que executam as tarefas *elementares* do processo, como trocar linhas, anular e homogeneisar elementos, etc... Abaixo se encontra a lista das *procedures e functions* de que se compõe *sistema*:

```
Procedure EntradaMatriz;
Function  LocalizaNaoZeroSubDiagonal(j:Integer):Boolean;
Procedure PermutaLinha(linha1,linha2:integer);
Procedure HomogeneisaAnula(i,j:integer;bMat:Matriz);
Function  DeterminanteTriangSup(dim:Integer;bMat:Matriz):Real;
Procedure PrintMatriz;
Procedure SupTriangulariza;
Procedure Resolve;
Procedure SolucaoPossivel;
{***  rotina  principal *****}
```

**Observação: 23** *A matriz de transformação .*

*A matriz de transformação 2.67 é uma matriz triangular inferior, todos os elementos acima da diagonal principal são nulos.*

**Observação: 24** *Complexidade do algoritmo.*

Uma outra forma de demonstrar este teorema é construtiva: produzindo a triangularização numa matriz formal, há um máximo de $m \leq \frac{(n+1)n}{2}$ operações inversíveis para serem executadas numa matriz $nxn$. Portanto a matriz inicial pode ser recuperada por m operações inversas: Simbolicamente

$$E = P_1 \cdots P_m$$

que é o produto de $m$ matrizes inversíveis.

Este número máximo de operações que se pode ter que fazer para que um algoritmo seja executado, se chama de de *complexidade do algoritmo*, no caso de triangularização superior de uma matriz o número é $\frac{(n+1)n}{2}$, muito menor que a complexidade do cálculo do determinante de uma matriz que é $n!$. Como podemos usar a triangularização para indiretamente calcular determinantes, o uso destes últimos fica assim recuperado.

Vamos enunciar um teorema que completa as ideias ventiladas acima, sem demonstração. Sua demonstração poderá ser encontrada num livro de Álgebra, ver [7].

**Teorema: 8** *Determinante do produto.* $det(\mathcal{AB}) = det(\mathcal{A})det(\mathcal{B})$

Com o teorema 8 podemos calcular facilmente o determinante de $\mathcal{M}$ :

$$det(\mathcal{E} \cdot \mathcal{M} = \det(\mathcal{S}) = det(\mathcal{E})det(\mathcal{M})$$

como $\mathcal{S}$ é triangular superior, seu determinante é o produto dos termos sobre a diagonal principal. Dividindo-se pelo $det(\mathcal{E})$ se tem $det(\mathcal{M})$. A complexidade deste algoritmo é

$$\frac{n(n+1)}{2} + n \approx \frac{n(n+1)}{2} \approx \frac{n^2}{2}$$

[14]

**Exercício: 16** *Solução automática de equações.*

[14] a complexidade é dada por pela sucessão simples da classe de equivalência a que pertença a sucessão que traduz o número de operações *elementares* do algoritmo.

*Resolve-se este sistema de traz para frente. A última linha da matriz produz* $-63w = 87$ *de onde se tira que* $w = -\frac{29}{21}$ *. A penúltima linha nos dá:*

$$-6z + 3w = -7$$

*da qual por substituição se conclue que* $z = \frac{10}{21}$*. Da segunda linha vem*

$$-3y - 3z + 3w = -5$$

*e portanto que* $y = \frac{4}{21}$*. Da primeira linha se tem*

$$x + 2y + 3z - w = 2$$

*portanto* $x = -\frac{25}{21}$

**Teorema: 7** *Triangularização superior de matrizes.*

*Toda matriz pode ser triangularizada superiormente e a matriz de transformação é uma matriz triangular inferior inversível.*

**Dem**: Que toda matriz pode ser triangularizada equivale a dizer que a equação $sa + tb = 0$ sempre tem solução em que $a$ é a entrada sobre a diagonal em uma determinada coluna. Se $a \neq 0$ a equação tem solução única. Se $a = 0$ basta escolher $t = 0$. Se houver alguma entrada diferente de zero na coluna ela pode ser transferida para a diagonal por troca de linhas. Isto mostra que toda matriz pode ser triangularizada. Resta-nos mostrar que as operações efetuadas são resultantes de produto de matrizes inversíveis, para garantir a inversibilidade da matriz $E$ 2.67. Acima descrevemos as *operações* elementares que produzem a triangularização :

1. Combinação linear de duas linhas em substituição a uma das linhas.
2. Troca de linhas.

É suficiente mostrar que estas operações são inversíveis e representáveis como um produto de matrizes. A troca de linha na matriz identidade produz uma matriz com determinante diferente de zero, portanto inversível. O produto desta matriz à esquerda por uma matriz $T$ qualquer produz a mesma troca de linhas na matriz T. A operação

$$sL_i + tL_j \to L_i$$

de combinação linear não trivial das linhas $L_i, L_j$ com os coeficientes $s, t \neq 0$ e substituição da linha $L_i$ por esta combinação na matriz identidade produz uma matriz cujo determinante será $s$. Como $s \neq 0$ esta matriz, que é também a matriz da trasnsformação , é inversível. q.e.d.

**Sexto Passo:**

$$\left|\begin{array}{cccc} 1 & 2 & 3 & -1 \\ 0 & -3 & -3 & 3 \\ 0 & 0 & -6 & 3 \\ 0 & 0 & 0 & -63 \end{array}\right| \left|\begin{array}{c} 2 \\ -5 \\ -7 \\ 87 \end{array}\right| \left|\begin{array}{cccc} 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 1 & -5 & -3 & 0 \\ 3 & -15 & -27 & -18 \end{array}\right| \tag{2.66}$$

*Anulamos todos os elementos que se encontravam abaixo da diagonal principal: triangularizamos superiormente a matriz.*

*A matriz:*

$$\mathcal{E} = \left|\begin{array}{cccc} 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 1 & -5 & -3 & 0 \\ 3 & -15 & -27 & -18 \end{array}\right| \tag{2.67}$$

*memoriza o conjunto das operações efetuadas. Se aplicarmos esta matriz à matriz incial, o resultado e a nova matriz dos coeficientes já triangularizada:*

$$\mathcal{E} \cdot \mathcal{M} = \left|\begin{array}{cccc} 1 & 2 & 3 & -1 \\ 0 & -3 & -3 & 3 \\ 0 & 0 & -6 & 3 \\ 0 & 0 & 0 & -63 \end{array}\right| \tag{2.68}$$

*Matricialmente, melhor, algebricamente, fizemos:*

$$\mathcal{M} \cdot x = b \;\Rightarrow\; \mathcal{E} \cdot \mathcal{M} \cdot x = \mathcal{E} \cdot b,$$

*multiplicamos toda a equação pela matriz inversível* $\mathcal{E}$ *. Agora* $\mathcal{E} \cdot \mathcal{M}$ *sendo uma matriz triangular o sistema* $\mathcal{E} \cdot \mathcal{M} \cdot x = \mathcal{E} \cdot b$ *é fácil de resolver e é equivalente ao primeiro* **do ponto de vista de solução do sistema de equações** *. A matriz de transformação* $\mathcal{E}$ *ser inversível é condição* **sine qua non** *para que se tenha a equivalência dos dois sistemas.*

$$\left|\begin{array}{cccc}1 & 2 & 3 & -1\\ 0 & -3 & -3 & 3\\ 0 & 5 & 3 & -4\\ -2 & 1 & -4 & 2\end{array}\right|\left|\begin{array}{c}2\\ -5\\ 6\\ -4\end{array}\right|\left|\begin{array}{cccc}1 & 0 & 0 & 0\\ -1 & -1 & 0 & 0\\ 2 & 0 & -1 & 0\\ 0 & 0 & 0 & 1\end{array}\right| \tag{2.62}$$

**Terceiro passo:** *Multiplicar a primeira linha por -2 e a quarta por -1, somá-las colocando o resultado na quarta linha:*

$$\left|\begin{array}{cccc}1 & 2 & 3 & -1\\ 0 & -3 & -3 & 3\\ 0 & 5 & 3 & -4\\ 0 & -5 & -2 & 0\end{array}\right|\left|\begin{array}{c}2\\ -5\\ 6\\ 0\end{array}\right|\left|\begin{array}{cccc}1 & 0 & 0 & 0\\ -1 & -1 & 0 & 0\\ 2 & 0 & -1 & 0\\ -2 & 0 & 0 & -1\end{array}\right| \tag{2.63}$$

*Conseguimos anular todos os elementos da primeira coluna exceto o primeiro elemento. As operações também se encontram registradas na última matriz. Continuaremos com o mesmo processo para anular todos os elementos da segunda coluna, exceto o primeiro e o segundo,* todos que se encontrem abaixo da diagonal principal.

**Quarto passo:** *Multiplicando a segunda linha por 5 e a terceira por -3, somando-as e colocando o resultado em lugar da terceira linha:*

$$\left|\begin{array}{cccc}1 & 2 & 3 & -1\\ 0 & -3 & -3 & 3\\ 0 & 0 & -6 & 3\\ 0 & -5 & -2 & 0\end{array}\right|\left|\begin{array}{c}2\\ -5\\ -7\\ 0\end{array}\right|\left|\begin{array}{cccc}1 & 0 & 0 & 0\\ -1 & -1 & 0 & 0\\ 1 & -5 & -3 & 0\\ -2 & 0 & 0 & -1\end{array}\right| \tag{2.64}$$

**Quinto passo:**

$$\left|\begin{array}{cccc}1 & 2 & 3 & -1\\ 0 & -3 & -3 & 3\\ 0 & 0 & -6 & 3\\ 0 & 0 & 9 & -15\end{array}\right|\left|\begin{array}{c}2\\ -5\\ -7\\ 25\end{array}\right|\left|\begin{array}{cccc}1 & 0 & 0 & 0\\ -1 & -1 & 0 & 0\\ 1 & -5 & -3 & 0\\ -1 & 5 & 0 & 3\end{array}\right| \tag{2.65}$$

*Considere o sistema:*

$$\begin{array}{rcl} x+2y+3z-w & = & 2 \\ -x+y+2w & = & 3 \\ 2x-y+3z+2w & = & -2 \\ -2x+y-4z+2w & = & -4 \end{array} \tag{2.59}$$

*que pode ainda ser representado pelo produto de matrizes:*

$$\left|\begin{array}{cccc} 1 & 2 & 3 & -1 \\ -1 & 1 & 0 & 2 \\ 2 & -1 & 3 & 2 \\ -2 & 1 & -4 & 2 \end{array}\right| \left|\begin{array}{c} x \\ y \\ z \\ w \end{array}\right| \left|\begin{array}{c} 2 \\ 3 \\ -2 \\ -4 \end{array}\right| \left|\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right| \tag{2.60}$$

*acrescentamos a matriz identidade ao final, todas as operações serão feitas simultaneamente na matriz dos coeficientes, na matriz dos dados e na matriz identidade. Em consequência ficará registrado na matriz identidade modificada o conjunto das operações efetuadas nas matrizes: a matriz resultante é a composição de todas as operações - linha efetuadas.*

**Primeiro passo:** *Multiplicar a primeira linha por -1, a segunda pro -1 e somá-las colocando o resultado na segunda linha:*

$$\left|\begin{array}{cccc} 1 & 2 & 3 & -1 \\ 0 & -3 & -3 & 3 \\ 2 & -1 & 3 & 2 \\ -2 & 1 & -4 & 2 \end{array}\right| \left|\begin{array}{c} 2 \\ -5 \\ -2 \\ -4 \end{array}\right| \left|\begin{array}{cccc} 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array}\right| \tag{2.61}$$

*Observe que omitimos a matriz das "variáveis" que aqui não faria nenhum sentido, elas apenas marcam a posição dos dados.*

**Segundo passo:** *Multiplicar a primeira linha por 2 e a terceira por -1, somar colocando o resultado na terceira linha:*

como $x_n$ é conhecido tem-se

$$x_{n-1} := \frac{b_{n-1} - a_{n-1n}x_n}{a_{n-1n-1}} \tag{2.57}$$

se $a_{n-1n-1} \neq 0$. Se $a_{n-1n-1} = 0$ só é possivel haver solução se $b_{n-1} - a_{n-1n}x_n = 0$ também, e neste caso $x_{n-1}$ é qualquer, ( o *programa* pode atribuir qualquer valor a variável para encontrar uma solução particular).

O mesmo raciocínio continua a se produzir com relação as demais linhas:

$$x_i := \frac{b_i - \sum_{k=i}^{n} a_{1k}x_k}{a_{ii}} \quad \Rightarrow \quad a_{ii} \neq 0 \tag{2.58}$$

porque $x_k$ são todos conhecidos desde $x_n$ até $x_k; k = i$. Se $a_{ii} = 0$ o sistema só terá solução caso $b_i - \sum_{k=2}^{n} a_{1k}x_k = 0$ e se for assim o *programa* pode atribuir qualquer valor a $x_i$ para encontrar uma solução particular.

Método de Gauss.

Este é o método da eliminação sucessiva de Gauss aplicado a um sistema de equação já triangularizado. Portanto, em síntese, para resolver um sistema pelo método de Gauss temos que:

1. Triangularizar a matriz do sistema.

2. Resolver o sistema por substituição a partir da última equação . Sempre analisando se $a_{ii} = 0$

Vejamos um exemplo de solução de um sistema de equações por este método.

**Exemplo: 16** *Método de eliminação de Gauss.*

tidade de elementos nulos. Zero é um número muito rápido de ser processado por que ocupa um byte, todos os outros números, *pequenos ou grandes* ocupam mais bytes e tomam mais tempo de processamento. Assim, matrizes esparsas, que contem uma grande quantidade de zeros são interessantes pela rapidez com que serão processadas.

Método de eliminação .

As matrizes triangularizadas são um exemplo de *matriz esparsa* equivalente, do ponto de vista de sistemas de equações à matriz dada. As matrizes mais esparsas que é possivel obter são as matrizes diagonais, nelas se reduziu ao mínimo possivel a *redundância do sistema: elas representam uma transformação linear numa base de vetores ortogonais.* Já sabemos, que por um defeito intrínseco do corpo $\mathbf{R}$, nem sempre é possivel diagonalizar matrizes reais. Mas sempre é possivel triangularizá-las.

O método de Gauss consiste na eliminação sucessiva das incognitas até que se tenha uma equação com uma única incógnita para resolver. Suponhamos que a triangulação superior é aquela em se anulam todos os elementos inferiores a diagonal principal. A última linha do sistema se reduz a:

$$a_{nn}x_n = b_n. \tag{2.54}$$

Assim, se $a_{nn} \neq 0$ então

$$x_n := \frac{b_n}{a_{nn}}, \tag{2.55}$$

Se $a_{nn} = 0$ então o sistema só terá solução se $b_n = 0$ e neste caso *programa* pode atribuir qualquer valor a $x_n$ para encontrar uma solução particular. A linha imediatamente superior é então :

$$a_{n-1n-1}x_{n-1} + a_{n-1n}x_n = b_{n-1} \tag{2.56}$$

um dos seus processadores, ver [24, vol 29, nº9 - Novembro 1996, *A parallel Climate Data Assimilation Package* - Ding, H. Q. e Ferraro, R. D.]. Este exemplo mostra um problema do cotidiano que envolve muitos cálculo matriciais sobre uma massa muito grande de dados e a necessiade de se dominarem métodos que diminuam o tempo de processamento. Em certa forma há uma sensação de que as otimizações puramente matemáticas já foram feitas e seguem-se fazendo as otimizações computacionais: melhora na arquitetura dos computadores, que permitam um processamento paralelo mais efetivo, nas linguagens para que se adaptem ao processamento paralelo e distribuido, (dentro em breve certamente não haverá diferença entre estes dois conceitos do ponto de vista de linguagem de processamento, será de pouca importância onde se encontra o processador, na *mesma máquina* em que se iniciou o *processo* ou n'outra máquina remota...

Gauss publicou o seu método há cerca de 150 anos e nada se conseguiu fazer de mais efetivo desde então . Há diversas variantes do seu método que se aplicam com bons resultados a casos muito especificos de matrizes.

**Matrizes esparsas.**

O método de Gauss consiste em sucessivamente irem-se eliminando "incognitas" por um processo de somas de equações . Na verdade não são as "incognitas" que se eliminam uma vez que elas nem siquer "existem", e sim os coeficientes que lhes correspondem.

Matricialmente falando este método consiste na *triangularização* superior ou inferior da matriz do sistema. Aplicam-se à matriz do sistema as operações acima referidas de "somas de equações" de modo a que paulatinamente se anulem todos os elementos da matriz que fiquem abaixo da diagonal principal, *triangularização superior*, ou alternativamente, se anulem os elementos que fiquem acima da diagonal principal, *triangularização inferior*.

Os dois métodos, do ponto de vista de uma matriz são equivalentes, mas não do ponto de vista de um sistema como ficará claro ao final. Vamos adotar triangularização superior. A nova matriz triangularizada é equivalente a anterior do ponto de vista do sistema de equações , ver (observação , 19). Como as alterações feitas na matriz do sistema são acompanhadas das mesmas modificações feitas também na matriz de dados, vamos obter assim um sistema de equações equivalente ao inicial, isto é, a solução dos dois sistemas, do inicial ou do novo são as mesmas.

Um classe de matrizes que surge na prática com frequência são as *matrizes esparsas*: elas tem uma grande quan-

*A equação característica fica:*

$$det(\mathcal{T} - \lambda\mathcal{I}) = \begin{vmatrix} cos(\theta) - \lambda & -sin(\theta) \\ sin(\theta) & cos(\theta) - \lambda \end{vmatrix} = (cos(\theta) - \lambda)^2 + sin^2(\theta) = 0$$

*que não tem raizes reais a não ser quando* $\theta = 2k\pi$ *para* $k \in \mathbf{Z}$.

Se uma matriz puder se colocada na forma diagonal os elementos da diagonal são as raizes da equação polinomial de grau n

$$Det(\mathcal{A} - \lambda I) = 0,$$

chamada *equação característica*. As soluções da equação característica se chamam *valores próprios* e os vetores solução da equação linear associada, para cada valor próprio se chamam *vetores próprios*

Conclusão, a forma ótima de simplificação de matrizes, a diagonal, nem sempre é atingível em sua plenitude. Vamos ver na próxima seção que sempre poderemos encontrar uma forma triangular e mostraremos lá as consequências disto para o problema importante de valores próprios.

## 2.6 Sistemas lineares: triangularição de matrizes.

Resolver sistemas de equações lineares é uma tarefa que surge pela frente em múltiplas situações em diversos ramos científicos representando possivelmente uma das mais importantes aplicações da Matemática até mesmo pelo seu aspecto elementar. Ao mesmo tempo esta é uma tarefa muito difícil por que a sua teoria algébrica é muito complexa não sendo acessivel a nível elementar. Os métodos elementares repousam em um número de operações que crescem muito rapidamente com a dimensão do sistema, (cerca de $n^2$ operações ). Apesar disto computadores rápidos resolvem sistemas de equações 100 x100 como aqueles necessários aos problemas que se encontram no capítulo 3, em alguns segundos. Num 386 com 20 MHz de frequência, (um máquina lenta), resolvemos equações até 10 x10 instantaneamente, (décimos de segundo). Entretanto fica claro que se o número de equações crescer muito o processamento pode ficar muito mais demorado. Na máquina a que nos referimos podemos levar algumas centenas de segundos, (alguns minutos), para resolver um sistema 100 x 100, entretanto já existem hoje computadores pequenos muito mais velozes que um 386 com 20 MHz de frequência...

Vale a pena indicar um problema atual envolvendo cálculos matriciais, um, que nos toca no dia a dia, é o problema da modelagem do clima. A agência americana NASA tem um centro, o *Goddard Space Flight Center*, em que parte substancial dos dados metereológicos colhidos ao redor do mundo são processados. Um *bloco* de informações se completa, chegando ao centro, em períodos de 6 horas e há uma estimativa de que estes dados são processados, (lidos e atribuidos os valores às matrizes de dados e depois efetuadas as diversas operações, muitas das quais matriciais, sobre estes dados), perfazendo assim um ciclo de processamento que dura entre 4 e 7 horas de trabalho num Cray C90 ocupando apenas

uma solução $x$ existir, podemos escrever o sistema, neste caso particular como:

$$\mathcal{A}x = \lambda x$$

ou equivalentemente

$$\mathcal{A}x = \lambda I x$$

em que $I$ representa a matriz identidade e agora subtraindo as duas equações:

$$(\mathcal{A} - \lambda I)x = 0$$

o que nos leva a dizer que estamos procurando uma solução , $x$, diferente de zero para a equação com matriz $\mathcal{A} - \lambda I$ e portanto, pela nossa discussão com determinantes, estamos procurando solução para a equação algébrica (nada óbvio):

$$Det(\mathcal{A} - \lambda I) = 0.$$

Como $\mathcal{A}$ é de ordem $n$, (nada óbvio), então estamos em presença de uma equação polinomial de grau $n$ que tem exatamente $n$ raizes *complexas*. Mas se estivermos trabalhando com números reais o número de soluções será *no máximo* $n$. Dependendo da paridade de $n$ pode não haver nenhuma solução e a conclusão é:

**Teorema: 6**

*Pode haver matriz para as quais a forma diagonal seja impossível.*

**Exemplo: 15** *Uma transformação linear impossível de diagonalizar.*

*Considere* $\mathbf{R}^2 \ni x \overset{\mathcal{T}}{\mapsto} y \in \mathbf{R}^2$ *definida geometricamente por:* $\mathcal{T}x$ *dá uma rotação de ângulo* $\theta$ *em* $x$. *A expressão algébrica para* $\mathcal{T}$ *é:*

**<u>Exercício</u>**:

$$\mathcal{T} = \begin{pmatrix} cos(\theta) & -sin(\theta) \\ sin(\theta) & cos(\theta) \end{pmatrix} \tag{2.53}$$

- *Se* $det(\mathcal{A}) \neq 0$ *então o sistema tem uma única solução que se pode expressar como*

$$x = \mathcal{A}^{-1}b$$

- *Se* $det(\mathcal{A}) = 0$ *então o sistema, se tiver solução, tem uma infinidade delas.*

Não é o objetivo a demonstração do teorema 5 agora. Ele representa apenas a síntese das discussões que fizemos até o momento.

Assim as matrizes com determinante nulo generalizam o número zero, neste contexto em que vemos as matrizes como números generalizados.

**Valores próprios, vetores próprios.**

Cheguemos ao ponto central. Queremos saber se uma matriz $\mathcal{A}$ pode ser diagonolizada. Isto equivale a perguntar se o sistema

$$\mathcal{A}x = \begin{bmatrix} a_{11} & \cdots & a_{1n} \\ \cdot & \cdots & \cdot \\ a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \equiv \tag{2.51}$$

$$\begin{bmatrix} \lambda_1 & 0 & \cdots & 0 & 0 \\ \vdots & 0 & \lambda_k \cdots & 0 & \\ 0 & 0 & \cdots & 0 & \lambda_n \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \tag{2.52}$$

Observe que se $x = (x_1, 0, \cdots, 0, 0)$ então se poderia ter $\mathcal{A}x = \lambda_1 x$. Claro, o vetor $x$ será um caso particular de solução de uma equação . Não é nada óbvio o que diremos agora, mas é intuitivo: Estamos procurando $n$ números,

$$\{\lambda_1, \cdots, \lambda_n\}$$

e naturalmente vetores $x$ tais que se possa escrever a equação $\mathcal{A}x$ na forma $\lambda x$. Se um tal número existir e naturalmente também

Na busca de algorítmos para transformar um sistema em sua forma mais simples, vemos que o método se concentra em *simplificar a matriz*.

Há dois grandes tipos de matrizes que se busca encontrar: *matrizes triangulares* ou *matrizes diagonais*. As matrizes diagonais são uma forma particular de matrizes triangulares. Elas são muito especiais por muitas razões que aos poucos mostraremos, mas obviamente elas tem uma forma obviamente simples. Seria ideal se encontrassemos um *método* geral para diagonalizar toda matriz, ou, colocado em outras palavras: encontrar a forma diagonal de uma matriz qualquer. Vamos usar agora os determinantes para descrever de forma bem geral o método e suas limitações e sobretudo mostrar que isto nem sempre é possível.

**Observação: 22** *Determinantes.*

O nosso uso dos determinantes é *ilegal*. Não definimos o que era $det(\mathcal{A})$ para uma matriz $\mathcal{A}$ qualquer. Continuaremos sem o fazer porque é dificil de fazê-lo corretamente e deixaremos que o leitor leia a respeito em algum outro livro. Ao mesmo tempo o convidamos para continuar usando os determinantes de forma extra-oficial. Esta nossa opção permite deixar Álgebra Linear confinada num capítulo de livro quando ela sozinha poderia ocupar um livro inteiro.

Apenas para poder fazer algumas contas, digamos que os Determinantes são funcionais definidos em $n$ vetores todos de dimensão $n$, ou seja são funções de $n$ variáveis vetoriais todas elas de dimensão $n$. Isto equivale a dizer que só existe determinantes de matrizes quadradas.

Além do mais digamos que os determinantes são funcionais $n$-lineares definidos sobre estas $n$ variáveis vetoriais, são funcionais multi-lineares. É este o caminho para definí-los corretamente, porisso tomaria tempo.

O determinante de uma matriz $\mathcal{A}$ define se a função

$$\mathbf{R}^n \ni \vec{x} \xrightarrow{\mathcal{A}} \mathcal{A}\vec{x} \in \mathbf{R}^n$$

é *injetiva* ou não. As funções injetivas tem solução única para as equações que elas definem, *se tiverem solução*. Se o determinante for zero a funcao $\mathcal{A}$ não é injetiva podendo ter uma infinidade de soluções. As alternativas são:

**Teorema: 5** *Soluções de um sistema linear. Considere o sistema linear:*

$$\mathcal{A}x = b$$

*da outra, e se as coordenadas $i, j$ do* vetor de dados *não estiverem na mesma proporção , então o sistema de equações é impossivel.*

2. *** Generalize o exercício anterior: Se o mesmo teste de "dependencia linear" aplicado a um sub-conjunto de linhas da matriz do sistem ao ser aplicado ao mesmo conjunto de linhas da matriz de dados tiver resultado diferente então o sistema é impossível.* solução : a igualdade de vetores representada pelo sistema...

**Observação: 20**

Esta segunda forma de simplificação de sistemas de equações difere da primeira num ponto fundamental: redução de linhas da matriz do sistema. Isto equivale a substituirem-se as linhas tiradas por linhas nulas o que significa que *a matriz do sistema equivale a uma matriz cujo determinante é zero.*

Determinante de $\mathcal{A}$ ser zero significa que a equação linear

$$\mathcal{A}x = b$$

é uma equação indeterminada. As equações indeterminadas são generalizações das equações numéricas da forma

$$0x = b$$

sendo que estas últimas só tem solução quando $b = 0$ enquanto que as equações matriciais com determinante nulo podem ter solução quando $b \neq 0$ mas neste caso o vetor $b$ tem que apresentar a mesma dependencia linear entre suas linhas que existir entre as linhas da matriz do sistema. Esta é a chave geral para solução de sistemas de equações e sua completa discussão . Posta neste termos a discussão é simples, mas o que complica é criar um algorítimo que se aplique a um sistema qualquer. É este objetivo da próxima seção.

### 2.5.3 Classes de matrizes.

Fizemos referência acima a dois tipos de equivalência de matrizes. Isto quer dizer que no conjunto de todas as matrizes há pelo menos dois métodos de classificar as matrizes.

**Observação: 21** *Classificação .*

Em vários momentos, em Matemática, passamos por classificações . Alguns casos são tão habituais que não os percebemos como é caso da classficação das frações. O conjunto de todas as frações está divido em classes e não vemos diferença entre elementos de uma mesma classe: $\frac{3}{4}$ , $\frac{6}{8}$ são tidas como iguais, na verdade são equivalentes, mas na prática funciona como se fossem iguais.

Uma dessas frações , $\frac{3}{4}$, é considerada o *padrão* , ou como se costuma dizer: a versão simplificada. Alguns professores primários cometem o absurdo de marcar com um erro se um aluno deixar $\frac{6}{8}$ em vez de escrever $\frac{3}{4}$.

Podemos, numa comparação , dizer que o sistema escalonado na página 88 é uma expressão mais simples do sistema original. Observe que dissemos *uma expressão mais simples* em vez de dizer *a expressão mais simples*, porque há varias expressões mais simples, há várias maneiras de se fazerem os escalonamentos.

Em geral a propriedade "universalidade" é dada como subentendida querendo dizer que todos os pares de elementos de um determinado conjunto onde uma relação estiver definida tem que pertencer ao domínio da relação. Consequentemente se costuma dizer que uma relação de equivalência fica definida pelas tres propriedades: *simetria, reflexividade, transitividade* e vamos aderir a esta prática ignorando a primeira. A equivalência de sistemas induz sobre o conjunto das matrizes uma relação de equivalência uma vez que as tres propriedades valem.

Uma outra forma de definir equivalência de matrizes é: Duas matrizes quadradas, $n \times n$, $A, B$, são equivalentes se houver uma matriz inversivel $\mathcal{T}$ tal que

$$\mathcal{T}A = B\mathcal{T}$$

essas duas formas de equivalência não coincidem.

### 2.5.2 Dependência linear das linhas.

A dependência linear das linhas é um outro exemplo que mostra uma forma diferente de simplificar uma matriz e por uma razão bem distinta da simplificação que estudamos acima.

O exemplo 13, página 67, mostra que as linhas de uma matriz podem trazer a mesma informação repetida: serem linearmente dependentes.

Uma maneira de verificá-lo é o cálculo do determinante da matriz, mas isto pode ser muito custoso e veremos na próxima seção, *triangularização de matrizes no método de Gauss*, que há maneiras alternativas de fazê-lo.

Se uma linha for um múltiplo de uma outra por uma constante não nula então uma discussão simples pode determinar se um sistema de equações é possível ou não : a matriz de dados tem que satisfazer a mesma condição de multiplicidade entre as correspondentes "linhas". A constante pode ser nula, mas o exemplo fica trivializado...

Se pudermos, de alguma forma, determinar quais são as linhas linearmente dependentes e encontrar um sub-conjunto delas que não o seja, o sistema fica simplificado pela eliminação do *complemento deste conjunto de linhas* da matriz do sistema e do *correspondente conjunto de linhas* da matriz de dados.

**Exercício: 15** *Teste de dependência linear.*

1. *Prove que se as linhas $i, j$ da matriz de um sistema de equações forem* linearmente dependentes, *logo múltiplas uma*

*Simplificando, ou melhor efetuando as operações com as matrizes, temos:*

$$\begin{bmatrix} 3 & 4 \\ 0 & 25 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 4 \times 7 + 3 \times 8 \end{bmatrix} = \begin{bmatrix} 7 \\ 52 \end{bmatrix} \tag{2.49}$$

*Então* $y = \frac{52}{25} = 2.08$ ; $x = -0.44$

Simplificamos o sistema substituindo a matriz do sistema primitivo por uma matriz *triangular superior*. As matrizes triangulares e as matrizes diagonais, são mais simples que as outras porque a quantidade de zeros é superior, ou pelo menos igual a dos números diferentes de zeros fora da diagonal. Tais matrizes se dizem *esparsas* e elas são excelentes para serem usadas por computadores uma vez que o trabalho com elas se reduz quase pela metade. Seus determinantes são fáceis de serem calculados: é o produto dos elementos da diagonal.

Mas dissemos que o *algebrismo usado* era pouco prático, foi uma expressão infeliz que não retiraremos em respeito ao leitor... a verdade é que o exemplo é pouco prático uma vez que nos mostrar que podemos fazer uma simplificação sem nos induzir em nenhuma pista de como fazê-lo em qualquer caso.

**Observação: 19** *Equivalência de matrizes.*

É comum dizer-se que as matrizes

$$\begin{bmatrix} 3 & 4 \\ 0 & 25 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix} \tag{2.50}$$

são equivalentes. Na verdade os sistemas de equações por elas definidos é que são equivalentes. Esta é uma forma de definir "objetos" equivalentes: pela natureza do efeito que eles causam. Há outras maneiras de definir equivalência.

Antes de ir adiante falemos de *equivalência* que é uma forma de *generalizar* o conceito de *igualdade*.

**Definição : 11** *Dizemos que uma relação de equivalência,*$\equiv$*, se encontra definida num conjunto U se:*

1. *universalidade Se* $\equiv$ *estiver definida para par de elementos do conjunto U.*
2. *simetria Se* $x \equiv y \quad \Rightarrow \quad y \equiv x$
3. *reflexividade Se* $x \equiv x$ *for verdadeira.*
4. *transitividade Se* $x \equiv y$ *e* $y \equiv z$ *então* $x \equiv z$ *for verdadeira.*

*Vamos resolver o sistema usando um algebrismo muito artístico mas que logo veremos ser nada prático. A justificativa do que faremos é seguinte. Se escrever matricialmente o sistema temos:*

$$\begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \end{bmatrix} = \mathcal{A}X = B \tag{2.46}$$

*Se multiplicarmos ambos os membros desta equação matricial pela mesma matriz a equação não se altera,* velha regra da álgebra*:*

$$\mathcal{D} \begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \mathcal{D} \begin{bmatrix} 7 \\ 8 \end{bmatrix} \equiv \mathcal{D}\mathcal{A}X = \mathcal{D}B \tag{2.47}$$

**Observação: 18** *Mudança de variável.*

Mas se a matriz $\mathcal{D}$ for *qualquer*, em vez de fazer uma operação "útil" poderemos ter feito uma operação "destrutiva". Isto é discutido em "mudanças de variável" sobre tudo em integração a várias variáveis: é preciso que a jacobiana da "mudança de variáveis" tenha um determinante não nulo: a Jacobiana é a matriz das derivadas parciais que se vê em Cálculo multivariado, e o determinante representa o a distorção da *medida* produzida pela mudança de variáveis que não pode ser zero caso contrário esta mudança de variáveis estará produzindo uma distorção irreversível. A mudança de variáveis está representada pela matriz $\mathcal{D}$

A matriz que nos interessa é aquela que efetua a clássica eliminação de variáveis: se faz uma soma das equações de tal modo que possamos eliminar uma das variáveis, por exemplo multiplicando a primeira equação por 3 e a segunda por 4 e substituindo a segunda pela soma das equações . A matriz que faz isto é:

$$\mathcal{D} = \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix}$$

verifique multiplicando esta equação pelo vetor $(a, b)$ que o resultado será $(a, 3a + 4b)$. O seu determinante é 4 o que significa que seu coeficiente de distorção é 4. Se voce for calcular uma integral com esta mudançde variáveis, depois terá que dividir o resultado por 4, ou dentro da integral deverá aparecer a fração $\frac{1}{4}$ que é o inverso do determinante da Jacobiana.

*O sistema seguinte é idêntico ao primeiro:*

$$\begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 3 & 4 \\ -4 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} 7 \\ 8 \end{bmatrix} \tag{2.48}$$

$$\begin{aligned} 3x_1 + 4x_2 + 5x_3 &= 0 \\ 3x_1 + 5x_2 - 6x_3 &= 0 \\ 7x_1 + 6x_2 - 8x_3 &= 0 \\ 2x_1 - 2x_2 - 3x_3 &= 0 \end{aligned}$$

4. *Qual é dimensão da imagem da matriz usada no item anterior vista como transformação linear. Qual é o espaço de saida ?*

## 2.5 Simplificação dos sistemas de equações.

A simplificação de um sistema de equações passa pela simplificação da matriz deste sistema. O que seria uma matriz "simples"? Para começar, esta frase sub-entende que podemos modificar os sistemas de equações sem alterá-los substancialmente. Um exemplo prova isto, exemplos *provam* a existência de "coisas", de "métodos", mas depois de analisar um exemplo a gente pode não conseguir encontrar nenhum método geral que o justifique...

### 2.5.1 Multiplicação por uma matriz não destrutiva.

**Exemplo: 14** *Simplificação de um sistema de equações.*
*Considere o sistema de equações :*

$$\begin{aligned} r_1 : &\quad 3x + 4y = 7 \\ r_2 : &\quad -4x + 3y = 8 \end{aligned}$$

*que representa a intereseção de duas retas* $r_1$ , $r_2$, *cujos coeficientes angulares são* $\frac{-3}{4}$ , $\frac{4}{3}$, *respectivamente. Logo as retas são perpendiculares entre si. A solução única do sistema é o ponto de interseção das retas que depois de algumas contas concluimos ser:* $(-0.44, 2.08)$.

Sugerimos que a dimensão de um espaço corresponde ao número *ótimo* de coordenadas com que se expressam os seus vetores. Vamos guardar esta definição , ela é aceitável. A teoria dos espaços vetoriais fica mais complicada com esta definição de dimensão , ganhamos em intuição [13] e porisso a vamos adotar. No espaço $\mathbf{R}^n$ precisamos de $n$ coordenadas para escrever os vetores, assim este espaço tem dimensão $n$. Intuitivamente, a dimensão é número mínimo de informações necessárias para descrever os elementos de um espaço, e é também o número ótimo.

É o que acontece com os tres vetores não colineares dois a dois do exercício 12, temos tres vetores que é um excesso de informações para descrever o plano: eles tinham que ser *linearmente dependentes* que é a caracterização do excesso de informações .

Vetores *linearmente independentes* geram um espaço que tem por dimensão a quantidade desses vetores.

A definição usual de dimensão é: "o número mínimo de vetores l.i. do espaço.

**Exercício: 14** *1. Como* $\mathbf{R}^3$ *deve ser um espaço de dimensão 3, mos-tre que os vetores*

$$(1,0,0),(0,1,0),(0,0,1)$$

*geram qualquer vetor* $(a,b,c) \in \mathbf{R}^3$.

*2. Verifique quais dos conjuntos abaixo é* l.i *ou* l.d. *Quando for o caso elimine vetores para tornar o conjunto restante* l.i..

*1) (1,2,3,0) , (0,3,2,1), (0, 1,1,1) , (1,-1,0,-3)*
*2) (-1,2,3). (0,1,0). (0,0,1) , (0,0,0)*
*3) (-1,2,0). (0,1,0). (1,0,1) , (1,1,1)*
*4) (-1,2,3). (0,1,0). (0,0,1)*

*3. Verifique que as linhas do sistema de equações abaixo são l.d. e, eliminando pelo menos uma delas, torne-o mais simples.*

[13] de certa forma esta maneira de fazer usa o conceito de isomorfismo, na verdade estamos dizendo que os espaços isomorfos ao $\mathbf{R}^n$ tem dimensão $n$

exatamente para que eles não colocassem "*lixo nas informações*". Mas você vê do exemplo que vetores não precisam ser ortogonais entre si para gerar o plano. A ortogonalidade e a ortonormalidade de $v = (1, 0), w = (0, 1)$ tornam a geometria mais simples.

Seria difícil repetir os exemplos dados acima com vetores do espaço *tridimensional*. A solução é concretizar o conceito de independência linear.

**Definição : 10** *Independência linear.*

*Dizemos que os vetores $u_1, \cdots, u_n$ são linearmente independentes, se a implicação seguinte for verdadeira:*

$$\alpha_1 u_1 + \cdots + \alpha_n u_n = 0 \;\Rightarrow\; \alpha_1 = \cdots = \alpha_n = 0.$$

*Dizemos que os vetores $u_1, \cdots, u_n$ são linearmente dependentes, se não forem independentes, isto é se*

$$\alpha_1 u_1 + \cdots + \alpha_n u_n = 0$$

*não forçar que*

$$\alpha_1 = \cdots = \alpha_n = 0,$$

*podendo haver pelo menos um dos $\alpha_i \neq 0$*

Análise a segunda parte da definição , a *dependencia*. Se houver $\alpha_i \neq 0$ então toda a expressão pode ser explicitada em termos de $u_i$ porque podemos dividir por $\alpha_i$ e assim $u_i$ *depende* dos outros vetores: o sistema é "redundante", $u_i$ pode ser eliminado.

**Observação: 17** *A otimização da independência linear.*

Relembre os nossos exemplos industriais, lá nos referimos a uma linha da matriz de taxa de variações sendo múltipla de outra, (a forma mais elementar de depêndencia linear). Se uma linha for linearmente dependente de outras, a informação que ela contém é redundante e isto tem como consequência maior tempo de processamento de dados que pode ser evitado eliminando-se esta linha.

Se um conjunto de vetores for "*l.i.*", eles geram um espaço, em outras palavras eles são o conjunto ótimo para gerar todas as informações contidas neles mesmos. Se um conjunto de vetores não for "*l.i.*" então entre os vetores existe alguma *redundância*, logo algum deles pode ser apagado.

Vamos usar isto para simplificar sistemas de equações , quando detectarmos que há linhas *linearmente dependentes* na matriz de um sistema de equações , apagaremos algumas delas tornando o sistema mais simples: *eliminando as redundâncias.*

*(a)* $v$
*(b)* $w$
*(c)* *2v*
*(d)* *u= (-4,0); u = (4,0); u = (1,-2)*
*(e)* $u = (a, b)$

O último item do exercício anterior é interessante, vamos resolvê-lo:

- $u = (a, b) = \alpha v + \beta w =$
- $= (-\alpha, 2\alpha) + (-\beta, 3\beta) =$
- $= (-\alpha - \beta, 2\alpha + 3\beta) = (a, b)$
- Queremos calcular $\alpha$ e $\beta$ e acima temos um sistema de equações :
- $-\alpha - \beta = a$ ; $2\alpha + 3\beta = b$
- cuja solução é, como dizem os burocratas, "salvo engano":
- $\beta = b + 2a$ ; $\alpha = -(b + 3a)$;

**Exercício: 13** *Escreva um programa para fornecer* as coordenadas *de um vetor qualquer do plano relativamente aos vetores* $v = (-1, 2), w = (-1, -3)$.

**Observação: 16** *Pacotes de Cálculo Numérico.* Veja que [21] e [8] são mais do que simples pacotes de Cálculo Numérico, ambos são linguagens muito evoluidas de programação. Com pequenos modificações os programas feitos em Pascal se adaptam a qualquer uma das duas.

Se os vetores fossem $v = (1, 0), w = (0, 1)$ as coordenadas de u = (x,y) seriam exatamente: x,y. É porisso que se escolhem estes vetores para gerar o plano, *eles não colocam "lixo" na informação* como fizeram os vetores $v = (-1, 2), w = (-1, -3)$. São dois vetores unitários e perpendiculares entre si. Lembre-se que discutimos isto com os coeficientes de Fourier... usamos um coeficiente $\frac{1}{\pi}$ para normalisar os vetores senk e cosk

O centro da questão , envolvendo os tres vetores cuja resultante é nula se encontra na questão da *independência linear*. Se a resultante dos tres vetores for nula então nos podemos escrever:

$$u := (x_1, y_1) \quad v := (x_2, y_2) \quad u := (x_3, y_3) \tag{2.42}$$

$$r := \mu u + \nu v + \omega w = 0 \;\Rightarrow \tag{2.43}$$

$$\Rightarrow \;\; \mu u = -(\nu v + \omega w) \;\Rightarrow \tag{2.44}$$

$$u = -\frac{(\nu v + \omega w)}{\mu} \tag{2.45}$$

conclusão : um dos vetores, no caso acima preferimos $u$ mas podia ser qualquer outro dos tres, *desde que seu coeficiente seja diferente de zero*, pode ser escrito como combinação linear dos outros dois. Aqui está a chave daquela questão deixada atraz: gerar o plano. Os dois vetores $v, w$ podem reproduzir qualquer vetor do plano, observe que $u$ pode ser qualquer, até mesmo colinear com algum dos outros dois, neste caso um dos coeficientes se anula, qual?

**Exercício: 12** *Independência linear, decomposição de vetores.*

*Em todos os itens abaixo, os vetores $u, v, w$ serão os mesmos definidos no primeiro item.*

1. *Verifique que os vetores $u = (1, 2), v = (-1, 2), w = (-1, -3)$ não são colineares dois a dois.*
2. *Calcule $\mu, \nu, \omega$ tal que $u = -\frac{(\nu v + \omega w)}{\mu}$.*
3. *Verifique que se podia ter dito: Calcule*

$$\alpha, \beta \quad \textit{tal que} \quad u = \alpha v + \beta w.$$

   Neste caso se diz que $\alpha$ e $\beta$ são os coeficientes de $u$ na combinação linear de $v$ com $w$. Na geometria analítica ainda se diz algumas vezes que $\alpha$ é o coeficiente de $u$ na direção de $v$ e $\beta$ é coeficiente de $u$ na direção de $w$ .

4. *Calcule os coeficientes da combinação linear dos vetores abaixo relativamente ao par de vetores $v, w$ definidos no primeiro exercício:*

origem, *porque o termo independente é zero* então eles se interceptam num ponto $(\mu, \nu, \omega) = (0, 0, 0)$ que é uma solução do sistema. Encontramos assim uma solução e queremos mostrar que há mais de uma. O conceito intuitivo de plano resolve a questão , dois planos se interceptam:

1. são paralelos:
   (a) com *interseção vazia* , ou
   (b) coincidentes;
2. se intersectam ao longo de uma reta.

são as tres possibilidades: { }, uma reta, um plano. Como há uma solução então a interseção vai ser uma reta ou um plano, em qualquer das duas circunstâncias um número infinito de soluções, $\boxed{\textbf{q.e.d .}}$

O raciocínio geométrico feito acima se estende a objetos de maior dimensão , (leia-se, com número maior de coordenadas), mas as hipóteses aumentam com a dimensão . Sempre que num sistema o número de equações for menor que o número de incógnitas, se o sistema for homogêneo então há um número infinito de soluções . Porque os sistemas homogêneos tem pelo menos uma solução, o zero, então; como é indeterminado pelo fato de ter o número de incónitas maior do que o de equações, então terá uma infinidade de soluções. Se não for homogêneo a análise tem que ser continuada. Vamos desenvolver abaixo a técnica "algébrica" com que estas questõesão tratadas.

O sistema acima é um caso particular de sistemas *sub- determinados*, número insuficiente de equações . Se trata de um sistema homogêneo: matriz de dados é nula, isto foi que nos permitiu concluir que os dois planos passavam pela origem e porisso havia uma solução .

Entretanto, se o sistema não for homogêneo este raciocí-nio já não mais se aplica e os planos *podem* ser paralelos sem coincidir um com o outro, neste caso a interseção é vazia e o sistema será *sub-determinado*, mas impossivel.

3. *Prove que se tres vetores não formarem um triângulo como no exercício 1, então eles não podem estar num mesmo plano. Neste caso, qual seria o espaço gerado por eles.*

Os conceitos de *espaço* e de *dimensão* não ficaram defini-dos até agora. *Espaço vetorial* é uma generalização da geometria. O plano é um espaço vetorial de dimensão dois. A Mecânica Relativista se passa num espaço vetorial de dimensão quatro, chamado *espaço-tempo*, mas a Mecânica Clássica se passa num espaço de dimensão 3. A melhor maneira de concretizar estas ideias vem atravéz da solução do exercício 1 da lista acima, que vamos resolver agora.

Vetores do $\mathbf{R}^2$ são pares ordenados $(x, y)$ de números reais. Os dados do exercício se traduzem com:

$$u := (x_1, y_1) \quad v := (x_2, y_2) \quad u := (x_3, y_3) \tag{2.35}$$

$$r := \mu u + \nu v + \omega w \tag{2.36}$$

$$r = 0 \equiv r_1 = r_2 = 0 \tag{2.37}$$

$$r_1 := \mu x_1 + \nu x_2 + \omega x_3 = 0 \tag{2.38}$$

$$r_2 := \mu y_1 + \nu y_2 + \omega y_3 = 0 \tag{2.39}$$

como $u, v, w$ são dados, então $\mu, \nu, \omega$ são as tres incongnitas do problema e temos assim um *sistema de equações* com duas equações e tres incónitas. Como o número de equações é menor do que o número de incógnitas, o sistema é *indeterminado*, isto significa que não temos dados suficientes para determinar os valores de $\mu, \nu, \omega$ ou ainda, haverá mais de uma solução para esta equação . Para demonstrá-lo, a maneira mais simples é artificiosa lançando mão de conceitos geométricos. O sistema se expressa assim:

$$\mu x_1 + \nu x_2 + \omega x_3 \quad = 0 \tag{2.40}$$

$$\mu y_1 + \nu y_2 + \omega y_3 \quad = 0 \tag{2.41}$$

que podemos interpretar pela Geometria Analítica, como equações de dois planos nas variáveis $\mu, \nu, \omega$. Como os planos passam pela

## 2.4 Independência linear.

Vamos discutir *independência linear*, que está intimamente ligado com o conceito de *dimensão*. Um exemplo inicial deixa as coisas mais claras.

Com apenas dois vetores do $\mathbf{R}^2$ podemos escrever qualquer outro vetor do plano?

A resposta é "não", se por exemplo este dois vetores forem colineares sua combinações lineares ficam dentro da reta que os contém. Então dois vetores quaisquer não ão suficientes, talvez, para *gerar o plano*.

Como estabelecer uma regra que nos garanta quando dois vetores podem gerar o plano? Possivelmente deveriamos nos perguntar porque é importante *gerar o plano*.

**Observação: 15** *Gerar um espaço.*

Vamos descer um pouco às origens. A ciência aos poucos descobriu a natureza íntima do que havia por traz de um "alfabeto" e a revolução que os alfabetos trouxeram para o processo de registrar a informação : Um "alfabeto" consegue codificar de modo simples a *quase totalidade* dos "sons" que emitimos quando falamos. Quer dizer que as letras do alfabeto representam os átomos da linguagem, elas "geram" as palavras de nossa lingua.

De modo semelhante os algarismos "geram" os números.

Assim é preciso descobrir "objetos mais simples" e leis de formação elementares que nos permitam "gerar" os *objetos mais complexos* de um certo universo que nos interessa. É o que acontece com os vetores unitários $i, j, k$ da mecânica ou da Geometria Analitica, as *partículas elementares* para a Física ou para a Química, as *moléculas* para a Biologia.

Outro conceito que vai junto com a idéia de "gerar" é a eliminação das redundâncias. No alfabeto da lingua portuguesa se tem redundâncias como "cedilha" e "s-duplo". Se poderia eliminar um dos dois e ainda expressar de modo adequado as palavras da lingua que dependam dum ou do outro. Razões de natureza histórica impedem que isto aconteça e a lingua fica "redundante". As redundâncias nós as chamaremos aqui de "dependencia". Dois vetores colineares não nulos são redundantes para definir uma reta, basta um deles. Nos diremos que eles são "linearmente dependentes". O adjetivo "linearmente" é típico do tratamento de vetores, dos espaços vetoriais.

A eliminação das redundâncias deixa mais simples a sintaxe do sistema, mas pode tornar mais *abstratas* as suas frases no sentido que se afastem de um modelo intuitivo, folclórico ou histórico.

**Exercício: 11** *Dependência linear.*

1. *Prove que se $u, v, w$ forem tres vetores do $\mathbf{R}^2$ que não sejam colineares dois a dois, então podemos encontrar tres escalares $\mu, \nu, \omega$ tal que $r = \mu u + \nu v + \omega w$ forme um triângulo, ou que, na linguagem da Física, a resultante $r$ seja nula.*

2. *Construa um algoritmo para resolver o problema anterior.*

Se diz *equação linear* mesmo que o termo independente não seja nulo. Neste caso há um nome para as equações em que a matriz de dados é nula, chamam-se *equações homogêneas*.

**Observação: 14** *Espaço vetorial, vetores, matrizes.*

Matrizes e vetores são *vetores*. *Vetor* é um elemento de um *espaço vetorial* e isto quer dizer um conjunto munido de duas operações , *soma* e *produto por escalar* satisfazendo as propriedades habituais dos vetores, veja num livro de *Álgebra* a definição precisa de espaço vetorial.

Queremos resolver equações lineares com um grande nú-mero de incognitas e equações . O conhecido método de Cramer que usa determinantes se torna logo inviável computacionalemente pelo grande tempo que exige para fazer as operações . A grande maioria das linguagens de processamento não tem capacidade numérica para calcular produtos que ultrapassem a ordem de "30!" (com trinta fatores). O número 30! tem 33 casas decimais e a maioria das linguagens estão estruturadas para oferecer 16 a 20 casas decimais de precisão . Há linguagens de programação voltadas para "precisão infinita", LISP é uma delas[12], Parece que *Python*, ver [19], também oferece a possibilidade de precisão infinita. Mas o problema do tempo de processamento continua crucial. Um método extremamente simples e efetivo é da *substituição sucessiva*, de Gauss, mas sua efetividade aumenta se primeiro a matriz do sistema for *transformada* numa matriz triangular. *Transformada* é uma palavra-chave, quer dizer que uma modificação será feita na matriz sem que o sistema de equações seja modificado no sentido de que suas soluções sejam preservadas. Nas duas próximas secções iremos discutir como modificar uma matriz preservando as soluções do sistema de equações que elas definem. As proximas seções se dedicam a ver como se podem modificar equações sem alterar sua solução.

[12] nas implementações de LISP se pode dar a uma variável de sistema o número de casas decimais com precisão exata com se quer trabalhar e um valor negativo significa que a precisão é total, o resto depende da memória que se dispuser na máquina... e do tempo que se tenha para esperar.

2. *Multiplique as matrizes $\mathcal{A}$ e $\mathcal{B}$ dos dois modos possiveis:*

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \ , \ \mathcal{B} = \begin{bmatrix} 1 & 1 & 20 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{2.32}$$

*verificando que numa das multiplicações , $\mathcal{AB}$ ou $\mathcal{BA}$ é nula e que a outra não é nula.*

3. *Conclua que na multiplicação de matrizes por acontecer que $\mathcal{AB} = 0$ mas $\mathcal{A} \neq 0$ ; $\mathcal{B} \neq 0$*

**Definição : 9**

*Um par de "numeros" $a, b$ que sejam diferentes de zero mas cujo produto $ab = ba = 0$, se dizem divisores de zero. No (exer. 2.32), as matrizes não são um exemplo de divisor de zero porque $\mathcal{AB} \neq \mathcal{BA}$, Para isto veja o (exer.2.33 ).*

4. *Multiplique as matrizes $\mathcal{A}$ e $\mathcal{B}$ dos dois modos possiveis:*

$$\mathcal{A} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} \ , \ \mathcal{B} = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \tag{2.33}$$

*verificando que as matrizes $\mathcal{A}, \mathcal{B}$ formam um par de divisores de zero.*

Com o que vimos podemos agora considerar $\mathcal{A}x = b$ como uma equação matricial, $x$ habitualmente é um vetor coluna, tem $p$ coordenadas assim como $b$ que é habitualmente chamada de *matriz de dados* apesar de ser um vetor. Dizemos então que temos um *sistema de equações lineares* com $p$ incónitas, que correspondem às coordenadas de $x$, e $n$ *equações* que corresponde ao número de linhas de $\mathcal{A}$:

$$\mathcal{A}x = \begin{pmatrix} a_{11}x_1 + \cdots + a_{1p}x_p \\ \cdots \\ a_{n1}x_1 + \cdots + a_{np}x_p \end{pmatrix} = \begin{bmatrix} b_1 \\ \cdots \\ b_p \end{bmatrix} = b \tag{2.34}$$

é diferente de (eq. 2.27), por exemplo, as dimensões de $x$ e de $b$ tem que ser diferentes. Não vamos demonstrar a seguinte afirmação, ela é intuitivamente óbvia, mas sua demonstração tomaria algum tempo:

**Teorema: 4** *Equivalência de equações lineares. Para toda equação linear à esquerda*

$$x^t\mathcal{B} = b^t \tag{2.29}$$

*existe uma equação linear à direita*

$$\mathcal{A}x = b \tag{2.30}$$

*que lhe é equivalente.*

A operação $x \mapsto x^t$ se chama *transposição*, e $x^t$ se diz a *transposta* de $x$.

**Observação: 13** *Hábitos no escrever equações.* O Teorema 2.30 nos libera para escrever as equações lineares no formato *a direita*, da equação (eq. 2.27), que é o que sempre faremos. Os autores que usam equações à esquerda estão certíssimos porque todos os *vetores* ficam expressos como matrizes-linha que ocupam menos espaço de folha de livro e parecem mais naturais no escrever, entretanto a equações assumem um aspecto operacional não usual relativamente ao hábito elementar de operação com os números.

Infelizmente nem sempre podemos resolver uma equação matricial multiplicando tudo pelo inverso multiplicativo de $\mathcal{A}$, porque, *eis a principal diferença com os números usuais*, as matrizes nem sempre tem inversos multiplicativos[11]. Os *números usuais* todos tem inversos multiplicativos com exceção de zero: *a álgebra das matrizes é um pouquinho mais complicada...* o exercício seguinte lhe mostra isto.

**Exercício: 10** *Divisores de zero.*

1. *Multiplique as matrizes $\mathcal{A}$ e $\mathcal{B}$ do único modo possivel:*

$$\mathcal{A} = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \ , \ \mathcal{B} = \begin{bmatrix} 1 & 1 & 20 & -3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{2.31}$$

[11] há várias teorias tentando contornar este problema de inversão .

*em Pascal que são de domínio público para transformá-lo em* C *ou n'outra linguagem de seu agrado*[9].

*Entretanto vamos nos dirigir a um público que pode não ter experiência em programação e que se limite a ser um usuário de programas de computador. Para estes apontaremos o uso de programas de cálculo numérico e nos limitaremos a [8] e [21] que são de domínio púlico com implementações para* Linux *mas também com versões que rodam sob outros sistemas operacionais.*

## 2.3 Sistemas de equações lineares.

Vamos de agora em diante usar tacitamente as matrizes fazendo generalização da multiplicação . Escreveremos expressões do tipo

$$\mathcal{A}x = b \tag{2.27}$$

sem nenhum comentário, considerando natural que se pense que $\mathcal{A}, x, b$ são *números generalizados*[10], são objetos que se comportam como números. As matrizes são números generalizados! Já vimos que em expressões deste tipo se podem usar matrizes sendo necessário ter cuidado com as dimensões para que tais equações se possam escrever. Na equação (eq. 2.27) nxm, mxp tem que ser as dimensões de $\mathcal{A}, x$, respectivamente, e nxp a dimensão de $b$, a igualdade obriga a isto. Em geral quando nos depararmos com uma expressão do tipo (eq. 2.27), as matrizes $x, b$ serão *vetores*, matrizes-coluna, quando estiver na forma da equação (eq. 2.27). Como o produto de matrizes não é comutativo, a equação

$$x\mathcal{A} = b \tag{2.28}$$

[9] existe uma linguagem de programação de domínio público, parecida com Pascal, chamada Python, ver [19], com implementações para *Linux* e outros sistemas operacionais.

[10] há uma generalização dos números reais, *non standard real numbers*, que nada tem o que ver com matrizes e são números generalizados, reconhecemos aqui um defeito de nomenclatura.

(c) *Mostre que se $T$ for escalar, então $Tx = \lambda x$, para qualquer vetor $x$.*

5. *matriz diagonal: As matrizes diagonais são a generalização mais imediata das matrizes escalares:*

**Definição : 8**

*Uma matriz $\mathcal{D}$, nxn, se diz diagonal se todos os seus termos forem nulos exceto, possivelmente, os que se encontrarem na diagonal principal.*

(a) *Verifique que a matriz nula é diagonal.*

(b) *Mostre que as matrizes escalares são diagonais.*

(c) *Mostre que as matrizes diagonais comutam entre si.*

(d) *Mostre que as matrizes diagonoais não comutam com as outras matrizes: se $\mathcal{D}$ for uma matriz diagonal então se pode ter $\mathcal{DA} \neq \mathcal{AD}$.*

6. *preparação para o resto do capítulo: Alternativamente:*

- *Construa um programa em Pascal, (ou aprenda a usar um) que consiga ler matrizes pelo teclado ou a partir de dados num disco. Faça o seu programa somar e multiplicar matrizes. Mais para frente o programa deverá* aprender *a resolver sistemas de equações e outras operações matriciais...*
- *Aprenda a usar um programa que saiba manipular matrizes, sugestão : [8], [21].*

**Observação: 12** *Programas, programação e matrizes.*

*Vamos deslocar o centro de atenção, no que diz respeito a* programação *para pacotes de cálculo numérico prontos. Se sua experiência computacional lhe permitir vivamente aconselhamos a que continue a construir você mesmos os seus programas. Para isto faça uso inclusive dos nossos programas*

*são os espaços* $\mathbf{R}^n$*, um funcional linear será da forma*

$$f : \mathbf{R}^n \to \mathbf{R}; f(x) = (a_1, \cdots, a_n) \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \in \mathbf{R} =$$
$$= a_1 x_1 + \cdots + a_n x_n$$

Exemplos de funcionais lineares são os produtos escalares, inclusive aquele que definimos entre funções, com auxílio de uma integral na secção sobre *polinômios de Fourier* no capítulo anterior. Se você se detiver um pouco na análise daqueles funcionais, ver a fórmula 1.5, página 42, sua conclusão deve ser que eles seguem o mesmo modelo, a integral funcionando como a generalização da soma.

**Exercício: 9** *Tipos particulares de transformações lineares.*

1. *Chama-se* $Tr(\mathcal{A})$ *a uma função que associa uma matriz quadrada à soma dos elementos de sua diagonal principal. Mostre que que* $Tr$ *é um funcional linear.*

2. *Considere a função vetorial* $f : \mathbf{R}^n \to \mathbf{R}^n$*. Escreva* $div(f)$ *como composição de duas funções que tenham* $f$ *como paramâtro.*

3. *A forma trivial do teorema de Green estabelece a igualdade entre duas integrais nulas quando aplicada a um gradiente. Escreva a expressão do integrando matricialmente.*

4. *matriz escalar: Por definição temos:*

   **Definição : 7**

   *Uma matriz é escalar se for múltipla da matriz identidade por um escalar:* $\lambda I$*, em que* $I$ *é matriz identidade de ordem* $n$ *e* $\lambda$ *um escalar.*

   (a) *Verifique que a matriz identidade de ordem* $n$ *é escalar.*

   (b) *Verifique que se* $T$ *for uma matriz escalar, então* $T\mathcal{A} = \mathcal{A}T$ *para qualquer matriz* $\mathcal{A}$ *de ordem* $n$*.*

2. *Escreva algumas funções lineares usando distintas matrizes no que diz respeito a dimensão .*

3. *No planejamento de uma indústria se consideram as variáveis $x_1, \cdots, x_n$ representando os diferentes itens de matéria prima necessários. A matriz de variação de custos da indústria tem m linhas em que foram registradas as taxas de variação dos custos dos diversos itens necessários à produção:*

$$\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix} \tag{2.26}$$

*O decente governo do país editou uma* medida provisória *estabelecendo uma série de* impostos definitivos *incidindo sobre compra de produtos industriais primários, a CPIP* [8]*, com valores distintos para cada produto e além disto sujeitos a algumas variações de mercado. Como todo o processamento da industria está pronto, será necessário introduzir um produto matricial para corrigir os custos que,* naturalmente*, devem ser repassados ao consumidor. Corrija a expressão inicial da função de planejamento:*

$$f(x) = f(a) + J(f)(x - a)$$

*introduzindo a* matriz dos impostos *no local adequado.*

As funções lineares transformam vetores em outros vetores ou números. Quando transformam em números, isto é, quando a imagem for um número, recebem um nome especial:

**Definição : 6** *Funcional linear.*

*Se f for uma transformação linear cuja imagem é um número se chama funcional linear.*

*Como os espaços vetoriais de que tratamos aqui*

[8] "C" na sigla do imposto significa "contribuição"...

na nova fórmula 2.25 a matriz está fazendo o papel de número multiplicando a matriz coluna $\Delta x$ e como são matrizes de ordens 1x4 e 4x1 o resultado desta multiplicação é um número real.

Vemos desta forma que as matrizes servem para definir nos *espaços vetoriais* funções semelhantes as funções do primeiro grau:

$$f(x) = b + Ax \ ; \ b \text{ um número}, x, A \ \text{ matrizes } n\text{x}1, 1\text{x}n$$

Há várias combinações possiveis de dimensão na construção de tais funções. Acima chamamos $x$ de matriz quando o habitual é chamar de vetor. Veja mais o seguinte exemplo:

$$f(x) = B + Ax \quad ; \quad B, A, x \text{ matrizes: } n\text{x}1, n\text{x}p, p\text{x}1.$$

Se costuma chamar funções do primeiro grau de *lineares*, na verdade deveriam ser chamadas de *lineares afins*. São *lineares* aquelas com o termo constante $b$ ou $B$ nulo:

$$f(x) = Ax \quad ; \quad A, x \text{ matrizes: } n\text{x}p, p\text{x}k.$$

definidas por uma simples multiplicação. Nestas valem as propriedas de linearidade:

**Definição : 5** *Transformações lineares. Se $f$ for uma transformação linear então*

1. $f(x + y) = f(x) + f(y)$ ;
2. $f(\lambda x) = \lambda f(x)$ .

Os termos "*transformações lineares*" e "*funções lineares*" são sinônimos, mas há quem dê um significado geométrico ao primeiro.

**Exercício: 8** *Funções lineares.*

1. *Verifique que as propriedades de linearidade valem tanto para $f(x) = ax$ em que $a, x$ são números, como para $f(x) = Ax$ em que $A, x$ são matrizes,* convenientemente definidas para que se possa fazer a multiplicação.

*Verifique se é a melhor palavra é* dum *ou* do, num *ou* no *na redação acima. Justifique.*

## 2.2 Transformações Lineares.

A fórmula 2.24 pode ser escrita com outro aspecto. Vamos supor que $f : \mathbf{R}^4 \to \mathbf{R}$ então se calcularmos $f'(a)$ no ponto $(a_1, \cdots, a_4)$ teremos uma matriz linha com 4 entradas formadas pelas 4 derivadas parciais [7] de $f$:

$$f'(a) = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \frac{\partial f}{\partial x_3}, \frac{\partial f}{\partial x_4})$$

Usando esta notação podemos re-escrever a fórmula 2.24:

$$f(x) \approx f(a) + (\tfrac{\partial f}{\partial x_1}, \tfrac{\partial f}{\partial x_2}, \tfrac{\partial f}{\partial x_3}, \tfrac{\partial f}{\partial x_4}) \begin{bmatrix} x_1 - a_1 \\ x_2 - a_2 \\ x_3 - a_3 \\ x_4 - a_4 \end{bmatrix} = \tag{2.25}$$

$$= f(a) + (\tfrac{\partial f}{\partial x_1}, \tfrac{\partial f}{\partial x_2}, \tfrac{\partial f}{\partial x_3}, \tfrac{\partial f}{\partial x_4}) \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta x_3 \\ \Delta x_4 \end{bmatrix} =$$

$$= f(a) + \tfrac{\partial f}{\partial x_1}\Delta x_1 + \tfrac{\partial f}{\partial x_2}\Delta x_2 + \tfrac{\partial f}{\partial x_3}\Delta x_3 + \tfrac{\partial f}{\partial x_4}\Delta x_4 =$$

$$= f(a) + \tfrac{\partial f}{\partial x_1}(x_1 - a_1) + \tfrac{\partial f}{\partial x_2}(x_2 - a_2) + \tfrac{\partial f}{\partial x_3}(x_3 - a_3) + \tfrac{\partial f}{\partial x_4}(x_4 - a_4)$$

em que vemos a matriz atuando como *um dispositivo operatório* na definição de uma função, (*uma nova função que é uma aproximação de* $f$). Observe que esta esta expressão é semelhante à expressão de uma função do primeiro grau:

$$f(x) = b + ax \quad ; \quad x, a, b \in \mathbf{R}$$

[7] A notação de derivadas parciais não deixa ver que as derivadas estão sendo calculadas no ponto a, isto causa dificuldade para o entendimento.

*$grad(f)_{(a,b)}$ é um vetor perpendicular à curva de nível que passa no ponto $(a,b)$. Conclua que o $grad(f)$ aponta na direção de créscimento, (ou decréscimento) máximo de $f$ a partir do ponto $(a,b)$.*

11. **direção de crescimento máximo: Considere um ponto $a \in \mathbf{R}^n$ no domínio de uma função $f$ e o círculo $C_{a,r} = \{x\ ;\ d(a,x) = r\}$ também contido no domínio de $f$. Verifique que o seguinte algoritmo determina a região onde se encontra o* caminho *de maior crescimento de $f$ :*

    - *Cálculo do valor médio*

      $$ValMed(f)_{C_{a,r}} = \frac{1}{2\pi r}\int_0^{2\pi} f(a + rcos(t), b + rsen(t))dt$$

      *Observe que se trata de uma integral univariada percorrendo a parametrização de um círculo.*
    - *Determinação dos pontos acima da média Gráfico dos pontos $\{(x,y)\ \ (x,y) \in C_{a,r}\ \ f(x,y) > ValMed(f)_{C_{a,r}}\}$*
    - *Determinação do máximo como função de $r$ nos conjuntos do item anterior. Determinação dos pontos em que ocorra o máximo. Determinação de uma ou mais curvas com os pontos assim encontrados.*

12. ** Construa um programa que determine curvas de crescimento máximo para uma função bivariada usando o algorítmo da questão anterior. Aplique-o nas funções do (exercicio , 8).*

13. **Mostre que se você estiver numa região montanhosa e no ponto a tiver $grad(f) = 0$ em que $f$ é a equação da montanha, então:*

    - *você chegou ao fundo* dum *vale.*
    - *você se encontra no topo* dum *monte.*
    - *você se encontra* num *passo de montanhas.*

*apresentado no texto. Faça um pequeno projeto de simulação industrial usando matrizes como $J(f)$ em que $f$ é uma amostragem de dados do processo industrial.*

7. *Construa um exemplo em que a matriz $J(f)$ representa a taxa de lucro dos distintos produtos. O vetor $a$ representa a taxa de venda dos produtos. Defina um teto de lucro aceitável e a partir deste teto verifique que $\frac{\partial f}{\partial x_i}$ depende do valor de $a_i$, mostre como.*

8. **curva de nível: Se $F : \Omega \subset \mathbf{R}^2 \to \mathbf{R}$, se definem os subconjuntos de $\Omega$*

$$\text{curva de nível}_k = \{(x, y) \in \Omega \ ; \ F(x, y) = k; k \in \mathbf{R}\}$$

[5] *Encontre as curvas de nível indicado:*

(a) $F(x, y) = x^2 + y^2 \ ; \ k \in \{0, 0.5, 1, 2\}$
(b) $F(x, y) = x^2 - y^2 \ ; \ k \in \{-1, -0.5, 0, 0.5, 1, 2\}$
(c) $F(x, y) = (x-3)^2+(y+4)^2 \ ; \ k \in \{-1, -0.5, 0, 0.5, 1, 2\}$
(d) $F(x, y) = (x-3)^2-(y+4)^2 \ ; \ k \in \{-1, -0.5, 0, 0.5, 1, 2\}$
(e) $F(x, y) = (x-a)^2+(y-b)^2 \ ; \ k \in \{-1, -0.5, 0, 0.5, 1, 2\}$
(f) $*F(x, y) = 5(x-2)^2+3(y-1)^2 \ ; \ k \in \{-1, -0.5, 0, 0.5, 1, 2\}$
(g) $F(x, y) = 5(x-2)^2+2xy-3(y-1)^2 \ ; \ k \in \{-1, -0.5, 0, 0.5, 1, 2\}$
(h) $F(x, y) = xy \ ; \ k \in \{-1, -0.5, 0, 0.5, 1, 2\}$

9. **reta tangente à curva de nível. Para cada função do item anterior, encontre um ponto $(a, b)$ sobre a cuurva de nível, calcule a equação da reta tangente acra curva no ponto $(a, b)$ e faça os gráficos correspondentes.*

10. *gradiente: Se chama $graf(f)$ a jacobiana $J(f)$ quando $f : \Omega \subset \mathbf{R}^n \to \mathbf{R}$. Verifique[6] que $grad(f)_{(a,b)}$, o gradiente de $f$ calculado no ponto $(a, b)$, é um vetor. Mostre que*

[5] este nome vem dos mapas dos topógrafos que indicam assim os diferentes níveis dos terrenos.
[6] em suma, $grad(f)$ é um nome para uma jacobiana particular.

**mega-nós** $a_\alpha$ aos principais, eles caracterizam os pontos centrais de uma discretização da região de trabalho;

**micro-nós** $a_{\alpha,\beta}$ aos secundários que ficam a uma pequena na distância dos outros e que servem para calcular derivadas direcionais aproximadas no nó principal associado $a_\alpha$.

A linguagem está um pouco mais complexa que a usada no capítulo anterior porque estamos tratando de problemas multi-dimensionais, agora também os índices tem que ter mais coordenadas, *em princípio o número de coordenadas das variáveis do problema*, para facilitar uma "visão geográfica" da distribuição dos nós dentro da região de trabalho. "Índices" são um sistema de numeração, da forma como estão sendo usados aqui, suas coordenadas são inteiras indicando a quantidade total de pontos indexados. Para indexar, por exemplo, os pontos de uma região do plano usariamos:

$$\alpha \in \{(1,1),(2,1),\cdots,(N,1),(1,2),(2,2),\cdots,(N,2),\cdots,(N,N)\}$$

que seria a mesma forma dos índices $\beta$, estamos indicando que há $N^2$ mega-nós e semelhante indicação viria da numeração dos índices $\beta$.

**Exercício: 7** *Matrizes.*

*No disco você encontra o arquivo* Sistema.pas *que contém as rotinas necessárias à solução destes exercícios.*

1. *Escreva um programa que receba pelo teclado matrizes e as multiplique na ordem em que foram dadas.*

2. *Modifique o programa anterior para, peguntando ao usuário a ordem dos fatores, multiplique as matrizes na ordem indicada.*

3. *Construa um exemplo de matrizes* $\mathcal{A}, \mathcal{B}$ *tal que* $\mathcal{A}\cdot\mathcal{B} \neq \mathcal{B}\cdot\mathcal{A}$

4. *Inclua no seu programa a possibilidade de somar duas matrizes com um* alarme *no caso de as matrizes serem incompatíveis para soma. No caso de incompatibilidade o programa deve perguntar ao usuário se as deve somar assim mesmo e então completar linhas ou colunas com zeros de modo a poder efetuar a soma.*

5. *Faça seu programa calcular a* $J(f)$ *usando derivadas aproximadas.*

6. *Pesquise e descreva caso real de aplicação de matrizes em sua área de formação à semelhança do* exemplo industrial

*cada caso como uma função separada:*

$$C_1(x) = C_1(a) + C_1'(a)\Delta x \tag{2.19}$$
$$C_2(x) = C_2(a) + C_2'(a)\Delta x \tag{2.20}$$
$$C_3(x) = C_3(a) + C_3'(a)\Delta x \tag{2.21}$$
$$C_1 : \mathbf{R}^4 \to \mathbf{R}\ ;\ C_2 : \mathbf{R}^4 \to \mathbf{R}\ ;\ C_3 : \mathbf{R}^4 \to \mathbf{R} \tag{2.22}$$

*sendo*

$$Custo_{total}(x) = C_1(x) + C_2(x) + C_3(x) \tag{2.23}$$

As tres funções definidas por produtos de matrizes podem ser somadas porque todas tem o mesmo número de variáveis, 4, e todas tem o mesmo número de componentes, 1.

Só podemos somar matrizes que sejam exatamente da mesma ordem.

**Observação: 10** *Muitas vezes conhecemos $f'(a)$ e não $f(a)$.*

Uma pergunta poderia ser feita: porque colocamos ênfase em $f'(a)$ e não em $f(a)$? O desenvolvimento do capítulo anterior em certa forma responde a esta questão. Em geral não conhecemos $f$ mas sim alguns de seus valores e possivelmente alguns valores muito próximos de alguns nós $a_\alpha$ e consequentemente conhecemos $f(a_\alpha)$ e estamos em condições de calcular $f'(a_\alpha)$ sabendo através de um levantamento de dados $f(a_{\alpha,\beta})$ em que $\alpha,\beta$ são multi-índices, sendo $\alpha$ o multi-índice que caracteriza os nós principais da rede e $\beta$ caracterizam os nós finos em volta de algum nó $a_\alpha$, bem a semelhança do que fizemos no capítulo 1,

Convém lembrar aqui a fórmula de Taylor em seu caso mais simples que é

$$f(x) \approx f(a) + f'(a)(x-a) = f(a) + f'(a)\Delta x \tag{2.24}$$

em que $f$ é uma função vetorial e portanto $f'(a)$ é uma matriz jacobiana não trivial, (não é um *número comum*).

Se conhecermos $f(a_\alpha$ e $f(a_{\alpha\beta}$, os valores de $f$ sobre os nós de uma rede com um levantamento de dados fino em volta de cada nó, podemos projetar os valores de $f$ numa vizinhança dos nós com auxílio do polinômio de Taylor.

**Observação: 11** *Multi-índices.*

Quando precisamos indexar pontos de uma região , usamos índices. Algumas vezes precisamos de caracterizar que alguns pontos tem uma hierarquia diferente de outros: *nós principais* ou *nós secundários*. Diferenciamos estes nós "funcionalmente" diferentes chaman-do-os de:

isto que uma função $f$ acelera ou desacelera o comportamento de outra $g$ se as duas representam fenômenos que atuem simultâneamente: duas forças atuando sobre um mesmo corpo o aceleram *se tiverem componentes na mesma direção.* Se tiverem mesma direção e *sentidos contrários* podem lhe dar aceleração zero se tiverem mesmo módulo. As forças *se superpuseram*, dizem os físicos, *se somaram* dizemos os matemáticos. Duas *forças* só se podem somar se as suas *variáveis* e *componentes* forem em mesmo número:

$$f : \mathbf{R}^n \to \mathbf{R}^m, g : \mathbf{R}^n \to \mathbf{R}^m$$

porque atuam sobre objetos do *mesmo "espaço"* e representam resultados que se encontram no *mesmo "espaço"*.

Contrariamente, se

$$f : \mathbf{R}^n \to \mathbf{R}^m, g : \mathbf{R}^n \to \mathbf{R}^q \ ; \ m \neq q$$

então não se podem somar as funções $f$ e $g$ nem

$$f : \mathbf{R}^n \to \mathbf{R}^m, g : \mathbf{R}^q \to \mathbf{R}^m \ ; \ n \neq q.$$

Como $f : \mathbf{R}^n \to \mathbf{R}^m, g : \mathbf{R}^n \to \mathbf{R}^m$ se podem somar, também se podem somar as suas derivadas calculadas no mesmo ponto $a = (a_1, \cdots, a_n)$ que serão matrizes $n$x$m$ porque ambas as funções tem $nm$ coeficientes parciais. Daí tiramos a regra, só podemos somar matrizes de mesmas dimensões .

Poderiamos tentar representar com um fundo industrial o significado da soma de matrizes, o faremos sem grandes detalhes.

**Exemplo: 13** *A soma dos custos.*

*No exemplo industrial acima, consideramos separadamente o efeito sobre o custo final duma ação de marketing, do custo laboral e do custo da materia prima. Podemos entender*

mxn por outra de ordem nxq não interessando o valor de m e de q.

**Exemplo: 12** *O esquema da ordem das matrizes na multiplicação.*

$$A_{n\mathrm{X}m} \cdot B_{m\mathrm{X}q} \to C_{n\mathrm{X}q}$$

*em que os índices se encontram indicados em cada matriz.*

Há mais alguma coisa que podemos explorar no exemplo acima: *que significaria se os coeficientes que formam a linha 3 fossem dependentes dos coeficientes que formam a linha 2*, proporcionais, queremos dizer? Representariam uma informação inútil e consequentemente é ter *custos sem benefícios* mantê-los no processo pois a terceira coordenada do ve*tor de variação de custos* seria proporcional à segunda coordenada e portanto poderia ser obtido a partir da segunda por simples multiplicação. A matriz *ótima* para esta analise econômica, neste caso seria 2x3 eliminando-se uma linha da matriz $\mathcal{A}$ e da matriz $b$. Assim, se uma matriz tiver linhas que dependam linearmente de outras, o problema pode ser simplificado eliminando-se as linhas *linearmente dependentes*, não todas, obviamente: de cada conjunto de linhas linearmente dependentes entre si se devem eliminar aquelas que fazem o conjunto ficar linearmente dependente.

**Observação: 9** *Dependência linear e otimização. A palavra chave aqui é* otimização*, se otimizou o controle eliminando linhas* linearmente dependentes *da matriz que contém os dados do processo industrial.*

Começamos por multiplicar matrizes, acima o fizemos com matrizes 3x4 e 4x1. Falemos agora da soma de matrizes. A soma de matrizes traduz um conceito da Física chamado de *superposição*. Se $\mathcal{A} = J(f) = f'(a)$ e $\mathcal{B} = J(g) = g'(a)$ e se pudermos somar as duas funções $f, g$ então também poderemos somar $f'(a), f'(b)$. É um princípio do Cálculo: *se pudermos somar duas funções, poderemos também somar suas derivadas.* Os físicos chamam esta soma de superposição significando com

*essas que acabamos de expor. O exemplo serve em sua simplicidade para ilustrar o* produto de matrizes*, mostrando que elas são* um novo tipo de número*, um número que contém múltiplas informações a um só tempo: um* multi-número*.*

**Observação: 8** *Aproximação diferencial e modelagem.*

*Uma das lições que podemos tirar do presente exemplo é que a existência de uma função, como a função de custos $C$, não se dá diretamente atravéz de uma equação mas sim tudo o que temos é sua aproximação diferencial:*

$$C(x) \approx C(a) + C'(a)dx = C(a) + C'(a)(x-a) \tag{2.18}$$

*a partir do valor contabilizado de custos no ponto $a$ e com as informções estatísticas que chegam indicando as distintas taxas de variação $J(C) = C'(a)$ é possível determinar-se o custo previsível em curto espaço de tempo futuro. É ainda interessante observar que a palavra "aproximação" está sendo usada num sentido* histórico e folclórico*: não existe nenhuma função $C$ para ser aproximada. A* aproximação diferencial *<u>é</u> a própria função.*

*A aproximação diferencial representa, desta forma uma modelagem da realidade a partir de dados obtidos estatisticamente.*

Vimos outra forma de ver o produto de matrizes em que uma matriz fixa define uma função, neste caso $dC$ é a imagem de $dx$ por uma função cuja equação é um produto pela matriz $\mathcal{A}$.

É também uma forma de ilustrar a *álgebra de matrizes* mostrando o uso de adições e multiplicações de elementos de dimensões distintas, mas compatíveis. As matrizes são *um novo tipo de número*, um número que contém múltiplas informações a um só tempo.

Ficou evidente a regra básica para fazer produto de matrizes: a *dimensão* intermediária entre elas coincida, no presente caso o 4. Podemos multiplicar uma matriz de ordem

*O custo da indústria será obtido aproximadamente por*

$$\mathcal{A}\cdot dx = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \cdot \begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \\ dx_4 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} = \tag{2.15}$$

$$= C'(a)\cdot dx = C'(a)\cdot(x-a) = dC \tag{2.16}$$

$$C(x) \approx C(a) + C'(a)(x-a). \tag{2.17}$$

*A informação* taxa de variação *é muito* fina *e produz pequenas variações no planejamento permitindo correções antes que um* erro *avantajado produza grandes estragos. Observe também que é absolutamente natural primeiro conhecermos as informações sobre as* taxas de variação parciais *do que* conhecer *a equação da função.*

*A última equação, (eq. 2.17) é a expressão do polinômio de Taylor do primeiro grau* multivariado *que voç deve observar que é exatamente a fórmula do caso univariado.*

*Uma outra forma de ver o produto de matrizes é como função linear, neste caso $dC$ é a imagem de $dx$ por uma função cuja equação é um produto pela matriz $\mathcal{A} = C'(a)$. O vetor $dx$ representa uma "pequena" variação da quantização dos insumos comprados pela empresa, e obviamente são função de uma pequena variação do tempo de planejamento, (um dia ou um mes, dependendo do processo inflacionário). A forma "natural" matemática para representar as variações, seria o tempo como variável, mas em geral, para uma empresa o que interessa no seu planejamento são as unidades de compra que internamente guardam um compasso com o tempo.*

*Vimos assim surgir o mesmo exemplo de dois modos diferentes os dois exemplos representam a mesma situação,* $\boxed{a_{ij} = \frac{\partial C_i}{\partial x_j}}$ *em que $C : \mathbf{R}^4 \to \mathbf{R}^3$ é função que modela o custo da economia em que se encontra inserida a empresa em questão cujo universo econômico se reduz aqui a quatro variáveis. Em geral um problema econômico tem muito mais variáveis do que*

*descrita acima linha por linha, representa o coeficiente angular* múltiplo *no* instante *em que foi colhida: dia ou mes*[4].

*Se* $a_{33} = 0$, *isto significa que houve uma estabilidade do custo de distribuição relativo a* $x_3$ *porque apenas se estabilizou a* demanda *deste insumo no produto. Nada garante que o custo de produção relativo a* $x_3$ *se estabilize e pode até aumentar por uma ociosidade que produza na "máquina".*

*Uma análise similar conduz a que* $a_{33} = 0$ *não garante que* $a_{13} = 0$.

$$a_{33} = 0 \rightarrow a_{23} = 0 \quad \textit{é falso} \; ;$$
$$a_{33} = 0 \rightarrow a_{13} = 0 \quad \textit{é falso} \; ;$$

*Este exemplo mostra que as linhas da matriz* $\mathcal{A}$ podem ser independentes.

*Por definição, duas linhas de uma matriz, ou dois vetores quaisquer, são* linearmente dependentes *se um for múltiplo do outro. Então, se forem dependentes, uma mesma coordenada não pode ser num deles zero enquanto que no outro é diferente de zero. A definição de* dependência linear *não fica tão simples para um conjunto com mais de dois vetores, voltaremos em seguida à definição de* dependência linear.

*Assim*

$$\mathcal{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} \tag{2.13}$$

*representa assim a matriz das variações dos custos da indústria. Se a função*

$$C = (C_1, C_2, C_3) \tag{2.14}$$

*for a função de custos desta empresa, então* $\mathcal{A}$ *representa a matriz de variação de custos e o produto das matrizes 3x4 com o a matriz 4x1 resulta na matriz* $dC$ *3x1 que é o vetor* diferencial de custos *da produção da indústria.*

[4] observe a *fragilidade* do conceito " instantâneo".

*Uma indústria depende de quatro itens, ou insumos, bási-cos na composição de seu produto final e descreve com 3 funções o seu custo de produção:*

$$C = \begin{cases} C_1(x_1, ..., x_4) = \textit{custo de insumos} \\ C_2(x_1, ..., x_4) = \textit{custo de produção} \\ C_3(x_1, ..., x_4) = \textit{custo de distribuição} \end{cases} \tag{2.11}$$

*Os* itens, *em alguma forma* quantificados, *são agora as* variáveis *no modelo matemático.*

*Estas funções não existem na prática*[2], *pelo menos não sob forma de uma equação algébrica, mas um processo estatístico, cuidadosamente levado em dia, permite que a empresa determine as flutuações*[3] *de mercado dos preços dos produtos assim como as flutuações dos custos de* produção *e de distribuição:*

*taxas, parciais, de variação de custo dos insumos:*

$$(a_{11} \quad a_{12} \quad a_{13} \quad a_{14}),$$

*taxas, parciais, de variação de custo de produção:*

$$(a_{21} \quad a_{22} \quad a_{23} \quad a_{24}),$$

*taxas, parciais, de variação de custo de distribuição:*

$$(a_{31} \quad a_{32} \quad a_{33} \quad a_{34}),$$

*Estas taxas de variação são colhidas na unidade mínima de tempo que seja natural para o planejamento da empresa, digamos, diariamente, numa economia de inflação alta, ou mensalmente numa economia de inflação reduzida. Assim, a matriz*

$$\mathcal{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \end{bmatrix} = \begin{bmatrix} \frac{\partial C_1}{\partial x_1} & \frac{\partial C_1}{\partial x_2} & \frac{\partial C_1}{\partial x_3} & \frac{\partial C_1}{\partial x_4} \\ \frac{\partial C_2}{\partial x_1} & \frac{\partial C_2}{\partial x_2} & \frac{\partial C_2}{\partial x_3} & \frac{\partial C_2}{\partial x_4} \\ \frac{\partial C_3}{\partial x_1} & \frac{\partial C_3}{\partial x_2} & \frac{\partial C_3}{\partial x_3} & \frac{\partial C_3}{\partial x_4} \end{bmatrix} \tag{2.12}$$

[2] aqui se repete o que vimos no capítulo anterior, conhecemos apenas alguns valores de $f$.

[3] leia: "taxa de variação"

*que é uma expressão semelhante a do* diferencial de funções de uma variável*:* $df = f'(a)dx$*. Aqui vemos a* generalização *feita por Cayley entendendo matrizes como* multi-números*. A matriz* $J(f)$ *é a matriz de todos os possíveis coeficientes angulares instantâneos* parciais *de* $f$ *relativos a um* referêncial *previamente escolhido. Este produto matricial pode ser expandido para se obter o que se chama em livros de Cálculo de* diferencial total*:*

$$df = J(f)\begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \\ dx_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}dx_1 + \frac{\partial f_1}{\partial x_2}dx_2 + \frac{\partial f_1}{\partial x_3}dx_3 + \frac{\partial f_1}{\partial x_4}dx_4 \\ \frac{\partial f_2}{\partial x_1}dx_1 + \frac{\partial f_2}{\partial x_2}dx_2 + \frac{\partial f_2}{\partial x_3}dx_3 + \frac{\partial f_2}{\partial x_4}dx_4 \\ \frac{\partial f_3}{\partial x_1}dx_1 + \frac{\partial f_3}{\partial x_2}dx_2 + \frac{\partial f_3}{\partial x_3}dx_3 + \frac{\partial f_3}{\partial x_4}dx_4 \end{bmatrix} \quad (2.10)$$

aqui uma matriz cujas linhas são diferenciais totais*, e observe que agora nesta última equação tem-se uma igualdade entre dois vetores-coluna ou matrizes 3*x*1.*

O conteúdo do exemplo anterior mostra que as matrizes se multiplicam de forma semelhante como se multiplicam os números e na comparação anterior temos o produto de números $df = f'(a)dx$ no caso duma função univariada ou o produto matricial $df = J(f)dx$ no caso duma função multivariada. Em ambos os casos poderiamos adotar a mesma notação, e é isto que faremos de agora em diante: $df = f'(a)dx$ passará a representar o diferencial de uma função em qualquer caso, se $f$ for uma função multivariada, $f'(a)$ representa a sua matriz jacobiana calculada no ponto $a$, o seu único coeficiente multi-angular, se não for, representa o seu coeficiente angular numérico único no ponto $a$. No primeiro caso, uma matriz, no segundo caso, um número comum.

Usamos este exemplo do Cálculo para mostrar que tem sentido a multiplicação de matrizes. O próximo exemplo pode também ser descrito com as palavras do Cálculo e nós o faremos em seguida. Ele também serve para ilustrar o aspecto *multi-facetado* da informação contida nas matrizes.

**Exemplo: 11** *A matriz de custos de uma indústria.*

*Podemos ai ver quatro vetores-coluna cada um com tres coordenadas ou podemos ver tres vetores-linha cada um com quatro coordenadas. As duas maneiras de ver são válidas. As matrizes gene-ralizam os números, enquanto que estes contém uma única informação de uma medida feita, agora as matrizes contém várias informações oriundas de distintas medições feitas que podem até ser de naturezas diferentes entre si.*

**Exemplo: 10** *Matriz dos coeficientes angulares.*

*Se* $f : \mathcal{U} \subset \mathbf{R}^4 \mapsto \mathbf{R}^3$ *então no ponto* $(a_1, \cdots, a_4)$, *a matriz*

$$J(f) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \end{bmatrix}$$

*representa o* coeficiente angular múltiplo *de* $f$, *cada um dos números* $\frac{\partial f_j}{\partial x_i}$ *representa um coeficiente angular parcial, também chamado de* derivada parcial *de* $f_j$ *com respeito à variável* $x_i$ *e calculado no ponto* $(a_1, \cdots, a_4)$. *Esta matriz recebe o nome de* matriz jacobiana de $f = J(f)$.

*Da mesma forma como uma função univariada*

$$f : \mathbf{R} \mapsto \mathbf{R}$$

*tem um único coeficiente angular num determinado ponto, se for diferenciável, também*

$$f : \mathcal{U} \subset \mathbf{R}^4 \mapsto \mathbf{R}^3$$

*tem* <u>*único*</u> *"coeficiente angular múltiplo" representado pela matriz* $J(f)$ jacobiana de $f$ *no ponto* $(a_1, \cdots, a_4)$ *em que estas derivadas parciais foram calculadas. O diferencial de* $f$ *no ponto* $(a_1, \cdots, a_4)$ *é*

$$df = J(f)dx = J(f) \cdot \begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \\ dx_4 \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \frac{\partial f_1}{\partial x_3} & \frac{\partial f_1}{\partial x_4} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \frac{\partial f_2}{\partial x_3} & \frac{\partial f_2}{\partial x_4} \\ \frac{\partial f_3}{\partial x_1} & \frac{\partial f_3}{\partial x_2} & \frac{\partial f_3}{\partial x_3} & \frac{\partial f_3}{\partial x_4} \end{bmatrix} \cdot \begin{bmatrix} dx_1 \\ dx_2 \\ dx_3 \\ dx_4 \end{bmatrix} \tag{2.9}$$

*geometria não der mais pé.* Assim, se o espaço de trabalho for um conjunto de funções, por exemplo $\mathcal{C}([a,b])$ então

$$[a,b] \xrightarrow{f} \mathbf{R} \qquad [a,b] \xrightarrow{g} \mathbf{R} \tag{2.4}$$

$$< f,g >= \int_a^b f(x)g(x)dx \tag{2.5}$$

$$||f|| = \sqrt{< f,f >} \quad ||g|| = \sqrt{< g,g >} \tag{2.6}$$

$$\text{ângulo(f,g)} = \frac{< f,g >}{||f|| \cdot ||g||} \tag{2.7}$$

como é usual fazer com os vetores na Geometria Analítica. Observe a diferença fundamental entre os dois sistemas de equações acima. No caso dos vetores do $\mathbf{R}^n$ definimos o produto escalar *geometricamente* usando o conceito de ângulo que existe como ideia natural, (inclusive para dois vetores, mesmo do $\mathbf{R}^n$, como determinam um plano, ângulo é "geometrico"...). No caso das funções[1], *vetores do espaço* $\mathcal{C}([a,b])$, primeiro definimos o produto escalar, depois definimos ângulo, com auxílio deste novo conceito. Este é o método da *generalização*, se estendem a novas situações as idéias antigas, através de outras que sejam naturais na nova situação.

Mas vamos nos afastar desta linha de trabalho que leva a um campo que está fora do objetivo deste livro. Vamos retomar o produto escalar e vê-lo de uma outra forma. Veja que lhe demos o nome de *produto* que indica uma semelhança com o *produto entre números.* De fato é esta semelhança que interessa, e o produto escalar define uma forma de *multiplicar* vetores e outras entidades parecidas de que trataremos agora.

Mais geral que os vetores é um objeto chamado *matriz*, porque os vetores são também matrizes. Vetores são matrizes de um tipo particular, tem uma única linha, ou uma única coluna.

**Exemplo: 9** *Uma matriz 3x2.*

*Considere o esquema formado por 12 números dispostos da maneira regular que abaixo se vê.*

$$\left| \begin{array}{cccc} 1 & 2 & 3 & -1 \\ -1 & 1 & 0 & 2 \\ 2 & -1 & 3 & 2 \end{array} \right] \tag{2.8}$$

[1] veja o preconceito, primeiro dissemos "funções", depois as chamamos de vetores...

# Capítulo 2

# Álgebra Linear Computacional.

## 2.1 A álgebra das matrizes.

O conceito de vetor surgiu na Física como muitas das noções da matemática. Este conceito físico estava ligado a uma entidade geométrica, associada a uma seta, porque tinha que ter *direção e intensidade*. Esta visão geométrica é primitiva e tem que ser generalizada para ser melhor aplicada em distintas situações, e, como sempre, é um processo *algébrico, ou formal* que produz a generalização adequada.

**Exemplo: 8** *Generalização da geometria.*

Os conceitos de ângulo, comprimento ou módulo, ficam todos generalizados pelo conceito de produto escalar de já falamos anteriormente no capítulo 1 quando descrevemos o método de Fourier, ver 1.5, página 42. Alí os vetores eram *funções* e o que fizemos foi calcular-lhes *a soma ponto a ponto*, é o que significa uma integral que os franceses chamam de *somme*. Em Geometria Analítica se define o produto escalar de dois vetores:

$$u = (x_1, \cdots, x_n) \quad v = (y_1, \cdots, y_n) \tag{2.1}$$

$$< u, v >= \sum_{i=1}^{n} x_i y_i \quad = |u| \cdot |v| \cos(\theta) \tag{2.2}$$

$$\text{ângulo}(u, v) = \cos(\theta) \qquad = \frac{<u,v>}{|u| \cdot |v|} \tag{2.3}$$

que é a forma *discreta* equivalente a integral que se encontra na página 42. A última igualdade, de natureza geométrica, pode ser utilizada para definir *ângulo* quando a

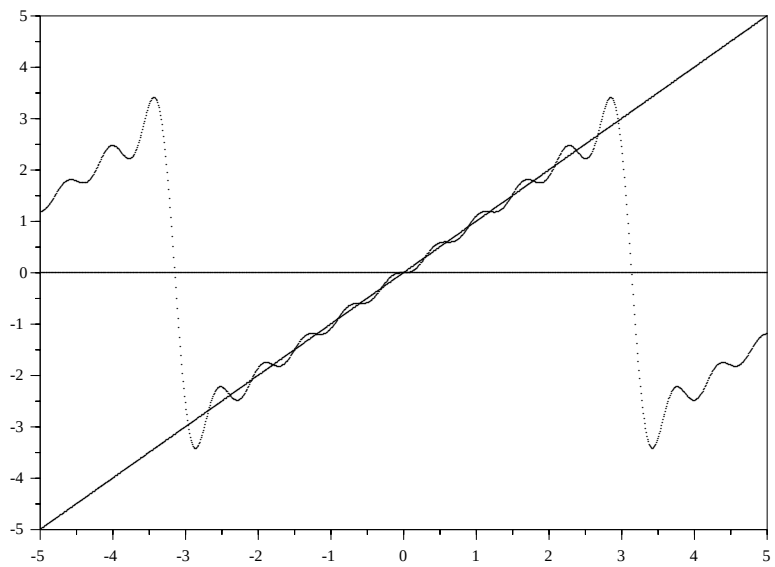



Figura 1.9: Aproximação da primeira bissetriz sobre o intervalo $[-\pi, \pi]$ por pol. trig.

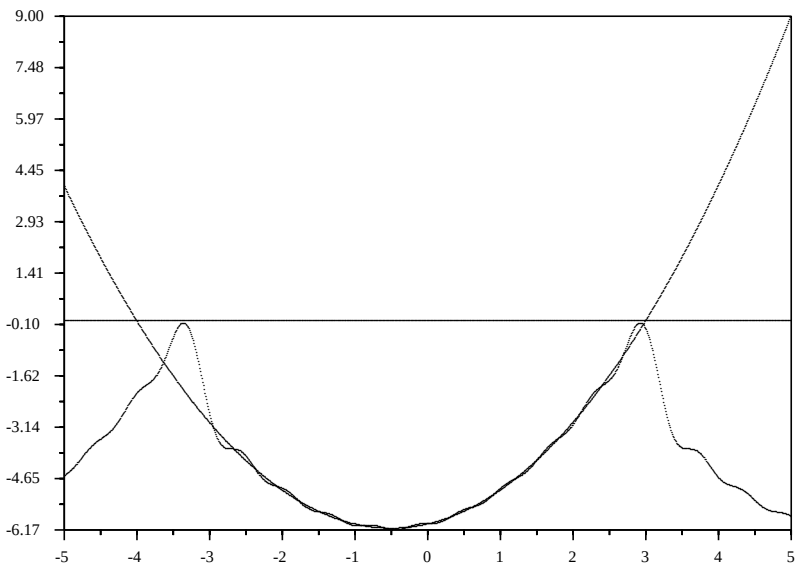



Figura 1.8: Aproximação com um polin. trigonométrico de 11 termos.

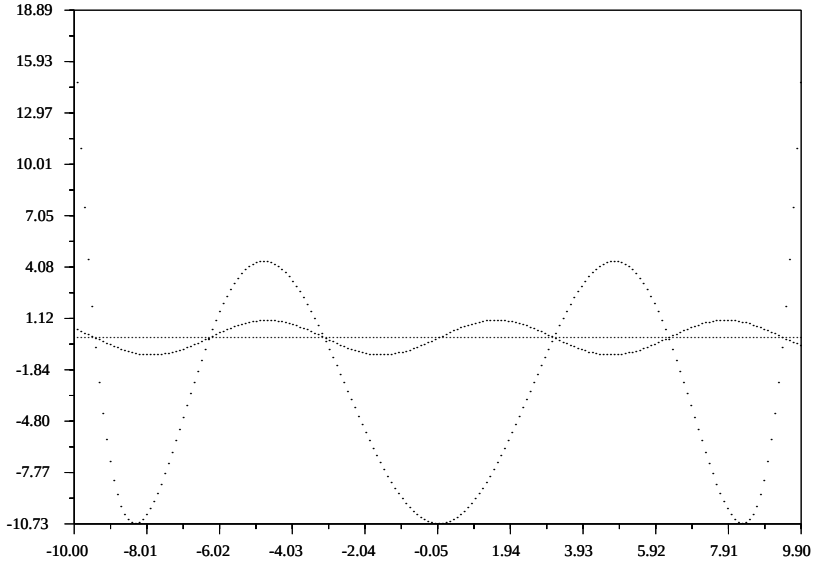



Figura 1.7: Interpolação polinomial clássica. Um polinômio que tem as mesmas raizes que *sen*.

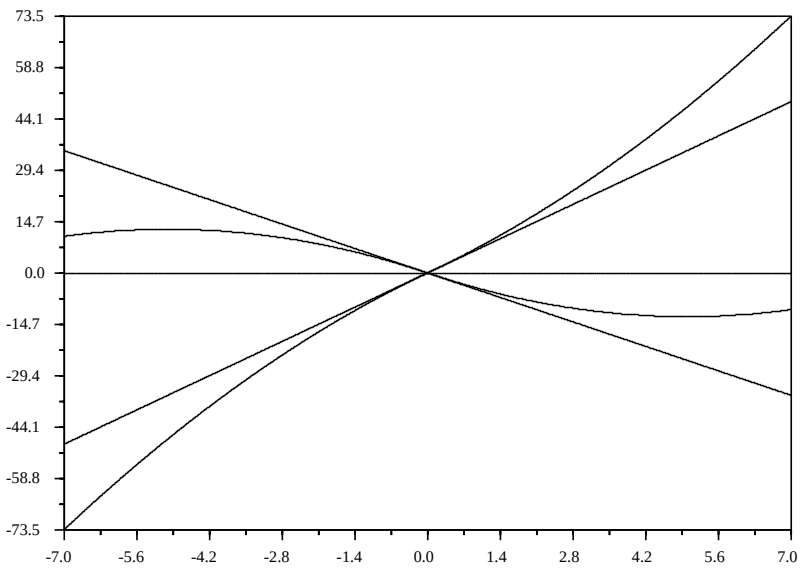



Figura 1.6: Splines tangentes a uma reta.

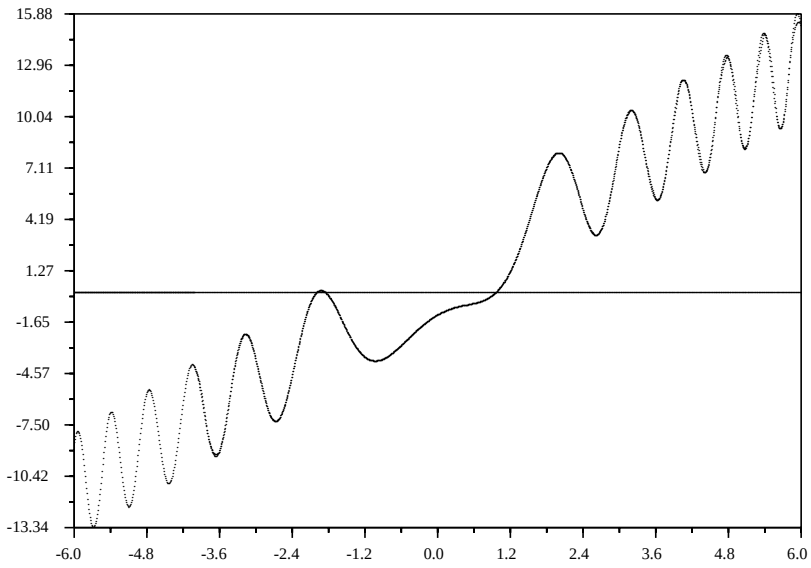



Figura 1.5: O erro decresce ao se refinar a partição. Neste caso partição uniforme de norma $\frac{1}{5}$. O número de nós da partição determina a ordem do elemento da sucessão de Cauchy que está sendo contruida.

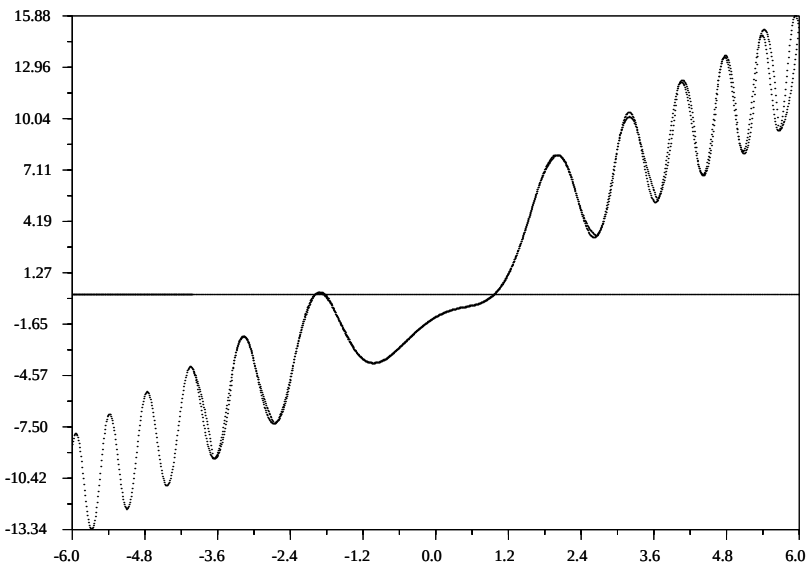



Figura 1.4: O erro cresce com a oscilação.Partição uniforme de norma $\frac{1}{3}$

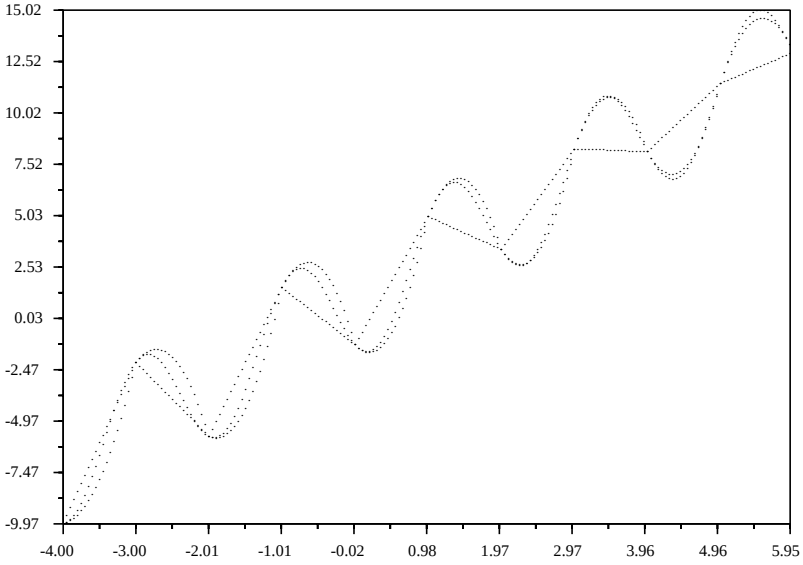



Figura 1.3: Interpolação polinomial linear e do segundo grau de $graf(f)$; com polinômio por pedaços de segundo grau contínuo.

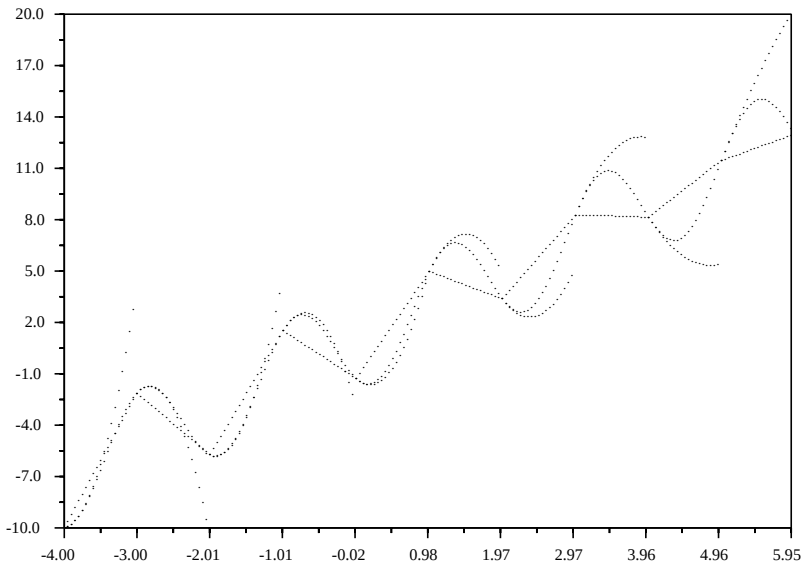



Figura 1.2: Interpolação polinomial linear e do segundo grau de $graf(f)$ com polinômio do segundo grau por pedaços descontínuo.

decomposição em *ondas*, finalmente servem para aproximação que é o objetivo último que *sempre* temos quando quisermos transmitir ou guardar informações uma vez que nos será impossível registrar ou recompor um fenômeno qualquer em sua completitude.

Se estas ideias tiverem excitado sua curiosidade, você poderá facilmente experimentar um pouco com elas. Veja a *procedure musica.pas* dentro da biblioteca *tipos.pas* com a qual você poderá aprender a fazer *música eletrônica* e inclusive ver o efeito de algumas *séries de Fourier* se aplicar o valor de $P_n(f)$ como argumento da função *sound* do Pascal. Tenha alguns cuidados, não execute o programa durante a noite se alguém já estiver dormindo no quarto ao lado ... alguma vezes isto se pode tornar um incômodo grave ...

emplos a sua volta e mais abaixo lhe diremos inclusive como você a poderá usar como laboratório. Se você ainda puder encontrar um instrumento de cordas, um violão, experimente e veja o que acontece com uma de suas cordas quando você quando nela produzir um impulso fornecendo-lhe energia. Ela vibrará de *modo perceptível* durante algum tempo transmitindo para o ar e finalmente para seus tímpanos vibrações. Serão praticamente as mesmas vibrações, apenas elas se irão amortecendo enquanto a corda perde energia para o ar a volta assim como para a própria caixa do instrumento. Acontece ai algo parecido com o péndulo. Num mesmo espaço de tempo que se repete, as vibrações voltam a se produzir e é isto que caracteriza uma *nota musical*, apenas sua intensidade vai diminuindo com a diluição da energia para ambiente: ar, caixa de ressonância do instrumento, até atingir o imperceptível para o seu ouvido. Claro que a corda continuará vibrando, até mesmo com outros tipos de vibrações vindas do meio ambiente, dos carros que passem por exemplo, e isto pode se encontrar fora da capacidade de percepção do ouvido humano.

Mas que é uma partitura musical? É a "soma", (dizem os matemáticos) ou a superposição, (dizem os físicos), destas vibrações provocadas no conjunto das cordas e alteradas por novos golpes dos dedos sobre as cordas do violão. Cada vez que o músico *interfere* sobre as cordas, ele altera a *equação* da música, criando assim *harmônicos por pedaços*, aqui a partição do intervalo de tempo segue o rítmo da música, que determina assim os intervalos em que cada novo *pedaço* de *harmônicos* será executado.

Para decompor música, as séries de Fourier são invencíveis, infelizmente parece que o ouvido humano recompõe as vibrações de uma forma que é mais compatível com outros tipos de ondas, ver [5, cap 1] ou também [13, cap 1], e sendo assim ganhamos, os humanos, em percepção se as ondas musicais forem decompostas com outros tipos de ondas diferentes dos *senk, cosk*. Infelizmente para as séries de Fourier ...

Resumindo, as séries de Fourier ou outros tipos de

Teriamos que discutir com maiores detalhes qual o significado da convergência representada pela série acima. Faremos isto em algum momento, posteriormente, no capítulo 5, entretanto vejamos logo qual é ideia intuitiva e geométrica que se encontra por traz desta convergência.

Para lhe dar uma ideia do que você poderia ter visto se tivesse rodado os programas sugeridos nos exercícios, e talvez estimulá-lo a fazê-lo agora, veja o gráfico (fig. 1.9), nele você tem os gráficos conjuntos da função identidade $f(x) = x$ e do polinômio trigonométrico $P_n(f)$ para $n = 11$. Os polinômios trigonométricos descrevem fenômenos oscilatórios como veremos em seguida, então $P_n(f)$ "oscila" em torno de $f$. É isto que destroi a "convergência" num sentido comum e mais intuitivo e que foi contestado no exemplo de Du Bois-Reymond, entretanto do ponto de vista da *energia* contida em $f$, ou mais exatamente no fenômeno modelado por $f$, a aproximação é excelente. A *energia* está representada pela integral de $f$ e agora sim: *a integral de $P_n(f)$ se aproxima muito da integral de $f$* no intervalo $[-\pi, \pi]$. Polinômios trigonométricos são aproximação de funções periódicas ou então de uma função , mas apenas sobre um intervalo em que ela é considerada como restrição de um função periódica, mas do ponto de vista da *da quantidade de fenômeno*, ou ainda, a integral de $f$ é bem aproximada pela integral de $P_n(f)$. Mas é preciso abrir uma ressalva que deixará um travo amargo: não estamos apresentando polinômios trigonométricos como um método para calcular integrais aproximadamente.

Para terminar esta breve introdução sobre os polinômios trigonométricos vamos mencionar qual é a sua função específica uma vez que dissemos que *aproximações* não é sua principal finalidade. Como eles são somas de oscilações, *senos* e em cosenos acelerados, então os polinômio trigonométricos descrevem bem os fenômenos oscilatórios. Eles podem usados para analisar a composição das *ondas* contidas nestes fenômenos. É isto que muitas vezes é chamado de *análise espectral*.

A música pode oferecer excelentes e múltiplos ex-

8. *Modifique as sucessões $s_1$ e $s_2$ dos coeficientes de Fourier e veja a função que disto vai resultar.*

9. *Tente definir uma função a partir do aspecto gráfico de suas experiências acima, e coloque-a em Fourier. Imprima os resultados gráficos e compare.*

10. *Modifique Fourier para imprimir em papel os coeficientes de Fourier (do Polinômio Trigonétrico). Tente descobrir uma lei de formação e teste o resultado em Fourier.*

11. *Modifique o programa Fourier para que seja calculada a derivada aproximada do Polin. Trig. e para que os dois gráficos sejam simultaneamente desenhados. Analise o resultado, do ponto de vista de variação do coeficiente angular do Pol. Trig. em alguns pontos.*

12. *Verifique se consegue descobrir uma função que gera um determinado Pol. Trigonométrico analisando os graficos de P e de P' (do polin. trigon. e de sua derivada). Observe que isto nem sempre será possivel e que não se trata de um método geral de trabalho.*

De acordo com os resultados que você conseguiu nos exercícios acima, podemos repetir a afirmação de Joseph Fourier feita no artigo apresentado à Academia Francesa de Ciências, em 1807: " *uma função qualquer, periódica*, $f$, pode ser escrita como combinação linear de $senk$ e $cosk$":

$$f(x) = \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k cos(kx) + b_k sen(kx). \tag{1.9}$$

Embora isto seja uma verdade, num sentido que Fourier mal podia imaginar em sua época, em 1873, Paul Du Bois-Reymond construiu um exemplo de função contínua cuja série de Fourier divergia em um determinado ponto, traduzindo em certa forma a desconfiança da validade da afirmação de Fourier em uma forma tão genérica.

*escalar definido por*

$$< f, g >= \int_0^{2\pi} f(x)g(x)dx. \tag{1.8}$$

*e depois inclua o coeficiente* $\frac{1}{\pi}$ *na integral e volte a calcular os módulos destes vetores. Tente uma demonstração formal dos resultados alcançados.*

3. *Faça um programa que calcule as projeções de* $f(x) = sen(4x+3)+3x+1$ *nas direções dos vetores* $senk$; $k \in \{0, 1, 2, 3, \cdots, 100\}$. *Estes números, como na Geometria Analítica, são os* coeficientes da decomposição do vetor $f(x) = sen(4x+3) + 3x + 1$ *relativamente ao conjunto de vetores* $senk$ ; $k \in \{0, 1, 2, 3, \cdots, 100\}$. Não se esqueça de manter presente que estamos trabalhando dentro de $\mathcal{C}([0, 2\pi])$, ou de $\mathcal{L}^2([0, 2\pi])$ .

4. *Chame* $a_k$ *aos coeficientes encontrados na questão anterior. Complete o programa para calcular o vetor* $g(x) = \sum_{k=0}^{10} a_k senk(x)$ *e faça os gráficos de de* $f$ *e* $g$ *num mesmo sistema de eixos.*

5. *Faça um programa que calcule as projeções de* $f(x) = sen(4x+3) + 3x + 1$ *nas direções dos vetores* $senk, cosk$ ; $k \in \{0, 1, \cdots, 10\}$. *Chame este coeficientes de* $a_k, b_k$, *respectivamente. Complete o programa para calcular o vetor*

$$g(x) = \sum_{k=0}^{10} a_k senk(x) + b_k cosk(x)$$

*fazendo os gráficos de* $f$ *e* $g$ *num mesmo sistema de eixos. Estes coeficientes se chamam* coeficientes de Fourier *de* $f$.

6. *Rode o programa Fourier. Ele lhe permite ver um polinômio trigonométrico cujos coeficientes estão previamente definidos como uma sucessão no próprio programa.*

7. *Defina uma das sucessões em Fourier como identicamente zero e verifique que o resultado é uma função par ou impar, dependendo de quem é identicamente zero.*

que lhe permitirão uma visão complementar caso você se decida a ler mais alguma coisa a este respeito. Ou brinque um pouco com estas ferramentas. O programa *Tutorial para o Ensino do Cálculo*, ver [15] tem rotinas dedicadas ao experimentos com Polinômios Trigonométricos também.

A figura (fig. 1.8) ilustra a aproximação do polinômio $P(x) = (x+4)(x-3)$ por um polinômio trigonométrico com 11 termos. Veja que fora do intervalo $[-\pi, \pi]$ a aproximação não existe. Ou seja, a aproximação se dá apenas no intervalo $[-\pi, \pi]$. Para mais detalhes sobre este assunto, consulte um livro de Cálculo. Observe em particular a discrepância que deixamos de propósito quando falamos dos coeficientes sobre o intervalo $[0, 2\pi]$, mas o exemplo gráfico corresponde ao intervalo $[-\pi, \pi]$. Não há nenhum erro, os coeficientes de Fourier se podem calcular em qualquer intervalo, naturalmente a fórmula deve ser adaptada para isto via uma transformação do tipo *mudança de variáveis*.

**Observação: 7** *Polinômio trigonométrico - nomenclatura.*
A denominação de "polinômio trigonométrico" fere um pouco o hábito, mas não está errada. *polinoômio* quer dizer uma expressão algébrica constituida de mais de um termo. Aqui os termos são expressões trigonométricas.

Os exercícios abaixo contém algumas experiências que não se constituem de métodos seguros para obter resultados. *São exemplos de experimentos que se podem fazer para ganhar mais intuição com o uso de Polinômios Trigonométricos, sendo exclusivamente este o objetivo delas aqui.*

**Exercício: 6** *Experiências com Polinômios Trigonométricos.*

1. *Use um programa de cálculo de integrais aproximadamente para verificar que* $\operatorname{sen} k$ *e* $\cos j$ *são ortogonais para quaisquer que sejam* $k$ *e* $j$.

2. *Verifique numericamente qual é o módulo dos vetores* $\operatorname{sen} k$ *e* $\cos j$ *para vários valores de* $k$ *e* $j$. *Primeiro use o produto*

intervalo fechado e limitada, *intervalo compacto, vamos particularizar o problema para apresentar uma teoria pequena, o intervalo base será* $[0, 2\pi]$. Para provar isto é preciso definir neste espaço $\mathcal{C}([0, 2\pi])$ um produto escalar, o conceito que nos permite calcular os ângulos entre vetores ou os módulos destes. Lembre-se do seu curso de Geometria Análitica, é o produto escalar que permite *generalizar* os conceitos geométricos, *ângulo, módulo, distância* a espaços mais gerais. O produto escalar poderia ser definido assim

$$< f, g >= \int_0^{2\pi} f(x)g(x)dx. \tag{1.5}$$

Duas integrações consecutivas, por partes, mostram que senk e cosj são ortogonais para quaisquer que sejam k e j, (mesmo iguais, neste caso por que a integral sobre o intervalo $[0, 2\pi]$ é igual a integral sobre o intervalo $[-\pi, \pi]$ e aí se pode usar o fato de que senk é ímpar enquanto que cosj é par).

De maneira análoga se pode mostrar que senk e senj são mutuamente ortogonais se $k \neq j$ o mesmo se dando com cosj e cosk .

Finalmente redefinindo o produto escalar acima se pode dizer que estes vetores são unitários:

$$< f, g >= \frac{1}{\pi} \int_0^{2\pi} f(x)g(x)dx. \tag{1.6}$$

A escolha da constante $\frac{1}{\pi}$ multiplicando a integral se liga ao fato de que

$$< sen, sen >= \int_0^{2\pi} sen^2(x)dx = \pi \tag{1.7}$$

Tem-se assim uma base de vetores ortonormais para um espaço vetorial que contém o espaço $\mathcal{C}([0, 2\pi])$ . Este espaço se chama $\mathcal{L}^2_{[0,2\pi]}$. Você pode ler mais a este respeito num livro de Cálculo Avança-do. Vamos lhe sugerir alguns experimentos que podem ser feitos com auxílio do programa Fourier e

título *Trigonometric Series*, tem perto de 1000 páginas em seus dois volumes. Estamos querendo fazer uma introdução experimental do assunto, apenas. Se conseguirmos despertar sua curiosidade, teremos atingido nosso objetivo.

O assunto é histórico. O uso de séries trigonométricas, (polinômios trigonométricos), para aproximar funções talvez esteja no fim dos seus dias uma vez que possivelmente será substituido por uma nova técnica que lhe robou a metodologia com inovações significativas: as Wavelets.

A base teórica para o conteúdo deste parágrafo será desenvolvida resumidamente no início do próximo capítulo, é a Álgebra Linear. Talvez fosse bom ler, de forma intercalada este parágrafo e o primeiro do capítulo seguinte. Este contém vários exemplos dos conceitos alí definidos.

Em 1822, num artigo apresentado à *Academia Francesa de Ciências*, Joseph Fourier, (1768-1830) afirmou que *todas as funções periódicas* podem ser decompostas em múltiplos das funções

$$x \longrightarrow sen(nx)$$

e

$$x \longrightarrow cos(kx)$$

com $n, k \in \mathbf{N}$.

Fourier não poderia entender o alcance do conceito de *aproximação* que estava enunciando nem os seus contemporâneos imediatos o poderiam, inclusive os que tentaram criticar a importância dos fatos usados por ele. A revolução que Fourier provocou no desenvolvimento da Matemática com as suas Séries Trigonométricas, chamadas ainda de *Séries de Fourier*, mas que eram conhecidas de alguns matemá-ticos anteriores a ele, como Euler, (1707-1783) e alguns dos irmãos Bernouilli, só puderam ser aquilatadas no século atual.

As funções

$$senk : \mathbf{R} \longrightarrow \mathbf{R}; x \mapsto sen(kx) \tag{1.3}$$

e

$$cosj : \mathbf{R} \longrightarrow \mathbf{R}; x \mapsto cos(jx) \tag{1.4}$$

formam um sistema de *vetores* linearmente independentes e ortogonais no espaço vetorial das funções contínuas definidas num

O segundo defeito é que para obter boas aproximações por interpolação polinomial, os polinômios tem de ser grau elevado e aí os seus valores também serão muito elevados, (ou muito pequenos) o que ocupa muito espaço de memória e consequentemente muito tempo de computação , ou imprecisão nos resultados. A *matriz* numérica dos computadores é fixa, ela é apenas "translatada" na medida em que você precisar de números maiores ou menores. Se os números forem muito grandes ou muito pequenos, se perde consideravelmente precisão.

**Exemplo: 7** *Interpolação polinomial de Lagrange.*

*O matemático francês Lagrange, se diz, construiu uma maneira engenhosa de corrigir a oscilação da interpolação polinomial comum. Não a vamos reconstruir aqui em detalhe, mas ideia é simples: cálcule o máximo da derivada do polinômio, é a constante de Lipschiz, e divida o polinômio por esta constante. O resultado será um polinômio que oscilará no máximo com coeficiente angular 1.*

*A solução de Lagrange, apesar de engenhosa permanece computacionalmente ineficiente uma vez que volta a usar o polinômio clássico para encontrar o máximo de sua derivada...*

## 1.3 Polinômios Trigonométricos.

*Aproximação de funções* ou o item mais geral *Teoria da Aproximação* é um capítulo imenso em Matemática e que apenas cresce nos dias atuais por sua importância natural. Nesta seção vamos deixar um pequeno testemunho de um dos tópicos importantes dentro da área de *aproximação de funções* : aproximação com polinômios trigonométricos. Embora os *polinômios trigonométricos* produzam aproximações de funções ou de dados discretos, esta não é a sua principal aplicação. Vamos apresentá-los aqui por duas razões:

1. um outro tipo de aproximação: Os polinômios trigonométricos oferecem aproximação do ponto de vista da energia do fenômeno ou do ponto de vista da integral.
2. uma razão histórica: Os polinômios trigonométricos são a base histórica das *wavelets*. Se podem estudar estas últimas sem passar pelos polinômios trigonométricos, mas se perde um dado evolutivo e intuição.

Para que você tenha uma ideia da superficialidade do que trataremos aqui, um dos livros mais famosos sobre o assunto, escrito por Antony Zigmund (1900-1993), sob o

(fig.1.7), página 54, os problemas deste método simples. O polinômio cujo gráfico está representado nesta figura traz como coeficiente lider 0.0005 do contrário o programa não conseguiria deixar visiveis os gráficos simultâneos do polinômio e do *sen*. Os polinômios são expressões relativamente faceis de se obter, entretanto eles tem uma oscilação violenta, veja [22] a respeito.

A determinação do polinômio que resolve esta questão se reduz à solução de um sistema de $n$ variáveis, *os $n$ coeficientes do polinômio* com $n$ equações obtidas a partir dos $n$ pontos dados.

**Definição : 4** *Interpolação polinomial.*

*Se chama* intepolação polinomial *a determinação de uma curva,* gráfico de um polinômio *, passando por um conjunto de pontos dados. Se diz ser o polinômio que interpola os pontos dados.*

A definição acima é antiga, hoje se estende o conceito de *interpolação polinomial aos splines assim como a qualquer aproximação polinomial por pedaços.* Adotaremos esta maneira de pensar quando nos referirmos a interpolação polinomial. Esta forma estendidade de entender *interpolação* coincide com o conceito de aproximação com que esta palavra já foi utilizada desde há séculos passados e assim se recupera a ideia de aproximação .

Mas os polinômios tem dois defeitos que forçaram a busca de outros métodos. O primeiro é que oscilam violentamente quanto maior for o seu grau. Use o programa *Grafun* e nele coloque como equação da função $f$ um polinômio de grau elevado. Se for muito elevado possivelmente o programa nem siquer vai rodar, (este é o segundo defeito). Como *Grafun* não faz nenhuma *maquilagem* no gráfico você irá ver algumas "*retas*" surgirem na tela se o gráfico for feito no intervalo em que se encontram as raizes. Isto porque entre as raizes os polinômios de grau alto oscilam violentamente.

A solução do exercício, caso parábolas, é:
há seis equações e seis incógnitas, (o sistema é determinado):

1. $P_1(a) = f(a)$
2. $P_1'(c) = P_2'(c) = -P_1'(a) = -P_2'(b) = \rho$
3. $P_2(b) = f(b)$
4. $P_2(c) = f(c) = P_1(c)$

observando que na segunda bateria de equações estamos usando a simetria das parábolas em torno do seu eixo que se reflete em anti-simetria para o coeficiente angular.

Se considerarmos que $\rho$ é uma variável, o sistema tem soluções mas é sub-determinado (tem o número de equações menor que o número de incógnitas. A indeterminação se chama de *grau de liberdade* por que nos permite diversas soluções e está representado na equações acima na escolha do coeficiente angular $\rho$ da reta $r$ que passa por $c$.

Voltaremos a este problema inspirado no que faremos na próxima secção.Mas o algoritmo desenvolvido no programa Spline00 lhe permite construir 2-splines sendo suficiente incluir nele uma saida de dados para informar ao usuário quais são os coeficientes calculados para os polinômios. Como está, o programa resolve o problema apenas graficamente, (calcula os coeficientes, mas não os escreve ...).

O conjunto dos coeficientes dum spline *ou quasi-spline*, recebe o nome de *matriz* de dados do spline ou *do quasi-spline*, ver (definição 2), página 18.

### 1.2.4 Interpolação polinomial clássica

A interpolação polinômial tem uma motivação simples que justifica que tenha sido muito popular: *se conhecermos, por uma razão qualquer, n pontos pelos quais passe o gráfico de uma função então podemos encontrar um polinômio do grau n cujo gráfico passa também por estes pontos.* Veja na figura

*ponto c. Escreva o sistema de equações correspondentes chamando de $P_1$ e $P_2$ aos arcos de polinômio do terceiro grau envolvidos.*

7. ** Enuncie e resolva os problemas anteriores com 3 pontos não colineares no plano em lugar de* $\{a, c, b\}$

Os exercícios acima oferecem de forma geométrica um algoritmo para construção de 2-splines, 3-splines inclusive mostrando a infinidade deles quando submetidos a tres condições com uma condição livre que representa a inclinação da reta $r$ que passa por $c$, (solução de um dos exercícios). Para os 2-splines os triângulos são isósceles. Para 3-splines, escalenos. O gráfico (fig. 1.6) ilustra estas ideias.

O argumento geométrico se explica melhor no caso de uma bi-partição de um intervalo [a , b] em que está definida a função f. Então o 1-spline que aproxima a função forma uma poligonal com tres lados. Qualquer par de triângulos construidos nos moldes do exercício anterior produz 2 ou 3 splines... Observe que estamos aproximando uma função usando os seus zeros como valores conhecidos: $(a, 0), (c, 0), (b, 0)$ são os tres pontos dados da função $f$. Ver o gráfico acima. A reta pode ser movimentada fazendo que as "bolhas" correspondentes às duas parábolas aumentem por igual.

No caso de 3-splines, é possivel mover a reta e aumentar as bolhas de maneiras diferentes porque é possivel manter condições de secância diferentes nos extremos dos intervalos, mantendo o ponto $c$ como ponto médio.

É esta propriedade geométrica destes polinômios por pe-daços, que lhes deu o nome de spline, um instrumento maleável de desenhistas para produzir curvas passando num ponto com um coeficiente angular escolhido.

O exercício pode ser *algebricamente* estendido para resolver o problema $(a, f(a)), (c, f(c)), (b, f(b))$ em que agora não se supõe mais que $f(a) = f(c) = f(b) = 0$.

tal instrumento usado por desenhistas para moldar curvas que passem por pontos determinados no plano).

**Exercício: 5** *Arcos diferenciavelmente soldados*[7].

1. *Trace um segmento de reta* $[a, b]$ *bipartido pelo ponto* $c$. *Isto é* $c$ *é o ponto médio do intervalo. Trace uma reta* $r$ *qualquer obliqua passando por* $c$. *Mostre que existem exatamente dois triângulos isósceles formado por retas passando por* $a$ *e por* $b$ *e* $r$ *nos quais se podem inscrever parábolas diferenciavelmente soldadas em* $c$. *Justifique porque os triângulos tem que ser semelhantes. Ver o gráfico na figura (fig. 1.6)*

2. * *Mostre com um argumento geométrico que o* número de arcos de parábola *do exercício anterior é infinito, se considerarmos que o coeficiente angular* $\rho$ *da reta* $r$ *varia.*

3. * *Chame de* $P_1$ *e* $P_2$ *aos arcos de parábola mencionados acima e construa o sistema de equações que resolve o problema algébricamente. Conclua que o sistema é determinado e solúvel, (número de equações corresponde ao número de incónitas e o sistema tem solução ).*

4. **generalização dos exercícios anteriores. Considere um ponto* $c$ *que divida o intervalo* $[a, b]$ *numa razão* $r$ *qualquer:*$\frac{c-a}{b-c} = r$. *Mostre que se* $r \neq 1$ *os triângulos mencionados no exercício 1 não serão semelhantes. Construa as parábolas, faça os gráficos.*

5. *Mostre que para um 3-spline as retas que passam pelos pontos* $a$ *e* $b$ *são também infinitas, (falamos agora de arcos de polinômios do terceiro grau). Mas que o ponto* $c$ *não precisa mais ser o ponto médio. Qual o grau de liberdade do sistema?*

6. *Inclua, no exercício anterior, condições sobre o coeficiente angular nos extremos do intervalo de modo a reduzir o grau de liberdade do sistema à inclinação da reta que passa no*

[7] os exercícios marcados com asterisco são "teóricos", decida se lhe convém fazê-los.

Como é uma média aritmética, o resultado geométrico é o segmento de reta que une os pontos

$$(a_j, f(a_j)) \quad , \quad (a_{j+1}, f(a_{j+1})).$$

Em outras palavras, dados n pontos conhecidos de uma função $f$:

$$f(a_0), ..., f(a_n)$$

num intervalo, 1.1 produz a poligonal que une estes pontos. Uma poligonal é um polinômio de primeiro grau por pedaços e contínua.

As funções características $\chi_{[a_j,a_{j+1}]}$ que multiplicam as parcelas em 1.1 tem um efeito localizador: garantem que nenhuma das funções do primeiro grau *existam* fora do seu intervalo "característico". Uma outra maneira de escrever esta expressão seria substituir a soma por uma sucessão.

Este método para construir splines é muito penoso como vamos mostrar no próximo pacote de exercícios. No capítulo 5 veremos uma outra construção de splines que é menos penosa, obviamente considerado que teremos aprendido mais coisas para chegar até lá...

Se tirarmos o adjetivo *convexa*, que apenas caracteriza o fato de que os coeficientes são menores que 1, positivos e cuja soma é 1, temos simplesmente uma combinação linear, logo resultando nos pontos de uma reta. Desta forma $S_1$ é uma função cujo gráfico é formado de segmentos de reta que ligam os pontos $(a_j, f(a_j))$ sendo assim uma interpolação linear destes pontos conhecidos de $f$. Vemos assim outra forma de entender a interpolação linear.

Um 2-spline associado a função $f$ seria uma "soma" de polinômios do segundo grau localizados nos sub-intervalos de uma partição usando condições sobre derivadas nos pontos dados além do valor da função nos pontos. Isto pode ser feito com um processo de integração sucessiva, por exemplo. Vamos evitar este método aqui. Em vez disto vamos ilustrar geométricamente a construção de splines que inclusive justificam o seu nome, (o

Para construir um spline, basta tomar uma função constante por pedaços e calcular sucessivas primitivas da mesma. A primitiva de ordem $k$ será um $(k)$-spline. O prefixo inteiro é utilizado para caracterizar o grau dos polinômios que constituem o spline.

Vamos analisar a construção de um algoritmo para construir splines.

Observemos que se conhecermos $f(a_0), ..., f(a_n)$ uma amos-tra dos valores de uma função num intervalo $[a, b]$ ; $a_0 = a$ e $a_n = b$ então o *único* 1-spline que aproxima esta função será a função linear por pedaços definida por

$$\begin{aligned} S_1(t) &= \sum_{i=0}^{n-1} [\tfrac{f(a_{j+1})-f(a_j)}{\Delta x_j}(t - a_j) + f(a_j)]\chi_{[a_j,a_{j+1}]}(t) = \\ &= \sum_{i=0}^{n-1} [\tfrac{t-a_j}{\Delta x_j} f(a_{j+1}) + (1 - \tfrac{t-a_j}{\Delta x_j}) f(a_j)]\chi_{[a_j,a_{j+1}]}(t) \end{aligned} \tag{1.1}$$

a segunda linha na equação acima é da forma:

$$s \cdot f(a_{j+1}) + t \cdot f(a_j) \ ; \ s, t \geq 0 \ ; \ s + t = 1 \tag{1.2}$$

em cada sub-intervalo da partição considerada em $[a, b]$. Uma soma deste tipo se chama de uma combinação linear convexa dos números $f(a_{j+1}), f(a_j)$ ou ainda uma média aritmética ponderada com pesos $s, t$. Quem faz o papel dos peso $s, t$ são os quocientes $\frac{t-a_j}{\Delta x_j}$ porque, como o tamanho do sub-intervalo em que estão $t, x_j$ é $\Delta x_j$.

**Observação: 6** *Média aritmética ponderada.*

A forma desta média aritmética é pouco usual porque os pesos já são menores que 1, mas o leitor pode facilmente ver que se trata de fato da média aritmética comum e corrente. Comumente se apresentam os pesos como números maiores que 1 e muito frequentemente como inteiros, mas se dividem as médias pela soma dos pesos, depois... É bem extranho que os *economistas* usem este método no cálculo da inflação, pesos inteiros, se usassem pesos menores que 1 poderiam confundir mais facilmente as pessoas, como parece ser o objetivo pelo menos dos economistas a serviço da classe dominante, se não o fazem deve ser por pura ignorância...

### 1.2.3 Splines.

*Esta seção se encontra aqui apenas por uma questão de compatibilidade com o restante do capítulo, afinal falamos de* splines *e seria incômodo nada construir a respeito deles. A construção original dos splines , ver [22], é bastante confusa e difícil, Aagu vamos lhe apresentar uma construção geométrica que inclusive lhe justifica o nome. No capítulo 5 uma outra construção será feita com uma metodologia diferente e atual e que inclusive conduz rapidamente ao núcleo da ideia por traz dos splines. Obviamente, será preciso mais material teórico para conseguir uma nova visão destes objetos*[5]*, é preciso completar a ideia, os splines não são difíceis o que é difícil é provar as suas propriedades.*

*Você pode saltar esta secção sem nenhum prejuizo para o resto do livro.*

Por volta de 1950 foram criados os splines, oficialmente[6]. Mas foi preciso chegar a década de 70 para que eles fossem notados como um objeto interessante, a razão disto é que sem computação o uso dos splines e mesmo qualquer construção que os envolva é impraticável.

**Definição : 3** *k-Spline. Um k-spline é uma função polinomial de grau k por pedaços cuja classe de continuidade é* $\mathcal{C}^{k-1}$ *.*

As poligonais que interpolam $n$ pontos no plano são um exemplo de 1-spline. O que chamamos de 1-quasi-spline coincide com os 1-splines.

O nome *spline* vem de um instrumento de arquitetura constituido de uma *régua* de borracha contendo em seu interior uma peça de ferro flexível e que pode ser encontrada nas livrárias ainda hoje. Com esta régua se podem traçar curvas passando por uma quantidade qualquer finita de pontos, isto se chama de interpolação .

[5] diziam os franceses, para convencer os alunos a estudarem, *il n'y a pas de méthodes faciles pour faires les choses dificiles*, não tem jeito de se fazerem fáceis as coisas dificeis...

[6] datas são os elementos da *história oficial*, se diz que as *wavelets* foram oficialmente criadas em 1985...mas Haar pensava nelas em 1900.

18. *Verifique que na questão anterior a escolha do intervalo $I$ não é única. Escolha $I$ tal que $F(x) = \int_{-\infty}^{x} f(t)dt$ seja constante fora do intervalo $I$, e a medida de $I$ seja a mínima possivel.*

19. *Duas parábolas $P_1$ e $P_2$ se cortam no ponto $(-\frac{1}{2}, \frac{1}{4})$ com o mesmo coeficiente angular $\frac{1}{2}$, isto é, são tangentes uma a outra neste ponto. Verifique de quantas maneiras isto se pode dar e determine dois arcos de parábola que satisfaçam estas condições e que, além disto, tenham concavidades diferentes no ponto $(-\frac{1}{2}, \frac{1}{4})$. Faça os gráficos.*

20. *Construa um quasi-spline diferenciável de grau mínimo para a função $f$ obtida amostralmente no intervalo de tempo $[-5, 5]$ e cuja listagem de dados se encontra no (exemplo , 5 ), página 20.*

21. *Considere o seguinte método para determinar uma aproximação quasi-spline de $f$, no intervalo $[a, b]$ no qual se tem a partição determinada pelos nós $\{a = x_0, \cdots, x_n = b\}$:*

    - coincidir com $f$ nos pontos $x_i, x_{i+1}$ .
    - diferenciabilidade nos extremos: ser-lhe tangente nestes pontos.
    - condição sobre curvatura: sua curvatura no intervalo $[x_i, x_{i+1}]$ seja dada pelo *Teorema do Valor Médio* aplicado às derivadas de $f$ nos extremos dos sub-intervalos.

    (a) *Qual é o grau do quasi-spline assim obtido?*

    (b) *Interprete geométricamente o método e aplique-o nos dados amostrais que definem $f$ na tabela (exemplo, 5), da página 20.*

22. *Sabemos que $g'(x) = x\sin(x)\cos(x)$ e que se $x \in \{1, 3, 5, 7, 9\}$ então $g(x) \in \{0, -0.972553, 0.753933, -0.274230, -1.855043\}$. Encontre um quasi-spline de grau mínimo que aproxime $g$ e calcule sua integral no intervalo $[1, 9]$.*

- *$Max(f), Max(|f - S_4|)$ divido pelo comprimento do intervalo de definição de f.*
- *Analise em particular qual é objetivo do item anterior.*

13. *Modifique o programa SpliExp para que ele se ocupe de todos os levantamentos de dados e lhe faça um relatório final circunstanciado, (use o computador como um laboratório...)*

    Sugestões : Use Case of, (veja GaPlana) e faça o programa selecionar assim diversas equações e repetir as experiências para cada uma delas.

14. *Procure adquirir uma rotina que lhe permita ler equações pelo teclado. Use-a para tornar os seus programas mais inteligentes.*

15. *Construa um quasi-spline de grau 3, $P_3$ para a função*

$$f(x) = 2 + \frac{1}{3}\sin(x/3) + \frac{\cos(3x)}{2}$$

    *no intervalo $[-5, 5]$ com uma partição uniforme do mesmo em 10 sub-intervalos.*

    - *Seu programa deve imprimir ou gravar em disco a matriz de dados do 3-quasi-spline.*
    - *Calcule a integral de $f$ e de $P_3$ no intervalo dado.*
    - *Encontre todas as raizes de $f$ e de $P_3$ no intervalo dado.*
    - *Repita o exercício com 20 nós na partição e compare os resultados com os anteriores.*

16. *Estenda a função $f(x) = (x+1)^2(x-1)^2$ para além do intervalo [-1 , 1], continuamente, por funções constantes. Chame esta nova função de $f$ e verifique se $f'$ , $f''$ são contínuas. Faça o gráfico de $f$ e de $f'$.*

17. *Encontre uma constante $k$ e um intervalo $I$ adequados tal que $f(x) = \cos(x) + K$ se possa estender no complementar $I^c$ de $I$ continuamente por uma função identicamente nula. Chame esta nova função de $f$ e verifique se $f'$ , $f''$ são contínuas. Faça os gráficos de $f$ e de $f'$.*

5. *Inclua linhas no* loop *para fazer gráfico, para calcular integral, e o máximo da diferença entre f e o quasi-spline que a aproxima.*

6. *Use o erro da fórmula de Taylor, erro integral e erro diferencial, e compare com o erro obtido no programa anterior: calcule-os no programa e mande imprimir os dados.*

7. *Modifique o programa para obter um 4-quasi-spline de maneira análoga na definição do quarto coeficiente, para que o resultado seja de classe* $\mathcal{C}^1$.

8. *Modifique o programa Spline00 para obter um 6-quasi-spline de classe* $\mathcal{C}^2$ *re-aproveitando os cálculos para obter derivada e segunda derivada nos dois extremos do intervalo.*

9. *Use o programa SpliExp, nele defina alternadamente* $a_2$ *igual a zero ou com sinal alternado e multiplicado por um pequeno fator. Analise os resultados das diversas experiências por ele executadas com distintas funções .*

10. *Repita as experiências anteriores omitindo o fator* $\frac{1}{n}$ *no cálculo de* $a_2$. *Analise os resultados contra aplicações em diversas funções . Compare com os casos anteriores.*

11. *Substitua a linha que define* $a_2$ *por comando lógico que, para cada novo uso do programa, alterne a definição deste coeficiente definindo como zero ou com sinal alternante. Ai mesmo inclua a definição da mensagem que informará ao usuário se a experiência é feita com* convexidade alternada *ou não .*

12. *Repita as experiências acima fazendo uma tabulação associe*

    - *Ponto de* $Max(f'), Max(|f - S_4|)$
    - *Tempo de processamento.*
    - *Precisão obtida.*

A única saida é a história do fenômeno. Se ela indicar um valor para $K$, a constante de Lipschiz, então o erro que nos expomos com os dados será $K \cdot ||\Pi||$. Por exemplo, é impossível refazer as medidas obtidas por um ondógrafo, da variação das ondas do mar de um determinado dia. Feitas as medidas é tudo que se pode ter, entretanto todos na região sabem qual é a máxima variação das ondas do mar e portanto $K$ é conhecida, e a estimativa pode ser feita com segurança.

Se os dados podem ser medidos diversas vezes então existe uma sucessão $K_n$ e podemos verificar se $K \cdot ||\Pi_n|| \to 0$. Se isto for *verdadeiro* diremos que o experimento é *convergente* o que significa que a sucessão de Cauchy $\mathcal{P}_n$ existe e se aproxima de uma função "ideal" $f$.

A preocupação com a continuidade na construção de modelos é fácil de se explicar com o Teorema de Weirstrass citado acima: nós sabemos que as funções contínuas se podem aproximar por funções também contínuas e a nossa hipótese nos diz que devemos sempre esperar que os fenômenos sejam representáveis por funções contínuas, e até mesmo diferenciáveis. Os quasi-splines são contínuos e, como são polinomiais, são também *computacionais*: fáceis de serem representados num programa de computação . Os exercícios seguintes devem conduzí-lo, experimentalmente, a compreender a avaliação do erro.

**Exercício: 4** *1. Introduza uma rotina para calcular o erro entre f e o seu quasi-spline e faça uma estatística deste erro para diversas funções .*

*2. Calcule por varredura o máximo de $f'$, e o ponto em que ele ocorre e compare com o resultado obtido no exercício anterior.*

*3. Use os coeficientes de Taylor para encontrar um quasi-spline de grau dois a partir de uma massa de dados amostrais $\mathcal{A} = (a_{ij})$ usando como aproximação de $f''(x)$ a media das diferenças de segunda ordem dos elementos da coluna da matriz $\mathcal{A}$ 2.*

*4. Modifique o seu programa para obter polinômios de grau tres calculando um quarto coeficiente com a condição do polinômio ter o valor $f(x_{i+1})$ no ponto $x_{i+1}$ portanto forçando a continuidade dos polinômios por pedaços. Encontre uma maneira de resolver a falta de dados nos extremos.*

**Dem**:

Vamos chamar de $f : [a, b] \to \mathbf{R}$ a função cujos dados amostrais se tem e que se deseja aproximar por k-quasi-splines.

O método de Riemann para integração e o Teorema da Aproximação de Weirstrass demonstram a existência de uma sucessão de k-quasi-splines convergindo para uma função contínua. O método de integração de Riemann também garante que se $f$ for integrável, então podemos produzir uma sucessão de Cauchy que convirja para $\int_a^b f(t)dt$ usando apenas partições uniformes[4]. Isto nos permite usar partições uniformes sem perda de generalidade.

Resta-nos demonstrar a condição sobre o erro. Vamos para isto considerar duas sucessões, $\mathcal{P}_n, K_n$ em que

- $\mathcal{P}_n$ é o k-quasi-spline associado a uma partição uniforme com com $m + 2^n$ sub-intervalos de medida $\frac{b-a}{m+2^n}$;
- $K_n$ é a sucessão das constantes de Lipschiz de cada termo da outra sucessão. O número $m$ é a quantida inicial de sub-intervalos antes que se inicie um processo de bipartição.

Como $K_n$ é uma aproximação da constante de Lipschiz da função $f$ e $\mathcal{P}_n$ é tangente ao gráfico de nos extremos de cada sub-intervalo, então o gráfico de $f$ se encontra contido em uma vizinhança de amplitude $K_n||\Pi_n||$ do gráfico de $\mathcal{P}_n$.

Como

$$K_n||\Pi_n|| = K_n \frac{b-a}{2^n} \to 0$$

porque $f$ é integrável, então então, dado um erro $\epsilon$ podemos encontrar $n$ tal que $\mathcal{P}_n$ se encontre numa faixa de erro uniforme $\epsilon$ em volta de $f$.

q.e.d.

## Observação: 4 *Hipóteses ocultas na demonstração do teorema*

No teorema (3) está implícita uma hipótese que na prática de levantamente de dados é irreal: que podemos ir fazendo refinamentos sucessivos da partição do intervalo de tempo em que um fenômenos está sendo observado.

Esta *ausência de realidade* não invalida (teorema, 3), porque este teorema representa apenas a garantia de que o método de aproximação por *quasi-splines* funciona. Se a única partição que pode ser feita é aquela em que foi feita a *aquisição* dos dados, tudo que se pode obter é uma estimativa grosseira $K \cdot ||\Pi||$ em que $K$ *é uma estimativa* da constante de Lipschiz e $||\Pi||$ é a norma da partição.

## Observação: 5 *Adequação à realidade...*

O que a prática nos oferece com frequência é uma simples amostragem de dados de $f$, ou do fenômeno que esta função representa. Este é o conjunto de precisão de que já falamos anteriormente, os pontos em que o fenômeno foi quantificado. Que precisão podemos ter para estes dados?

Frente ao que vimos acima a resposta é nenhuma. Um único levantamento de dados não oferece absolutamente nenhuma segurança. Somente uma série estatística é que pode oferecer uma resposta para uma tendência.

[4] se uma função for integrável, qualquer cadeia de partições cujas normas tendam a zero, produzem uma sucessão convergindo para a integral, em particular cadeias de partições uniformes obtidas por bi-partição, por exemplo.

Esta análise sugere que podemos encontrar um ponto, $p$, e uma vizinhança deste ponto $v(p)$, onde se dá a maior oscilação de uma função:

**Teorema: 2** *da vizinhança de um ponto de alta oscilação. Seja* $f : [a, b] \to \mathbf{R}$ *uma função contínuamente diferenciável. Então existe uma* melhor *constante* $K$ *tal que*

$$|f(x) - f(y)| \leq K|x - y|$$

*e um ponto* $p$ *numa vizinhança do qual se dá esta oscilação máxima.*

**Dem**: Como $f$ é contínuamente diferenciável, podemos escrever a desigualdade do teorema como

$$\frac{|f(x) - f(y)|}{|x - y|} \leq K$$

e vemos que se considerarmos sempre $y$ próximo de $x$ a expressão esquerda representa uma aproximação da derivada $f'(x)$ portanto a constante $K$ é o máximo de $f'$, que existe porque $f \in \mathcal{C}^1$ por hipótese. Como por hipótese também $f'$ é contínua, então existe pelo menos um ponto $p$ onde este máximo se realiza. q.e.d.

O teorema fala de uma *melhor* constante porque se $K' > K$ tanbém se aplica na desigualdade. Esta *melhor constante* se chama *constante de Lipschiz* de $f$ no intervalo $[a, b]$.

A análise feita acima nos conduz a um método semelhante ao da *integração no sentido de Riemann* que garante quando uma *função é integrável* e entre estas se encontram as funções contínuas. O método de integração de Riemann cria as condições para construção de sucessões de Cauchy equivalentes, qualquer uma delas convergindo para o valor da integral. O Teorema de Aproximação de Weirstrass garante a existência do limite. Temos, por semelhança o Teorema:

**Teorema: 3** *Se* $f$ *for uma função diferenciável definida no intervalo* $[a, b]$*, existe uma "sucessão de Cauchy" de k-quasi-splines de classe* $\mathcal{C}^1$ *que se aproximam uniformemente de* $f$ *o* erro uniforme *sendo* $K||\Pi||$ *em que* $K$ *é a constante de Lipschiz de* $f$ *e* $||\Pi||$ *é a norma da partição* $\Pi$ *do intervalo* $[a, b]$.

6. *mas* podemos comparar $\mathcal{P}_n$ e $\mathcal{P}_m$ em que $m > n$ são dois inteiros e $m$ está associado com um refinamento da partição a que está associado $\mathcal{P}_n$. Esta é revolucionária ideia de Cauchy, o *teste de Cauchy*: não precisamos conhecer o limite, se soubermos que uma sucessão é convergente, podemos encontrar elementos arbitrariamente próximos do limite, analisando apenas o comportamento da sucessão.

7. *mais* um dado para completar a análise, precisamos de um método para calcular o erro, veja a (fig. 1.4) e observe que os erros aumentam na vizinhança de 6 e de -4, onde a oscilação da função também aumenta. No intervalo $[-2.5, 2.5]$ o erro não é visível a partir do gráfico. A função que está sendo aproximada tem oscilações cada vez mais fortes a medida que $|x|$ cresce:

$$f(x) = 3sen(x^2 + 4) + 2x.$$

Quando a função oscila, a derivada cresce... é o caso *diametralmente* oposto ao das funções cujo gráfico é uma reta. Temos que estudar o comportamento da derivada, para descobrir onde o erro se acentua. Analise também a (fig. 1.5) onde a partição foi refinada e veja que o erro já não é mais visível a partir do gráfico. A versão de *polsegpedcon* que permite esta simulação é:

```
function [y] = polsegpedcon(x,den)
if or([x<-4 ; x>=6]) then y = 0
else p = (floor(den*x))/den; a=6*p*cos(p^2+4)+2;
b = den*den*(f(p+(1/den)) -  a/den - f(p));
y = f(p) + a*(x-p)+ b*(x-p)^2;
end
end,
```

em que o paramétro $den > 1$ é usado para refinar a partição. O código é do [21].

*módulo da diferença das duas funções. Esta última forma de medir caracteriza a diferença de energia entre as duas funções.*

*Aqui está a dificuldade em usar o* Teorema da aproximação de Weirstrass, *como vamos saber qual é o erro máximo, se todos os valores que temos são precisos,* estamos trabalhando com o conjunto de precisão...

Mas as funções são objetos que tem valor numa infinidade de pontos e *erro* aqui pode significar:

1. erro em cada ponto de um intervalo;
2. *mas*, ... analisar o erro em cada ponto do intervalo pode não ser possivel, então talvez possamos calcular o erro médio, ou o erro máximo;
3. *mas*, ... como analisar os erros se tudo que conhecemos da função são alguns valores medidos nos nós de uma partição do intervalo de tempo em que o fenômenos foi observado. Ora, se os dados experimentais que tivermos definirem todos o mesmo coeficiente angular, então eles se encontram sobre uma reta e a interpolação linear definirá uma aproximação de erro uniforme zero;
4. como é muito esperar que os dados sejam lineares como acima se propõe, vamos considerar que eles oscilem, isto é que
$$\frac{y_{i+1} - y_i}{x_{i+1} - x_i}$$
seja uma sucessão não constante. No intervalo $[x_i, x_{i+1}]$ em que a variação for maior, mesmo pelo raciocínio linear acima, é onde teremos a maior probabilidade de erro, e portanto onde se encontrará a sugestão de *erro uniforme*. Se o experimento puder ser repetido com partição mais fina, poderemos localizar melhor onde se dá está alta variação e assim obter uma estimativa mais precisa do *erro uniforme*.
5. *mas*, tudo o que temos é um k-quasi-spline $\mathcal{P}_n$ e não temos nenhuma chance de compará-lo com a função $f$;

*encontra em nossa coleção de programas.*

Com esta hipótese o nosso grau de liberdade aumenta sensivelmente porque as funções mesmos as descontínuas, podem ser aproximadas uniformente por funções de classe $\mathcal{C}^\infty$ com alguns cuidados a serem tomados nas extremidades dos intervalos e nos pontos de salto onde a aproximação uniforme falha. O instrumento teórico em que nos apoiamos é um Teorema do fim do século passado, de Weirstrass, [20, cap 15, pag. 336]:

**Teorema: 1** *Teorema da aproximação de Weirstrass.*
*As funções contínuas se podem aproximar uniformemente sobre intervalos fechados, por polinômios.*

Aqui se encontra o cuidado com os extremos: eles tem ser incluidos na determinação dos polinômios, como fizemos nos algoritmos descritos, é a razão dos intervalos fechados do *Teorema da aproximação de Weirstrass.*

**Observação: 3** *Aproximação uniforme.*
*O* Teorema da aproximação de Weirstrass *é uma garantia de que podemos fazer aproximações, entretanto ele não nos oferece um meio de verificarmos os erros de nossas aproximações. Quando fazemos aproximações, precisamos saber qual é a margem de erro com que trabalhamos.*

*Vamos logo discutir o que é* uniforme*. Quando calculamos valores aproximados para números, existe uma única forma de analisar o erro cometido que na prática se resume em indicar a partir de que* casa decimal *se perde a confiabilidade. Como as funções tem valores em todos os pontos de um intervalo* $[a, b]$*, em tese poderiamos calcular o máximo dos erros cometidos e depois afirmar que todos os demais erros são menores ou iguais a este máximo.* Erro uniforme *quer dizer isto, uma margem de erro que vale para todos os pontos em que o fenômenos foi medido. Há várias outras maneiras de medir-se o erro, ou a distância entre duas funções, uma delas é o valor da integral do*

**Exemplo: 6** *Funções que passam por um conjunto de pontos do plano.*

*Marque alguns pontos no plano correspondentes a valores sobre os nós de uma partição dum intervalo* $[a, b]$ *dado. Quer dizer, para cada nó* $x_i$ *de uma partição* $\Pi([a, b])$ *escolha um valor* $y_i$ *de modo a ter* $n$ *pares* $(x_i, y_i)$*. Você pode agora desenhar duas funções* $f, g$ *que sejam fortemente diferentes uma da outra mas ambas passando pelos* $n$ *pontos* $(x_i, y_i)$*. Você pode até mesmo impor a condição de que tanto* $f(x_i) = g(x_i)$ *como* $f'(x_i = g'(x_i)$ *para todos os nós e ainda assim tê-las muito diferentes: por exemplo suas integrais difiram muito. Mas em qualquer dos casos* $f, g$ *que você tenha desenhado, refinando a partição* $\Pi([a, b])$ *mesmo a aproximação linear por pedaços se torna gradativamente melhor.*

*Experimente graficamente para se convencer. Quem nos garante isto é o Teorema de Weirstrass que enunciaremos em seguida.*

*Entretanto, existem algumas hipóteses razoáveis às quais podemos aderir, dentro de uma concepção mais ampla de* teoria de modelos *que se pode dizer que é a Matemática. Uma delas é a* hipótese de alta diferenciabilidade dos fenômenos, *que num certo sentido se opõe tecnicamente e filosoficamente a da escola do* caos *quanto a construção de modelos. Ambas as tendências tem bons resultados e críticas mútuas bem fundadas... aderir a uma ou a outra termina sendo uma questão de gosto pessoal. Nós preferimos admitir que os fenômenos são altamente diferenciáveis, sem dúvida as técnicas disponíveis ficam bem mais confortáveis dirão os* outros *com um certo ar de superioridade, e é verdade...*

**Hipótese: 1** *Hipótese da suavidade dos fenômenos.*

*Os fenômenos têm, em quase-todos os pontos, em que podemos medí-los, um comportamento que pode ser modelado por uma função de classe* $\mathcal{C}^1$.

*Este exemplo geométrico pode ser repetido computacional com qualquer programa spl*.pas dos que você pode*

acompanha este livro. Você, alternativamente, pode fazer um *download* do arquivo *pas.zip* em *ftp.super.furg.br/pub/calculo.*

Rodando o programa Spline00 você terá uma visão concreta de como se dá esta aproximação de dados obtidos amostralmente. O programa lhe permite diversas experiências para testar o modelo de aproximação uma vez que *usa funções algoritmicamente bem definidas em Pascal* e os gráficos simultâneos da aproximação e da própria função evidenciam, graficamente, os erros.

A partir deste momento não vamos mais garantir-lhe que os programas que se encontram no disco, façam exatamente o que dizemos no texto do livro. Tendo chegado a este ponto você deve ser considerado, computacionalmente, um adulto, *esta é a sina de todos nós, em algum momento nos consideram adultos e responsáveis por nossas próprias ações ....* Nossos programas passam a ser apenas uma base para que você construa os seus próprios, se você julgar que ainda precisa de algum apoio. Também você poderá fazer uso de programas de simulação numérica que existem, *se* não quiser fazer os seus próprios programas. Neste caso sugerimos [21] como uma excelente alternativa, ou [8], ambos são programas de domínio público e podem ser encontrados nos locais citados na bibliografia. Os nossos programas também se encontram disponíveis, o código está aberto, em *ftp.super.furg.br/pub/calculo.*

**Erro na aproximação por 3-quasi-splines.**

Vamos aqui discutir a margem de erro na aproximação por quasi-splines. Há poucos dados para esta discussão , se quisermos nos colocar num ambito muito geral de aplicações. Não teriamos como discutir o tamanho de um erro de aproximação de uma *função qualquer.* Habitualmente se impõem condições sobre a *oscilação* da função como hipótese para que se possa garantir alguma forma de aproximação. Faça você mesmo a seguinte experiência:

*diferença entre os dois resultados. Inclua também o cálculo fórmla da integral e verifique qual dos métodos fornece melhor aproximação . Repita o teste com funções que tenham grande oscilação, por exemplo que tenham um termo* $sin(n * x)$ *ou* $cos(n * x)$.

6. *Modifique o programa anterior para gerar um polinômio como pontências de* $(x-a)$ *para um valor* $a$ *dado, à semelhança de um polinômio de Taylor.*

7. *Construa um programa que leia os valores de uma função nos nós de uma partição do intervalo* $[a, b]$ *de definição da mesma.*

8. *Melhore o programa anterior para fazer a leitura dos valores de uma função e além disto com passo fino ler valores muito proximos de cada nó. Encontre uma solução para eventual falta de informações no último nó.*

9. *Melhore os programas anteriores guardando os dados numa matriz e gerando polinômios de Taylor de grau tres desenvolvidos em cada nó calculando as derivadas aproximadamente.*

10. *Construa um programa que produza um 3-quasi-spline para uma função dada a partir de uma amostra dos valores desta em nós de uma partição do seu intervalo de definição , acréscidos de* micro-nós *na vizinhança de cada nó.*

11. *Use os valores finos próximos dos nós para calcular uma derivada segunda com diferenças de segunda ordem, produza um 5-quasi-spline de classe* $\mathcal{C}^2$ *. Veja as simulações feitas em [16, cap. 2, derivação aproximada] para se convencer de que este exercício pode representar algo inútil do ponto de vista de aproximação. É apenas um exercício sobre polinômios...*

A solução destes exercícios se encontram em parte no arquivo *Tipos* e nos programas *splineK, SpliExp* no disco que

```
f(2.0)=  2.250
===medicao fina
f(2.1)=  2.33884      f(2.2)=  2.42623      f(2.3)=  2.51220
===
f(3.0)=  3.07692
===medicao fina
f(3.1)=  3.15267      f(3.2)=  3.22727      f(3.3)=  3.30075
===
f(4.0)=  3.78571
===medicao fina
f(4.1)=  3.85106      f(4.2)=  3.91549      f(4.3)=  3.97902
===
f(5.0)=  4.4
```

Os exercícios que se seguem dirigem-no para a construção de quasi-splines. O objetivo maior é o dominio da técnica envolvida.

**Exercício: 3** *Construção de quasi-splines para dados amostrais.*

1. *Faça um programa que recebendo quatro números, produza um polinômio do terceiro grau. Desenvolva este polinômio num ponto genérico* $\underline{a}$ *e complete o programa para fazer o gráfico na tela.*

2. *Altere seu programa para que ele de um arquivo em disco os dados como os do (exemplo 5). Um tal programa estaria pronto para construir um quasi-spline produzido pela leitura de um aparelho desde que a formatação dos dados fosse a mesma.*

3. *Modifique o programa Simpson para calcular integrais com um 3-quasi-spline. Não se esqueça de trocar o nome do programa.*

4. *Modifique o programa anterior para calcular a integral da função representada pelos dados amostrais (exemplo, 5). Aplique o programa numa função cuja integral você saiba calcular e compare os resultados.*

5. *Inclua no programa feito no exercício anterior a rotina de cálculo da integral por Simpson e imprima ao final a*

É inútil tentar calcular derivadas aproximadas de ordem maior que dois. Ver as simulações feitas em [16, cap. 2, derivação aproximada] que mostram como a precisão se degrada muito. Ver [14, introdução ] em que inclusive um novo conceito se delinea, o de *derivação algorítmica*.

Uma última observação sobre o *algoritmo* do (exemplo, 4). O quase-spline assim definido é continuamente diferenciável uma vez que as derivadas à esquerda e à direita coincidem, isto com apenas quatro coeficientes que podem ser obtidos de dados como os da tabela do (exemplo, 5) seguinte.

**Exemplo: 5** *Função definida por dados amostrais.*

*Os dados abaixo foram obtidos por um programa em Pascal. Eles representam os valores de uma determinada função nos pontos indicados como argumento da função* `f`*. Os mega-nós estão aqui representados pelos números inteiros do intervalo considerado.*

```
 Intervalo:  [-5.0,5.0]
f(-5.0)= -12.8
===medicao fina
f(-4.9)= -12.29412    f(-4.8)= -11.80769   f(-4.7)= -11.33962
         ===
f(-4.0)= -8.5
===medicao fina
f(-3.9)= -8.14754    f(-3.8)= -7.80645     f(-3.7)= -7.47619
===
f(-3.0)= -5.42857
===medicao fina
f(-2.9)= -5.16901     f(-2.8)= -4.91667    f(-2.7)= -4.67123 \
===
f(-2.0)= -3.125
===medicao fina
f(-1.9)= -2.92593     f(-1.8)= -2.73171     f(-1.7)= -2.54217
===
f(-1.0)= -1.33333
===medicao fina
f(-0.9)= -1.17582     f(-0.8)= -1.02174     f(-0.7)= -0.87097
===
f(0.0)=  0.1
===medicao fina
f(0.1)=  0.22772      f(0.2)=  0.35294      f(0.3)=  0.47573
===
f(1.0)=  1.27273
===medicao fina
f(1.1)=  1.37838      f(1.2)=  1.48214      f(1.3)=  1.58407
===
```

*é a média das derivadas aproximadas no nó $x_{j+1}$, a semelhança de $a_{2,j}$ mas agora usando a outra ponta do sub-intervalo. Fizemos isto antes, ver (fig. , 1.3), alí os polinômios eram do segundo grau, e não calculamos nenhuma média das derivadas porque usamos derivação formal, (exata).*

**Definição : 2** *Matriz de dados de um quasi-spline.*
*A matriz mXn*

$$\mathcal{A} = (aMat[i][j])_{i=1..m j=1..n}$$

*que contém os coeficientes dos* segmentos de polinômios *obtidos por um algoritmo semelhante ao que temos acima em que são usados as informações sobre os valores ou as taxas de variação de uma função sobre uma partição de um intervalo, será aqui chamada de* Matriz de dados de um quasi-spline.

Falamos de *algoritmo semelhante* na definição, porque há muitas maneiras de construir-se um polinômio por pedaços.

De posse de uma tal matriz se pode recompor com um programa de computador o quasi-spline, temos assim um modelo para a função que teoricamente descreve um determinado fenômeno.

No algoritmo acima preferimos definir um 3-quasi-spline. O refinamente dos dados ainda nos poderia permitir o cálculo de uma derivada segunda. Bastava para isto que o instrumento disparasse a fazer refinamentos de distinta ordem de grandeza em cada nó isto nos permitiria criar um modelo na forma de um 4-quasi-spline. As diferenças de segunda ordem com os mesmos dados é também possivel e nós usamos este método nos programas. Entretanto a precisão que se pode conseguir com segundas derivadas não é grande e não compensa o esforço de ir tão longe. O fato é que as aproximações polinomiais se restringem às de grau tres na prática, e a razão no fundo volta a ser a da oscilação grande de polinômios de grau muito alto.

(a) *Supondo que o problema é o descrito acima, temos uma matriz com 3 dados refinados em cada coluna: uma matriz* $3 \times n$ *em que* $n$ *é o número de* mega *nós, e 3 é o número de* micro *nós associado a cada um dos nós principais:*

$$\mathcal{A} = (aMat[i][j])_{i=1..3 j=1..n}$$

(b) *O método consiste num uso modificado da fórmula de Taylor, faz uso do desenvolvimento de um polinômio em um dado ponto:* $\mathcal{P}(x) = A + B(x-a) + C(x-a)^2 + D(x-a)^3$. *Vamos usar polinômios do grau* 3 *para construir um 3-quasi-spline de classe* $\mathcal{C}^1$.

(c) *Em cada sub-intervalo* $[x_j, x_{j+1}]$ *define-se* $P_j$ *por*

i. $P_j = (a_{1,j}, a_{2,j}, a_{3,j}, a_{4,j})$ *em que estamos identificando um polinômio pela matriz dos seus coeficientes. Rompemos com um hábito multi-secular iniciando os índices do polinômio com* 1 *para coincidir com a notação matricial dos nós.*

ii. $a_{1,j} = aMat[1][j] = f(x_j)$, *é o valor da função no ponto inicial do intervalo.*

iii. $a_{2,j} = \frac{\sum_{i=1}^{2} \frac{aMat[i+1][j] - aMat[i][j]}{\Delta x'}}{2} \approx f'(x_j)$, *é a média das derivadas aproximadas no nó* $x_j$. Estamos usando o método da fórmula de Taylor para construir uma aproximação polinomial. A média serve de garantia contra erros de dados na medição e poderia ter sido usada também no cálculo de $a_{1,j}$.

iv. *Para obter* $a_{i,3}$ *resolvemos a equação*

$$P_j(x_{j+1}) = f(x_{j+1}) = aMat[1][j+1],$$

*a semelhança do cálculo de* $a_{1,j}$ *mas agora usando a outra ponta do sub-intervalo.*

v. *Para obter* $a_{i,4}$ *resolvemos a equação*

$$P'_j(x_{j+1}) = f'(x_{j+1}) \approx \frac{\sum_{i=1}^{2} \frac{aMat[i+1][j+1] - aMat[i][j+1]}{\Delta x'}}{2},$$

nossa linguagem, $f$ é uma função ideal que descreveria exatamente o fenômeno, enquanto que $P_n$ é a aproximação de ordem $n$ que conseguimos para *modelar* $f$ a partir de alguns dados que temos ou podemos deduzir de $f$. O parâmetro $n$ indica o número de nós da partição do intervalo $[a, b]$ onde a função está definida, este número será com frequência $n + 2^m$ porque começaremos nossas partições com um número baixo de nós e depois faremos sucessivas bi-partições dos sub-intervalos para ir em busca de mais precisão.

**Exemplo: 4** *Um processo de aquisição de dados.*

*O processo de aquisição dos dados pode ser o seguinte: Veja o exemplo 5, página 20 acompanhando a descrição deste processo.*

1. *Um aparelho monitora um experimento registrando os dados a cada intervalo de tempo* $\Delta x$ .

2. *No momento de cada registro, o aparelho faz uma série de medidas a pequeníssimos intervalos de tempo, por exemplo de tamanho* $\Delta x' = \frac{\Delta x}{100}$, *durante tres destes micro-intervalos.*

   Mais do que isto é inútil e anti-econômico, porque com pequenas subdivisões podemos chegar a uma derivação aproximada da função que desejamos modelar, entretanto é irreal conseguir-se derivadas de ordem superior com métodos numéricos. No caso multivariado estes micro nós levantariam coeficientes parciais de derivada em direções determinadas sobre um círculo de centro no mega-nó e em direções distintas escolhidas sobre este círculo.

3. *Tem-se, assim, não somente os valores de* $f$ *em cada um dos* mega nós $x$ *mas também os seus valores na família de* micro nós $x, x+\Delta x', x+2\Delta x', \ldots$ *.Isto nos permite calcular uma taxa de variação de* $f$ *na vizinhança de cada "*mega nó*" e assim ter* $f(x)$ *e* $f'(x)$ *para cada nó* $x$.

4. *Quatro coeficientes em cada sub-intervalo é o suficiente para construirmos um um 3-quasi-spline que sirva de modêlo para o fenôme-no que estamos estudando. Para fazê-lo* um *algoritmo é:*

Ora, para que fazer uma aproximação de uma função que se conhece tão bem? nos exemplos do Cálculo, estas funções são apresentadas inclusive com as respectivas equações ... Da mesma forma, objetos como os quasi-splines podem parecer ridículos ao fazermos uso deles para aproximar funções como

$$f(x) = 5sen(x/4) + x/4 + 1.$$

Veja que o programa *grafun* é muito mais simples e roda muito mais rápido que qualquer dos seus similares *grafunK*, exatamente porque ele usa a equação da função $f$ para encontrar os pontos $(x, f(x))$ que vai colocar na tela enquanto que os outros primeiro calculam um polinômio em um sub-intervalo, para aproximar a função por este polinômio no intervalo. Obviamente, se conhecermos a equação de uma função e se a linguagem de programação que usarmos tiver as funções de que ela se compõe como primitivas *algorítmicas*, não tem muito sentido procurarmos uma aproximação para $f$. *Nós já a conhecemos* !

Este parece ser um grave *defeito* na apresentação dos cursos de Cálculo... Com grande frequência se construem métodos de aproximação que se aplicam perfeitamente a funções altamente conhecidas, *como é o caso dos polinômios de Taylor*.

A razão desta aparente contradição se encontra na necessidade de exemplos com os quais se possam testar os métodos de aproximação , *controlar se produzem bons objetos aproximantes* . Se você rodar *grafun* e os programas *grafunK* poderá avaliar o grau crescente de aproximação que um k-quasi-spline produz para a função $f$ e inclusive deve ter observado que, maior k, menor é a quantidade de nós necessários para se obter a mesma boa aproximação . Ora esta constatação só é possível se tivermos um elemento de comparação. Esta é razão por que aplicamos os métodos de aproximação às funções bem conhecidas. Portanto não está errado o Cálculo nem o Cálculo Numérico, em fazer assim apenas ficam incompletos se não mostrarem que podemos recompor fenômenos com os modelos estudados. Esta incompletitude se deve à falta de condições didáticas, *(de financiamento)*, em que vivemos: sem laboratórios computacionais para o ensino do Cálculo e de Matemática em geral, o ensino tem que ficar pobre como está, e o nosso país continuará carente de tecnologia e de técnicos capazes de gerar esta tecnologia. Este é um problema político que nós, os cientístas, temos que denunciar a cada momento, ou alternativamente, aceitarmos a conivência.

A seguir veremos uma aplicação mais razoável deste instrumento tão poderoso que visivelmente produz bôas aproximações para as funções . A fórmula de Taylor encontra aqui diversas aplicações e voltará a ser usada no capítulo final sobre *equações diferenciais*.

**Quasi-spline: modelagem a partir de dados discretos.**

Invertendo o caminho, agora queremos construir um mo-dêlo polinomial para uma massa de dados que podemos ter obtido a partir de medições feitas por um aparelho. Precisamos encontrar uma função $P_n$, que descreva quantitativamente um fenômeno a partir destas medições, em que o índice $n$ indica que a função $P_n$ depende de uma sequência de operações e no presente caso foi obtida ao final de $n$ dessas operações . Em

veremos algumas formas mais simples desta ideia. A definição seguinte representa apenas uma simplificação vocabular, como toda definição.

**Definição : 1** *quasi-splines.*

*Diremos que $\mathcal{P}_n$ é um* k-quasi-splines *se for uma função polinomial por pedaços de grau k, que significa:*

- *$\mathcal{P}_n$ é uma função contínua;*
- *definida em um intervalo $[a, b]$ ao qual está associada uma partição com n nós;*
- *em cada um dos sub-intervalos desta partição $\mathcal{P}_n$ coincide com um polinômio de grau menor ou igual k.*

A poligonal, função linear afim por pedaços e contínua que apareceu no primeiro exemplo acima, é um 1-quasi-spline, é também um 1-spline. O pre-fixo "1" diz respeito ao grau dos polinômios por pedaços usados para construir a aproximação. Se necessário, indicaremos a classe de diferenciabilidade. Os quasi-splines que vamos construir aqui são de classe inferior ou igual a $\mathcal{C}^2$.

Nos programas *grafunK* do disco são produzidos k-quasi-splines aproximando a mesma função :

$$f(x) = 5sen(x/4) + x/4 + 1$$

o que lhe permite uma boa comparação entre eles e a aproximação que eles representam desta função .

**Crítica do Cálculo.**

Vamos fazer uma crítica a apresentação usual do Cálculo e mesmo dos métodos numéricos. Veja que *crítica* significa *análise*, é o que faremos aqui.

Você já se deve ter perguntado alguma vez porque calculamos aproximações de funções que conhecemos muito bem. No Cálculo, por exemplo, se apontam os polinômios de Taylor, (*séries de Taylor* ...), como aproximações de funções bem conhecidas. Para construir um polinômio de Taylor é necessário conhecer várias derivadas da função que se está aproximando.

3. *Inclua na análise do exercício anterior, o uso da derivada calculada algebricamente ou aproximadamente. Faça registro do $\Delta x$ usado para o cálculo da derivada aproximada em sua estatística.*

### 1.2.2 Aproximação polinomial por pedaços.

Na seção anterior construimos *aproximação linear por pedaços* e uma aproximação do *segundo grau por pedaços.* como as funções do primeiro grau são polinômios então já demos os primeiros passos na construção de aproximações polinomiais. Agora queremos trabalhar com polinômios de um grau qualquer. Mas, por que aproximação polinomial *por pedações* ? Neste livro não trataremos de outras aproximações polinomiais, se você quer saber a razão , consulte a penúltima seção deste capítulo onde se encontra uma crítica das aproximações polinomiais em geral. Há diversos livros, ver por exemplo [4], que tratam a aproximação polinomial com mais detalhes e seria inútil repetí-lo aqui sem nada trazer de novo. Os métodos de aproximação polinomial perderam terreno para os splines definitivamente, ver [22]. O livro de Larry Schumaker é muito complicado e confuso, como certamente teria que ser pela época em que foi escrito, mas representa uma referência importante sobre a história e os passos mais decisivos na construção da teoria em se baseiam os splines.

Os objetos que vamos inicialmente construir não são **splines** mas se aproximam muito deles e porisso os vamos chamar de quasi-splines. A classe de continuidade de um $k$-quasi-spline será inferior a k-1, *em geral apenas diferenciáveis, mas não continuamente diferenciáveis*, mas será formado de polinômios de grau $k$. Esta é diferença com os $k$-splines cuja classe de diferenciabilidade é $\mathcal{C}^{k-1}$, isto é, se $\mathcal{P}$ for um $k$-spline então $\mathcal{P}$ é uma função definida por pedaços, cada um desses pedaços é um polinômio de grau $k$ e a função pode ser derivável continuamente $k-1$ vezes. No capítulo 3 ficará inteiramente claro como construir uma tal função . Neste capítulo

que ajuda a ver o efeito gráfico que nos interessa mostrar: escolha o intervalo [-50 , 50] com um passo 2.

Escolha depois distintos valores para o passo. Analise os resultados.

Este exemplo mostra quando você escolhe um passo maior, o efeito das tangentes que, se superpondo, acentuam em cada nó o erro da aproximação tangencial de uma função.Esta aproximação tangencial corresponde a um polinômio de Taylor de primeiro grau.

O trecho central do programa é :

```
program grafun02    (trecho)
.....................................
  deltay := 2*deltax {* comprimento das tangentes *}
  Eixos; x := inicio; GotoXY(01,23);
  Write('f(x) = 5*sin(x/4) + x/4 +1 ');
  While  x < fim Do
  Begin
     teta := arcTan(df(x));
     s := cos(teta)*deltay; t:=sin(teta)*deltay;
     Reta(x-s,f(x)-t,x+s,f(x)+t);
     Reta(x,0,x,f(x));  {* reta vertical em cada no' *}
     x := x + deltax;
  End;
..................................
```

O programa grafun02 se encontra no disco. Este não é um método muito usado, se você rodar o programa verá que o aspecto visual da aproximação pode ser ruim dependendo do comprimento dos segmentos de reta tangentes, mas se os segmentos de reta tangentes forem *pequenos* o resultado é uma boa aproximação . Também é dificil definir uma aproximação contínua com este método. A razão de sua inclusão aqui é que a ideia nele contida será aproveitada mais a frente num método de solução aproximada de equações diferenciais.

Ao calcularmos aproximadamente a integral de $f$ usando a regra do trapésio, este cálculo corresponde à soma das áreas dos trapésios cujas bases são os sub-intervalos $[x_i, x_{i+1}]$ da partição tendo as duas alturas $f(x_i), f(x_{i+1})$.

A função assim resultante é contínua, entretanto a sua derivada pode não sê-lo, o que representa um *defeito* da aproximação obtida frente a uma hipótese de alta diferenciabilidade dos fenômenos que é muito comum ser feita. Temos um modelo que não é diferenciável. Na próxima seção usaremos polinômios por pedaços de maior e melhoraremos a continuidade de nossas aproximações.

**Exercício: 2** 1. *Produza uma versão distinta de* grafun *fazendo as tangentes se iniciarem no nó $x$ e se prolongarem ao longo do intervalo $[x, x + \Delta x]$ .*

2. *Use um sistema para cronometrar o tempo de computação gasto e faça um estudo comparativo dos tres métodos de integração apro-ximada que discutimos até agora. Use distintas funções no seus testes.*

```
## comentario: coeficiente b calcula com a formula de Taylor.
function polsegped(x)
        if or([x<-10 ; x>=10]) then y = 0
        else p = floor(x);  a=9*cos(3*p+4)+2; b= -27*sin(3*p+4)/2;
                y = f(p) + a*(x-p)+ b*(x-p)^2
        end
end
```

e no caso do polinômio por pedaços contítinuo, (fig. 1.3) é:

```
## comentario: coeficiente b calculado de modo a produzir
##     polinomios p.p. continuos.
function [y] = polsegpedcon(x)
        if or([x<-10 ; x>=10]) then y = 0
        else    p=floor(x); a=9*cos(3*p+4)+2; b = f(p+1) - a - f(p);
                y = f(p) + a*(x-p)+ b*(x-p)^2
        end
end
```

A diferença reside na definição do coeficiente $b$ do termo do segundo grau.

Nos dois casos estamos usando o código do [21], um programa do INRIA de domínio público que segue a sintaxe do *MatLab*. Com pequenas modificações estas definições podem ser usadas em *Maple* também. Ver também [8].

A definição da interpolação linear é:

```
function [y] = trapesio(x)
        if or([x<-10 ; x>=10]) then y = 0
        else inicio = floor(x); fim = ceil(x); m = f(fim)-f(inicio)
        y = m*(x+inicio)+f(inicio)
        end
end
```

Em todos os casos usamos a partição inteira do intervalo $[-4, 6]$ e isto se encontra refletido na definição das funções acima.

**Exemplo: 2** *Funções lineares por pedaços.*

Os exemplos que apresentamos acima com auxílio do SciLab, você poderá executá-los com auxílio do TurboPascal.

Rode os programas *grafun01 e grafun* do disco. O primeiro vai apresentar-lhe a função aproximada por retas como descrevemos acima. O segundo a propria função com o intervalo discretizado de acordo com a sua escolha do $\Delta x$ . Observe a mensagem pedindo o *passo* deltax que é diferente nos dois programas.

**Exemplo: 3** *Método de aproximação tangencial.*

Uma forma diferente de aproximar uma função pode ser pelas tangentes tomadas nos nós da partição do intervalo. Se voce substituir a parte central do programa grafun1 pelo trecho abaixo, o resultado será este. A função que se encontra alí definida é

$$f(x) = 5sen(x/4) + x/4 + 1$$

intuiva da aproximação de funções, o seu conteúdo será generalizado nas seguin-tes.

Considere $f : [a, b] \mapsto \mathbf{R}$. Com frequência $f$ é apenas um nome para uma tabela de dupla de entrada que relaciona os nós de uma partição do intervalo $[a, b]$ com os valores colhidos por um aparelho associados a estes nós. Uma forma de aproximar a função $f$ é a linear por pedaços que já usamos acima dentro da regra do trapézio. Na *regra do trapésio*, substituimos a função $f$ no sub-intervalo $[x, x + \Delta x]$ pelo segmento de reta que passa nos pontos

$$(x, f(x)) \text{ e } (x + \Delta x, f(x + \Delta x)) \ ,$$

de modo que o conjunto destes segmentos de reta formam uma poligonal que coincide com $f$ sobre os nós da partição do intervalo. A área limitada por esta poligonal e o eixo OX é sua integral, (da poligonal vista como função ), e representa uma aproximação da integral de $f$. Este é o conteúdo da *regra do trapésio* vista sob o enfoque de aproximações de funções: *calculamos a integral duma função que é uma aproximação de outra função.*

**Exemplo: 1** *Interpolação polinomial. Função polinomial por pedaços.*

Vamos considerar $f(x) = 3sen(3x + 4) + 2x + 1$. Na (fig. , 1.2) e na (fig., 1.3) você encontra $graf(f)$ junto com o gráfico de uma função linear por pedaços que coincide com $f$ sobre os nós de uma partição do intervalo $[-4, 6]$ com passo $\Delta x = 1$. Os nós da partição são assim chamados de *pontos de precisão* e o seu conjunto de *conjunto de precisão* porque neles se obtem precisão total.

No primeiro gráfico, os polinômios do segundo grau foram construidos apenas considerando o método dos Polinômios de Taylor, isto é, calculando os coeficientes como as sucessivas derivadas de $f$ no extremo inicial do intervalo. Veja os saltos de descontinuidadde que ocorrem no final de cada intervalo em que $P_i(x_{i+1}) \neq P_{i+1}(x_i)$. No segundo gráfico o método é o que descrevemos com o sistema de equações da regra de Simpson. Neste caso se tem $P_i(x_{i+1}) = P_{i+1}(x_i)$ e esta condição foi usada para modificar o método de Taylor no cálculo do último coeficiente.

Um defeito do exemplo é de sugerir que a interpolação linear não é boa. O objetivo não é este, na verdade a interpolação linear é tão boa quanto as interpolações polinomiais de maior grau, apenas exige geralmente uma partição mais fina do intervalo para que uma alta precisão seja atingida, logo maior tempo de processamento. Entretanto, nem mesmo isto pode ser considerada uma verdade absoluta...

A definição dos *polinômios parciais* do segundo grau correspondente a figura (fig. , 1.2) é:

*estendemos a regra de Simpson às funções multivariadas*? nós o fizemos, no último exercício acima, com a regra do trapésio.

Com os métodos do capítulo 3 isto será possivel e re-lativamente fácil.

## 1.2 Aproximação polinomial de funções .

Aproximação de funções *esteve sempre presente em quase tudo que o estudante fez em Cáculo, apesar de certas ideias mal expressas que de certa forma não podiam mesmo ser apresentadas de outra maneira. Por exemplo, uma percepção muito comum que fica no estudante é de que as funções* sempre *tem equações, que* sempre *as podemos derivar ou até demonstrar que são contínuas. Ora, funções são meios de guardar tabelas de dupla entrada em que se associam valores de duas naturezas diversas, como quando se coloca um medidor elétrico ou eletrônico para captar as variações de um fenômenos ao longo do tempo. Depois precisamos calcular a quantidade do fenômeno o que consiste em calcular uma integral de dados discretos: temos que calcular aproximadamente uma integral. Em outras palavras, temos que calcular a integral de uma* função que aproxima a função a ser integrada, *Vamos reverter o sentido da seção inicial, em vez de calcular uma integral aproximada, vamos estudar métodos de aproximar as funções, depois calcularemos a integral da aproximação. O método de aproximação polinomial é a característica deste livro.*

*O material discutido nas duas primeiras seções não é novo. As duas seções finais tratam de assunto que ainda é raro em livros didáticos:* as aproximações polinomiais por pedaços, quasi-splines e splines.

*Entretanto o nosso estudo de splines neste capítulo é introdutório, voltaremos a eles no capítulo 3.*

### 1.2.1 Aproximação linear por pedaços.

Esta subseção é introdutória, desenvolve a parte

5. Seria obviamente interessante executar esta experiência diversas vezes, com funções distintas, medindo o tempo gasto em cada caso e analisar os resultados frente a estas variáveis e a aproximação obtida, e concluir qual é o método ótimo entre os dois: somas de Riemann ou Simpson. Observe que o passo da Soma de Riemann final é menos de 10 vezes menor que em Simpson. Entretanto uma divisão por 10 já pode levar para a área de risco da precisão no computador...

6. O *teste de Cauchy* pode ser especializado afim de parar quando as aproximações começarem a se degradar: *se aproximar da área de risco da precisão do computador.*

**Exercício: 1** *1. Calcule as integrais univariadas dos exercícios do seu livro de Cálculo usando tanto a regra de Simpson como a regra do trapésio. Compare os resultados para uma mesma precisão nas partições escolhidas.*

*2. Estenda a regra do trapésio para funções multivariadas e calcule as integrais múltiplas dos exercícios do seu livro de Cálculo*[3]*.*

*3. Crie, com* **Case of***, uma coletânea de funções e aplique às mesmas as regras de Simpson e do Trapésio assim como Somas de Riemann, estas com precisão 10 vezes superior. Calcule para cada função a média das integrais, o desvio padrão da média. Faça o seu programa analisar automaticamente os resultados e emitir um laudo indicando qual o melhor método excluindo os casos em que o desvio padrão for alto, (*indicativo de dado estatístico não confiável*).* A solução deste exercício é o programa IntEstos (Integral Estocástica) que você encontra no disco.

O leitor deve observar um defeito neste texto e deve obviamente buscar uma solução para o mesmo: *porque não*

[3] Observe que a regra do trapésio é soma de Riemann calculada duas vezes com uma translação nos extremos...

```
End;
{*** inicio da rotina principal  ***}
Begin
   deltax := 0.1; soma:=0;
   soma := 0; x:=inicio;
   Repeat
      {*** calculo dos coeficientes em cada sub-intervalo ***}
      a0 := f(x); a1 := df1(x,deltax/10000);
      a2 := ( f(x+deltax) - a0 -a1*deltax )/( deltax * deltax );
        {** atualizando soma com integrais exatas de polinomios **}
      soma := soma + a0 + a1*deltax/2 + a2*deltax*deltax/3;
      x := x+deltax;
   Until x > fim; soma:=soma*deltax;
   Write('integral(',deltax:8:8,') =  ',soma:8:8);
End.
```

Observe que o programa *Riem02.pas* faz mais do que calcular uma *soma de Riemann*, ele contém um teste *tipo de Cauchy* e portanto ele calcula várias *somas de Riemann* até que atingir um erro estipulado. Acima nós eliminamos o *teste de Cauchy* do trecho de programa apresentado.

Este trecho de programa pode ser inserido em *Riem02*. Vamos chamar este novo programa de *Simpson.pas*, nós o testamos com a função $f(x) = x^2$ no intervalo $[0, 1]$ com um passo inicial 0.1 tendo obtido o seguinte resultado:

1. usando derivação exata e com erro na condição de Cauchy de 0.0001 o programa parou depois de 11 iterações com o resultado 0.333431 correspondente ao passo 0.00004883.

2. usando derivação aproximada, *veja df1*, e com o mesmo erro anterior, o programa parou de depois de 10 iterações com o resultado 0.33343099 correespondente ao passo 0.00009766.

3. usando apenas soma de Riemann mas com erro 0.00001 na condição de Cauchy, o programa parou depois de 14 iterações com o valor de 0.33333941 correspondente ao passo 0.00000610.

4. Não medimos o tempo de processamento gasto nos tres casos acima. Isto pode ser feito usan a função **GetTime**, veja com Ctrl-F1.

através das seguintes condições:

1. $a_{i,0} = f(x_i)$
2. $a_{i,1} = f'(x_i)$
3. $P_i(x_{i+1}) = f(x_{i+1})$

A última condição serve para calcular $a_{1,2}$ de modo a garantir a continuidade da função *polinomial por pedaços*.

Desta maneira estamos substituindo $f$ por uma função polinômial por pedaços, que é contínua, quem no-lo garante são as condições (1) e (3). Na regra do trapésio estavamos substituindo a função $f$ por uma função linear por pedaços, também contínua.

Finalmente vamos calcular *exatamente* a integral do poli-nômio em cada sub-intervalo $[x_i, x_{i+1}]$:

$$a_{i,0}\Delta x + a_{i,1}\frac{\Delta x^2}{2} + a_{i,2}\frac{\Delta x^3}{3}$$

observando que $\Delta x$ pode ser posto em evidência para ser multiplicado ao final pela variável *soma* em que, tradicionalemte, guardamos o valor de nossas integrais. Dentro do programa não precisaremos de tantos sub-índices a não ser que seja propósito do programa fornecer a matriz dos coeficientes da *função polinomial por pedaços* que aproxima $f$, *neste caso você deve usar uma variável de tipo vetorial para guardar os dados*[2]. Se o único resultado que interessar for o valor da integral, bastam-nos tres variáveis a0,a1,a2 para fazer o cálculo em cada sub-intervalo.

Veja o algoritmo abaixo:

```
{Trecho de programa}
program simpson;
.....
{*** calculo da derivada aproximadamente  ***}
Function df1(x,delta:Real):Real;
Begin
  df1 := ( f(x + delta) - f(x) )/delta;
```

[2] uma alternativa que consome menos memória, mas aumenta o tempo de processamento, lançar os dados num arquivo em disco.

A conclusão que se tira é de que *se deve* tentar otimizar os programas, sim, porque o tempo de processamento deles se reduz. Entretanto se deve ter o cuidado de *documentar* com comentários estas modificações para facilitar os trabalhos posteriores de manutenção dos sistemas. O que se perde é apenas espaço em disco, e isto pode ser contornado com programas de compactação de arquivos.

### 1.1.2 A regra de Simpson.

A **regra de Simpson** se constitue da aproximação de uma função por um polinómio do segundo grau. No próximo capítulo iremos mais longe do que isto quando aproximarmos funções por polinô-mios de grau tres ou quatro.

A regra de Simpson melhora a regra do trapésio interpolando com um polinômio do segundo grau e calculando a integral destes polinômios. Se você se recorda da aproximação de uma função feita com um Polinômio de Taylor, os polinômios de Taylor de grau elevado, maior do que primeiro grau, *memorizam* a concavidade da função, portanto a aproximam melhor do que retas, consequentemente se precisa de um número menor de sub-divisões do intervalo para conseguir uma boa aproximação . Tem-se assim mais rapidez aliada a uma boa precisão .

**Observação: 2** *Polinômio de Taylor*

*O teorema de Taylor diz que se uma função $f$ for muito diferenciável, até a ordem $n$, digamos, no ponto $\underline{a}$ interior a um intervalo, então*

$$f(x) \approx f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Há várias maneiras de se fazer isto, vamos escolher uma que terá sua continuidade depois.

Se conhecermos a equação de uma função e se ela for derivável, o método de Taylor nos permite encontrar alguns dos coeficientes de um polinômio que aproxime $f$ num ponto $x = a$ . Vamos usar em parte o método de Taylor.

No intervalo $[x_i, x_{i+1}]$ vamos definir o polinômio

$$P_i(x) = a_{i,0} + a_{i,1}(x - x_i) + a_{i,2}(x - x_i)^2$$

importante feita foi na linha

```
soma := soma + f(inicio);
```

que foi modificada para:

```
soma := soma + ( f(inicio) + f(inicio + passo))/2;
```

observe que substituimos o valor da função no ponto f(inicio) pela sua média aritmética com f(inicio + passo).

**Observação: 1** *Otimização dum programa.*

Um dos pontos críticos do processamento de dados ainda é o *tempo de processamento*. Isto conduz algumas vezes os programadores a eliminarem operações inúteis dentro de um programa. Uma consequência nefasta disto é que os programas se podem tornar *ilegíveis* porque a lógica do processamento fica invisível. O tempo que se ganha com um programa que roda mais rápido pode depois se perder em proporções muito maiores em compreender este mesmo programa posteriormente seja por quem o escreveu ou por seus colaboradores.

Uma forma de contornar este problema consiste em incluir mensagens explicativas dentro das rotinas indicando o que elas fazem. Estas mensagens vão tornar o código mais longo mas serão ignoradas pelo compilador e assim não irão aumentar o tempo de processamento.

A principal rotina do programa anterior é:

```
Begin
  soma:=soma+(f(inicio)+f(inicio+passo))/2;
  inicio:=inicio+passo;
  WriteLn('S = ',soma*passo:9:4);
End;
```

que pode ser otimizada. Observe que nela estamos calculando a soma quase duas vezes, e ainda dividindo por dois! Apenas o ponto inicial e o final não são duplicados. Então conseguiremos mais rapidez se

1. colocarmos de lado estes dois pontos: inicio e fim,
2. somarmos o restante,
3. ao final acrescentarmos ao dobro da soma, os dois pontos que faltaram,
4. multiplicarmos a soma assim obtida pelo valor do passo divido por dois.

Economisaremos assim n divisões , n somas. Observe a colocação de mensagens que indicam onde houve otimizações .

```
{** rotina otimizada para calculo medias  **}
x := inicio + passo; {** se perde o ponto inicial **}
 While x < fim - passo Do
Begin
  soma:=soma+f(x);
  x:=x+passo;
End;
{** nao se alcanca o ponto final **}
{** pontos que faltaram + dobro da soma ***}
soma := (2*soma + f(fim) + f(inicio))*(deltax/2);
WriteLn(soma);
```

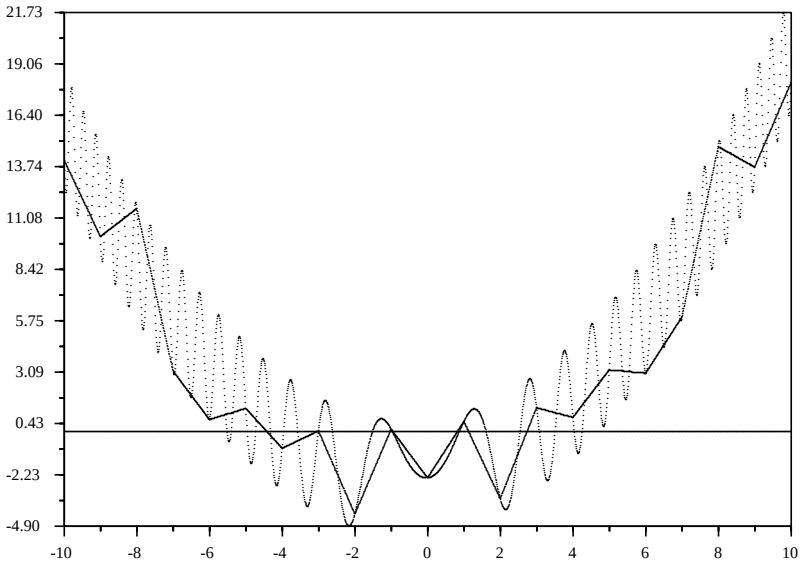



Figura 1.1: Interpolação linear do $graf(f)$

```
  WriteLn('S = ',soma*passo:9:4);
End;
soma := soma*deltax;
ReadLn;
```

O segmento de programa acima foi tirado e modificado de Riem02 que se encontra no disco. Observe que o programa Riem02 faz testes do *tipo Cauchy* com somas de Riemann, estes testes foram omitidos no trecho acima, também foram tirados alguns comandos que melhoram o Lay-Out do programa afim de que o código ficasse mais curto. A única modificação

nificativa da História das Ciências, mas a ciência avança e deixa algumas coisas muito bonitas para traz. Certamente ficará para os *polinômios trigonométricos* uma função nobre: a de acompanhar os neófitos em seus primeiros passos pela *análise harmônica e não harmônica* porque, pelo menos por enquanto, as *ondas* mais naturais para os computadores ainda são os senos e os cosenos, ... por enquanto.

## 1.1 Integração .

### 1.1.1 Integração aproximada.

Nos livros de Cálculo Numérico se citam dois métodos de Integração aproximada: regra do trapésio e regra de Simpson. Ambos necessitam das somas de Riemann para serem executados e os dois são um, a generalização do outro, e de uma certa forma ambos se enquadram nos conceitos do último parágrafo deste capítulo, como veremos a seu tempo.

A **regra do trapésio** se constitue de aproximar a integral de uma função usando uma *interpolação linear* da mesma sobre os sub-intervalos duma partição . Veja a (fig. 1.1) em que uma função fortemente oscilante se encontra "*aproximada*" por interpolação linear.

Como a integral *de Riemann* de uma função do primeiro grau corresponde à área dum trapésio[1], este método se resume a usarem-se as médias aritméticas entre

$$f(x), f(x + \Delta x)$$

numa soma de Riemann. O seguinte trecho de programa faz isto:

```
soma:=0;
While inicio < fim Do
Begin
  soma:=soma+(f(inicio)+f(inicio+passo))/2;
  inicio:=inicio+passo;
```

[1] claro, um retângulo é tipo de trapésio...

# Capítulo 1

# Aproximação de funções .

Introdução :

A definição matemática de integração foi feita com "somas de Riemann" e mesmo a maior parte das generalizações que se fizeram do método de Riemann usam somas semelhantes a esta para definir o conceito. Mas quando se quer calcular aproximadamente uma integral e atingir uma boa precisão tudo isto submetido a restrição do tempo de processamento, é preciso buscar um nível maior de sofisticação.

Há dois métodos clássicos de integração aproximada: *regra do trapésio* e *regra de Simpson* e o nosso objetivo é de colocá-los dentro do contexto mais amplo de *aproximação de funções*. Veremos na parte final do capítulo que estas duas *regras* são casos particulares de *interpolação polinomial*, que será a nossa *ferramenta de trabalho* principal no resto do livro, em especial no último capítulo.

A parte matemática deste capítulo se vai voltar para os métodos de aproximação de funções . Introduziremos aqui uma formulação fraca de Splines que chamamos de "quasi-Splines" e apresentaremos uma definição de Splines junto com uma interpretação "geometrica" dos mesmos que lhes justifica o nome. Splines serão tratados algoritmicamente no capítulo 3.

Terminaremos este capítulo com uma *introdução* a um assunto histórico: *aproximação de funções por polinômios trigonométricos.* Será apenas um registro, não queriamos deixar de mencionar esta vertente que corre sério risco de se tornar *assunto de museu*, apesar de sua importância científica e tecnológica, mas uma nova técnica, que lhe robou o método, *as wavelets*, já lhe supera as vantagens em diversas aplicações. Observe que *ser assunto de museu não representa uma mancha na reputação*, a não ser em paises em que o desenvolvimento cultural cai na faixa dos supérfluos. Nos museus se encontra a História da Humanidade. Os polinômios trigonométricos, ou as séries de senos e cosenos fizeram parte sig-

aplicar na discretização de operadores diferenciais a coeficientes constantes, caso univariado: equações diferenciais ordinárias lineares.

Os programas em Pascal foram feitos e testados com TurboPascal da Borland, e os programas em Maple foram testados e rodados no Departamento de Engenharia Química da UEM. Todos os programas podem ser usados e modificados livremente, espero apenas que seja feita a devida citação .

Este livro foi composto com o processador de textos TEX produzido por Donald Knuth, junto com a implementação para micros EmTEX de Eberhard Mattes, que intermedia TEX via LaTEX. LaTEX foi produzido por Leslie Lamport. Foi usada também a "casca" TEX$telm^extel$ que gerencia os módulos do EmTEX, produzido por Andreas Krebs. Os gráficos foram traduzidos para TEX com auxílio do programa bm2font de Heinrich-Heine-Universität Düsseldorf Universitätsrechenzentrum Friedhelm Sowa. Todos estes programas são de domínio público e podem ser obtidos em algum dos CTAN, Comprehensive TeX Archive NetWork, que se encontram em diversos nós, ver por exemplo em

"ftp.dante.de /pub/tex/systems/msdos/emtex" "ftp.dante.de /pub/tex/systems/unix/Latex" "ftp.debian /pub"

Sou grato aos autores dos programas.

Tarcisio Praciano Pereira
e-mail tppereirsuper.furg.br
Sobral, 5 de agosto de 2001

.

gral que deseje motivar os seus alunos usando computação como um meio de *concretizar* os conceitos do Cálculo.

Este último ponto é para mim de particular importância, estou convencido de que o uso de computação no ensino de Matemática enriquece fortemente a experiência do aluno por que permite introduzir uma dinâmica que giz e quadro negro não conseguem mais gerar ante uma nova mentalidade gráfica que está presente em nós todos. Tome-se como exemplo o caso dos *polinômios trigonométricos* que podem ser encontrados aqui no final do terceiro capítulo, é interessante observar o interesse que se pode suscitar nos alunos pelo simples fato de ver a aproximação que eles produzem para uma função e poder discutir com eles, frente ao video, o tipo de aproximação que se obtem do ponto de vista da energia do fenômeno aproximado contra uma aproximação ponto a ponto.

A segunda parte poderia ser caracterizada como um curso avançado de Cálculo Numérico, voltado para teoria da Aproximação .

1. No capítulo 3 começo uma introdução sobre teoria de aproximação de funções . O primeiro passo se volta para integração numérica aprofundando o uso de Somas de Riemann feito no capítulo 1. Aqui introduzo quasi-Splines que é uma forma fraca de aproximação polinomial. Splines são tratados no capítulo 5. Termino com uma pequena apresentação de polinômios trigonométricos.
2. O capítulo 4 é uma "Álgebra Linear computacional" dirigida para solução de sistemas de equações .
3. O capítulo 5 é se dedica a soluções numéricas de equações dife-renciais. Nele defino rigorosamente Splines que usarei para definir projetores e com eles resolver aproximadamente equações diferenciais periódicas. Termino o capítulo com regularização por convolução que se vai

Nesta primeira parte o objetivo principal é transmitir ao leitor os rudimentos de Pascal e quebrar as dificuldades iniciais com o aprendizado da linguagem. Espero que posteriormente a esta *quebra de gelo*, o leitor se encontre em condições de prosseguir o seu aprendizado de Pascal sozinho.

Não seria muito realístico supor que os alunos saibam Pascal, em muitos casos até mesmo supor que saibam programar é irreal. Mas acho que é preciso *forçar a barra*, o venho fazendo com os meus alunos de Cálculo Numérico, com resultados positivos: ao final do segundo mes de aula a maioria deles já sabe fazer programas. Os programas crescem de nível de modo que, *se o aluno se empenhar em entendê-los*, ficará gradualmente no nível dos mesmos. Ao longo do primeiro capítulo mantivemos a preocupação de explicar detalhadamente os programas, mas aos poucos vamos deixando que o leitor comece a voar sozinho...obviamente, na companhia de um bom livro sobre Pascal e com a certeza de que sempre encontrará a sua volta algum "*hacker*", que a menos de alguma *empáfia típica*, poderá ser-lhe de muito ajuda.

Associo esta introdução ao Pascal ao estudo da parte mais elementar de Cálculo Numérico que indentifico como a *nume-rização das técnicas do Cálculo*: integral, derivada, máximos e mínimos, caso univariado e multivariado. Todos estes problemas se podem resolver com uma técnica básica de programação , *varredura*, mais o acréscimo de alguma técnicas matemáticas auxiliares.

Assim esta primeira parte pode tanto representar material para um curso de introdução à ciência da computação para estudantes das áreas de exatas ou tecnológicas como pode representar um curso mais elementar de Cálculo Númerico, ou ainda, espero, material auxiliar para ser usado pelo professor de Cálculo Diferencial e Inte-

*aprendido os rudimentos de programação em Pascal, passar para Maple, seria sem dúvida uma saida interessante.*

*Na segunda parte deste livro não estarei usando nenhuma linguagem de programação explicitamente. Farei menção a programas em Pascal ou Maple conforme for mais conveniente.*

Este livro tem um disco flexível que o acompanha. Não se esqueça de fazer-lhe um *backup* logo.

**Vocabulário: 1** *Backup.* **Backup** *significa cópia de reserva. Os disco de computador ainda são muito frágeis e podem se danificar com relativa facilidade até mesmo por questões climáticas. Devido a isto é muito importante ter cópias de reserva de todo material que se tenha em discos e naturalmente guardar tais cópias em locais diferentes livres de humidade e de alta temperatura, e afastadas de campos magnéticos.*

No disco você poderá encontrar todos os programas que se encontram listados no livro e até algumas variantes dos mesmos. Todos os programas foram testados várias vezes inclusive por alguns alunos. Mas ainda podem ter ficado erros, pelo que me desculpo.

O livro se divide em duas partes e de acordo com a declaração de princípios feita acima.

A primeira parte, *de desenvolvimento tecnológico básico* contendo dois capítulos:

1. O primeiro capítulo trata de problemas elementares de apro-ximação , basicamente faz aquilo que deveria ter sido feito num curso moderno de Cálculo I.
2. O segundo capítulo já representa um vôo um pouco mais alto, trata da solução de equações da forma $f(x) = 0$ em que $f$ é uma função "qualquer".

*começo para nela vasar os algoritmos usando o material aqui apresentado como exemplos.*

2. *Desenvolvimento científico avançado: Computação Algébrica.*

*Um outro objetivo, num certo sentido* contraditório *com o anterior, dentro de um livro como este, é o desenvolvimento tecnológico avançado e de ponta. Com uma linguagem como Pascal, em geral se usa C, pode ser construida uma linguagem apropriada para computação algébrica. Mas de imediato podem os nossos alunos usar uma linguagem já feita para Computação Algébrica: Maple, Mathematica, MatLab, Derive são algumas das opções . Derive em particular não é uma linguagem de programação , é um poderoso pacote de Computação Algébrica feito em LISP.*

*A contradição se encontra em que* um usuário de uma linguagem de computação algébrica **nunca** será um construtor de uma tal linguagem. *Terá tecnologia, mas não a vai dominar.*

*Não sei resolver esta contradição , mas acredito poder contribuir n'alguma medida para o nosso desenvolvimento se instrumentar os alunos com uma linguagem como Pascal que está a um pequeno passo de C. Mas quero também abrir o caminho para o uso imediato de uma tecnologia avançada.*

*Uma excelente alternativa para Pascal seria Maple. Maple é um pacote de Computação Algébrica desenvolvido por professores da Universidade de Waterloo, Canada. Junto com ser um "tutor para matemática ", por que permite que você aprenda matemática pelo simples fato de usá-lo, Maple contém uma linguagem de programação que reune características do Pascal junto com com comandos de Computação Algébrica. Em Maple se pode definir uma função e em seguida trabalhar com a derivada da mesma, inclusive com derivadas sucessivas, por exemplo. Depois de ter*

*dentro dos círculos de pesquisas universitárias, embora hoje já venha se tornando uma linguagem quase popular, pelo menos um dos seus dialetos: Scheme.*

*Com Pascal veiu a compreensão de estrutura da infor-mação . Os dados agora se classificam, os problemas passam a ser resolvidos com os olhos voltados para o tipo de objetos em que eles se aplicam. Isto abriu o campo da abstração para a* arte de programar computadores*: agora não fazemos mais programas apenas para resolver um problema específico que temos objetivamente pela frente. Mesmo quando o fizermos o nosso pensamento logo se volta para uma generalização do programa para que ele possa ser aplicado* em uma classe de problemas semelhantes*. Isto é consequência direta da* estruturação *dos dados e dos programas, (*programação orientada a objeto é o fato mais recente neste direção *).*

*Se por um lado Pascal é uma excelente linguagem didá-tica por conduzir o programador a obter hábitos regulares de es-truturação do seu trabalho, por outro lado traz algumas complica-ções que o leitor terminará por ver em algum momento. A forte estrutura que têm os próprios programas torna praticamente impossível que um programa modifique outro programa que é um ponto chave da Inteligência Artificial. Pascal é o que se chama de uma linguagem* fortemente tipada*. Fica isto dito para alertá-lo, possivelmente você deverá escolher uma outra linguagem posteriormente, ou possivelmente fique com Pascal para sempre como faz muita gente.*

*Uso o compilador Pascal da Borland, TurboPascal, por que estou convencido de sua excelência entre as implementações de Pascal que conheço. Não tenho dúvidas em sugerir-lhe o uso. Assim, estarei usando TurboPascal em vez de Pascal. Mas o leitor é vivamente incentivado a escolher sua própria linguagem de programação desde o*

sivelmente o uso de métodos computacionais na disciplina. O objetivo é duplo:

1. *Desenvolvimento tecnológico básico.*

   *Um país não se desenvolve se não possuir pesquisa básica inclusive tecnológica. Nossos estudantes para isto tem que ser expostos às ferramentas básicas para produzir pacotes computacionais e algorítmos.*

   *Escolhi Pascal como linguagem para representar os algoritmos. Isto não significa, de minha parte, nenhuma indicação de que Pascal é a melhor linguagem para processamento numérico. Entretanto não há dúvida que Pascal é a melhor linguagem de apresentação didática para algorítmos, ela é usada frequentemente como uma* pseudo linguagem *de programação . Esta maneira de usar Pascal chega a ter aspectos ridículos; usa-se Pascal em português colocando-se o estudante sob duas falsas visões : (1) que podem adiar indefinidamente o aprendizado do inglês, (2) lhes ensinam a escrever* algoritmos *numa linguagem inútil para aplicar em computadores que afinal seria o objetivo último dos mesmos.*

   *Pascal é uma língua que foi criada pelo Matemático holandês, N. Wirth, com objetivos didáticos. Precisamente* o inverso de aprender a usar um programa que funcione, *o objetivo é aprender a fazer programas que funcionem. Pascal absorveu o que se encontrava no ar no início da década de 70, num momento de amadurecimento das linguagens de programação : se compreendia que as tarefas na verdade não eram atômicas, elas se dividiam em pedaços, ou blocos, que podem muitas vezes ter existência própria. Foi assim que surgiram as linguagens modulares de que alguns exemplos são Pascal, C, LISP entre outras, possivelmente estas as mais simples. LISP é uma linguagem mais velha que Pascal, mas foi criada com uma filosofia bem diferente das suas contemporâneas tendo ficado com seu uso restrito*

INTRODUÇÃO .

O material contido neste livro foi testado em três cursos de Cálculo Numérico na Universidade Estadual de Maringá, e mais recentemente, nos cursos de Engenharia da Universidade Federal de Rio Grande. Gostaria de prestar minha sincera homenagem aos alunos que participaram desta experiência com uma consciência clara de participação na mesma e que, com suas perguntas críticas, muitas vezes me nortearam na determinação do melhor caminho. A primeira versão do livro tinha o objetivo extremamente modesto de representar uma introdução ao Pascal para pessoas envolvidas com Matemática, somente aos poucos se solidificou a idéia de escrever um curso de Cálculo Numérico.

Análise numérica vem sendo objeto de muita discussão assim como também a disciplina mais elementar, o Cálculo Numérico, veja-se o artigo de Peter Linz, [12], a respeito. O mito de que se poderiam resolver problemas *numéricos* exclusivamente com computação já se encontra defininhando. No momento é clara a idéia de que uma "parceria" do homem com o computador é a melhor saída. Discute-se a propriedade de disciplinas que se originaram numa época em que o uso de computação mal podia delinear o avanço atual que veio desembocar, entre outras coisas, em Computação Algébrica com a qual é possivel resolver exatamente muitos dos problemas que a Análise Numérica só resolve aproximadamente.

Mas é preciso também combater o preconceito contra "aproximação" em que valores exatos, na maioria das vezes inexistentes, são colocados como um paradigma de perfeição em contraposição com aproximações "imperfeitas" mas inevitáveis.

Queremos pensar na aproximação como a única solução possível e ver a soluções exatas como um acidente. Ou, uma outra forma de ver as coisas, consiste em ver a *solução* como o resultado de uma simulação construida a partir de um grande número de experimentos que foram testando e melhorando as hipóteses.

Neste livro assumi o objetivo de aumentar sen-

3.5.1 Partição da unidade. . . . . 141
3.5.2 Projetores definidos com Partições da Unidade.148
3.5.3 Splines. . . . . 151
3.5.4 Solução de $y' = g$ com 2-Splines. . . . . 154
3.5.5 Solucao periódica de $y' = g(t, y)$. . . . . 155
3.6 Regularização por convolução . . . . . 156
3.7 Solução de equações lineares. . . . . 172
3.7.1 O método. . . . . 172
3.7.2 As demonstrações . . . . . 177
Bibliografia .....193
Índice remissivo alfabético.....196

**1 Aproximação de funções . 1**
1.1 Integração . . . . . 2
1.1.1 Integração aproximada. . . . . 2
1.1.2 A regra de Simpson. . . . . 5
1.2 Aproximação polinomial de funções . . . . . 9
1.2.1 Aproximação linear por pedaços. . . . . 9
1.2.2 Aproximação polinomial por pedaços. . . . 13
1.2.3 Splines. . . . . 33
1.2.4 Interpolação polinomial clássica . . . . . 38
1.3 Polinômios Trigonométricos. . . . . 40

**2 Álgebra Linear Computacional. 57**
2.1 A álgebra das matrizes. . . . . 57
2.2 Transformações Lineares. . . . . 71
2.3 Sistemas de equações lineares. . . . . 76
2.4 Independência linear. . . . . 80
2.5 Simplificação dos sistemas de equações. . . . . 87
2.5.1 Multiplicação por uma matriz não destrutiva. . . . . 87
2.5.2 Dependência linear das linhas. . . . . 90
2.5.3 Classes de matrizes. . . . . 91
2.6 Sistemas lineares: triangularição de matrizes. . . . 95
2.6.1 O programa *Sistema*. . . . . 104
2.7 O método de Gauss-Jordan. . . . . 108

**3 Equações Diferenciais Ordinárias. 117**
3.1 Equações Diferenciais. . . . . 118
3.1.1 Exemplos e notação. . . . . 118
3.1.2 A solução de uma equação diferencial. . . 119
3.2 Métodos *passo a passo*. . . . . 124
3.2.1 O método de Euler. . . . . 125
3.2.2 Métodos de Runge-Kutta. . . . . 128
3.2.3 Precisão da solução . . . . . 132
3.3 O método das Isóclinas. . . . . 134
3.4 Um método iterativo. . . . . 135
3.5 Interpolação polinomial. . . . . 141

Copyleft Tarcisio Praciano Pereira

Este livro pode ser livremente copiado para uso individual ou não comercial, desde que o seja em todo, de capa a capa, e que esta descrição do `copyleft` seja também mantida. Não fazer assim representa um crime contra os direitos do autor.

Para distribuir comercialmente o livro o autor deve ser contactado, no endereço tarcisio@e-math.ams.org

P496c

Pereira, Tarcisio Praciano
Cálculo Numérico Computacional:
Teoria da Aproximação
Laboratório de Matemática Computacional - LMC
Sobral: UVA, 2001
214.p
Bibliografia
ISBN:85-87906-09-7

1 - Splines - 2 - Aproximação de funções
3 - Cálculo Numérico Computacional.
I. Título

CDD 515.1

Tarcisio Praciano Pereira
PhD in Mathematics

# CÁLCULO NUMÉRICO COMPUTACIONAL

Teoria da Aproximação

Publicações Eletrônicas
do
Laboratório de Matemática Computacional
UVA
Sobral - Ce

# CÁLCULO NUMÉRICO COMPUTACIONAL.
# Teoria da Aproximação .

Tarcisio Praciano Pereira

Laboratório de Matemática Computacional
Universidade Estadual Vale do Acaraú
Sobral, 5 de agosto de 2001
tarcisio@e-math.ams.org - fone 55 88 677 42 46