Notas em Matemática Aplicada e-ISSN 2236-5915

Volume 36, 2012

Editado por

Célia A. Zorzo Barcelos
Universidade Federal de Uberlândia - UFU
Uberlândia, MG, Brasil

Eliana X.L. de Andrade
Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil

Maurílio Boaventura
Universidade Estadual Paulista - UNESP
São José do Rio Preto, SP, Brasil

Sociedade Brasileira de Matemática Aplicada e Computacional

2012

A Sociedade Brasileira de Matemática Aplicada e Computacional - SBMAC publica, desde as primeiras edições do evento, monografias dos cursos que são ministrados nos CNMAC.

Para a comemoração dos 25 anos da SBMAC, que ocorreu durante o XXVI CNMAC em 2003, foi criada a série **Notas em Matemática Aplicada** para publicar as monografias dos minicursos ministrados nos CNMAC, o que permaneceu até o XXXIII CNMAC em 2010.

A partir de 2011, a série passa a publicar, também, livros nas áreas de interesse da SBMAC. Os autores que submeterem textos à série Notas em Matemática Aplicada devem estar cientes de que poderão ser convidados a ministrarem minicursos nos eventos patrocinados pela SBMAC, em especial nos CNMAC, sobre assunto a que se refere o texto.

O livro deve ser preparado em **Latex (compatível com o Miktex versão 2.7), as figuras em eps** e deve ter entre **80 e 150 páginas**. O texto deve ser redigido de forma clara, acompanhado de uma excelente revisão bibliográfica e de **exercícios de verificação de aprendizagem** ao final de cada capítulo.

Veja todos os títulos publicados nesta série na página
http://www.sbmac.org.br/notas.php

SBMAC **Sociedade Brasileira de Matemática Aplicada e Computacional**

2012

GEOMETRIA DE ALGORITMOS NUMÉRICOS

Gregorio Malajovich
gregorio@ufrj.br

Departamento de Matemática, Aplicada
Instituto de Matemática
Universidade Federal do Rio de Janeiro

Sociedade Brasileira de Matemática Aplicada e Computacional

São Carlos - SP, Brasil
2012

Coordenação Editorial: Sandra Mara Cardoso Malta

Coordenação Editorial da Série: Eliana Xavier Linhares de Andrade

Editora: SBMAC

Capa: Matheus Botossi Trindade

Patrocínio: SBMAC

Copyright ©2012 by Gregorio Malajovich.
Direitos reservados, 2012 pela SBMAC. A publicação nesta série não impede o autor de publicar parte ou a totalidade da obra por outra editora, em qualquer meio, desde que faça citação à edição original.

Catalogação elaborada pela Biblioteca do IBILCE/UNESP
Bibliotecária: Maria Luiza Fernandes Jardim Froner

Malajovich, Gregorio
Geometria de Algoritmos Numéricos - São Carlos, SP: SBMAC, 2012, 76 p.; 20,5cm - (Notas em Matemática Aplicada; v. 36)

e-ISBN 978-85-86883-97-2

1. Algoritmos Numéricos. 2. Complexidade.
I. Malajovich, Gregorio. II. Título. III. Série.

CDD - 51

Esta é uma republicação em formato de e-book do livro original do mesmo título publicado em 2008 nesta mesma série pela SBMAC.

Conteúdo

Prefácio

Estas são as notas de um curso oferecido durante o Congresso Nacional de Matemática Aplicada e Computacional, Belém do Pará, de 8 a 11 de setembro de 2008.

Vou começar com uma afirmação polêmica. Matemática Computacional não é (nem pode ser) utilizar métodos computacionais para resolver um modelo matemático.

Meu argumento é que esse procedimento pode satisfazer engenheiros ou físicos, mas nunca satisfaz a curiosidade de um matemático. Não produz teoremas. Passa ao largo de problemas matemáticos interessantes.

São objetos susceptíveis de estudo matemático não apenas o modelo, mas ainda os algoritmos, a classe de todos os algoritmos resolvendo o mesmo problema, e ainda classes de modelos semelhantes (dependendo por exemplo de um ou mais parâmetros).

Isso difere do ponto de vista do engenheiro, para quem um algoritmo é uma regra de bolo. Funciona ou não funciona, se não funcionar, troca-se. Já para um matemático, algoritmos têm que ser provados corretos. Afirmações sobre a velocidade do algoritmo, ou a complexidade de um problema, exigem provas.

Quando o algoritmo não funciona na prática, ou quando funciona melhor do que o previsto, isso precisa ser explicado. Essa explicação vem geralmente em forma de teoremas.

Provar teoremas sobre algoritmos (e algoritmos numéricos em particular) exige um ferramental matemático de ponta. Em alguns casos, esse ferramental precisa ainda ser construído. Problemas de complexidade como $P \neq NP$ se transformaram em desafios para a comunidade matemática em geral.

Matemáticos gostam de problemas difíceis. Quanto mais conexões com

Gregorio Malajovich, *Geometria de Algoritmos Numéricos*. Notas em Matemática Aplicada **37** (XXXI CNMAC), SBMAC, São Carlos, 2008.

Copyright © Gregorio Malajovich, 2008.

áreas de pesquisa atuais, maior a garantia de que os resultados atuais serão *não-triviais*.

Nas duas primeiras aulas, ilustraremos algumas conexões entre a análise de algoritmos numéricos e tópicos como geometria diferencial ou algébrica.

Nas duas aulas seguintes, mostraremos como matemática de boa qualidade tem o poder de voltar, influenciando a nossa vida cotidiana. Para isso, vou mostrar quais são os fundamentos matemáticos da procura na Internet e da compressão de imagens e de som.

O único pré-requisito para este curso é ter conhecimentos básicos de Álgebra Linear, no nível de um curso de graduação. Alguns dos principais conceitos e teoremas de Álgebra Linear necessários para a comprensão do texto estão resumidos no Apêndice.

Os exemplos numéricos estão escritos em linguagem de programação C, ou na linguagem do aplicativo *Octave*. Este aplicativo é *software* livre, logo todos os exemplos podem ser executados em um computador com GNU-Linux bem configurado. Quem dispuser da versão em PDF destas notas pode utilizar o mouse para copiar e executar os exemplos numéricos.

Parte do material foi extraído do texto de Álgebra Linear do autor, em preparação (mas disponível eletronicamente), em `www.labma.ufrj.br/~gregorio`.

Agradeço a um revisor anônimo e a Beatriz Malajovich por ajudarem a depurar este texto.

A pesquisa do autor é apoiada pelo CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico) e pela FAPERJ (Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro).

Aula 1

Condicionamento, e a Variedade Discriminante

1.1 O que não se ensina no curso de Álgebra Linear

Nos cursos de Álgebra Linear, ensinamos a resolver sistemas de equações afins, ou seja, sistemas da forma

$$A\mathbf{x} = \mathbf{b} \ , \tag{1.1.1}$$

onde A é uma matriz $n \times n$, e $\mathbf{b}$ e $\mathbf{x}$ são vetores em $\mathbb{R}^n$. Essa equação tem solução $\mathbf{x}$ única se e somente se $\det(A) \neq 0$.

Nos bons cursos de Álgebra Linear, ensina-se a resolver o sistema (1.1.1) por eliminação Gaussiana ou por fatoração PLU. Isso só vale para sistemas com matriz inversível (e as outras situações são tratadas caso a caso).

Por exemplo, o sistema

$$\begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}$$

admite uma reta de soluções se e somente se $b_3 = b_1 + b_2$.

Gregorio Malajovich, *Geometria de Algoritmos Numéricos*. Notas em Matemática Aplicada **37** (XXXI CNMAC), SBMAC, São Carlos, 2008.
Copyright © Gregorio Malajovich, 2008.

Teoricamente, o sistema

$$\begin{bmatrix} \epsilon & 0 & 1 \\ 0 & 1 & \epsilon \\ 0 & 1 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \tag{1.1.2}$$

tem sempre solução para $\epsilon \neq 0$.

Vamos escolher $\mathbf{b} = [1, 0, 1]^T$ e resolver este último sistema por eliminação: subtraindo a segunda linha da terceira, obtemos:

$$\begin{bmatrix} \epsilon & 0 & 1 \\ 0 & 1 & \epsilon \\ 0 & 0 & 1-\epsilon \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} .$$

Ou seja, $x_3 = 1/(1-\epsilon)$, $x_2 = -\epsilon x_3$ e $x_1 = (1 - x_3)/\epsilon = 1/(1-\epsilon)$. No limite,

$$\lim_{\epsilon \to 0} x_1 = 1.$$

Esses são resultados exatos e teóricos, obtidos simbolicamente. Ao refazer o mesmo exemplo numericamente, fica evidente que a teoria não descreve de maneira adequada a realidade de um computador digital.

Vamos utilizar diversos valores de ϵ:

```
#include <stdio.h>
#include <math.h>

void resolve(float epsilon)
{
float x[4] ;


x[3] = 1.0 /(1.0-epsilon) ;
x[2] = -epsilon * x[3] ;
x[1] = (1.0 - x[3]) / epsilon ;

printf("x = [% 15.12e % 15.12e % 15.12e]\n", x[1],x[2],x[3]) ;
}
```

Podemos agora olhar para os resultados:

```
main()
{
```

```
resolve(pow(2.0, -21)) ;
resolve(pow(2.0, -22)) ;
resolve(pow(2.0, -23)) ;
resolve(pow(2.0, -24)) ;
resolve(pow(2.0, -25)) ;
resolve(pow(2.0, -26)) ;
}
gregorio@momentum:~/cnmac$ ./arredonda
x = [-1.000000000000e+00 -4.768373855768e-07  1.000000476837e+00]
x = [-1.000000000000e+00 -2.384186359450e-07  1.000000238419e+00]
x = [-1.000000000000e+00 -1.192093037616e-07  1.000000119209e+00]
x = [-2.000000000000e+00 -5.960465188082e-08  1.000000119209e+00]
x = [ 0.000000000000e+00 -2.980232238770e-08  1.000000000000e+00]
x = [ 0.000000000000e+00 -1.490116119385e-08  1.000000000000e+00]
```

Quando ϵ é suficientemente pequeno, o resultado obtido para x_1 é zero ! Isso se deve à maneira como os números reais são representados. Computadores trabalham com grandezas aproximadas, representadas como números de ponto flutuante. O formato padrão em uso nos computadores modernos (figuras 1.1 e 1.2) prevê 23 bits de mantissa.

O modelo de aritmética utilizado hoje em todos os computadores é o padrão IEEE754/854 [4, Cap. 2.3]. *Este modelo constitui uma definição axiomática rigorosa da aritmética digital, embora permita diferentes implementações. As regras ou axiomas do padrão permitem provar teoremas sobre algoritmos numéricos.*

Por outro lado, $x_1 = 0$ está longe da solução do sistema (1.1.2). Para entender as implicações de se utilizar aritmética IEEE, é preciso entender não apenas o efeito do arredondamento no algoritmo, mas ainda o conceito de condicionamento, que é independente do algoritmo.

1.2 Arredondamento e condicionamento

Um estudo cuidadoso do algoritmo de eliminação Gaussiana sob a aritmética de ponto flutuante [2, Seção 2.4.2] mostra que, se o computador produzir uma "solução" $\mathbf{x}$ para o problema $A\mathbf{x} = \mathbf{b}$, então $\mathbf{x}$ é a solução exata para um problema aproximado,

$$(A + \delta A)\,\mathbf{x} = \mathbf{b}\ .$$

Além disso, prova-se que $\|\delta A\|_F \leq 3n\epsilon\|L\|_F\|U\|_F$, onde $A = PLU$ é a fatoração PLU de A e $\|\cdot\|_F$ denota a *norma de Frobenius* (Ver Apêndice, seção A.4).

Subtraindo a equação $A\mathbf{x} = \mathbf{b}$, obtemos:

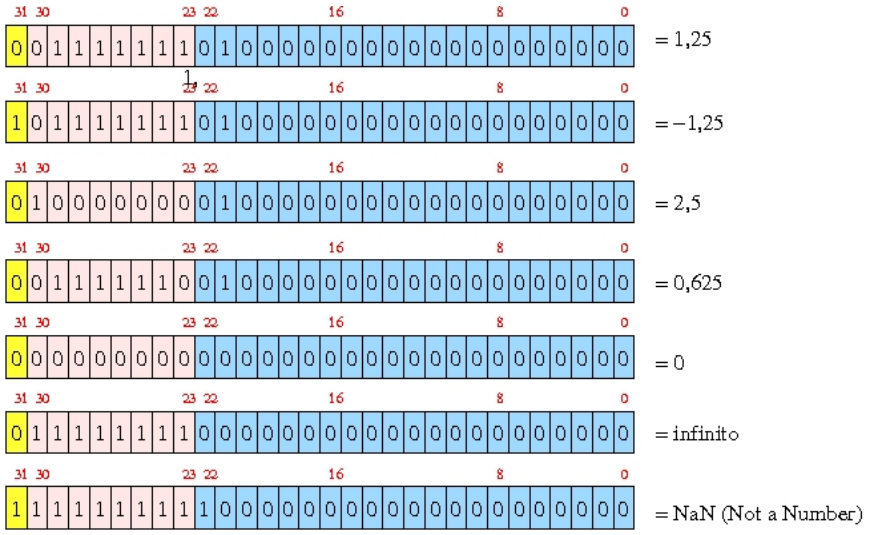

Figura 1.1: Representação de números de ponto flutuante em precisão simples (padrão IEEE 754). Números de ponto flutuante em precisão dupla são representados de maneira análoga com 52 bits de mantissa, 11 de expoente e um de sinal.

Formato	Precisão simples	Precisão dupla	IEEE-854 dupla estendida
Linguagem C	*float*	*double*	*long double*
Bits de mantissa	23	52	64
Bits de expoente	8+1	11+1	15+1
Menor representável > 1	$1 + 2^{-23}$	$1 + 2^{-52}$	$1 + 2^{-64}$
Menor arredondado a 1:	$1 + 2^{-24}$	$1 + 2^{-53}$	$1 + 2^{-65}$

Figura 1.2: Padrão IEEE 754. Dados tirados de `/usr/include/ieee754.h`

$$\delta A\ \mathbf{x} + A\ \delta\mathbf{x} = 0\ .$$

Ou seja,

$$\delta\mathbf{x} = A^{-1}\ \delta A\mathbf{x}$$

e o erro relativo da solução aproximada pode ser estimado por:

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}+\delta\mathbf{x}\|} \leq \|A^{-1}\|_2\|\delta A\|_F \frac{\|\mathbf{x}\|}{\|\mathbf{x}+\delta\mathbf{x}\|}\ .$$

Isso implica:

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}+\delta\mathbf{x}\|} \leq \|A^{-1}\|_2\|A\|_F \frac{\|\delta A\|_F}{\|A\|_F} \frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}+\delta\mathbf{x}\|}\ .$$

Seja $\kappa_A = \|A\|_2\|A^{-1}\|_F$. Nesse caso, obtemos:

$$\frac{\|\delta\mathbf{x}\|}{\|\mathbf{x}+\delta\mathbf{x}\|} \leq \kappa_A \frac{\|\delta A\|_F}{\|A\|_F}\ .$$

Em outras palavras: *O processo numérico utilizado para computar* $\mathbf{x}$ *produziu a solução exata de um problema aproximado. O erro relativo na solução* $\mathbf{x}$ *pode ser estimado pelo produto do erro relativo no problema, vezes o número* κ_A.

O número κ_A **não depende** do algoritmo utilizado para resolver o sistema de equações. Ele é um invariante que depende apenas dos coeficientes do sistema. É chamado de número de condicionamento da matriz A, associado ao problema de resolver $A\mathbf{x} = \mathbf{b}$.

No exemplo numérico acima, $\kappa_A \simeq \sqrt{2}\epsilon^{-1}$.

1.3 O polinômio pérfido

O polinômio *pérfido* de grau d (também conhecido como polinômio de Pochammer) é definido por:

$$p_d(x) = (x-1)(x-2)\cdots(x-d)\ .$$

Por exemplo,

$$\begin{aligned} p_{10}(x) &= x^{10} - 55x^9 + 1320x^8 - 18150x^7 + 157773x^6 - 902055x^5 + \\ &+ 3416930x^4 - 8409500x^3 + 12753576x^2 - 10628640x^1 + 3628800. \end{aligned}$$

Uma maneira de se encontrar a raízes de polinômios de grau baixo é produzir a *matriz companheira* associada a eles. Se $f(x) = x^d + f_{d-1}x^{d-1} + \cdots + f_1 x + f_0$, então a matriz companheira de f é

$$C_f = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ & \ddots & \ddots & & \vdots \\ & & \ddots & \ddots & \vdots \\ & & & 0 & 1 \\ -f_0 & -f_1 & \cdots & -f_{d-2} & -f_{d-1} \end{bmatrix}.$$

A matriz companheira foi construída de maneira que f fosse o seu polinômio característico (a menos do sinal):

$$\det C_f - \lambda I = (-1)^d f(\lambda).$$

Assim, reduzimos o problema de resolver um polinômio de grau d a outro problema, que é o de achar os autovalores de uma matriz $d \times d$. Existe excelente *software* numérico para achar autovalores. Vamos aplicar essa idéia ao polinômio pérfido.

```
p=poly(1:10)
C=[ [zeros(9,1),eye(9)]; -p(11:-1:2)]
x=eig(C)
```

```
x =

  10.00000
   9.00000
   8.00000
   7.00000
   6.00000
   5.00000
   4.00000
   3.00000
   2.00000
   1.00000
```

A solução **parece** correta. Mas conhecemos a solução exata, e podemos conferir:

```
x - [10:-1:1]´
ans =
```

```
   4.3965e-11
  -2.1128e-10
   4.4744e-10
  -5.4329e-10
   4.0626e-10
  -1.8595e-10
   4.9097e-11
  -6.6769e-12
   4.0634e-13
  -1.5654e-14
```

Mais uma vez, a solução **parece** correta. O fato do *Octave* usar aritmética de dupla precisão deveria no entanto levantar suspeitas. O erro relativo de cada operação aritmética em precisão dupla é de no máximo $2^{-53} \simeq 10^{-16}$. Não está claro se o resultado é acurado. O mesmo acontece se utilizamos o comando `roots`.

Vamos agora repetir o experimento, com grau 20.

```
 p=poly(1:20)                                    13.0553
 C=[ [zeros(19,1),eye(19)]; -p(21:-1:2)]         11.9753
 x=eig(C)                                        11.0092
                                                  9.9975
x =                                               9.0005
                                                  7.9999
  19.9994                                         7.0000
  19.0056                                         6.0000
  17.9769                                         5.0000
  17.0524                                         4.0000
  15.9092                                         3.0000
  15.1021                                         2.0000
  13.9168                                         1.0000
```

O seguinte experimento mostra que a dificuldade em se resolver polinômios pérfidos não é *bug* do *software* ou deficiência do algoritmo:

```
p=poly(1:10)
p(11)=p(11)*1.0001
C=[ [zeros(9,1),eye(9)]; -p(11:-1:2)]
x=eig(C)
x =

  9.9990
```

```
9.0089
7.9624
7.0807
5.8673
5.1327
3.9193
3.0376
1.9911
1.0010
```

Uma perturbação de 0.01% em um dos coeficientes provocou uma perturbação de 2% nos valores das raízes.

Para explicar o ocorrido, introduzimos o número de condicionamento de Wilkinson para polinômios: se ζ é um zero isolado de f, define-se o condicionamento por

$$\kappa_f(\zeta) = \frac{\|f\|}{|f'(\zeta)|} \ .$$

Se z for uma solução exata do polinômio $f + \delta f$, então podemos definir implicitamente:

$$(f + t\delta f)(z(t)) = 0$$

com $z(1) = z$. Derivando em relação a t,

$$(\delta f)(z(t)) + (f + \delta f)'(\dot{z}(t)) = 0$$

e logo

$$\|\dot{z}(t)\| \leq \frac{1}{|(f + \delta f)'(z(t))|} \|\delta f\| \left\| \begin{bmatrix} 1 \\ z(t) \\ z(t)^2 \\ \vdots \\ z(t)^{\deg f} \end{bmatrix} \right\| \ .$$

Podemos concluir (com um pouco mais de trabalho) que o erro $\|\delta z\|$ é no máximo da ordem de

$$\kappa_f(\zeta) \frac{\|\delta f\|}{\|f\|} \sqrt{\deg f} \max(1, |\zeta|)^{\deg f} \ .$$

1.4 Teoria geral

Números de condicionamento foram estudados e definidos para os mais diversos problemas, como por exemplo problemas de autovalores, solução de sistemas de polinômios, mínimos quadrados, etc... (Ver [1, 2]).

A teoria geral de números de condicionamento utiliza conceitos de geometria diferencial. Vamos introduzir abaixo alguns conceitos fundamentais de *Cálculo em Variedades*, essenciais para o bom entendimento do que segue.

Em primeiro lugar, uma função diferenciável é uma função derivável, com derivada contínua. Se for uma função vetorial, exigimos que todas as derivadas parciais em relação a todas as variáveis sejam diferenciáveis, e representamos a derivada pela matriz das derivadas parciais.

Definição 1.1. Seja

$$\begin{array}{rccc} F: & \mathbb{R}^n & \to & \mathbb{R}^m \\ & \mathbf{x} & \mapsto & F(\mathbf{x}) \end{array}$$

uma função diferenciável, com $n \geq m$. Um *ponto crítico* de F é um $\mathbf{x} \in \mathbb{R}^n$ tal que $DF_{\mathbf{x}}$ tenha posto $< m$, ou seja, que as colunas de $DF_{\mathbf{x}}$ não sejam independentes, ou ainda que a aplicação linear $DF_{\mathbf{x}}$ não seja sobrejetora. Um *ponto regular* é um $\mathbf{x} \in \mathbb{R}^n$ que não é crítico. Definimos ainda um *valor crítico* de F como um $y \in \mathbb{R}^m$ que seja imagem por F de um ponto crítico. Um $y \in \mathbb{R}^m$ é dito *valor regular* se ele não é imagem de nenhum ponto crítico.

Definição 1.2. Uma subvariedade diferenciável d-dimensional M implícita em $\mathbb{R}^n$ é um conjunto da forma $F^{-1}(\mathbf{0})$, onde

$$\begin{array}{rccc} F: & \mathbb{R}^n & \to & \mathbb{R}^m \\ & \mathbf{x} & \mapsto & F(\mathbf{x}) \end{array}$$

é uma função diferenciável, e $\mathbf{0}$ é valor regular de F.

Neste texto, uma *variedade* é sempre uma subvariedade diferenciável d-dimensional implícita de algum espaço linear. Por exemplo, o conjunto das matrizes $n \times n$ de determinante zero é uma variedade.

Se $M = F^{-1}(\mathbf{0})$ é uma variedade (no nosso sentido) e $\mathbf{x}$ um ponto de M, definimos o espaço tangente a M em x como

$$T_{\mathbf{x}}M = \ker DF_{\mathbf{x}} \ .$$

Existe uma parametrização de $T_{\mathbf{x}}M$ em uma vizinhança de $\mathbf{x} \in M$. Dessa maneira, podemos (localmente) introduzir sistemas de coordenadas em M. Se $F : \mathbf{R}^n \to \mathbb{R}^m$, então $T_{\mathbf{x}}M$ tem dimensão $n - m$ e dizemos que M tem dimensão $m - n$.

Se M e N são variedades e $\Phi : M \to N$, então definimos $D\Phi_{\mathbf{x}} : T_{\mathbf{x}} \to T_{\Phi(\mathbf{x})}$ como a melhor aproximação linear de Φ (se existir). Dessa maneira, podemos fazer cálculo em variedades.

Uma definição diferente de variedade é utilizada em *geometria algébrica*. Um *fechado de Zariski* é o conjunto dos zeros de uma coleção de polinômios (reais, complexos, etc...). Uma *variedade algébrica* é um fechado de Zariski, que não se escreve como união trivial de fechados de Zariski. A variedade das matrizes A tais que $\det A = 0$, por exemplo, é uma variedade algébrica.

Um problema numérico é definido por um espaço (ou variedade !) $\mathcal{F}$ de entrada (por exemplo, as matrizes $n \times n$) e um espaço X de soluções (exemplo: autovalor $\lambda \in \mathbb{C}$). Existe uma regra associando soluções às entradas. Essa regra não é nem pode ser uma função unívoca, pois certos problemas (por exemplo, o problema de autovalores) admitem várias soluções.

No exemplo, um número complexo λ é autovalor de A se e somente se $\det(A - \lambda I) = 0$. Definimos portanto a seguinte variedade de $\mathcal{F} \times \mathbb{C}$, chamada de *variedade de incidência* (Figura 1.3):

$$V = \{(A, \lambda) \in \mathcal{F} \times \mathbb{C} : \det(A - \lambda I) = 0\} \tag{1.4.3}$$

(A prova de que V é variedade é um exercício). $\pi_1 : V \to \mathcal{F}$ e $\pi_2 : V \to \mathbb{C}$ denotam as projeções canônicas.

Dado $(M, \lambda) \in V \subset \mathcal{F} \times X$, podem acontecer duas situações:

- (A, λ) é ponto crítico da projeção $\pi_1 : V \to \mathcal{F}$. Nesse caso, a entrada A é chamada de mal-posta, ou degenerada. Por definição, o número de condicionamento $\mu(A, \lambda)$ em (A, λ) é infinito.

- (A, λ) é ponto regular da projeção $V \to F$. Nesse caso (Teorema da Função Implícita !) podemos estender λ como função $(\pi_2 \circ \pi_1^{-1})$de A, *localmente*, em uma certa vizinhança de A. O número de condicionamento $\mu(A, \lambda)$ em (A, λ) é a norma da derivada da função implícita $(\pi_2 \circ \pi_1^{-1})$ em A.

Em geral, *o número de condicionamento é proporcional à norma da derivada da solução em função da entrada,* podendo ser infinito. Se for grande, a entrada é dita "mal condicionada". Se for infinita, ela é degenerada ou "mal posta". Se pudéssemos impunemente brincar com a precisão de nossos computadores, *o número de bits necessário para resolver um problema com uma certa entrada seria proporcional ao logaritmo do condicionamento desta.*

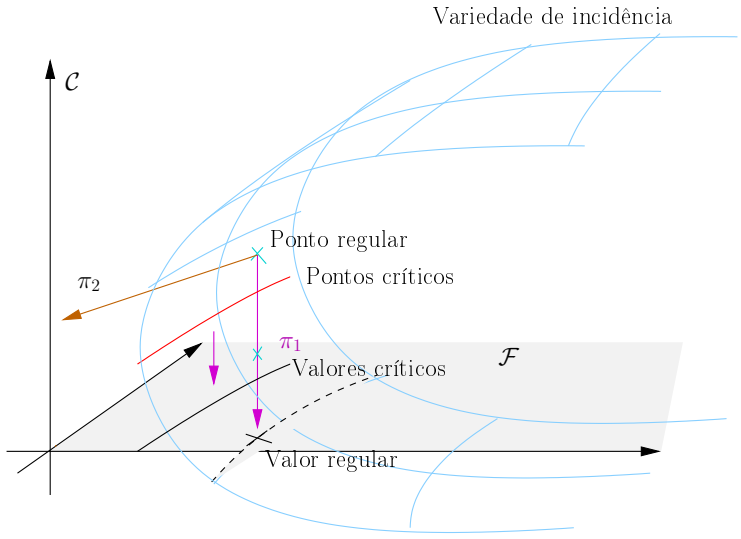

Figura 1.3: Formulação geral.

1.5 Os problemas mal postos

Os números de condicionamento medem a dificuldade de resolver um problema para determinada entrada. A esse título, constituem a noção central em análise numérica, e números de condicionamento altos são a principal obstrução para algoritmos numéricos.

Identificada a obstrução, cumpre entendê-la, para depois tomar as devidas providências (contornar? remover? explodir? conformar-se?).

Na maioria dos problemas numéricos estudados em dimensão finita, o conjunto das entradas mal-postas tem estrutura de variedade algébrica. Essa variedade não é, em geral, diferenciável. Por exemplo, no caso do problema de resolver sistemas lineares, as entradas mal-postas são precisamente:

$$\{A : \det A = 0\} \ .$$

No caso de polinômios em uma variável, a classe das entradas mal postas é a variedade dos polinômios com uma raiz múltipla. Em função dos coeficientes, isso equivale a escrever:

$$\det \begin{bmatrix} f_0 & & & f_1 & & & \\ f_1 & \ddots & & \vdots & \ddots & & \\ \vdots & & f_0 & (d-1)f_{d-1} & & \ddots & \\ \vdots & & f_1 & df_d & \ddots & & f_1 \\ f_d & & \vdots & & \ddots & \ddots & \vdots \\ & \ddots & \vdots & & & \ddots & (d-1)f_{d-1} \\ & & f_d & & & & df_d \end{bmatrix} = 0 \ .$$

A variedade das entradas mal postas é um objeto extremamente complicado, mas tem sempre codimensão ≥ 1 (i.e. dimensão $N-1$ em um espaço de dimensão N).

Podemos deduzir que

- A probabilidade de uma entrada aleatória ser mal posta é zero.
- A variedade das entradas mal postas não desconecta a variedade de todas as entradas... **desde que se trate de variedades complexas.**
- Superfície e área de vizinhança tubular podem ser estimadas.

1.6 A decomposição em valores singulares

Precisamos agora de alguns teoremas de Álgebra Linear. O principal deles é o teorema da decomposição em valores singulares, e vamos deduzi-lo do Teorema Espectral.

Teorema 1.3 (T. Espectral para matrizes simétricas)**.** *Seja S uma matriz simétrica real $n \times n$. Então todos os autovalores de S são reais, e S admite uma base ortonormal de autovetores.*

A prova foi deixada para os exercícios. Admitindo esse resultado, podemos passar ao Teorema seguinte.

Seja $A : \mathbb{R}^n \to \mathbb{R}^m$ uma aplicação linear. Os espaços R^n e R^m são munidos do produto interno canônico.

Vamos mostrar que existe uma base ortonormal $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ de $\mathbb{R}^n$, e uma base ortonormal $(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ de $\mathbb{R}^m$, tais que a matriz associada A relativa a essas duas bases é diagonal.

Uma matriz Σ de tamanho $m \times n$ é diagonal positiva se e somente se para todo i, $\Sigma_{ii} \geq 0$, e para $i \neq j$, $\Sigma_{ij} = 0$.

Teorema 1.4. *Seja A uma matriz real de tamanho $m \times n$. Então existem $U \in O(m)$, $V \in O(n)$ e Σ diagonal positiva de tamanho $m \times n$, tais que*

$$A = U \Sigma V^T \ .$$

Demonstração: Para fixar as idéias, vamos assumir que $m \geq n$. (No caso $m < n$, basta substituir A por A^T).

A matriz $A^T A$ é real e simétrica. Pelo Teorema Espectral, ela admite uma base ortonormal $(\mathbf{v}_1, \ldots, \mathbf{v}_n)$ de autovetores. Denotamos por λ_i os autovalores de $A^T A$. Para todo $i = 1, \ldots, n$, definimos

$$\sigma_i = \|A\mathbf{v}_i\|$$

e assumimos que a base $\mathbf{v}_i$ está ordenada de maneira que $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n$. Seja r o posto de A, então $\sigma_1, \ldots, \sigma_r \neq 0$ e para todo $i \in \{1, 2, \ldots, r\}$, podemos definir

$$\mathbf{u}_i = \sigma_i^{-1} A \mathbf{v}_i \ .$$

Por construção, $\|\mathbf{u}_i\| = 1$. (Note que isso implica que $\sigma_i^2 = \lambda_i$). Como para todo $i \neq j$, $\mathbf{v}_i (A^T A) \mathbf{v}_j = \lambda_j \mathbf{v}_i^T \mathbf{v}_j = 0$, teremos que $\langle \mathbf{u}_i, \mathbf{u}_j \rangle = 0$ e os $\mathbf{u}_i$ formam um conjunto ortonormal.

Existe uma base $(\mathbf{u}_1, \ldots, \mathbf{u}_r, \mathbf{w}_{r+1}, \ldots, \mathbf{w}_m)$ de $\mathbb{R}^n$. Aplicando Gram-Schmidt a essa base, obtemos um base ortonormal $(\mathbf{u}_1, \ldots, \mathbf{u}_m)$. Como

todo vetor ortogonal à imagem de A pertence ao núcleo de A^T,

$$A = U \begin{bmatrix} \sigma_1 & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & 0 & \cdots & 0 \\ 0 & 0 & \sigma_3 & & \\ \vdots & \vdots & & \ddots & \\ 0 & 0 & & & \sigma_r \\ 0 & 0 & & & 0 \\ \vdots & \vdots & & & \vdots \\ 0 & 0 & & & 0 \end{bmatrix} V^T \ .$$

■

Os σ_i's são chamados de valores singulares de A, e os $\mathbf{u}_i$'s e $\mathbf{v}_i$'s de vetores singulares.

Uma interpretação geométrica é a seguinte. Assuma que A é de tamanho $m \times n$, com $m \geq n$ e posto n). Seja

$$\mathcal{E} = \{A\mathbf{x} : \|\mathbf{x}\| \leq 1\} \ .$$

O conjunto $\mathcal{E}$ é o elipsóide de centro zero, e semieixos $\sigma_i \mathbf{u}_i$.

1.7 Os problemas mal condicionados

Lembremos que o número de condicionamento de uma matriz A é dado por

$$\kappa_A = \|A\|_2 \|A^{-1}\|_F \ .$$

Seja $\Sigma = \{A : \det A = 0\}$ a variedade das entradas mal postas.

Teorema 1.5 (Eckart-Young)**.**

$$\kappa_A = \frac{1}{d(\Sigma, A)/\|A\|_F}$$

onde $d(\Sigma, A) = \min_{(A+\delta A)\in\Sigma} \|\delta A\|_F$.

Prova: Vamos utilizar a decomposição em valores singulares de A. Pelo Teorema 1.4, toda matriz A se escreve como:

$$A = U\Sigma V$$

onde U e V são matrizes ortogonais, e Σ é uma matriz diagonal real não-negativa. Seja

$$\Sigma = \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix}$$

com $\sigma_1 \geq \sigma_2 \geq \cdots \sigma_n > 0$. Então, podemos escrever explicitamente:

$$\|A\|_F = \sqrt{\sum \sigma_i^2}$$

e

$$\kappa_A = \sqrt{\sum \sigma_i^2}/\sigma_n \ .$$

A menor perturbação que torna A singular é:

$$\delta_A = U \begin{bmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & -\sigma_n \end{bmatrix} V^T$$

que tem norma σ_n. Logo, $d(A, \Sigma)/\|A\|_F = \sigma_n/\|A\|_F = 1/\kappa_A$, q.e.d.

Resultados análogos foram descobertos para problemas como o problema de autovalores, problema de mínimos quadrados, solução de polinômios, de sistemas de polinômios.

Esses resultados sugerem o seguinte paradigma: *O número de condicionamento pode ser interpretado como o inverso da distância à variedade das entradas mal postas.*

Como todo paradigma, deve ser tomado com um certo cuidado. No caso de polinômios e sistemas de polinômios, a igualdade é um pouco mais sutil: o número de condicionamento $\mu(f, \zeta)$ é a inversa da distância medida dentro da "fibra" das entradas f com solução em ζ ([1, Cap.12]).

No caso do problema de autovalores não-simétrico, a distância é menor ou igual a $1/\sqrt{c^2 - 1} \simeq 1/c$, onde c é o número de condicionamento ([2, Teorema 4.6]).

No entanto, o paradigma acima permite atacar a perguntas como a seguinte: *qual é a probabilidade de uma matriz ter número de condição menor do que um certo* ϵ^{-1} ? Para isso precisamos definir uma medida de probabilidade sobre as matrizes. Por exemplo, se as coordenadas são variáveis aleatórias Gaussianas, identicamente e independentemente distribuídas, Alan Edelman mostrou [3, Cor. 7.1] que

$$\lim_{n \to \infty} \text{Prob}[\kappa < \epsilon^{-1}] = e^{-2n\epsilon - 2n^2\epsilon^2} \ .$$

Pelo Teorema de Eckart-Young, estamos calculando o volume (probabilidade total) de uma vizinhança tubular de raio ϵ da variedade algébrica $\det A = 0$.

1.8 Exercícios

Exercício 1.1. Explique os resultados obtidos pelo programa da página 10

Exercício 1.2. Mostre que se S é uma matriz real simétrica, então seus autovalores e autovetores são reais.

Exercício 1.3. Mostre que se u é autovalor de S, e se $v \perp u$, então $Sv \perp u$.

Exercício 1.4. Prove o Teorema Espectral.

Exercício 1.5. Mostre que a variedade algébrica $\{A : \det(A) = 0\}$, onde A é uma matriz $n \times n$ e $n \geq 2$, não é uma variedade diferenciável implícita.

Exercício 1.6. Mostre que toda matriz complexa quadrada A se escreve como $A = QRQ^*$, onde R é triangular superior e $QQ^* = I$.

Exercício 1.7. Mostre que o conjunto V da equação (1.4.3) é uma variedade.

Exercício 1.8. Ache uma expressão para o número de condicionamento do problema de autovalores.

Referências

[1] Lenore Blum, Felipe Cucker, Michael Shub, and Steve Smale, *Complexity and real computation*, Springer-Verlag, New York, 1998.

[2] James W. Demmel, *Applied numerical linear algebra*, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1997.

[3] Alan Edelman, *Eigenvalues and condition numbers of random matrices*, Ph.D. Thesis, Department of Mathematics, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[4] Nicholas J. Higham, *Accuracy and stability of numerical algorithms*, 2nd ed., Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2002.

Aula 2

A iteração de Gräffe, Dandelin ou Lobachevskii

Na década de 1830, a Academia de Ciências de Berlim ofereceu um prêmio para o melhor método de se achar zeros de polinômios.

O agraciado foi Karl Heinrich Gräffe (1799-1873) cujo trabalho (publicado em 1837) propõe elevar as raízes ao quadrado, por meio da iteração $f(x) \mapsto \pm f(\sqrt{x}) f(\sqrt{-x})$. Após um pequeno número k de iterações, é possível recuperar as raízes 2^k-ésimas das raízes de f.

O método já havia sido esboçado em 1826 por Germinal Pierre Dandelin (1794-1847) (Fig.2.1), matemático, engenheiro militar e futuro rev-

Gregorio Malajovich, *Geometria de Algoritmos Numéricos*. Notas em Matemática Aplicada **37** (XXXI CNMAC), SBMAC, São Carlos, 2008.
Copyright © Gregorio Malajovich, 2008.

Figura 2.1: Germinal Dandelin (1794-1847) e Nicolai Ivánovich Lobachevskii (1792-1856).

olucionário Belga. Esse método passou a ser conhecido no ocidente como método de Dandelin-Gräffe.

Foi descoberto independentemente por Nicolai Ivánovich Lobachevskii (1792-1856) (Fig.2.1), hoje mais conhecido pela invenção da geometria não Euclidiana. O Tratado de Álgebra de Lobachevskii foi entregue ao censor em 1832, mas só foi publicado em 1834. Pesou contra ele o fato de estar em Kazan, longe dos centros intelectuais da Europa. Nos textos Soviéticos, falava-se do algoritmo de Lobachevskii.

Cada um deles apresentou contribuições importantes para o "método", que foi tornado mais rigoroso e eficiente ao longo de dois séculos.

Nesta aula, pretendo apresentar uma visão moderna da iteração de Gräffe-Dandelin-Lobachevskii. Quero mostrar que apesar de quase bicentenário, o algoritmo ilustra vários tipos de conexões de problemas numéricos com áreas correntes da matemática.

A principal referência para a versão moderna do algoritmo é o artigo [3], escrito em parceria com Jorge P. Zubelli. A referência histórica é [1].

2.1 A iteração

Seja

$$f(x) = \sum_{j=0}^{d} f_j x^j \tag{2.1.1}$$

um polinômio de grau d, com coeficientes reais ou complexos. Para simplificar as contas, vamos assumir que $f_d = 1$ (f é *mônico*).

Do Teorema Fundamental da Álgebra, sabemos que f admite d raízes complexas, contadas com multiplicidade. Sejam $\zeta_1, \ldots, \zeta_d$ essas raízes, e assumimos que estão ordenadas de modo que

$$|\zeta_1| \leq |\zeta_2| \leq \cdots \leq |\zeta_d| \ .$$

O polinômio $f(x)$ também se escreve

$$f(x) = (x - \zeta_1)(x - \zeta_2) \cdots (x - \zeta_d) \ . \tag{2.1.2}$$

Igualando (2.1.1) a (2.1.2), deduzimos que os coeficientes do polinômio f são (a menos de um sinal) as funções simétricas das raízes:

$$f_j = (-1)^{d-j} \sigma_{d-j}(\zeta_1, \ldots, \zeta_d) \ . \tag{2.1.3}$$

com

$$\begin{aligned}
\sigma_0(\zeta_1,\ldots,\zeta_d) &= 1\\
\sigma_1(\zeta_1,\ldots,\zeta_d) &= \zeta_1+\zeta_2+\cdots+\zeta_d\\
\sigma_2(\zeta_1,\ldots,\zeta_d) &= \zeta_1\zeta_2+\zeta_1\zeta_3+\cdots+\zeta_{d-1}\zeta_d\\
&\vdots\\
\sigma_d(\zeta_1,\ldots,\zeta_d) &= \zeta_1\zeta_2\cdots\zeta_d
\end{aligned}$$

A iteração de Gräffe leva o polinômio $f(x)$ no polinômio $(Gf)(x) = (-1)^d f(\sqrt{x})f(\sqrt{-x})$. Dados os coeficientes de $f(x)$, podemos calcular os coeficientes de $(Gf)(x)$ pelo algoritmo da multiplicação ou *convolução*:

$$(Gf)_j = \sum_{i=-\min(j,d-j)}^{\min(j,d-j)} (-1)^{j-i} f_{j+i} f_{j-i}.$$

Essa conta pode ser feita de maneira extremamente rápida utilizando a transformada rápida de Fourier.

Por outro lado,

$$\begin{aligned}
Gf(x) &= (\sqrt{x}-\zeta_1)(\sqrt{(}x)+\zeta_1)\cdots(\sqrt{x}-\zeta_d)(\sqrt{(}x)+\zeta_d)\\
&= (\sqrt{x}-\zeta_1^2)\cdots(\sqrt{(}x)+\zeta_d^2)
\end{aligned}$$

Vamos agora iterar esse processo. Escrevemos

$$g(x) = G^N f(x) = \left((\underbrace{G\circ G\circ\cdots\circ G}_{N \text{ vezes}})(f) \right)(x) \ .$$

Vamos denotar por g_i os coeficientes de $g(x) = G^N f(x)$. Como

$$g(x) = (x-\zeta_1^{2^N})(x-\zeta_2^{2^N})\cdots(x-\zeta_d^{2^N}) \ ,$$

teremos:

$$\begin{aligned}
g_d &= 1\\
g_{d-1} &= -\left(\zeta_1^{2^N}+\zeta_2^{2^N}+\cdots+\zeta_d^{2^N}\right)\\
g_{d-2} &= \zeta_1^{2^N}\zeta_2^{2^N}+\zeta_1^{2^N}\zeta_3^{2^N}+\cdots+\zeta_{d-1}^{2^N}\zeta_d^{2^N}\\
&\vdots\\
g_0 &= (-1)^d\zeta_1^{2^N}\zeta_2^{2^N}\cdots\zeta_d^{2^N}
\end{aligned}$$

Vamos introduzir agora uma hipótese simplificadora: assumimos temporariamente que

$$|\zeta_1| < |\zeta_2| < \cdots < |\zeta_d| \ . \tag{2.1.4}$$

Seja $R = \min \frac{\zeta_j}{\zeta_{j-1}}$. Então $R > 1$. Temos agora:

$$\begin{aligned} g_d &= 1 \\ g_{d-1} &= -\zeta_d^{2^N}(1+\delta_{d-1}) \\ g_{d-2} &= \zeta_{d-1}^{2^N}\zeta_d^{2^N}(1+\delta_{d-2}) \\ &\vdots \\ g_1 &= (-1)^{d-1}\zeta_2^{2^N}\zeta_3^{2^N}\cdots\zeta_d^{2^N}(1+\delta_1) \\ g_0 &= (-1)^d\zeta_1^{2^N}\zeta_2^{2^N}\cdots\zeta_d^{2^N} \end{aligned}$$

onde os δ_j são pequenos (quando R é fixo e k é grande): uma estimativa óbvia é

$$|\delta_j| < R^{-2^N}\binom{d}{j} \ .$$

Até a primeira metade do século XX, era usual representar os g_i em papel logarítmico (com a escala certa) e aproximar $\log|\zeta_i|^{2^N}$ pelo coeficiente da reta contendo $(j, -\log|g_j|)$ e $(j+1, -\log g_{j+1})$. Esse gráfico era conhecido como *diagrama de Newton* (Fig 2.2).

Sobraram dois problemas:

1. Como lidar com raízes com o mesmo módulo ? Por exemplo, $f_1(x) = x^2 - 2x + 1$?
2. Como recuperar os argumentos ? Se $f_2(x) = x^2 - 1$, então $Gf_1(x) = Gf_2(x) = f_2(x)$, e perdemos informação !

A solução histórica era iterar $f(x+\epsilon)$ e $f(x-\epsilon)$, obtendo assim $|\zeta_i - \epsilon|$ e $|\zeta_i + \epsilon|$.

2.2 Papel logarítmico, renormalização e Geometria Algébrica Tropical

Vimos que não é natural assumir que as raízes de um polinômio têm módulo diferente. De fato, existe uma probabilidade positiva de um polinômio real

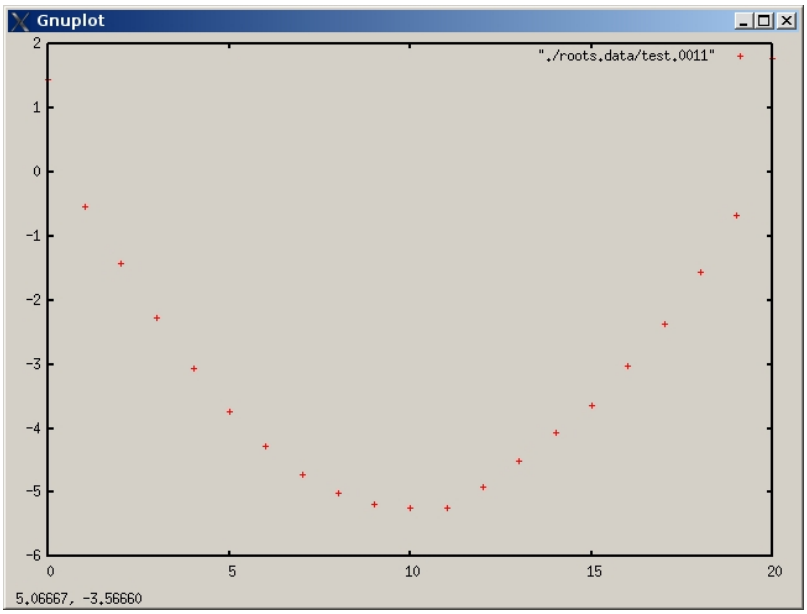

Figura 2.2: Diagrama de Newton, polinômio complexo aleatório de grau 20, após 11 iterações.

aleatório ter raízes complexas conjugadas. Vamos portanto substituir a hipótese (2.1.4) por uma hipótese muito mais fraca:

Hipótese: *Se duas raízes de f têm mesmo módulo, é por uma das seguintes razões: elas são iguais, ou f é real e as raízes são conjugadas.*

Nesse caso, dizemos que f é *livre de círculos*.

Essa é uma hipótese genérica e de probabilidade um. Se f não for livre de círculos, sempre podemos aplicar uma transformação conforme aleatória e obter (com probabilidade 1) um polinômio livre de círculos. De agora em diante, assumimos que f é livre de círculos.

A cada etapa de iteração, representamos os coeficientes da N-ésima iterada $g(x) = G^N f(x)$ em coordenadas logarítmicas *renormalizadas*:

$$g_k = e^{2^N r_k^i(N)} e^{\sqrt{-1}\alpha_i^{(N)}} .$$

Ou ainda:

$$\begin{aligned} r_k^{(N)} &= 2^{-N} \log |g_k| \\ \alpha_i^{(N)} &= \arg g_k \end{aligned}$$

Isso era feito *implicitamente* no início do século XX, utilizando papel logarítmico. A mudança de escala era feita, mas não escrita.

Se valesse a hipótese (2.1.4), teríamos o seguinte fato: embora a seqüência dos $(G^N f(x))_{N\in\mathbb{N}}$ seja divergente, a seqüência dos $(r^{(N)})_{N\in\mathbb{N}}$ converge para um limite que nos fornece informação relevante.

Esse é um exemplo de *renormalização*, técnica utilizada em autômatos celulares e sistemas dinâmicos. (Renormalização em física pode ser um pouco diferente).

Por outro lado, a dinâmica dos argumentos $\alpha_k^{(N)}$ é tipicamente caótica, no limite $\alpha_k^{(N)} \simeq 2\alpha_k^{(N-1)} \mod 2\pi$. É importante entender que esse fato é trivial e absolutamente irrelevante. A informação guardada nos $\alpha_k^{(N)}$ é *perdida*, e aqui entender sistemas dinâmicos caóticos não ajuda em nada.

A convergência da seqüência $(r^{(N)})_{N\in\mathbb{N}}$ depende da hipótese (2.1.4). O que acontece com polinômios livres de círculos é mais complicado. Esse fenômeno foi descrito por Ostrowskii [4, 5], mas o fato de renormalizarmos os $r^{(N)}$ permite um entendimento melhor:

Seja $\hat{r}(N)$ a maior função convexa em $[0, d]$ tal que $\hat{r}^{(N)}(k) \leq r_k^{(N)}$, $k = 0, \ldots, d$.

Se f é livre de círculos, então $\hat{r}^{(N)}$ é convergente (Figura 2.3). A velocidade de convergência pode ser estimada em função de R.

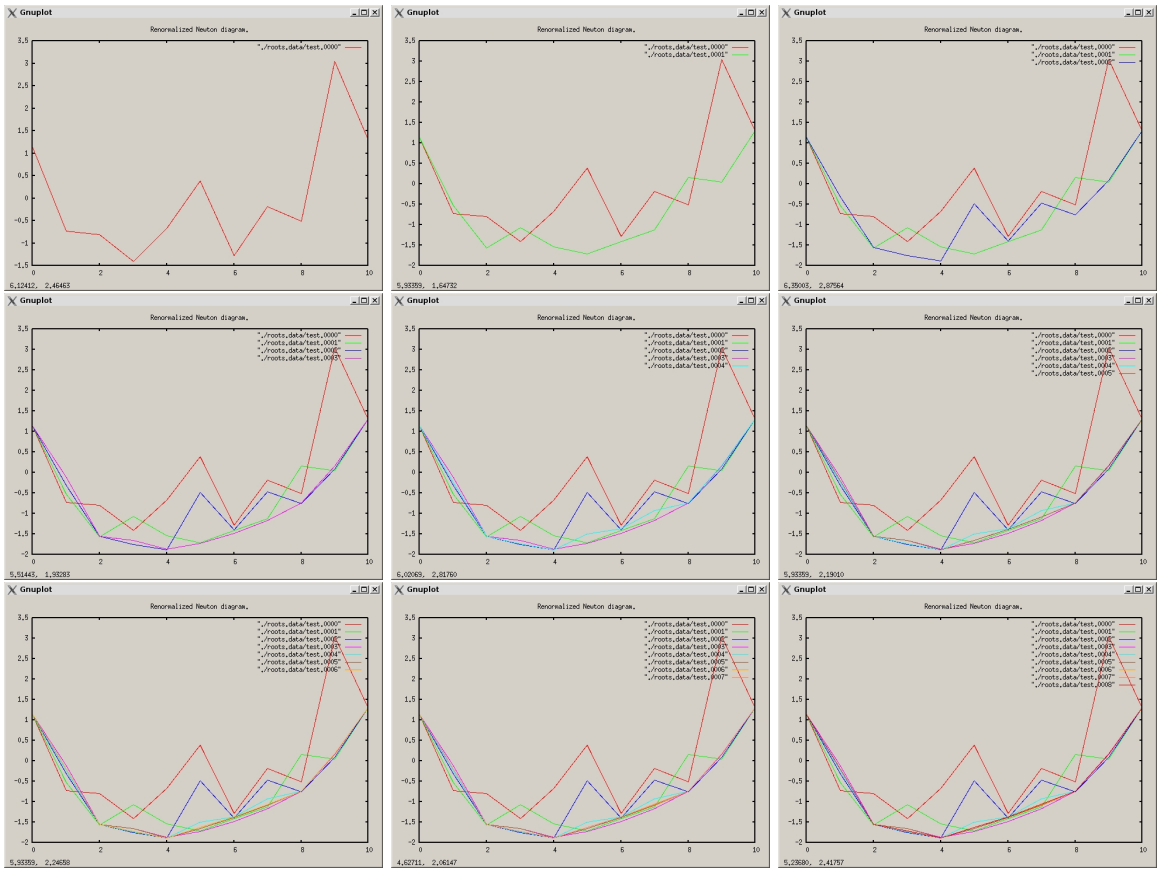

Figura 2.3: Diagrama de Newton renormalizado, polinômio aleatório real de grau 10.

Até a invenção do computador digital, a iteração de Gräffe-Dandelin-Lobachevskii e suas variantes eram o algoritmo escolhido para resolver polinômios.

Isso mudou radicalmente com o advento do computador digital. Após as primeiras experiências numéricas, ficou claro que o *expoente* "estourava" rapidamente (Fig 1.1).

Por exemplo, se começamos com coeficientes da ordem de 2, em 10 iterações apenas teremos coeficientes da ordem de 2^{2^10}, que precisam de 2^10 bits em representação inteira exata, e de em torno de 10 bits de expoente em representação mantissa-potência de dois. Hoje utilizamos em torno de 10 bits de expoente.

Como apareceram algoritmos mais simples de implementar (ou implementados por outras razões), o algoritmo ficou esquecido.

Utilizando a idéia de renormalização, não ocorre "estouro" do expoente, e a cada passo podemos fazer contas renormalizadas da maneira seguinte:

Representação usual:	Renormalização de ordem N:
$x = e^{2^N r+i\alpha}$	(r,α)
$y = e^{2^N s+i\beta}$	(s,β)
$x+y = e^{2^N t+i\gamma}$	$(t,\gamma) = (r,\alpha) \boxed{+}_k (s,\beta)$
$xy = e^{2^N u+i\delta}$	$(u,\delta) = (r,\alpha)+(s,\beta)$

Onde (assumindo $r \geq s$:)

$$\begin{aligned} t &= 2^{-N}\log\left|e^{2^N r+i\alpha}+e^{2^N s+i\beta}\right| \\ &= r\log\left|1+e^{2^N(s-r)+i(\beta-\alpha)}\right| \end{aligned}$$

Note que

$$e^{2^N(s-r)+i(\beta-\alpha)} = \frac{e^{i(\beta-\alpha)}}{1+2^N(r-s)+\frac{1}{2}2^{2N}(r-s)^2+\cdots}\ .$$

A *soma renormalizada* pode ser calculada de maneira estável. Quando $2^N \gg |r-s|$, pode-se até aproximar $(r,\alpha)\boxed{+}_k(s,\beta)$ pelo limite,

$$(r,\alpha)\boxed{+}_\infty(s,\beta) = \begin{cases} (r,\alpha) & \text{Se } r > s \\ (s,\beta) & \text{Se } s < r \\ (r,\text{Indefinido}) & \text{Se } r = s \end{cases}\ .$$

A primeira coordenada desse limite é conhecida hoje como *soma tropical* de r e s.

Definição 2.1. O semianel tropical é o anel $\mathbb{R}\cup -\infty$ munido das operações de *soma tropical* $r\boxed{+}s = \max(r,s)$ e *produto tropical* $r\boxed{\times}s = r+s$.

Existe também uma conexão com geometria simplética. Podemos relacionar o sistema de coordenadas renormalizado (r, α) ao espaço $\mathbb{R} \times S^1$ munido da métrica induzida pelo pull-back da métrica de Fubini pela imersão de Veronese

$$\begin{array}{rccl} v: & \mathbb{R} \times S^1 & \to & \mathbb{P}^d \\ & (r, \alpha) & \mapsto & c_0 \left[1 : c_1 e^{r+i\alpha} : \cdots : c_d e^{dr+di\alpha}\right] \end{array}$$

onde c_i são constantes reais positivas. O fecho de $\mathbb{R} \times S^1$ é a *variedade tórica* associada aos polinômios a uma variável densos de grau d. O número de soluções é proporcional ao volume dessa variedade. Esse fato se generaliza a dimensão qualquer (Mais informações em [2]).

2.3 Perturbação ou Derivação

Como mencionei acima, um dos métodos para se recuperar também o módulo das raízes era perturbativo: aplicar o algoritmo em $f(x + \epsilon)$ e em $f(x - \epsilon)$. Em análise numérica, esse tipo de algoritmos costumam ser uma péssima idéia por conta dos arredondamentos.

É muito melhor utilizar cálculo diferencial. *Algoritmos numéricos podem ser derivados.*

Vamos considerar inicialmente uma *curva* de polinômios,

$$f(x - \epsilon) = (x - \zeta_1 - \epsilon) \cdots (x - \zeta_d - \epsilon) \ .$$

Só iremos derivar o algoritmo *uma* vez e para $\epsilon \simeq 0$. Por isso, escrevemos:

$$f(x - \epsilon) = f(x) + \epsilon \dot{f}(x) + O(\epsilon^2)$$

onde podemos calcular

$$\dot{f}(x) = -f'(x) \ .$$

Agora podemos aproximar cada iteração por uma função afim em ϵ:

$$\begin{array}{rcl} f(x) & \mapsto & Gf(x) = (-1)^d f(x) f(-x) \\ f(x) + \epsilon \dot{f}(x) & \mapsto & Gf(x) + \epsilon \left((-1)^d f(x) \dot{f}(-x) + \dot{f}(x) f(-x) \right) \end{array}$$

A aplicação $f + \epsilon \dot{f} \mapsto Gf + \epsilon DG_{|f} \dot{f}$ é chamada em Cálculo em Variedades de *aplicação tangente*.

Iterando a aplicação tangente, obtemos uma "linha" de polinômios $g(x) + \epsilon \dot{g}(x)$, de raízes

$$Z_j + \epsilon \dot{Z}_j = (\zeta_j + \epsilon)^{2^N} + O(\epsilon^2) = \zeta_j^{2^N} (1 + 2^N \epsilon \zeta_j^{-1}) + O(\epsilon^2) \ .$$

Assim,

$$\dot{Z}_j = 2^N Z_j \zeta_j^{-1}$$

e podemos recuperar

$$\zeta_j = 2^{-N} \frac{\dot{Z}_j}{Z_j} \ .$$

(Ainda é necessário reescrever essa fórmula em termos de coordenadas renormalizadas. Isso é feito no artigo [3]).

2.4 Conclusões

Um dos principais objetivos desta aula era convencer a audiência de que algoritmos numéricos são um objeto cujo estudo demanda ferramental matemático clássico:

1. Cálculo Diferencial: algoritmos podem *e devem* ser diferenciados.
2. Geometria Diferencial: algoritmos existem naturalmente em variedades.
3. Geometria Simplética (dessa falei pouco).

Uma outra lição importante: *A Matemática do século XIX está cheia de resultados brilhantes, muitos dos quais fora de moda.* Ao estudar problemas computacionais, nunca se deve desprezar a sabedoria de quem era obrigado a fazer as contas na mão.

2.5 Exercícios

Exercício 2.1. Defina uma variação da iterada de Gräffe, que eleve ao cubo cada solução de um polinômio f.

Exercício 2.2. Mostre que com probabilidade um, todo polinômio é "livre de círculos".

Exercício 2.3. Uma *reta tropical* de equação $a \boxed{\times} r \boxed{+} b \boxed{\times} s \boxed{+} c$ é o conjunto de $(\mathbb{R} \cup -\infty)^2$ dos pontos (r, s) onde o máximo de $a \boxed{\times} r$, $b \boxed{\times} s$ e c é atingido pelo menos duas vezes. Caracterize todas as retas tropicais.

Exercício 2.4. Mostre que duas retas tropicais têm sempre pelo menos um ponto de intersecção.

Exercício 2.5. Seja $f(x, y) = \sum_{k=1}^{S} f_k x^{\alpha_k} y^{\beta_k}$ um polinômio em duas variáveis, $f_k \neq 0$. Seja V o conjunto dos zeros de f em $\mathbb{C}^2$. A *ameba* $\mathcal{A}$ associada a f é o conjunto $\mathcal{A} = \{(-\log|x|, -\log|y|) : (x, y) \in V\}$. Mostre

que $\lim 2^{-k}\mathcal{A}$ está contido na *curva tropical* dos $\{(r,s) \in (\mathbb{R} \cup -\infty)^2$ tais que

$$\max_k r^{\boxed{\alpha_k}} \boxed{\times} s^{\boxed{\beta_k}}$$

é atingido pelo menos duas vezes.

Referências

[1] Alston S. Householder, *Dandelin, Lobačevskiĭ, or Graeffe?*, Amer. Math. Monthly **66** (1959), 464–466.

[2] Gregorio Malajovich and J. Maurice Rojas, *High probability analysis of the condition number of sparse polynomial systems*, Theoret. Comput. Sci. **315** (2004), no. 2-3, 524–555.

[3] Gregorio Malajovich and Jorge P. Zubelli, *Tangent Graeffe iteration*, Numer. Math. **89** (2001), no. 4, 749–782.

[4] Alexandre Ostrowski, *Recherches sur la méthode de Graeffe et les zéros des polynomes et des séries de Laurent*, Acta Math. **72** (1940), 99–155 (French).

[5] ———, *Recherches sur la méthode de Graeffe et les zéros des polynomes et des séries de Laurent. Chapitres III et IV*, Acta Math. **72** (1940), 157–257 (French).

Esta lista não é completa. Uma bibliografia mais extensa pode ser encontrada em [3].

Aula 3

A procura na *Web*

3.1 Introdução à teoria dos grafos

Talvez uma das aplicações mais importantes e menos entendidas da Álgebra Linear sejam os algoritmos de de busca na internet. Os conceitos fundamentais são o Teorema Espectral (Teorema 1.3) e a decomposição SVD (Teorema 1.4).

Definição 3.1. Um *grafo* simples é um par $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ onde $\mathcal{V}$ é um conjunto finito (seus elementos são chamados de *vértice* e $\mathcal{E}$ é um conjunto de pares não ordenados de vértices diferentes (chamados de *arestas*).

Um *caminho* é uma lista finita de vértices, tais que cada dois vértices consecutivos formam uma aresta. Também podemos representar um caminho pela lista de arestas correspondentes. Um *ciclo* é um caminho onde o último vértice é idêntico ao primeiro vértice.

Exemplo 3.2. A internet (Fig. 3.1) pode ser modelada por um conjunto (gigantesco mas finito) de computadores (vértices), cada um conectado a um número pequeno de outros computadores. Cada ligação é uma aresta. Um modelo mais realista associaria também a cada aresta, a sua velocidade ou largura de banda.

Exemplo 3.3. A malha rodoviária nacional também pode ser descrita como um conjunto de vértices (localidades), e as arestas correspondem às estradas diretas entre essas localidades.

Gregorio Malajovich, *Geometria de Algoritmos Numéricos*. Notas em Matemática Aplicada **37** (XXXI CNMAC), SBMAC, São Carlos, 2008.
Copyright © Gregorio Malajovich, 2008.

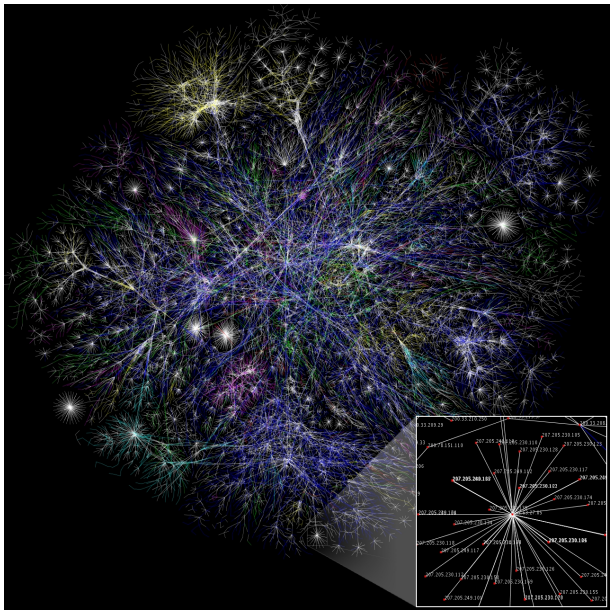

Figura 3.1: Mapa parcial da internet. Imagem publicada por Matt Britt, *http.wikimidia.org*, sob o título *Internet map 1024.jpg*. Copyright © Creative Commons Attribution 2.5 License.

Exemplo 3.4. Matemáticos costumam escrever artigos em parceria. O *grafo de colaboração* é o grafo cujos vértices correspondem a cada Matemático com artigos publicados, e as arestas à existência de uma colaboração publicada entre eles. A *distância de colaboração* entre dois Matemáticos é a distância entre eles no grafo, e pode ser calculada[1]. O *número de Erdös* de um Matemático é a distância de colaboração entre ele e Paul Erdös (1913-1996), que foi aparentemente o mais colaborador e prolífico dentre os grandes matemáticos do século passado. O *grau* de um vértice é o número de arestas contendo esse vértice. No grafo de colaboração, Paul Erdös tem grau 507.

Propriedades métricas e de conexidade de grafos são extremamente importantes. Em uma rede de comunicações, é importante que existam múltiplos caminhos entre dois pontos mas também é crucial que a distância entre dois pontos quaisquer seja pequena.

Isso é uma característica importante de redes de comunicações ou de redes sociais, conhecida como propriedade do "mundo pequeno".

A internet tem essa propriedade (vocês podem listar o caminho entre o seu computador e outro computador qualquer usando o comando `traceroute`. Uma distância de 30 é incomum. Entre Matemáticos, a distância de colaboração costuma ser bem menor (4 é razoável).

Uma maneira de estudar grafos é introduzir a matriz de adjacência.

Definição 3.5. A *matriz de Adjacência* $A_{\mathcal{G}}$ associada a um grafo simples $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ é a matriz de tamanho $\#\mathcal{V} \times \#\mathcal{V}$ definida por

$$(A_{\mathcal{G}})_{a,b} = \begin{cases} 1 & \text{Se } \{a,b\} \in \mathcal{E} \\ 0 & \text{Em todos os outros casos.} \end{cases}$$

3.2 A Equação do Calor em grafos

Para se estudar as propriedades de conexidade de grafos do mundo real (em geral com milhares ou milhões de vértices, talvez bilhões) é necessário recorrer a invariantes "estatísticos". Por exemplo, é possível estudar caminhos aleatórios em grafos. Como veremos a seguir, isso está relacionado com a equação do calor.

Seja $\mathcal{G}$ um grafo simples. A *matriz Laplaciana* associada a $\mathcal{G}$ é definida por:

$$\Delta_{\mathcal{G}} = A_{\mathcal{G}} - D_{\mathcal{G}}$$

[1] Ver em: ams.impa.br

onde a matriz $D_{\mathcal{G}}$ é diagonal, e $(D_{\mathcal{G}})_{vv} = \sum_w A_{vw}$ é o *grau* do vértice v (número de arestas incidentes). A transmissão do calor entre dois compartimentos é proporcional à diferença de temperatura. A equação diferencial do calor em uma barra de metal pode ser obtida discretizando o espaço e o tempo e passando ao limite: $\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2}$. Em uma placa ou barra de metal, a equação é $\frac{\partial u(\mathbf{x},t)}{\partial t} = \Delta_{\mathbf{x}} u(\mathbf{x},t)$.

A equação do calor em grafos é

$$\dot{\mathbf{u}}(t) = \Delta_{\mathcal{H}} \mathbf{u}(t) \ .$$

Essa equação modela um processo de difusão em grafos. Se $\mathcal{H}$ for conexo, $\lim_{t\to\infty} \mathbf{u}(t)$ existe, e é constante entre vértices conectados por caminhos. Quanto mais rápida (em geral) a convergência, mais bem-conexo é o grafo.

Podemos também considerar o análogo discreto:

$$\mathbf{u}(t+1) = I + \epsilon \Delta_{\mathcal{H}} \mathbf{u}(t) \ .$$

Note que para $\epsilon < 1/(max(D_{\mathcal{G}})_{vv})$, a matriz $I + \epsilon\Delta_{\mathcal{H}}$ é uma matriz de Márkov! De fato é uma matriz duplamente estocástica, e a distribuição estacionária é $\mathbf{u}^*_v = 1$.

A matriz $\Delta_{\mathcal{H}}$ é simétrica, e pode portanto ser diagonalizada. Seus autovalores são portanto números reais. Como $I + \epsilon\Delta_{\mathcal{H}}$ é estocástica, os autovalores de $\Delta_{\mathcal{H}}$ são menores ou iguais a zero. Sendo $\mathcal{H}$ conexo, o autovalor zero terá multiplicidade 1.

Definição 3.6. O *espectro* de um grafo $\mathcal{H}$ é a lista $0 = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\#\mathcal{V}}$ dos autovalores de $\Delta_{\mathcal{H}}$.

Pelo Teorema Espectral, a matriz $\Delta_{\mathcal{H}}$ admite uma base ortonormal de autovetores. Isso permite estimar a velocidade de convergência de $u(t)$ por:

$$\|u(t) - u^*\| \leq e^{\lambda_2 t} \|u(0)\| \ .$$

Quanto mais negativo for λ_2, mais robusta e eficiente é uma rede de comunicações.

3.3 As Leis de Kirchhoff

As Leis de Kirchhoff (Fig.3.2) permitem "resolver" circuitos elétricos com resistências conectadas de maneira arbitrária. Para isso, precisamos representar circuitos elétricos de alguma maneira. Poderíamos utilizar um grafo, mas precisamos ainda de mais informação:

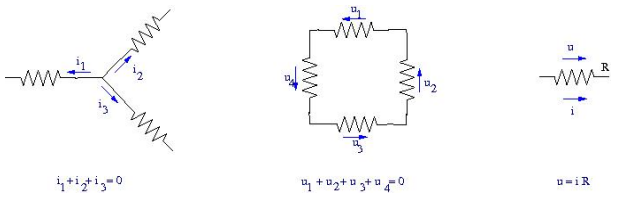

Figura 3.2: Leis de Kirchhoff e de Ohm.

1. Precisamos convencionar uma orientação para cada aresta.

2. Além disso, precisamos conhecer cada uma das resistências.

Um grafo simples onde se especifica uma orientação para cada aresta é chamado de *grafo orientado*. Agora, o conjunto de arestas é um subconjunto $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$. onde, se $(v, w) \in \mathcal{E}$, $(w, v) \notin \mathcal{E}$. Em particular, não existe aresta da forma (v, v).

Definição 3.7. A *matriz de Incidência* $I_{\mathcal{G}}$ associada a $\mathcal{G}$ é a matriz de tamanho $\#\mathcal{E} \times \#\mathcal{V}$, onde

$$(I_{\mathcal{G}})_{(a,b),c} = \begin{cases} 1 & \text{Se } b = c \\ -1 & \text{Se } c = a \\ 0 & \text{Em todos os outros casos.} \end{cases}$$

Seja $\mathcal{G}$ portanto o grafo orientado de uma malha elétrica, onde a cada aresta (a, b) associamos uma resistência $R_{(a,b)}$. Seja $\mathbf{R} \in \mathbb{R}^{\#\mathcal{E}} \times \mathbf{R} \in \mathbb{R}^{\#\mathcal{E}}$ a matriz diagonal das resistências, $\mathbf{i} \in \mathbb{R}^{\#\mathcal{E}}$ o vetor da corrente em cada aresta e $\mathbf{q} \in \mathbb{R}^{\#\mathcal{V}}$ o vetor de potencial elétrico.

Assumimos que o circuito está em equilíbrio.

Lei de Kirchhoff para a corrente: *A corrente elétrica entrando em um vértice é igual à corrente saindo.*

Do ponto de vista matricial,

$$I_{\mathcal{G}}^T \mathbf{i} = \mathbf{0} \ .$$

Lei de Kirchhoff para a voltagem *A soma de diferenças de potencial entre arestas correspondendo a um ciclo fechado é zero.*

O vetor das diferenças de potencial é $\mathbf{u} = I_{\mathcal{G}}\mathbf{q} \in \mathbb{R}^{\#\mathcal{E}}$.

Os caminhos fechados pertencem todos ao núcleo de $I_{\mathcal{G}}^T$. O que a Segunda Lei afirma é que $\mathbf{u}$ é ortogonal ao núcleo de $I_{\mathcal{G}}^T$. Isso segue do Teorema do Posto.

Agora aplicamos a **Lei de Ohm:** $u = iR$.

No nosso caso, temos a igualdade matricial $\mathbf{u} = R\mathbf{i}$. Podemos resolver o circuito:

$$I_{\mathcal{G}}^T R^{-1} I_{\mathcal{G}} \mathbf{q} = 0 \ .$$

A dimensão do espaço das soluções é $\dim \ker I_{\mathcal{G}}$, que é o número de componentes conexos do grafo. Isso é razoável se o nosso circuito está no equilíbrio.

Agora suponhamos que prescrevemos entrada ou saída de corrente em alguns dos vértices. Teremos agora:

$$I_{\mathcal{G}}^T R^{-1} I_{\mathcal{G}} \mathbf{q} = \mathbf{j} \ ,$$

onde $\mathbf{j}$ corresponde ao intercâmbio de corrente. A matriz $I_{\mathcal{G}}^T R^{-1} I_{\mathcal{G}}$ é simétrica e pode ser assimilada a um Laplaciano. A Lei de Ohm diz que a derivada da carga em um vértice é igual à soma das diferenças de potencial, ponderadas pela condutância (inversa da resistência). Nesse sentido, a equação acima corresponde também a um processo de difusão. A maneira correta de se definir a matriz de adjacência para um circuito de resistências é como a matriz das condutâncias:

$$(A_{\mathcal{G}})_{a,b} = \begin{cases} R_{a,b}^{-1} & \text{Se } (a,b) \text{ ou } (b,a) \in \mathcal{E} \\ 0 & \text{Em todos os outros casos.} \end{cases}$$

Nesse caso, definimos o grau de a como a soma das condutâncias das arestas incidindo em a. Recuperamos portanto que:

$$I_{\mathcal{G}}^T R^{-1} I_{\mathcal{G}} = A_{\mathcal{G}} - D_{\mathcal{G}} = \Delta_{\mathcal{G}} \ .$$

3.4 Digrafos e o *Google*

Definição 3.8. Um *digrafo simples* ou *directed simple graph*) $\mathcal{G}$ é um par $(\mathcal{V}, \mathcal{E})$ onde $\mathcal{V}$ é um conjunto finito (seus elementos são chamados de *vértices*) e $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. Os elementos de $\mathcal{E}$ são chamados de *arestas* . Arestas de um vértice nele mesmo são permitidas.

Exemplo 3.9. A *World Wide Web* é modelada por um conjunto de *páginas* (associadas a um endereço *http* ou *uniform ressource locator*) (vértices) e um conjunto de ligações orientadas (*links*) entre os vértices.

Exemplo 3.10. O cérebro humano é composto de mais de 100 bilhões de neurônios. Cada neurônio é uma célula com dois prolongamentos (áxil e dendrítico). Cada um desses prolongamentos se ramifica em possivelmente

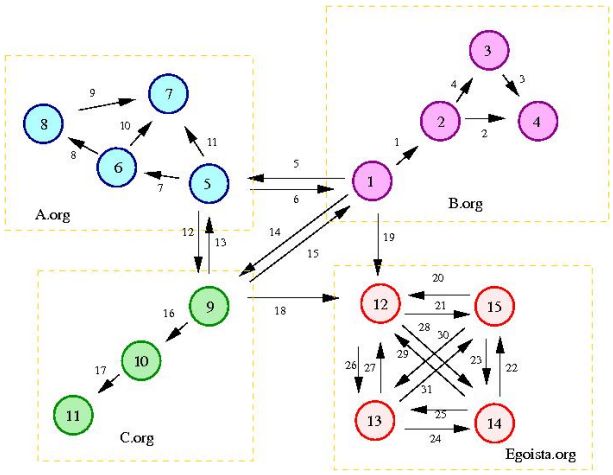

Figura 3.3: Domínio fictício de *Web*.

milhares ou dezenas de milhares de extremidades. Potencial elétrico no centro da célula (soma) é transmitido ao prolongamento áxil. Isso afeta o terminal dendrítico de outros neurônios em contato (sinapse), transmitindo assim a informação. A transmissão é unidirecional. Neurônios podem ser ativadores ou inibidores. De qualquer maneira, podemos modelar o fluxo de informação por um digrafo, onde os neurônios são os vértices e as sinapses são as arestas. Um neurônio pode ter mil ou dez mil arestas.

O invariante natural de um digrafo é a matriz de transferência, versão orientada da matriz de adjacência:

$$(T_{\mathcal{G}})_{a,b} = \begin{cases} 1 & \text{Se } (b,a) \in \mathcal{E} \\ 0 & \text{Em todos os outros casos.} \end{cases}$$

Essa matriz não é simétrica.

Vamos supor que um *programa rastejador* (que vamos chamar de *bot*) percorre um grafo orientado $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ escolhendo, a cada vértice, uma aresta aleatória. Para fixar as idéias, vamos considerar apenas o domínio da Figura 3.3. O *bot* se desloca de maneira aleatória, escolhendo arestas ao acaso.

Quatro domínios disputam a atenção do *bot*, e todos têm uma *bot trap* : uma vez que o bot entrou em uma página, ele não consegue mais sair do sub-domínio.

Se o bot chega em um vértice sem saída, ele pula para um vértice aleatório. Ainda, poderia ficar preso em um ciclo.

Uma solução possível é a seguinte: a cada passo, o bot tem uma probabilidade δ de pular para uma página escolhida de maneira totalmente aleatória.

Larry Page, Sergey Brin e coautores[3] , então estudantes em Stanford, modelaram a relevância de uma página como o tempo médio que um desses bots ficaria nessa página. Obviamente é impossível simular isso com um trilhão de bots. Mas o algoritmo que Page desenvolveu, em conjunto com Serguei Brin, permite estimar esse tempo de maneira conveniente.

Para isso, ele modelou o passeio aleatório do bot por um processo de Márkov. Dado um vetor de probabilidade $\mathbf{p}_t$ (que mede a probabilidade do bot se encontrar em cada página, no tempo t), pode-se escrever o vetor $\mathbf{p}_{t+1}$ como $\mathbf{p}_{t+1} = M\mathbf{p}_t$, onde:

$$M = \frac{\delta}{N} \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \begin{bmatrix} 1 & \dots & 1 \end{bmatrix} + (1-\delta) T D^{-1} \ ,$$

N é o número de vértices e D é a matriz diagonal contendo o número de arestas saindo de cada vértice.

A Matriz M é uma matriz de Márkov, e portanto $(\mathbf{p}_t)$ converge para um estado estacionário $\mathbf{p}^*$. O índice de relevância da página v é p^*_v.

Os algoritmos que Page e Brin utilizaram se mostraram muito mais eficientes do que os disponíveis para a concorrência (que funcionava como as páginas amarelas, cobrando dos anunciantes). *Google* , o sistema de busca deles, ganhou imediatamente o favor dos usuários da internet.

Vamos digitar a matriz de transferência do domínio fictício da Fig. 3.3:

```
T = zeros( 15, 15) ;

T( 1, 2) = 1; T( 2, 4) = 1; T( 3, 4) = 1; T( 2, 3) = 1; T( 1, 5) = 1;
T( 5, 1) = 1; T( 5, 6) = 1; T( 6, 8) = 1; T( 8, 7) = 1; T( 6, 7) = 1;
T( 5, 7) = 1; T( 5, 9) = 1; T( 9, 5) = 1; T( 1, 9) = 1; T( 9, 1) = 1;
T( 9,10) = 1; T(10,11) = 1; T( 9,12) = 1; T( 1,12) = 1; T(15,12) = 1;
T(12,15) = 1; T(14,15) = 1; T(15,14) = 1; T(13,14) = 1; T(14,13) = 1;
T(12,13) = 1; T(13,12) = 1; T(12,14) = 1; T(14,12) = 1; T(15,13) = 1;
T(13,15) = 1;
T = T'
```

Agora produzimos a matriz M, e iteramos.

```
delta = 0.15 ;
D = sum(T) ;

for j=1:15
```

```
  if (D(j) == 0) T(:,j) = ones(15,1) ;
  end ;
end ;
D = sum(T) ;

M = delta * ones(15,1) * ones(1,15) / 15 + (
(1-delta) * T * diag(D.^(-1))) ;

p = ones(15,1)/15 ;
for k=1:100
  p = M * p ;
  end ;
p'
eps = norm(p - M*p)
```

Obtemos:

```
octave:58> p'
ans =

Columns 1 through 8:

0.033144 0.026101 0.030151 0.055779 0.033144 0.026101 0.062822 0.030151

Columns 9 through 15:

0.033144 0.026101 0.041244 0.158762 0.147785 0.147785 0.147785

octave:59> eps = norm(p - M*p)
eps =  2.2153e-16
```

Note que *Egoísta.org* foi quem apresentou os melhores índices de relevância ! O domínio *Egoísta.org* montou uma *fazenda de links* , que aumenta o número de arestas apontando para cada uma das suas páginas.

Uma maneira de evitar essas manipulações é escolher a priori um número pequeno de páginas "confiáveis" e só permitir o salto para essas páginas. Outra é utilizar mais matemática.

O algoritmo atualmente utilizado pelo Google não é público. O autor destas linhas está convencido de que se trata de uma variante do algoritmos abaixo.

3.5 Busca na rede e a svd

A matriz de transferência T foi definida assim: $T_{vu} = 1$ se existe uma aresta orientada (u, v), senão $T_{vu} = 0$. Jon M. Kleinberg [2] observou o seguinte: $(T^T T)_{u_1 u_2}$ conta o número de vezes que u_1 e u_2 apontam para a mesma página. Isso mede quanto u_1 e u_2 concordam enquanto fontes de referências.

Vamos supor que um engenho de busca atribui peso a_u para a página u enquanto fonte de referência. Quão bom é o vetor de pesos $\mathbf{a}$? Do ponto de vista da página u_1, uma boa medida é $(T^T T\mathbf{a})_{u_1}$. Uma medida de quão consensual é o vetor $\mathbf{a}$ é a norma $\|T^T T\mathbf{a}\|$. Kleinberg sugere utilizar o autovetor principal de $T^T T$, que maximiza $\|T^T T\mathbf{a}\|$, como peso para as páginas *enquanto fonte de referência*.

Já $(TT^T)_{v_2 v_1}$ conta o número de referências que apontam simultaneamente para v_1 e v_2. Kleinberg também propões utilizar o autovetor principal $\mathbf{b}$ de TT^T como peso para as páginas *enquanto conteúdo*. (Ver exercício 3.3 para verificar que podemos escolher $\mathbf{b}$ de tal maneira que $b_u \geq 0$).

Esse algoritmo pode ser interpretado em termos da decomposição em valores singulares (Teorema 1.4). Os vetores $\mathbf{a}$ e $\mathbf{b}$ podem ser escolhidos de tal maneira que $\|a\| = \|b\| = 1$. Nesse caso, eles são o vetor singular principal à direita (resp. vetor singular principal à esquerda) de T, e são relacionados por

$$\sigma_1 \mathbf{b} = T\mathbf{a}$$

onde $\sigma_1 = \|T\|_2$ é o valor singular principal.

Os índices de Kleinberg são manipuláveis. Uma página pode obter um alto índice de relevância enquanto fonte apontando para toda a internet.

Para evitar esse tipo de manipulação, podemos substituir a matriz de transferência T pela matriz estocástica M.

Vamos voltar ao nosso exemplo. Uma maneira pouco eficiente de se calcular os vetores singulares é:

```
octave:28> [u,sigma,v]=svd(M)
octave:29> p=u(:,1); q=v(:,1); p=p/sum(p); q=q/sum(q) ; [p,q]
ans =

  0.040813  0.041361
  0.027973  0.072657
  0.050063  0.102729
  0.137383  0.054022
  0.036762  0.060426
  0.032024  0.093969
```

```
0.190207  0.054022
0.059120  0.139113
0.040813  0.041361
0.027973  0.051422
0.062892  0.054022
0.087595  0.055486
0.068793  0.059803
0.068793  0.059803
0.068793  0.059803
```

Note que a "fazenda de links" da Egoísta.com perdeu a liderança nas buscas !

O algoritmo mais eficiente para se achar autovetores principais (ou vetores singulares principais) é a iteração: Escolher $\mathbf{p}_1$ ao acaso, depois iterar

$$\mathbf{p}_{i+1} = \frac{(M^T M)\mathbf{p}_i}{\|(M^T M)\mathbf{p}_i\|} \ .$$

A velocidade de convergência é estimada facilmente utilizando o fato de que $M^T M$ é simétrica, e portanto (Teorema 1.3) admite uma base **ortonormal** de autovetores.

Seja $\mathbf{q}$ o autovetor principal de $M^T M$. Vamos escrever:

$$\mathbf{p}_i = x_i \mathbf{q} + \mathbf{r}_i,$$

com $\mathbf{r}_i \perp \mathbf{q}$. Então, a velocidade de convergência pode ser estimada por:

$$\frac{\|r_{i+1}\|}{x_i + 1} \leq \frac{\lambda_2}{\lambda_1} \frac{\|r_i\|}{x_i}$$

onde $\lambda_1 \geq \lambda_2 \geq \ldots$ são os autovalores de $M^T M$ (e $\lambda_j = \sigma_j^2$).

A indústria de engenhos de busca na internet é altamente competitiva, e precisa se atualizar constantemente para combater *Web spam* , práticas desonestas para obter (e vender) mais visibilidade. O combate ao *Web spam* pode exigir intervenção humana acoplada aos algoritmos [1] , ou reprogramação dos *programas rastejadores* que podem ser instruídos a não freqüentar certos domínios.

A maioria dos algoritmos são segredos industriais ou foram patenteados [2].

Este capítulo foi escrito com base na informação disponível publicamente, mas omiti aspectos computacionais importantes.

[2] Algoritmos são objetos matemáticos e portanto não são patenteáveis. No entanto, o departamento de patentes de alguns países aceita objetos matemáticos como parte de um processo industrial.

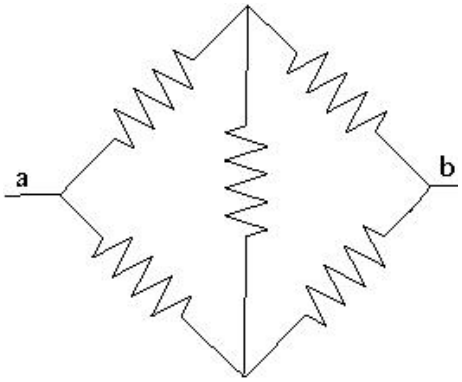

Figura 3.4: Ponte de resistências.

3.6 Conclusões

Alguns problemas em grafos *grandes* podem ser modelados utilizando idéias de processos de difusão, que levam a uma matemática similar à da equação do calor. A partir desse momento, os modelos podem ser resolvidos utilizando idéias de Álgebra Linear.

Uma das idéias principais é utilizar, sempre que possível, bases ortonormais. No exemplo acima, só precisamos calcular um dos vetores da base ortonormal (o autovetor principal).

Transformações ortogonais (ou seja, matrizes cujas colunas são ortonormais) têm número de condicionamento 1. Isso implica que mudanças de coordenadas ortonormais são numericamente estáveis, e que os autovetores de $M^T M$, no nosso exemplo, não vão ser muito afetados por mudanças pequenas na matriz M ou por erros de arredondamento.

Trabalhar com matrizes *grandes* é difícil por outra razão, além de eventual instabilidade numérica: as matrizes não entram na memória de um único computador. É preciso Matrizes do tamanho da *World Wide Web* exigem algoritmos distribuídos computacionalmente eficientes, que exploram a estrutura do grafo.

3.7 Exercícios

Exercício 3.1. A figura 3.4 mostra uma "ponte de resistências". Assuma que o valor de cada resistência é de 1Ω. Sabemos que passa uma corrente de $1A$ entre os pontos a e b. Qual é a diferença de tensão ?

Exercício 3.2. O *grafo perfeito* K_d de ordem d é o grafo com d vértices, conectados todos com todos. Quais são os autovalores de Δ_{K_d} ?

Exercício 3.3. Seja T a matriz de transferência de um grafo orientado, e assuma que de todo vértice sai ao menos uma aresta. Mostre que o autovetor principal de $T^T T$ pode ser escolhido com coordenadas positivas. Dica: mostre que o ortante positivo é mapeado nele mesmo pela iteração $\mathbf{x} \mapsto T^T T \mathbf{x}$.

Exercício 3.4. Considere agora o grafo com d vértices 1 a d, onde j e $j+1$ estão conectados e d está conectado a 1. Calcule numericamente os autovalores de seu Laplaciano.

Exercício 3.5. Considere a seguinte variante para o modelo de Kleinman: substitua a matriz T, não pela matriz estocástica M mas pela matriz $\tilde{M}$ obtida normalizando as colunas da matriz estocástica. Dessa maneira, interprete a recorrência $\mathbf{p}_{i+1} = \tilde{M}^T \tilde{M} \mathbf{p}_i$ como a versão discreta do processo continuo

$$\frac{\partial}{\partial t}\mathbf{p}(t) = (\tilde{M}^T \tilde{M} - I)\mathbf{p}(t) \ .$$

Mostre que essa variante daria peso igual para cada uma das páginas. Dica: interprete esse processo como a equação do calor em um certo grafo.

Referências

[1] Zoltán Gyöngi, Hector Garcia-Molina, and Jan Pedersen, *Combating Web Spam with TrustRank*, Proceedings of the International Conference on Very Large Data Bases **30**, 576. `http://www.vldb.org/conf/2004/RS15P3.PDF`.

[2] Jon Michael Kleinberg, *US Patent 6112202: Method and system for identifying authoritative information resources in an environment with content-based links between information resources.*, 1997, 2000.

[3] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd, *The PageRank citation ranking: Bringing order to the Web*, Preprint, Stanford University, 1999. `http://dbpubs.stanford.edu:8090/pub/1999-66`.

Aula 4

A compressão do som, o mp4 e a televisão digital

4.1 Sinais sonoros

O som que ouvimos corresponde a variações de pressão nos nossos tímpanos, que são capturadas e transformadas em um sinal nervoso no nosso ouvido interno.

Podemos capturar um sinal sonoro com um microfone, e o sinal elétrico analógico pode ser gravado, transmitido, reproduzido e retransformado em sinal sonoro.

Hoje em dia preferimos armazenar som sob forma digital. Para isso, o sinal elétrico produzido por um microfone pode ser digitalizado por um conversor analógico-digital. O circuito de áudio dos computadores modernos tem um conversor embutido.

Se o computador estiver rodando GNU-linux ou similar, existe um "dispositivo de áudio" `/dev/dsp` que funciona como um arquivo (pode ser lido e escrito) e permite capturar o sinal do microfone. Mas vamos fazer o experimento no *Octave:*

```
f = record (1) ;
plot(f) ;
playaudio(f)
```

Gregorio Malajovich, *Geometria de Algoritmos Numéricos*. Notas em Matemática Aplicada **37** (XXXI CNMAC), SBMAC, São Carlos, 2008.

Copyright © Gregorio Malajovich, 2008.

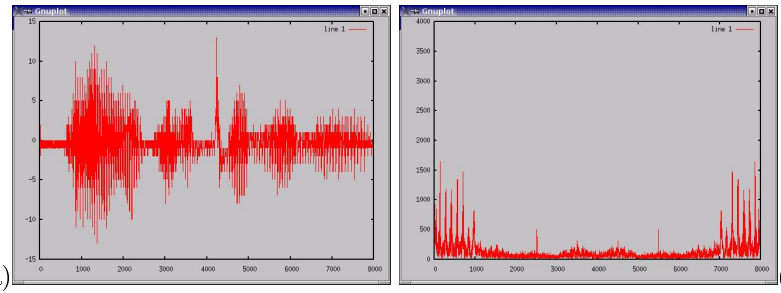

Figura 4.1: Sinal sonoro (a) original, (b) transformada de Fourier, (c) transformada de Fourier comprimida, (d) comprimido.

O primeiro comando captura o sinal do microfone durante 1 segundo, o segundo mostra o gráfico do sinal (Figura 4.1a) e o terceiro toca o sinal de volta. O sinal foi armazenado em um vetor de $\mathbb{R}^{8000}$ (São 8000 leituras do sinal por segundo). O gráfico está desenhado na figura 4.1a.

Para gravar música em qualidade de CD, precisamos de 44,100 leituras por segundo, vezes dois canais. Para armazenar uma hora de música, precisaríamos armazenar um vetor de $\mathbb{R}^{318\times10^6}$. Guardando dois bytes por coordenada, precisaremos de aproximadamente 600 MB, que é o tamanho dos atuais CDs.

Um método de compressão possível (utilizado em redes de telefonia) é aplicar uma escala logarítmica à intensidade do sinal, e armazenar utilizando menos *bits* (por exemplo, 8). Por exemplo, de x é um sinal sonoro, a codificação de x pela chamada $\mu - law$ ou $u - law$ é definida por:

$$y_i = \frac{\text{sinal}(|x_i|)}{\ln(1+\mu)} \ln\left(1 + \mu \frac{|x_i|}{\max(|x_j|)}\right)$$

onde $\mu = 255$.

Esse método é (apenas) razoável para comunicações telefônicas.

4.2 A transformada de Fourier

Vamos denotar por $L^2([0,1])$ o espaço de todas as funções a valores complexos, definidas para $t \in [0,1]$, integráveis e com quadrado integrável. Se temos um sinal (função real) f definido no intervalo de tempo $[0,1]$, podemos considerar ele como um elemento de $L^2([0,1])$. O *produto interno* de duas funções f e g em $L^2([0,1])$ é

$$\langle f, g\rangle = \int_0^1 \bar{f}(t)g(t)\ dt.$$

Note que esse produto está bem definido em $L^2([0,1])$.

A *transformada de Fourier* de f é definida por:

$$\hat{f}(s) = \int_0^1 f(t)e^{-2\pi ist}\ dt$$

para $s \in \mathbb{Z}$. Uma definição equivalente é

$$\hat{f}(s) = \langle t \mapsto e^{-2\pi ist}, t \mapsto f(t)\rangle\ .$$

A transformada de Fourier pode ser interpretada como uma aplicação linear de $L^2([0,1])$ no espaço $l^2(\mathbb{Z})$, que é o espaço de todas as biseqüências $(\hat{f}_s)_{s\in\mathbb{Z}}$ a valores complexos com "norma Euclidiana" $\|\hat{f}\| = \sqrt{\sum_{s\in\mathbb{Z}} |\hat{f}_s|^2}$ finita. Esse espaço admite o produto interno

$$\langle \hat{f}_s, \hat{g}_s\rangle = \sum_{s\in\mathbb{Z}} \hat{\bar{f}}\hat{g}_s\ .$$

Prova-se ainda que vale a seguinte fórmula de reconstrução:

$$f(t) = \sum_{s\in\mathbb{Z}} e^{2\pi its}\hat{f}(s)\ .$$

Nesse sentido, diz-se que o conjunto das funções $t \mapsto e^{2\pi its}$ é uma *base* do espaço $L^2([0,1])$.

Observação 4.1. A definição usual de combinação linear, nos textos de Álgebra, costuma ser a da combinação linear finita. Em análise de Fourier ou processamento de sinais, assume-se que uma combinação linear é qualquer soma ou integral, com norma Euclidiana dos coeficientes (ou integral do valor absoluto do módulo da função-coeficiente) limitada.

Note também que

$$\|t \mapsto e^{2\pi its}\|^2 = \int_0^1 1 \; dt = 1$$

e que, para $s_1 \neq s_2$ inteiros,

$$\langle t \mapsto e^{2\pi its_1}, t \mapsto e^{2\pi its_2} \rangle = \int_0^1 e^{2\pi it(s_2 - s_1)} \; dt = 0.$$

Assim, $(t \mapsto e^{2\pi its})_{s \in \mathbb{Z}}$ é uma *base ortonormal* do espaço $L^2([0,1])$.

Observação 4.2. Uma maneira clássica de resolver a equação do calor e a equação da onde é escrever o operador $f(x) \mapsto \frac{\partial^2}{\partial x^2} f(x)$ nessa base ortonormal. Obtemos o operador

$$\hat{f}_s \mapsto -4\pi^2 s^2 \hat{f}_s \; .$$

Note que esse operador tem norma infinita, e que ele é infinitamente mal condicionado. (Em geral, a derivação é mal condicionada e por isso algoritmos de diferenciação numérica são sempre problemáticos).

Vamos utilizar a base de Fourier como primeira aproximação para processamento de sinais. O comando `fft` do *Octave* aproxima a transformada de Fourier. Se temos um vetor $\mathbf{f}$ de $\mathbb{R}^N$ ou $\mathbb{C}^N$, o comando produz um vetor $\hat{\mathbf{f}} = \mathcal{F}(\mathbf{f})$ de $\mathbb{C}^N$ onde

$$\begin{array}{rccl} \mathcal{F}: & \mathbb{C}^N & \to & \mathbb{C}^N \\ & \mathbf{f} & \mapsto & \hat{\mathbf{f}} = \mathcal{F}(\mathbf{f}) \text{ onde } \hat{f}_j = \sum_{k=0}^{N-1} f_k e^{-2\pi ijk/N} \end{array} .$$

A transformação $\mathcal{F}$ é conhecida como *transformada de Fourier discreta* , ou *DFT*.

Se o vetor $\mathbf{f}$ é discretização de um sinal de duração T, a j-ésima coordenada de $\hat{f}_j$ corresponde à freqüência $\frac{\min(j, N-j)}{T}$. (Ver exercício 4.6).

O comando `ifft` calcula a transformada discreta inversa de Fourier $\mathcal{F}^{-1}$. Podemos gerar a nota Lá a 440 Hz fazendo:

```
y=zeros(8000,1);            % 1s corresponde a 8000 leituras.
y(440-1)=1e+6;
playaudio(real(ifft(y))) ;
```

(Compare com o sinal telefônico). Do ponto de vista matemático, essa transformada corresponde a uma *projeção ortogonal* do sinal original.

Agora vamos utilizar isso para "comprimir" o sinal de voz que havíamos gravado. O ouvido humano consegue perceber freqüências entre 20Hz e

20kHz. Se gravamos uma mensagem de 1 segundo com 8000 leituras, então perdemos totalmente as freqüências mais altas (Acima de 4kHz). Isso é perceptível na qualidade do sinal, mas não prejudica a sua compreensão. Vamos olhar para o espectro das freqüências do sinal:

```
Ff = fft(f) ;
plot(abs(Ff)) ;
```

Vemos na figura 4.1b que o sinal parece concentrado em umas poucas freqüências.

As seguintes linhas mostram um processo rudimentar de compressão. Vamos primeiro ordenar as coordenadas de $\hat{\mathbf{f}} =$ `Ff` por valor absoluto decrescente. O índice `I` armazena a ordem das coordenadas. Vamos depois medir quanto do sinal está concentrado nas 25% das freqüências com maior amplitude.

```
[ES,I]=sort(abs(Ff),'descend');
plot(ES) ;
norm(ES(1:2000))/norm(ES)

ans = 0.95853
```

Vemos que, se zeramos as 75% das freqüências restantes, perdemos "menos de 5% do sinal". Vamos fazer isso:

```
Ff(I(2001:8000))=zeros(6000,1);
plot(abs(Ff))
g=real(ifft(Ff));
plot(g)
playaudio (g)
```

(Veja a figura 4.1c–d). O sinal comprimido `g` é quase indistinguível do original `f`. Ao armazenar apenas um quarto das amplitudes, podemos obter uma compressão significativa. Ao executar o programa acima, vocês estão *ouvindo* o efeito de uma projeção ortogonal.

Aplicando esse procedimento a pequenos intervalos de sinais mais longos, é possível também equalizar, eliminar freqüências indesejadas ou "remasterizar" gravações antigas.

Uma maneira de "cortar" um sinal $f(t)$ em pedaços de tamanho T é substituir, para cada k, a função $f(t)$ por

$$f_k(\tau) = \begin{cases} 0 & \text{Se } \tau < Tk/2 \\ f(t)\text{sen}^2(\frac{\pi}{T}(\tau - Tk/2)) & \text{Se } Tk/2 \leq \tau \leq T(k/2+1) \\ 0 & \text{Se } \tau > T(k/2+1). \end{cases}$$

Teremos então $f(t) = \sum_k f_k(t + Tk/2)$. Se depois aplicamos a Transformada de Fourier Discreta em cada f_k, obtemos a chamada *transformada de Fourier discreta de curto prazo* ou *STDFT*.

Um procedimento de remasterização usual é calcular a *STDFT* e retirar as freqüências que aparecem em menor intensidade (exatamente como fizemos no exemplo do sinal sonoro).

Outra transformada usual é a transformada do cosseno, que pode substituir a transformada discreta de Fourier (Ver exercício 4.4). Para um sinal de tamanho T, ela é definida por:

$$F_k = \sum_{t=0}^{T-1} f_t \cos\left(\frac{\pi k}{T}\left(t + \frac{1}{2}\right)\right) \ .$$

4.3 A base de Haar

A transformada de Fourier não é a única maneira razoável de se representar um sinal. Uma das grandes desvantagens da transformada de Fourier é que ela representa "mal" transientes ou "picos" de um sinal. Isso quer dizer que um transiente, transformado pela transformada de Fourier, não pode facilmente ser comprimido.

Mesmo quando se utiliza a transformada de Fourier de curto prazo, o tamanho da "janela" é fixo, e transientes com duração muito inferior a esse tamanhos serão mal representados. A transformada de Haar é uma maneira razoável de representar sinais com transientes.

A *função de Haar* é definida por:

$$\begin{array}{rccl} H: & \mathbb{R} & \to & \mathbb{R} \\ & x & \mapsto & H(x) = \left\{ \begin{array}{ll} 0 & \text{Se } x < 0 \\ 1 & \text{Se } 0 \leq x < 1/2 \\ -1 & \text{Se } 1/2 \leq x < 1 \\ 0 & \text{Se } x > 1. \end{array} \right. \end{array}$$

A *base* de Haar que utilizaremos para representar funções em $L^2([0,1])$ será composta de dilações e translações de H. Definimos

$$H_{m,n}(t) = 2^{-m/2} H(2^{-m}t - n)$$

onde $-m \in \mathbb{N}$ e $n = 0, \ldots, 2^{-m} - 1$. O fator multiplicativo $2^{-m/2}$ garante que o conjunto das $H_{m,n}$ seja ortonormal. Para geral o espaço $L^2([0,1])$ inteiro, definimos ainda $H_{0,0}(t) \equiv 1$. Mostra-se que os H_{mn} formam uma base ortonormal de $L^2([0,1])$.

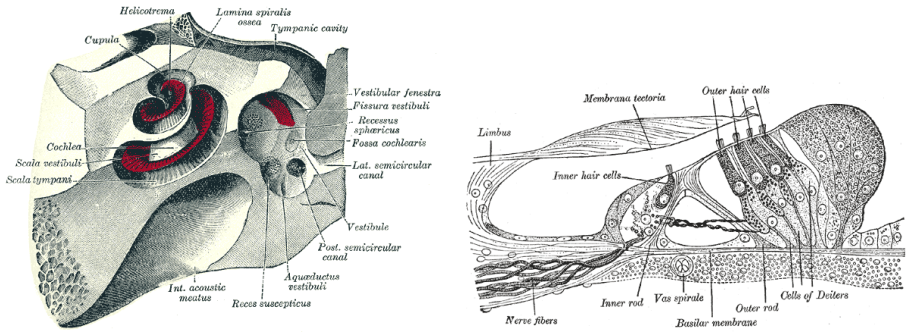

Figura 4.2: Cóclea e órgão de Corti. [4]

A transformada de Haar ou transformada de Wavelets de uma função real f é definida por:

$$T_{\mathrm{H}aar} f(m,n) = \int_0^1 H_{m,n}(t) f(t)\ dt.$$

No caso de uma mensagem discreta, é conveniente assumir que o número de leituras é potência de dois. Por exemplo, se temos 8 leituras, a mensagem é um vetor de $\mathbb{R}^8$ e a base de Haar é dada pelas colunas da matriz

$$H_3 = \begin{bmatrix} \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{1}{2} & 0 & \frac{\sqrt{2}}{2} & 0 & 0 & 0 \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & \frac{1}{2} & 0 & -\frac{\sqrt{2}}{2} & 0 & 0 & 0 \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{1}{2} & 0 & 0 & \frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{4} & \frac{\sqrt{2}}{4} & -\frac{1}{2} & 0 & 0 & -\frac{\sqrt{2}}{2} & 0 & 0 \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & \frac{1}{2} & 0 & 0 & \frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & \frac{1}{2} & 0 & 0 & -\frac{\sqrt{2}}{2} & 0 \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & -\frac{1}{2} & 0 & 0 & 0 & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{4} & -\frac{\sqrt{2}}{4} & 0 & -\frac{1}{2} & 0 & 0 & 0 & -\frac{\sqrt{2}}{2} \end{bmatrix}.$$

As matrizes H_n (ou *matrizes de Haar* assim construídas são ortogonais.

4.4 O ouvido humano e a transformada de Wavelets.

A cóclea é o órgão responsável pela audição humana. O sinal sonoro é amplificado mecanicamente no tímpano, e se propaga em meio líquido no interior da cóclea (Ver figura 4.2).

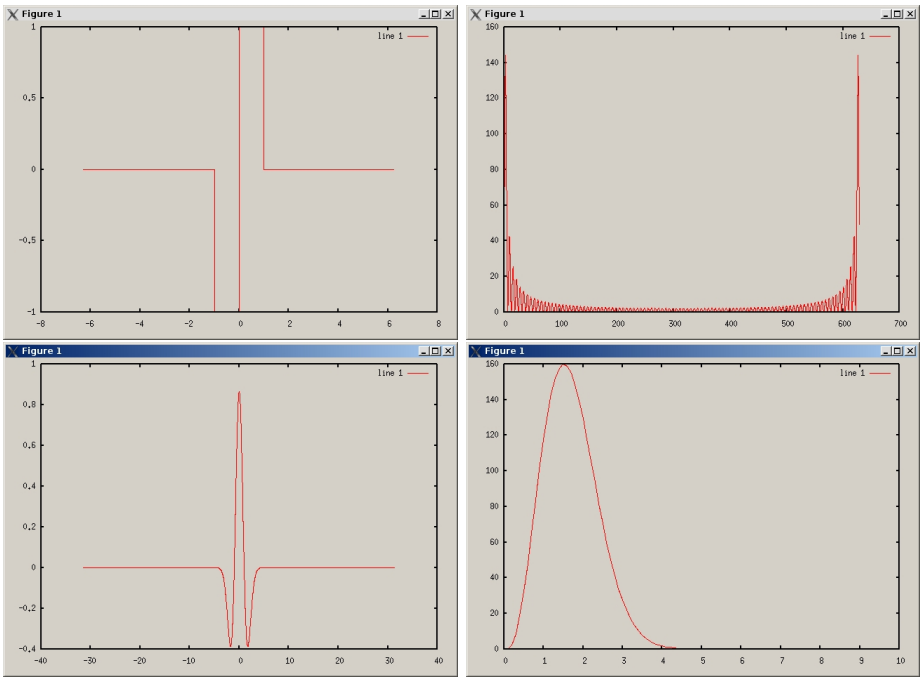

Figura 4.3: Acima: função de Haar e sua transformada de Fourier. Embaixo: função chapéu Mexicano e detalhe da sua transformada de Fourier

O órgão de Corti, dentro da cóclea, está recoberto de células ciliares, que fazem a transição para o nervo auditivo. Há dois tipos de células ciliares.

As curtas, ou estereocílios, ao se deslocar por força do sinal sonoro, abrem canais iônicos pelos quais estimulam (ou desestimulam) os terminais do nervo auditivo. Por conta desse mecanismo, o sinal nervoso gerado por cada estereocílio é localizado no tempo.

As células ciliares longas ou quinecílios entram em ressonância com o sinal sonoro, amplificando-o. A freqüência de vibração dos quinecílios é afetada por sinais nervosos. Assim, o ouvido "sintoniza" as principais freqüências recebidas em determinado momento.

Cada seção da cóclea corresponde a uma certa faixa de freqüências. Dessa forma, a cóclea já "decompõe" o sinal em freqüências, e os quinecílios fazem a sintonia fina.

As *bases de Wavelets* tentam imitar o processo acima. A compressão por transformada de Wavelets trunca exatamente a parte do sinal sonoro

que nós não ouvimos.

Um exemplo de Wavelets é gerada pela função do chapéu Mexicano,

$$\psi(x) = \frac{2\sqrt{3}}{3}\pi^{-1/4}(1-x^2)e^{-x^2/2} \ .$$

(Ver figura 4.3). A base de Wavelets é gerada da mesma maneira do que a base de Haar:

$$\psi_{m,n}(t) = 2^{-m/2}\psi(2^{-m}t - n) \ .$$

Existe hoje uma quantidade absurda de bases de Wavelets disponíveis. Em processamento de sinais, utiliza-se inclusive "pacotes" de Wavelets, que são conjuntos geradores do espaço das funções (os vetores da "base" não precisam ser linearmente independentes).

4.5 O padrão MP3 e os CODECs

O padrão *MP3* ou *MPEG-1 Audio Layer 3* está baseado em um modelo psicoacústico. Vimos na seção anterior que os quinecílios no órgão de Corti "sintonizam" as principais freqüências do sinal sonoro.

Se o ouvido está sintonizado em uma certa gama de freqüências (por exemplo, as utilizadas em uma composição musical), sinais relativamente fracos em freqüências vizinhas não encontram ressonância, pois os quinecílios correspondentes estão nas freqüências principais. Por isso, deixamos de perceber essa parte do sinal.

A compressão MP3 utiliza esse fato. O sinal é primeiro dividido em pequenos intervalos de tempo, cada um correspondendo a 1152 leituras para cada canal de áudio.

A primeira etapa da codificação é feita pelo chamado *filtro de quadratura polifase* . O som em cada intervalo é dividido por freqüências em 32 bandas.

Simultaneamente, o som sofre uma transformada de Fourier discreta e é analisado pelo modelo psicoacústico. Esse modelo permite eliminar as freqüências não audíveis (escondidas por sinais de maior intensidade em freqüências na mesma banda) e realçar sinais que exigem maior resolução no domínio do tempo.

Mais uma transformada de Fourier (de fato, transformada do cosseno) é aplicada a cada uma das 32 bandas, dividindo cada banda em 18 freqüências. Os parâmetros dessa última transformada são fornecidos pelo modelo psicoacústico.

Depois disso, o sinal é discretizado (ainda sob controle do modelo psicoacústico) e os valores discretos são comprimidos pelo código de Huffman (Ver exercícios 4.7–4.9).

Os detalhes da codificação são extremamente complicados, e não são definidos pelo padrão MP3. O que o padrão define é a decodificação.

Mais detalhes podem ser encontrados em [2]. Em particular, sugiro o artigo de Rassol Raissi, *The theory behind MP3*, Dezembro de 2002.

Existe uma quantidade enorme de padrões de áudio e vídeo disponíveis. Como é impossível escrever um programa capaz de ler todos os padrões existentes, os codificadores/decodificadores (os *codec* infames) podem ser distribuídos a parte. Por exemplo, MP3 e Vorbis são padrões definidos por *codec*.

O padrão MP3 é hoje um padrão industrial dominante. Tem a desvantagem de ser protegido por patentes.

O padrão Vorbis tem a vantagem de ser software livre. O algoritmo é aparentemente mais simples, e inclui uma transformada discreta do cosseno e a codificação em separado do "piso" e do resíduo. O piso é uma função linear por partes que aproxima o espectro do sinal em um dado intervalo de tempo. O resíduo é a diferença entre o espectro e o piso. Resíduo e espectro são depois truncados e armazenados via código de Huffman (Ex. 4.7–4.9).

4.6 Compressão de imagem e de vídeo

A compressão de imagens e de vídeos é potencialmente muito mais complicada do que a compressão de áudio.

Por exemplo, a compressão no padrão *JPEG* segue os seguintes passos: em primeiro lugar, as cores são codificadas em um sistema conhecido como YCbCr (usado por exemplo no padrão PAL-M de televisão). O olho humano é mais sensível ao brilho Y do que à cor (representada por Cr e Cb). Assim, a cor é truncada. A partir deste momento, cada sinal (Y, Cr e Cb) é tratado separadamente.

Cada um dos canais Y, Cr e Cb é dividido em blocos de 8×8 píxeis. Cada bloco é então objeto de uma transformada discreta do cosseno bidimensional.

O resultado é uma descrição de cada bloco por 8×8 "bifreqüências". Nossos olhos são mais sensíveis às baixas do que às altas "freqüências". Por isso, a precisão utilizada para as baixas freqüências é muito maior do que a das altas freqüências, que são truncadas.

Depois disso, o sinal é discretizado e codificado com código de Huffman.

4.7 A televisão digital.

O padrão *MP4* (aliás MPEG-4 Parte 14) é na verdade um meta-padrão, com vários *codec* de áudio e vídeo disponíveis. Por exemplo, o *H.264* (também conhecido como *MPEG*-4 Parte 10, ou ainda *AVC*) é o *codec* mais popular para a compressão de vídeo, e aparentemente será o padrão de alta resolução utilizado pela TV digital Brasileira[1].

No padrão *H.264*, cada quadro de vídeo (imagem) é dividida em blocos (4×4, 8×8 ou 16×16 pixeis). O algoritmo básico de codificação é a aplicação de uma transformada discreta do cosseno bidimensional (mesma estratégia do que no *JPEG*), e os resultados são truncados.

Para se obter uma compressão significativamente melhor do que a do *JPEG*, o padrão *H.264* permite codificar cada bloco nos modos *intra* e *inter*.

No modo intra, cada bloco comprimido é comparado com outros blocos já codificados. Apenas a diferença precisa ser armazenada.

No modo inter, o bloco é armazenado como combinação de blocos de quadros já armazenados. Um vetor de deslocamento pode ser utilizado para comprimir objetos em movimento[3].

4.8 Conclusões

Assim, vemos que uma das peças fundamentais em processamento de sinais é a aplicação de transformações ortogonais ao espaço de sinais. Transformações ortogonais preservam o produto interno do espaço de sinais (é um *grupo* de transformações que preserva a *geometria* desse espaço). Elas não amplificam ruídos no sinal.

Existem algoritmos extremamente rápidos para calcular as transformações mais usuais (*fft* e assemelhadas, e transformada de Wavelets) em tempo real. Esses algoritmos funcionam como uma decomposição da transformada em transformadas mais simples.

Se escolhemos a base certa, podemos obter uma boa compressão (ou uma boa filtragem) do sinal por meio de uma projeção ortogonal (que também não amplifica ruídos).

Aqui descrevemos alguma peças fundamentais. Mas a escolha da base depende de uma boa modelagem do processo acústico ou visual, que é não-linear.

4.9 Exercícios

Exercício 4.1. Escreva a matriz de $\mathcal{F}$.

Exercício 4.2. Mostre que $\frac{1}{\sqrt{N}}\mathcal{F}$ é unitária.

Exercício 4.3. Mostre que $\mathcal{F}^2 = -NI$.

Exercício 4.4. Mostre que a transformada discreta do cosseno do sinal f é proporcional à transformada discreta de Fourier do sinal

$$g = [0\ f_1\ 0\ f_2\ 0 \cdots f_T\ 0\ f_T\ 0 \cdots f_2\ 0\ f_1\ 0]\ .$$

Exercício 4.5. Dado um sinal f de 2^m leituras, mostre que a sua transformada de Haar pode ser calculada em tempo $O(n)$. Dica: em uma primeira passagem, separe as "altas freqüências" $T_{\mathrm{H}aar}f(-m, n)$ das "baixas freqüências", representadas por um vetor de $\mathbb{R}^{2^{m-1}}$. Continue recursivamente.

Exercício 4.6. No topo da Figura 4.3, explique por quê aparecem freqüências acima de 500, e por que motivo essa transformada de Fourier parece simétrica.

Exercício 4.7. Uma árvore binária é um grafo sem ciclos, com um vértice privilegiado chamado de raiz, e tal que todo vértice tem grau 1 (e nesse caso é chamado de *folha*) ou grau 3.

- Desenhe todas as árvores binárias com 4 folhas
- Seja $\mathbf{p} \in \mathbb{R}^n$ um vetor de probabilidade: $p_i \geq 0$ e $\sum p_i = 0$. Mostre que existe uma árvore binária contendo folhas numeradas 1 a n, e tal que o comprimento do caminho entre a raiz e a i-ésima folha seja menor ou igual do que $-\log_2 p_i$.

Exercício 4.8. Nas hipóteses do exercício anterior, mostre como associar a todo caminho de comprimento c (saindo da raiz e chegando em uma folha i) uma seqüência binária única $s(i)$ de $c(i)$ dígitos, e com a seguinte propriedade:

Se x é uma variável aleatória discreta assumindo o valor i com probabilidade p_i, mostre que $E(c(x)) \leq -\sum p_i \log_2 p_i$. O número da direita é chamado de *entropia* do vetor de probabilidade $\mathbf{p}$.

Exercício 4.9. Mostre como codificar um sinal discreto aleatório $\mathbf{x} \in \{1, \ldots, n\}^N$ como uma seqüência binária, tal que se $p_i = \mathrm{Prob}[x = i]$, então o valor esperado do tamanho da seqüência codificada é menor ou igual do que a entropia de $\mathbf{p}$, vezes o comprimento N da mensagem original. (Assuma o vetor $\mathbf{p}$ conhecido do codificador e do decodificador). O código construído acima é o *código de Huffman* utilizado para compressão de mensagens discretas.

Referências

[1] Brasil, Ministério das Comunicações, Sistema Brasileiro de Televisão Digital, *Especificação técnica de referência*. `http://sbtvd.cpqd.com.br/`.

[2] Gabriel Bouvigne, *MP3'TECH (página internet)*. `http://www.mp3-tech.org`.

[3] Iain Richardson, *Vcodex (página internet)*. `http://www.vcodex.com`.

[4] Wikemedia Commons, *Gray923.png*. digitalização da *Gray's Anatomy of the Human Body*, domínio público. (1918).

Apêndice A

Complementos de Álgebra Linear

Aqui listamos, por conveniência, alguns resultados importantes de Álgebra Linear, mencionados no corpo do texto. A prova desses resultados pode ser achada em qualquer bom livro de Álgebra Linear.

A.1 Bases e ortogonalidade

Seja E um subespaço de $\mathbb{R}^n$. Uma *base* de E é uma lista ordenada de vetores $(\mathbf{u}_1, \ldots, \mathbf{u}_n)$ tal que:

1. Qualquer vetor $\mathbf{x}$ de $\mathbb{R}^n$ se escreve como combinação linear:

   $$\mathbf{x} = x_1\mathbf{u}_1 + x_2\mathbf{u}_2 + \cdots + x_d\mathbf{u}_d$$

   onde $x_1, \ldots, x_d \in \mathbb{R}$.

2. Qualquer combinação linear

   $$\mathbf{x} = x_1\mathbf{u}_1 + x_2\mathbf{u}_2 + \cdots + x_d\mathbf{u}_d,$$

   onde $x_1, \ldots, x_d \in \mathbb{R}$, é um elemento de E.

Gregorio Malajovich, *Geometria de Algoritmos Numéricos*. Notas em Matemática Aplicada **37** (XXXI CNMAC), SBMAC, São Carlos, 2008.
Copyright © Gregorio Malajovich, 2008.

O número d não depende da escolha da base (isto é um Teorema), e é chamado de *dimensão*. Os números $x_1, \ldots, x_d$ são chamados de *coordenadas* de $\mathbf{x}$.

A definição de base para subespaços de $\mathbb{C}^n$ é a mesma, mas as coordenadas (coeficientes da combinação linear) são números complexos. Dessa maneira, $\mathbb{C}^n$ tem dimensão (complexa) n.

A base canônica de $\mathbb{R}^n$ é $(\mathbf{e}_1, \ldots, \mathbf{e}_n)$ onde

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad , \quad \mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad , \cdots , \quad \mathbf{e}_n = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \ .$$

A base canônica de $\mathbb{C}^n$, também denotada por $(\mathbf{e}_1, \ldots, \mathbf{e}_n)$, é definida da mesma maneira.

Se $\mathbf{x}$ é um vetor real ou complexo, então denotamos por x_k a sua k-ésima coordenada na base canônica. Temos portanto, por definição: $x = \sum_k x_k \mathbf{e}_k$.

O *produto interno canônico* em $\mathbb{R}^n$ é definido por:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{k=1}^{n} u_i v_i \ .$$

Já o produto interno canônico em $\mathbb{C}^n$ é:

$$\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{k=1}^{n} \bar{u}_i v_i \ .$$

Em ambos casos, definimos $\|\mathbf{u}\| = \sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}$ e dizemos que $\mathbf{u}$ e $\mathbf{v}$ são ortogonais se e somente se $\langle \mathbf{u}, \mathbf{v} \rangle = 0$. Se E é um subespaço vetorial de $\mathbb{R}^n$ ou $\mathbb{C}^n$, então $E^\perp$ é o espaço de todos os vetores perpendiculares a todos os vetores de E. Prova-se que $(E^\perp)^\perp) = E$.

A seguir, vamos utilizar a letra $\mathbb{K}$ para enunciar definições e resultados que valem para $K = \mathbb{R}$ ou $K = \mathbb{C}$:

Seja $A : \mathbb{K}^n \to \mathbb{K}^m$ uma transformação linear (que associamos a uma matriz $m \times n$). A *adjunta* de A é a única transformação linear $A^* : \mathbb{K}^m \to \mathbb{K}^n$ tal que, para todos $\mathbf{x}, \mathbf{y}$:

$$\langle A\mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{x}, A^*\mathbf{y} \rangle.$$

Quando $\mathbb{K} = \mathbb{R}$, A^* é associada à transposta A^T de A. Quando $\mathbb{K} = \mathbb{C}$, A^* é associada à conjugada transposta $A^H = \bar{A}^T$ de A.

Definimos o núcleo e a imagem de A por

$$\ker(A) = \{\mathbf{x} \in \mathbb{K}^n : Ax = 0\} \subseteq \beth^n$$

e

$$\mathrm{im}(A) = \{A\mathbf{x} : \mathbf{x} \in \mathbb{K}^n\} \subseteq \beth^m.$$

Teorema A.1 (do posto). *Seja $A : \beth^n \to \beth^m$ uma transformação linear. Então*

1. $$\ker(A) = \mathrm{im}(A^*)^\perp,$$

2. $$\ker(A^*) = \mathrm{im}(A)^\perp, e$$

3. $$\dim \mathrm{im}(A) = \dim \mathrm{im}(A^*) = n - \dim(\ker(A)) = m - \dim(\ker A^*) \ .$$

O número $\dim \mathrm{im}(A)$ é chamado de *posto* de A.

A.2 Bases ortogonais, matrizes ortonormais

Uma base $(\mathbf{u}_1, \ldots, \mathbf{u}_d)$ de um subespaço E é dita *ortonormal* se e somente se

$$\langle \mathbf{u}_i, \mathbf{u}_j \rangle = \begin{cases} 1 & \text{Se } i = j \\ 0 & \text{Se } i \neq j. \end{cases}$$

Uma matriz real Q é dita *ortogonal* quando as suas colunas formam uma base **ortonormal** de $\mathrm{im}(Q)$. Isso é equivalente a:

$$Q^T Q = I.$$

No caso de matrizes complexas, utilizamos uma palavra diferente (a matriz é dita *unitária*) para a mesma definição. Uma matriz complexa Q é unitária se e somente se

$$Q^* Q = I.$$

O produto de duas matrizes ortogonais $n \times n$ é uma matriz ortogonal $n \times n$. O conjunto das matrizes $n \times n$, munido da operação de produto satisfaz aos axiomas de *grupo* (ver qualquer texto de álgebra). Esse grupo é denotado por $O(n)$ (grupo ortogonal de ordem n).

O mesmo vale para matrizes unitárias $n \times n$, e o grupo (denotado por $U(n)$ é chamado de grupo unitário de ordem n.

Além da estrutura de grupo, $O(n)$ e $U(n)$ são variedades diferenciáveis. Objetos combinando essas duas propriedades são chamados de *grupos de Lie*. Do ponto de vista geométrico, $O(n)$ e $U(n)$ são as transformações lineares que preservam ângulos e distâncias.

A.3 Obtendo bases ortonormais

Seja $\mathbf{u}_1, \ldots, \mathbf{u}_d$ base de $E \subseteq \mathbb{K}^n$. O *processo de Gram-Schmidt* permite produzir uma base ortonormal de E a partir dos $\mathbf{u}_i's$.

No primeiro passo, fazemos $\mathbf{v}_1 = \frac{\mathbf{u}_1}{\|\mathbf{u}_1\|}$.

Indutivamente, fazemos

$$\mathbf{w}_i = \mathbf{u}_i - \sum_{1 \leq j < i} \mathbf{v}_j \langle \mathbf{v}_j, \mathbf{u}_i \rangle$$

e

$$\mathbf{v}_i = \frac{\mathbf{w}_i}{\|\mathbf{w}_i\|} \ .$$

Desse modo, $\mathbf{w}_i$ é ortogonal a $\mathbf{v}_1, \ldots, \mathbf{v}_{i-1}$, etc...

Na prática numérica, o processo de Gram-Schmidt é substituído por um algoritmos sofisticado, a *fatoração QR*.

A.4 Normas de matrizes

Definimos a *norma de operador* de uma matriz por:

$$\|A\|_2 = \max_{\|\mathbf{x}\|=1} \|Ax\| \ .$$

Essa definição depende da norma de vetores que foi utilizada. Neste texto, é a norma canônica. Em textos de análise numérica, é conveniente utilizar outras normas mais fáceis de calcular.

Outra norma interessante é a norma de Frobenius :

$$\|A\|_F = \sqrt{\sum_{i,j} |A_{ij}|^2}.$$

A relação entre elas é:

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{n}\|A\|_2.$$

Uma interpretação, utilizando o Teorema 1.4 , é:

$$\|A\|_2 = \sigma_1, \quad \|A\|_F = \sqrt{\sum \sigma_j^2} \ .$$

Agora podemos explicar a popularidade das matrizes ortogonais em análise numérica. Se A é uma matriz (real, complexa) $n \times n$, e queremos calcular $A\mathbf{x}$, mas conhecemos $\mathbf{x}$ com precisão relativa δ, então vamos conhecer $A\mathbf{x}$ com precisão relativa

$$\frac{\|A\|_2\|x\|\delta}{\|Ax\|} \leq \|A\|_2\|A^{-1}\|_2 \ .$$

Se $A \in O(n)$, então teremos $\|A\|_2 = \|A^{-1}\|_2 = 1$. Multiplicar por A é benigno em relação ao erro acumulado.

A.5 Matrizes de Márkov

As *matrizes de Márkov* ou matrizes estocásticas modelam um sistema com n estados, e probabilidades M_{ij} de passar de um estado i a um estado j.

Definição A.2. Uma *Matriz de Márkov* ou *Matriz Estocástica* é uma matriz quadrada M de tamanho $n \times n$, com $M_{ij} \geq 0$ e, para toda coluna j, $\sum_i M_{ij} = 1$. Ela é *positiva* se e somente se $M_{ij} > 0$ para todos i e j.

Observação A.3. Em parte da literatura, matrizes de Márkov são definidas como matrizes com coordenadas não-negativas e onde a soma das coordenadas de cada linha é 1. Assim, estamos transpondo a definição, para poder trabalhar com um vetor coluna de probabilidades.

O principal resultado sobre matrizes de Márkov garante (sob certas circunstâncias) a existência de um *estado estacionário*, que representa a probabilidade do sistema se encontrar em cada estado após tempo infinito.

Teorema A.4 (Perron-Frobenius, caso Markoviano). *Seja M uma matriz de Márkov. Então,*

1. *Se λ é autovalor de M, então $|\lambda| \leq 1$*
2. *1 é autovalor de M.*
3. *Todo autovalor λ de M diferente de 1 verifica $|\lambda| < 1$.*
4. *Existe um autovetor à direita, associado ao autovalor 1, cujas coordenadas são todas não-negativas.*
5. *Se M for positiva, então o espaço dos auto-espaços com autovalor associado 1 tem dimensão 1.*

A.6 Exercícios

Exercício A.1. Mostre que um subespaço vetorial complexo de $\mathbb{C}^n$ de dimensão d pode ser interpretado como um subespaço vetorial $2d$-dimensional real de $\mathbb{R}^{2n}$, mas que a recíproca não é verdadeira.

Exercício A.2. Prove o Teorema de Perron-Frobenius

Referências

[1] Gilbert Strang, *Introduction to Linear Algebra*, Wellesley Cambridge Press, 2003,2005. 3rd edition.

[2] Gregorio Malajovich, *Álgebra Linear*, 2007. Em preparação. `http://www.labma.ufrj.br/~gregorio/`.

Índice

Notas em Matemática Aplicada

Arquivos em pdf disponíveis em http://www.sbmac.org.br/notas.php

1. Restauração de Imagens com Aplicações em Biologia e Engenharia
   Geraldo Cidade, Antônio Silva Neto e Nilson Costa Roberty
2. Fundamentos, Potencialidades e Aplicações de Algoritmos Evolutivos
   Leandro dos Santos Coelho
3. Modelos Matemáticos e Métodos Numéricos em Águas Subterrâneas
   Edson Wendlander
4. Métodos Numéricos para Equações Diferenciais Parciais
   Maria Cristina de Castro Cunha e Maria Amélia Novais Schleicher
5. Modelagem em Biomatemática
   Joyce da Silva Bevilacqua, Marat Rafikov e Cláudia de Lello Courtouke Guedes
6. Métodos de Otimização Randômica: algoritmos genéticos e "simulated annealing"
   Sezimária F. Pereira Saramago
7. "Matemática Aplicada à Fisiologia e Epidemiologia"
   H.M. Yang, R. Sampaio e A. Sri Ranga
8. Uma Introdução à Computação Quântica
   Renato Portugal, Carlile Campos Lavor, Luiz Mariano Carvalho e Nelson Maculan
9. Aplicações de Análise Fatorial de Correspondências para Análise de Dados
   Homero Chaib Filho

10. Modelos Matemáticos baseados em autômatos celulares para Geoprocessamento

    Marilton Sanchotene de Aguiar, Fábia Amorim da Costa, Graçaliz Pereira Dimuro e Antônio Carlos da Rocha Costa

11. Computabilidade: os limites da Computação

    Regivan H. N. Santiago e Benjamín R. C. Bedregal

12. Modelagem Multiescala em Materiais e Estruturas

    Fernando Rochinha e Alexandre Madureira

13. Modelagem em Biomatemática (Coraci Malta ed.)

    1 - "Modelagem matemática do comportamento elétrico de neurônios e algumas aplicações"

    Reynaldo D. Pinto

    2 - "Redes complexas e aplicações nas Ciências"

    José Carlos M. Mombach

    3 - "Possíveis níveis de complexidade na modelagem de sistemas biológicos"

    Henrique L. Lenzi, Waldemiro de Souza Romanha e Marcelo Pelajo- Machado

14. A lógica na construção dos argumentos

    Angela Cruz e José Eduardo de Almeida Moura

15. Modelagem Matemática e Simulação Numérica em Dinâmica dos Fluidos

    Valdemir G. Ferreira, Hélio A. Navarro, Magda K. Kaibara

16. Introdução ao Tratamento da Informação nos Ensinos Fundamental e Médio

    Marcilia Andrade Campos, Paulo Figueiredo Lima

17. Teoria dos Conjuntos Fuzzy com Aplicações

    Rosana Sueli da Motta Jafelice, Laércio Carvalho de Barros, Rodney Carlos Bassanezi

18. Introdução à Construção de Modelos de Otimização Linear e Inteira

    Socorro Rangel

19. Observar e Pensar, antes de Modelar
    Flavio Shigeo Yamamoto, Sérgio Alves, Edson P. Marques Filho, Amauri P. de Oliveira

20. Frações Contínuas: Propriedades e Aplicações
    Eliana Xavier Linhares de Andrade, Cleonice Fátima Bracciali

21. Uma Introdução à Teoria de Códigos
    Carlile Campos Lavor, Marcelo Muniz Silva Alves, Rogério Monteiro de Siqueira, Sueli Irene Rodrigues Costa

22. Análise e Processamento de Sinais
    Rubens Sampaio, Edson Cataldo, Alexandre de Souza Brandão

23. Introdução aos Métodos Discretos de Análise Numérica de EDO e EDP
    David Soares Pinto Júnior

24. Representações Computacionais de Grafos
    Lílian Markenzon, Oswaldo Vernet

25. Ondas Oceânicas de Superfície
    Leandro Farina

26. Técnicas de Modelagem de Processos Epidêmicos e Evolucionários
    Domingos Alves, Henrique Fabrício Gagliardi

27. Introdução à teoria espectral de grafos com aplicações
    Nair Maria Maia de Abreu, Renata Raposo Del-Vecchio, Cybele Tavares Maia Vinagre e Dragan Stevanović

28. Modelagem e convexidade
    Eduardo Cursi e Rubens Sampaio

29. Modelagem matemática em finanças quantitativas em tempo discreto
    Max Oliveira de Souza e Jorge Zubelli

30. Programação não linear em dois níveis: aplicação em Engenharia Mecânica
    Ana Friedlander e Eduardo Fancello

31. Funções simétricas e aplicações em Combinatória
    José Plinio de Oliveira Santos e Robson da Silva
32. Semigrupos aplicados a sistemas dissipativos em EDP
    Carlos Raposo da Cunha