Introdução à Álgebra Linear Computacional

Licio Hernanes Bezerra
Departamento de Matemática - UFSC
Florianópolis - SC - 88040-900

setembro 1993

Conteúdo

Introdução

A Álgebra Linear Computacional assume cada vez mais importância no âmbito da Matemática Aplicada. Isso se deve principalmente ao desenvolvimento da arquitetura de computadores que projeta máquinas com processamento numérico cada vez mais rápido e mais preciso. Problemas de grande porte, que envolvem grande armazenamento de dados e intenso processamento numérico, procuram na Álgebra Linear Computacional o suporte teórico para a sua resolução e assim o estudo dessa matéria se torna cada vez mais necessária e complexa. A sua conexão com outras áreas da Matemática é às vezes surpreendente. Por exemplo, demonstra-se que o metodo QR converge para matrizes simétricas em aritmética exata por fluxo de matrizes. Esse método, que a cada passo conjuga uma dada matriz por matrizes ortogonais e que no limite tende a uma matriz diagonal, é na realidade a avaliação em tempos inteiros da solução de uma equação diferencial matricial.

Basicamente, a Álgebra Linear Computacional se envolve com dois problemas: a resolução de sistemas lineares e o cálculo de autovalores e autovetores de uma matriz, procurando dar a esses problemas uma resposta com o máximo de precisão em um tempo mínimo de computação. Em aritmética de ponto flutuante, os truncamentos e os arredondamentos nas operações algébricas podem resultar em respostas muito diferentes das respostas em aritmética exata. Otimizar o número de operações algébricas é, por isso, uma preocupação constante na hora de se propor uma solução para um determinado problema. A Transformada de Fourier Rápida (FFT) é um exemplo de um algoritmo de multiplicação de matriz por vetor que utiliza $\frac{n^2}{2} log_2 n$ multiplicações em vez das n^3 usuais. Isso se torna possível pelas propriedades intrínsecas da matriz de Fourier. A diferença se traduz num resultado mais preciso e obtido mais rapidamente.

Os capítulos a seguir estao divididos da seguinte forma: no primeiro capítulo, aborda-se a resolução de sistemas lineares, cujas matrizes de coeficientes são inversíveis, por decomposição PLU. No segundo capítulo, sistemas lineares determinados e indeterminados são formulados como um problema de mínimos quadrados (por exemplo, os problemas de ajuste linear de dados). Para resolvê-lo, então, estudamos mais duas decomposições de matriz: QR e DVS (decomposição em valores singulares). No terceiro capítulo, estudam-se alguns métodos computacionais para calcular autovalores de uma matriz. Ao final de cada capítulo há dois tipos de exercícios: os usuais (dedutivos ou construtivos) e os computacionais. Estes foram elaborados para se utilizar o MATLAB, sistema interativo no qual se processam algoritmos matriciais (em alto nível). Para isso, incluiu-se uma pequena introdução ao MATLAB, com alguns comandos básicos, suficientes para resolver os exercícios propostos.

1

Decomposição LU

1.1 Eliminação Gaussiana

O modelo mais simples e o mais utilizado em Matemática Aplicada é um sistema de equações lineares. Nesse capítulo, discutiremos a resolução de sistemas lineares da forma Ax = b, em que A é uma matriz quadrada e inversível. Mais precisamente, focalizaremos nossa atenção sobre a eliminação gaussiana e a álgebra matricial nela envolvida. Vamos começar com o seguinte exemplo :

$$\begin{aligned} x + y + z &= 4 \\ 3x + 2y + 2z &= 10 \\ 2x - y + 3z &= -2 \end{aligned} \tag{1.1}$$

O primeiro passo é eliminar a variável x das duas últimas equações. O pivô dessa operação é o coeficiente de x na primeira equação. Algebricamente, isso é equivalente a

a) subtrair 3 vezes a primeira equação da segunda ;

b) subtrair 2 vezes a primeira equação da terceira.

Essas operações elementares resultam no seguinte sistema

$$\begin{aligned} x + y + z &= 4 \\ -y - z &= -2 \\ -3y + z &= -10 \end{aligned} \tag{1.2}$$

que é equivalente ao anterior (isto é, ambos têm o mesmo conjunto de soluções). No próximo passo, ignoremos a primeira equação e consideremos apenas as duas últimas equações. O objetivo agora é eliminar a variável y da última equação. Para isso, devemos subtrair dessa equação três vezes a segunda. O pivô é o coeficiente de y na segunda equação, (-1). Chegamos então ao seguinte sistema triangular:

$$\begin{aligned} x + y + z &= 4 \\ -y - z &= -2 \\ 4z &= -4 \end{aligned} \tag{1.3}$$

que se resolve por retrossubstituição. Ou seja, calcula-se o valor de z na terceira equação: z = -1. Substitui-se esse valor na segunda equação e então calcula-se o valor de y : y = 3. Finalmente, substituindo-se esses valores na primeira equação, obtemos x = 2.

Essa sistemática se generaliza para sistemas de n equações lineares com n variáveis: elimina-se uma variável após outra até que reste somente uma variável, x_n. Resolve-se então esta variável que, em seguida, é substituída na equação anterior. Esse procedimento se repete assim por diante até que finalmente se resolve x_1. Esse método de resolução de sistemas lineares, que é conhecido como eliminação gaussiana, é limitado aos casos em que os pivôs são não nulos. Um exemplo no qual o método falha é o seguinte:

$$\begin{aligned} x + 2y + z &= 4 \\ 2x + 4y - z &= 11 \\ 5x + 2y + 3z &= 6 \end{aligned}$$

Para eliminar a variável x nas duas últimas equações, subtrai-se da segunda equação duas vezes a primeira e da terceira, cinco vezes a primeira. O resultado dessas operações é o seguinte sistema equivalente:

$$\begin{aligned} x + 2y + z &= 4 \\ -3z &= 3 \\ -8y - 2z &= -14 \end{aligned} \tag{1.4}$$

Não é possível eliminar a variável y da terceira equação subtraindo-se dela um múltiplo da segunda, pois o pivô é zero. Contudo, esse problema se resolve simplesmente permutando-se as duas últimas equações:

$$\begin{aligned} x+2y+z &= 4 \\ -8y-2z &= -14 \\ -3z &= 3 \end{aligned} \tag{1.5}$$

1.2 Matrizes

1.2.1 Matrizes Elementares

Sejam C e A duas matrizes tais que seja possível o seu produto CA. As linhas de CA são combinações das linhas de A, como se pode ver no seguinte exemplo :

$$(x \quad y \quad z) \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} =$$

$$= x \,(a_{11} \quad a_{12} \quad a_{13}) + y \,(a_{21} \quad a_{22} \quad a_{23}) + z \,(a_{31} \quad a_{32} \quad a_{33}) \,.$$

Por exemplo, o seguinte produto transforma a matriz de coeficientes do sistema 1.4 na matriz de coeficientes do sistema 1.5 :

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 & 4 \\ 0 & 0 & -3 & 3 \\ 0 & -8 & -2 & -14 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 1 & 4 \\ 0 & -8 & -2 & -14 \\ 0 & 0 & -3 & 3 \end{pmatrix}.$$

A matriz que multiplicou a matriz de coeficientes do sistema 1.4, á esquerda, é uma matriz de permutação. Em geral, uma matriz de permutação é qualquer matriz cujas linhas são os vetores da base canônica em alguma ordem. Elas serão vistas novamente na próxima seção. Outro exemplo de produto de matrizes é o que transforma o sistema 1.1 no sistema 1.2 :

$$\begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 4 \\ 3 & 2 & 2 & 10 \\ 2 & -1 & 3 & -2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 4 \\ 0 & -1 & -1 & -2 \\ 0 & -3 & 1 & -10 \end{pmatrix}.$$

A transformação operada no sistema 1.1 e que resultou no sistema 1.2 é composta de duas operações elementares. Isso é traduzido em matrizes do seguinte modo: a matriz que multiplica a matriz das constantes numéricas do sistema 1.1 (ver acima) é o produto de duas matrizes elementares. Ou seja,

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}$$

Matrizes elementares são matrizes que multiplicadas à esquerda de uma matriz A resultam em operações elementares nas linhas da matriz A. São de três tipos:

- diagonal, que equivale a multiplicar uma linha por um número;
- de permutação, que equivale a permutar duas linhas;
- do tipo $E_{ij} = I + k_{ij} e_i e_j^T$, onde e_i é a matriz nx1 correspondente ao i-ésimo vetor da base canônica do $\mathbb{R}^n$, que equivale a somar à linha i um múltiplo da linha j.

Exercício 1 *Calcule a inversa de uma matriz elementar e verifique que é também uma matriz elementar.*

Exercício 2 *Mostre que, fixada uma coluna j, quaisquer que sejam as linhas $i_1, i_2 > j$,* $\quad E_{i_1 j} E_{i_2 j} = E_{i_2 j} E_{i_1 j}$.

Exercício 3 *Mostre que se $E_{i_1 j_1}, E_{i_2 j_2}$ são matrizes elementares triangulares inferiores então*

$$E_{i_1 j_1} E_{i_2 j_2} = E_{i_2 j_2} E_{i_1 j_1}, \text{ se } i_1 < i_2 \text{ e } j_1 \leq j_2.$$

Consideremos o sistema linear

$$Ax = b,$$

onde A é uma matriz nxn e b é uma matriz nx1. Seja $[A|b]$ a matriz aumentada correspondente ao sistema. A eliminação gaussiana do ponto de vista matricial é a seguinte multiplicação sucessiva de matrizes elementares, triangulares inferiores :

$$E_{n\,n-1} \cdots E_{ni} \cdots E_{i+1\,i} \cdots E_{n1} \cdots E_{21}[A|b] = [U|\hat{b}],$$

onde U é triangular superior. O pivô do i-ésimo passo de eliminação é o elemento u_{ii}, da diagonal de U. Isso obviamente se não precisarmos permutar linhas durante o processo.

Exercício 4 *Mostre que a eliminação gaussiana chega ao mesmo resultado se computada de modo que a cada linha i são eliminadas as $i-1$ primeiras variáveis, isto é,*

$$E_{n1} \cdots E_{n\,n-1} \cdots E_{i1} \cdots E_{i\,i-1} \cdots E_{21} =$$

$$E_{n\,n-1} \cdots E_{ni} \cdots E_{i+1\,i} \cdots E_{n1} \cdots E_{21}.$$

Esse produto de matrizes que triangulariza A é uma matriz triangular inferior, com todas as entradas na diagonal iguais a 1. Denotando-a por L^{-1}, temos que $L^{-1}A = U$ e assim

$$A = LU.$$

Observemos que

$$L = L_1 L_2 \cdots L_{n-1},$$

em que $L_j = E_{j+1\,j}^{-1} \cdots E_{nj}^{-1}$. Se $E_{ij} = I + k_{ij}e_i e_j^T$, temos que

$$L_j = I - \sum_{i=j+1}^{n} k_{ij}e_i e_j^T.$$

Logo,

$$L = I - \sum_{i>j} k_{ij}e_i e_j^T.$$

A cada passo da eliminação gaussiana, para eliminar uma variável de uma equação, subtraímos dela um múltiplo da equação pivô. L é a matriz triangular inferior formada por esses números e com todos os elementos da diagonal iguais a 1.

Se a matriz A é fatorada em LU, o sistema Ax = b é agora equivalente a dois sistemas triangulares: $Lz = b$ e $Ux = z$. Observemos que L contém a informação de como A se transforma numa matriz triangular superior por operações elementares. Ao se resolver o sistema triangular $Lz = b$, transformamos o vetor b no vetor $z = \hat{b}$, que é a última coluna da matriz aumentada $[U|\hat{b}]$ após eliminação gaussiana em $[A|b]$.

Exercício 5 *Mostre que a inversa de uma matriz triangular inferior (resp. superior) é uma matriz triangular inferior (resp. superior).*

Exercício 6 *Mostre que o produto de matrizes triangulares inferiores (resp. superiores) é uma matriz triangular inferior (resp. superior).*

1.2.2 Pivôs e determinantes

A decomposição LU de uma matriz só é possível se os pivôs na eliminação gaussiana forem não nulos. Se $A = LU$,

$$det(A) = det(L).det(U) = det(U).$$

Mais ainda, se denotarmos por $A(1:i, 1:i)$ a matriz ixi formada pelas entradas de A que estão ao mesmo tempo nas linhas e colunas de 1 a i (submatriz principal de A de ordem i), temos

$$det A(1:i, 1:i) = det U(1:i, 1:i) = u_{11} \cdots u_{ii}$$

Assim, $u_{11} = a_{11}$ e para $i > 1$

$$u_{ii} = \frac{det A(1:i, 1:i)}{det A(1:i-1, 1:i-1)}$$

Temos então a seguinte proposição:

Proposição 1.1 *Uma matriz inversível tem decomposição LU, onde L é uma matriz triangular inferior com 1 na diagonal e U é uma matriz triangular superior, se e só se para todo i ($1 \leq i \leq n$)*

$$det A(1:i, 1:i) \neq 0$$

Exercício 7 *Mostre que a decomposição LU, quando existe, é única. Ou seja, se* $A = L_1U_1 = L_2U_2$*,* L_1, L_2 *são triangulares inferiores com 1 na diagonal e* U_1, U_2 *são triangulares superiores então* $L_1 = L_2$ *e* $U_1 = U_2$*.*

Exercício 8 *Mostre que podemos escrever de modo único* $A = LD\tilde{U}$*, onde* D *é uma matriz diagonal,* L *e* $\tilde{U}$ *são matrizes triangulares respectivamente inferior e superior, com todos as entradas diagonais iguais a 1.*

Exercício 9 *Use a unicidade da decomposição LU para mostrar que a fatoração LDU de uma matriz simétrica* A $(A = A^T)$*, quando existe, é da forma* LDL^T*. Conclua ainda que a fatoração LDU de uma matriz hermitiana* A $(A = A^H$*, em que* A^H *é a transposta conjugada de* $A)$*, quando existe, é da forma* LDL^H*.*

Exercício 10 *Mostre que uma matriz real simétrica* A *é positiva definida*[1] *se e só se* $A = LDL^T$*, com* D *positiva. Analogamente, uma matriz hermitiana* A *é positiva definida*[2] *se e só se* $A = LDL^H$*, com* D *positiva.*

Exercício 11 *Mostre que uma matriz real e simétrica (complexa e hermitiana)* A *é positiva definida se e só se* $A = LL^T$ $(A = LL^H)$*, em que* L *é uma matriz triangular inferior (decomposição de Cholesky).*

Exercício 12 *Considere uma matriz* $A \in \mathbb{C}^{n\times n}$*. Mostre que se* $(\forall x \in \mathbb{C}^n)\,(x \neq 0)\, x^H A x > 0$ *então* $A = A^H$*.*

1.2.3 Pivotamento - fatoração PLU

Vimos na seção anterior que uma matriz A inversível admite fatoração LU se e só se para todo i a submatriz A(1:i,1:i) é inversível. Esse é o

[1] Uma matriz real simétrica $A \in \mathbb{R}^{n\times n}$ é positiva definida se e só se para todo $x \in \mathbb{R}^n$, $x \neq 0$, $x^T A x > 0$.

[2] Uma matriz $A \in \mathbb{C}^{n\times n}$ é hermitiana positiva definida se e só se para todo $x \in \mathbb{C}^n$, $x \neq 0$, $x^H A x > 0$.

caso das matrizes hermitianas positivas definidas, como também, o das matrizes diagonal-dominantes. Fato é que se uma matriz A é inversível existe uma permutação de linhas que transforma A em uma matriz cujas submatrizes principais são todas inversíveis. A demonstração desse fato é o algoritmo de eliminação gaussiana com permutação :

- Se a_{11} é zero permuta-se a primeira linha com uma linha cujo primeiro elemento seja não nulo. Isso é possível, pois A é não singular e, logo, não possui coluna formada por zeros.

- Suponha que já foram eliminadas as i primeiras variáveis permutando-se eventualmente as linhas buscando-se um pivô não nulo. A matriz de coeficientes do sistema é mais ou menos o seguinte :

$$\begin{pmatrix} u_{11} & \cdots & \times & \times & \cdots & \times \\ & \ddots & \vdots & \vdots & \ddots & \vdots \\ & & u_{ii} & \times & \cdots & \times \\ & & & \times & \cdots & \times \\ & & & \vdots & \ddots & \vdots \\ & & & \times & \cdots & \times \end{pmatrix}$$

  Essa matriz é inversível, logo pelo menos uma das n-i últimas linhas tem o elemento da coluna i+1 diferente de zero. Caso não seja a linha i+1, permuta-se ela com alguma não nula.

Demonstramos acima a seguinte proposição:

Proposição 1.2 (Decomposição PLU) *Se A é uma matriz inversível então existe uma matriz de permutação P, uma matriz triangular inferior L com entradas diagonais iguais a 1 e uma matriz triangular superior U tais que $A = PLU$.*

Se a cada passo escolhêssemos o maior pivô possível em valor absoluto, apenas permutando linhas, teríamos o algoritmo que é chamado de pivotamento parcial. Na resolução do sistema triangular $Ux = z$ ($z = L^{-1}b$), há n operações de divisão, que ocorrem quando dividimos expressões algébricas pelos pivôs. O algoritmo de pivotamento parcial procura pivôs grandes para otimizar o processamento numérico, os erros causados por arredondamento na divisão são relativamente menores quanto maior for o quociente.

1.2.4 Matrizes Especiais

Uma fatoração resultante da fatoração LU é a decomposição LDU, em que D é a matriz diagonal formada pelos pivôs e U é agora uma matriz triangular superior com 1 na diagonal. As matrizes hermitianas se decompõem então em LDL^H e são positivas definidas se e somente se $D > 0$ (exercícios 9 e 10). Nesse caso, $A = LDL^H = L\sqrt{D}\sqrt{D}\,L^H = PP^H$. A decomposição de uma matriz hermitiana positiva definida na forma $A = PP^H$, onde P é triangular inferior, é chamada de fatoração de Cholesky.

Exercício 13 *Mostre que se uma matriz A é hermitiana, diagonal estritamente dominante ($\forall i\, |a_{ii}| > \sum_{j \neq i} |a_{ij}|$) e, para todo i, $a_{ii} > 0$ então A é positiva definida.*

Um fato interessante é que a decomposição LDU de matrizes de banda (quando existem) preserva bandas. Por exemplo, os fatores triangulares L e U de uma matriz tridiagonal são bidiagonais, de uma pentadiagonal, tridiagonais, e assim por diante.

A Análise Numérica gerou uma outra classe de matrizes não singulares, que são sensíveis no que diz respeito a solução de sistemas lineares a pequenas perturbações do vetor independente b e que, em geral, resistem a inversão por algoritmos tipo LU[3] - as matrizes mal condicionadas. Um exemplo interessante é o seguinte [2]:

$$\begin{array}{rcrcl} u & + & v & = & 2 \\ u & + & 1.0001v & = & 2 \end{array} \quad \text{e} \quad \begin{array}{rcrcl} u & + & v & = & 2 \\ u & + & 1.0001v & = & 2.0001 \end{array}$$

A solução do primeiro sistema é $u = 2$, $v = 0$; a solução do segundo, $u = v = 1$. A variação relativa do vetor independente do primeiro para o segundo sistema foi

$$\frac{||\Delta b||}{||b||} = \frac{\sqrt{2}}{4}\,10^{-4}\,;$$

a variação relativa da solução,

$$\frac{||\Delta x||}{||x||} = \frac{\sqrt{2}}{2}.$$

[3]Os métodos de resolução de sistemas algébricos lineares baseados em alguma fatoração da matriz de coeficientes do sistema, que dão a solução em um número finito de passos, são chamados em geral de métodos diretos

Ou seja, a variação do vetor independente foi amplificada 20000 vezes !

1.3 Condição de uma Matriz

Dado $p \geq 1$, a p-norma de um vetor de $\mathbb{C}^n$ é definida por

$$\|x\|_p = (|x_1|^p + \cdots + |x_n|^p)^{\frac{1}{p}}.$$

Dois exemplos de p-normas são

$$\|x\|_2 = (|x_1|^2 + \cdots + |x_n|^2)^{\frac{1}{2}} = (x^H x)^{\frac{1}{2}} \quad \text{e}$$

$$\|x\|_\infty = \max_{1 \leq i \leq n} |x_i|.$$

As p-normas podem ser estendidas às matrizes do seguinte modo:

$$\|A\|_p = \max_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}.$$

As p-normas satisfazem a seguinte propriedade (além obviamente das que definem uma norma num espaço vetorial):

$$\|Ax\|_p \leq \|A\|_p \, \|x\|_p.$$

A 2-norma satisfaz ainda mais uma:

$$\|AB\|_2 \leq \|A\|_2 \, \|B\|_2, \tag{1.6}$$

quaisquer que sejam as matrizes $A \in \mathbb{C}^{m \times n}$, $B \in \mathbb{C}^{n \times r}$.

1.3.1 A Sensibilidade de Sistemas Algébricos Lineares

Consideremos os sistemas

$$Ax = b \quad \text{e} \quad A(x + \Delta x) = b + \Delta b.$$

Logo, $A\Delta x = \Delta b$. Ou seja, $\Delta x = A^{-1}\Delta b$. Passando à norma,

$$\|b\| \leq \|A\| \, \|x\| \quad \text{e} \quad \|\Delta x\| \leq \|A^{-1}\| \, \|\Delta b\|.$$

Logo,

$$\frac{\|\Delta x\|}{\|x\|} \leq \|A\|\,\|A^{-1}\|\,\frac{\|\Delta b\|}{\|\Delta b\|} = \kappa(A)\,\frac{\|\Delta b\|}{\|\Delta b\|}. \tag{1.7}$$

Dizemos que

$$\kappa(A) = \|A\|\,\|A^{-1}\|$$

é o número de condição da matriz A, que é sempre maior ou igual a 1. Quanto maior $\kappa(A)$, maior pode ser o erro relativo na solução do sistema, mais mal condicionada é a matriz A.

Se perturbarmos a matriz A no lugar de b,

$$(A + \Delta A)(x + \Delta x) = b,$$

obtemos se $Ax = b$

$$\Delta x = -A^{-1}\,\Delta A\,(x + \Delta x).$$

Passando à 2-norma (ou qualquer norma que satisfaz (1.6)),

$$\|\Delta x\| \leq \|A^{-1}\|\,\|\Delta A\|\,\|x + \Delta x\|,$$

ou seja,

$$\frac{\|\Delta x\|}{\|x + \Delta x\|} \leq \|A^{-1}\|\,\|\Delta A\| \;=\; \kappa(A)\,\frac{\|\Delta A\|}{\|A\|}. \tag{1.8}$$

As desigualdades (1.7) e (1.8) revelam que os erros de arredondamento têm duas fontes: a sensibilidade do problema, que tem o número de condição como uma medida, e os erros Δb e ΔA. Esses erros são cometidos, por exemplo, na fatoração LU em aritmética de ponto flutuante, em que computamos na verdade $A + \Delta A = \hat{L}\hat{U}$ (e não $A = LU$), e na resolução dos sistemas triangulares, onde computamos $\hat{x} = x + \Delta x$ tal que $A\hat{x} = b + \Delta b$. Uma classe de matrizes mal condicionadas (número de condição grande) é a das matrizes de Hilbert, cujas entradas são definidas por $a_{ij} = 1/(i + j - 1)$ (ver MATLAB-exercícios na próxima seção).

1.4 Fórmula de Sherman-Morrison

Suponhamos que $A \in \mathbb{R}^{n\times n}$ é inversível. Sejam u e v dois vetores tais que $v^T A^{-1} u \neq -1$. Então

$$(A + uv^T)^{-1} = A^{-1} - \frac{A^{-1}uv^T A^{-1}}{1 + v^T A^{-1} u}.$$

Essa é a fórmula de Sherman-Morrison para inversão de perturbações de posto 1 de uma matriz A. Essa fórmula se generaliza se $U \in \mathbb{R}^{n\times p}$ e $V \in \mathbb{R}^{p\times n}$:

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1} U)^{-1} V^T A^{-1}.$$

Perturbações de posto p de matrizes facilmente inversíveis são muito frequentes. Um exemplo clássico é a matriz resultante da discretização da equação

$$-\frac{d^2u}{dx^2} = f(x)$$

em que u é periódica:

$$A = \begin{pmatrix} 2 & -1 & 0 & -1 \\ -1 & 2 & -1 & 0 \\ 0 & -1 & 2 & -1 \\ -1 & 0 & -1 & 2 \end{pmatrix}.$$

Um exemplo de perturbação de posto 1 é o de uma matriz que é diagonal em blocos a menos de uma coluna, que representa a comunicação de um dos blocos com todos os outros. Isso é o que acontece, por exemplo, na geração de energia em usinas hidrelétricas: a velocidade (angular) de cada máquina síncrona pode ser formulada tomando-se uma delas como referência. Assim, na coluna associada a velocidade angular dessa máquina aparecerão elementos não nulos, nas linhas correspondentes à variação de velocidade angular das outras máquinas. Outro exemplo (dramático) é o de uma matriz que, a menos de uma entrada não nula, é triangular superior (ver exercício 11 da próxima seção).

Exercício 14 *Verifique a fórmula de Sherman-Morrison.*

1.5 Exercícios

Os exercícios seguintes foram especialmente elaborados para o sistema iterativo MATLAB. A descrição detalhada dos comandos desse sistema pode ser encontrado em *The MATLAB User's Guide[1] ou mais sucintamente no próprio on-line help (é só digitar* **help** *comando). A seguir algumas dicas.*

- *Matrizes são definidas por*

$$A = [a_{11} \cdots a_{1n}; a_{21} \cdots a_{2n}; \cdots; a_{m1} \cdots a_{mn}]$$

Por exemplo, se eu digito

$$\mathbf{A} = [\mathbf{1} \quad \mathbf{2}\ ;\ \mathbf{1} \quad -\mathbf{1}]$$

MATLAB me responde:

$$A = \begin{pmatrix} 1 & 2 \\ 1 & -1 \end{pmatrix}$$

- *Se não se quer eco na tela, deve se por ponto e vírgula ao final do comando. Nesse caso, no exemplo acima, digitar-se-ia*

$$\mathbf{A} = [\mathbf{1} \quad \mathbf{2}\ ;\ \mathbf{1} \quad -\mathbf{1}];$$

- *O comando*

$$\mathbf{B} = \mathbf{A};$$

define a matriz B como sendo igual a A. A matriz A não se perde.

- *MATLAB gera matrizes mxn com números randômicos através dos comandos*

$$\mathbf{A} = \mathbf{rand(m, n)}$$

ou, no caso de uma matriz nxn,

$$\mathbf{A} = \mathbf{rand(n)}$$

- *A matriz identidade é gerada por*

$$\mathbf{eye(n)}$$

- *Eis algumas matrizes especiais definidas por comandos MATLAB:*

compan	*companheira*
diag	*diagonal*
gallery	*exemplos*
hadamard	*Hadamard*
hankel	*Hankel*
hilb	*Hilbert*
invhilb	*inversa de Hilbert*
magic	*quadrado mágico*
pascal	*triângulo de Pascal*
toeplitz	*Toeplitz*
vander	*Vandermonde*

  Para saber como definir um quadrado mágico, por exemplo, digite **help** *magic.*

- *A transposta conjugada de uma matriz A pode ser definida pelo comando*

$$\mathbf{B} = \mathbf{A}'$$

  e o produto de matrizes A e B,

$$\mathbf{C} = \mathbf{A} * \mathbf{B}$$

  Por exemplo, se x é um vetor de n coordenadas (uma matriz nx1) então

$$\mathbf{A} = \mathbf{x} * \mathbf{x}'$$

  é uma matriz nxn de posto 1.

- *Se for necessário continuar o comando na linha seguinte, digite dois pontos um ao lado do outro (**..**) para se interpretar como uma continuação.*

- *Subtrair 3 vezes a primeira linha de A da segunda linha de A é traduzido pelo comando*

$$\mathbf{A(2,:) = A(2,:) - 3 * A(1,:)}$$

  A matriz A é modificada. Se se digitar **A**, *aparecerá a matriz resultante da operação elementar acima.*

- *Em geral, se se quer alterar ou destacar uma submatriz de uma matriz A, os seguintes comandos são possíveis.*
  - **A(:,2) = eye(:,4) faz com que a segunda coluna de A seja substituída pelo quarto vetor da base canônica;**
  - **B = A(2:3,3:5) define a matriz formada pelos elementos que estão ao mesmo tempo nas linhas 2 e 3 e nas colunas 3,4 e 5;**
  - **A([1 3],[2 4]) = [0 1; -1 2] define** $A_{12} = 0$, $A_{14} = 1$, $A_{32} = -1$ **e** $A_{34} = 2$.

  Um exemplo é a programação de uma rotação de Givens:
  $J = eye(n)$
  $J([i\,j],[i\,j]) = [cos(\theta)\ sen(\theta),\ -sen(\theta)\ cos(\theta)]$

- *Se se quer trabalhar com complexos, define-se por comando* $i = sqrt(-1)$; *e então durante toda a sessão, se não for redefinido,* i *será* $\sqrt{-1}$.

- *Se se quer saber o número de flops de uma sequência de operações, digita-se* **flops(0)** *logo antes da sequência se iniciar, para zerar o contador. Após a sequência, digita-se então* **flops**. *Para números reais, cada operação aritmética equivale a um flop. Para complexos, somas e subtrações equivalem a dois flops enquanto multiplicações e divisões, seis flops.*

- *A variável* **eps** *é a precisão da máquina, o menor número positivo tal que* $1 + eps > 1$. *Digita* **eps** *que MATLAB te dará esse número.*

Matlab 1
1. *Gere uma matriz A, 3x3, randômica, e compute passo a passo as operações elementares para triangularizá-la (* $U = A$, $U(2,:) = U(2,:) - U(2,1)/U(1,1) * U(1,:)$ *etc).*
2. *Construa as matrizes elementares E, F e G correspondentes às operações acima.*
3. *Faça o produto G*F*E*A e compare com U acima.*

4. *Para $i > 1$, calcule det(A(1:i,1:i)) / det(A(1:i-1,1:i-1)) e compare com U_{ii}.*

Matlab 2 *Gere uma matriz real 4x4, diagonal dominante, e ache sua decomposição LDU.*

Matlab 3 *Gere uma matriz tridiagonal simétrica positiva definida e encontre sua fatoração de Cholesky (comando L = chol(A)).*

Matlab 4 *O comando [L,U] = lu(A) não gera a "verdadeira" fatoração LU de A. O algoritmo de pivotamento parcial calcula a decomposição $PA = \hat{L}\hat{U}$, enquanto MATLAB gera uma decomposição LU, onde*

$$L = P^T * \hat{L} \quad \text{e} \quad U = \hat{U}.$$

Se for dado o comando [L,U,P] = lu(a), MATLAB exibe a matriz P. Use-o para computar os fatores P, L e U de uma matriz de Toeplitz (o comando A = Toeplitz(c,r) gera uma matriz tal que $a_{ij} = c_{i-j+1}$, se $i \geq j$, e $a_{ij} = r_{j-i+1}$, se $i < j$).

Matlab 5 *Se b é um vetor, o comando $x = A\backslash b$ calcula a solução x do sistema $Ax = b$ por eliminação gaussiana com pivotamento parcial. Faça alguns testes com a matriz do exercício anterior.*

Matlab 6 *De modo geral, se B é uma matriz $n \times p$, o comando*

$$\mathbf{X} = \mathbf{A}\backslash\mathbf{B}$$

calcula a solução X do sistema $AX = B$ por eliminação gaussiana com pivotamento parcial. Inverta matrizes de Hilbert por esse comando. Compare com as inversas das matrizes de Hilbert dadas pelo comando **invhilb**.

Matlab 7 *O comando* **inv(A)** *inverte uma matriz quadrada A, por eliminação gaussiana com pivotamento parcial, computando primeiro inv(L) e inv(U): $inv(A) = inv(U) *$*

$inv(L)$. Inverta matrizes de Hilbert com esse comando e compare os resultados com os obtidos no exercício anterior, com o comando $A \backslash eye(n)$.

Matlab 8 *Pode-se definir uma matriz por um programa. Exemplo:*

```
for i = 1:n
    for j = 1:n
        A(i,j) = 1/(i+j+1);
    end
end
```

Faça $n = 5$ e gere uma matriz A como acima. Ache sua decomposição LU, sua inversa e seu determinante.

Matlab 9 *O comando* **cond(A)** *computa o número de condição da matriz A, segundo a 2-norma. Teste-o com matrizes de Hilbert, por exemplo.*

Matlab 10 *Seja A a parte triangular superior de uma matriz de Hilbert, por exemplo, 7×7 (comando* **A=triu(hilb(7))***). Compute $A^{-1}v$, onde v é o vetor cujas coordenadas são todas iguais a 1. Lembre-se que, para calcular $A^{-1}v$, não se calcula A^{-1} e, sim, resolve-se $Ax = v$. Compare a 'flopagem' de $A^{-1} * v$ com a de $A \backslash v$.*

Matlab 11 *Seja B igual a A, a matriz do exercício anterior, a menos da entrada $(7,1)$: $B(7,1) = 1/3$. Resolva $Bx = v$, onde v é o mesmo vetor do exercício acima, por eliminação gaussiana. Depois resolva a mesma equação usando a fórmula de Sherman-Morrison. Compare as 'flopagens' dos dois processos.*

2

Ortogonalidade

Nesse capítulo, discutiremos a solução de equações do tipo $Ax = b$, em que $A \in \mathbb{C}^{m\times n}$, uma matriz não necessariamente quadrada, e b é um vetor que pode não pertencer ao espaço coluna de A. A fatoração triangular ainda é válida para matrizes retangulares:

Proposição 2.3 *Se $A \in \mathbb{C}^{m\times n}$, existe uma matriz de permutação P, uma matriz quadrada triangular inferior L e uma matriz retangular triangular superior[1] U tais que $PA = LU$.*

Entretanto, abordaremos aqui o problema de resolver sistemas lineares de outra forma, como um problema de minimização de uma forma quadrática, o que dá origem a uma outra fatoração de matriz - a fatoração QR.

2.1 O Problema de Mínimos Quadrados

Um sistema de equações lineares do tipo

$$Ax = b,$$

onde A é uma matriz real $m \times n$, pode ter a seguinte interpretação: de que modo o vetor b pertence ao espaço coluna da matriz A, isto

[1] Nesse capítulo, uma matriz triangular superior (ou inferior) M é uma matriz tal que $M_{ij} = 0$ se $i > j$ ($i < j$); uma matriz diagonal, uma matriz triangular ao mesmo tempo superior e inferior.

é, que combinação linear dos vetores-coluna de A resulta no vetor b ? Por Álgebra Linear, sabemos então que esse sistema é possível se e só se o vetor b pertence ao espaço coluna de A. Além disso, o sistema será determinado se e só se esses vetores-coluna são linearmente independentes. Um outro modo ainda de se resolver o sistema acima é minimizando-se a forma quadrática

$$E^2(x) = \|Ax - b\|_2^2.$$

Se o sistema for impossível ainda assim poderemos encontrar um vetor x tal que a sua imagem por A, $p = Ax$, seja a mais próxima possível de b. Geometricamente, o vetor p corresponde à projeção ortogonal do vetor b sobre o espaço-coluna da matriz A. Denotaremos por x_{mq} o vetor cuja imagem é p e tem norma euclidiana mínima - a solução de mínimos quadrados. Se o posto da matriz A for n, verifica-se que

- $x_{mq} = (A^TA)^{-1}A^Tb$;
- $p = A(A^TA)^{-1}A^Tb$.

Exercício 1 *Mostre que $E^2(x) = \|Ax - b\|_2^2$ tem um mínimo absoluto e verifique que $E(x)$ é mínimo absoluto se e só se $A^TAx = A^Tb$.*

Exercício 2 *Seja $M = \{\|x\|_2 \mid E(x) \text{ é mínimo}\}$. Mostre que M tem um único mínimo.*

Exercício 3 *Mostre que as colunas de uma matriz A são linearmente independentes se e só se A^TA é inversível.*

Exercício 4 *Mostre que um operador linear P é uma projeção sobre um subespaço vetorial se e só se P é simétrica e $P^2 = P$. Verifique que $P = A(A^TA)^{-1}A^T$ é uma projeção.*

Achar p é uma tarefa simples se conhecemos uma base ortonormal do espaço-coluna de A: p é a soma das projeções ortogonais de b sobre cada vetor da base. Em termos matriciais, se Q é a matriz cujas colunas são os vetores daquela base ortonormal,

$$p = QQ^Tb.$$

Se as colunas da matriz A são linearmente independentes, x_{mq} é a única solução de $Ax = p$. Calcular x_{mq} se torna um problema simples, basta fatorar A como um produto QR, onde Q é uma matriz com os vetores-coluna ortonormais e R, triangular inferior inversível. O sistema $Ax = p$ é equivalente então a $QRx = QQ^Tb$, ou seja,

$$Rx = Q^T b.$$

Exercício 5 Ajuste de Dados por Mínimos Quadrados
O problema de se aproximar uma função por um polinômio tem sentido como um problema de mínimos quadrados. Por exemplo, para ajustar os seguintes dados,

$$y = 2 \text{ em } t = 1, \quad y = 1 \text{ em } t = 2 \quad e \quad y = 3 \text{ em } t = 3,$$

por uma função do tipo $y = c + dt$, resolve-se por mínimos quadrados o seguinte sistema

$$\begin{cases} c + d.1 & = & 2 \\ c + d.2 & = & 1 \\ c + d.3 & = & 3 \end{cases}$$

Exercício 5 .1 *Qual é a reta que melhor ajusta os seguintes dados:*

y = 2 em t = -1,   y = 0 em t = 0,
y = -3 em t = 1,   y = -5 em t = 2 ?

Exercício 5 .2 *Qual é a parábola que melhor ajusta os seguintes dados:*

y = 2 em t = -1,   y = 0 em t = 0,
y = 2 em t = 2,   y = 6 em t = 3 ?

Exercício 5 .3 *A tabela abaixo fornece dados experimentais obtidos com machos albinos de tilápia do Nilo pelo Centro de Pesquisas Ictiológicas de Pentecostes (CE):*

tempo t (mês)	**comprimento médio (cm)**
0	**11.0**
1	**15.0**
2	**17.4**
3	**20.5**
4	**22.7**
5	**25.3**
6	**27.4**
7	**28.0**
8	**29.3**

Encontre a função linear que melhor ajusta esses dados. [2]

Exercício 6 *Qual a função da forma* $y = ax^2 + bx + c$, $a \neq 0$, *mais próxima de* $y = \cos x$, *em* $[-\frac{\pi}{2}, \frac{\pi}{2}]$ *?*
$\left(\quad \| f - g \| = \sqrt{\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} (f-g)^2(x)dx} \quad \right)$

2.2 Decomposição QR

Se as colunas de uma matriz A *são linearmente independentes podemos ortonormalizá-las pelo método de Gram-Schmidt.*

Exemplo: *Consideremos a matriz*

$$A = \begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} v_1 & v_2 \end{pmatrix}.$$

Por Gram-Schmidt,

$$q_1 = \frac{v_1}{\|v_1\|}$$

$$q_2 = \frac{v_2 - q_1(v_2^T q_1)}{\|v_2 - q_1(v_2^T q_1)\|},$$

[2]Exercício do livro Equações Diferenciais com Aplicações, de Bassanezi e Ferreira Jr.

ou seja,

$$v_1 = \sqrt{3}\, q_1$$

$$v_2 = \frac{4}{\sqrt{3}}\, q_1 + \frac{\sqrt{6}}{3}\, q_2.$$

Em notação matricial,

$$\begin{pmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\sqrt{3}} & \frac{-\sqrt{6}}{6} \\ \frac{1}{\sqrt{3}} & \frac{\sqrt{6}}{3} \\ \frac{1}{\sqrt{3}} & \frac{-\sqrt{6}}{6} \end{pmatrix} \begin{pmatrix} \sqrt{3} & \frac{4}{\sqrt{3}} \\ 0 & \frac{\sqrt{6}}{3} \end{pmatrix}$$

De modo geral, se A é uma matriz $m \times n$, de posto $1 \leq r \leq n$, pode-se ortonormalizar r colunas de A por Gram-Schmidt de tal modo que $AP = QR$, em que P é uma matriz de permutação $n \times n$, Q é uma matriz $m \times r$ com colunas ortonormais e R é uma matriz retangular rxn tal que $R_{ij} = 0$, se $i > j$. Se $r < n$, há várias soluções para o problema de mínimos quadrados e a determinação da solução x_{mq} é mais complicada (exercício 11).

A implementação computacional do método de ortogonalização de Gram-Schmidt pode ser feita de dois modos equivalentes em aritmética exata, mas com resultados diferentes computacionalmente, em aritmética de ponto flutuante, que é como chamamos a matemática do processamento numérico de uma máquina. O primeiro, chamado de Gram-Schmidt Clássico, é o procedimento utilizado para ortonormalizar bases nos cursos usuais de Álgebra Linear. O segundo, o Gram-Schmidt Modificado, é uma reorganização dos cálculos do primeiro, que tem se mostrado mais eficiente que o Gram-Schmidt Clássico no que diz respeito à ortogonalidade dos vetores computados. A seguir são descritos os dois algoritmos, que decompõem uma matriz A, cujos vetores-coluna são denotados por v_i, $i = 1, \ldots, n$, no produto QR. O segundo algoritmo sobrepõe a matriz Q à matriz A.

1. *Gram-Schmidt Clássico*

   Para $k = 1$

   $\quad q_1 := \frac{v_1}{||v_1||}$

   Para $k = 2, \ldots, n$

$$\begin{array}{ll} s_{ik} := q_i^H v_k & (i = 1, \ldots, k-1) \\ w_k := v_k - \sum_{i=1}^{k-1} s_{ik} q_i & \\ r_{kk} := \sqrt{w_k^H w_k} & \\ q_k := w_k / r_{kk} & \\ r_{ik} := s_{ik} / r_{kk} & (i = 1, \ldots, k-1) \end{array}$$

2. *Gram-Schmidt Modificado*

Para $k = 1, \ldots, n$

$\quad r_{kk} := \sqrt{v_k^H v_k}$

$\quad v_k := v_k / r_{kk}$

$\quad$ *Para* $j = k+1, \ldots, n$

$\quad\quad r_{kj} := v_k^H v_j$

$\quad\quad v_j := v_j - r_{kj} v_k$

Exercício 7 *Faça o seguinte teste dos dois algoritmos GS acima: seja a matriz diagonal D, 20×20, formada pelos números de 1 a 20, nessa ordem. A seguir, partindo do vetor $v_0^T = (1 \ldots 1)$, calcule*

$$v_i = \frac{Dv_{i-1}}{\|Dv_{i-1}\|}, \qquad i = 1, \ldots, 20.$$

Compute os fatores QR da matriz A, cujos vetores-coluna são os $v_i's$ acima. Para cada método, compute também os produtos internos entre os vetores $q_i's$.

Vimos acima como decompor uma matriz A, $m \times n$, de posto n, em um produto QR, onde Q tem colunas ortonormais e R é triangular superior. No caso em que a matriz A tem posto $r < n$, ainda assim pode-se fatorá-la como $A = QRP^T$, onde P é de permutação. Porém, o que usualmente chamamos de decomposição QR de uma matriz é o produto de uma matriz ortogonal (unitária) Q por uma matriz R, tal que $r_{ij} = 0$ se $i > j$.

Uma matriz ortogonal (unitária) é uma matriz real (complexa) **quadrada** *tal que todas as suas colunas são ortonormais. Poderíamos completar os vetores ortonormais dados por Gram-Schmidt até formar* *uma base do $\mathbb{R}^m$, mas tem um modo de decompor uma matriz A muito mais preciso no que diz respeito a ortogonalidade: Householder. Esse método aplica transformações unitárias seguidamente na matriz A até que todos os elementos abaixo da diagonal principal (linha = coluna) se anulem.*

O sucesso desse método se sustenta no fato de que essas transformações unitárias, chamadas de transformações de Householder, são muito simples. O método constrói passo a passo uma sequência de matrizes unitárias

$$Q_1, \ldots, Q_{n-1}$$

tais que

$$Q_1 A = \begin{pmatrix} \times & \times & \cdots & \times \\ 0 & \times & \cdots & \times \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \times & \cdots & \times \end{pmatrix}$$

$$Q_2 Q_1 A = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \times & \cdots & \times \end{pmatrix}$$

$$Q_{n-1} \cdots Q_2 Q_1 A = \begin{pmatrix} \times & \times & \times & \cdots & \times \\ 0 & \times & \times & \cdots & \times \\ 0 & 0 & \times & \cdots & \times \\ \vdots & \vdots & 0 & \ddots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \times \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}$$

As únicas condições impostas sobre a natureza de Q_1 são que ela seja unitária e que leve o primeiro vetor-coluna de A, v_1, em um múltiplo do primeiro vetor da base canônica, e_1. A simetria em relação a um hiperplano bissetor desses dois vetores satisfaz essas condições e, mais ainda, é muito simples: é uma perturbação de posto um da matriz identidade, como veremos a seguir.

Existem dois hiperplanos bissetores aos vetores v_1 e e_1 e para determinar as duas transformações correspondentes, H_1 e H_2, basta calcular a normal de cada hiperplano, n_1 e n_2. Como são unitárias (logo, $\|Hv\| = \|v\|$),

$$H_1(v_1) = \|v_1\|e_1,$$

$$H_2(v_1) = -\|v_1\|e_1.$$

Logo, as duas normais são

$$n_1 = v_1 - H_1(v_1) \quad e \quad n_2 = v_2 - H_2(v_1).$$

Um cálculo simples resulta em

$$H_1 = I - 2\frac{n_1 n_1^H}{n_1^H n_1} \quad e \quad H_2 = I - 2\frac{n_2 n_2^H}{n_2^H n_2}.$$

Essas transformações são ditas de **Householder**. *Agora, um critério prático para a escolha de qual das duas transformações será Q_1: aquela que tiver o maior denominador na fração acima (que ocasionará menor erro de arredondamento). Ou seja, escolhe-se a transformação definida pela normal*

$$n = v_1 + sinal(e_1^T v_1)\|v_1\|e_1.$$

Continuando a sequência, Q_2 deve preservar a primeira coluna de Q_1A, além de satisfazer as condições impostas sobre Q_1. Para isso, basta que

$$Q_2 = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ \vdots & & \tilde{Q}_2 & \\ 0 & & & \end{pmatrix}$$

e $\tilde{Q}_2$ seja uma transformação de Householder. Observemos que dessa forma a dimensão do problema diminui a cada passo de obtenção das matrizes da sequência. Assim, obtemos a seguinte proposição:

Proposição 2.4 (Decomposição QR) *Seja $A \in \mathbb{C}^{m\times n}$, uma matriz de posto n. Então existe uma única matriz unitária Q e uma única matriz triangular superior R, $(\forall i \leq n)$ $R_{ii} > 0$, tal que $A = QR$.*

Exercício 8 *Calcule a expressão da transformação de Householder e verifique que ela é ortogonal e simétrica (unitária e hermitiana, no caso complexo).*

Exercício 9 *Mostre que toda matriz unitária é um produto de matrizes de Householder.*

Exercício 10 *Mostre que a matriz de Fourier F definida por*

$$F_{ij} = \frac{1}{\sqrt{n}}\,\omega^{(i-1)(j-1)},$$

em que $\omega = e^{\frac{2\pi i}{n}}$ (ou $e^{-\frac{2\pi i}{n}}$, como em MATLAB), é unitária.

2.2.1 QR com pivotamento de coluna

Se a matriz A, $m \times n$, é de posto $r < n$, o algoritmo acima com uma estratégia de pivotamento nos fornece uma decomposição do tipo $AP = QR$, em que P é uma matriz de permutação e

$$R = \begin{pmatrix} R_{11} & R_{12} \\ 0 & 0 \end{pmatrix},$$

com R_{11}, $r \times r$, triangular inferior. A estratégia é a seguinte: suponha que no passo k, $k < r$, temos

$$Q_k \cdots Q_1 A P_1 \cdots P_k = R_k,$$

onde

$$R_k = \begin{pmatrix} R_{11}^{(k)} & R_{12}^{(k)} \\ 0 & R_{22}^{(k)} \end{pmatrix},$$

$R_{11}^{(k)}$, $k \times k$, e

$$R_{22}^{(k)} = \begin{pmatrix} v_{k+1}^{(k)} \cdots v_n^{(k)} \end{pmatrix}.$$

Seja $k+1 \le s \le n$ tal que

$$\|v_s^{(k)}\| = \max_{k+1 \le i \le n} \|v_i^{(k)}\|.$$

Seja P_{k+1} a matriz de permutação que troca as colunas s e k+1. Agora, é só obter pelo algoritmo acima a matriz Q_{k+1}. Ao final, temos que para todo $1 \leq i \leq r$, para todo $i+1 \leq j \leq n$,

$$r_{ii}^2 \leq \sum_{k=i}^{j} r_{kj}^2.$$

2.2.2 Decomposição em Valores Singulares - DVS

A decomposição de uma matriz em valores singulares nos revela que qualquer transformação linear entre espaços vetoriais de dimensão finita, analisada a partir de referenciais apropriados no domínio e no contradomínio, nada mais é que uma transformação diagonal não negativa.

Proposição 2.5 (Decomposição em Valores Singulares) *Se $A \in \mathbb{C}^{m\times n}$ é uma matriz de posto $r > 0$ então existem matrizes unitárias $V \in \mathbb{C}^{n\times n}$ e $U \in \mathbb{C}^{m\times m}$ tais que $A = U\Sigma V^H$, onde Σ é uma matriz diagonal positiva, isto é, $\Sigma_{11} \geq \Sigma 22 \geq \cdots \geq \Sigma rr > 0$. Esses elementos são ditos os valores singulares de A.*

A demonstração é a seguinte: seja $\sigma_1 = \max\{\|Av\|\,; \|v\| = 1\}$. Sejam v_1 e u_1 tais que $Av_1 = \sigma_1 u_1$. Sejam V_1 e U_1 duas matrizes unitárias tais que suas primeiras colunas são respectivamente v_1 e u_1. É fácil verificar que

$$U_1^H A V_1 = \begin{pmatrix} \sigma_1 & 0 \\ 0 & A_1 \end{pmatrix}.$$

Repetindo o procedimento com a matriz A_1 e assim por diante, chega-se na matriz Σ.

Exercício 11 *Seja A uma matriz $m \times n$ de posto r e*

$$A = U\Sigma V^H = \sum_{i=1}^{r} \sigma_i u_i v_i^H,$$

a sua decomposição em valores singulares. Verifique que

$$\|Ax - b\|_2^2 = \sum_{i=1}^{r} (\sigma_i y_i - u_i^H b)^2 + \sum_{i=r+1}^{m} (u_i^H b)^2,$$

em que $y = V^H x$. *Mostre que* x *é uma solução para o problema de mínimos quadrados se e só se*

$$y_i = u_i^H b/\sigma_i, \text{ para } i = 1, \ldots, r.$$

Conclua que $x_{mq} = Vy$ *é a solução de mínimos quadrados se* $y_i = u_i^H b/\sigma_i$, *para* $i = 1, \ldots, r$ *e* $y_i = 0$, *para* $i = r+1, \ldots, n$.

Pelo exercício acima,

$$x_{mq} = V\,\Sigma^+\,U^H\,b,$$

em que Σ^+ *é a matriz diagonal tal que* $(\forall i \leq r)\ \Sigma^+_{ii} = \sigma_i$. *A matriz*

$$A^+ = V\Sigma^+ U^H$$

é dita a **pseudoinversa** *de* A.

O problema de encontrar a DVS de uma matriz A *é não linear, os valores singulares são as raízes quadradas dos autovalores não nulos de* $A^H A$; V *e* U *são matrizes de autovetores ortonormais de* $A^H A$ *e* AA^H, *respectivamente. Por operações lineares, chegamos no máximo a uma decomposição da forma*

$$A = U\Theta V^H,$$

em que U *e* V *são unitárias e* Θ *é bidiagonal.*

Se $A = U\Sigma V^H$ *é uma DVS da matriz* A, *temos que*

$$\|A\|_2 = \sigma_1.$$

Se A *é inversível, o número de condição de* A *em relação à 2-norma é então*

$$\kappa_2(A) = \frac{\sigma_1}{\sigma_n}.$$

Uma interpretação geométrica para os valores singulares de uma matriz de posto r *é a seguinte: o elipsóide dado pela equação* $\|Ax\|_2 = 1$ *tem os eixos principais na direção dos vetores coluna de* V *e os comprimentos desses eixos são* $1/\sigma_1, \ldots, 1/\sigma_r$. *Para vermos isso, notemos que*

$$x^H A^H A x = \sum_{i=1}^{r} \sigma_i^2 |y_i|^2,$$

em que $Vy = x$. Matrizes inversíveis mal condicionadas estão relacionadas a elipsóides muito achatados. As matrizes unitárias são as que têm melhor condicionamento, para essas matrizes o elipsóide acima é uma esfera.

No capítulo 1, vimos que a sensibilidade de um sistema linear $Ax = b$, A inversível, era medida pela desigualdade

$$\frac{\|\Delta x\|}{\|x\|} \leq \kappa(A)\,\frac{\|\Delta b\|}{\|b\|}.$$

Se b está na direção do último vetor coluna de V, $b = \alpha v_n$, e Δb, na direção do primeiro vetor coluna de V, $\Delta b = \epsilon v_1$, temos pela DVS que

$$x = A^{-1}b = \sigma_n \alpha u_n \qquad e \qquad \Delta x = A^{-1}\Delta b = \sigma_1 \epsilon u_1.$$

Logo,

$$\frac{\|\Delta x\|}{\|x\|} = \frac{\sigma_1}{\sigma_n}\,\frac{\epsilon}{\alpha} = \kappa_2(A)\,\frac{\|\Delta b\|}{\|b\|}.$$

Isso é o pior que pode acontecer.

Exercício 12 *Dada uma matriz $A \in \mathbb{C}^{m\times n}$, mostre que existem matrizes unitárias U e V tais que*

$$A = U\Theta V^H,$$

em que Θ é bidiagonal (sugestão: multiplique os dois lados de A por matrizes de Householder apropriadas).

Exercício 13 *(**Decomposição Polar**) Seja $A \in \mathbb{C}^{n\times n}$. Mostre que existe matriz unitária Q e matriz hermitiana não negativa P tais que $A = QP$.*

2.3 Exercícios

Matlab 1 *O comando $[Q, R, P] = qr(A)$ computa a fatoração QR com pivotamento de colunas de A. Teste-o com a matriz A, 4×3, tal que $A_{ij} = i + j$.*

Matlab 2 *Compute a decomposição QR da matriz de Hilbert de ordem 5 (comandos* $A = hilb(5)$ *e* $[Q, R] = qr(A)$*).*

Matlab 3 *O comando* $[U, S, V] = svd(A)$ *computa a DSV de* A*. Teste-o com as matrizes acima.*

Matlab 4 *O comando* $Q = orth(A)$ *computa uma base ortonormal para o espaço coluna de* A *e o comando* $Q = null(A)$*, uma base ortonormal para o núcleo. Teste-os, pelo menos com as matrizes acima.*

Matlab 5 *O comando* $B = pinv(A)$ *computa a pseudoinversa de* A*. Teste-o na matriz do primeiro exercício acima.*

Matlab 6 *Ache o plano* $z = ax + by + c$ *que melhor ajuste os seguintes dados: z = 0 em x = y = 0, z = 1 no mesmo ponto, e z = 1 em x = y = 1.*

Matlab 7 *Computar uma transformação de Householder* H *tal que* Hx *é zero abaixo da sua primeira coordenada, em que* x *é uma matriz* $n \times 1$*, é feito em MATLAB a partir do comando* $[H, R] = qr(x)$*.*

Tome uma matriz A real. Compute sua fatoração QR. Em seguida, bidiagonalize $\mathbb{R}^T$ *por transformações de Householder. Uma dica: para gerar uma matriz em blocos do tipo*

$$A = \begin{pmatrix} I & 0 \\ 0 & H \end{pmatrix},$$

onde I *é a matriz identidade* $r \times r$ *e* H*, uma matriz de ordem* $n - r$ *anteriormente definida, dê o comando*

$$\mathbf{a = [eye(r)zeros(1, n - r); zeros(n - r, 1)H]}.$$

Seja P *o produto dessas matrizes de Householder. Verifique que* $Q^T A P^T$ *é bidiagonal superior.*

3

Autovalores

Calcular autovalores de uma matriz $A \in \mathbb{C}^{n\times n}$ *é equivalente a resolver o problema não linear de achar as raízes do seu polinômio característico,*

$$\det(A - \lambda I) = 0.$$

Para cada raiz λ *dessa equação, temos então que* $A - \lambda I$ *é uma matriz singular e o espaço solução do sistema homogêneo a ela associada, o núcleo de* $A - \lambda I$*, tem dimensão não nula. Chamamos esse espaço e cada vetor não nulo pertencente a ele de autoespaço e de autovetor de* A*, respectivamente, ambos associados ao autovalor* λ*.*

Exercício 1 *Ache os autovalores, e respectivos autoespaços, da matriz*

$$A = \begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

Uma matriz é diagonalizável se existe uma matriz inversível P *tal que*

$$AP = PD,$$

em que D *é diagonal. É fácil concluir que nesse caso as colunas de* P *são autovetores de* A *e que as entradas diagonais de* D*, os autovalores.*

Exercício 2 *Mostre que se* A *é uma matriz quadrada então existe matriz* P *tal que* $AP = PD$*,* D *diagonal, se e só se as colunas de* P *são autovetores de* A*. Conclua então que* A *é diagonalizável se e só se existe uma matriz* P *cujos vetores coluna são uma base de autovetores de* A*.*

A seguir um exemplo da utilização de autovalores: considere a sequência de Fibonacci

$$x_0 = 0, \quad x_1 = 1 \quad \text{e, para } k > 1, \quad x_k = x_{k-1} + x_{k-2}.$$

Em termos matriciais, para $k \geq 1$,

$$\begin{pmatrix} x_k \\ x_{k+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} x_{k-1} \\ x_k \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^k \begin{pmatrix} x_0 \\ x_1 \end{pmatrix},$$

ou seja, $u_k = A^k u_0$. *Os autovalores de* A *são*

$$\lambda_1 = \frac{1+\sqrt{5}}{2} \quad e \quad \lambda_2 = \frac{1-\sqrt{5}}{2};$$

$(1, \lambda_1)$ *e* $(1, \lambda_2)$ *são dois autovetores de* A *associados respectivamente a* λ_1 *e a* λ_2. *Assim,*

$$A = SDS^{-1},$$

onde

$$S = \begin{pmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{pmatrix} \qquad e \qquad D = \begin{pmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{pmatrix}.$$

Logo, $u_k = SD^k S^{-1} u_0$ *e, como*

$$S^{-1}u_0 = \begin{pmatrix} 1 & 1 \\ \lambda_1 & \lambda_2 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{\lambda_1 - \lambda_2} \\ \frac{1}{\lambda_2 - \lambda_1} \end{pmatrix},$$

$$u_k = \frac{1}{\lambda_1 - \lambda_2} \begin{pmatrix} \lambda_1^k - \lambda_2^k \\ \lambda_1^{k+1} - \lambda_2^{k+1} \end{pmatrix}.$$

Assim,

$$x_k = \frac{\lambda_1^k - \lambda_2^k}{\lambda_1 - \lambda_2} = \frac{1}{\sqrt{5}} \left[\left(\frac{1+\sqrt{5}}{2} \right)^k - \left(\frac{1-\sqrt{5}}{2} \right)^k \right].$$

Agora, para todo k, $\frac{|\lambda_2|^k}{\sqrt{5}} < 0.5$. *Concluímos então que* x_k *é o inteiro mais próximo de* $\frac{\lambda_1^k}{\sqrt{5}}$.

Como calcular os autovalores de uma matriz? Computar o seu polinômio característico implica em muitas operações numéricas, ocasionando muitos erros de arredondamento. Uma vez computado, calcular suas raízes deve ser feito por métodos iterativos, se a ordem da *matriz for maior que 4, pois não há formas fechadas de resolução de polinômios de grau ≥ 5, conforme teoria de Galois. Há outros métodos para a computação de autovalores que não o cálculo de raízes de um polinômio, baseados em álgebra matricial. Dois deles se destacam na literatura: um, os métodos de potência, que calculam alguns autovalores; outro é o método QR, que computa a forma de Schur da matriz.*

Exercício 3 *Mostre o* **Teorema de Schur**: *$(\forall A \in \mathbb{C}^{n\times n})$ existe uma matriz unitária Q tal que $Q^H AQ = T$, em que T é triangular superior. (sugestão: princípio de indução)*

Exercício 4 *Mostre que se A é uma matriz real e simétrica então existe uma matriz ortogonal Q tal que $Q^T AQ$ é diagonal. Conclua então que os autovalores de uma matriz simétrica são reais e associados a eles existe uma base ortonormal do $\mathbb{R}^n$ de autovetores (sugestão: use o teorema de Schur).*

Exercício 5 *Mostre que se T é uma matriz triangular e $T^H T = TT^H$, T é diagonal. Conclua então que uma matriz A é normal ($AA^H = A^H A$) se e somente se existe uma base ortonormal de $\mathbb{C}^n$ de autovetores de A.*

Exercício 6 *Mostre que se P é inversível, A e $P^{-1}AP$ têm o mesmo polinômio característico.*

Exercício 7 *Mostre que se T é uma matriz triangular (superior ou inferior) e $p(x)$ é o seu polinômio característico, $p(T) = 0$. Conclua então, pelo teorema de Schur, que se A é uma matriz complexa e $p_A(x)$, o seu polinômio característico, $p_A(A) = 0$ (***Teorema de Cayley-Hamilton***).*

Exercício 8 *O polinômio mínimo de uma matriz A é o polinômio mônico de menor grau que se anula em A (logo, divide p_A). Mostre que se P é inversível, as matrizes A e $P^{-1}AP$ têm o mesmo polinômio mínimo.*

Uma consequência interessante do Teorema de Schur (exercício 3) é a seguinte: **os autovalores dependem continuamente das entradas da matriz**. *Para provar, seja uma sequência $\{A_k\}$ de matrizes* *tais que* $A_k \to A$*, quando* $k \to \infty$*. Seja* $A_k = Q_k T_k Q_k^H$*, onde* Q_k *é unitária e* T_k*, triangular superior. O grupo das matrizes unitárias é compacto e, portanto, existe uma subsequência* $\{Q_{k_i}\}$ *de* $\{Q_k\}$ *tal que* Q_{k_i} *converge a uma matriz unitária* Q*. Assim,*

$$\lim_{k\to\infty} Q_k^H A_k Q_k = \lim_{k_i\to\infty} Q_{k_i}^H A_{k_i} Q_{k_i} = Q^H A Q$$

Mas, $\forall k\, Q_k^H A_k Q_k$ *é triangular superior e, assim,* $Q^H A Q$ *só pode ser triangular superior. Conclusão: os autovalores de* A_k*, que são as entradas diagonais de* T_k*, convergem para os autovalores de* A *que estão, todos, na diagonal de* $Q^H A Q$*.*

3.1 Métodos de Potência

Vimos que na sequência de Fibonacci o autovalor $\frac{1+\sqrt{5}}{2}$ *é dominante na avaliação de um termo qualquer da sequência. Em geral, se* $\{x_1, \ldots, x_n\}$ *é uma base de autovetores de* A *associados respectivamente a* $\lambda_1, \ldots, \lambda_n$*, tais que*

$$|\lambda_1| > |\lambda_2| \geq \ldots \geq |\lambda_n|,$$

e

$$u_0 = c_1 x_1 + \ldots + c_n x_n, \quad c_1 \neq 0,$$

temos que a sequência

$$u_k = A u_{k-1},$$

quando k *cresce, tende a um autovetor associado ao autovalor* λ_1*. Isso porque*

$$u_k = A^k u_0 = c_1 \lambda_1^k x_1 + \ldots + c_n \lambda_n^k x_n =$$
$$= \lambda_1^k \left[c_1 x_1 + c_2 \left(\frac{\lambda_2}{\lambda_1}\right)^k x_2 + \ldots + c_n \left(\frac{\lambda_n}{\lambda_1}\right)^k x_n \right]$$

e assim, quanto maior for k*, menor a influência das direções dos autovetores* $x_2, \ldots, x_n$ *em* u_k*.*

Agora, se $|\lambda_1| >> 1$ *(ou* $|\lambda_1| << 1$*),* $\|A^k u_0\|$ *pode ser muito grande (ou muito pequeno). Para evitar erros de arredondamento, é melhor que se divida eventualmente (ou a cada passo) o vetor de iteração por algum número, por exemplo, a sua primeira coordenada. Melhor ainda* *dividir pela coordenada de maior valor absoluto. Dados A e u_0 como acima, o algoritmo*

$$\begin{aligned} \hat{u}_k &= Au_{k-1} \\ a_k &= e_k^T \hat{u}_k, \quad |a_k| = \|\hat{u}_k\|_\infty \\ u_k &= \hat{u}_k / a_k \end{aligned}$$

gera vetores $u_k's$ cada vez mais próximos de $[x_1]$, o subespaço gerado pelo vetor x_1. A rapidez com que esse método converge depende do quociente $|\frac{\lambda_2}{\lambda_1}|$: quanto menor, mais rápido converge.

Os métodos de potência têm a seguinte forma geral

$$u_k = \frac{f(A)u_{k-1}}{a_k},$$

onde $f(A)$ é em geral uma função analítica e a_k, um normalizador, para evitar números muito grandes ou muito pequenos (note que o conceito de número grande ou pequeno faz sentido em aritmética de ponto flutuante). Se tudo correr bem, o vetor de iteração converge para um autovetor associado ao maior autovalor de $f(A)$ em valor absoluto, isto é,

$$\max_{1 \le i \le n} |f(\lambda_i)|.$$

Exercício 9 *Suponha que A é uma matriz diagonalizável. Seja f um polinômio. Mostre que se v é autovetor de A associado a λ então v é autovetor de $f(A)$ associado a $f(\lambda)$. Ou seja, se $A = PDP^{-1}$ então $f(A) = Pf(D)P^{-1}$. Generalize, usando série de Taylor, para funções analíticas.*

Vimos acima o método de potência clássico, com $f(A) = A$. Observamos também que a taxa de convergência depende da razão dos dois maiores autovalores em valor absoluto. Interessante seria se tivéssemos uma função que fosse ao mesmo tempo fácil de ser computada e para a qual esse quociente fosse o menor possível. Observe que o quociente agora é entre os dois maiores autovalores em valor absoluto de $f(A)$; logo, se eu estiver interessado em computar λ_{12}, por exemplo, f deve ser *de tal modo que $f(\lambda_{12})$ seja o maior dos autovalores, em valor absoluto. Uma boa escolha é a função*

$$f(A) = (A - \mu I)^{-1},$$

onde μ é uma estimativa do autovalor em que eu estou interessado.

3.1.1 Iteração Inversa com Deslocamento

Vamos supor que A é diagonalizável, com autovetores $x_1, \ldots, x_n$, associados respectivamente a $\lambda_1, \ldots, \lambda_n$. Seja $u_0 = c_1 x_1 + \ldots + c_n x_n$. Então

$$(A - \mu I)^{-k} u_0 = \frac{c_1}{(\lambda_1 - \mu)^k}\, x_1 + \ldots + \frac{c_n}{(\lambda_n - \mu)^k}\, x_n.$$

Na expressão acima, a direção mais importante é aquela associada ao autovalor mais próximo de μ, ou seja, a

$$|\frac{1}{\lambda - \mu}| = \max_{1 \le i \le n} |\frac{1}{\lambda_i - \mu}|.$$

Quanto mais próximo for μ de algum autovalor, mais rápido o método convergirá. Assim, além da escolha do vetor inicial, é importante a escolha do deslocamento (shift) *inicial μ_0; inicial, porque o próprio método pode dar melhores estimativas para deslocamentos durante o processamento. Por exemplo, o algoritmo seguinte utiliza uma estratégia em que deslocamentos são atualizados a partir de iterações anteriores:*

$u_0 \neq 0$
Para $k \geq 1$
$\quad \hat{u}_k = (A - \mu_{k-1} I)^{-1} u_{k-1}$
$\quad a_k = e_k^T \hat{u}_k, \quad |a_k| = \|\hat{u}_k\|_\infty$
$\quad u_k = \hat{u}_k / a_k$
$\quad \mu_k = \mu_{k-1} + 1/a_k$

Quando A é real e simétrica, se x é um vetor real não nulo,

$$r(x) = \frac{x^T A x}{x^T x}$$

minimiza a forma quadrática $\|Ax - \lambda x\|^2$ *e, logo, é uma boa escolha para deslocamento inicial. Esse número é chamado de* **quociente de Rayleigh de** x. *Quando* A *é simétrica, seus autovalores e respectivos autovetores são reais e o algoritmo de iteração inversa, nesse caso, pode ser reescrito da seguinte forma:*

$$
\begin{array}{l}
u_0 \neq 0 \\
\textit{Para } k \geq 1 \\
\qquad \mu_{k-1} = r(u_{k-1}) \\
\qquad \hat{u}_k = (A - \mu_{k-1} I)^{-1} u_{k-1} \\
\qquad a_k = \|\hat{u}_k\|_2 \\
\qquad u_k = \hat{u}_k / a_k
\end{array}
$$

Esse algoritmo converge globalmente e sua convergência é rápida, até cúbica [3]. Ou seja, a partir de um certo momento o erro diminui cubicamente (por exemplo, $10^{-1}, 10^{-3}, \ldots$*).*

3.1.2 Métodos de Iteração Simultânea

Seja $A \in \mathbb{C}^{n \times n}$. *Vamos supor que* A *é diagonalizável, que* $\lambda_1, \ldots, \lambda_n$ *são seus autovalores e que*

$$|\lambda_1| \geq \ldots \geq |\lambda_p| > |\lambda_{p+1}| \geq \ldots \geq |\lambda_n|.$$

Aplicando o método de potência com $f(A) = A$ *a* p *vetores, simultaneamente, a tendência é dos vetores se aproximarem de um subespaço invariante sob a matriz* A, *de dimensão* p, *associado a*$\lambda_1, \ldots, \lambda_p$ *desde que se tomem alguns cuidados. Um é normalizá-los de algum modo, para que os vetores não fiquem muito "grandes" ou muito "pequenos"; outro, evitar que se tornem linearmente dependentes, por exemplo, ortogonalizando-os eventualmente, ou a cada passo. Esse método é chamado de* **Iteração Ortogonal**. *Há outros aspectos práticos e teóricos envolvidos nesses métodos ([4],[5]), que se tornam particularmente interessantes quando se quer localizar autovalores em uma região do plano complexo, por exemplo, os autovalores mais próximos de um complexo* μ $(f(A) = (A - \mu I)^{-1})$.

3.2 Método QR

Seja $A \in \mathbb{C}^{n\times n}$ *tal que seus autovalores,* $\lambda_1, \ldots, \lambda_n$*, são tais que*

$$|\lambda_1| > \ldots > |\lambda_n| > 0.$$

Seja $A = U_0R_0$*, a fatoração QR de* A*. Tomando* U_0 *como ponto de partida para o método de iteração ortogonal, obteríamos a seguinte sequência:*

$$\begin{array}{l} \text{Para } k \geq 0 \\ \quad A_{k+1} := AU_K \\ \quad U_{k+1}R_{k+1} := A_{k+1} \text{ (fatoração QR de} A_{k+1}) \end{array}$$

Da sequência acima, concluímos que

$$A^2 = U_1R_1R_0, \; \ldots, \; A^{k+1} = U_kR_kR_{k-1}\cdots R_1R_0,$$

ou seja, a cada passo k *a matriz* U_k *é a matriz unitária da decomposição* QR *de* A^{k+1}*. Assim, o primeiro vetor coluna de* U_k *tende ao espaço* $[x_1]$*, o segundo, ao espaço* $[x_1, x_2]$*, etc. Logo,* $U_k^H AU_k$ *tende a uma matriz triangular superior - a forma de Schur de* A*, com os autovalores na diagonal, em ordem decrescente por valor absoluto. Esse ainda não é o método QR, mas tem estreita ligação com ele. O método QR se baseia no seguinte algoritmo:*

$$\begin{array}{l} A_0 := A \\ \text{Para } k \geq 0 \\ \quad Q_kR_k := A_k \text{ (fatoração QR de } A_k) \\ \quad A^{k+1} := R_kQ_k \;\; (= Q_k^H A_kQ_k) \end{array}$$

Exercício 10 *Mostre por indução que*

$$U_k = Q_0Q_1\cdots Q_k,$$

em que U_k *é a matriz unitária da fatoração QR de* A^{k+1} *e* $Q_0, \cdots, Q_k$ *são as matrizes unitárias geradas pelo algoritmo acima (lembre-se que a fatoração QR de uma matriz inversível é única se exigimos que a diagonal de* R *seja positiva).*

Apliquemos agora o algoritmo acima à matriz $A = \begin{pmatrix} 6 & 2 \\ 2 & 3 \end{pmatrix}$, *cujos autovalores são 7 e 2:*

$$Q_0 R_0 = \frac{1}{\sqrt{40}} \begin{pmatrix} 6 & -2 \\ 2 & 6 \end{pmatrix} \begin{pmatrix} \sqrt{40} & 9\sqrt{40}/20 \\ 0 & 7\sqrt{40}/20 \end{pmatrix} := A$$

$$A_1 := R_0 Q_0 = \begin{pmatrix} 7 - 1/10 & 7/10 \\ 7/10 & 2 + 1/10 \end{pmatrix}$$

$$Q_1 R_1 = \frac{1}{\sqrt{4810}} \begin{pmatrix} 69 & -7 \\ 7 & 69 \end{pmatrix} \begin{pmatrix} \sqrt{4810}/10 & 63/\sqrt{4810} \\ 0 & 140/\sqrt{4810} \end{pmatrix} := A_1$$

$$A_2 := R_1 Q_1 = \begin{pmatrix} 7 - 4/481 & 98/481 \\ 98/481 & 2 + 4/481 \end{pmatrix}$$

Observemos que a sequência, já nos dois primeiros termos, mostra tendência de convergir para a matriz diagonal formada por 7 e 2. Mas, para cada decomposição QR são necessárias $O(n^3)$ *operações se a matriz não é esparsa. Há um modo desse número de operações diminuir: operar com uma matriz conjugada à matriz original por uma matriz unitária, a sua forma de Hessemberg* H. *Uma matriz de Hessemberg superior é uma matriz da forma*

$$\begin{pmatrix} \times & \times & \cdots & \cdots & \times \\ \times & \times & \cdots & \cdots & \times \\ 0 & \times & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \times & \times \end{pmatrix}$$

Ou seja, é uma matriz H *tal que* $h_{ij} = 0$, *se* $i \geq j + 2$. *A matriz de Hessemberg inferior é a transposta de uma Hessemberg superior.*

Proposição 3.6 (Forma de Hessemberg) $\forall A \in \mathbb{C}^{n \times n}$ *existe uma matriz unitária* Q *tal que* $Q^H A Q = H$, *matriz de Hessemberg superior.*

A prova da proposição acima se baseia num procedimento que gera a matriz unitária Q *como um produto de transformações de Householder [6]. Notemos que uma matriz de Hessemberg conjugada a uma matriz hermitiana é uma matriz tridiagonal. A seguir algumas propriedades interessantes das matrizes de Hessemberg:*

- *a fatoração* QR *de uma matriz de Hessemberg envolve* $O(n^2)$ *operações e, se ela for tridiagonal, apenas* $O(n)$*;*

- *se* $H = QR$*,* Q *e* $RQ = Q^H HQ$ *são também de Hessemberg.*

Exercício 11 *Mostre que se* H *é uma matriz de Hessemberg e* $H = QR$*,* Q *é unitária e* R *é triangular superior,* Q *e* RQ *são ambas matrizes de Hessemberg.*

Agora, observemos o seguinte exemplo:

$$QR = \frac{1}{\sqrt{20}} \begin{pmatrix} 4 & 2 \\ 2 & -4 \end{pmatrix} \begin{pmatrix} \sqrt{20} & \sqrt{20}/2 \\ 0 & 0 \end{pmatrix} := \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}$$

$$A_1 := RQ = \begin{pmatrix} 5 & 0 \\ 0 & 0 \end{pmatrix}$$

O método convergiu acima numa só iteração. De modo geral, se A *é de Hessemberg, o posto de* A *é* $r < n$ *e* $A = QR$*, então* R *tem* r *linhas nulas correspondentes às* r *colunas de* Q *que estão no espaço ortogonal ao espaço coluna de* A*. Assim, o produto* RQ *tem* r *linhas nulas e o problema de autovalores de* A *se desacopla em problemas menores de autovalores. A idéia é fazer então um deslocamento em* A *próximo a algum autovalor. Que deslocamento escolher? Já vimos que o método* QR *tem estreita ligação com o método de potência* ($f(A) = A$)*, que privilegia as direções associadas aos maiores autovalores de* A *em valor absoluto. Se a matriz* A *for real e seus autovalores* $\lambda_1, \ldots, \lambda_n$ *forem tais que* $|\lambda_1| > \cdots > |\lambda_n|$ *(logo, todos reais), a entrada* $(n, n-1)$ *de* A_k *tende a ficar pequena,*

$$A_k = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \epsilon & \hat{\lambda}_n \end{pmatrix}$$

ou seja, a entrada (n, n) *de* A_k *tende a* λ_n *e, assim por diante. Para matrizes reais e simétricas com espectro simples (autovalores distintos dois a dois), já foi mostrado que esse algoritmo converge para uma matriz diagonal [7].*

Vejamos então como fica o método QR com esta estratégia de deslocamento aplicado à mesma matriz A de um exemplo anterior:

$$Q_0R_0 = \frac{1}{\sqrt{13}}\begin{pmatrix} 3 & 2 \\ 2 & -3 \end{pmatrix}\begin{pmatrix} \sqrt{13} & 6/\sqrt{13} \\ 0 & 4/\sqrt{13} \end{pmatrix} := A - 3I$$

$$A_1 := R_0Q_0 + 3I = \begin{pmatrix} 7 - 1/13 & 8/13 \\ 8/13 & 2 + 1/13 \end{pmatrix}$$

$$\begin{aligned} Q_1R_1 &= \frac{1}{\sqrt{4033}}\begin{pmatrix} 63 & 8 \\ 8 & -63 \end{pmatrix}\begin{pmatrix} \sqrt{4033}/13 & 8/\sqrt{4033} \\ 0 & 64/13\sqrt{4033} \end{pmatrix} \\ &:= A_1 - \left(2 + \frac{1}{13}\right) I \end{aligned}$$

$$\begin{pmatrix} 7 - 1/52429 & 512/52429 \\ 512/52429 & 2 + 1/52429 \end{pmatrix} = R_1Q_1 + \left(2 + \frac{1}{13}\right) I$$

Vemos acima que o método QR com deslocamento converge muito mais rapidamente que o método QR simples (pelo menos nas duas primeiras iterações).

Método QR com deslocamento *Seja $A \in \mathbb{C}^{n\times n}$ uma matriz de Hessemberg superior. O algoritmo seguinte é dito o algoritmo QR com deslocamento simples:*

$A_0 := A$
Para $k \geq 0$
  $\mu_k := (A_k)_{nn}$
  $Q_kR_k := A_k - \mu_k I$ *(fatoração QR de* $A_k - \mu I$*)*
  $A^{k+1} := R_kQ_k + \mu_k I$ $(= Q_k^H A_k Q_k)$

Se A é uma matriz real não simétrica, uma estratégia de deslocamento para detectar possíveis autovalores complexos seria a seguinte: suponhamos que no passo k

$$A_k = \begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & a & b \\ 0 & 0 & c & d \end{pmatrix}$$

é uma matriz real e que a matriz $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ *tenha autovalores complexos conjugados,* λ *e* $\overline{\lambda}$*. Fazemos então dois deslocamentos simples consecutivos, um com* λ*, outro com* $\overline{\lambda}$*:*

$$\begin{aligned} &Q_k R_k := A_k - \lambda I \\ &A^{k+1} := R_k Q_k + \lambda I \\ &Q_{k+1} R_{k+1} := A_{k+1} - \overline{\lambda} I \\ &A^{k+2} := R_{k+1} Q_{k+1} + \overline{\lambda} I \end{aligned}$$

Há porém um modo [6] de se fazer os dois deslocamentos acima apenas em aritmética real, pelo fato de

$$(Q_k Q_{k+1})(R_{k+1} R_k) = (A_k - \lambda I)(A_k - \overline{\lambda} I).$$

Como a matriz que está à direita da igualdade é real (A_k *é real), a equação acima é a sua fatoração QR. Logo,* $Q_k Q_{k+1}$ *e* $R_{k+1} R_k$ *são reais. Lembremos que*

$$A_{k+2} = Q_{k+1}^H Q_k^H A_K Q_k Q_{k+1},$$

ou seja, podemos passar de A_k *para* A_{k+2} *achando a fatoração QR de uma matriz real, tudo em aritmética real. Se tudo der certo, chegaremos a uma matriz do tipo*

$$\begin{pmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \epsilon & e & f \\ 0 & 0 & g & h \end{pmatrix},$$

em que ϵ *é bem pequeno e, logo, os autovalores da matriz*

$$\begin{pmatrix} e & f \\ g & h \end{pmatrix}$$

são autovalores da matriz A.

3.3 Exercícios

Matlab 1 *O comando* **p = poly(A)** *dá os coeficientes do polinômio característico de A. O comando* **roots(p)** *dá as raízes do polinômio p. Compute os autovalores de matrizes do tipo*

$$\begin{pmatrix} 0 & 1 & & & \\ n-1 & 0 & 2 & & \\ & n-2 & 0 & \ddots & \\ & & \ddots & \ddots & n-1 \\ & & & 1 & 0 \end{pmatrix}.$$

Matlab 2 *O comando* **polyvalm(p,A)** *avalia o polinômio p na matriz A. Verifique o Teorema de Cayley-Hamilton em matrizes randômicas ou em matrizes de sua escolha.*

Matlab 3 *Seja $A = \begin{pmatrix} 2 & -2 \\ -1 & 2 \end{pmatrix}$. Aplique o método QR em A.*

Matlab 4 *Aplique o método QR com deslocamento na mesma matriz do exercício anterior.*

Matlab 5 *Defina uma matriz A, compute $p = poly(A)$ e suas raízes. Escolha uma das raízes ou um número próximo para deslocamento no método de iteração inversa. Para isso, escolha um vetor inicial v_0 e inverta a matriz por eliminação gaussiana. O comando* **max(v)** *calcula a coordenada de v de maior valor absoluto. Se escolher um número próximo de uma das raízes do polinômio característico, mude o deslocamento eventualmente usando a fórmula $\mu_k = \mu_{k-1} + 1/s$, onde s é a coordenada de maior valor absoluto do vetor resultante da iteração $(A - \mu_{k-1} I)^{-1} v_{k-1}$.*

Matlab 6 *O comando* **eig(A)** *acha os autovalores de A. Se eu der o comando* **[X,D] = eig(A)**, *MATLAB computa a matriz X de autovetores de A e a matriz diagonal D, com seus autovalores. Calcule os autovalores e autovetores de matrizes de Vandermonde (digite* **help vander**).

Matlab 7 *O comando* **fft(A)** *aplica a transformada rápida de Fourier a cada coluna de A, se a ordem de A for uma potência de 2. Se $A = eye(n)$, a matriz identidade, $n \times n$, a matriz de Fourier é definida por*

$$F = \frac{1}{\sqrt{n}} fft(A).$$

Mostre que as colunas de F são autovetores da matriz de permutação P, matriz cuja primeira linha é e_2^T, o segundo vetor da base canônica, a segunda linha é e_3^T e, assim por diante, até a última linha, que é e_1^T.

Matlab 8 *MATLAB computa os autovalores e autovetores de uma matriz pelo método QR. Ou seja,*

- *primeiro acha H, a forma Hessemberg de A. Teste o comando* **H=hess(A)** *em alguma A de sua escolha, de preferência, real com autovalores complexos.*
- *Depois, se H é complexa, tenta computar a forma de Schur de H; se for real e tiver autovalores complexos, computa a forma real de Schur, QSQ^T, em que S é quase triangular superior (uma matriz de Hessemberg superior desacoplada em blocos 2×2, correspondentes a autovalores complexos, ou em blocos 1×1, correspondentes a autovalores reais). Teste o comando* **S=schur(H)** *e verifique se S coincide com a forma de Schur de A.*
- *Depois computa os autovetores pelo método de iteração inversa (faça isso também).*

Esses são os passos utilizados na implementação computacional do método QR [13]. Vimos que tudo deve correr bem se os autovalores não forem repetidos. Se você der o comando **help gallery**, *MATLAB vai te mostrar exemplos em que nem tudo corre tão bem. Teste o comando* **eig(A)** *em matrizes A do tipo $A = PJP^{-1}$, em que J é bidiagonal com autovalores muito próximos, quase Jordan.*

Referências Bibliográficas

[1] C.B.Moler, J.Little e S.Bangert, PC-Matlab User's Guide, *The MathWorks Inc., 1987.*

[2] G.Strang, Linear Algebra and its Applications, 3rd edition, *Harcourt Brace Jovanovich, Publishers, 1988.*

[3] B.N.Parlett, The Rayleigh Quocient Iteration and Some Generalizations for Nonnormal Matrices, *Math.Comp. 28, 695-710, 1974.*

[4] M.Clint e A.Jennings, A Simultaneous Iteration Method for the Unsymmetric Eigenvalue Problem, *J.Inst.Math.Applic.8,111-121,1971.*

[5] G.W.Stewart, Simultaneous Iteration for Computing Invariant Subspaces of Non-Hermitian Matrices, *Numer.Math. 25, 123-136, 1976.*

[6] G.H.Golub e C.F.Van Loan, Matrix Computations, 2nd edition, *The Johns Hopkins University Press, 1989.*

[7] P.Deift. T.Nanda e C.Tomei, Ordinary Differential Equations and the Symmetric Eigenvalue Problem, *SIAM J.Numer.Anal. 20, 1-22, 1983.*

[8] B.N.Parlett, The Symmetric Eigenvalue Problem, *Prentice-Hall, 1980.*

[9] G.Strang, Introduction to Applied Mathematics, *Wellesley-Cambridge Press, 1986.*

[10] T.F.Coleman e C.F.Van Loan, Handbook for Matrix Computations, *SIAM, 1988.*

[11] J.Dongarra, J.R.Bunch, C.B.Moler e G.W.Stewart, LINPACK User's Guide, *SIAM, 1978.*

[12] B.T.Smith, J.M.Boyle, J.Dongarra, B.Garbow, Y.Ikebe, V.C. Klema e C.B.Moler, Matrix Eigensystem Routines: EISPACK Guide, 2nd edition, *Springer-Verlag, 1976.*

[13] R.C.Bassanezi e W.C.Ferreira Jr., Equações Diferenciais com aplicações, *Harbra, 1988.*