Everton Boos

Métodos Iterativos para a Pseudo-Inversa de Moore-Penrose e Aplicações na Resolução de Sistemas Lineares

Florianópolis
2015

Everton Boos

Métodos Iterativos para a Pseudo-Inversa de Moore-Penrose e Aplicações na Resolução de Sistemas Lineares

Trabalho de Conclusão de Curso apresentado ao Curso de Matemática do Departamento de Matemática do Centro de Ciências Físicas e Matemáticas da Universidade Federal de Santa Catarina para obtenção do título de Bacharel em Matemática e Computação Científica.

Universidade Federal de Santa Catarina – UFSC
Centro de Ciências Físicas e Matemáticas
Departamento de Matemática

Orientador: Prof. Dr. Fermín S. V. Bazán

Florianópolis
2015

Esta monografia foi julgada adequada como **TRABALHO DE CONCLUSÃO DE CURSO** no Curso de Matemática — Habilitação Bacharelado em Matemática e Computação Científica e aprovada em sua forma final pela Banca Examinadora designada pela Portaria n. 15/2015/CCM.

Florianópolis, 26 de novembro de 2015:

Prof. Dra. Silvia Martini de Holanda Janesch
Coordenadora do Curso

Everton Boos
Acadêmico

Banca Examinadora:

Prof. Dr. Fermín S. V. Bazán
Orientador

Prof. Dr. Douglas Soares Gonçalves
Membro

Prof. Dr. Leonardo Silveira Borges
Membro

Florianópolis
2015

Resumo

A busca por soluções para sistemas de equações lineares é assunto de estudo em muitas áreas das ciências e matemática. Uma maneira de encontrar soluções para estes sistemas é a partir do uso da matriz pseudo-inversa, que possui teoria bem desenvolvida, mas é pouco aplicada na prática devido ao alto custo computacional de calculá-la. Visando evitar o cálculo explícito da matriz pseudo-inversa, partimos para o uso de métodos iterativos, aplicando os mesmos na construção de uma sequência vetorial que converge para a solução do sistema linear. Para tanto, foram estudados aspectos teóricos sobre convergência bem como um conjunto de implementações computacionais no ambiente interativo científico MATLAB. Percebemos que ainda há muito o que desenvolver e melhorar, para termos soluções mais precisas e mais rápidas do ponto de vista computacional.

Palavras-chave: Matriz pseudo-inversa. Métodos iterativos. Soluções aproximadas para sistemas lineares. Implementação computacional.

Abstract

The search for solutions to systems of linear equations is subject of study in many areas of science and mathematics. One way to find solutions for these systems is using the pseudoinverse matrix, which has a well developed theory, but it is rarely applied in practice due to the high computational cost to calculate it. In order to avoid the explicit computation of the pseudoinverse matrix, we make use of iterative methods, applying them in the construction of a vector sequence that converges to the solution of the linear system. To this end, we studied theoretical aspects of convergence as well as a set of computational implementations in scientific interactive environment MATLAB. It is noticed that there is still much to develop and improve, in order to have more accurate and faster solutions on the computational point of view.

Keywords: Pseudoinverse matrix. Iterative methods. Approximate solutions for linear systems. Computational implementation.

Sumário

Introdução

O problema de resolver sistemas de equações lineares é assunto recorrente em diversas áreas das ciências aplicadas. Em geral, o objetivo é resolver problemas da forma

$$\text{a)} \qquad Ax = b, \qquad \text{ou} \qquad \text{b)} \qquad \min_{x \in \mathbb{C}^n} \|Ax - b\|_2,$$

com $A \in \mathbb{C}^{m \times n}$, $b \in \mathbb{C}^m$, e $x \in \mathbb{C}^n$ (vetor de incógnitas). Quando o problema a) é tal que $m = n$ e A tem posto completo, então a solução é única e dada por $x = A^{-1}b$. Porém, em certos casos, o problema a) não admite solução. Para estes, buscamos resolver o problema de mínimos quadrados b). Por outro lado, podemos ter um sistema com várias soluções, e fica claro que o conceito de matriz inversa não é suficiente. Assim, foi desenvolvido o estudo da chamada matriz pseudo-inversa de Moore-Penrose, denotada por $A^\dagger$, que busca generalizar o conceito da inversa.

Baseado na matriz pseudo-inversa (que existe para qualquer matriz), o vetor $\hat{x} = A^\dagger b$ representa a solução de norma mínima do problema de mínimos quadrados associado. Note que esta caracterização compreende os casos apresentados acima, o que torna seu estudo de interesse teórico e prático.

É sabido que computar a matriz inversa não é recomendável devido ao alto custo computacional, portanto, o cálculo do produto $A^{-1}b$ pode ser inviável. A mesma observação é válida para o caso do produto $A^\dagger b$. Para contornar estas dificuldades, partimos para métodos iterativos para aproximar a pseudo-inversa, mas tendo em mente que, ao invés de aproximarmos a própria pseudo-inversa, o principal objetivo é calcular aproximações do efeito dela quando aplicada ao vetor b.

Em termos matemáticos, numa primeira etapa a ideia é gerar uma sequência $\{X_k\}_{k \geq 0}$ de matrizes de modo que $X_k \longrightarrow A^\dagger$, quando $k \longrightarrow \infty$. Tendo garantida a convergência $X_k \longrightarrow A^\dagger$, a ideia da segunda etapa é construir uma sequência de vetores $x_k \in \mathbb{C}^n$, baseada na sequência X_k, e analisar sob quais condições, $x_k \longrightarrow A^\dagger b$.

Com relação à organização do trabalho, o Capítulo 1 apresenta uma série de conceitos introdutórios necessários às construções dos capítulos seguintes. Em especial, notamos uma breve teoria da decomposição em valores singulares (SVD) e de projeções, e o caso particular da projeção ortogonal.

No Capítulo 2, apresentamos os principais aspectos teóricos da matriz pseudo-inversa, como definição, teorema de existência e unicidade, propriedades e aplicações na solução de sistemas lineares. O material dos Capítulos 1 e 2 é extraído principalmente de [1, 3, 5, 9, 12].

A teoria voltada para a construção das sequências matriciais convergentes para $A^\dagger$ é encontrada no Capítulo 3, em que mostramos três classes principais de métodos

iterativos, com suas respectivas análises e condições para convergência, cujo material se baseia em [3, 13, 14].

A partir disto, no Capítulo 4 construímos sequências vetoriais que convergem para $A^{\dagger}b$, baseadas nos métodos iterativos desenvolvidos no capítulo anterior. Aqui, nos focamos principalmente em dois dos métodos iterativos vistos, por serem de maior interesse numérico.

Finalmente, no Capítulo 5 temos alguns resultados numéricos da aplicação do melhor método vetorial desenvolvido. Aplicamos o mesmo em problemas de teste reconhecidamente mal condicionados, provenientes de [7] e da Galeria do MATLAB, buscando capturar a solução do problema original a partir do problema perturbado. Para tanto, fizemos uso de alguns critérios de parada especiais, sendo eles o princípio de discrepância [8, 11], um critério baseado no conceito de curva-L [6, 8] e a regra do produto mínimo [2, 4].

1 Preliminares

Neste capítulo, procuramos desenvolver os conceitos que serão utilizados ao longo de todo o texto. Estes conceitos e pequenos resultados normalmente são vistos em um curso de Álgebra Linear, por isso serão somente citados aqui.

Seja $\mathbb{C}$ o *corpo dos números complexos* e $\mathbb{C}^{m\times n}$ o *espaço das matrizes* $m \times n$ *com coeficientes em* $\mathbb{C}$. Se $x, y \in \mathbb{C}^n$, definimos o *produto interno canônico* $\langle \cdot, \cdot \rangle : \mathbb{C}^n \times \mathbb{C}^n \to \mathbb{C}$ e a *norma canônica* $\| \cdot \|_2 : \mathbb{C}^n \to \mathbb{R}$ em $\mathbb{C}^n$ por

$$\langle x, y \rangle := \sum_{j=1}^{n} \overline{x_j} y_j \text{ e } \|x\|_2 := \sqrt{\langle x, x \rangle} = \left(\sum_{j=1}^{n} \overline{x_j} x_j \right)^{\frac{1}{2}},$$

em que $\overline{x_j}$ representa o *conjugado* de x_j. Daqui em diante, assumiremos que os vetores em $\mathbb{C}^n$ são vetores coluna, isto é,

$$x \in \mathbb{C}^n \Leftrightarrow x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}, x_i \in \mathbb{C}.$$

Portanto, tomando x e y em $\mathbb{C}^n$, o produto interno é $\langle x, y \rangle = \overline{x}^T y$, em que $\overline{x}^T$ representa o vetor *transposto* de $\overline{x}$.

Se $A \in \mathbb{C}^{m\times n}$, considere a matriz $A^* \in \mathbb{C}^{n\times m}$ definida por $A^* := \overline{A}^T \equiv \overline{A^T}$. A^* é chamada de *matriz adjunta* associada a A. Em particular, se $A \in \mathbb{C}^{m\times n}$, então, para todo $x \in \mathbb{C}^n$ e todo $y \in \mathbb{C}^m$, tem-se $\langle Ax, y \rangle = \langle x, A^* y \rangle$. A é dita *auto-adjunta* ou *hermitiana* se $A = A^*$. Além disso, A é dita *unitária* se $A^* = A^{-1}$. Um caso de matriz unitária segue se tomarmos $\{v_1, \ldots, v_n\}$ uma base ortonormal de $\mathbb{C}^n$ e definirmos $A := [v_1, \ldots, v_n] \in \mathbb{C}^{n\times n}$, com $v_i \in \mathbb{C}^n$.

Considerando uma matriz $A \in \mathbb{C}^{m\times n}$, definimos o *núcleo* (ou *kernel*) de A por $Ker(A) := \{x \in \mathbb{C}^n \mid Ax = 0\}$ e a *imagem* de A por $Im(A) := \{y \in \mathbb{C}^m \mid Ax = y, \text{ para algum } x \in \mathbb{C}^n\}$.

Agora, seja $\mathcal{V}$ um espaço vetorial com produto interno e $\mathcal{S}$ um subconjunto de $\mathcal{V}$. Definimos o *complemento ortogonal* $\mathcal{S}^\perp$ de $\mathcal{S}$ por

$$\mathcal{S}^\perp := \{x \in \mathcal{V} \mid \langle s, x \rangle = 0, \forall s \in \mathcal{S}\}.$$

Além disso, dois subespaços $\mathcal{X}$ e $\mathcal{Y}$ de um espaço $\mathcal{V}$ são ditos *complementares* sempre que $\mathcal{V} = \mathcal{X} + \mathcal{Y}$ e $\mathcal{X} \cap \mathcal{Y} = \{\mathbf{0}\}$, lembrando que $\mathcal{X} + \mathcal{Y} = \{x + y \mid x \in \mathcal{X} \text{ e } y \in \mathcal{Y}\}$. Neste caso, $\mathcal{V}$ é a *soma direta* de $\mathcal{X}$ e $\mathcal{Y}$ e é representada por $\mathcal{V} = \mathcal{X} \oplus \mathcal{Y}$.

Teorema 1 (Subespaços Ortogonais Complementares)**.** [9] *Se $\mathcal{S}$ é um subespaço de um espaço $\mathcal{V}$ de dimensão finita com produto interno, então $\mathcal{V} = \mathcal{S} \oplus \mathcal{S}^\perp$. Além disso, se $\mathcal{N}$ é*

subespaço tal que $\mathcal{V} = \mathcal{S} \oplus \mathcal{N}$ *e* $\mathcal{N} \perp \mathcal{S}$ *(isto é, todo vetor de* $\mathcal{N}$ *é ortogonal a todo vetor de* $\mathcal{S}$*), então* $\mathcal{N} = \mathcal{S}^{\perp}$.

Teorema 2 (Decomposição Ortogonal)**.** [3] *Para cada* $A \in \mathbb{C}^{m\times n}$*, tem-se*

$$Im(A)^{\perp} = Ker(A^*) \text{ e } Ker(A)^{\perp} = Im(A^*).$$

Assim, cada matriz $A \in \mathbb{C}^{m\times n}$ *produz uma decomposição ortogonal de* $\mathbb{C}^m$ *e* $\mathbb{C}^n$ *no seguinte sentido:*

$$\mathbb{C}^m = Im(A) \oplus Im(A)^{\perp} = Im(A) \oplus Ker(A^*) \text{ e}$$
$$\mathbb{C}^n = Ker(A) \oplus Ker(A)^{\perp} = Ker(A) \oplus Im(A^*).$$

1.1 Decomposição em Valores Singulares

A decomposição em valores singulares, mais conhecida como SVD (devido ao termo em inglês ser *singular value decomposition*), é uma maneira de fatorar matrizes. De fato, dada uma matriz A, buscamos matrizes U, V e Σ de modo a termos $A = U\Sigma V^*$. Esta decomposição será muito útil na construção e demonstração de vários fatos relacionados à matriz pseudo-inversa.

Agora, vejamos um pequeno resultado antes de construirmos a decomposição.

Lema 1. *Seja* $A \in \mathbb{C}^{m\times n}$*. Então* $A^*A \in \mathbb{C}^{n\times n}$ *é hermitiana e todos os seus autovalores são reais não negativos.*

Demonstração. Perceba que $(A^*A)^* = A^*(A^*)^* = A^*A$. Portanto, A^*A é hermitiana. Agora, considere $\lambda \in \mathbb{C}$ um autovalor qualquer de A^*A. Assim, existe $v \in \mathbb{C}^n$, $v \neq 0$, tal que $A^*Av = \lambda v$. Logo,

$$\lambda\langle v, v\rangle = \langle \lambda v, v\rangle = \langle A^*Av, v\rangle = \langle Av, Av\rangle = \|Av\|_2^2 \geq 0.$$

Como $\langle v, v\rangle > 0$, temos que $\lambda \geq 0$, obrigatoriamente. Além disso, fica claro que $\lambda \in \mathbb{R}$, pois $\lambda = \frac{\|Av\|_2^2}{\langle v,v\rangle}$, em que ambos (numerador e denominador) são reais. Para esclarecer, o passo $\langle A^*Av, v\rangle = \langle Av, Av\rangle$ segue diretamente do fato de A^* ser o adjunto de A. □

Vale observar que o resultado acima também é válido para AA^*.

Definição 1. *Sejam* $A \in \mathbb{C}^{m\times n}$ *e* $0 < \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_p$*, para algum* $p \leq min\{m, n\}$*, os autovalores de* A^*A*. Assim, definimos os* valores singulares *não nulos de* A *por* $\sigma_k := \sqrt{\lambda_k}$*, para todo* $k = 1, \ldots, p$*. Note que* $0 < \sigma_1 \leq \sigma_2 \leq \ldots \leq \sigma_p$.

Perceba que esta definição é precisa, já que o Lema 1 acima nos garante a não-negatividade de todos os autovalores de A^*A. De posse disso, podemos enunciar o teorema seguinte.

Teorema 3 (Existência da SVD)**.** [5] *Sejam $A \in \mathbb{C}^{m\times n}$ e $0 < \sigma_1 \leq \sigma_2 \leq \ldots \leq \sigma_p$ os valores singulares não nulos de A. Então existem $U \in \mathbb{C}^{m\times m}$ e $V \in \mathbb{C}^{n\times n}$ unitárias tais que*

$$U^*AV = \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{m\times n}, \text{ em que } C := \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \end{bmatrix} \in \mathbb{C}^{p\times p}.$$

Demonstração. Seja $\{v_1, \ldots, v_n\}$ uma base ortonormal de $\mathbb{C}^n$ tal que $A^*Av_i = \sigma_i^2 v_i$, para $i \in \{1, \ldots, p\}$ (ou seja, $\{v_1, \ldots, v_p\}$ é base ortonormal de autovetores), e $A^*Av_i = 0$, para $i \in \{p+1, \ldots, n\}$ (estes vetores podem ser gerados completando a base $\{v_1, \ldots, v_p\}$). Assim, defina $V := [v_1, \ldots, v_n] \in \mathbb{C}^{n\times n}$. V é claramente unitária. Agora, estamos buscando $U \in \mathbb{C}^{m\times m}$ unitária de modo que

$$AV = U \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{1.1}$$

Para isso, defina, para $i \in \{1, \ldots, p\}$, $u_i := \frac{1}{\sigma_i} Av_i$. Logo, para $i, j \in \{1, \ldots, p\}$,

$$\langle u_i, u_j \rangle = \frac{1}{\sigma_i \sigma_j} \langle Av_i, Av_j \rangle = \frac{1}{\sigma_i \sigma_j} \langle A^*Av_i, v_j \rangle = \frac{\sigma_i^2}{\sigma_i \sigma_j} \langle v_i, v_j \rangle = \delta_{ij},$$

em que $\delta_{ij} = 1$ se $i = j$ e $\delta_{ij} = 0$ se $i \neq j$ (conhecido como *delta de Kronecker*). Agora, completamos $\{u_1, \ldots, u_p\}$ a uma base ortonormal $\{u_1, \ldots, u_m\}$ de $\mathbb{C}^m$ e definimos $U := [u_1, \ldots, u_m] \in \mathbb{C}^{m\times m}$. Deste modo, temos que U é unitária.
Falta verificar que U satisfaz (1.1). Para $j \in \{1, \ldots, p\}$, temos que a j-ésima coluna de AV (aqui denotada por $c_j(AV)$) é:

$$c_j(AV) = Av_j = \sigma_j u_j, \text{ já que, para estes } j\text{'s, } u_j := \frac{1}{\sigma_j} Av_j.$$

Por outro lado,

$$c_j \left(U \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) = U\sigma_j e_j = \sigma_j U e_j = \sigma_j u_j.$$

Agora, se considerarmos $j \in \{p+1, \ldots, n\}$, temos:

$$c_j(AV) = Av_j = 0 \text{ pois } \langle Av_j, Av_j \rangle = \langle A^*Av_j, v_j \rangle = \langle \mathbf{0}, v_j \rangle = 0$$

e

$$c_j \left(U \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \right) = U\mathbf{0} = \mathbf{0}.$$

Logo, vale (1.1) para as matrizes U e V acima definidas. Assim, temos o resultado. □

Em geral, é introduzida a seguinte notação:

$$\Sigma := \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}.$$

Tendo em vista isso, o teorema acima nos diz que $U^*AV = \Sigma$. Mas, por U e V serem unitárias, segue que

$$A = U\Sigma V^*,$$

que é conhecida como *SVD* de A.

1.2 Projeções

A breve teoria sobre projeções desenvolvida aqui tem por objetivo central seu uso nas propriedades relacionadas à matriz pseudo-inversa.

Definição 2. *Suponha que $\mathcal{V} = \mathcal{X} \oplus \mathcal{Y}$, isto é, para todo $v \in \mathcal{V}$, existem únicos $x \in \mathcal{X}$ e $y \in \mathcal{Y}$ tal que $v = x + y$. Assim, o vetor x é chamado de* projeção *de v em $\mathcal{X}$ ao longo de $\mathcal{Y}$. Analogamente, o vetor y é chamado de* projeção *de v em $\mathcal{Y}$ ao longo de $\mathcal{X}$.*

Se considerarmos $\mathcal{V} = \mathbb{C}^n = \mathcal{X} \oplus \mathcal{Y}$, para um par de subespaços complementares $\mathcal{X}$ e $\mathcal{Y}$, e um vetor $v \in \mathbb{C}^n$, a pergunta é: como construir um operador (chamado de *projetor*) $P \in \mathbb{C}^{n\times n}$ de modo que Pv é a projeção de v em $\mathcal{X}$ ao longo de $\mathcal{Y}$?

Observação 1. Em alguns casos, P é representado por $P_{\mathcal{X},\mathcal{Y}}$, de modo a explicitar que P é o projetor em $\mathcal{X}$ ao longo de $\mathcal{Y}$.

Considere $\beta_{\mathcal{X}} = \{x_1, \ldots, x_r\}$ e $\beta_{\mathcal{Y}} = \{y_1, \ldots, y_{n-r}\}$ bases de $\mathcal{X}$ e $\mathcal{Y}$, respectivamente, de modo que $\beta_{\mathcal{X}} \cup \beta_{\mathcal{Y}}$ é base de $\mathbb{C}^n$. Assim, a matriz $B := [x_1, \ldots, x_r \mid y_1, \ldots, y_{n-r}] = [X \mid Y]$ é não-singular.

Como procuramos P projetor, então $Px_i = x_i$, para $i = 1, \ldots, r$, e $Py_j = 0$, para $j = 1, \ldots, n-r$. Assim, $PB = P[X \mid Y] = [PX \mid PY] = [X \mid \mathbf{0}]$. Consequentemente,

$$P = [X \mid \mathbf{0}]B^{-1} = B \begin{bmatrix} I_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} B^{-1}.$$

Precisamos verificar que P assim definida é, de fato, a projeção de v em $\mathcal{X}$ ao longo de $\mathcal{Y}$. Para isso, seja $x = Pv$ e $y = (I - P)v$ e observe que $v = x + y$, em que

$$x = Pv = [X \mid 0]B^{-1}v \in Im(X) = \mathcal{X}$$

e

$$y = (I - P)v = B \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n-r} \end{bmatrix} B^{-1}v = [0 \mid Y]B^{-1}v \in Im(Y) = \mathcal{Y}.$$

O argumento acima estabelece que o *projetor complementar*, isto é, o projetor em $\mathcal{Y}$ ao longo de $\mathcal{X}$, é dado por

$$Q = I - P = [\mathbf{0} \mid Y]B^{-1} = B \begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \mathbf{0} & I_{n-r} \end{bmatrix} B^{-1}.$$

A proposição abaixo resume propriedades dos projetores.

Proposição 1. [9] *Sejam $\mathcal{X}$ e $\mathcal{Y}$ subespaços complementares de um espaço vetorial $\mathcal{V}$ tal que para cada $v \in \mathcal{V}$, existem únicos $x \in \mathcal{X}$ e $y \in \mathcal{Y}$ tal que $v = x + y$. O único operador linear P definido por $Pv = x$ é chamado de projetor em $\mathcal{X}$ ao longo de $\mathcal{Y}$ e tem as seguintes propriedades:*

(i) $P^2 = P$ *(ou seja, P é* idempotente*);*

(ii) $I - P$ *é o projetor complementar em $\mathcal{Y}$ ao longo de $\mathcal{X}$;*

(iii) $Im(P) = \{x \mid Px = x\}$;

(iv) $Im(P) = Ker(I - P) = \mathcal{X}$ *e* $Im(I - P) = Ker(P) = \mathcal{Y}$;

(v) Se $\mathcal{V} = \mathbb{C}^n$, então P é dado por

$$P = [X \mid \mathbf{0}][X \mid Y]^{-1} = [X \mid Y] \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [X \mid Y]^{-1},$$

em que as colunas de X e Y são bases de $\mathcal{X}$ e $\mathcal{Y}$, respectivamente.

Demonstração. Unicidade: se P_1 e P_2 satisfazem a condição para serem projetores, então $P_1 v = P_2 v$, para cada $v \in \mathcal{V}$, de modo que $P_1 = P_2$.
Linearidade: se $v_1 = x_1 + y_1$ e $v_2 = x_2 + y_2$, com $x_1, x_2 \in \mathcal{X}$ e $y_1, y_2 \in \mathcal{Y}$, e $\lambda \in \mathbb{C}$, então

$$P(\lambda v_1 + v_2) = P((\lambda x_1 + x_2) + (\lambda y_1 + y_2)) = \lambda x_1 + x_2 = \lambda P v_1 + P v_2.$$

(i) Perceba que, para cada $v \in \mathcal{V}$, tem-se

$$P^2 v = P(Pv) = Px = x = Pv.$$

Logo, $P^2 = P$.

(ii) Observe que $v = x + y = Pv + y$. Portanto, $y = v - Pv = (I - P)v$, de modo que $I - P$ é o projetor em $\mathcal{Y}$ ao longo de $\mathcal{X}$.

Os itens (iii) e (iv) seguem facilmente da definição de projetor. Já o item (v) foi desenvolvido imediatamente acima desta proposição. □

Teorema 4. [14] *Para toda matriz idempotente $A \in \mathbb{C}^{n \times n}$, temos que $Im(A)$ e $Ker(A)$ são subespaços complementares de $\mathbb{C}^n$, com $A = P_{Im(A), Ker(A)}$.*
Reciprocamente, se $\mathcal{L}$ e $\mathcal{M}$ são subespaços complementares de $\mathbb{C}^n$, então existe uma única matriz idempotente $P_{\mathcal{L},\mathcal{M}} \in \mathbb{C}^{n \times n}$ tal que $Im(P_{\mathcal{L},\mathcal{M}}) = \mathcal{L}$ e $Ker(P_{\mathcal{L},\mathcal{M}}) = \mathcal{M}$.

Demonstração. Seja $A \in \mathbb{C}^{n \times n}$ uma matriz idempotente, então podemos escrever $x \in \mathbb{C}^n$ da forma $x = Ax + (I - A)x$. Segue, pela teoria de projeção desenvolvida acima, que $\mathbb{C}^n$ é soma de $Im(A)$ e $Ker(A)$. Para mostrar que $Im(A) \cap Ker(A) = \{\mathbf{0}\}$, note que do fato de $x \in Im(A)$ segue que $Ax = x$, e como $x \in Ker(A)$ então $Ax = \mathbf{0}$, portanto $x = \mathbf{0}$.

Assim, $\mathbb{C}^n = Im(A) \oplus Ker(A)$, e como $x = Ax + (I - A)x$ para todo $x \in \mathbb{C}^n$, então Ax é a projeção de x sobre $Im(A)$ ao longo de $Ker(A)$.
Reciprocamente, sejam $\mathcal{L}$ e $\mathcal{M}$ subespaços complementares de $\mathbb{C}^n$ com bases $\{x_1, x_2, \ldots, x_l\}$ e $\{y_1, y_2, \ldots, y_m\}$, respectivamente. Se existe uma projeção $P_{\mathcal{L},\mathcal{M}}$ tal que $Im(P_{\mathcal{L},\mathcal{M}}) = \mathcal{L}$ e $Ker(P_{\mathcal{L},\mathcal{M}}) = \mathcal{M}$, então ela deve satisfazer

$$P_{\mathcal{L},\mathcal{M}} x_i = x_i, i = 1, 2, \ldots, l,$$
$$P_{\mathcal{L},\mathcal{M}} y_i = \mathbf{0}, i = 1, 2, \ldots, m.$$

Vamos construir uma matriz idempotente $P_{\mathcal{L},\mathcal{M}}$ que satisfaça as condições acima, assim tal matriz será a projeção sobre $\mathcal{L}$ ao longo de $\mathcal{M}$. Seja $X = [x_1, x_2, \ldots, x_l]$ e $Y = [y_1, y_2, \ldots, y_m]$, então $P_{\mathcal{L},\mathcal{M}}$ deve satisfazer $P_{\mathcal{L},\mathcal{M}}[X \mid Y] = [X \mid \mathbf{0}]$. Além disso, pelo fato de que $\{x_1, x_2, \ldots, x_l, y_1, y_2, \ldots, y_m\}$ é uma base de $\mathbb{C}^n$, então $[X \mid Y]$ é não singular. Assim, defina $P_{\mathcal{L},\mathcal{M}} = [X \mid \mathbf{0}][X \mid Y]^{-1}$.
Vejamos que $P_{\mathcal{L},\mathcal{M}}$ é idempotente. De fato, como $P_{\mathcal{L},\mathcal{M}}[X \mid Y] = [X \mid \mathbf{0}]$, ou seja, $P_{\mathcal{L},\mathcal{M}} X = X$ e $P_{\mathcal{L},\mathcal{M}} Y = \mathbf{0}$, temos

$$P^2_{\mathcal{L},\mathcal{M}} = P_{\mathcal{L},\mathcal{M}}[X \mid \mathbf{0}][X \mid Y]^{-1} = [X \mid \mathbf{0}][X \mid Y]^{-1} = P_{\mathcal{L},\mathcal{M}}.$$

Basta mostrar que $P_{\mathcal{L},\mathcal{M}}$ é única. Para isso, suponha que existe outra matriz idempotente E tal que $Im(E) = \mathcal{L}$ e $Ker(E) = \mathcal{M}$, então

$$Ex_i = x_i, i = 1, 2, \ldots, l,$$
$$Ey_i = \mathbf{0}, i = 1, 2, \ldots, m,$$

De onde segue que $E[X \mid Y] = [X \mid \mathbf{0}]$ e portanto $E = [X \mid \mathbf{0}][X \mid Y]^{-1} = P_{\mathcal{L},\mathcal{M}}$. □

O teorema a seguir nos dá uma condição necessária e suficiente para um dado operador ser um projetor. Em particular, o resultado garante que todo operador idempotente é um projetor.

Teorema 5. [9] *Um operador linear P em $\mathcal{V}$ é um projetor se, e somente se, $P^2 = P$.*

Demonstração. ($\Rightarrow$) Se P é projetor, segue diretamente da Proposição 1 que $P^2 = P$.
($\Leftarrow$) Vejamos que por $P^2 = P$, temos que $Im(P)$ e $Ker(P)$ são subespaços complementares. Para provar isto, perceba que $\mathcal{V} = Im(P) + Ker(P)$, pois para cada $v \in \mathcal{V}$, tem-se $v = Pv + (I - P)v$, em que $Pv \in Im(P)$ e $(I - P)v \in Ker(P)$. Além disso, $Im(P) \cap Ker(P) = \{\mathbf{0}\}$, pois se

$$x \in Im(P) \cap Ker(P) \Rightarrow x = Py \text{ e } Px = 0 \Rightarrow x = Py = P^2 y = P(Py) = Px = 0.$$

Logo, temos que $Im(P)$ e $Ker(P)$ são complementares. Portanto, cada $v \in \mathcal{V}$ pode ser escrito unicamente como $v = x + y$, em que $x \in Im(P)$ e $y \in Ker(P)$. Por $x \in Im(P)$, existe $z \in \mathcal{V}$ tal que $Pz = x$. Logo,

$$Pv = P(x + y) = Px + Py = P(Pz) + 0 = P^2 z = Pz = x.$$

Assim, P é projetor. □

1.2.1 Projeção Ortogonal

Definição 3. *Para $v \in \mathcal{V}$, seja $v = u+w$, em que $u \in \mathcal{M}$ e $w \in \mathcal{M}^{\perp}$. Assim, u é chamada* projeção ortogonal *de v em $\mathcal{M}$. Além disso, o projetor $P_{\mathcal{M}}$ em $\mathcal{M}$ ao longo de $\mathcal{M}^{\perp}$ é chamado de* projetor ortogonal *em $\mathcal{M}$.*

É imediato do que foi desenvolvido acima que $P_{\mathcal{M}}$ é o único operador linear tal que $P_{\mathcal{M}}v = u$.

A partir da Proposição 1, se $\mathcal{M}$ e $\mathcal{N}$ são subespaços complementares de $\mathbb{C}^n$, então o projetor P em $\mathcal{M}$ ao longo de $\mathcal{N}$ é dado por

$$P = [M \mid \mathbf{0}][M \mid N]^{-1} = [M \mid N] \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} [M \mid N]^{-1},$$

em que M e N formam uma base para $\mathcal{M}$ e $\mathcal{N}$, respectivamente. Agora, se $\mathcal{N} = \mathcal{M}^{\perp}$, pode-se simplificar a expressão acima. Observe que, neste caso, $N^*M = \mathbf{0}$ e $M^*N = \mathbf{0}$. Além disso, se $\dim\mathcal{M} = r$, então M^*M é $r \times r$ e $posto(M^*M) = posto(M) = r$, de modo que M^*M é não-singular. Por outro lado, escolhendo as colunas de N de modo a ser base ortonormal de $\mathcal{M}^{\perp}$, então

$$\begin{bmatrix} (M^*M)^{-1}M^* \\ N^* \end{bmatrix} [M \mid N] = \begin{bmatrix} I & \mathbf{0} \\ \mathbf{0} & I \end{bmatrix} \Longrightarrow [M \mid N]^{-1} = \begin{bmatrix} (M^*M)^{-1}M^* \\ N^* \end{bmatrix}.$$

Assim, juntando as informações, temos que o projetor ortogonal em $\mathcal{M}$ é dado por

$$P_{\mathcal{M}} = [M \mid \mathbf{0}][M \mid N]^{-1} = [M \mid \mathbf{0}] \begin{bmatrix} (M^*M)^{-1}M^* \\ N^* \end{bmatrix} = M(M^*M)^{-1}M^*.$$

Como sabemos, o projetor associado a quaisquer dois subespaços complementares é único, de modo que podemos escolher qualquer base dos espaços respectivos para formar M e N. Assim, a fórmula $P_{\mathcal{M}} = M(M^*M)^{-1}M^*$ é válida sempre, independente da escolha de M (respeitando, é claro, que as colunas de M sejam base para $\mathcal{M}$).

Agora, se escolhermos uma base ortonormal de $\mathcal{M}$ para formar M, então temos $M^*M = I$ e, logo, $P_{\mathcal{M}} = MM^*$. Mais ainda, se escolhermos M e N de modo que suas colunas formem uma base ortonormal de $\mathcal{M}$ e $\mathcal{M}^{\perp}$, respectivamente, então $U := [M \mid N]$ é tal que $U^{-1} = U^*$. Portanto, tem-se outra fórmula para o projetor:

$$P_{\mathcal{M}} = U \begin{bmatrix} I_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} U^*.$$

De forma mais concisa, a proposição a seguir faz um apanhado geral das ideias acima.

Proposição 2. [9] *Seja $\mathcal{M}$ um subespaço de $\mathbb{C}^n$ com $\dim\mathcal{M} = r$. Além disso, considere $M \in \mathbb{C}^{n\times r}$ e $N \in \mathbb{C}^{n\times n-r}$ de modo que as colunas formam uma base para $\mathcal{M}$ e $\mathcal{M}^{\perp}$, respectivamente. Assim, os projetores ortogonais em $\mathcal{M}$ e $\mathcal{M}^{\perp}$ são:*

$$P_{\mathcal{M}} = M(M^*M)^{-1}M^* \text{ e } P_{\mathcal{M}^{\perp}} = N(N^*N)^{-1}N^*.$$

Caso M e N contém bases ortonormais de $\mathcal{M}$ e $\mathcal{M}^\perp$, então:

- $P_{\mathcal{M}} = MM^*$;
- $P_{\mathcal{M}^\perp} = NN^*$;
- $P_{\mathcal{M}} = U \begin{bmatrix} I_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} U^*$, *em que* $U := [M \mid N]$.

Ademais, em todos os casos é válido: $P_{\mathcal{M}} = I - P_{\mathcal{M}^\perp}$ *e, por consequência,* $P_{\mathcal{M}^\perp} = I - P_{\mathcal{M}}$.

Agora, podemos perguntar: é possível, dado um projetor qualquer, verificar se ele é ortogonal ou não? A resposta é sim, e a proposição a seguir traz algumas maneiras de fazer esta verificação.

Proposição 3. [9] *Suponha que $P \in \mathbb{C}^{n\times n}$ é um projetor, isto é, $P^2 = P$. Então, os seguintes itens são equivalentes para dizer que P é um projetor ortogonal:*

(i) $Ker(P) \perp Im(P)$;

(ii) $P^* = P$ *(ou seja, P é projetor ortogonal se, e somente se, $P^2 = P = P^*$);*

(iii) $\|P\|_2 = 1$, *em que* $\|\cdot\|_2$ *é chamada de* 2-norma matricial *e é dada por*

$$\|A\|_2 = \max_{x\in\mathbb{C}^n\setminus\{\mathbf{0}\}} \frac{\|Ax\|_2}{\|x\|_2} = \max_{\|x\|_2=1} \|Ax\|_2 = \sqrt{\lambda_{max}},$$

com $A \in \mathbb{C}^{m\times n}$ e λ_{max} o maior autovalor de A^*A.

Demonstração. Uma prova para este resultado pode ser encontrada em [9], p. 433. □

O resultado a seguir apresenta uma aplicação para as projeções ortogonais, que as tornam importantes na prática: o fato da projeção minimizar a distância entre um vetor e um espaço. Mais à frente, veremos essa propriedade em ação, já que o produto $AA^\dagger$ (em que $A^\dagger$ é a pseudo-inversa de A) é o projetor ortogonal em $Im(A)$.

Teorema 6. [9] *Sejam $\mathcal{M}$ um subespaço de um espaço $\mathcal{V}$ com produto interno e b um vetor em $\mathcal{V}$. Assim, o ponto em $\mathcal{M}$ mais próximo de b é $p = P_{\mathcal{M}}b$, a projeção ortogonal de b em $\mathcal{M}$. Em outros termos,*

$$\min_{u\in\mathcal{M}} \|b - u\|_2 = \|b - P_{\mathcal{M}}b\|_2 =: dist(b, \mathcal{M}),$$

que é a chamada distância ortogonal *entre $\mathcal{M}$ e b.*

Demonstração. Se $p = P_{\mathcal{M}}b$, então $p - u \in \mathcal{M}$, para todo $u \in \mathcal{M}$. Por outro lado, $b - p = b - P_{\mathcal{M}}b = (I - P_{\mathcal{M}})b \in \mathcal{M}^\perp$. Portanto, $(p - u) \perp (b - p)$, de modo que o

Teorema de Pitágoras pode ser aplicado, pois dados vetores x e y tais que $x \perp y$, então $\|x+y\|^2 = \|x\|^2 + \|y\|^2$. Logo,

$$\|b-u\|_2^2 = \|b-p+p-u\|_2^2 = \|b-p\|_2^2 + \|p-u\|_2^2 \geq \|p-u\|_2^2.$$

Resumindo, $\min_{u\in\mathcal{M}} \|b-u\|_2 = \|b-p\|_2$. Portanto, p é ponto em $\mathcal{M}$ de menor distância. Falta mostrar a unicidade. Para isso, considere que exista um outro ponto $q \in \mathcal{M}$ tal que $\|b-p\|_2 = \|b-q\|_2$, ao mesmo tempo considerando que não existe em $\mathcal{M}$ nenhum outro ponto mais próximo de b que p. Assim, novamente utilizando o Teorema de Pitágoras,

$$\|b-q\|_2^2 = \|b-p+p-q\|_2^2 = \|b-p\|_2^2 + \|p-q\|_2^2 = \|b-q\|_2^2 + \|p-q\|_2^2 \Longrightarrow \|p-q\|_2^2 = 0.$$

Logo, temos que $p = q$, garantindo a unicidade. □

2 A Matriz Pseudo-Inversa

Em diversos problemas da álgebra linear, é necessário o uso da matriz inversa. Mas não é toda matriz que possui uma inversa, visto que algumas condições precisam ser verdadeiras. Entre estas condições, está o fato de a matriz ser necessariamente quadrada, o que restringe a uma classe menor das matrizes em $\mathbb{C}^{m\times n}$ gerais. Mesmo assim, não é toda matriz quadrada que possui inversa.

A partir disto, podemos perguntar: existe alguma maneira de construir uma matriz que "imite" a inversa, mas que possa ser definida para qualquer matriz? A resposta é positiva, e foi conseguida com a definição da matriz pseudo-inversa.

Neste capítulo, veremos as propriedades teóricas principais da pseudo-inversa, bem como algumas de suas aplicações na busca por soluções para sistemas lineares.

2.1 Definição e Propriedades Básicas

Definição 4. *Sejam $A \in \mathbb{C}^{m\times n}$ e $X \in \mathbb{C}^{n\times m}$. Considere as seguintes condições, conhecidas como* condições de Penrose *(ou* equações de Penrose*):*

(I) $AXA = A$;

(II) $XAX = X$;

(III) $(AX)^* = AX$;

(IV) $(XA)^* = XA$.

Se X satisfaz essas quatro propriedades, então X é conhecida como a inversa de Moore-Penrose*, ou simplesmente* pseudo-inversa*, e é denotada por $A^\dagger$.*

A tentativa de generalização para o conceito de inversas para matrizes iniciou-se em 1920, quando Moore definiu uma inversa generalizada e provou sua unicidade (veja [10]). Mas foi apenas em 1955 que a ideia foi melhor formulada, quando Penrose (em [12]) mostrou que toda matriz A possuía uma única matriz X que satisfizesse as condições (I)-(IV) acima definidas.

Foi verificado que uma matriz X que satisfaz as equações de Penrose também coincide com a definição de inversa generalizada de Moore e, por este motivo, a matriz $X := A^\dagger$ ficou conhecida como inversa de Moore-Penrose, em homenagem aos principais matemáticos envolvidos na sua construção.

Observação 2. Vale dizer que existem casos em que X satisfaz apenas algumas das equações de Penrose, mas não todas. Assim, dizemos que X é *inversa*-$\{i, j, k\}$ de A se

satisfaz a i-ésima, a j-ésima e a k-ésima equações de Penrose. Por exemplo, se X é inversa-$\{1\}$ de A, então X satisfaz somente a equação (I), ou seja, $AXA = A$. Da mesma maneira, se X é inversa-$\{1, 2\}$, então satisfaz (I) e (II), isto é, $AXA = A$ e $XAX = X$, e assim por diante.

Existem várias maneiras de computar a pseudo-inversa, e uma destas possibilidades é usando a SVD da matriz A, que já desenvolvemos no Capítulo 1. Deste modo, podemos utilizar a SVD para construir um candidato à pseudo-inversa. Assim, temos $A = U\Sigma V^*$, em que $U \in \mathbb{C}^{m\times m}$ e $V \in \mathbb{C}^{n\times n}$ são matrizes unitárias, V^* é a matriz adjunta de V e, para $\sigma_i = \sqrt{\lambda_i}$, $\forall i = 1, \ldots, p$, com $\lambda_1, \ldots, \lambda_p$ os autovalores não nulos de A^*A,

$$\Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \sigma_p & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right] \in \mathbb{C}^{m\times n}.$$

Desta maneira, afirmamos que a pseudo-inversa é dada por $A^\dagger = V\Sigma^\dagger U^*$, em que

$$\Sigma^\dagger := \left[\begin{array}{ccc|c} \frac{1}{\sigma_1} & & & \\ & \ddots & & \mathbf{0} \\ & & \frac{1}{\sigma_p} & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right] \in \mathbb{C}^{n\times m}.$$

Como visto no Lema 1, todos os autovalores λ_i de A^*A são tais que $\lambda_i \geq 0$. Assim, como escolhemos σ_i, $i = 1, \ldots, p$, tais que $\sigma_i > 0$, os possíveis problemas com relação à definição das matrizes Σ e $\Sigma^\dagger$ estão removidos.

Agora, vejamos que $A^\dagger$ assim definida satisfaz as equações de Penrose. Para tanto, verifiquemos inicialmente que $\Sigma^\dagger$ é pseudo-inversa de Σ, ou seja, que

$$\begin{aligned} \Sigma\Sigma^\dagger\Sigma &= \Sigma, \\ \Sigma^\dagger\Sigma\Sigma^\dagger &= \Sigma^\dagger, \\ (\Sigma\Sigma^\dagger)^* &= \Sigma\Sigma^\dagger, \text{ e} \\ (\Sigma^\dagger\Sigma)^* &= \Sigma^\dagger\Sigma. \end{aligned}$$

Assim, fazendo uso da definição das matrizes Σ e $\Sigma^\dagger$, vejamos que, de fato, uma é pseudo-inversa da outra. Antes, observe que podemos escrever

$$\Sigma = \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{m\times n} \text{ e } \Sigma^\dagger = \begin{bmatrix} C^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{n\times m},$$

em que C é tal que

$$C = \begin{bmatrix} \sigma_1 & & 0 \\ & \ddots & \\ 0 & & \sigma_p \end{bmatrix} \in \mathbb{C}^{p\times p}.$$

Claramente, C é inversível, pois os valores singulares $\sigma_1, \ldots, \sigma_p$ são reais positivos. Também é interessante observar que $p \leq \min\{m, n\}$, o que evita outros problemas relacionados aos produtos, como faremos a seguir. Além disso,

$$C^{-1} = \begin{bmatrix} \frac{1}{\sigma_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sigma_p} \end{bmatrix} \in \mathbb{C}^{p\times p},$$

o que torna as definições precisas. Assim, perceba que

$$\Sigma\Sigma^\dagger = \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} C^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{m\times m}.$$

Analogamente,

$$\Sigma^\dagger\Sigma = \begin{bmatrix} C^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

Tendo isto, podemos finalmente partir para as verificações finais. Logo,

$$\Sigma\Sigma^\dagger\Sigma = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{m\times m} \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{m\times n} = \begin{bmatrix} C & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{m\times n} = \Sigma$$

e

$$\Sigma^\dagger\Sigma\Sigma^\dagger = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{n\times n} \begin{bmatrix} C^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{n\times m} = \begin{bmatrix} C^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{n\times m} = \Sigma^\dagger.$$

Os subíndices nas matrizes representam as dimensões das mesmas, afim de facilitar a compreensão dos produtos efetuados. Por último, veja que

$$(\Sigma\Sigma^\dagger)^* = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}^*_{m\times m} = \begin{bmatrix} I_p & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}_{m\times m} = \Sigma\Sigma^\dagger.$$

Exatamente pelo mesmo raciocínio segue que $(\Sigma^\dagger\Sigma)^* = \Sigma^\dagger\Sigma$. Assim, realmente temos que $\Sigma^\dagger$ satisfaz as equações de Penrose para Σ, o que a caracteriza como pseudo-inversa de Σ.

Finalmente, vejamos que $X := V\Sigma^\dagger U^*$ satisfaz as equações de Penrose para $A = U\Sigma V^*$. Logo:

(I) $AXA = U\Sigma V^*V\Sigma^\dagger U^*U\Sigma V^* = U\Sigma\Sigma^\dagger\Sigma V^* = U\Sigma V^* = A$;

(II) $XAX = V\Sigma^\dagger U^*U\Sigma V^*V\Sigma^\dagger U^* = V\Sigma^\dagger\Sigma\Sigma^\dagger U^* = V\Sigma^\dagger U^* = X$;

(III) $(AX)^* = (U\Sigma V^*V\Sigma^\dagger U^*)^* = (U\Sigma\Sigma^\dagger U^*)^* = U(\Sigma\Sigma^\dagger)^*U^* = U\Sigma\Sigma^\dagger U^* = U\Sigma V^*V\Sigma^\dagger U^* = AX$;

(IV) $(XA)^* = (V\Sigma^\dagger U^*U\Sigma V^*)^* = (V\Sigma^\dagger\Sigma V^*)^* = V(\Sigma^\dagger\Sigma)^*V^* = V\Sigma^\dagger\Sigma V^* = V\Sigma^\dagger U^*U\Sigma V^* = XA$.

Assim, X é pseudo-inversa de A, isto é, $X = A^\dagger$. Podemos, portanto, dizer que $A^\dagger = V\Sigma^\dagger U^*$.

Note que o raciocínio acima nada mais é do que a demonstração da existência da matriz pseudo-inversa para uma matriz $A \in \mathbb{C}^{m\times n}$ qualquer. O teorema abaixo afirma isso, além de também verificar a unicidade de $A^\dagger$.

Teorema 7. *Seja $A \in \mathbb{C}^{m\times n}$. Então $A^\dagger \in \mathbb{C}^{n\times m}$ existe e é única.*

Demonstração. **Unicidade:** sejam B e C duas matrizes pseudo-inversas de A. Em particular, elas satisfazem as condições de Penrose. Assim,

$$AB = (AB)^* = B^*A^* = B^*(ACA)^* = B^*A^*C^*A^* = (AB)^*(AC)^* = ABAC = AC.$$

Analogamente, temos $BA = CA$. Logo,

$$B = BAB = BAC = CAC = C.$$

Com isto temos $B = C$ e, portanto, a pseudo-inversa é única.
Existência: se $A \in \mathbb{C}^{m\times n}$ é uma matriz qualquer, então, pelo teorema da SVD, sempre existem as matrizes U, V e Σ como definidas acima, de modo que $A = U\Sigma V^*$. Assim, definimos $X := V\Sigma^\dagger U^* \in \mathbb{C}^{n\times m}$ como matriz candidata a ser a pseudo-inversa, em que $\Sigma^\dagger$ é como definida acima. O desenvolvimento exatamente anterior a este teorema verifica que de fato X assim definida satisfaz as equações de Penrose, ou seja, X é pseudo-inversa de A. □

O teorema a seguir apresenta diversas propriedades interessantes de $A^\dagger$, algumas necessárias na teoria a ser desenvolvida mais a frente.

Teorema 8. *Seja $A \in \mathbb{C}^{m\times n}$ e $A^\dagger \in \mathbb{C}^{n\times m}$ a sua pseudo-inversa. Então, são válidas as propriedades seguintes:*

(i) $(A^\dagger)^\dagger = A$;

(ii) *Se A é inversível, então* $A^\dagger = A^{-1}$;

(iii) $(A^\dagger)^T = (A^T)^\dagger$ *(comutatividade com transposição);*

(iv) $\overline{A^\dagger} = \overline{A}^\dagger$ *(comutatividade com conjugação);*

(v) $(A^\dagger)^* = (A^*)^\dagger$ *(comutatividade com adjunto);*

(vi) $(\alpha A)^\dagger = \alpha^\dagger A^\dagger$, *em que* $\alpha \in \mathbb{C}$ *e* $\alpha^\dagger = \begin{cases} \frac{1}{\alpha}, & \text{se } \alpha \neq 0 \\ 0, & \text{se } \alpha = 0 \end{cases}$;

(vii) $(AA^*)^\dagger = (A^*)^\dagger A^\dagger$ *e* $(A^*A)^\dagger = A^\dagger(A^*)^\dagger$;

(viii) $A^\dagger = (A^*A)^\dagger A^* = A^*(AA^*)^\dagger$;

(ix) $A^* = A^*AA^\dagger = A^\dagger AA^*$*;*

(x) Se $posto(A) = n$*, então* $A^\dagger A = I_n$ *e se* $posto(A) = m$*, então* $AA^\dagger = I_m$*.*

Demonstração. Considere a SVD de A como vista acima, isto é, $A = U\Sigma V^*$. Logo, $A^\dagger = V\Sigma^\dagger U^*$. Agora, vamos aos itens:

(i) $(A^\dagger)^\dagger = (V\Sigma^\dagger U^*)^\dagger = U(\Sigma^\dagger)^\dagger V^* = U\Sigma V^* = A$;

(ii) Como A é inversível, então $m = n$ e $posto(A) = m$. Assim, $\Sigma \in \mathbb{C}^{m\times m}$ é matriz diagonal sem zeros na diagonal principal, logo inversível. Assim,

$$\begin{aligned} AA^\dagger &= U\Sigma V^*V\Sigma^\dagger U^* = U\Sigma\Sigma^\dagger U^* = \\ &= U\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_m \end{bmatrix}\begin{bmatrix} \frac{1}{\sigma_1} & & \\ & \ddots & \\ & & \frac{1}{\sigma_m} \end{bmatrix} U^* = UU^* = I_m \end{aligned}$$

em que I_m é a matriz identidade $m \times m$. Logo, $A^\dagger = A^{-1}$.

(iii) $(A^T)^\dagger = ((V^*)^T\Sigma^T U^T)^\dagger = ((V^T)^*\Sigma^T U^T)^\dagger = (U^T)^*(\Sigma^T)^\dagger V^T = (U^*)^T(\Sigma^\dagger)^T V^T = (V\Sigma^\dagger U^*)^T = (A^\dagger)^T$;

(iv) $\overline{A}^\dagger = (\overline{U}\,\overline{\Sigma}\,\overline{V}^*)^\dagger = \overline{V}\Sigma^\dagger\overline{U}^* = \overline{V}\,\overline{\Sigma}^\dagger\overline{U^*} = \overline{(V\Sigma^\dagger U^*)} = \overline{A^\dagger}$;

(v) $(A^\dagger)^* = (V\Sigma^\dagger U^*)^* = U(\Sigma^\dagger)^* V^* = U(\Sigma^*)^\dagger V^* = (V\Sigma^* U^*)^\dagger = (A^*)^\dagger$;

(vi) Para o caso em que $\alpha = 0$, é trivial. Seja $\alpha \neq 0$. Assim,

$$\begin{aligned} (\alpha A)^\dagger &= (\alpha U\Sigma V^*)^\dagger = (U(\alpha\Sigma)V^*)^\dagger = \\ &= V\begin{bmatrix} \alpha\sigma_1 & & & \\ & \ddots & & \\ & & \alpha\sigma_p & \\ & & & \mathbf{0} \end{bmatrix}^\dagger U^* = V\begin{bmatrix} \frac{1}{\alpha\sigma_1} & & & \\ & \ddots & & \\ & & \frac{1}{\alpha\sigma_p} & \\ & & & \mathbf{0} \end{bmatrix} U^* = \\ &= V(\alpha^{-1}\Sigma^\dagger)U^* = \alpha^{-1}V\Sigma^\dagger U^* = \alpha^{-1}A^\dagger = \alpha^\dagger A^\dagger; \end{aligned}$$

(vii) Utilizando as equações de Penrose para A e o item (v) deste teorema, verifiquemos que valem as equações de Penrose para AA^*:

(I) $AA^*(A^*)^\dagger A^\dagger AA^* = AA^*(A^*)^\dagger(A^\dagger A)^* A^* = AA^*(A^*)^\dagger A^*(A^*)^\dagger A^* = AA^*(A^*)^\dagger A^* = AA^*$;

(II) $(A^*)^\dagger A^\dagger AA^*(A^*)^\dagger A^\dagger = (A^*)^\dagger A^\dagger A(A^\dagger A)^* A^\dagger = (A^*)^\dagger A^\dagger AA^\dagger AA^\dagger = (A^*)^\dagger A^\dagger AA^\dagger = (A^*)^\dagger A^\dagger$;

(III) $(AA^*(A^*)^\dagger A^\dagger)^* = (A^\dagger)^*((A^*)^\dagger)^* AA^* = (A^\dagger)^* A^\dagger AA^* = (A^\dagger)^* A^*(A^\dagger)^* A^* = (A^\dagger)^* A^* = (AA^\dagger)^* = AA^\dagger = AA^\dagger AA^\dagger = A(A^\dagger A)^* A^\dagger = AA^*(A^*)^\dagger A^\dagger$;

(IV) $((A^*)^\dagger A^\dagger AA^*)^* = AA^*(A^\dagger)^*((A^*)^\dagger)^* = AA^*(A^\dagger)^*A^\dagger = A(A^\dagger A)^*A^\dagger =$
$AA^\dagger AA^\dagger = AA^\dagger = (AA^\dagger)^* = (A^\dagger)^*A^* = (A^*)^\dagger A^* =$
$(A^*)^\dagger A^*(A^*)^\dagger A^* = (A^*)^\dagger (A^\dagger A)^* A^* = (A^*)^\dagger A^\dagger AA^*.$

O caso $(A^*A)^\dagger = A^\dagger(A^*)^\dagger$ segue pelo mesmo raciocínio.

(viii) Utilizando as equações de Penrose e os itens (v) e (vii), temos:

$$A^\dagger = A^\dagger AA^\dagger = A^\dagger(AA^\dagger)^* = A^\dagger(A^*)^\dagger A^* = (A^*A)^\dagger A^*.$$

Por outro lado,

$$A^\dagger = A^\dagger AA^\dagger = (A^\dagger A)^* A^\dagger = A^*(A^*)^\dagger A^\dagger = A^*(AA^*)^\dagger.$$

(ix) Utilizando as equações de Penrose e os itens (v) e (vii), temos:

$$A^* = A^*(A^*)^\dagger A^* = A^*(A^\dagger)^* A^* = A^*(AA^\dagger)^* = A^*AA^\dagger$$

e

$$A^* = A^*(A^*)^\dagger A^* = A^*(A^\dagger)^* A^* = (A^\dagger A)^* A^* = A^\dagger AA^*.$$

(x) Se $posto(A) = n$, então $posto(A^*A) = n$ e, junto com o fato de $A^*A \in \mathbb{C}^{n\times n}$, temos que A^*A é não singular. Agora, utilizando os itens (ii) e (viii), temos

$$A^\dagger A = (A^*A)^\dagger A^* A = (A^*A)^{-1}A^*A = I_n.$$

Da mesma maneira, se $posto(A) = m$, então $posto(AA^*) = m$ e, junto com o fato de $AA^* \in \mathbb{C}^{m\times m}$, temos que AA^* é não singular. Pelos itens (ii) e (viii), temos

$$AA^\dagger = AA^*(AA^*)^\dagger = AA^*(AA^*)^{-1} = I_m.$$

□

Proposição 4. *Seja $A \in \mathbb{C}^{m\times n}$ e $A^\dagger \in \mathbb{C}^{n\times m}$ a sua pseudo-inversa. Então:*

(i) $Ker(A^\dagger) = Ker(A^*)$;

(ii) $Im(A^\dagger) = Im(A^*)$.

Demonstração. (i) Defina $P := AA^\dagger$. Note que pelas equações de Penrose

$$P^2 = AA^\dagger AA^\dagger = AA^\dagger = P \text{ e } P = AA^\dagger = (AA^\dagger)^* = P^*.$$

Assim, pela Proposição 3 (item (ii)), temos que P é um projetor ortogonal.
Vejamos que $Im(P) = Im(A)$. Se $y \in Im(A)$, então existe x tal que $Ax = y$ e, portanto,

$$Py = PAx = AA^\dagger Ax = Ax = y,$$

o que implica que $y \in Im(P)$. Reciprocamente, se $y \in Im(P)$, então $Py = y$. Deste modo,

$$y = Py = AA^{\dagger}y = A(A^{\dagger}y) \in Im(A).$$

Logo, temos que $Im(P) = Im(A)$.
Note que, pela Proposição 1, por P ser projetor ortogonal em $Im(A)$, então $I - P$ é projetor ortogonal em $Im(A)^{\perp}$. Assim, $Im(I - P) = Im(A)^{\perp}$. Por outro lado, o Teorema 2 nos garante que $Im(A)^{\perp} = Ker(A^*)$. Portanto,

$$Im(I - P) = Ker(A^*). \tag{2.1}$$

Vejamos que $Im(P) = Im((A^{\dagger})^*)$. Se $y \in Im((A^{\dagger})^*)$, então existe x tal que $(A^{\dagger})^*x = y$. Perceba que $P(A^{\dagger})^* = P^*(A^{\dagger})^* = (A^{\dagger}P)^* = (A^{\dagger})^*$ e, assim,

$$Py = P(A^{\dagger})^*x = (A^{\dagger})^*x = y.$$

Logo, $y \in Im(P)$. Agora, se $y \in Im(P)$, então $Py = y$. Por $P = P^* = (A^{\dagger})^*A^*$, então

$$y = Py = (A^{\dagger})^*A^*y = (A^{\dagger})^*(A^*y) \in Im((A^{\dagger})^*).$$

Portanto, $Im(P) = Im((A^{\dagger})^*)$.
Pelo mesmo raciocínio acima, temos que

$$Im(I - P) = Im((A^{\dagger})^*)^{\perp} = Ker(((A^{\dagger})^*)^*) = Ker(A^{\dagger}). \tag{2.2}$$

Finalmente, juntando as equações (2.1) e (2.2), temos

$$Ker(A^{\dagger}) = Im(I - P) = Ker(A^*).$$

(ii) Defina $Q := A^{\dagger}A$. Pelas equações de Penrose, temos

$$Q^2 = A^{\dagger}AA^{\dagger}A = A^{\dagger}A = Q \text{ e } Q = A^{\dagger}A = (A^{\dagger}A)^* = Q^*,$$

o que caracteriza Q como projetor ortogonal, pela Proposição 3 (item (ii)).
Vejamos que $Im(Q) = Im(A^*)$. Se $y \in Im(A^*)$, então existe x tal que $A^*x = y$. Por outro lado, note que

$$QA^* = Q^*A^* = (A^{\dagger}A)^*A^* = A^*(A^{\dagger})^*A^* = (AA^{\dagger}A)^* = A^*.$$

Assim, temos que

$$Qy = QA^*x = A^*x = y,$$

ou seja, $y \in Im(Q)$. Reciprocamente, se $y \in Im(Q)$, então $Qy = y$. Deste modo,

$$y = Qy = Q^*y = (AA^{\dagger})^*y = A^*(A^{\dagger})^*y = A^*((A^{\dagger})^*y) \in Im(Q).$$

Portanto,

$$Im(Q) = Im(A^*). \tag{2.3}$$

Da mesma maneira, vejamos que $Im(Q) = Im(A^\dagger)$. Se $y \in Im(A^\dagger)$, então existe x tal que $A^\dagger x = y$. Logo,

$$Qy = QA^\dagger x = A^\dagger AA^\dagger x = A^\dagger x = y,$$

de modo que $y \in Im(Q)$. Agora, se $y \in Im(Q)$, então $Qy = y$. Assim,

$$y = Qy = A^\dagger Ay = A^\dagger(Ay) \in Im(A^\dagger).$$

Deste modo,

$$Im(Q) = Im(A^\dagger). \tag{2.4}$$

Finalmente, das equações (2.3) e (2.4), temos que

$$Im(A^\dagger) = Im(Q) = Im(A^*),$$

o que finaliza a demonstração. □

Proposição 5. *Sejam $A \in \mathbb{C}^{m\times n}$ e $B \in \mathbb{C}^{n\times m}$. Considere as seguintes condições:*

(i) A tem colunas ortogonais, ou seja, $A^*A = I$;

(ii) B tem linhas ortogonais, ou seja, $BB^* = I$;

(iii) A tem todas as colunas linearmente independentes e B tem todas as linhas linearmente independentes.

Se alguma das condições acima é válida, então $(AB)^\dagger = B^\dagger A^\dagger$.

Demonstração. Para cada um dos itens, a ideia é verificar se as condições de Penrose são válidas. Se sim, pelo fato da matriz pseudo-inversa ser única, temos os resultados. Para isso, considere $C := AB$.

(i) Como A tem colunas ortogonais, então $A^*A = I$, ou seja, $A^\dagger = A^*$. Seja $D := B^\dagger A^\dagger = B^\dagger A^*$. Verifiquemos que D satisfaz as condições de Penrose:

(I) $CDC = ABB^\dagger A^*AB = ABB^\dagger B = AB = C$;

(II) $DCD = B^\dagger A^*ABB^\dagger A^* = B^\dagger BB^\dagger A^* = B^\dagger A^* = D$;

(III) $(CD)^* = D^*B^*A^* = A(B^\dagger)^*B^*A^* = A(BB^\dagger)^*A^* = ABB^\dagger A^* = CD$;

(IV) $(DC)^* = B^*A^*D^* = B^*A^*A(B^\dagger)^* = (B^\dagger B)^* = B^\dagger B = B^\dagger A^*AB = DC$.

Logo, $D = C^\dagger$.

(ii) Como B tem linhas ortogonais, então $BB^* = I$, ou seja, $B^\dagger = B^*$. Seja $D := B^\dagger A^\dagger = B^*A^\dagger$. Verifiquemos que D satisfaz as condições de Penrose:

(I) $CDC = ABB^*A^\dagger AB = AA^\dagger AB = AB = C$;

(II) $DCD = B^*A^\dagger ABB^*A^\dagger = B^*A^\dagger AA^\dagger = B^*A^\dagger = D$;

(III) $(CD)^* = D^*B^*A^* = (A^\dagger)^*BB^*A^* = (A^\dagger)^*A^* = (AA^\dagger)^* = AA^\dagger = ABB^*A^\dagger = CD$;

(IV) $(DC)^* = B^*A^*D^* = B^*A^*(A^\dagger)^*B = B^*(A^\dagger A)^*B = B^*A^\dagger AB = DC$.

Logo, $D = C^\dagger$.

(iii) Como A tem todas as colunas linearmente independentes, A^*A é inversível e $(A^*A)^\dagger = (A^*A)^{-1}$. Analogamente, por B ter todas as linhas linearmente independentes, BB^* é inversível e $(BB^*)^\dagger = (BB^*)^{-1}$. Seja $D := B^\dagger A^\dagger = B^*(BB^*)^{-1}(A^*A)^{-1}A^*$. Verifiquemos que D satisfaz as condições de Penrose:

(I) $CDC = ABB^*(BB^*)^{-1}(A^*A)^{-1}A^*AB = AB = C$;

(II) $DCD = B^*(BB^*)^{-1}(A^*A)^{-1}A^*ABB^*(BB^*)^{-1}(A^*A)^{-1}A^* = B^*(BB^*)^{-1}(A^*A)^{-1}A^* = D$;

(III) $CD = ABB^*(BB^*)^{-1}(A^*A)^{-1}A^* = A(AA^*)^{-1}A^* = (A(AA^*)^{-1}A^*)^*$ $\Rightarrow (CD)^* = CD$;

(IV) $DC = B^*(BB^*)^{-1}(A^*A)^{-1}A^*AB = B^*(BB^*)^{-1}B = (B^*(BB^*)^{-1}B)^*$ $\Rightarrow (DC)^* = DC$.

Logo, $D = C^\dagger$.

□

O teorema a seguir não é diretamente necessário nesta parte da teoria, mas apresenta uma caracterização interessante para o uso do conceito de inversa generalizada. Na realidade, este resultado será utilizado mais a frente, na teoria de métodos iterativos para a pseudo-inversa de Moore-Penrose.

Teorema 9. [3] *Sejam $A \in \mathbb{C}^{m\times n}$, $B \in \mathbb{C}^{p\times q}$ e $D \in \mathbb{C}^{m\times q}$, então a equação matricial*

$$AXB = D$$

é consistente se, e somente se, para matrizes $A^{(1)}$ e $B^{(1)}$ inversas-$\{1\}$ de A e B (isto é, $AA^{(1)}A = A$ e $BB^{(1)}B = B$), respectivamente, vale $AA^{(1)}DB^{(1)}B = D$.
Neste caso, a solução geral da equação matricial é dada por

$$X = A^{(1)}DB^{(1)} + Y - A^{(1)}AYBB^{(1)},$$

para $Y \in \mathbb{C}^{n\times p}$ arbitrária.

Demonstração. ($\Rightarrow$) Supondo que a equação seja consistente, então existe X tal que $AXB = D$, e usando a definição das inversas generalizadas para A e B, temos

$$D = AXB = AA^{(1)}AXBB^{(1)}B = AA^{(1)}DB^{(1)}B.$$

($\Leftarrow$) Agora, supondo que vale $AA^{(1)}DB^{(1)}B = D$, então temos claramente que $X = A^{(1)}DB^{(1)}$ é uma solução da equação $AXB = D$, ou seja, a equação é consistente.

Assim, agora supondo que a equação é consistente, temos que se $X = A^{(1)}DB^{(1)} + Y - A^{(1)}AYBB^{(1)}$, para $Y \in \mathbb{C}^{n\times p}$ arbitrária, então

$$\begin{aligned} AXB &= A(A^{(1)}DB^{(1)} + Y - A^{(1)}AYBB^{(1)})B \\ &= AA^{(1)}DB^{(1)}B + AYB - AA^{(1)}AYBB^{(1)}B \\ &= AA^{(1)}DB^{(1)}B + AYB - AYB \\ &= AA^{(1)}DB^{(1)}B \\ &= D, \end{aligned}$$

ou seja, X é solução da equação. Por outro lado, se X é uma solução, então

$$X = A^{(1)}DB^{(1)} + X - A^{(1)}DB^{(1)} = A^{(1)}DB^{(1)} + X - A^{(1)}AXBB^{(1)},$$

que é da forma buscada. □

Podemos utilizar o teorema acima para dar uma caracterização para a consistência de um sistema linear, como pode ser visto no seguinte corolário.

Corolário 1. [3] *Sejam $A \in \mathbb{C}^{m\times n}$ e $b \in \mathbb{C}^m$. Então, o sistema de equações lineares $Ax = b$ é consistente se, e somente se, para alguma $A^{(1)}$ inversa-$\{1\}$ de A, tenhamos $AA^{(1)}b = b$.*
Neste caso, a solução geral do sistema é dado por $x = A^{(1)}b + (I - A^{(1)}A)y$, para $y \in \mathbb{C}^n$ arbitrário.

Demonstração. Segue diretamente do Teorema 9, tomando $B = I$ e $D = b$. □

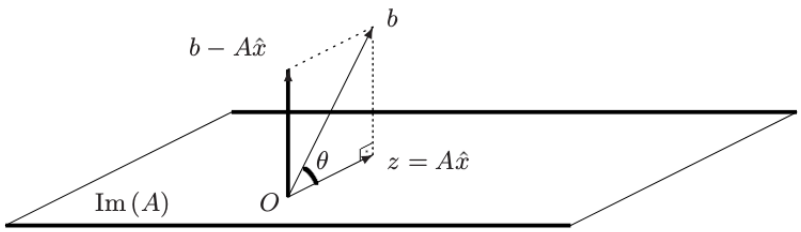
2.2 O Problema de Mínimos Quadrados

Dado um sistema linear $Ax = b$, nem sempre é possível encontrar uma solução no sentido clássico, ou seja, se $b \notin Im(A)$, não há $\hat{x}$ tal que $A\hat{x} = b$. Assim, a ideia agora é encontrar algum vetor $x \in \mathbb{C}^n$ que faça com que, de alguma maneira, Ax esteja o "mais próximo" possível do vetor b. Matematicamente, considerando $A \in \mathbb{C}^{m\times n}$ e $b \in \mathbb{C}^m$, estamos buscando $\hat{x} \in \mathbb{C}^n$ tal que

$$\|b - A\hat{x}\|_2 = \min_{y\in\mathbb{C}^n} \|b - Ay\|_2. \tag{2.5}$$

Utilizando o Teorema 6, o problema de mínimos quadrados (2.5) tem como interpretação geométrica a busca pelo projetor ortogonal de b em $Im(A)$. De fato, $A\hat{x}$ é o vetor de $Im(A)$ mais próximo de b. Além disso, sabemos que $b - A\hat{x}$ é perpendicular a $Im(A)$. A Figura 1 apresenta a visão geométrica do problema.

Figura 1 – O problema de mínimos quadrados: projeção de b em $Im(A)$.



Fonte: [1], p. 126.

Existem vários resultados clássicos da Álgebra Linear sobre o problema de mínimos quadrados. Entre eles, é notório que $x \in \mathbb{C}^n$ é uma solução de (2.5) se, e somente se, x satisfaz as chamadas *equações normais*, isto é,

$$A^*Ax = A^*b. \tag{2.6}$$

Para qualquer matriz $A \in \mathbb{C}^{m\times n}$, sempre existe ao menos uma solução para o sistema de equações normais (2.6). Além disso, esta solução é única se, e somente se, $Ker(A) = \{\mathbf{0}\}$.

Tendo estes conceitos e resultados em mente, vejamos agora um resultado que conecta diretamente o problema de mínimos quadrados com a matriz pseudo-inversa.

Proposição 6. [1] *O vetor $x_b := A^\dagger b$ é uma solução do problema de mínimos quadrados (2.5). Além disso, quando (2.5) admite mais de uma solução, então x_b é solução de norma mínima, ou seja, para todo $x \neq x_b$ tal que $\|Ax_b - b\|_2 = \|Ax - b\|_2$, temos que*

$$\|x_b\|_2 < \|x\|_2.$$

Demonstração. Note que, para todo $x \in \mathbb{C}^n$, podemos decompor $Ax - b$ da seguinte forma:

$$Ax - b = A(x - x_b) - (I - AA^\dagger)b.$$

Lembre que, na demonstração da Proposição 4, verificamos que $AA^\dagger$ é projetor ortogonal em $Im(A)$. Assim, a decomposição para $Ax - b$ acima é ortogonal, pois $A(x - x_b) \in Im(A)$ e $(I - AA^\dagger)b \in Im(A)^\perp$. Isto ocorre pelo fato de $AA^\dagger$ ser projetor ortogonal em $Im(A)$, logo $I - AA^\dagger$ é projetor ortogonal em $Im(A)^\perp$. Portanto, segue que

$$\langle A(x - x_b), (I - AA^\dagger)b\rangle = 0. \tag{2.7}$$

Deste modo,

$$\|Ax - b\|_2^2 = \|Ax - Ax_b\|_2^2 + \|b - Ax_b\|_2^2 \geq \|b - Ax_b\|_2^2,$$

o que caracteriza x_b como uma solução de (2.5), pois a desigualdade acima é válida para $x \in \mathbb{C}^n$ arbitrário.

Agora, para mostrar a segunda parte, considere $\|Ax_b - b\|_2 = \|Ax - b\|_2$. Pela equação (2.7), temos que $Ax = Ax_b$, o que implica que $z := x - x_b \in Ker(A)$. Assim, optamos por escrever $x = z + x_b$. Note que esta decomposição de x é ortogonal, pois $z \in Ker(A)$ e $x_b = A^\dagger b \in Ker(A)^\perp$, uma vez que $Im(A^\dagger) = Im(A^*) = Ker(A)^\perp$ (pela Proposição 4 e pelo Teorema 2). Logo,

$$\langle z, x_b \rangle = 0,$$

o que permite escrever

$$\|x\|_2^2 = \|z\|_2^2 + \|x_b\|_2^2 > \|x_b\|_2^2,$$

uma vez que estamos assumindo $x \neq x_b$, logo $\|z\|_2 = \|x - x_b\|_2 > 0$. □

O resultado acima apresenta uma das propriedades teóricas mais interessantes da pseudo-inversa, que certamente é utilizada na prática: dado uma sistema $Ax = b$, então $A^\dagger b$ é solução do problema de mínimos quadrados associado e, mais ainda, é a solução de norma mínima. Este fato torna importante e motiva o estudo de maneiras práticas para aplicar a pseudo-inversa na busca por soluções para $Ax = b$.

Como pontuação final desta seção, apresentamos o teorema a seguir, que formaliza todos os conceitos debatidos aqui e trata das equivalências entre o problema de mínimos quadrados, as equações normais e soluções com o uso da matriz pseudo-inversa.

Proposição 7. [9] *Considere o sistema inconsistente $Ax = b$ e o problema de mínimos quadrados associado a este sistema. Assim, cada um dos itens a seguir é equivalente a dizer que $\hat{x}$ é uma solução de mínimos quadrados para $Ax = b$.*

(i) $\|A\hat{x} - b\|_2 = \min_{x \in \mathbb{C}^n} \|Ax - b\|_2$;

(ii) $A\hat{x} = P_{Im(A)}b$;

(iii) $A^*A\hat{x} = A^*b$;

(iv) $\hat{x} \in A^\dagger b + Ker(A)$. *Além disso, $A^\dagger b$ é a solução de norma mínima para mínimos quadrados (no caso de 2-norma).*

Demonstração. Praticamente já discutimos todos os aspectos deste resultado nos comentários acima. Para uma demonstração completa, veja [9], p. 439. □

3 Métodos Iterativos para a Pseudo-Inversa

Apesar de extremamente rica do ponto de vista teórico, a matriz pseudo-inversa é, em geral, não utilizada na prática, visto que calculá-la pode ser demorado computacionalmente. De fato, é comparável ao custo de computar inversas de matrizes, algo que também não é recomendado.

Mesmo assim, ao longo dos anos e com o avanço da computação, diversos métodos foram desenvolvidos, visando novas abordagens na busca por calcular a pseudo-inversa de maneira mais barata e segura computacionalmente. Neste trabalho, abordaremos alguns dos métodos iterativos desenvolvidos, com o intuito de futuramente aplicá-los na construção de sequências vetoriais que convergem para a solução do problema de mínimos quadrados associado a $Ax = b$.

Por método iterativo, entendemos que se trata de um algoritmo que constrói uma sequência de matrizes $\{X_0, X_1, X_2, \ldots\}$, em que X_0 é uma aproximação inicial, e tal que, quando $k \longrightarrow \infty$, temos que $X_k \longrightarrow A^\dagger$.

O problema é certamente relacionado com a aproximação inicial X_0 e a maneira como é criada a sequência de matrizes X_k. Uma vez desenvolvida a ideia, a dificuldade se torna outra: determinar ordens de convergência e a aplicabilidade do método.

Aqui, veremos basicamente três tipos de métodos: um com convergência linear, outro que é uma família de métodos com convergência de ordem $p \geq 2$ e um último, que se baseia nas equações de Penrose. Destes, nos focaremos principalmente no segundo tipo, em especial para o caso $p = 2$, em que temos convergência quadrática.

Antes, precisamos de alguns resultados sobre normas matriciais, para podermos tratar da análise de convergência de alguns métodos.

Definição 5. *Uma função* $\|\cdot\| : \mathbb{C}^{m\times n} \longrightarrow \mathbb{R}$ *é uma* norma *no espaço vetorial* $\mathbb{C}^{m\times n}$ *se satisfaz as seguintes propriedades:*

(i) $\|A\| \geq 0, \forall A \in \mathbb{C}^{m\times n}$, *e* $\|A\| = 0 \Leftrightarrow A = \mathbf{0}$;

(ii) $\|\alpha A\| = |\alpha|\|A\|, \forall \alpha \in \mathbb{C}, \forall A \in \mathbb{C}^{m\times n}$;

(iii) $\|A + B\| \leq \|A\| + \|B\|, \forall A, B \in \mathbb{C}^{m\times n}$.

Além disso, se $\|\cdot\|$ *é norma e satisfaz* $\|AB\| \leq \|A\|\|B\|, \forall A, B \in \mathbb{C}^{m\times n}$, *então dizemos que* $\|\cdot\|$ *é uma* norma submultiplicativa.

Existem diversas normas matriciais, mas as mais comuns são as normas induzidas por normas vetoriais, que são da forma

$$\|A\|_p = \max_{x\in\mathbb{C}^n,\ x\neq\mathbf{0}} \frac{\|Ax\|_p}{\|x\|_p},$$

em que

$$\|y\|_p = \left(\sum_{i=1}^{n} |y_i|^p\right)^{\frac{1}{p}}, \text{ com } y \in \mathbb{C}^n \text{ e } 1 \leq p < \infty.$$

É possível mostrar que todas as normas induzidas são submultiplicativas. Outra norma clássica, e que também é submultiplicativa, é definida por

$$\|A\|_\infty = \max_{x \in \mathbb{C}^n,\ x \neq \mathbf{0}} \frac{\|Ax\|_\infty}{\|x\|_\infty}, \text{ em que } \|y\|_\infty = \max_{1 \leq i \leq n} |y_i|, y \in \mathbb{C}^n.$$

Agora, dada uma matriz $A \in \mathbb{C}^{n \times n}$, definimos o *raio espectral* de A, denotado por $\rho(A)$, da seguinte maneira:

$$\rho(A) = \max\{|\lambda| : \lambda \text{ é autovalor de } A\}.$$

Lema 2. [5] *Seja $A \in \mathbb{C}^{n \times n}$ e $\varepsilon > 0$. Então existe uma norma matricial induzida $\|\cdot\|_{A,\varepsilon}$, que depende de A e de ε, tal que $\|A\|_{A,\varepsilon} \leq \rho(A) + \varepsilon$.*

Demonstração. Seja $S^{-1}AS = J$ a forma de Jordan da matriz A e defina a matriz $D_\varepsilon \in \mathbb{C}^{n \times n}$ por $D_\varepsilon = \text{diag}(1, \varepsilon, \varepsilon^2, \ldots, \varepsilon^{n-1})$. Então, perceba que

$$(SD_\varepsilon)^{-1}A(SD_\varepsilon) = D_\varepsilon J D_\varepsilon = \left[\begin{array}{ccccc|ccc|c} \lambda_1 & \varepsilon & & & & & & & \\ & \ddots & \ddots & & & & & & \\ & & \ddots & \varepsilon & & & & & \\ & & & \lambda_1 & & & & & \\ \hline & & & & & \lambda_2 & \varepsilon & & \\ & & & & & & \ddots & \ddots & \\ & & & & & & & \ddots & \varepsilon \\ & & & & & & & & \lambda_2 \\ \hline & & & & & & & & \ddots \end{array}\right].$$

Assim, usando a norma vetorial $\|\cdot\|_{A,\varepsilon}$ definida por

$$\|x\|_{A,\varepsilon} = \|(SD_\varepsilon)^{-1}x\|_\infty,$$

e denotando $y = (SD_\varepsilon)^{-1}x$, geramos a norma matricial induzida por esta norma acima:

$$\begin{aligned} \|A\|_{A,\varepsilon} &= \max_{x \neq \mathbf{0}} \frac{\|Ax\|_{A,\varepsilon}}{\|x\|_{A,\varepsilon}} \\ &= \max_{x \neq \mathbf{0}} \frac{\|(SD_\varepsilon)^{-1}Ax\|_\infty}{\|(SD_\varepsilon)^{-1}x\|_\infty} \\ &= \max_{y \neq \mathbf{0}} \frac{\|(SD_\varepsilon)^{-1}ASD_\varepsilon y\|_\infty}{\|y\|_\infty} \\ &= \|(SD_\varepsilon)^{-1}ASD_\varepsilon\|_\infty \\ &= \max_i |\lambda_i| + \varepsilon \\ &= \rho(A) + \varepsilon. \end{aligned}$$

Logo, o resultado segue. □

Perceba que este resultado acima nos garante que, em geral, $\rho(A) \leq \|A\|$, para qualquer norma matricial.

Lema 3. [3] *Seja $A \in \mathbb{C}^{n\times n}$. Então,*

$$\lim_{k\to\infty} A^k = \mathbf{0} \Leftrightarrow \rho(A) < 1.$$

Demonstração. ($\Rightarrow$) Seja λ um autovalor de A e $x \neq \mathbf{0}$ um autovetor associado a este autovalor. Assim, $A^k x = \lambda^k x$. Portanto,

$$\mathbf{0} = \lim_{k\to\infty} A^k x = \lim_{k\to\infty} \lambda^k x.$$

Pelo fato de $x \neq \mathbf{0}$, temos que $\lim_{k\to\infty} \lambda^k = 0$, de onde segue que $|\lambda| < 1$. Como λ foi tomado arbitrário, temos que $\rho(A) < 1$.
($\Leftarrow$) Como $\rho(A) < 1$, então existe $\varepsilon > 0$ tal que $\rho(A) < 1 - \varepsilon$. Pelo Lema 2, existe uma norma matricial induzida $\|\cdot\|$ tal que $\|A\| \leq \rho(A) + \varepsilon < 1$.
Como essa norma é uma norma induzida, então satisfaz a propriedade submultiplicativa e, assim, temos que $\|A^k\| \leq \|A\|^k$. Agora, tomando o limite quando k tende ao infinito, temos

$$\lim_{k\to\infty} \|A^k\| \leq \lim_{k\to\infty} \|A\|^k = 0,$$

ou seja,

$$\lim_{k\to\infty} A^k = \mathbf{0}.$$

□

Agora já temos algumas ferramentas necessárias para tratar da convergência dos métodos. Portanto, vejamos, a partir de agora, alguns dos métodos iterativos para encontrar $A^\dagger$.

3.1 Métodos do Tipo $X_{k+1} = X_k + C_k(P_{Im(A)} - AX_k)$

Como visto na demonstração da Proposição 4, temos que

$$AA^\dagger = P_{Im(A)}$$

e, assim, definimos o *resíduo na iteração k* por $R_k = P_{Im(A)} - AX_k$. Obviamente, R_k converge a zero quando X_k converge para $A^\dagger$.

Definição 6. *Seja $\|\cdot\|$ alguma norma matricial submultiplicativa. Dizemos que um método iterativo para calcular $A^\dagger$ tal que $R_k \longrightarrow \mathbf{0}$ (ou, equivalentemente, $X_k \longrightarrow A^\dagger$):*

(i) converge linearmente *se existe* $r \in [0,1)$ *tal que*

$$\lim_{k\to\infty} \frac{\|R_{k+1}\|}{\|R_k\|} = r.$$

Caso $r = 0$*, dizemos que a convergência é* superlinear.

(ii) tem ordem de convergência p*, para* $p > 1$*, se existe uma constante* $M > 0$ *tal que*

$$\lim_{k\to\infty} \frac{\|R_{k+1}\|}{\|R_k\|^p} = M.$$

Definição 7. *Sejam* $A, B \in \mathbb{C}^{m\times n}$*. Chamamos de* imagem de (A,B) *e* núcleo de (A,B)*, respectivamente,*

$$Im(A,B) = \{Y = AXB \in \mathbb{C}^{m\times n} \mid X \in \mathbb{C}^{n\times m}\} \text{ e}$$
$$Ker(A,B) = \{X \in \mathbb{C}^{n\times m} \mid AXB = \mathbf{0}\}.$$

Definição 8. *Considere* $A = (a_{ij}) \in \mathbb{C}^{m\times n}$ *e* $B \in \mathbb{C}^{p\times q}$*. O* produto de Kronecker $A \otimes B$ *é a matriz* $mp \times nq$ *dada por*

$$A \otimes B = \begin{bmatrix} a_{11}B & a_{12}B & \cdots & a_{1n}B \\ a_{21}B & a_{22}B & \cdots & a_{2n}B \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1}B & a_{m2}B & \cdots & a_{mn}B \end{bmatrix}.$$

É possível verificar que o produto de Kronecker definido acima satisfaz diversas propriedades, entre elas as seguintes:

$$(A \otimes B)^* = A^* \otimes B^*,$$
$$(A \otimes B)^T = A^T \otimes B^T \text{ e}$$
$$(A \otimes B)(P \otimes Q) = AP \otimes BQ,$$

para matrizes A, B, P, Q com dimensões de modo que os produtos acima façam sentido.

Lema 4. [3] *Considere o produto interno em* $\mathbb{C}^{m\times n}$ *dado por*

$$\langle X, Y\rangle = \mathrm{tr}(Y^*X) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij}\overline{y_{ij}},$$

então, dados $A, B \in \mathbb{C}^{m\times n}$*, os conjuntos* $Im(A,B)$ *e* $Ker(A^*, B^*)$ *são subespaços ortogonais complementares de* $\mathbb{C}^{m\times n}$*.*

Demonstração. Considere uma transformação linear que leva uma matriz $X \in \mathbb{C}^{m\times n}$ em um vetor $\mathrm{vet}(X)$ dado por

$$\mathrm{vet}(X) = \mathrm{vet}\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{pmatrix}$$
$$= (x_{11}, x_{12}, \cdots, x_{1n}, x_{21}, x_{22}, \cdots, x_{2n}, \cdots, x_{m1}, x_{m2}, \cdots, x_{mn})^T.$$

Claramente, temos uma bijeção entre os espaços $\mathbb{C}^{m\times n}$ e $\mathbb{C}^{mn}$. Além disso, perceba que

$$\langle X, Y\rangle = \langle \text{vet}(X), \text{vet}(Y)\rangle = \text{vet}(Y)^*\text{vet}(X),$$

ou seja, no espaço $\mathbb{C}^{mn}$ esse produto interno corresponde ao produto interno euclidiano. Também, é fácil de verificar que $\text{vet}(AXB) = (A \otimes B^T)\text{vet}(X)$. Assim, deduzimos que $Im(A, B)$ e $Ker(A^*, B^*)$ correspondem a $Im(A\otimes B^T)$ e $Ker(A^*\otimes (B^*)^T)$, respectivamente. Pelas propriedades do produto de Kronecker vistas acima, segue que $Ker(A^* \otimes (B^*)^T)$ é o mesmo que $Ker((A \otimes B^T)^*)$, que é complemento ortogonal de $Im(A \otimes B^T)$ em $\mathbb{C}^{mn}$. Além disso, como $\text{vet} : \mathbb{C}^{m\times n} \to \mathbb{C}^{mn}$ é uma bijeção, então $Im(A, B)$ e $Ker(A^*, B^*)$ são subespaços ortogonais complementares de $\mathbb{C}^{m\times n}$. □

Agora, buscamos um método de modo que $X_{k+1} = X_k + C_kR_k$, em que C_k é alguma sequência de matrizes. Cada escolha dessa sequência origina métodos diferentes.

Claramente, não é prático calcular $P_{Im(A)}$; assim, buscamos C_k de modo que este problema possa ser contornado. Uma maneira de escolher C_k para que isto ocorra é impor $C_k = C_kP_{Im(A)}$. Assim,

$$C_kR_k = C_k(P_{Im(A)} - AX_k) = C_kP_{Im(A)} - C_kAX_k = C_k - C_kAX_k = C_k(I - AX_k).$$

Logo, $X_{k+1} = X_k + C_k(I - AX_k)$. A seguir, veremos algumas das escolhas de C_k possíveis, de modo que a condição $C_k = C_kP_{Im(A)}$ seja satisfeita.

Caso escolhemos $C_k = X_0$, para todo k, então obtemos um método iterativo de convergência linear para aproximar $A^\dagger$, como pode ser visto no teorema a seguir.

Teorema 10. [3] *Sejam $A \in \mathbb{C}^{m\times n}$, $X_0 \in \mathbb{C}^{n\times m}$ e $R_0 = P_{Im(A)} - AX_0$ tais que $X_0 \in Im(A^*, A^*)$ e $\rho(R_0) < 1$, então a sequência*

$$X_{k+1} = X_k + X_0(I - AX_k), \text{ para } k = 0, 1, 2, \ldots,$$

converge para $A^\dagger$. Além disso, a sequência de resíduos satisfaz

$$\|R_{k+1}\| \leq \|R_0\|\|R_k\|, \text{ com } k = 0, 1, 2, \ldots,$$

para alguma norma matricial submultiplicativa $\|\cdot\|$.

Demonstração. Primeiro vejamos que $X_0P_{Im(A)} = X_0$. De fato, como $X_0 \in Im(A^*, A^*)$, então existe uma matriz $B \in \mathbb{C}^{m\times n}$ tal que $X_0 = A^*BA^*$. Além disso, podemos escrever $P_{Im(A)} = AA^\dagger$, e pelo item (ix) do Teorema 8, temos que $A^*AA^\dagger = A^*$. Assim,

$$X_0P_{Im(A)} = X_0AA^\dagger = A^*BA^*AA^\dagger = A^*BA^* = X_0.$$

Portanto, podemos reescrever a iteração, que é $X_{k+1} = X_k + X_0(I - AX_k)$, para

$$X_{k+1} = X_k + X_0(P_{Im(A)} - AX_k) = X_k + X_0R_k,$$

de onde segue que

$$\begin{aligned} R_{k+1} &= P_{Im(A)} - AX_{k+1} \\ &= P_{Im(A)} - AX_k - AX_0R_k \\ &= P_{Im(A)} - AA^\dagger AX_k - AX_0R_k \\ &= P^2_{Im(A)} - P_{Im(A)}AX_k - AX_0R_k \\ &= P_{Im(A)}(P_{Im(A)} - AX_k) - AX_0R_k \\ &= P_{Im(A)}R_k - AX_0R_k \\ &= (P_{Im(A)} - AX_0)R_k \\ &= R_0R_k. \end{aligned}$$

Assim, considerando $\|\cdot\|$ alguma norma matricial submultiplicativa, temos que $\|R_{k+1}\| = \|R_0R_k\| \leq \|R_0\|\|R_k\|$. Além disso, como $R_k = R_0R_{k-1} = R_0^2R_{k-2} = \ldots = R_0^{k+1}$, então $\|R_k\| = \|R_0^{k+1}\| \leq \|R_0\|^{k+1}$, e tomando o limite quando $k \longrightarrow \infty$, segue da hipótese de que $\rho(R_0) < 1$ e do Lema 3 que

$$\lim_{k\to\infty} \|R_k\| = 0.$$

Portanto, $P_{Im(A)} - AX_k \longrightarrow \mathbf{0}$, e reescrevendo o método da forma

$$\begin{aligned} X_{k+1} &= X_k + X_0R_k \\ &= X_{k-1} + X_0R_{k-1} + X_0R_k \\ &= \ldots \\ &= X_0 + X_0R_0 + X_0R_1 + \ldots + X_0R_k \\ &= X_0 + X_0R_0 + X_0R_0^2 + \ldots + X_0R_0^{k+1} \\ &= X_0(I + R_0 + R_0^2 + \ldots + R_0^{k+1}), \end{aligned}$$

como $\rho(R_0) < 1$, do Lema 3 implica que X_k converge para algum limite X_∞, ou seja, temos que $AX_\infty = P_{Im(A)}$. Logo, $AX_\infty A = P_{Im(A)}A = AA^\dagger A = A$. Disto, segue que X_∞ é inversa generalizada de A, ou seja, satisfaz a equação (I) de Penrose: $AX_\infty A = A$.
Além disso, como $X_0 \in Im(A^*, A^*)$, e supondo que $X_k \in Im(A^*, A^*)$, podemos escrever $X_0 = A^*BA^*$ e $X_k = A^*CA^*$, para $B, C \in \mathbb{C}^{n\times m}$, o que implica em

$$\begin{aligned} X_{k+1} &= X_k + X_0(I - AX_k) \\ &= X_k + X_0 - X_0AX_k \\ &= A^*CA^* + A^*BA^* - A^*BA^*AA^*CA^* \\ &= A^*(C + B - BA^*AA^*C)A^*, \end{aligned}$$

de modo que $X_{k+1} \in Im(A^*, A^*)$. Segue por indução que $X_k \in Im(A^*, A^*)$, para $k = 0, 1, 2, \ldots$ e, portanto, $X_\infty \in Im(A^*, A^*)$.
Além disso, pelo Teorema 9 temos que a solução geral da equação $AXA = A$ é dada por

$$X = A^\dagger AA^\dagger + Y - A^\dagger AYAA^\dagger = A^\dagger + Y - A^\dagger AYAA^\dagger,$$

para $Y \in \mathbb{C}^{n\times m}$ arbitrária. Porém, note que $A^\dagger \in Im(A^*, A^*)$ e $(Y - A^\dagger AYAA^\dagger) \in Ker(A, A)$. De fato,

$$A^\dagger = A^\dagger AA^\dagger AA^\dagger = (A^\dagger A)^* A^\dagger (AA^\dagger)^* = A^*(A^\dagger)^* A^\dagger (A^\dagger)^* A^*$$

e

$$A(Y - A^\dagger AYAA^\dagger)A = AYA - AA^\dagger AYAA^\dagger A = AYA - AYA = \mathbf{0}.$$

Pelo Lema 4 temos que $Im(A^*, A^*)$ e $Ker(A, A)$ são subespaços complementares de $\mathbb{C}^{n\times m}$, o que implica que a representação $X = A^\dagger + Y - A^\dagger AYAA^\dagger$ é única. E como $X_\infty \in Im(A^*, A^*)$ e satisfaz $AX_\infty A = A$, segue que $X_\infty = A^\dagger$, o que finaliza a demonstração. □

O teorema acima garante que o método tem convergência linear, pois

$$\lim_{k\to\infty} \frac{\|R_{k+1}\|}{\|R_k\|} \leq \lim_{k\to\infty} \frac{\|R_0\|\|R_k\|}{\|R_k\|} = \|R_0\| < 1.$$

O último passo vem do fato de que, por hipótese, $\rho(R_0) < 1$ e do Lema 2.

Podemos também efetuar a seguinte escolha para C_k:

$$C_k = X_k(I + T_k + T_k^2 + \ldots + T_k^{p-2}),$$

em que $T_k = I - AX_k$ e $p \geq 2$ inteiro. O teorema seguinte diz que esta escolha resulta num método iterativo de ordem p.

Teorema 11. [3] *Sejam $A \in \mathbb{C}^{m\times n}$, $X_0 \in \mathbb{C}^{n\times m}$ e $R_0 = P_{Im(A)} - AX_0$ tais que $X_0 \in Im(A^*, A^*)$ e $\rho(R_0) < 1$, então para $p \geq 2$ inteiro, a sequência*

$$X_{k+1} = X_k + C_k T_k, \text{ para } k = 0, 1, 2, \ldots,$$

em que

$$C_k = X_k(I + T_k + T_k^2 + \ldots + T_k^{p-2})$$

e

$$T_k = (I - AX_k),$$

converge para $A^\dagger$. Além disso, a sequência de resíduos satisfaz

$$\|R_{k+1}\| \leq \|R_k\|^p, \text{ com } k = 0, 1, 2, \ldots,$$

para alguma norma matricial submultiplicativa $\|\cdot\|$.

Demonstração. Vejamos que $X_k \in Im(A^*, A^*)$, para todo $k = 0, 1, 2, \ldots$. De fato, procedendo por indução, temos que, por hipótese, $X_0 \in Im(A^*, A^*)$. Agora, supondo que $X_k \in Im(A^*, A^*)$, temos que $X_k T_k^i = X_k(I - AX_k)^i \in Im(A^*, A^*)$, para $i = 1, 2, \ldots, p-1$. Além disso,

$$\begin{aligned} X_{k+1} &= X_k + C_k T_k \\ &= X_k + X_k(I + T_k + T_k^2 + \ldots + T_k^{p-2})T_k \\ &= X_k + X_k T_k + X_k T_k^2 + \ldots + X_k T_k^{p-1}. \end{aligned}$$

Logo, $X_{k+1} \in Im(A^*, A^*)$. Deste modo, podemos escrever $X_k = A^*BA^*$ e $X_k T_k^i = A^* D_i A^*$, para $i = 1, 2, \ldots, p-1$, de onde temos

$$\begin{aligned} C_k P_{Im(A)} &= X_k(I + T_k + T_k^2 + \ldots + T_k^{p-2})AA^\dagger \\ &= X_k AA^\dagger + X_k T_k AA^\dagger + X_k T_k^2 AA^\dagger + \ldots + X_k T_k^{p-2} AA^\dagger \\ &= A^*BA^*AA^\dagger + A^*D_1A^*AA^\dagger + A^*D_2A^*AA^\dagger + \ldots + A^*D_{p-2}A^*AA^\dagger \\ &= A^*BA^* + A^*D_1A^* + A^*D_2A^* + \ldots + A^*D_{p-2}A^* \\ &= X_k + X_k T_k + X_k T_k^2 + \ldots + X_k T_k^{p-2} \\ &= X_k(I + T_k + T_k^2 + \ldots + T_k^{p-2}) \\ &= C_k, \ k = 0, 1, 2, \ldots, \end{aligned}$$

de modo que podemos reescrever o método para a seguinte forma: $X_{k+1} = X_k(I + R_k + R_k^2 + \ldots + R_k^{p-1})$. Disto, segue que

$$\begin{aligned} R_{k+1} &= P_{Im(A)} - AX_{k+1} \\ &= P_{Im(A)} - AX_k(I + R_k + R_k^2 + \ldots + R_k^{p-1}) \\ &= P_{Im(A)} - AX_k - AX_k(R_k + R_k^2 + \ldots + R_k^{p-1}) \\ &= R_k - AX_k(R_k + R_k^2 + \ldots + R_k^{p-1}). \end{aligned}$$

Utilizando propriedades da projeção, temos que $R_k = P_{Im(A)} - AX_k = P_{Im(A)} - P_{Im(A)}AX_k = P_{Im(A)}(P_{Im(A)} - AX_k) = P_{Im(A)}R_k$, de onde segue que, para $i = 1, 2, \ldots, p-1$,

$$\begin{aligned} R_k^i - AX_kR_k^i &= R_kR_k^{i-1} - AX_kR_k^i \\ &= P_{Im(A)}R_kR_k^{i-1} - AX_kR_k^i \\ &= P_{Im(A)}R_k^i - AX_kR_k^i \\ &= (P_{Im(A)} - AX_k)R_k^i \\ &= R_kR_k^i \\ &= R_k^{i+1}. \end{aligned}$$

Assim, juntado estes resultados, temos

$$\begin{aligned} R_{k+1} &= R_k - AX_k(R_k + R_k^2 + \ldots + R_k^{p-1}) \\ &= R_k - AX_kR_k - AX_k(R_k^2 + \ldots + R_k^{p-1}) \\ &= R_k^2 - AX_k(R_k^2 + \ldots + R_k^{p-1}) \\ &= \ldots \\ &= R_k^{p-1} - AX_kR_k^{p-1} \\ &= R_k^p. \end{aligned}$$

Assim, considerando agora uma norma submultiplicativa $\|\cdot\|$, temos $\|R_{k+1}\| = \|R_k^p\| \leq \|R_k\|^p$. Além disso, $R_k = R_{k-1}^p = R_{k-2}^{p^2} = \ldots = R_0^{p^k}$, pelo que foi visto acima. Deste modo,

juntando isto com o fato de $\rho(R_0) < 1$, pelo Lema 3, temos que

$$\lim_{k\to\infty} \|R_k\| = 0,$$

ou seja, $P_{Im(A)} - AX_k \longrightarrow 0$, e de $\rho(R_0) < 1$ e $R_{k+1} = R_k^p$, podemos concluir que $X_k \longrightarrow X_\infty$, a qual satisfaz $AX_\infty = P_{Im(A)}$, e portanto é solução de $AXA = A$. Como $X_k \in Im(A^*, A^*)$ para $k = 0, 1, 2, \ldots$, segue que $X_\infty \in Im(A^*, A^*)$ e procedendo da mesma forma que na demonstração do Teorema 10 concluímos que $X_\infty = A^\dagger$, o que conclui a demonstração. □

O teorema nos garante que o método tem convergência de ordem p:

$$\lim_{k\to\infty} \frac{\|R_{k+1}\|}{\|R_k\|^p} \leq \lim_{k\to\infty} \frac{\|R_k\|^p}{\|R_k\|^p} = 1.$$

Por este resultado, vemos que é possível construir métodos iterativos com a taxa de convergência desejada, porém recaímos no problema de eficiência do método, já que quanto maior a taxa de convergência, mais cara se torna cada iteração. De fato, na prática, como é mostrado em [3], o método mais vantajoso dentro desta classe é o caso em que $p = 2$.

Perceba que, nos teoremas acima, pedimos que $X_0 \in Im(A^*, A^*)$ e $\rho(R_0) < 1$. Logo, precisamos escolher X_0 corretamente, de modo às condições serem satisfeitas. Em geral, utilizamos $X_0 = \beta A^*$, para $\beta \in \mathbb{R}$, e, assim, $X_0 \in Im(A^*, A^*)$, pois $\beta A^* = \beta(AA^\dagger A)^* = A^*\beta(A^\dagger)^* A^*$.

No resultado a seguir, vemos como escolher β para que a outra condição seja satisfeita, isto é, $\rho(R_0) < 1$.

Teorema 12. [3] *Sejam $\mathbf{0} \neq A \in \mathbb{C}^{m\times n}$ com posto$(A) = r$, $\beta \in \mathbb{R}$ e $R_0 = P_{Im(A)} - \beta AA^*$. Então, $\rho(R_0) < 1$ se, e somente se,*

$$0 < \beta < \frac{2}{\rho(AA^*)}.$$

Demonstração. Como AA^* é hermitiana, então, pelo mesmo raciocínio do Lema 1, seus autovalores são reais e não negativos. Sejam $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > \lambda_{r+1} = \ldots = \lambda_m = 0$ os autovalores de AA^*. Assim, os autovalores não negativos de R_0 são da forma $\mu_i = 1 - \beta\lambda_i$, para $i = 1, \ldots, r$.

Agora, vamos às implicações:

($\Rightarrow$) Supondo $\rho(R_0) < 1$, temos que $-1 < \mu_i < 1$, para $i = 1, \ldots, r$. Como $\mu_i = 1 - \beta\lambda_i$,

$i = 1, \ldots, r$, temos

$$\begin{aligned} -1 < 1 - \beta\lambda_i < 1 \\ -2 < -\beta\lambda_i < 0 \\ 0 < \beta\lambda_i < 2 \\ 0 < \beta < \frac{2}{\lambda_i} \\ 0 < \beta < \frac{2}{\lambda_1} \\ 0 < \beta < \frac{2}{\rho(AA^*)}. \end{aligned}$$

($\Leftarrow$) Como os autovalores não negativos de R_0 são dados por $\mu_i = 1 - \beta\lambda_i$, e da hipótese de $\beta > 0$, temos

$$\mu_i = 1 - \beta\lambda_i < 1 - 0 = 1.$$

Por outro lado, de $\beta < \frac{2}{\rho(AA^*)} = \frac{2}{\lambda_1}$, temos

$$\mu_i > 1 - \frac{2}{\lambda_1}\lambda_i \geq 1 - \frac{2}{\lambda_i}\lambda_i = 1 - 2 = -1.$$

Portanto, temos $-1 < \mu_i < 1$, para $i = 1, \ldots, r$, do que segue que $\rho(R_0) < 1$. □

Agora, como um apanhado geral do que foi desenvolvido acima, temos, basicamente, dois métodos iterativos:

(i) O método com convergência linear dado por

$$X_{k+1} = X_k + X_0(I - AX_k), \text{ para } k = 0, 1, 2, \ldots. \tag{3.1}$$

(ii) $X_{k+1} = X_k + C_kT_k$, para $k = 0, 1, 2, \ldots$, em que $C_k = X_k(I + T_k + T_k^2 + \ldots + T_k^{p-2})$ e $T_k = (I - AX_k)$, para algum $p \geq 2$ inteiro escolhido. Este método tem convergência de ordem p. Podemos reescrevê-lo utilizando C_k e T_k. Assim,

$$X_{k+1} = X_k + X_k(I + (I - AX_k) + (I - AX_k)^2 + \ldots + (I - AX_k)^{p-2})(I - AX_k). \tag{3.2}$$

Perceba que (3.2) pode ser, ainda, escrito como

$$\begin{aligned} X_{k+1} &= X_k(I + (I - AX_k) + (I - AX_k)^2 + \ldots + (I - AX_k)^{p-1}) \\ &= X_k(I + T_k + T_k^2 + \ldots + T_k^{p-1}). \end{aligned} \tag{3.3}$$

Como observado acima, o caso mais vantajoso é quando $p = 2$, que nos gera um método da forma

$$X_{k+1} = X_k + X_k(I - AX_k) = X_k(2I - AX_k). \tag{3.4}$$

Estes dois métodos convergem para $A^\dagger$ sob as seguintes condições: $X_0 \in Im(A^*, A^*)$ e $\rho(R_0) < 1$, em que $R_0 = P_{Im(A)} - AX_0$. O Teorema 12 nos diz como fazer uma escolha acertada de X_0 de modo que estas duas condições sejam satisfeitas. Assim, seja $X_0 := \beta A^*$, em que

$$0 < \beta < \frac{2}{\rho(AA^*)}.$$

O problema agora é calcular $\rho(AA^*)$, que pode ser caro. Vejamos como contornar isto.

Definição 9. *Seja $A \in \mathbb{C}^{m\times n}$. Definimos a* norma de Frobenius*, denotada por $\|\cdot\|_F$, por*

$$\|A\|_F = \left(\sum_{i=1}^{m}\sum_{j=1}^{n} |a_{ij}|^2\right)^{1/2}.$$

De fato, $\|\cdot\|_F$ é uma norma. Além disso, ela possui outra propriedade interessante, que a torna útil aqui: dada $A \in \mathbb{C}^{m\times n}$, então $\|A\|_2 \leq \|A\|_F$, em que $\|A\|_2 = \rho(A)$. Assim,

$$\rho(AA^*) = \|AA^*\|_2 \leq \|A\|_2\|A^*\|_2 = \|A\|_2^2 \leq \|A\|_F^2,$$

ou seja, com isso, podemos estimar β de outra maneira:

$$0 < \beta \leq \frac{1}{\|A\|_F^2} < \frac{2}{\rho(AA^*)}.$$

Note que nada impedia uma escolha de β diretamente a partir de sua condição, ou seja, calculando $\rho(AA^*)$. O que deve ser observado é que o custo computacional de calcular $\rho(AA^*)$ é significativamente mais elevado que computar a norma de Frobenius. Por este motivo, escolhemos utilizar a norma de Frobenius no cálculo de β em todas as implementações aqui apresentadas.

Observação 3. Sabemos que β deve satisfazer a condição $0 < \beta < \frac{2}{\rho(AA^*)}$, ou seja, temos mais de uma possibilidade de escolha para o seu valor. Logo, podemos perguntar: qual a influência da escolha de β nas iterações? Como resposta a essa pergunta, temos que a norma $\|R_k\|_2$ é minimizada ao escolher

$$\beta = \frac{2}{\lambda_1(AA^*) + \lambda_r(AA^*)},$$

em que $\lambda_1(AA^*)$ e $\lambda_r(AA^*)$ representam, respectivamente, o maior e o menor autovalor positivo de AA^*. Uma demonstração para este fato pode ser encontrada em [3], p. 277. Certamente, calcular este β corresponde a efetuar ainda mais cálculos do que para computar $\rho(AA^*)$, o que buscamos evitar aqui. Por isso, consideramos suficiente para nossas aplicações o uso de $\beta = \frac{1}{\|A\|_F^2}$.

Finalmente, temos como resultado os seguintes dois algoritmos, com os métodos linear e quadrático principais, e que serão utilizados mais a frente.

Algoritmo 1: Método (3.1) (*linear*)

Entrada: A

Saída: $A^{\dagger}$

início

Escolha β tal que $0 < \beta < \frac{2}{\rho(AA^*)}$. Em geral, escolhemos $\beta = \frac{1}{\|A\|_F^2}$;

Faça $X_0 = \beta A^*$;

para $k = 0, 1, 2, \ldots$ **faça**

$X_{k+1} = X_k + X_0(I - AX_k)$;

fim

fim

Algoritmo 2: Método (3.4) (*quadrático*)

Entrada: A

Saída: $A^{\dagger}$

início

Escolha β tal que $0 < \beta < \frac{2}{\rho(AA^*)}$. Em geral, escolhemos $\beta = \frac{1}{\|A\|_F^2}$;

Faça $X_0 = \beta A^*$;

para $k = 0, 1, 2, \ldots$ **faça**

$X_{k+1} = X_k + X_k(I - AX_k)$;

fim

fim

Exemplo 1. Este exemplo é apenas para mostrar como os métodos atuam, isto é, como se dá a convergência. Assim, seja $A \in \mathbb{C}^{5\times 5}$ a seguinte matriz:

$$A = \begin{bmatrix} 9 & 3 & 6 & 8 & 6 \\ 3 & 7 & 6 & 4 & 8 \\ 10 & 5 & 10 & 6 & 10 \\ 4 & 4 & 3 & 1 & 2 \\ 2 & 9 & 8 & 1 & 6 \end{bmatrix}.$$

Utilizando MATLAB, podemos calcular a matriz pseudo-inversa de A, que é dada abaixo. Observe que A é não-singular, o que implica que $A^{\dagger} = A^{-1}$.

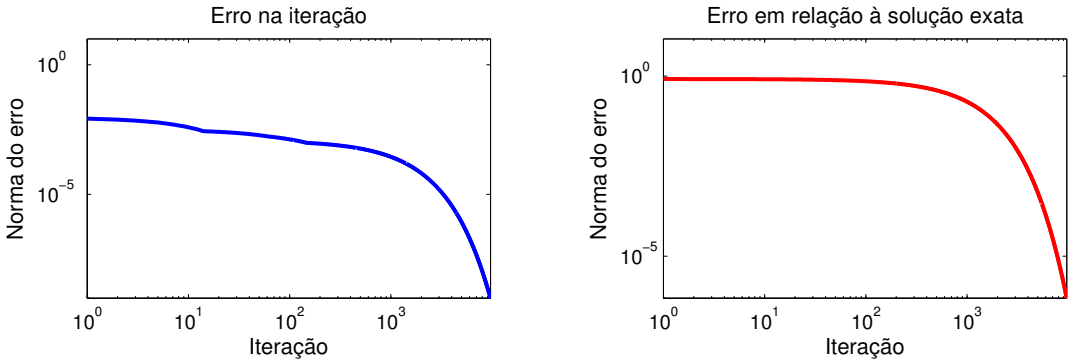
$$A^{\dagger} = \begin{bmatrix} -0.0769 & 0.0000 & 0.0769 & 0.3077 & -0.1538 \\ 0.0287 & 0.1111 & -0.1500 & 0.2284 & -0.0031 \\ 0.1235 & -0.3333 & 0.0583 & -0.3427 & 0.3380 \\ 0.2797 & -0.0000 & -0.1888 & -0.2098 & 0.1049 \\ -0.2288 & 0.2778 & 0.1531 & 0.0466 & -0.2455 \end{bmatrix}.$$

Para as implementações, consideramos dois critérios de parada: um por tolerância e outro pelo número máximo de iterações, i.e., o método para quando $\|X_k - X_{k-1}\|_2 < \varepsilon$, para $\varepsilon > 0$ dado, ou então para quando $k = k_{max}$, para $k_{max} > 0$ natural dado.

Utilizamos, também, para ambos os métodos, $X_0 = \beta A^*$, com $\beta = \frac{1}{\|A\|_F^2}$. Como A é uma matriz real neste caso, então $A^* = A^T$, logo, $X_0 = \beta A^T$.

Para o método iterativo (3.1), consideramos $k_{max} = 10^4$ e $\varepsilon = 10^{-9}$. Na Figura 2, temos o resultado do método. De fato, atingimos a tolerância esperada, com parada na iteração $k = 9651$, na qual temos que o valor atingido por $\|X_k - X_{k-1}\|_2$ é 9.9861×10^{-10}. Além disso, o valor para $\|A^\dagger - X_k\|_2$ é 6.8759×10^{-7}, nesta mesma iteração. Perceba a suavidade no declínio nos gráficos, característica da convergência linear do método.

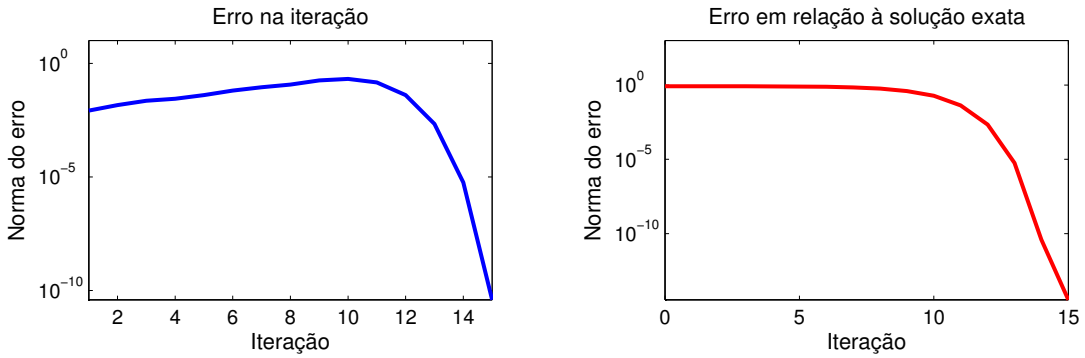
Figura 2 – Valores de $\|X_k - X_{k-1}\|_2$ (à esquerda) e de $\|A^\dagger - X_k\|_2$ (à direita), em cada iteração, para o método linear.



Fonte: o autor, 2015.

Agora, aplicando o método iterativo (3.4), para $k_{max} = 50$ e a mesma tolerância $\varepsilon = 10^{-9}$, temos na Figura 3 os resultados. Para este método, o critério de parada por tolerância também foi atingido, já que o método parou com 15 iterações, na qual o valor de $\|X_k - X_{k-1}\|_2$ é 3.9289×10^{-11}. Além disso, $\|A^\dagger - X_k\|_2$ assume o valor de 3.4998×10^{-15}. Logo, para este método, a aproximação da pseudo-inversa é de boa qualidade e com poucas iterações.

Figura 3 – Valores de $\|X_k - X_{k-1}\|_2$ (à esquerda) e de $\|A^\dagger - X_k\|_2$ (à direita), em cada iteração, para o método quadrático.



Fonte: o autor, 2015.

É interessante observar o comportamento de $\|X_k - X_{k-1}\|_2$ para o método (3.4): temos um crescimento do erro na iteração, inicialmente, apesar da convergência rápida em seguida. De fato, esse crescimento é comum em todos os testes feitos até o momento (inclusive para matrizes de ordens maiores), e representa simplesmente o "caminho do algoritmo", ou seja, está intrinsecamente relacionado com o funcionamento do método. Perceba que, apesar disso, temos que $\|A^\dagger - X_k\|_2$ está sempre decaindo. Ou seja, apesar de termos as matrizes X_k cujo erro cresce em algumas iterações, nos aproximamos cada vez mais de $A^\dagger$, junto com o crescimento de k.

Já o método (3.1) funciona de maneira mais uniforme. Também temos X_k se aproximando de $A^\dagger$ em cada iteração, mas muito mais lentamente que o método (3.4) o que já era esperado, pois (3.1) tem convergência linear, enquanto (3.4) tem convergência quadrática. De fato, essa diferença na convergência é bem clara, facilmente observada pela quantidade de iterações necessárias a cada um dos métodos para atingir a precisão desejada.

3.2 Métodos Baseados nas Equações de Penrose

É comum encontrarmos métodos iterativos para calcular a inversa $A^\dagger$ de A baseados nas equações de Penrose (veja a Definição 4). Perceba que o método (3.4) é desta forma, pois considerando a equação (II) de Penrose, isto é, $X = XAX$, então é fácil ver que

$$X = X + X - X = X + X - XAX = X + X(I - AX).$$

Em um trabalho recente, Petković em [13] propôs um método baseado nas equações (II) e (IV), fazendo uso do raciocínio de que $X^* = (XAX)^* = X^*(XA)^* = X^*XA$ e, assim,

para $\beta \in \mathbb{R}$, temos

$$\begin{aligned} X^* &= X^* - \beta(X^* - X^*) \\ &= X^* - \beta(X^*XA - X^*) \\ &= X^*(I - \beta XA) + \beta X^* \\ \Rightarrow X &= (I - \beta XA)^* X + \beta X, \end{aligned}$$

que sugere o método

$$X_{k+1} = (I - \beta X_k A)^* X_k + \beta X_k. \tag{3.5}$$

O lema a seguir apresenta alguns resultados interessantes referentes a esta sequência.

Lema 5. [13] *Sejam $A \in \mathbb{C}^{m\times n}$, $X_0 = \beta A^*$ e $X = A^\dagger$. Então, a sequência de matrizes gerada por (3.5) satisfaz:*

$$(X_k A)^* = X_k A,\ XAX_k = X_k,\ X_k AX = X_k,\ k = 0, 1, \ldots \tag{3.6}$$

Demonstração. A ideia é utilizar a indução. Para $k = 0$ e pelo item (ix) do Teorema 8, temos

$$\begin{aligned} (X_0 A)^* = (\beta A^* A)^* = \beta A^* A = X_0 A, \\ XAX_0 = \beta XAA^* = \beta A^* = X_0 \text{ e} \\ X_0 AX = \beta A^* AX = \beta A^* = X_0. \end{aligned}$$

Assim, supondo-se válidas as equações em (3.6) para k, então para o caso $k+1$ teremos

$$\begin{aligned} (X_{k+1}A)^* &= (((I - \beta X_k A)^* X_k + \beta X_k)A)^* \\ &= ((I - \beta X_k A)^* X_k A + \beta X_k A)^* \\ &= (X_k A)^*(I - \beta X_k A) + \beta (X_k A)^* \\ &= X_k A(I - \beta X_k A) + \beta X_k A \\ &= (I - \beta X_k A) X_k A + \beta X_k A \\ &= (I - \beta (X_k A)^*) X_k A + \beta X_k A \\ &= (I - \beta X_k A)^* X_k A + \beta X_k A \\ &= ((I - \beta X_k A)^* X_k + \beta X_k) A \\ &= X_{k+1} A. \end{aligned}$$

Procedendo de maneira análoga, comprovamos a veracidade das outras duas equações. □

Este lema nos leva a uma nova maneira de escrever a sequência, dando origem ao método

$$X_{k+1} = (I - \beta X_k A) X_k + \beta X_k = (1 + \beta) X_k - \beta X_k A X_k. \tag{3.7}$$

O próximo teorema dá resultados referentes à convergência de (3.7). Perceba que existe uma condição necessária à convergência que, à primeira vista, pode ser difícil de analisar na prática.

Teorema 13. [13] *Sejam* $A \in \mathbb{C}^{m\times n}$ *e* $X_0 = \beta A^*$. *Se*

$$0 < \beta \leq 1 \ e \ \|(X_0 - X)A\| \leq 1,$$

para alguma norma submultiplicativa $\|\cdot\|$, *então a sequência* $\{X_k\}$ *de matrizes gerada por* (3.7) *converge para* $X = A^\dagger$, *quando* $k \to \infty$.
Além disso, para $\beta < 1$, *o método tem convergência linear, enquanto que para* $\beta = 1$ *sua convergência é quadrática.*

Demonstração. Veja [13], p. 5. □

Como observado acima, para garantirmos a convergência da sequência (3.7) é necessário que $\|(X_0 - X)A\| \leq 1$ e que $0 < \beta \leq 1$. A condição para β é em geral satisfeita, principalmente para matrizes maiores, pois $0 < \beta < \frac{2}{\rho(AA^*)}$ e $\rho(AA^*)$ só em casos bem especiais assume valores menores ou iguais a 2, como é o caso da matriz identidade I_n, em que $\rho(I_n I_n^*) = 1$. Mas, mesmo neste caso, teríamos $0 < \beta < 2$, e nada impede uma escolha tal que $\beta \leq 1$, como é o caso de escolhermos $\beta = \frac{1}{\|I_n\|_F^2} = \frac{1}{n} < 1$, para todo $n \geq 2$.

Com relação à condição $\|(X_0 - X)A\| \leq 1$, que aparentemente é mais difícil de controlar, também não apresenta problemas: a desigualdade é verdadeira nas condições aqui utilizadas, ou seja, com β e X_0 como escolhidos (veja [13], p. 7, para uma demonstração completa deste fato).

O Teorema 13 também nos dá uma caracterização quanto à convergência do método. Perceba que se $\beta < 1$ temos convergência linear e se $\beta = 1$ temos convergência quadrática. É interessante notar que, para $\beta = 1$, o método (3.7) é equivalente ao método quadrático (3.4), pois

$$X_{k+1} = (1+\beta)X_k - \beta X_k A X_k = 2X_k - X_k A X_k = X_k + X_k(I - AX_k).$$

Finalmente, o algoritmo abaixo apresenta o pseudo-código do método.

Algoritmo 3: Método (3.7) (*baseado nas equações de Penrose*)

Entrada: A
Saída: $A^\dagger$
início
  Escolha β tal que $0 < \beta \leq \min\{1, \frac{2}{\rho(AA^*)}\}$. Em geral, escolhemos $\beta = \frac{1}{\|A\|_F^2}$;
  Faça $X_0 = \beta A^*$;
  para $k = 0, 1, 2, \ldots$ **faça**
    $X_{k+1} = (1+\beta)X_k - \beta X_k A X_k$;
  fim
fim

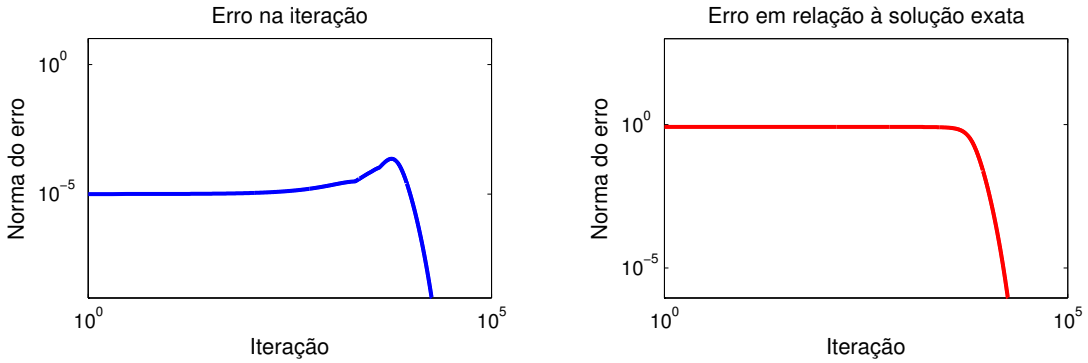
Exemplo 2. Como ilustração e comparação com os métodos já vistos acima, considere a mesma matriz do Exemplo 1, isto é,

$$A = \begin{bmatrix} 9 & 3 & 6 & 8 & 6 \\ 3 & 7 & 6 & 4 & 8 \\ 10 & 5 & 10 & 6 & 10 \\ 4 & 4 & 3 & 1 & 2 \\ 2 & 9 & 8 & 1 & 6 \end{bmatrix}.$$

Observe que $\|(X_0 - A^\dagger)A\|_2 = 0.9985 < 1$ e $\beta = \frac{1}{\|A\|_F^2} = 0.0010$, de modo que as condições do Teorema 13 estão satisfeitas e, ao aplicarmos o método, temos convergência garantida.

Para este método, consideramos $k_{max} = 10^5$, ou seja, suficientemente grande para que a tolerância de $\varepsilon = 10^{-9}$ seja atingida antes (ao menos para este exemplo). Na Figura 4 temos os resultados. O critério de parada por tolerância foi atingido para $k = 18119$, em que $\|X_k - X_{k-1}\|_2 = 9.9972 \times 10^{-10}$ e $\|A^\dagger - X_k\|_2 = 8.9698 \times 10^{-7}$.

Figura 4 – Valores de $\|X_k - X_{k-1}\|_2$ (à esquerda) e de $\|A^\dagger - X_k\|_2$ (à direita), em cada iteração, para o método (3.7).



Fonte: o autor, 2015.

Perceba que para a resolução de problemas reais (em que a ordem de A pode ser muito grande) o método tende a se tornar impraticável. Para este exemplo A é matriz 5×5 e é perceptível a convergência lenta. Para matrizes maiores, executar tantas iterações não é recomendado, não só pelo tempo computacional elevado, mas também pelos problemas com a aritmética de ponto flutuante.

Além disso, o melhor caso para este método, isto é, quando $\beta = 1$, é equivalente ao método quadrático. Logo, vamos nos focar em aplicações utilizando neste último método, que sempre apresenta convergência de ordem 2.

4 Sequências Vetoriais baseadas nos Métodos Iterativos

Considere o problema $Ax = b$, em que $A \in \mathbb{C}^{m\times n}, x \in \mathbb{C}^n$ e $b \in \mathbb{C}^m$. A ideia aqui é criar uma sequência de vetores $\{x_k\}$ tal que $x_k \longrightarrow \hat{x}$, quando $k \longrightarrow \infty$, em que $\hat{x} := A^\dagger b$. De fato, estamos tentando resolver o problema de mínimos quadrados associado a $Ax = b$. Como visto, se A é inversível então $A^\dagger = A^{-1}$, portanto, ao encontrar $A^\dagger b$ estamos encontrando a solução exata para o sistema linear $Ax = b$.

Umas das maneiras de construir a sequência $\{x_k\}$ é utilizando a teoria desenvolvida no Capítulo 3, com os métodos iterativos para a pseudo-inversa. Assim, considere $\{X_k\}$ sequência de matrizes que converge para $A^\dagger$, quando $k \longrightarrow \infty$. Defina, desta maneira, $x_k = X_k b$, para $k = 0, 1, \ldots$, e perceba que $x_k \longrightarrow \hat{x}$, como na seguinte proposição.

Proposição 8. *Sejam $A \in \mathbb{C}^{m\times n}$ e $\{X_k\}$ sequência de matrizes tal que $X_k \longrightarrow A^\dagger$, quando $k \longrightarrow \infty$. Além disso, considere o sistema linear $Ax = b$, para $b \in \mathbb{C}^m$. Então, a sequência de vetores $\{x_k\}$ dada por*

$$x_k = X_k b, \;\; k = 0, 1, \ldots,$$

converge para $\hat{x} := A^\dagger b$.

Demonstração. Seja $\|\cdot\|$ alguma norma submultiplicativa. Se $\|b\| = 0$, então $b = \mathbf{0}$, e, desta maneira, $x_k = X_k b = \mathbf{0}$, para todo k. Além disso, $\hat{x} = A^\dagger b = \mathbf{0}$, ou seja, claramente, $x_k \longrightarrow \hat{x}$. Note que neste caso estaríamos resolvendo $Ax = \mathbf{0}$, cuja solução de norma mínima é $\hat{x} = \mathbf{0}$.
Agora, considere $\|b\| \neq 0$. Seja $\varepsilon > 0$ qualquer. Como $\lim_{k\to\infty} X_k = A^\dagger$, então existe $k_0 \in \mathbb{N}$ tal que $\|X_k - A^\dagger\| < \frac{\varepsilon}{\|b\|}$, para todo $k \geq k_0$.
Assim, para todo $k \geq k_0$,

$$\|x_k - \hat{x}\| = \|X_k b - A^\dagger b\| = \|(X_k - A^\dagger)b\| \leq \|X_k - A^\dagger\|\|b\| < \frac{\varepsilon}{\|b\|}\|b\| = \varepsilon.$$

Ou seja, $x_k \longrightarrow \hat{x}$, quando $k \longrightarrow \infty$, e o resultado segue. □

Observação 4. Apesar de termos desenvolvido uma teoria geral para os métodos iterativos no capítulo anterior, não usaremos todos aqui, na construção das sequências vetoriais. De fato, vamos construir a sequência $x_k = X_k b$ para os métodos linear (3.1) e quadrático (3.4), o primeiro a título de motivação e o segundo por apresentar os melhores resultados numéricos.

É claro que podemos construir outras sequências vetoriais, já que não vimos apenas os dois métodos citados. De fato, temos um método de convergência de ordem p, dado em (3.3), e o método baseado nas equações de Penrose.

Para (3.3), o caso mais vantajoso numericamente é quando $p = 2$, que vamos tratar.

Já o método baseado nas equações de Penrose não apresenta bons resultados práticos, como discutido e exemplificado na Seção 3.2. Logo, o desenvolvimento da sequência vetorial apresenta as mesmas dificuldades. Além de que, no melhor caso, este método tem o desempenho do método quadrático. Por estes motivo, não demos foco em seu desenvolvimento.

Agora, considere os métodos iterativos linear e quadrático desenvolvidos no capítulo anterior. Iniciemos com o caso linear dado pela recorrência $X_{k+1} = X_k + X_0(I - AX_k)$, ou seja, para $x_k = X_k b$, temos

$$x_{k+1} = X_{k+1}b = X_k b + X_0(I - AX_k)b = x_k + x_0 - X_0 A x_k = x_0 + (I - X_0 A)x_k.$$

Perceba que podemos, ainda, escrever $U_0 = I - X_0 A$, o que gera o primeiro método vetorial deste capítulo, dado por

$$x_{k+1} = x_0 + U_0 x_k. \tag{4.1}$$

Desta maneira, temos um método que converge para $A^\dagger b$, de maneira linear, já que o método iterativo utilizado para a pseudo-inversa tem convergência linear.

Algoritmo 4: Método Vetorial (4.1) (*linear*)

Entrada: A, b

Saída: $\hat{x}$ (solução do sistema $Ax = b$)

início

  Escolha β tal que $0 < \beta < \frac{2}{\rho(AA^*)}$. Em geral, escolhemos $\beta = \frac{1}{\|A\|_F^2}$;

  Faça $X_0 = \beta A^*$, $x_0 = X_0 b$ e $U_0 = I - X_0 A$;

  para $k = 0, 1, 2, \ldots$ **faça**

    $x_{k+1} = x_0 + U_0 x_k$;

  fim

fim

4.1 O Caso Quadrático

Considerando o método $X_{k+1} = X_k + X_k(I - AX_k) = X_k(2I - AX_k)$, começam a surgir problemas, já que, neste caso, temos

$$x_{k+1} = X_{k+1}b = (X_k(2I - AX_k))b = 2X_k b - X_k A X_k b = 2x_k - X_k A x_k. \tag{4.2}$$

Ou seja, ficamos com o método em função das matrizes X_k em cada iteração, o que não é procurado, uma vez que calcular as matrizes X_k é o mesmo que computar a pseudo-inversa, algo que está sendo evitado aqui.

Nas próximas subseções apresentaremos algumas das maneiras utilizadas aqui para contornar a construção das matrizes X_k em cada iteração. A primeira ideia visa computar o produto X_kA, de alguma maneira. A segunda opção encontrada foi a partir de manipulações algébricas do método.

4.1.1 Computando o Produto X_kA

Uma maneira de contornar essa situação é computando o produto X_kA, com algum método iterativo. Um resultado neste sentido é apresentado na proposição a seguir, em que encontramos um método iterativo para computar o produto $BA^\dagger$, em que $B \in \mathbb{C}^{q\times n}$ é matriz qualquer. Observe que na verdade buscamos algo da forma $A^\dagger B$.

Proposição 9. [3] *Considere o método iterativo (3.3), ou seja,*

$$X_{k+1} = X_k(I + T_k + T_k^2 + \ldots + T_k^{p-1}),$$

em que $T_k = I - AX_k$, para $k = 0, 1, \ldots$, com X_0 satisfazendo as condições do Teorema 11. Assim, dada uma matriz $B \in \mathbb{C}^{q\times n}$ qualquer, seja a sequência $\{Z_k\}$ dada por

$$\begin{aligned} Z_0 &= BX_0, \\ Z_{k+1} &= Z_kM_k, \ \textit{para } k = 0, 1, \ldots, \end{aligned} \tag{4.3}$$

em que

$$\begin{aligned} M_k &= I + T_k + T_k^2 + \ldots + T_k^{p-1}, \ \textit{para } k = 0, 1, \ldots, \\ T_{k+1} &= I + M_k(T_k - I), \ \textit{para } k = 0, 1, \ldots \end{aligned}$$

e $T_0 = I - AX_0$. Então, $Z_k \longrightarrow BA^\dagger$ quando $k \longrightarrow \infty$.

Demonstração. O resultado é praticamente uma decorrência do Teorema 11. Veja [3], p. 280, para maior aprofundamento. □

Agora, se considerarmos $p = 2$, então perceba que

$$T_{k+1} = I + M_k(T_k - I) = I + (I + T_k)(T_k - I) = T_k^2.$$

Portanto,

$$M_k = I + T_k = I + (T_{k-1})^{2^1} = I + (T_{k-2})^{2^2} = \ldots = I + (T_0)^{2^k},$$

e, assim, a sequência dada em (4.3) pode ser escrita como

$$Z_{k+1} = Z_kM_k = Z_k(I + (T_0)^{2^k}), \ \text{para } k = 0, 1, \ldots \tag{4.4}$$

Desta maneira, Z_k se torna relativamente simples de computar.

Vamos utilizar a sequência de (4.4) para calcular o produto $A^\dagger A$. Para isto, perceba que $(A^\dagger A)^* = A^*(A^\dagger)^* = A^*(A^*)^\dagger$. Desta maneira, podemos alterar a sequência $\{Z_k\}$ (que converge originalmente para $BA^\dagger$) para que Z_k convirja para $A^*(A^*)^\dagger$, quando k tende ao infinito. Ou seja, se utilizarmos $B = A^*$ e $\{X_k\}$ tal que $X_k \longrightarrow (A^*)^\dagger$, quando $k \longrightarrow \infty$, então $\{Z_k\}$ converge para $A^*(A^*)^\dagger$.

Em termos práticos, para (4.4), utilizando $X_0 = \beta(A^*)^* = \beta A$, então

$$Z_0 = A^* X_0 \text{ e } T_0 = I - A^* X_0$$

e, assim, a sequência $Z_{k+1} = Z_k(I + (T_0)^{2^k})$, para $k = 0, 1, \ldots$, é tal que $Z_k \longrightarrow A^*(A^*)^\dagger$, quando $k \longrightarrow \infty$. Logo,

$$(Z_k)^* \longrightarrow (A^*(A^*)^\dagger)^* = A^\dagger A,$$

quando $k \longrightarrow \infty$.

Como as sequências $\{X_k A\}$ e $\{(Z_k)^*\}$ convergem para o mesmo limite $A^\dagger A$, então a ideia é trocar o termo $X_k A$ por $(Z_k)^*$ no método vetorial (4.2), resultando em

$$x_{k+1} = 2x_k - (Z_k)^* x_k, \text{ para } k = 0, 1, \ldots, \tag{4.5}$$

de modo que os problemas iniciais com calcular $X_k A$ estão resolvidos.

Algoritmo 5: Método Vetorial (4.5) (*quadrático, computando* $X_k A$)

Entrada: A, b

Saída: $\hat{x}$ (solução do sistema $Ax = b$)

início

Escolha β tal que $0 < \beta < \frac{2}{\rho(AA^*)}$. Em geral, escolhemos $\beta = \frac{1}{\|A\|_F^2}$;

Faça $X_0 = \beta A^*$, $x_0 = X_0 b$, $T_0 = I - A^* X_0^*$ e $Z_0 = A^* X_0^*$;

para $k = 0, 1, 2, \ldots$ **faça**

$x_{k+1} = 2x_k - (Z_k)^* x_k$;

$Z_{k+1} = Z_k(I + T_0)$;

$T_0 \leftarrow T_0^2$;

fim

fim

Note que mesmo com esta alternativa, ainda temos que construir a matriz Z_k em cada iteração, o que nos leva a efetuar, na melhor possibilidade, dois produtos matriz-matriz por iteração. Certamente, quanto maior a dimensão do problema, maiores serão as dificuldades computacionais para efetuar estes produtos.

4.1.2 Alternativa por Manipulação Algébrica

No Algoritmo 5 utilizamos a ideia contida no Proposição 9 para resolver o problema de calcular $X_k A$ em cada iteração, a partir da construção de uma nova sequência de

matrizes Z_k. Mas existe outra possibilidade, diretamente a partir do método quadrático, isto é, considere

$$X_{k+1} = X_k + X_k(I - AX_k).$$

Claramente, podemos reescrever a equação acima como

$$X_{k+1} = X_k + (I - X_kA)X_k.$$

Denotando $U_k := I - X_kA$, temos $X_{k+1} = X_k + U_kX_k$. Desta equação, decorre que

$$\begin{aligned} X_{k+1} &= X_k + U_kX_k \\ \Leftrightarrow X_{k+1}A &= X_kA + U_kX_kA \\ \Leftrightarrow I - X_{k+1}A &= I - X_kA - U_kX_kA = (I - X_kA) - U_kX_kA \\ \Leftrightarrow U_{k+1} &= U_k - U_kX_kA = U_k(I - X_kA) \\ \Leftrightarrow U_{k+1} &= U_k^2. \end{aligned}$$

Logo, temos que

$$U_{k+1} = U_k^{2^1} = U_{k-1}^{2^2} = U_{k-2}^{2^3} = \cdots = U_0^{2^{k+1}},$$

de modo que

$$X_{k+1} = X_k + U_kX_k = X_k + U_0^{2^k}X_k, \tag{4.6}$$

em que $U_0 = I - X_0A = I - \beta A^*A$.

Agora, considerando novamente a sequência vetorial, isto é, $x_k = X_kb$, temos

$$x_{k+1} = x_k + U_0^{2^k}x_k. \tag{4.7}$$

A equação acima sugere dois tipos de implementação numérica. A primeira busca computar a matriz $U_0^{2^k}$ a partir de um produto matriz-matriz em cada iteração. A segunda possibilidade é calcular $U_0^{2^k}x_k$ por 2^k produtos matriz-vetor por iteração. As duas ideias podem ser vistas nos algoritmos seguintes.

Algoritmo 6: Método Vetorial (4.7) (*quadrático, versão matriz-matriz*)

Entrada: A, b
Saída: $\hat{x}$ (solução do sistema $Ax = b$)
início
  Escolha β tal que $0 < \beta < \frac{2}{\rho(AA^*)}$. Em geral, escolhemos $\beta = \frac{1}{\|A\|_F^2}$;
  Faça $X_0 = \beta A^*$, $x_0 = X_0b$, $U_0 = I - X_0A$;
  para $k = 0, 1, 2, \ldots$ **faça**
    $x_{k+1} = x_k + U_0x_k$;
    $U_0 \leftarrow U_0^2$;
  fim
fim

Algoritmo 7: Método Vetorial (4.7) (*quadrático, versão matriz-vetor*)

Entrada: A, b

Saída: $\hat{x}$ (solução do sistema $Ax = b$)

início

Escolha β tal que $0 < \beta < \frac{2}{\rho(AA^*)}$. Em geral, escolhemos $\beta = \frac{1}{\|A\|_F^2}$;

Faça $X_0 = \beta A^*$, $x_0 = X_0 b$, $U_0 = I - X_0 A$;

para $k = 0, 1, 2, \ldots$ **faça**

$y = x_k$;

para $i = 1, 2, \ldots, 2^k$ **faça**

$y \leftarrow U_0 y$;

fim

$x_{k+1} = x_k + y$;

fim

fim

Exemplo 3. Considere o sistema linear $Ax = b$, em que

$$A = \begin{bmatrix} 17 & 1 & 5 & 4 & 20 & 3 & 13 \\ 7 & 5 & 16 & 2 & 9 & 3 & 3 \\ 2 & 1 & 12 & 18 & 2 & 20 & 10 \\ 8 & 12 & 14 & 12 & 13 & 17 & 17 \\ 2 & 5 & 6 & 8 & 3 & 2 & 5 \end{bmatrix} \text{ e } b = \begin{bmatrix} 1 \\ 3 \\ 9 \\ 2 \\ 19 \end{bmatrix}.$$

Estamos buscando o vetor $\hat{x} = A^\dagger b$, que é solução do problema de mínimos quadrados associado a $Ax = b$. Para tanto, vamos utilizar os métodos vetoriais desenvolvidos nesta seção. Novamente, escolhemos $X_0 = \beta A^*$, para $\beta = \frac{1}{\|A\|_F^2}$. Os critérios de parada são os mesmos do Exemplo 1: o algoritmo para ao atingir tolerância ε desejada ou o número máximo de iterações k_{max}.

Computando a solução exata, temos:

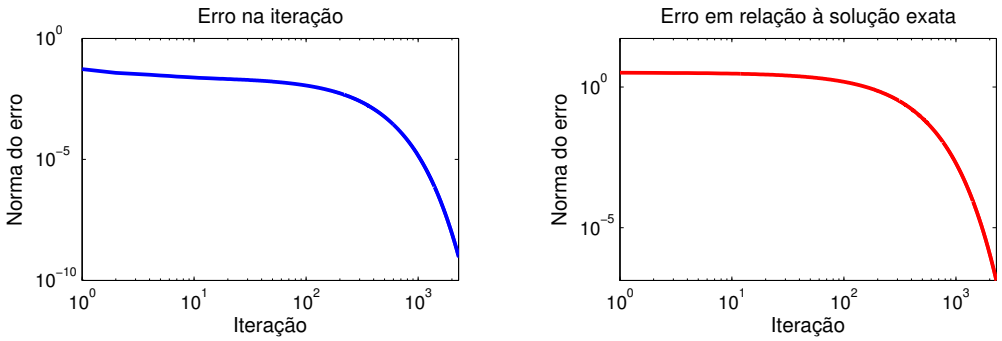
$$\hat{x} = A^\dagger b = \begin{bmatrix} 0.0295 \\ 0.3708 \\ 0.2861 \\ 2.4826 \\ -0.3229 \\ -1.9913 \\ 0.0921 \end{bmatrix}.$$

Obviamente, nos problemas reais, não temos acesso a $\hat{x}$, apresentamos aqui o seu valor apenas a título de ilustração.

Para o método vetorial (4.1), utilizamos $\varepsilon = 10^{-9}$ e $k_{max} = 3000$. Como este método tem convergência linear, ele funciona lentamente, como pode ser visto nos resultados da

Figura 5. Apesar disso, o método parou em $k = 2290$ iterações, na qual os valores atingidos por $\|x_k - x_{k-1}\|_2$ e $\|x_k - \hat{x}\|_2$ são 9.9921×10^{-10} e 1.3424×10^{-7}, respectivamente.

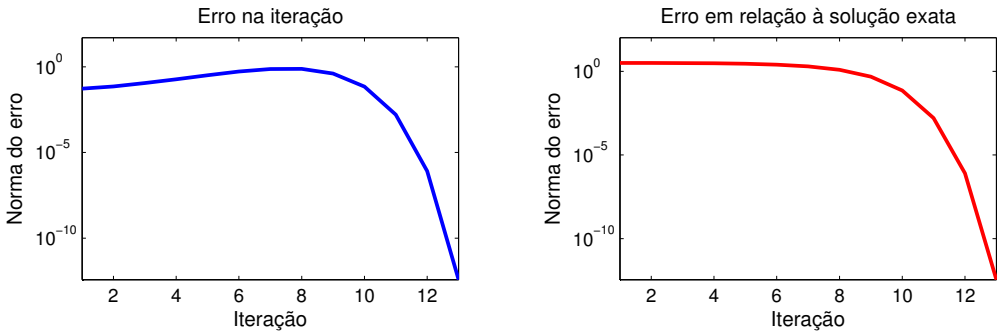
Figura 5 – Valor de $\|x_k - x_{k-1}\|_2$ (à esquerda) e valor de $\|x_k - A^\dagger b\|_2$ (à direita), em cada iteração.



Fonte: o autor, 2015.

Para o método (4.5), a convergência é quadrática, ou seja, encontramos rapidamente a solução, como apresentado na Figura 6. Aqui, o algoritmo parou com $k = 13$ iterações, assumindo, para este k, os valores de $\|x_k - x_{k-1}\|_2 = 3.0486 \times 10^{-13}$ e $\|x_k - \hat{x}\|_2 = 4.3936 \times 10^{-13}$. Portanto, atingimos uma aproximação de boa qualidade da solução exata.

Figura 6 – Valor de $\|x_k - x_{k-1}\|_2$ (à esquerda) e valor de $\|x_k - A^\dagger b\|_2$ (à direita), em cada iteração.



Fonte: o autor, 2015.

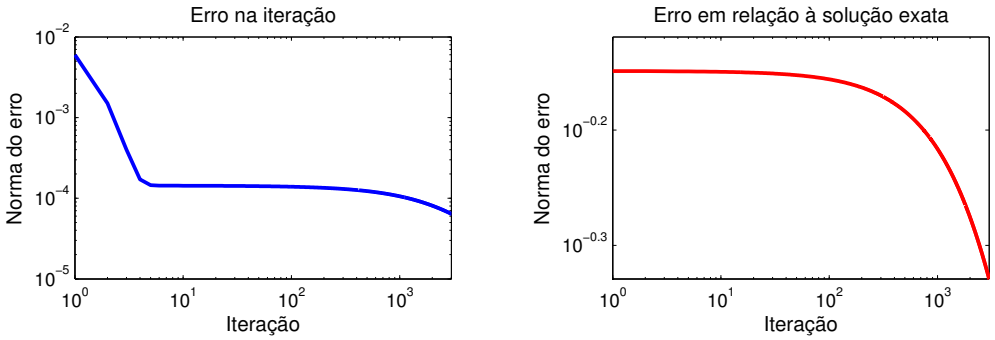
Exemplo 4. Nos moldes do Exemplo 3, considere o sistema linear $Ax = b$, em que $A \in \mathbb{C}^{3000\times1000}$ e $b \in \mathbb{C}^{3000}$ são gerados randomicamente, de modo que todo elemento de A e b pertence ao intervalo $(0, 1)$. Novamente, utilizamos $\varepsilon = 10^{-9}$ para a tolerância.

Aplicando o método (4.1), temos os resultados na Figura 7. Consideramos $k_{max} = 3000$ e não atingimos a precisão desejada, já que o algoritmo parou pelo número de iterações. Assim, para $k = k_{max}$, temos $\|x_k - x_{k-1}\|_2 = 6.4134 \times 10^{-5}$ e $\|x_k - \hat{x}\|_2 = 0.4688$.

Perceba que, apesar do número de iterações, temos pequena redução em $\|x_k - \hat{x}\|_2$, o que leva a várias hipóteses, entre elas a possibilidade de erros na aritmética computacional

(devido ao número de operações matriciais) e a óbvia lentidão causada pela convergência linear, que certamente tem grande influência no resultado.

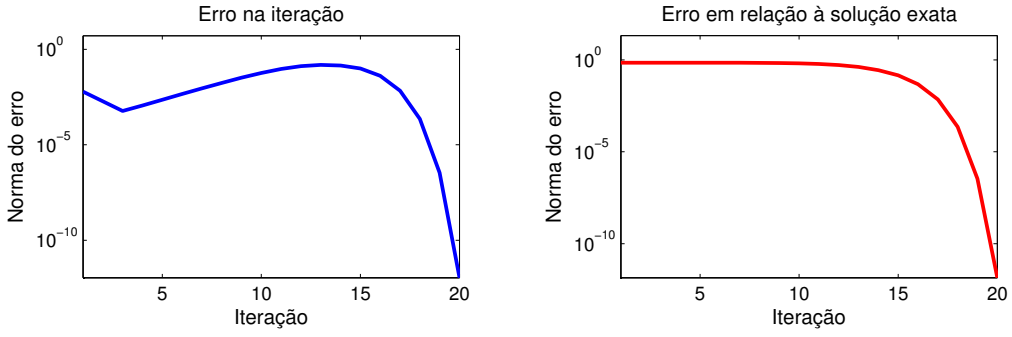
Figura 7 – Valor de $\|x_k - x_{k-1}\|_2$ (à esquerda) e valor de $\|x_k - A^\dagger b\|_2$ (à direita), em cada iteração.



Fonte: o autor, 2015.

Já para o método (4.5), temos um bom resultado, como pode ser constatado na Figura 8. Aqui, apesar de considerarmos $k_{max} = 30$, o algoritmo parou em $k = 20$, com os valores $\|x_k - x_{k-1}\|_2 = 1.0887 \times 10^{-12}$ e $\|x_k - \hat{x}\|_2 = 1.4088 \times 10^{-12}$. A convergência quadrática faz grande diferença na eficiência do método. O que deve ser notado é que, apesar das ordens de A e b serem elevadas, o método quadrático continua funcionando eficientemente, enquanto que o método linear apresenta piora significativa.

Figura 8 – Valor de $\|x_k - x_{k-1}\|_2$ (à esquerda) e valor de $\|x_k - A^\dagger b\|_2$ (à direita), em cada iteração.



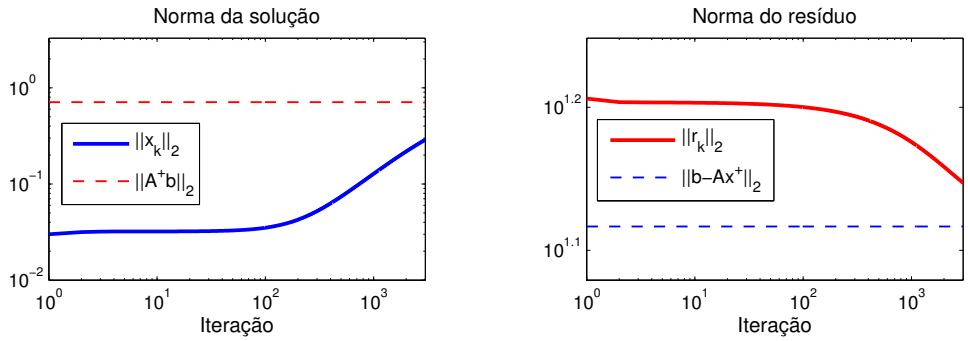
Fonte: o autor, 2015.

Outra análise interessante pode ser feita quando consideramos o comportamento de $\|x_k\|_2$ e de $\|r_k\|_2$, em que $r_k := b - Ax_k$ é conhecido como *resíduo* na iteração k (vale notar que em muitas situações a análise do resíduo é utilizada como critério de parada dos algoritmos). Nas Figuras 9 e 10, temos a possibilidade de analisar $\|x_k\|_2$ e $\|r_k\|_2$ para este exemplo.

Nas figuras, as linhas pontilhadas representam o objetivo de cada uma das sequências: $\|x_k\|_2$ deve convergir para $\|\hat{x}\|_2 = \|A^\dagger b\|_2$ e $\|r_k\|_2$ para $\|b - A\hat{x}\|_2$, que minimiza a distância entre b e $Im(A)$. Esclarecendo a notação, o vetor $x^\dagger$ nas figuras representa $\hat{x} = A^\dagger b$.

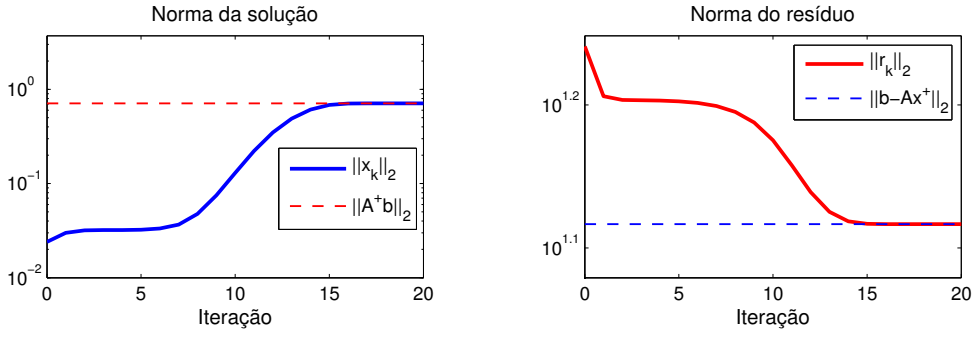
Note um aparente comportamento monotônico das sequências de normas: neste exemplo, $\|x_k\|_2$ é crescente e $\|r_k\|_2$ é decrescente, tanto para o método linear quanto para o quadrático. A partir disso, surge a pergunta: esta propriedade é pontual deste exemplo, ou pode ser generalizada? Veremos uma resposta para isto na próxima subseção.

Figura 9 – Comportamento de $\|x_k\|_2$ (à esquerda) e de $\|r_k\|_2$ (à direita), em cada iteração, para o método linear.



Fonte: o autor, 2015.

Figura 10 – Comportamento de $\|x_k\|_2$ (à esquerda) e de $\|r_k\|_2$ (à direita), em cada iteração, para o método quadrático.



Fonte: o autor, 2015.

4.1.3 Propriedades do Método Quadrático

Considere o problema $Ax = b$ e a sequência de vetores $\{x_k\}$ tal que $x_k = X_k b$, em que $\{X_k\}$ é sequência matricial que converge para $A^\dagger$, quando $k \longrightarrow \infty$. Pela construção da sequência, segue imediatamente a convergência de x_k para $\hat{x} = A^\dagger b$ (também verificada

na Proposição 8), ou seja, x_k converge para uma solução de mínimos quadrados de $Ax = b$, de fato a solução de norma mínima.

Além disso, como $x_k = X_k b$, isto é, a sequência vetorial deriva de uma sequência matricial, temos que a convergência de $\{x_k\}$ possui a mesma ordem que a convergência de $\{X_k\}$. Por exemplo, considere o método da forma

$$x_{k+1} = x_k + U_0^{2^k} x_k,$$

dado nos Algoritmos 6 e 7, que deriva da relação (3.4), isto é,

$$X_{k+1} = X_k + X_k(I - AX_k).$$

Assim, como (3.4) possui convergência quadrática, temos que $\{x_k\}$ também converge quadraticamente. O mesmo vale para o método linear desenvolvido em (4.1), que possui, pelos mesmos motivos, convergência linear, já que decorre de uma sequência matricial que converge linearmente.

Agora, antes de continuarmos, vamos enunciar e mostrar a seguinte proposição, vital no estabelecimento do principal resultado desta subseção. Para tanto, vamos lembrar que, dada uma matriz $B \in \mathbb{C}^{n\times n}$ hermitiana, então todos os seus autovalores são reais. De fato, se λ é autovalor de B, então existe $x \in \mathbb{C}^n$, com $x \neq \mathbf{0}$, autovetor de B associado a λ, isto é, $Bx = \lambda x$. Assim, usando propriedades do produto interno e de B, que é hermitiana, temos

$$\lambda\langle x, x\rangle = \langle \lambda x, x\rangle = \langle Bx, x\rangle = \langle x, B^*x\rangle = \langle x, Bx\rangle = \langle x, \lambda x\rangle = \overline{\lambda}\langle x, x\rangle.$$

Como $\langle x, x\rangle \neq 0$, então $\lambda = \overline{\lambda}$ e, logo, $\lambda \in \mathbb{R}$.

Proposição 10. *Seja $B \in \mathbb{C}^{n\times n}$ hermitiana e $\lambda_n \leq \lambda_{n-1} \leq \ldots \leq \lambda_1$ os seus autovalores. Defina*

$$\lambda_{min} = \min_{1\leq i\leq n} |\lambda_i| \text{ e } \lambda_{max} = \max_{1\leq i\leq n} |\lambda_i|.$$

Assim, para todo $x \in \mathbb{C}^n$, é válido que

$$\lambda_{min}\|x\|_2 \leq \|Bx\|_2 \leq \lambda_{max}\|x\|_2.$$

Demonstração. Como B é hermitiana, segue que B é normal, isto é, a matriz satisfaz $BB^* = B^*B$. Logo, temos que B é diagonalizável e podemos escrevê-la da forma $B = Q\Lambda Q^*$ (para maiores detalhes, veja [9], p. 547), com $Q \in \mathbb{C}^{n\times n}$ unitária e

$$\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_n) = \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{bmatrix} \in \mathbb{C}^{n\times n}.$$

Agora, seja $x \in \mathbb{C}^n$ qualquer e denote $y = Q^*x$. Perceba que

$$\|y\|_2 = \|Q^*x\|_2 = (Q^*x)^*Q^*x = x^*QQ^*x = x^*x = \|x\|_2.$$

Além disso,

$$\|Bx\|_2 = \|Q\Lambda Q^* x\|_2 = \|Q\Lambda y\|_2 = (Q\Lambda y)^* Q\Lambda y = y^*\Lambda^* Q^* Q\Lambda y = (\Lambda y)^*\Lambda y = \|\Lambda y\|_2 =$$
$$= \left\| \begin{pmatrix} \lambda_1 y_1 \\ \vdots \\ \lambda_n y_n \end{pmatrix} \right\|_2 = \sqrt{\lambda_1^2 y_1^2 + \ldots + \lambda_n^2 y_n^2} = \sqrt{|\lambda_1|^2 y_1^2 + \ldots + |\lambda_n|^2 y_n^2}.$$

Assim, temos que

$$\|Bx\|_2 = \sqrt{|\lambda_1|^2 y_1^2 + \ldots + |\lambda_n|^2 y_n^2} \geq \sqrt{\lambda_{min}^2 (y_1^2 + \ldots + y_n^2)} = \lambda_{min}\|y\|_2 = \lambda_{min}\|x\|_2.$$

Da mesma maneira, segue a outra parte da desigualdade:

$$\|Bx\|_2 = \sqrt{|\lambda_1|^2 y_1^2 + \ldots + |\lambda_n|^2 y_n^2} \leq \sqrt{\lambda_{max}^2 (y_1^2 + \ldots + y_n^2)} = \lambda_{max}\|y\|_2 = \lambda_{max}\|x\|_2.$$

Logo, $\lambda_{min}\|x\|_2 \leq \|Bx\|_2 \leq \lambda_{max}\|x\|_2$ e o resultado segue. □

O objetivo central desta subseção é a construção de relações de monotonicidade para as normas das soluções e resíduos do método vetorial quadrático desenvolvido acima, que serão necessárias futuramente (utilizaremos alguns critérios de parada em problemas de teste que necessitam destas propriedades teóricas). De fato, queremos mostrar que para a sequência vetorial x_k dada em (4.7) temos que

$$\|x_{k+1}\|_2 \geq \|x_k\|_2, \text{ para todo } k = 1, 2, \ldots$$

e, se considerarmos o resíduo na k-ésima iteração como $r_k := b - Ax_k$, então

$$\|r_{k+1}\|_2 \leq \|r_k\|_2, \text{ para todo } k = 0, 1, \ldots$$

Para tanto, lembre que no desenvolvimento da equação (4.7) extraímos algumas propriedades e modos de reescrever o método quadrático. Assim, vimos que

$$X_{k+1} = X_k + U_k X_k = (I + U_k)X_k,$$

em que $U_k = I - X_k A$. Além disso, temos que $U_k = U_0^{2^k}$. Analogamente, verificamos que

$$X_{k+1} = X_k + U_k X_k = X_k + X_k T_k,$$

com $T_k = I - AX_k$, e que vale $T_{k+1} = T_k^2$, o que implica que $T_k = T_0^{2^k}$.

Agora, finalmente, vamos ao resultado. Note que no caso da monotonicidade de $\|x_k\|_2$ começamos com $k = 1$. De fato, veremos que não é necessariamente verdade que $\|x_1\|_2 \geq \|x_0\|_2$, a partir de um contraexemplo.

Teorema 14. *Considere o sistema $Ax = b$, com $A \in \mathbb{C}^{m\times n}$, $x \in \mathbb{C}^n$ e $b \in \mathbb{C}^m$, e a sequência vetorial dada por $x_k = X_k b$, com X_k sequência de matrizes que converge para $A^\dagger$ quadraticamente a partir da seguinte recorrência:*

$$X_{k+1} = X_k + X_k(I - AX_k),$$

com $X_0 = \beta A^*$ e $0 < \beta < \frac{2}{\rho(AA^*)}$. Assim, são válidos:

(i) $\|x_{k+1}\|_2 \geq \|x_k\|_2$, *para todo* $k = 1, 2, \ldots$;

(ii) $\|r_{k+1}\|_2 \leq \|r_k\|_2$, *para todo* $k = 0, 1, \ldots$, *em que* $r_k := b - Ax_k$.

Demonstração. O ponto chave para esta demonstração é o uso da Proposição 10. Para isso, veremos que diversas das matrizes envolvidas aqui são hermitianas e que podemos construir relações para os seus autovalores, de modo a garantir o resultado.

(i) A sequência x_k pode ser escrita pela seguinte recorrência:

$$x_{k+1} = X_{k+1}b = (I + U_k)x_k.$$

Vejamos que $B_k := I + U_k$ é hermitiana. Para tanto, lembre que $U_k = U_0^{2^k}$ e, deste modo, basta verificar que U_0 é hermitiana. Mas $U_0 = I - X_0A = I - \beta A^*A$ e

$$U_0^* = (I - \beta A^*A)^* = I^* - \beta(A^*A)^* = I - \beta A^*A = U_0.$$

Assim, U_0 é hermitiana, o que implica que U_k também é. Logo, B_k é hermitiana também, e todos os seus autovalores são reais. Agora, considerando a notação $\lambda_i(C)$ para representar o i-ésimo autovalor de uma matriz C qualquer, então para todo $k = 1, 2, \ldots$,

$$\lambda_i(U_k) = \lambda_i(U_0^{2^k}) = (\lambda_i(U_0))^{2^k} \geq 0, \text{ para todo } i = 1, 2, \ldots, n.$$

Logo,

$$\lambda_i(B_k) = \lambda_i(I + U_k) = \lambda_i(I) + \lambda_i(U_k) = 1 + \lambda_i(U_k) \geq 1, \text{ para todo } i = 1, 2, \ldots, n.$$

Definindo

$$\lambda_{min} = \min_{1 \leq i \leq n} |\lambda_i(B_k)| \geq 1,$$

temos, pela Proposição 10,

$$\|x_{k+1}\|_2 = \|B_kx_k\|_2 \geq \lambda_{min}\|x_k\|_2 \geq \|x_k\|_2, \text{ para todo } k = 1, 2, \ldots$$

(ii) Por $T_{k+1} = T_k^2$ e lembrando que $x_k = X_kb$, temos que

$$\begin{aligned} I - AX_{k+1} = T_{k+1} = T_k^2 &= T_k(I - AX_k) \\ \Leftrightarrow b - AX_{k+1}b &= T_k(b - AX_kb) \\ \Leftrightarrow b - Ax_{k+1} &= T_k(b - Ax_k) \\ \Leftrightarrow r_{k+1} &= T_kr_k. \end{aligned}$$

Assim,

$$\|r_{k+1}\|_2 = \|T_kr_k\|_2.$$

Vejamos que T_k é hermitiana. Para isto, perceba que basta verificar que T_0 é hermitiana, pois $T_k = T_0^{2^k}$. Mas $T_0 = I - AX_0 = I - \beta AA^*$ é claramente hermitiana.

Logo, temos que T_k é hermitiana. Por esta propriedade, segue que os autovalores de T_k são reais.

Por outro lado, se $\lambda_i(AA^*)$ é o i-ésimo autovalor de AA^* (que são todos reais e não negativos, pelo Lema 1) e λ_1 representa o seu maior autovalor (de AA^*), sabemos que β satisfaz

$$0 < \beta < \frac{2}{\rho(AA^*)} = \frac{2}{\lambda_1}.$$

Portanto,

$$\begin{aligned} 0 &> -\beta > -\frac{2}{\lambda_1} \\ \Leftrightarrow\ 0 &> -\beta\lambda_i(AA^*) > -\frac{2\lambda_i(AA^*)}{\lambda_1} \\ \Leftrightarrow\ 1 &> 1-\beta\lambda_i(AA^*) > 1-\frac{2\lambda_i(AA^*)}{\lambda_1}. \end{aligned}$$

Note que $\frac{\lambda_i(AA^*)}{\lambda_1} \leq 1$, para todo $i = 1, 2, \ldots, m$, por λ_1 ser o maior autovalor de AA^*. Logo, temos

$$1 > 1-\beta\lambda_i(AA^*) > 1-\frac{2\lambda_i(AA^*)}{\lambda_1} \geq 1-2 = -1. \tag{4.8}$$

Perceba que, como $T_0 = I - \beta AA^*$, então

$$\lambda_i(T_0) = \lambda_i(I) - \beta\lambda_i(AA^*) = 1 - \beta\lambda_i(AA^*).$$

Assim, pela desigualdade (4.8), temos que $-1 < \lambda_i(T_0) < 1$, para todo $i = 1, 2, \ldots, m$. Desta maneira, como $\lambda_i(T_k) = \lambda_i(T_0^{2^k}) = [\lambda_i(T_0)]^{2^k}$, então segue que

$$-1 < \lambda_i(T_k) < 1, \text{ para todo } i = 1, 2, \ldots, m.$$

Agora, definindo

$$\lambda_{max} = \max_{1 \leq i \leq m} |\lambda_i(T_k)|,$$

tem-se que $\lambda_{max} < 1$. Finalmente, pela Proposição 10,

$$\|r_{k+1}\|_2 = \|T_k r_k\|_2 \leq \lambda_{max}\|r_k\|_2 < \|r_k\|_2 \leq \|r_k\|_2, \text{ para todo } k = 0, 1, \ldots$$

□

Exemplo 5. Vejamos que não é necessariamente verdade que $\|x_1\|_2 \geq \|x_0\|_2$. Para tanto, considere o sistema $Ax = b$, em que $A := I_n \in \mathbb{C}^{n\times n}$ e $b := e_1 \in \mathbb{C}^n$, em que e_i representa o i-ésimo vetor da base canônica de $\mathbb{C}^n$. Poderíamos usar outros vetores b que teríamos o mesmo resultado, mas para simplificar vamos usar e_1.

Note que $A^\dagger = I_n = I_n^{-1}$, o que implica que $Ax = b$ tem única solução dada por $\hat{x} = A^\dagger b = b = e_1$. Além disso, veja que $\rho(I_n I_n^*) = \rho(I_n) = 1$, logo a escolha de β para o método fica condicionada a

$$0 < \beta < \frac{2}{\rho(I_n I_n^*)} = 2.$$

Também lembre que

$$U_0 = I_n - X_0 A = I_n - \beta A^* A = (1-\beta) I_n.$$

Agora, escrevendo x_0 e x_1 explicitamente (lembrando que estamos utilizando o método vetorial quadrático), temos:

$$x_0 = X_0 b = \beta A^* b = \beta I_n e_1 = \beta e_1 \text{ e}$$
$$x_1 = (I_n + U_0)x_0 = (1 + (1-\beta)) I_n (\beta e_1) = \beta(2-\beta) e_1$$

Assim, por $\|e_1\|_2 = 1$, temos que

$$\|x_0\|_2 = \beta \text{ e } \|x_1\|_2 = \beta(2-\beta).$$

Como $0 < \beta < 2$, podemos escolher $\beta := \frac{3}{2}$. Logo,

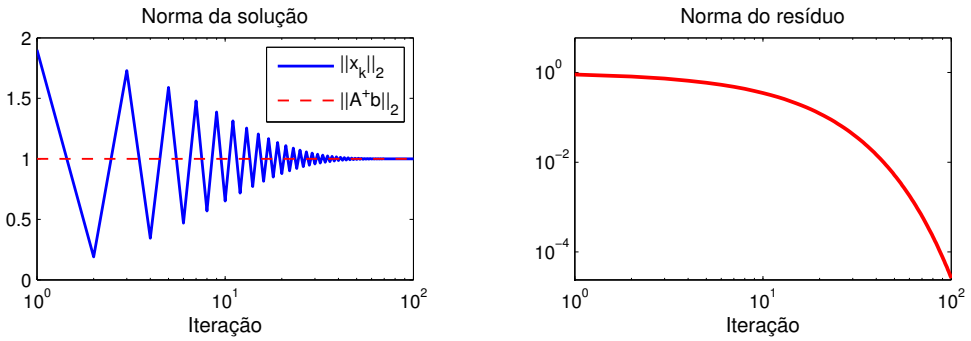
$$\|x_0\|_2 = \frac{3}{2} \text{ e } \|x_1\|_2 = \frac{3}{2}\left(2 - \frac{3}{2}\right) = \frac{3}{4}.$$

Portanto, $\|x_0\|_2 = \frac{3}{2} > \frac{3}{4} = \|x_1\|_2$, e temos o que queremos mostrar. Vale observar que, para qualquer β tal que $1 < \beta < 2$, temos o mesmo resultado.

O mesmo exemplo (isto é, $I_n x = e_1$) pode ser utilizado para ver que não é válida a monotonicidade de $\|x_k\|_2$ para nenhum k, se considerarmos o método vetorial linear dado em (4.1). No entanto, é fácil provar que a norma do resíduo é decrescente sempre.

Na Figura 11 abaixo, podemos ver oscilação na norma de x_k, mas decrescimento do resíduo r_k. Para este exemplo, utilizamos $n = 100$ e $\beta = 1.9$. Outros valores menores para β já apresentam os mesmos resultados, mas para este a diferença é mais expressiva.

Figura 11 – Comportamento de $\|x_k\|_2$ (à esquerda) e de $\|r_k\|_2$ (à direita), em cada iteração, para o método vetorial linear.



Fonte: o autor, 2015.

4.2 Contagem de Operações nos Algoritmos

Essencialmente, a contagem de operações consiste em verificar a quantidade de somas, subtrações, multiplicações, divisões, etc. efetuadas por algum algoritmo específico.

Por exemplo, se considerarmos $\alpha \in \mathbb{C}$ e $B \in \mathbb{C}^{m\times n}$, então para efetuarmos αB computamos mn operações (de fato, mn multiplicações). Seguindo a mesma ideia, se $y \in \mathbb{C}^n$, então para computar By efetuamos, em cada linha da matriz B, n multiplicações e $n-1$ somas, totalizando $2n-1$ operações. Mas como B tem m linhas, então computamos, no total, $m(2n-1)$ operações.

O mesmo raciocínio pode ser aplicado para produtos matriz-matriz, como é o caso de considerarmos o produto BC, para $C \in \mathbb{C}^{n\times p}$ e B dada acima. Assim, podemos ver que efetuamos, neste produto, $mp(2n-1)$ operações. Outras operações seguem a mesma ideia e podem ser facilmente deduzidas.

Assim, vamos efetuar a contagem de operações dos algoritmos vetoriais dados neste capítulo, a fim de analisarmos seus respectivos custos computacionais com maior facilidade. Basicamente, os algoritmos deste capítulo possuem duas partes: a primeira, de inicialização, define vetores e matrizes iniciais, além da constante β; a segunda parte é um *loop*, que depende diretamente de k.

Faremos a contagem detalhada para o Algoritmo 4, que representa o pseudo-código para o método vetorial linear. Para os outros algoritmos, a ideia é a mesma. Inicialmente, computamos a constante β. Apesar de existirem vários β possíveis, vamos fixar que $\beta = \frac{1}{\|A\|_F^2}$. Pela definição da norma de Frobenius, conseguimos computar $\|A\|_F^2$ com $2mn$ operações. Contando mais uma pela divisão, temos que β custa $2mn+1$ operações. Agora, $X_0 = \beta A^*$ e $x_0 = X_0 b$ custam, respectivamente, mn e $n(2m-1)$ operações. Para $U_0 = I - X_0 A$, efetuamos o produto $X_0 A$ (que representa $n^2(2m-1)$ operações) e uma soma de duas matrizes de ordem $n \times n$, que consistem em n^2 operações. Assim, totalizamos $n^2(2m-1)+n^2 = 2mn^2$ operações para computar U_0. Somando todos os valores até agora, temos o custo para inicialização do algoritmo:

$$2mn + 1 + mn + n(2m-1) + 2mn^2 = 5mn + 2mn^2 - n + 1.$$

Partindo para o *loop* em k, temos um produto matriz-vetor ($n(2n-1)$ operações) e uma soma de vetores (n operações), que totalizam $2n^2$. Como k inicia em 0, o custo do *loop* até a iteração k é dado por $(k+1)2n^2$. Finalmente, se denotarmos $f(m,n,k)$ a função que computa as operações do algoritmo, temos que

$$f(m,n,k) = (5mn + 2mn^2 - n + 1) + (k+1)2n^2$$

é o total de operações para o Algoritmo 4, para $A \in \mathbb{C}^{m\times n}$, $b \in \mathbb{C}^m$, e $x \in \mathbb{C}^n$, até alguma iteração k. O mesmo procedimento pode ser efetuado para os outros algoritmos deste capítulo, e a Tabela 1 abaixo contém os custos dos mesmos.

O único caso especial é para o Algoritmo 7, que possui dois *loops* que dependem de k. Para solucionar o problema, basta lembrar que

$$1 + 2 + \ldots + 2^k = \sum_{i=0}^{k} 2^i = 2^{k+1} - 1,$$

que pode ser facilmente mostrada por indução.

Tabela 1 – Contagem de operações dos algoritmos vetoriais.

Algoritmo	$f(m,n,k)$
4	$(5mn + 2mn^2 - n + 1) + (k+1)2n^2$
5	$(5mn + 4mn^2 - n^2 - n + 1) + (k+1)(4n^3 + n^2 + n)$
6	$(5mn + 2mn^2 - n + 1) + (k+1)(2n^3 + n^2)$
7	$(5mn + 2mn^2 - n + 1) + (k+1)[-2n^2 + 2n + 2^{k+1}(2n^2 - n)]$

Lembre que os três últimos algoritmos da tabela representam maneiras de computar o método vetorial quadrático. Claramente, entre os Algoritmos 5 e 6, o segundo tem vantagem, com coeficientes menores, principalmente os que acompanham k. Já entre 6 e 7, o segundo tem uma vantagem temporária, dependendo de quantas iterações k são efetuadas. À medida que crescemos k, o termo $2^{k+1}(2n^2 - n)$ ultrapassa (com certa facilidade) o termo $2n^3 + n^2$, o que torna o Algoritmo 7 difícil de aplicar, visto que n teria que assumir valores muito grandes para sair em vantagem em relação ao Algoritmo 6, ao qual temos mais controle.

Sobre o método vetorial linear, o problema se situa na quantidade de iterações para convergência, o que o torna tão caro quanto os algoritmos para o método quadrático, principalmente em problemas que apresentam maior dificuldade computacional, como os que apresentamos como problemas de teste no próximo capítulo. Para estes, a matriz A do sistema $Ax = b$ é mal condicionada, dificultando ainda mais a busca por soluções de boa qualidade.

5 Resultados Numéricos

Considere o problema $Ax = b$, em que sabemos a solução exata $\hat{x}$. Agora, seja $b_p = A\hat{x} + e$, em que e é vetor randômico gerado via a função randn. Perceba que o vetor b_p é uma perturbação do vetor b. Com isto, buscamos a solução do sistema $Ax = b_p$, para verificarmos o quanto perturbada pode ser em relação a $\hat{x}$. Na realidade, o objetivo é tentar capturar a solução do problema original a partir do problema perturbado, uma vez que na prática sempre temos os dados perturbados e não os valores exatos.

Os problemas utilizados são provenientes do pacote *Regularization Tools* (de [7]) e da Galeria do MATLAB (*MATLAB's Gallery*). São eles:

1) foxgood, 2) phillips, 3) heat, 4) shaw, 5) gravity,
6) baart, 7) deriv2, 8) moler, 9) lotkin, 10) prolate,
11) lehmer, 12) cauchy, 13) fiedler, 14) frank, 15) hilb.

A Tabela 2 abaixo mostra o número de condicionamento e o posto numérico (ambos calculados via MATLAB) das matrizes envolvidas em cada problema. O objetivo é verificar o mal condicionamento das matrizes, apesar de o posto deixar explícito que em vários casos não é possível calcular (teoricamente) a matriz inversa para cada problema, pelo fato da maioria não possuir posto completo, ou seja, $posto(A) = n$ (neste caso, $n = 1000$).

Tabela 2 – Posto numérico e número de condicionamento κ dos problemas de teste.

	Problema	*posto*	κ
1	foxgood	30	2.8795×10^{20}
2	phillips	1000	2.6415×10^{10}
3	heat	588	2.5979×10^{232}
4	shaw	20	1.1476×10^{21}
5	gravity	45	4.7467×10^{20}
6	baart	13	3.6112×10^{18}
7	deriv2	1000	1.2159×10^{6}
8	moler	999	5.0406×10^{15}
9	lotkin	22	9.644×10^{21}
10	prolate	521	4.727×10^{17}
11	lehmer	1000	1.0748×10^{6}
12	cauchy	23	1.4338×10^{21}
13	fiedler	1000	6.9481×10^{5}
14	frank	999	1.6599×10^{19}
15	hilb	24	2.5685×10^{21}

Todos são problemas conhecidos, principalmente pela sua dificuldade numérica, ou seja, são problemas mal postos. Alguns deles geram automaticamente A, x e b (os exemplos utilizados do pacote *Regularization Tools* são desta forma), enquanto que outros geram apenas a matriz A. Para estes casos, escolhemos $\hat{x}$ como a solução exata do problema shaw, e consideramos $b := A\hat{x}$.

De maneira a esclarecer a notação utilizada daqui para frente, considere δ como a perturbação causada ao vetor b, de modo a gerar o vetor b_p. Aqui, realizamos testes com δ assumindo três valores: 0.025, 0.01 e 0.001. Isto é, consideramos que o vetor b_p possui 2.5%, 1% e 0.1% de perturbação em relação ao vetor b original.

Para cada um dos problemas teste, efetuamos um número n_r de resoluções do problema, ou seja, aplicamos o algoritmo de resolução n_r vezes sobre cada um dos problemas. A intenção é verificar divergências nas soluções, a partir da análise da média dos erros e das iterações máxima e mínima de parada.

Com relação à notação nas tabelas de resultados, k_m representa a menor quantidade de iterações necessárias até o critério de parada ser atingido. Analogamente, k_M representa a maior quantidade de iterações. Como resolvemos cada problema n_r vezes, ϵ_{med} representa a média aritmética dos erros relativos. Isto é, em cada resolução $i = 1, 2, \ldots, n_r$ computamos

$$\epsilon_i = \frac{\|x_k - \hat{x}\|_2}{\|\hat{x}\|_2},$$

com k a iteração de parada, e fazemos

$$\epsilon_{med} = \frac{\epsilon_1 + \epsilon_2 + \cdots + \epsilon_{n_r}}{n_r}.$$

Por último, devemos ressaltar que os testes foram executados utilizando o método vetorial quadrático estudado na Seção 4.1. Em geral, utilizamos uma implementação do Algoritmo 6 para resolver os problemas, por efetuar menos produtos matriz-matriz que o Algoritmo 5 e por se manter constante no custo computacional por iteração, algo que não acontece com o Algoritmo 7, que pode crescer muito o custo com o crescimento de k. Não há, porém, diferença com relação à solução encontrada, visto que cada algoritmo efetua os mesmos cálculos, apenas usam processos diferentes para tanto.

Além disso, como os problemas são mal condicionados, pequenas perturbações no vetor b tendem a gerar grandes mudanças no vetor de soluções x encontrado. Portanto, precisamos introduzir critérios de parada eficientes, de maneira a escolher a "iteração certa" na qual parar, visto que estamos trabalhando com problemas instáveis, que em algum momento divergem da solução do problema original.

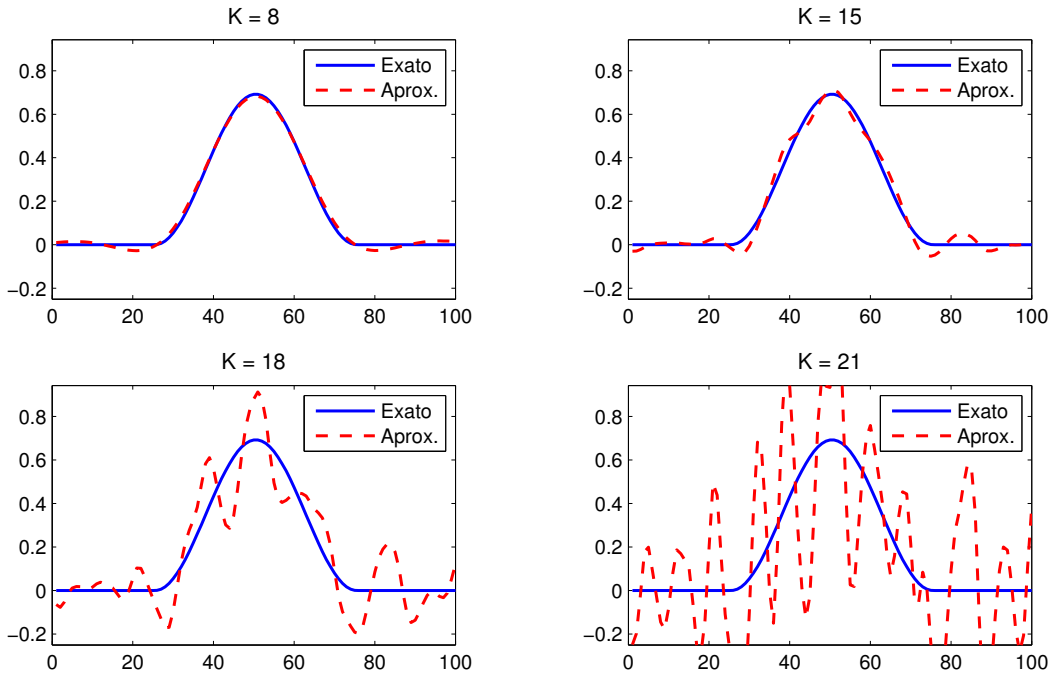
Lembre que estamos tentando capturar $\hat{x}$ solução de $Ax = b$ a partir da solução de $Ax = b_p$. Iterativamente, enquanto buscamos resolver $Ax = b_p$, os vetores x_k se aproximam de $\hat{x}$ em algumas iterações iniciais, para em seguida convergirem a $A^\dagger b_p$, que em geral pode diferir muito de $\hat{x}$. A intenção é parar o algoritmo na iteração que melhor aproxima $\hat{x}$. Para isso, os critérios de parada especiais.

Aqui, utilizaremos três critérios de parada: o princípio de discrepância, um baseado no critério da curva-L e a regra do produto mínimo. Nas seções seguintes exploraremos cada um dos casos. Vale notar que também consideramos um critério de parada por número máximo de iterações, para evitar processos sem fim. Tomamos, assim, o número máximo de iterações como sendo $k_{max} = 35$.

O Exemplo 6 abaixo apresenta uma noção da instabilidade dos problemas, quando se passa da iteração que melhor aproxima a solução do problema original.

Exemplo 6. Considere $n = 100$ e o problema phillips. Na Figura 12, temos o resultado da aplicação do método quadrático para resolver o problema com $\delta = 0.01$. Apesar de ainda não abordado, a solução encontrada satisfaz o princípio de discrepância quando $k = 8$ iterações (e $\tau = 1.1$, como veremos).

Figura 12 – Diferença entre solução exata e aproximada para problema phillips, em algumas iterações.



Fonte: o autor, 2015.

Ao efetuarmos mais iterações (isto é, ao elevarmos o valor de k), perdemos cada vez mais as características da solução do problema original, as quais estamos buscando. Portanto, precisamos que o critério de parada possa perceber esta divergência e parar na melhor iteração, antes de divergirmos da solução $\hat{x}$ do problema inicial.

5.1 Parada com Princípio de Discrepância

O *princípio de discrepância* (PD) tem como ideia básica a parada na iteração k em que

$$\|Ax_k - b_p\|_2 \leq \tau \|e\|_2,$$

com τ algum valor próximo de 1. O método é atribuído a Morozov (veja [8, 11]) e é um dos principais métodos que se baseiam no conhecimento do erro e adicionado ao vetor de dados b. Na prática, é feito uso de estimativas para o valor da perturbação; nos casos apresentados aqui, sabemos exatamente qual a perturbação adicionada em cada problema.

Como dito, para a aplicação do princípio de discrepância, τ deve ser próximo de 1. Evidentemente, seu valor não pode ser grande, pois teríamos uma solução muito grosseira, que pode não ser útil, por divergir muito da solução original. Por outro lado, se reduzirmos muito τ, perdemos as características da perturbação, o que implica em resolver somente o sistema de mínimos quadrados associado a $Ax = b_p$.

Nas Tabelas 3, 4 e 5 podemos ver os resultados numéricos para cada δ. Escolhemos $n = 1000$, ou seja, as matrizes envolvidas são todas pertencentes a $\mathbb{C}^{1000\times1000}$.

Para cada δ, efetuamos os cálculos variando τ também, assumindo valores de 0.9, 1.0, 1.1 e 1.2. Note que para $\tau = 0.9$ temos o critério por número máximo de iterações atingido em vários casos.

Tabela 3 – Resultados numéricos para PD, $\delta = 0.025$, $n = 1000$ e $n_r = 30$ resoluções.

Problema	$\tau = 0.9$		$\tau = 1.0$		$\tau = 1.1$		$\tau = 1.2$	
	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}
1	35(35)	105.88	8(11)	0.0307	7(7)	0.0580	6(6)	0.1251
2	35(35)	372.91	8(10)	0.0253	6(7)	0.0614	6(6)	0.0715
3	35(35)	218.87	12(13)	0.1105	10(10)	0.1898	9(9)	0.2363
4	35(35)	42.601	8(14)	0.1181	7(7)	0.1754	7(7)	0.1755
5	35(35)	108.69	7(10)	0.0420	6(6)	0.0701	5(5)	0.0948
6	35(35)	42.993	12(17)	0.1680	7(7)	0.3299	7(7)	0.3298
7	28(30)	91.093	11(13)	0.2567	8(8)	0.3435	7(7)	0.3754
8	30(30)	194.56	12(14)	0.1476	7(7)	0.4281	6(6)	0.4428
9	35(35)	91.984	11(13)	0.4549	6(7)	0.5011	1(1)	0.5222
10	12(12)	0.0177	11(11)	0.0238	11(11)	0.0240	11(11)	0.0240
11	27(28)	71.375	11(12)	0.1196	7(9)	0.3406	2(4)	0.4247
12	35(35)	36.034	12(14)	0.4435	10(10)	0.4611	9(9)	0.4764
13	29(31)	148.13	11(12)	0.0799	9(10)	0.2195	8(9)	0.3512
14	15(15)	0.9755	9(10)	0.0867	8(8)	0.1046	7(7)	0.1499
15	35(35)	38.352	12(14)	0.445	10(10)	0.4685	9(9)	0.4833

Tabela 4 – Resultados numéricos para PD, $\delta = 0.01$, $n = 1000$ e $n_r = 30$ resoluções.

Problema	$\tau = 0.9$		$\tau = 1.0$		$\tau = 1.1$		$\tau = 1.2$	
	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}
1	35(35)	42.241	9(14)	0.0251	8(8)	0.0319	7(7)	0.0589
2	35(35)	146.99	9(10)	0.0236	8(8)	0.0291	8(8)	0.0290
3	35(35)	87.422	13(14)	0.0785	11(12)	0.1104	11(11)	0.1447
4	35(35)	17.385	12(15)	0.0780	8(8)	0.1657	8(8)	0.1657
5	35(35)	41.143	9(11)	0.0295	7(7)	0.0552	7(7)	0.0555
6	35(35)	15.958	13(18)	0.1620	11(11)	0.2009	10(10)	0.2484
7	28(30)	40.019	13(15)	0.2173	10(10)	0.2895	9(10)	0.3132
8	29(31)	79.18	13(15)	0.0833	11(12)	0.2528	9(10)	0.3439
9	35(35)	34.371	12(15)	0.4517	9(9)	0.4736	8(9)	0.4760
10	12(12)	0.0071	12(12)	0.0071	12(12)	0.0071	12(12)	0.0071
11	27(29)	30.6	12(13)	0.0631	11(11)	0.1430	10(10)	0.2194
12	35(35)	16.389	13(15)	0.4412	11(11)	0.4540	10(10)	0.4618
13	29(31)	61.947	12(13)	0.0502	11(11)	0.1110	11(11)	0.1108
14	14(15)	0.3624	10(11)	0.0602	9(9)	0.0769	8(8)	0.1026
15	35(35)	15.051	13(16)	0.4416	12(12)	0.4476	11(11)	0.4561

Tabela 5 – Resultados numéricos para PD, $\delta = 0.001$, $n = 1000$ e $n_r = 30$ resoluções.

Problema	$\tau = 0.9$		$\tau = 1.0$		$\tau = 1.1$		$\tau = 1.2$	
	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}
1	35(35)	4.4723	15(17)	0.0076	9(9)	0.0301	9(9)	0.0301
2	35(35)	14.621	14(15)	0.0082	10(12)	0.0201	9(10)	0.0229
3	35(35)	8.4426	16(17)	0.0252	15(15)	0.0403	14(15)	0.0460
4	35(35)	1.8912	16(20)	0.0471	15(15)	0.0533	14(14)	0.0717
5	35(35)	4.1327	13(17)	0.0135	11(11)	0.0241	10(11)	0.0257
6	35(35)	1.4708	18(20)	0.1170	13(13)	0.1649	13(13)	0.1649
7	28(30)	3.7602	17(19)	0.1482	15(16)	0.1798	15(15)	0.1873
8	30(31)	7.9366	16(17)	0.0245	15(15)	0.0413	14(14)	0.0648
9	35(35)	3.049	15(21)	0.4477	13(13)	0.4513	13(13)	0.4513
10	12(12)	0.0010	12(12)	0.0010	12(12)	0.0010	12(12)	0.0010
11	27(29)	3.0715	14(15)	0.0165	13(14)	0.0336	13(13)	0.0369
12	35(35)	1.8722	17(20)	0.4393	14(15)	0.4409	14(14)	0.441
13	30(31)	6.2048	15(15)	0.0115	14(14)	0.0219	13(13)	0.0384
14	15(15)	0.0393	13(13)	0.0169	12(12)	0.0189	12(12)	0.0184
15	35(35)	1.3055	16(17)	0.4395	15(15)	0.4401	15(15)	0.4401

Algumas observações podem ser feitas sobre as tabelas acima. Diretamente, os melhores resultados são encontrados para $\tau = 1$, porém com maior custo em iterações (na maioria dos casos). Além disso, note que o método apresenta maiores dificuldades de capturar a solução exata quanto maior é a perturbação. Ou seja, quanto maior δ, menor a precisão. De certa maneira, este efeito era esperado, pois quanto mais perturbações, menores são as informações dadas ao sistema sobre o problema inicial.

É também claro que $\tau < 1$ não deve ser aplicado na prática, visto os grandes erros e a necessidade do critério de parada por iterações ser ativado, e não o princípio de discrepância. Entretanto, considerando as iterações máximas e mínimas de paradas para todos os valores de τ, em geral temos boa estabilidade nestes dois números, com um desvio padrão pequeno, isto é, k_m e k_M tem pouca variação um do outro.

Também é interessante analisar alguns dos casos específicos, como os problemas lotkin, cauchy e hilb, numerados por 9, 12 e 15, respectivamente. Estes problemas apresentam matrizes muito próximas e altamente mal condicionadas. Note que o erro foi maior para estes casos, com valores próximos um do outro, novamente apresentando a questão da estabilidade do método.

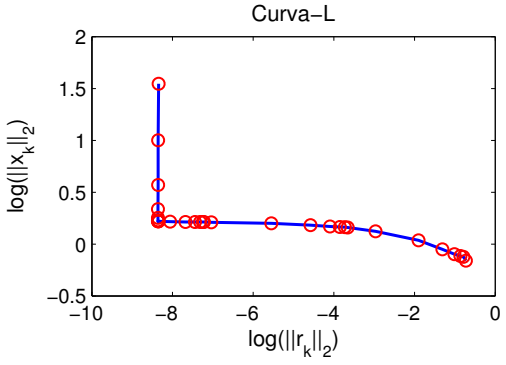
5.2 Critério de Parada com Uso de Curva-L

A curva-L consiste, de maneira grosseira, no gráfico gerado pela norma da solução e pela norma do resíduo correspondente. Sua ideia geral foi formalizada por Hansen (veja [6, 8]). Aqui, consiste na curva discreta definida pelos pontos

$$(\log(\|r_k\|_2), \log(\|x_k\|_2)),$$

para $k = 1, \ldots, k_{max}$, com k_{max} natural definido. Na Figura 13 abaixo, temos um exemplo de curva-L para o problema baart, em que consideramos dimensão $n = 500$, perturbação de 0.1% nos dados iniciais e $k_{max} = 35$. A forma da curva deixa clara a escolha para a sua nomeação.

Figura 13 – Curva-L para problema baart, com $\delta = 0.001$, $n = 500$ e $k_{max} = 35$.



Fonte: o autor, 2015.

Intuitivamente, buscamos k tal que $(\log(\|r_k\|_2), \log(\|x_k\|_2))$ esteja próximo ao canto da curva-L, pois minimiza tanto $\|r_k\|_2$ quanto $\|x_k\|_2$. Aumentando ou reduzindo k a partir deste ponto, temos aumento em alguma das normas, por isso a parada no canto. A pergunta é: como escolher k? Mesmo no exemplo acima, percebemos um grande acúmulo de iterações nas proximidades da esquina da curva, o que aumenta as dificuldades. Além disso, nem sempre a curva terá forma tão clara e que permita uma boa ideia quanto à iteração de parada.

Uma das abordagens aqui utilizadas na busca pela parada na iteração correta é utilizando o programa corner, que faz parte do pacote *Regularization Tools*, idealizado por Hansen em [7]. O programa é complexo e não é foco de estudo aqui, apenas o seu uso nos é interessante no momento. O programa recebe os valores de $\|r_k\|_2$ e $\|x_k\|_2$, necessários à construção da curva-L, e retorna a iteração próxima da esquina da curva. Porém, para enviarmos os dados necessitamos efetuar várias iterações, o que certamente é um problema para dimensões maiores, devido ao custo do método quadrático.

Abaixo, temos uma tabela com os resultados da aplicação do programa corner. Aqui, k_{max} é utilizado para sinalizar a quantidade de iterações efetuadas, ou seja, a quantidade de dados fornecidas ao programa. Utilizamos $k_{max} = 35$, o que informa que enviamos $\|r_1\|_2, \|r_2\|_2, \ldots, \|r_{35}\|_2$ e $\|x_1\|_2, \|x_2\|_2, \ldots, \|x_{35}\|_2$. Além disso, n_r continua representando o número de vezes que cada um dos problemas foi resolvido.

Tabela 6 – Resultados numéricos para programa corner, $n = 1000$, $k_{max} = 35$ e $n_r = 30$ resoluções.

	Problema	$\delta = 0.025$		$\delta = 0.01$		$\delta = 0.001$	
		$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}
1	foxgood	8(8)	0.0321	9(13)	0.0250	16(20)	0.0131
2	phillips	9(13)	0.0281	14(16)	0.0441	21(22)	0.0688
3	heat	12(12)	0.1098	13(16)	0.0713	22(22)	0.0755
4	shaw	8(11)	0.1603	13(14)	0.0802	17(21)	0.0459
5	gravity	10(12)	0.0367	12(15)	0.0306	20(21)	0.0252
6	baart	11(11)	0.2014	12(12)	0.1712	20(20)	0.1166
7	deriv2	9(11)	0.2874	12(14)	0.2270	20(21)	0.1524
8	moler	10(10)	0.3449	12(13)	0.1955	20(20)	0.0705
9	lotkin	9(9)	0.4735	12(12)	0.4530	19(20)	0.4468
10	prolate	4(21)	0.5306	4(23)	0.2775	4(30)	0.6512
11	lehmer	10(10)	0.2256	12(12)	0.0820	20(20)	0.1025
12	cauchy	10(13)	0.4582	13(14)	0.4420	18(22)	0.4394
13	fiedler	10(11)	0.1776	12(12)	0.0668	21(21)	0.0964
14	frank	11(12)	0.1688	14(14)	0.2352	22(34)	1.8986
15	hilb	12(12)	0.4479	13(15)	0.4405	19(22)	0.4393

É preciso notar que o programa corner necessita de monotonicidade em $\|r_k\|_2$ e $\|x_k\|_2$ para funcionar. Para tanto, o Teorema 14 nos garante que $\|r_k\|_2$ é decrescente

e $\|x_k\|_2$ é crescente. Como o crescimento de $\|x_k\|_2$ é somente válido para $k \geq 1$, então enviamos os dados a partir de $k = 1$.

5.3 Regra do Produto Mínimo

Uma outra abordagem foi apresentada recentemente em [2] e [4], na chamada *regra do produto mínimo* (MPR). No trabalho, é apresentada uma maneira heurística (tal qual o critério da curva-L) para a busca da iteração de parada k, que se baseia na ideia de minimizar a função discreta

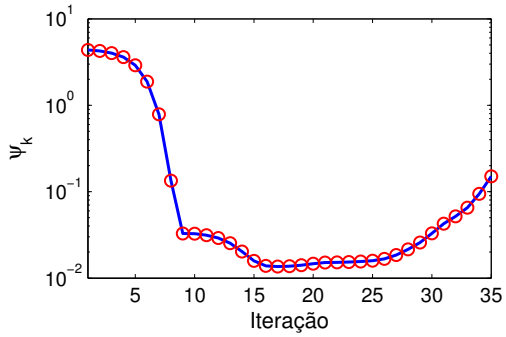
$$\psi_k = \|r_k\|_2 \|x_k\|_2,$$

isto é, buscamos $\hat{k}$ tal que

$$\hat{k} = \arg\min \psi_k.$$

Na Figura 14 abaixo, apresentamos um exemplo da função ψ_k para o problema foxgood, com $n = 500$, perturbação de 0.1% nos dados iniciais e $k_{max} = 35$. Note que mesmo visualmente temos dificuldade em encontrar a iteração $\hat{k}$ que minimiza ψ_k.

Figura 14 – Função ψ_k para problema foxgood, com $\delta = 0.001$, $n = 500$ e $k_{max} = 35$.



Fonte: o autor, 2015.

Assim como para o programa corner, vamos apenas aplicar o MPR, sem nos preocuparmos com seu funcionamento. Novamente, necessitamos de enviar os dados para construção da função ψ_k, isto é, $\|r_k\|_2$ e $\|x_k\|_2$. Este método também pede monotonicidade das normas de r_k e x_k, que garantimos pelo Teorema 14.

A Tabela 7 abaixo apresenta os resultados, sob as mesmas condições que para a aplicação de corner.

Sobre estas últimas duas tabelas (isto é, Tabelas 6 e 7), os resultados são próximos aos encontrados na aplicação do princípio de discrepância, com iterações e precisão aumentando de acordo com o decrescimento da perturbação δ. Com relação aos erros, em alguns casos corner captura a solução com mais qualidade, enquanto que em outros MPR tem melhor desempenho.

Tabela 7 – Resultados numéricos para MPR, $n = 1000$, $k_{max} = 35$ e $n_r = 30$ resoluções.

	Problema	$\delta = 0.025$		$\delta = 0.01$		$\delta = 0.001$	
		$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}	$k_m(k_M)$	ϵ_{med}
1	foxgood	8(8)	0.0321	9(9)	0.0301	9(9)	0.0301
2	phillips	9(9)	0.0246	9(9)	0.0239	13(13)	0.0110
3	heat	12(12)	0.1098	13(14)	0.0661	16(17)	0.0291
4	shaw	9(11)	0.1511	13(13)	0.0994	16(16)	0.0477
5	gravity	9(10)	0.0371	10(11)	0.0306	12(13)	0.0175
6	baart	11(11)	0.2014	12(12)	0.1712	14(14)	0.1634
7	deriv2	10(10)	0.2920	13(13)	0.2253	17(18)	0.149
8	moler	10(10)	0.3449	12(12)	0.1981	17(17)	0.0251
9	lotkin	9(9)	0.4735	12(12)	0.4530	16(18)	0.4490
10	prolate	12(14)	0.0179	12(12)	0.0071	13(13)	0.0009
11	lehmer	10(10)	0.2256	12(12)	0.0820	16(20)	0.0692
12	cauchy	11(12)	0.4493	13(14)	0.4425	16(16)	0.4395
13	fiedler	10(11)	0.1890	12(13)	0.0648	15(15)	0.0114
14	frank	11(11)	0.1279	14(14)	0.2352	25(35)	1.9183
15	hilb	12(12)	0.4479	13(14)	0.4438	16(16)	0.4396

Vale notar que corner e MPR também não melhoram a solução encontrada para os problemas em que a matriz de Hilbert está envolvida. Além disso, note que corner tem mal funcionamento para o problema prolate, numerado por 10, algo que não ocorre com MPR ou PD. Finalmente, vemos dificuldade no problema frank, numerado por 14, para o programa corner e para MPR, especialmente no caso $\delta = 0.001$, dificuldade não encontrada na aplicação do princípio de discrepância.

6 Conclusão

De forma sucinta, este trabalho pode ser dividido em três partes: a primeira, relacionada diretamente à matriz pseudo-inversa e suas propriedades; uma relacionada aos métodos iterativos para o cálculo da pseudo-inversa; e, finalmente, uma com aplicações dos métodos iterativos na construção de soluções aproximadas para sistemas de equações lineares. As duas primeiras são fundamentais ao estudo e já são bem desenvolvidas na literatura (principalmente a primeira). Já a terceira parte é o trabalho propriamente dito: é nesta parte que aplicamos os conceitos dos capítulos anteriores em uma nova estrutura.

As sequências vetoriais apresentam resultados promissores, mas ainda de certa maneira brutos: é preciso mais estudo e desenvolvimento, principalmente no que diz respeito ao custo computacional dos métodos. Por outro lado, os métodos desenvolvidos são robustos, podendo ser aplicados a qualquer sistema linear, com convergência teórica garantida, e de boa qualidade, se considerarmos o método quadrático.

Esta abordagem para a construção de sequências vetoriais a partir de métodos iterativos para a matriz pseudo-inversa é inédita na literatura, e abre portas para muitos estudos e possibilidades, que aqui não foi possível abordar. De fato, para trabalhos futuros, temos ideias de combinar os métodos estudados/desenvolvidos neste projeto com técnicas de projeção em subespaços de Krylov, como tentativa de reduzir o custo operacional dos algoritmos. Outro assunto de interesse é o estudo das mesmas técnicas e métodos no ambiente dos espaços de Hilbert, visando resolver equações que envolvem operadores lineares.

Referências

[1] ALLAIRE, G.; KABER, S. M. **Numerical Linear Algebra**. New York: Springer, 2008. 271 p. (Texts in Applied Mathematics, v. 55).

[2] BAZÁN, F. S. V.; CUNHA, M. C.; BORGES, L. S. Extension of GKB-FP algorithm to large-scale general-form Tikhonov regularization. **Comput. Appl. Math.**, 2013.

[3] BEN-ISRAEL, A.; GREVILLE, T. N. E. **Generalized Inverses**: Theory and Applications. 2nd. ed. New York: Springer, 2003. 420 p.

[4] BORGES, L. S.; BAZÁN, F. S. V.; CUNHA, M. C. Automatic stopping rule for iterative methods in discrete ill-posed problems. **Comp. Appl. Math., DOI 10.1007/s40314-014-0174-3**, 2014.

[5] DEMMEL, J. W. **Applied Numerical Linear Algebra**. Philadelphia: SIAM, 1997. 421 p.

[6] HANSEN, P. C. Analysis of discrete ill-posed problems by means of the L-curve. **SIAM Rev.**, v. 34, p. 561–580, 1992.

[7] HANSEN, P. C. Regularization Tools: A MATLAB package for analysis and solution of discrete ill-posed problems. **Numerical Algorithms**, v. 6, p. 1–35, 1994.

[8] KILMER, M. E.; O'LEARY, D. P. Choosing regularization parameters in iterative methods for ill-posed problems. **SIAM J. Matrix Anal. Appl.**, v. 22, n. 4, p. 1204–1221, 2001.

[9] MEYER, C. D. **Matrix Analysis and Applied Linear Algebra**. Philadelphia: SIAM, 2000. 718 p.

[10] MOORE, E. H. On the reciprocal of the general algebraic matrix. **Bull, Amer. Math. Soc.**, v. 26, p. 394–395, 1920.

[11] MOROZOV, V. A. On the solution of functional equations by the method of regularization. **Soviet Math. Dokl.**, v. 7, p. 414–417, 1966.

[12] PENROSE, R. A generalized inverse for matrices. **Proc. Cambridge Phil. Soc.**, v. 51, p. 406–413, 1955.

[13] PETKOVIĆ, M. D.; STANIMIROVIĆ, P. S. Iterative method for computing the Moore-Penrose inverse based on Penrose equations. **J. Comput. Appl. Math.**, v. 235, p. 1604–1613, 2011.

[14] ZONTINI, D. D. **Métodos Computacionais para Inversas Generalizadas**. Tese (Doutorado) — Universidade Federal do Paraná, Curitiba, 2014.