O MÉTODO DO GRADIENTE CONJUGADO COM PRODUTO INTERNO GERAL

Um dos requisitos do Mestrado em Matemática Aplicada

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE MATEMÁTICA
DEPARTAMENTO DE MATEMÁTICA PURA E APLICADA
CONVÊNIO COM A UNIVERSIDADE DE CAXIAS DO SUL

O MÉTODO DO GRADIENTE CONJUGADO COM PRODUTO INTERNO GERAL

DISSERTAÇÃO DE MESTRADO

VÂNIA MARIA PINHEIRO SLAVIERO

ORIENTADOR: PROF. DR. OCLIDE JOSÉ DOTTO

PORTO ALEGRE, DEZEMBRO DE 1997

Conteúdo

Resumo

O método do gradiente conjugado, na sua forma geral, pode ser aplicado a um sistema de equações lineares algébricas **Ax = b**, quando **A** é autoadjunta e positiva definida em relação a um produto interno qualquer. As formas de recorrência de dois termos ou três, que fornecem uma aproximação da solução do sistema, independem do produto interno fixado no espaço universo. A generalidade teórica envolvida em tal contexto encontra-se, nesse trabalho, devidamente justificada. O precondicionamento e a sua relação com o produto interno utilizado, e o método para SELAS singulares e quase singulares também fazem parte da exposição.

Abstract

The conjugated gradient method, in its general form, can be applied on an algebraic linear system **Ax = b**, when **A** is selfadjoint and positive definite with respect to an arbitrary inner product. The three-term recurrence form and the two-term one that give an approximation to the solution of the system do not depend on the inner product in the environment space. The theoretical generality involved in that context is properly justified in this dissertation. The preconditioning and its relationship with the relevant inner product and the conjugate gradient method for the singular and nearly singular systems are also part of this work.

Introdução

Era verão de 1949. O Professor Wassily Leontief da Universidade de Harvard estava introduzindo o último cartão perfurado no computador da universidade, MARK II. Os cartões continham informações econômicas a respeito da economia dos Estados Unidos e representavam um resumo de mais de 250 000 itens de informação, produzidos pelo *Bureau of Labor Statistics* desse país, resultado de um trabalho intenso que levou dois anos. Leontief dividiu a economia dos Estados Unidos em 500 setores, tais como indústria do carvão, indústria automotiva, comunicações, etc. Para cada setor, escreveu uma equação linear que descrevia como o setor distribuía sua produção para o outro setor da economia. Como o MARK II, um dos maiores computadores daqueles dias, não podia manejar o sistema resultante de 500 equações com 500 incógnitas, Leontief simplificou o sistema, reduzindo-o a 42 equações com 42 incógnitas .

Para programar o MARK II para as 42 equações, foram necessários vários meses de esforço, e Leontief estava ansioso para ver quanto tempo o computador levaria para resolver o problema. O MARK II resmungou e piscou durante 56 horas, até que produziu a solução.

Leontief, que recebeu em 1973 o Prêmio Nobel de Economia, abriu a porta para uma nova era na modelagem matemática na economia. Seus esforços em Harvard em 1949 indicaram um dos usos significativos do computador para analisar o que era então um modelo matemático em grande escala. Desde então os pesquisadores em muitos outros campos empregaram computadores para analisar modelos matemáticos. Por causa da grande quantidade de dados envolvidos, os modelos são normalmente *lineares*, isto é, são descritos por Sistemas de Equações Lineares AlgébricaS (SELAS). Há estimativas que apontam para o fato de que, a cada quatro problemas de simulação em matemática, três convertem-se em solução de SELAS [29].

A importância da Álgebra Linear nas aplicações cresceu em proporção direta com a capacidade computacional, demandando, após cada nova geração de *hardware* e *software*, capacidade ainda maior. É por isso que a ciência da computação está imbricada com a Álgebra Linear através do crescimento explosivo do processamento paralelo e cálculos em grande escala.

Cientistas e engenheiros agora trabalham em problemas muito mais complexos do que os sonhados algumas décadas passadas. Em 1974, nos Estados Unidos, a *National Geodetic Survey* montou um projeto, que levaria, segundo estimativa, 10 anos para executá-lo, com o fim de atualizar dados da *North American Datum*. Tal projeto visou a construir um mapa completo e preciso dos Estados Unidos e resultaria num SELAS sobredeterminado de 1 800 000 equações com 900 000 incógnitas. As medidas foram coleccionadas durante 140 anos e corrigidas ao máximo e adaptadas para entrar no computador. Os cálculos finais envolveram em torno de 1 800 000 observações, cada uma ponderada de acordo com seu erro relativo e originou uma equação. A solução dos mínimos quadrados foi achada resolvendo um SELAS quadrado de 928 735 equações. O banco de dados ficou completo em 1983 e o computador levou 940 horas para resolver o maior SELAS até a data.

Muitas técnicas ou métodos foram desenvolvidos com propósito de resolver SELAS. Respeitando a característica de cada um, é comum catalogá-los em dois grupos: *métodos diretos* e *métodos iterativos*. Os primeiros conduzem à solução exata (a menos de erros de arredondamento) após um número finito de passos, e os segundos se baseiam na construção de seqüências de aproximações, onde, em cada passo, são usados valores calculados anteriormente para melhorar a aproximação. Esses últimos ainda se classificam em: *métodos estacionários*, que são fáceis de compreender e implementar, mas normalmente não são muito eficientes; e *métodos não-estacionários*, que diferem dos anteriores em que os cálculos

envolvem informações que mudam a cada iteração, são mais sofisticados e adequados para as necessidades hodiernas de SELAS de grande porte. Nessa última classe encontramos uma variedade de processos sob a designação geral de **M**étodos do **G**radiente **C**onjugado (MGC), considerados hoje muito úteis, particularmente para SELAS esparsos, superando em eficiência e rapidez métodos clássicos como o SOR.

É sobre o MGC que versa esta dissertação. O processo do gradiente conjugado envolve de maneira essencial um produto interno. Normalmente usamos o produto interno canônico, também dito usual, mas hoje é importante, por exemplo no precondicionamento, que possamos utilizar produtos internos mais gerais. O mérito desse trabalho consiste precisamente em apresentar a validade do MGC acoplado a um produto interno geral, ao qual corresponde ortogonalidade não-usual de vetores.

Consideremos um SELAS $\mathbf{Ax} = \mathbf{b}$ com n equações e n incógnitas e um produto interno $\langle \bullet, \bullet \rangle$ em $\mathbb{R}^n$, definido por uma matriz simétrica positiva definida **W**, chamado de *matriz de ponderação*. Esse produto interno [08] é dado por

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t\mathbf{W}\mathbf{y}, \text{ para quaisquer } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

Aqui o índice superior t indica transposição e o estilo negrito é usado para simbolizar matrizes e vetores. Supomos que a matriz **A** dos coeficientes do SELAS seja *autoadjunta* em relação ao produto interno $\langle \bullet, \bullet \rangle$, o que significa que

$$\langle \mathbf{x}, \mathbf{Ay} \rangle = \langle \mathbf{Ax}, \mathbf{y} \rangle, \text{ para quaisquer } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

Além disso, admitimos que **A** seja *positiva definida* em relação a $\langle \bullet, \bullet \rangle$, isto é,

$$\langle \mathbf{x}, \mathbf{Ax} \rangle > \mathbf{0}, \text{ se } \mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n.$$

As expressões *autoadjunta positiva definida* e *simétrica positiva definida* serão abreviadas, respectivamente, com a. p. d. e s. p. d.

Uma matriz **A** é a. p. d. em relação ao produto interno $\langle \bullet, \bullet \rangle$, representado por uma matriz s. p. d. **W**, se e somente se $\mathbf{A}^t\mathbf{W} = \mathbf{WA}$ e **WA** é positiva definida. De fato, ser $\langle \mathbf{x}, \mathbf{Ay} \rangle = \langle \mathbf{Ax}, \mathbf{y} \rangle$ para quaisquer **x** e **y** aqui significa $\mathbf{x}^t\mathbf{WAy} = \mathbf{x}^t\mathbf{A}^t\mathbf{Wy}$ para quaisquer **x** e **y**, e ainda, equivalentemente, $\mathbf{A}^t\mathbf{W} = \mathbf{WA}$; segue, e vice-versa, que **WA** é simétrica. Além disso, requerer que seja $\langle \mathbf{x}, \mathbf{Ax} \rangle > 0$ para todo $\mathbf{x} \neq 0$, eqüivale a requerer que **WA** seja positiva definida.

Também, se **A** é a. p. d. em relação ao produto interno $\langle \bullet, \bullet \rangle$, então $\mathbf{A}^{-1}$ é a. p. d. em relação ao mesmo produto interno. A prova dessa implicação é imediata, pois

$$\mathbf{WA} = \mathbf{A}^t\mathbf{W} \Rightarrow (\mathbf{WA})^{-1} = (\mathbf{A}^t\mathbf{W})^{-1} \Rightarrow \mathbf{A}^{-1}\mathbf{W}^{-1} = \mathbf{W}^{-1}(\mathbf{A}^t)^{-1} \Rightarrow \mathbf{WA}^{-1}\mathbf{W}^{-1} = (\mathbf{A}^{-1})^t \Rightarrow \mathbf{WA}^{-1} = (\mathbf{A}^{-1})^t\mathbf{W},$$

isto é, $\mathbf{A}^{-1}$ é autoadjunta em relação ao produto interno $\langle \bullet, \bullet \rangle$. Como **A** é não singular, para $\mathbf{x} \neq \mathbf{0}$, existe **y**, necessariamente não nulo, tal que $\mathbf{x} = \mathbf{Ay}$. Então,

$$\langle \mathbf{x}, \mathbf{A}^{-1}\mathbf{x} \rangle = \mathbf{x}^t\mathbf{WA}^{-1}\mathbf{x} = (\mathbf{Ay})^t\,\mathbf{WA}^{-1}(\mathbf{Ay}) = (\mathbf{Ay})^t\,\mathbf{WAy} = \langle \mathbf{Ay}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{Ay} \rangle > 0,$$

ou seja, $\mathbf{A}^{-1}$ é positiva definida em relação ao produto interno $\langle \bullet, \bullet \rangle$.

É imediato que se $\mathbf{W} = c\mathbf{I}$ é uma matriz escalar com $c > 0$, então **A** é a. p. d. em relação ao produto interno representado por **W** se e somente se **A** é s. p. d. Nesse caso $c\mathbf{A}$ e **W** definem o mesmo produto interno. Em particular, ser **A** a. p. d. em relação ao produto interno usual é o mesmo que ser s. p. d.

É importante observar que o fato de uma matriz **A** ser a. p. d. em relação a um produto interno não implica que **A** seja s. p. d., e nem mesmo simétrica. Um simples exemplo comprova isso: tomemos

$$\mathbf{W}:=\begin{bmatrix}1 & 1\\ 1 & 2\end{bmatrix} \quad \text{e} \quad \mathbf{A}=\begin{bmatrix}2 & 2\\ 1 & 2\end{bmatrix}; \text{ teremos } \mathbf{A}^{t}\mathbf{W} = \mathbf{AW} = \begin{bmatrix}3 & 4\\ 4 & 6\end{bmatrix},$$

ou seja, a simetria não é mantida.

Feitas as considerações iniciais, seguimos com uma breve explanação sobre o princípio do MGC. Os MGC podem ser vistos como métodos iterativos que minimizam o funcional quadrático f, definido por

$$f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{Ax}\rangle - \langle \mathbf{x}, \mathbf{b}\rangle, \tag{I.1}$$

ou a *função residual*, dada por

$$F(\mathbf{x}) = \frac{1}{2}\langle \mathbf{Ax-b}, \mathbf{A}^{-1}(\mathbf{Ax-b})\rangle. \tag{I.2}$$

Estamos supondo que **A** seja a. p. d. em relação ao produto interno $\langle \bullet, \bullet \rangle$. Minimizar (I.2) é equivalente a minimizar (I.1), pois,

$$\begin{aligned}
F(\mathbf{x}) &= \frac{1}{2}\langle \mathbf{Ax-b}, \mathbf{A}^{-1}(\mathbf{Ax-b})\rangle \\
&= \frac{1}{2}(\mathbf{Ax-b})^{t}\mathbf{WA}^{-1}(\mathbf{Ax-b}) \\
&= \frac{1}{2}[(\mathbf{Ax})^{t}\mathbf{WA}^{-1}(\mathbf{Ax}) - (\mathbf{Ax})^{t}\mathbf{WA}^{-1}\mathbf{b} - \mathbf{b}^{t}\mathbf{WA}^{-1}(\mathbf{Ax}) + \mathbf{b}^{t}\mathbf{WA}^{-1}\mathbf{b}] \\
&= \frac{1}{2}[\mathbf{x}^{t}\mathbf{A}^{t}\mathbf{Wx} - \mathbf{x}^{t}\mathbf{A}^{t}\mathbf{WA}^{-1}\mathbf{b} - \mathbf{x}^{t}\mathbf{A}^{t}\mathbf{WA}^{-1}\mathbf{b} + \mathbf{b}^{t}\mathbf{WA}^{-1}\mathbf{b}] \\
&= \frac{1}{2}\langle \mathbf{x}, \mathbf{Ax}\rangle - \langle \mathbf{x}, \mathbf{b}\rangle + \frac{1}{2}\langle \mathbf{b}, \mathbf{A}^{-1}\mathbf{b}\rangle \\
&= f(\mathbf{x}) + c,
\end{aligned}$$

onde c é constante.

O mínimo estrito e global de f ocorre na solução $\bar{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$ do SELAS $\mathbf{Ax} = \mathbf{b}$, pois, para $\mathbf{d}\in\mathbb{R}^{n}$, indicando com $\nabla f(\mathbf{x})$ o gradiente de f,

$$\begin{aligned}
\langle \nabla f(\mathbf{x}), \mathbf{d}\rangle &= \frac{1}{2}\big(\langle \mathbf{x}, \mathbf{Ad}\rangle + \langle \mathbf{d}, \mathbf{Ax}\rangle\big) - \langle \mathbf{b}, \mathbf{d}\rangle \\
&= \langle \mathbf{Ax}, \mathbf{d}\rangle + \langle -\mathbf{b}, \mathbf{d}\rangle \\
&= \langle \mathbf{Ax} - \mathbf{b}, \mathbf{d}\rangle.
\end{aligned}$$

Por aí vemos que: $\langle \nabla f(\mathbf{x}), \mathbf{d}\rangle = 0$ para todo $\mathbf{d}\in\mathbb{R}^{n} \Leftrightarrow \mathbf{Ax} = \mathbf{b}$. O fato de o mínimo de f ser estrito e global decorre de **A** ser a. p. d. em relação ao produto interno em questão.

A minimização ocorre sobre transladados de certos espaços de vetores, chamados *espaços de Krylov*, de dimensão crescente, definidos recursivamente por:

$$\mathscr{K}_{k} := \mathscr{K}_{k-1} \oplus \mathscr{G}(\mathbf{A}^{k}\mathbf{r}_{o}),$$

onde indicamos com $\mathscr{G}(S)$ o subespaço de $\mathbb{R}^{n}$ gerado por um conjunto S de vetores, e a operador $\oplus$ indica soma direta. Aqui $\mathscr{K}_{o} := \mathscr{G}(\mathbf{r}_{o})$, e $\mathbf{r}_{o} := \mathbf{Ax}_{o} - \mathbf{b}$ é o *resíduo* do vetor $\mathbf{x}_{o}$, este uma aproximação inicial da

solução $\mathbf{x}$ do SELAS. Conseqüentemente, $\mathscr{K}_k$ é gerado pelos vetores $\mathbf{r}_o$, $\mathbf{A}\mathbf{r}_o$, ..., $\mathbf{A}^k\mathbf{r}_o$, é determinado por $\mathbf{r}_o$ e $\mathbf{A}$, e, por isso, escrevemos

$$\mathscr{K}_k(\mathbf{r}_o,\mathbf{A}) := \mathscr{G}(\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^k\mathbf{r}_o).$$

Para a busca do mínimo do funcional (I.1) ou, equivalentemente, de (I.2), procedemos, nos MGC, da seguinte forma: conhecida uma aproximação inicial $\mathbf{x}_o$ da solução $\mathbf{x}$ do SELAS $\mathbf{A}\mathbf{x} = \mathbf{b}$, calculamos

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{p}_k,$$

de modo que $f(\mathbf{x}_{k+1}) < f(\mathbf{x}_k)$. As escolhas do parâmetro α_k (real) e do vetor $\mathbf{p}_k$ vão definir os diferentes métodos.

No trabalho que estamos apresentando, a escolha é feita de forma que a seqüência $(\mathbf{p}_k)$ constitua uma base ortogonal, ou conjugadamente ortogonal, do subespaço $\mathscr{K}_k$, em relação a um produto interno $\langle \bullet, \bullet \rangle$. *Conjugadamente ortogonal*, assim como *A-ortogonal*, significa

$$\langle \mathbf{p}_i, \mathbf{A}\mathbf{p}_j \rangle = 0, \text{ se } i \neq j.$$

No primeiro caso, ou seja, quando a seqüência $(\mathbf{p}_k)$ forma uma base ortogonal de $\mathscr{K}_k$, se escolhermos $\mathbf{p}_k := \mathbf{r}_k$, onde $\mathbf{r}_k := \mathbf{A}\mathbf{x}_k - \mathbf{b}$ denota o *$k^{\text{ésimo}}$ resíduo*, a construção dessa base é feita concomitantemente com o cálculo das aproximações $\mathbf{x}_{k+1}$, definidas através de uma forma de recorrência que envolve três termos. Essa forma de recorrência para o cálculo das aproximações $\mathbf{x}_{k+1}$ pode envolver apenas dois termos, se a escolha for $\mathbf{p}_k := \mathbf{d}_k$, de modo que a seqüência $(\mathbf{d}_k)$ forme uma base $\mathbf{A}$-ortogonal de $\mathscr{K}_k$, o que constitui o segundo caso aqui abordado.

O capítulo 1 é dedicado à forma de recorrência de três termos do MGC. Nele expomos como definir indutivamente a seqüência das aproximações $(\mathbf{x}_k)$ que converge para a solução de um SELAS $\mathbf{A}\mathbf{x} = \mathbf{b}$, apresentamos o algoritmo do método e sua complexidade computacional, e discutimos o método com precondicionamento. Lemas e teoremas, propostos e demonstrados, reúnem resultados. Como exemplo, citamos o Lema 1.2, que enuncia e prova que, sob certas condições iniciais, o conjunto $\{\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k\}$ é uma base ortogonal de $\mathscr{K}_k$. Observamos que a construção da forma de recorrência de três termos do MGC é baseada nesse fato, o que justifica a existência e a importância do lema, que é de nossa autoria.

No capítulo 2 a forma de recorrência de dois termos, também chamada forma padrão, ou ainda forma de recorrência curta do método, é apresentada, sempre com um produto interno geral. Ela surge de uma conveniente escolha de n direções linearmente independentes, ou, mais especificamente, de direções $\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}$ $\mathbf{A}$-ortogonais, na direção das quais a seqüência das aproximações $(\mathbf{x}_k)$ é construída concomitantemente. De forma semelhante ao capítulo 1, os aspectos mais relevantes do método são desenvolvidos e apresentados em forma de lemas e teoremas. Dentre esses, destacamos dois, de nossa autoria, de grande importância dentro do contexto: o Teorema 2.4.1 que prova o seguinte: se $\mathbf{d}_0 := -\mathbf{r}_o$, então o *$k^{\text{ésimo}}$* espaço de Krylov é gerado tanto pelas $k + 1$ primeiras direções de procura quanto pelos $k + 1$ resíduos; o Teorema 2.4.2 que prova a $\mathbf{A}$-ortogonalidade da seqüência $(\mathbf{d}_k)$ dos vetores de direção, sob condições preestabelecidas. A complexidade computacional da forma de recorrência de dois termos e uma comparação com a de três termos, algoritmos para a forma padrão, com e sem precondicionamento, também são assuntos tratados nesse capítulo. Através da comparação entre as formas, uma desenvolvida no primeiro capítulo e outra nesse, veremos que, na forma de recorrência curta, o número de operações realizadas por iteração é menor que na de três termos, o que torna a escolha das direções $\mathbf{A}$-ortogonais particularmente interessante. O Exemplo 2.10 comprova essa afirmação. Para a ilustração do referido exemplo foi feita a implementação computacional do método no MATLAB para as formas de recorrência de três e dois termos pelos *m-files mgc3t* e *mgc2t*, respectivamente, que se encontram em apêndice.

Um dos aspectos teóricos do método, discutido na secção 2.6, é a propriedade de terminação finita, que consiste no seguinte: teoricamente, o MGC fornece a solução exata de um SELAS $\mathbf{A}\mathbf{x} = \mathbf{b}$ em, no máximo, n iterações, sendo n a ordem de $\mathbf{A}$. Esse fato até classificou o mesmo, na década de 60, como

um método direto. Contudo, na presença de erros de arredondamento, os vetores gerados não são exatamente ortogonais (ou **A**-ortogonais), e o método pode necessitar mais de n iterações para alcançar a precisão numérica de máquina onde ele é calculado. Mas, e isto é muito importante, pois os SELAS provindos do mundo real costumam ser grandes, veremos que o MGC pode gerar boas aproximações do vetor solução, em menos que n passos.

O capítulo 3 trata de SELAS $\mathbf{Ax} = \mathbf{b}$ singulares e quase singulares, considerando **A**, no primeiro caso, simétrica positiva semidefinida, isto é, além de simétrica,

$$\langle \mathbf{x}, \mathbf{Ax} \rangle \geq \mathbf{0}, \text{ para todo } \mathbf{x} \in \mathbb{R}^n,$$

onde $\langle \bullet, \bullet \rangle$ denota aqui o produto interno usual. No segundo caso, **A** é tomada s. p. d. e a expressão *quase singular* significa que **A** tem um ou mais autovalores muito próximos de zero.

Se o SELAS é singular e inconsistente, podemos resolver, por exemplo, as equações normais que traduzem SELAS consistentes. Nesta última condição e sendo **A** simétrica, o MGC, e aqui, por questão de eficiência, nos referimos à forma de recorrência curta, fornece uma solução particular $\tilde{\mathbf{x}}$ de $\mathbf{Ax} = \mathbf{b}$, e aplicado sobre o sistema homogêneo associado fornece uma solução **s** no núcleo de **A**, de forma que todas as soluções **x** são obtidas por $\mathbf{x} = \tilde{\mathbf{x}} + t\mathbf{s}$, $t \in \mathbb{R}$.

No caso de o SELAS ser quase singular, se houver apenas um autovalor λ_1 próximo de zero, associado a um autovetor $\mathbf{v}_1$, podemos escrever sua solução $\tilde{\mathbf{x}}$ na forma

$$\tilde{\mathbf{x}} = \frac{\mathbf{b}^t \mathbf{v}_1}{\lambda_1} \mathbf{v}_1 + \tilde{\mathbf{x}}_d,$$

onde $\tilde{\mathbf{x}}_d$ é expresso como combinação linear dos demais autovetores.

Multiplicando ambos os membros por **A**, obtemos:

$$\mathbf{A}\tilde{\mathbf{x}}_d = \mathbf{c},$$

onde $\mathbf{c} := \mathbf{b} - (\mathbf{b}^t \mathbf{v}_1)\mathbf{v}_1$. Podemos, conhecendo o autopar $(\lambda_1, \mathbf{v}_1)$, determinar o vetor **c** e aplicar o MGC a esse último sistema para obter a solução $\tilde{\mathbf{x}}_d$. A solução $\tilde{\mathbf{x}}$ fica então completamente determinada.

Para resolver SELAS quase singulares são discutidos dois métodos: o método do sistema aumentado e o método de Lanczos. O primeiro, não tão eficiente quanto o segundo, mas útil quando **A** tem poucos autovalores quase nulos, é criado circundando a matriz **A** com um número de linhas e colunas igual ao número desses autovalores, obtendo assim uma matriz $\tilde{\mathbf{A}}$ singular, com o papel de matriz dos coeficientes de um novo SELAS consistente, estreitamente relacionado com o SELAS original. A nova matriz $\tilde{\mathbf{A}}$ é mais bem condicionada que **A** e existe também uma relação importante entre os autovalores de ambas. Tais fatos encontra-se colecionados e provados no Teorema 3.7.

O método de Lanczos, ou melhor dito, o método de Lanczos para gerar vetores **A**-ortogonais propõe, como o próprio nome já diz, uma fórmula de recorrência que gera uma seqüência $(\mathbf{d}_j)$ de vetores **A**-ortogonais. O precondicionamento desse método, a versão **C**-ortogonal precondicionada, a resolução de SELAS com o uso da versão **A**-ortogonal precondicionada, e a tridiagonalização da matriz **A** através da base de Lanczos **A**-ortogonal e da **A**-ortonormal, são tópicos também incluídos no capítulo 3. Dentre esses, o que mereceu maior atenção, por estar diretamente na linha de nosso objetivo, que é determinar auto-soluções extremais de **A**, é o que transforma **A** numa matriz tridiagonal **H**, semelhante à **A**. O cálculo dos autovalores de **H**, que não é direto, pode ser mais simples que o cálculo dos de **A**, devido à esparsidade das matrizes tridiagonais. Também podemos estimar os autovalores extremos de **A** conhecendo os autovalores de uma submatriz $\mathbf{H}_k$ da matriz **H**. O Teorema 3.8.7 (Lanczos-Kaniel-Paige) é enunciado com esse propósito.

Embora o objetivo principal do desenvolvimento do método de Lanczos seja o cálculo do auto-sistema de **A**, não podemos deixar de citar que a resolução de um SELAS através da versão **A**-ortogonal precondicionada é muito interessante, tanto que motivou a construção de um algoritmo que foi implementado no MATLAB com o *m-file lanczos.m* e que foi utilizado para a resolução do Exemplo 3.8.4.1. O algoritmo *lanczos.m* é encontrado em apêndice.

Alguns resultados nesse trabalho, como deixamos entrever acima (Lema 1.2, Lema 2.3, Teorema 2.4.1, Teorema 2.4.2, Lema 3.8.3.1) e diversos outros sem título, foram criados por nós, embora, provavelmente, se encontrem implicitamente na literatura, outros foram adaptados ou modificados, e ainda outros foram apresentados com maior clareza e maiores detalhes que os correspondentes encontrados em periódicos.

Histórico

Na organização deste trabalho percebemos que não poderíamos deixar de colocar aqui um breve histórico do desenvolvimento dos MGC e dos métodos do tipo de Lanczos. Veremos que eles se situam dentro um certo período e são historicamente relacionados, de modo que não poderíamos falar de um sem citar outros. Quanto ao desenvolvimento do MGC e suas variações, o histórico avança até 1996.

Mediante levantamento bibliográfico redigimos, então, uma breve história dos métodos iterativos não-estacionários, isto é, métodos iterativos nos quais os cálculos em cada passo são feitos sobre informações que se atualizam em cada iteração.

Métodos baseados em ortogonalização, também chamados métodos de Krylov, começaram a ser desenvolvidos por volta de 1950. Os algoritmos dos métodos do gradiente conjugado e de Lanczos, para resolver problemas de autovalores e sistemas lineares, representaram importantes inovações computacionais naquela década, embora a aplicação dos mesmos só tenha sido difundida na década de 70, após passar por um refinamento dos algoritmos.

A primeira versão do algoritmo de um método de direções conjugadas foi dada por Fox, Huskey e Wilkinson, em 1948 [50]. Os autores desse artigo apresentaram um *método de vetores ortogonais*, como um método direto que requer a formação de uma base **A**-conjugada pela ortogonalização de Gram-Schmidt, com a representação do vetor solução nessa base. Lanczos [88], em 1950, propôs um algoritmo para um método iterativo, que resolve problemas de autovalores e envolve formas de recorrência de três termos. Apresentou ainda um algoritmo de biortogonalização de vetores para encontrar autovalores de matrizes não simétricas. Esse artigo motivou o desenvolvimento da forma de recorrência de três termos do método do gradiente conjugado para um sistema simétrico positivo definido, matéria também apresentada por Lanczos [88], [89], ainda no ano de 1950 e em 1952.

Na seqüência, em 1951, Arnoldi [3] publicou um novo algoritmo que deriva do algoritmo não simétrico de Lanczos, originando vetores biortogonais que reduzem a matriz do sistema para a forma de Hessenberg superior. Apresentado na sistemática de iterações minimizadas no método de Galerkin, este algoritmo foi conhecido como o *algoritmo de Arnoldi*. Nesse mesmo ano, vários artigos referentes a métodos iterativos e, em particular, ao método gradiente conjugado, foram publicados por Hestenes e outros pesquisadores [66], [49], [69], [70], [71], [115], entre os quais Lanczos. Entre esses trabalhos comparece o caso de sistemas singulares, para os quais a solução é obtida pelo processo dos mínimos quadrados, a discussão do caso do sistema ser semidefinido, a recomendação do uso das equações normais para problemas não simétricos e uma fórmula para o cálculo da inversa de uma matriz. O método do gradiente conjugado foi desenvolvido independentemente por Stiefel, do Instituto de Matemática Aplicada em Zurich e por Hestenes, com a cooperação de Rosser, Forsythe, e Paige, do Instituto de Análise Numérica. Relatos desse método, feitos por Stiefel e Rosser, aparecem na conferência da National Bureau Standards, que ocorreu de 23 a 25 de agosto de 1951, mas as primeiras publicações devem-se a Hestenes [66], em 1951, e a Stiefel [126], em 1952. Finalmente, esses dois matemáticos, juntos, ainda em 1952, publicaram a forma de recorrência de três termos para o resíduo e a de dois termos do método do gradiente conjugado [72], da qual a forma de três termos de Lanczos é derivada através da eliminação das direções de procura. Semelhantemente, Fletcher [46], em 1975 e 76, propôs uma implementação do método de Lanczos, com dois pares de recorrência de dois termos, que ele chamou de método do gradiente bi-conjugado (BiCG).

Até 1952, o método do gradiente conjugado era considerado um método direto. Usá-lo como um método indireto ficava a nível de sugestão, embora já fosse observado que, como tal, requeria menos que

n passos para problemas bem condicionados e mais que n passos para os mal condicionados, sendo n a ordem do sistema. Resultados computacionais do método foram apresentados por Hestenes e Stiefel [72] em 1952, Stiefel [128] em 1958, Engeli *et al.* [41] em 1959, entre outros. Artigos importantes, como [89], [125], [30], [67], [127], [90], [1], [41], publicados até o final da década de 50, fizeram diferentes abordagens aos métodos de ortogonalização.

Na década de 60 a reputação dos métodos historiados acima encontrava-se dividida. Em 1960, Frank [51] testou o algoritmo do método gradiente conjugado com uma matriz tridiag(-1,2,-1) de ordem 50×50, com o elemento da posição (1,1) mudado para 1, e cujos autovalores se relacionavam aos polinômios de Chebychev, um caso difícil para o método do gradiente conjugado, e constatou convergência lenta. Aplicações em análise estrutural [91] tentadas em 1960 fracassaram, mas Bothner-By e Naer-Collin [14] em 1962 mostraram satisfação com os resultados obtidos em problemas de análise de espectros químicos, e Pitha e Norman [108], já em 1967, eram usuários constantes do algoritmo. De 1962 a 1969, pesquisadores como Antosiewicz e Rheinboldt [02], Nashed [100], Daniel [31], Horwitz e Sarachik [74], e Kawamura e Volz [83], discutiam o algoritmo do método do gradiente conjugado em espaços de Hilbert, e Kratochvil [87] estudava o mesmo numa classe de operadores em espaços de Banach.

Um importante avanço se processou na direção de soluções de equações não lineares, e algoritmos para otimização foram desenvolvidos, de maneira que certos métodos puderam resolver muitos problemas sem o cálculo da matriz derivada. Os primeiros algoritmos dessa classe, que aplicam o processo do gradiente conjugado a funções quadráticas, foram apresentados por diversos pesquisadores em 1962, entre os quais citamos Powell [112]. Outros trabalharam nessa linha, a saber, Fletcher e Powell [47] em 1963, Fletcher e Reeves [48] em 1964, Shah, Buchler e Kempthome [120] em 1964, e Broyden [18] em 1965, Polak e Ribiere [110] em 1969, Polyak [111] em 1969, Zoutendijk [140] em 1960, Sinnott e Luenberger [122] em 1967, Pagurek e Woodside [101] em 1968, Luenberger [92] em 1969, e Miele, Huang e Heideman [97] em 1969 resolveram problemas condicionados com o uso do método do gradiente conjugado. Ainda nessa década, a publicação de Faddeev e Faddeeva [45], datada de 1963, apresenta discussões no que se refere ao uso de algoritmos para a **A**-biortogonalização, do método da descida mais íngreme, e de direções conjugadas, para resolver sistemas lineares.

A teoria de Kaniel [81], apresentada em 1966, contribuiu fortemente para a compreensão das propriedades de convergência do método do gradiente conjugado e de Lanczos. Ainda na década de 60 os algoritmos foram usados em problemas de aplicações, de espectros infravermelhos por Eu [42] em 1968, teoria da difusão por Garibotti e Vilani [57] em 1969, análise de redes por Marshall [93] em 1969, e no mesmo ano, numa análise de ogiva nuclear, por Sebe e Nachamkin [119].

Apesar de os algoritmos serem bastante usados na década de 60, a comunidade de análise numérica não estava satisfeita com eles, ou com a velocidade de convergência deles. Técnicas de precondicionamento não eram suficientemente conhecidas - embora muito progresso tenha surgido em técnicas de separação - e só no início dos anos 70 os desenvolvimentos chaves transformaram-se em algoritmos práticos de precondicionamento. A primeira aplicação do precondicionamento para melhorar a convergência de um método iterativo pode ser atribuída a Evans [43] em 1968 e [44] em 1973, e, no caso específico da aplicação ao método do gradiente conjugado, a Axelson [4] em 1972 que sugere um precondicionamento para o método por um operador SSOR. Nesse artigo apresenta também uma simulação numérica. Outros precondicionamentos foram discutidos por Evans [44] em 1973, Bartelo e Daniel [12] em 1974, Chandra, Eisenstat e Schultz [22] em 1975, Axelson [5] em 1976, Concus, Golub e O'Leary [26] em 1976, Douglas e Dupont [38] em 1976, e Meijerink e Van der Vorst [96] em 1977.

Reid [114], em 1971, chamou a atenção de muitos pesquisadores para o potencial do algoritmo do gradiente conjugado como um método iterativo para sistemas lineares esparsos, o que originou muitos trabalhos subseqüentes sobre o método. A tese de doutorado de Paige, datada do mesmo ano e intitulada *The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices*, defendida na Universidade de Londres, com publicações [102] em 1972 e [103] em 1976, serviu ao mesmo propósito em relação ao algoritmo de Lanczos, onde ocorreu o primeiro passo na compreensão da perda da ortogonalidade dos vetores de Lanczos, dando a chave para o desenvolvimento de algoritmos estáveis que não requerem completa reortogonalização. Isso tornou o algoritmo praticável em grandes problemas esparsos, reduzindo o tempo de armazenamento e computação. Mais desenvolvimentos em torno desse as-

sunto foram feitos por Takahasi e Natori [130] em 1971 e 1972, e por Kaham e Parlett [79] em 1976. A primeira versão estável do algoritmo do gradiente conjugado padrão para sistemas indefinidos é devida a Paige e Saunders [104], que a criaram em 1975, e Concus e Golub [25] propuseram um algoritmo para a classe de matrizes não simétricas em 1976. O algoritmo de Lanczos para matrizes de blocos foi desenvolvido por Cullum e Donath [27] e [28] em 1974, e Underwood [132] em 1975.

Inúmeras foram as aplicações dadas ao algoritmo do gradiente conjugado na década de 70, através dos trabalhos, por exemplo, de De e Davies [33] em 1970, de Kamoshida, Kani, Sato e Okada [80] em 1970, de Kobayashi [86] em 1970, de Powers [113] em 1973, de Wang e Treitel [136] em 1973, e de Dodson, Isaacs e Rollett [35] em 1976. Aplicações do algoritmo de Lanczos foram dadas por Chang e Wing [23] em 1970, Emilia e Bodvarsson [40] em 1970, Weaver e Yoshida [137] em 1971, Whitehead [138] em 1972, Harms [63] em 1974, Hausman, Bloan e Bender [64] em 1975, Ibarra, Vallieres e Feng [76] em 1975, Platzman [109] em 1975, Cline, Golub e Platzman [24] em 1976, e Kaplan e Gray [82] também em 1976. Mas a redescoberta do algoritmo de Lanczos é devida a Haydok, Heine e Kelley, cujos estudos realizados no período de 1972 a 1975, culminaram com a publicação [65] em 1975. Enfim, devemos à década de 70 a credibilidade e a aceitação dos métodos do gradiente conjugado e de Lanczos como métodos iterativos. Variações desses métodos desde então foram desenvolvidas, dentre as quais citamos as que seguem:

- O *método do resíduo mínimo generalizado* (GMRES), com importantes publicações: [118] em 1986, [36] em 1991, e [34] em 1993. Foi proposto com o objetivo de resolver sistemas lineares não-simétricos grandes e esparsos e baseou-se no método de Arnoldi clássico [3], que constrói bases ortonormais nos subespaços de Krylov;
- O *método do resíduo quase mínimo* (QMR), com publicações expressivas: [54] em 1991 e [55] em 1994. Este adquiriu popularidade nos últimos anos, quando o problema era resolver sistemas lineares esparsos e não-simétricos. Caracteriza-se pelo fato de reduzir o sistema original a um sistema tridiagonal, para, depois resolvê-lo no sentido dos mínimos quadrados. Em verdade é uma variante do método de Lanczos.
- Os *métodos do gradiente conjugado sobre as equações normais* (CGNE e CGNR), encontrados em [105] e [13], por exemplo. São baseados na aplicação do método gradiente conjugado em cada uma das duas formas das equações normais de $\mathbf{Ax} = \mathbf{b}$, quando $\mathbf{A}$ é não singular e não simétrica. O CGNE resolve o sistema $(\mathbf{AA}^t)\mathbf{y} = \mathbf{b}$ onde $\mathbf{x} = \mathbf{A}^t\mathbf{y}$, e o CGNR resolve $(\mathbf{A}^t\mathbf{A})\mathbf{x} = \mathbf{c}$ onde $\mathbf{c} = \mathbf{A}^t\mathbf{b}$.
- O *método do gradiente biconjugado* (BiCG), publicado por Lanczos [89] em 1952, seguido por Fletcher [46] em 1975 e 76, e com mais recentes publicações encontradas em [9] e [107]. Esse método gera duas seqüências de vetores, mutuamente ortogonais, de forma semelhante ao método gradiente conjugado, uma baseada na matriz $\mathbf{A}$ e outra em $\mathbf{A}^t$. É proposto para $\mathbf{A}$ não singular e não simétrica. A convergência pode ser irregular e existe a possibilidade de o método ser interrompido antes da aproximação esperada ser obtida.
- O *método conjugate gradient squared* (CGS), de Sonneveld [124]. Surgiu em 1989 como uma variação do método BiCG. Esse método aplica as operações de atualização da $\mathbf{A}$-seqüência e da $\mathbf{A}^t$-seqüência aos mesmos vetores. Na prática a convergência pode ser mais irregular do que em BiCG, mas tem a vantagem sobre esse de não necessitar cálculos de multiplicações com a matriz $\mathbf{A}^t$.
- O *método do gradiente biconjugado estabilizado* (Bi-CGSTAB), de Van der Vorst [133]. Foi proposto em 1992 como uma variação do método BiCG, de forma semelhante ao CGS, mas usando diferentes atualizações para a $\mathbf{A}^t$-seqüência a fim de obter convergência mais regular do que no método CGS.

A tese de doutorado de Elman [39] em 1982, artigos, por exemplo, de Axelsson e Vassilevski [7] em 1991, e de Vassilevski [135] em 1992, mostram que os métodos do gradiente conjugado podem, em qualquer caso, funcionar de forma satisfatória, se eles forem, adequadamente, precondicionados. Assim, a combinação de precondicionadores com a propriedade de otimização desses métodos é o que parece mostrar maior solidez.

No período de 1983 a 1996, técnicas de precondicionamento foram desenvolvidas e aprimoramentos aos métodos GMRES, BiCG, QMR, SYMMLQ, CGS, Bi-CGSTAB, e TFQMR foram propostos em [77], [117], [99], [116], [134], [95], [131], [59], [10], [60], [61], [62], [53], [56], [52], [123], [15], [16], [17], [19], [21], [37], [78], [106], [131], [85]. Combinações dos métodos clássicos foram originando novos métodos, como, por exemplo, a versão QMR do método Bi-CGSTAB deu origem ao método QMRCGSTAB, que comparece no artigo de Chan *et al.* [20] de 1994. Importante também é o artigo de Nachtigal, Reddy e Trefethen [98] que mostra que, para problemas diferentes, a classificação de um determinado grupo de métodos, quanto à eficiência, pode variar, ou seja, um método pode ser excelente em uma classe de problemas e não possuir o mesmo comportamento em outra classe de problemas. Isso atribui um certo grau de importância a cada um dos métodos existentes.

1. A Forma de Recorrência de Três Termos do Método do Gradiente Conjugado

1.1. Preliminares. Neste capítulo vamos expor, de maneira rigorosa e geral, como definir indutivamente uma seqüência $(\mathbf{x}_k)$ em $\mathbb{R}^n$, que integra o **M**étodo do **G**radiente **C**onjugado (MGC), que convirja para a solução de um Sistema de **E**quações **L**ineares **A**lgébrica**S** (SELAS)

$$\mathbf{Ax} = \mathbf{b}, \tag{1.1}$$

com n equações e n incógnitas, que satisfaz certas condições. Cada termo dessa seqüência será chamado uma *aproximação da solução*.

Basicamente distinguem-se duas formas de recorrência para o MGC: a de três termos e a de dois termos. Dedicamos um capítulo para cada uma das formas. No presente capítulo tratamos a de três termos.

Nesta exposição usaremos sistematicamente letras em negrito para simbolizar matrizes ou vetores e letras gregas para indicar escalares.

Consideremos um produto interno $\langle \bullet, \bullet \rangle$ em $\mathbb{R}^n$, definido por uma matriz simétrica positiva definida $\mathbf{W}$, isto é, para quaisquer $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$, $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t\mathbf{W}\mathbf{y}$. Supomos inicialmente que a matriz $\mathbf{A}$ dos coeficientes em (1.1) seja *autoadjunta* em relação ao produto interno $\langle \bullet, \bullet \rangle$, o que significa que

$$\langle \mathbf{x}, \mathbf{Ay} \rangle = \langle \mathbf{Ax}, \mathbf{y} \rangle, \text{ para quaisquer } \mathbf{x}, \mathbf{y} \in \mathbb{R}^n.$$

Além disso, admitimos que $\mathbf{A}$ seja *positiva definida* em relação a $\langle \bullet, \bullet \rangle$, isto é,

$$\langle \mathbf{x}, \mathbf{Ax} \rangle > \mathbf{0}, \text{ se } \mathbf{0} \neq \mathbf{x} \in \mathbb{R}^n.$$

As expressões *autoadjunta positiva definida* e *simétrica positiva definida* serão abreviadas, respectivamente, com a. p. d. e s. p. d.

Uma matriz $\mathbf{A}$ é a. p. d. em relação ao produto interno $\langle \bullet, \bullet \rangle$ representado por uma matriz s. p. d. $\mathbf{W}$, se e somente se $\mathbf{A}^t\mathbf{W} = \mathbf{WA}$ e $\mathbf{WA}$ é positiva definida [Vd. prova em Introdução].

Se $\mathbf{W} = c\mathbf{I}$ é uma matriz escalar com $c > 0$, então $\mathbf{A}$ é a. p. d. em relação ao produto interno representado por $\mathbf{W}$ se e somente se $\mathbf{A}$ é s. p. d. Nesse caso $c\mathbf{A}$ e $\mathbf{W}$ definem o mesmo produto interno. Em particular, ser $\mathbf{A}$ a. p. d. em relação ao produto interno usual é o mesmo que ser s. p. d.

É importante observar que o fato de uma matriz $\mathbf{A}$ ser a. p. d. em relação a um produto interno não implica que $\mathbf{A}$ seja s. p. d., e nem mesmo simétrica [Vd. exemplo em Introdução].

Concomitantemente com a seqüência $(\mathbf{x}_k)$ de aproximações da solução de (1.1) será gerada uma seqüência $(\mathbf{r}_k)$ de *resíduos*

$$\mathbf{r}_k := \mathbf{Ax}_k - \mathbf{b},$$

ortogonal em relação ao nosso produto interno. Os resultados dessa proposta serão enunciados e ampliados claramente no Teorema 1.3, que demonstraremos com cuidado, pois constitui a base de todo o desenvolvimento do capítulo. O que segue, então, é apresentado em preparação da compreensão do Teorema 1.3.

Seja $\mathbf{x}_o \in \mathbb{R}^n$ um vetor inicializador dado e definamos

$$\mathbf{x}_1 := \mathbf{x}_o - \beta_o \mathbf{r}_o, \tag{1.2}$$

onde

$$\beta_o := \frac{\langle \mathbf{r}_o, \mathbf{r}_o \rangle}{\langle \mathbf{r}_o, \mathbf{A}\mathbf{r}_o \rangle}. \tag{1.3}$$

Podemos supor $\mathbf{r}_o \neq \mathbf{0}$, pois, caso contrário, $\mathbf{x}_o$ é a solução de (1.1). Como $\mathbf{A}$ é positiva definida em relação ao produto interno em questão, para nosso objetivo essa escolha do escalar β_o é a única possível para que tenhamos $\mathbf{r}_1$ ortogonal a $\mathbf{r}_o$, uma vez que

$$\mathbf{r}_1 = \mathbf{A}\mathbf{x}_1 - \mathbf{b} = \mathbf{A}(\mathbf{x}_o - \beta_o \mathbf{r}_o) - \mathbf{b} = \mathbf{r}_o - \beta_o \mathbf{A}\mathbf{r}_o,$$

e, portanto,

$$\langle \mathbf{r}_1, \mathbf{r}_o \rangle = 0 \Leftrightarrow \beta_o = \frac{\langle \mathbf{r}_o, \mathbf{r}_o \rangle}{\langle \mathbf{r}_o, \mathbf{A}\mathbf{r}_o \rangle}.$$

Além disso, a positividade definida de $\mathbf{A}$ implica $\beta_o > 0$.

As definições (1.2) e (1.3) inicializam a recursão (seqüência em $\mathbb{R}^n$ definida por indução)

$$\mathbf{x}_{k+1} := \alpha_k \mathbf{x}_k + (1-\alpha_k)\mathbf{x}_{k-1} - \beta_k \mathbf{r}_k, \quad k = 0, 1, 2, \ldots, \tag{1.4}$$

onde

$$\alpha_k := \beta_k \frac{\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}, \quad k = 0, 1, 2, \ldots . \tag{1.5}$$

Claro, a *forma de recorrência de três termos* (1.4) só ficará definida após estabelecermos independentemente a seqüência de escalares (β_k). De (1.4) obtemos também recursivamente a seqüência dos resíduos:

$$\begin{aligned}
\mathbf{r}_{k+1} &:= \mathbf{A}\mathbf{x}_{k+1} - \mathbf{b} = \mathbf{A}(\alpha_k \mathbf{x}_k + (1-\alpha_k)\mathbf{x}_{k-1} - \beta_k \mathbf{r}_k) - \mathbf{b} \\
&= \alpha_k \mathbf{A}\mathbf{x}_k + (\mathbf{A}\mathbf{x}_{k-1} - \mathbf{b}) - \alpha_k \mathbf{A}\mathbf{x}_{k-1} - \beta_k \mathbf{A}\mathbf{r}_k \\
&= (\alpha_k \mathbf{A}\mathbf{x}_k - \alpha_k \mathbf{b}) + \mathbf{r}_{k-1} + (-\alpha_k \mathbf{A}\mathbf{x}_{k-1} + \alpha_k \mathbf{b}) - \beta_k \mathbf{A}\mathbf{r}_k \\
&= \alpha_k \mathbf{r}_k + (1-\alpha_k)\mathbf{r}_{k-1} - \beta_k \mathbf{A}\mathbf{r}_k .
\end{aligned}$$

Então

$$\mathbf{r}_{k+1} = \alpha_k \mathbf{r}_k + (1-\alpha_k)\mathbf{r}_{k-1} - \beta_k \mathbf{A}\mathbf{r}_k, \quad k = 0, 1, 2, \ldots . \tag{1.6}$$

Podemos admitir que $\mathbf{r}_k \neq \mathbf{0}$ para todo k, porque, se $\mathbf{r}_k = \mathbf{0}$ para algum k, então $\mathbf{x}_k$ é a solução de (1.1) e o processo recursivo termina.

Mostremos indutivamente que há uma única seqüência (β_n) de escalares, que torna a seqüência dos resíduos ortogonal. Conforme a inicialização, $\beta_o \neq 0$ e $\{\mathbf{r}_o, \mathbf{r}_1\}$ é ortogonal. Fixemos arbitrariamente k e suponhamos que seja $\beta_j \neq 0$ para $j = 1, 2, \ldots, k-1$, e que o conjunto $\{\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k\}$ seja ortogonal; esta última condição, em vista da indução, é equivalente a que o resíduo $\mathbf{r}_k$ seja ortogonal a todos os resíduos anteriores. O resíduo $\mathbf{r}_{k+1}$, dado como em (1.6), é ortogonal aos resíduos $\mathbf{r}_{k-1}$ e $\mathbf{r}_k$ se e somente se

$$\begin{cases} \alpha_k\langle \mathbf{r}_k, \mathbf{r}_{k-1}\rangle + (1-\alpha_k)\langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle - \beta_k\langle \mathbf{r}_{k-1}, \mathbf{A}\mathbf{r}_k\rangle = 0 \\ \alpha_k\langle \mathbf{r}_k, \mathbf{r}_k\rangle + (1-\alpha_k)\langle \mathbf{r}_k, \mathbf{r}_{k-1}\rangle - \beta_k\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\rangle = 0. \end{cases}$$

Usando a ortogonalidade de $\{\mathbf{r}_o, \mathbf{r}_1, \cdots, \mathbf{r}_k\}$, vemos que esse sistema linear em α_k e β_k, é equivalente ao

$$\begin{cases} \alpha_k\langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle + \beta_k\langle \mathbf{r}_{k-1}, \mathbf{A}\mathbf{r}_k\rangle = \langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle \\ \alpha_k\langle \mathbf{r}_k, \mathbf{r}_k\rangle - \beta_k\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\rangle = 0. \end{cases} \tag{1.7}$$

A segunda equação em (1.7) mostra que uma condição necessária e suficiente para que $\mathbf{r}_{k+1}$, dado como em (1.6), seja ortogonal a $\mathbf{r}_k$ é que α_k e β_k satisfaçam a igualdade em (1.5).

Como $\mathbf{A}$ é autoadjunta, mediante (1.5) e (1.6), com k-1 em lugar de k, e ainda usando a hipótese da indução, vem

$$\begin{aligned} \langle \mathbf{r}_{k-1}, \mathbf{A}\mathbf{r}_k\rangle &= \langle \mathbf{A}\mathbf{r}_{k-1}, \mathbf{r}_k\rangle \\ &= \frac{1}{\beta_{k-1}}\langle \alpha_{k-1}\mathbf{r}_{k-1} + (1-\alpha_{k-1})\mathbf{r}_{k-2} - \mathbf{r}_k,\ \mathbf{r}_k\rangle \\ &= -\frac{\langle \mathbf{r}_k, \mathbf{r}_k\rangle}{\beta_{k-1}}. \end{aligned}$$

Levando esse resultado na primeira equação de (1.7), o sistema (1.7) se escreve

$$\begin{cases} \alpha_k\langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle - \dfrac{\beta_k}{\beta_{k-1}}\langle \mathbf{r}_k, \mathbf{r}_k\rangle = \langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle \\ \alpha_k\langle \mathbf{r}_k, \mathbf{r}_k\rangle - \beta_k\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\rangle = 0. \end{cases}$$

Substituindo α_k na primeira equação, extraído da segunda, obtemos

$$\beta_k\frac{\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\rangle}{\langle \mathbf{r}_k, \mathbf{r}_k\rangle}\langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle - \frac{\beta_k}{\beta_{k-1}}\langle \mathbf{r}_k, \mathbf{r}_k\rangle = \langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle,$$

ou

$$\beta_k^{-1} = \frac{\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\rangle}{\langle \mathbf{r}_k, \mathbf{r}_k\rangle} - \frac{\langle \mathbf{r}_k, \mathbf{r}_k\rangle}{\langle \mathbf{r}_{k-1}, \mathbf{r}_{k-1}\rangle}\beta_{k-1}^{-1},\ k = 1, 2, \ldots. \tag{1.8}$$

Então, se pudermos provar que o segundo membro de (1.8) não se anula, a (1.8) define recursiva e univocamente a seqüência de escalares (β_n) de forma que seja ortogonal a seqüência dos resíduos, gerada pela seqüência $(\mathbf{x}_k)$ das aproximações da solução de (1.1), sendo essas aproximações produzidas

pela fórmula recursiva (1.4). De fato, mostraremos no Teorema 1.3 que $\beta_n > 0$ e $\alpha_k > 1$, além de vários outros resultados fundamentais, particularmente que, para cada k, o resíduo calculado por (1.6) e (1.8) tem norma (certa norma) mínima. Ainda em preparação a esse teorema, apresentamos o lema seguinte.

1.2. Lema. *O espaço vetorial* $R_k := \mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k)$, *gerado pelos vetores residuais* $\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k$, *definidos por* (1.5), (1.6) e (1.8), *é igual ao espaço de Krylov* $\mathcal{K}_k := \mathcal{K}_k(\mathbf{r}_o, \mathbf{A})$, *se esses vetores forem ortogonais e* $\beta_i \neq 0$ *para* $i = 0, 1, \ldots, k$. *Nesse caso* $\{\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k\}$ *é uma base ortogonal de* $\mathcal{K}_k(\mathbf{r}_o, \mathbf{A})$.

Demonstração. Claramente, $\dim(\mathcal{K}_k) \leq \dim(R_k) = k + 1$. Então basta mostrar que $R_k \subseteq \mathcal{K}_k$. É imediato que $R_0 = \mathcal{K}_0$ e $R_1 = \mathcal{K}_1$. Suponhamos que $R_i \subseteq \mathcal{K}_i$ para $i = 0, 1, \ldots, j\text{-}1$. Então

$$\mathbf{r}_i := a_{io}\mathbf{r}_o + a_{i1}\mathbf{A}\mathbf{r}_o + \ldots + a_{ii}\mathbf{A}^i\mathbf{r}_o, \text{ para } i = 0, 1, \ldots, j\text{-}1$$

e convenientes escalares a_{ij}. Para provar que $R_j \subseteq \mathcal{K}_j$ basta mostrar que $\mathbf{r}_j$ é uma combinação linear de $\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^j\mathbf{r}_o$. Mas, pela (1.6),

$$\mathbf{r}_j = \alpha_{j-1}\mathbf{r}_{j-1} + (1 - \alpha_{j-1})\mathbf{r}_{j-2} - \beta_{j-1}\mathbf{A}\mathbf{r}_{j-1}.$$

Então

$$\begin{aligned}
\mathbf{r}_j &= \alpha_{j-1}\left(a_{(j-1)o}\mathbf{r}_o + a_{(j-1)1}\mathbf{A}\mathbf{r}_o + \ldots + a_{(j-1)(j-1)}\mathbf{A}^{j-1}\mathbf{r}_o\right) \\
&\quad + (1 - \alpha_{j-1})\left(a_{(j-2)o}\mathbf{r}_o + a_{(j-2)1}\mathbf{A}\mathbf{r}_o + \ldots + a_{(j-2)(j-2)}\mathbf{A}^{j-2}\mathbf{r}_o\right) \\
&\quad - \beta_{j-1}\left(a_{(j-1)o}\mathbf{A}\mathbf{r}_o + a_{(j-1)1}\mathbf{A}^2\mathbf{r}_o + \ldots + a_{(j-1)(j-1)}\mathbf{A}^j\mathbf{r}_o\right) \\
&= a_o\mathbf{r}_o + a_1\mathbf{A}\mathbf{r}_o + \cdots + a_j\mathbf{A}^j\mathbf{r}_o,
\end{aligned}$$

onde os escalares a_i decorrem do agrupamento conveniente dos termos. ∎

Adotaremos as seguintes convenções notacionais:

$$\delta_k := \langle \mathbf{r}_k, \mathbf{r}_k \rangle, \quad \mu_k := \frac{\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle},$$
$$\Delta\mathbf{x}_k := \mathbf{x}_k - \mathbf{x}_{k-1}, \quad \Delta\mathbf{r}_k := \mathbf{r}_k - \mathbf{r}_{k-1}, \quad \widetilde{\alpha}_k := \alpha_k - 1.$$

1.3. Teorema. *Seja* $\mathbf{A}\mathbf{x} = \mathbf{b}$ *um SELAS de ordem* $n \times n$, *cuja matriz* $\mathbf{A}$ *dos coeficientes é a. p. d. em relação a* $\langle \bullet, \bullet \rangle$, *e seja o método iterativo, definido por* (1.4), (1.6), *onde*

$$\alpha_k := \beta_k\mu_k.$$

Então, para todo $k \in \{0, 1, 2, \ldots\}$,

(*a*) $\Delta\mathbf{x}_{k+1} = \widetilde{\alpha}_k\Delta\mathbf{x}_k - \beta_k\mathbf{r}_k$,

e, se $k > 0$,

$$\Delta\mathbf{r}_{k+1} = \widetilde{\alpha}_k\Delta\mathbf{r}_k - \beta_k\mathbf{A}\mathbf{r}_k; \tag{1.9}$$

(*b*) $\langle \mathbf{r}_{k+1}, \mathbf{r}_{k-j} \rangle = 0$ *e* $\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\Delta\mathbf{r}_{k-j+1} \rangle = 0$, *para* $j = 0, 1, 2, \ldots, k$;

(*c*) *as seqüências* (α_k) e (β_k), *esta última definida por* (1.8) *ou seja* $\beta_k^{-1} = \mu_k - \dfrac{\delta_k}{\delta_{k-1}}\beta_{k-1}^{-1}$, *com* $k > 0$, *satisfazem*

$$\begin{bmatrix} \langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle & -\delta_k \\ -\delta_k & \langle \Delta\mathbf{r}_k, \mathbf{A}^{-1}\Delta\mathbf{r}_k \rangle \end{bmatrix} \begin{bmatrix} \beta_k \\ \tilde{\alpha}_k \end{bmatrix} = \begin{bmatrix} \delta_k \\ 0 \end{bmatrix}; \tag{1.10}$$

se $\mathbf{r}_k \neq \mathbf{0}$; *a* (1.10) *implica que* $\alpha_k > 1$ *e* $\beta_k > 0$, *o que mostra que a seqüência* (β_k) *está bem definida por* (1.9);

(d) consideremos o subespaço S_k *de* $\mathbb{R}^n$, *gerado pelo conjunto* $\{\mathbf{x}_o, \mathbf{x}_1, \ldots, \mathbf{x}_k, \mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k\}$, *e indiquemos com*

$$\tilde{\mathbf{x}}_{k+1} := \xi_o\mathbf{x}_o + \xi_1\mathbf{x}_1 + \cdots + \xi_k\mathbf{x}_k - \eta_o\mathbf{r}_o - \eta_1\mathbf{r}_1 - \cdots - \eta_k\mathbf{r}_k$$

um vetor arbitrário de S_k; *teremos que o mínimo sobre* S_k *de* $\langle \tilde{\mathbf{x}}_{k+1} - \bar{\mathbf{x}},\ \mathbf{A}(\tilde{\mathbf{x}}_{k+1} - \bar{\mathbf{x}}) \rangle$, *onde* $\bar{\mathbf{x}}$ *é a solução do SELAS* $\mathbf{Ax} = \mathbf{b}$, *ocorre quando* $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1}$.

Demonstração. (a) A primeira igualdade é produzida subtraindo $\mathbf{x}_k$ de ambos os membros de (1.4) e a segunda, subtraindo $\mathbf{r}_k$ de ambos os membros de (1.6).

(b) Usemos a indução. Pela escolha da inicialização da fórmula de recursão de três termos, $\mathbf{r}_o$ e $\mathbf{r}_1$ são ortogonais. Suponhamos que, para um k arbitrário, sejam ortogonais os resíduos $\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k$. Pela definição de α_k e β_k, temos também que $\mathbf{r}_{k+1}$ é ortogonal a $\mathbf{r}_{k-1}$ e $\mathbf{r}_k$; além disso, para $j = 2, 3, \ldots, k$,

$$\begin{aligned} \langle \mathbf{r}_{k+1}, \mathbf{r}_{k-j} \rangle &= \langle \alpha_k\mathbf{r}_k + (1-\alpha_k)\mathbf{r}_{k-1} - \beta_k\mathbf{A}\mathbf{r}_k, \mathbf{r}_{k-j} \rangle \\ &= \alpha_k\langle \mathbf{r}_k, \mathbf{r}_{k-j} \rangle + (1-\alpha_k)\langle \mathbf{r}_{k-1}, \mathbf{r}_{k-j} \rangle - \beta_k\langle \mathbf{A}\mathbf{r}_k, \mathbf{r}_{k-j} \rangle \\ &= -\beta_k\langle \mathbf{A}\mathbf{r}_k, \mathbf{r}_{k-j} \rangle \\ &= -\beta_k\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_{k-j} \rangle = 0. \end{aligned}$$

O último produto interno é nulo, pois, sendo $\mathbf{A}\mathbf{r}_{k-j} \in \mathcal{K}_{k-j+1} \subset \mathcal{K}_{k-1}$, pelo Lema 1.2, $\mathbf{r}_k$ é ortogonal a todo vetor de $\mathcal{K}_{k-1}$, e a primeira igualdade em (b) está provada.

Para provar a segunda igualdade em (b), escrevemos, para $j = 0, 1, 2, \ldots, k$, a sucessão das seguintes igualdades, válidas e facilmente entendidas diante da ortogonalidade dos resíduos já estabelecida e da (1.9),

$$\begin{aligned} \langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\Delta\mathbf{r}_{k-j+1} \rangle &= \langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}(\tilde{\alpha}_{k-j}\Delta\mathbf{r}_{k-j} - \beta_{k-j}\mathbf{A}\mathbf{r}_{k-j}) \rangle \\ &= \langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\tilde{\alpha}_{k-j}\Delta\mathbf{r}_{k-j} - \beta_{k-j}\mathbf{r}_{k-j} \rangle \\ &= \tilde{\alpha}_{k-j}\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\Delta\mathbf{r}_{k-j} \rangle - \beta_{k-j}\langle \mathbf{r}_{k+1}, \mathbf{r}_{k-j} \rangle \end{aligned}$$

$$
\begin{aligned}
&= \widetilde{\alpha}_{k-j}\left\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\Delta\mathbf{r}_{k-j}\right\rangle \\
&= \widetilde{\alpha}_{k-j-1}\widetilde{\alpha}_{k-j}\left\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\Delta\mathbf{r}_{k-j-1}\right\rangle \\
&\vdots \\
&= \widetilde{\alpha}_1\widetilde{\alpha}_2\cdots\widetilde{\alpha}_{k-j}\left\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\Delta\mathbf{r}_1\right\rangle \\
&= \widetilde{\alpha}_1\widetilde{\alpha}_2\cdots\widetilde{\alpha}_{k-j}\left\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}(\mathbf{r}_1 - \mathbf{r}_o)\right\rangle \\
&= -\beta_o\widetilde{\alpha}_1\widetilde{\alpha}_2\cdots\widetilde{\alpha}_{k-j}\left\langle \mathbf{r}_{k+1}, \mathbf{r}_o\right\rangle \\
&= 0.
\end{aligned}
$$

Para escrever a penúltima igualdade, usamos $\mathbf{r}_1 = \mathbf{r}_o - \beta_o\mathbf{A}\mathbf{r}_o$.

(c) Fazendo o produto interno de (1.9) por $\mathbf{r}_k$, obtemos

$$\left\langle \mathbf{r}_k, \Delta\mathbf{r}_{k+1}\right\rangle = \widetilde{\alpha}_k\left\langle \mathbf{r}_k, \Delta\mathbf{r}_k\right\rangle - \beta_k\left\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\right\rangle,$$

ou

$$\left\langle \mathbf{r}_k, \mathbf{r}_{k+1}\right\rangle - \left\langle \mathbf{r}_k, \mathbf{r}_k\right\rangle = \widetilde{\alpha}_k\left\langle \mathbf{r}_k, \mathbf{r}_k\right\rangle - \widetilde{\alpha}_k\left\langle \mathbf{r}_k, \mathbf{r}_{k-1}\right\rangle - \beta_k\left\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\right\rangle,$$

ou ainda

$$\left\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\right\rangle\beta_k - \delta_k\widetilde{\alpha}_k = \delta_k. \tag{1.11}$$

Por outro lado, fazendo o produto interno de (1.9) por $\mathbf{A}^{-1}\Delta\mathbf{r}_k$, vem

$$\left\langle \mathbf{A}^{-1}\Delta\mathbf{r}_k,\ \Delta\mathbf{r}_{k+1}\right\rangle = \widetilde{\alpha}_k\left\langle \mathbf{A}^{-1}\Delta\mathbf{r}_k, \Delta\mathbf{r}_k\right\rangle - \beta_k\left\langle \mathbf{A}^{-1}\Delta\mathbf{r}_k, \mathbf{A}\mathbf{r}_k\right\rangle. \tag{1.12}$$

Como $\left\langle \mathbf{A}^{-1}\Delta\mathbf{r}_k, \mathbf{A}\mathbf{r}_k\right\rangle = \delta_k$ e, por (b), $\left\langle \mathbf{A}^{-1}\Delta\mathbf{r}_k,\ \Delta\mathbf{r}_{k+1}\right\rangle = 0$, a (1.12) é equivalente à equação

$$-\delta_k\beta_k + \widetilde{\alpha}_k\left\langle \mathbf{A}^{-1}\Delta\mathbf{r}_k, \Delta\mathbf{r}_k\right\rangle = 0. \tag{1.13}$$

As equações (1.11) e (1.13) formam um SELAS em β_k e $\widetilde{\alpha}_k$, cuja representação matricial é (1.10).

Calculemos o determinante da matriz dos coeficientes do SELAS (1.10):

$$d := \left\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k\right\rangle\left\langle \Delta\mathbf{r}_k, \mathbf{A}^{-1}\Delta\mathbf{r}_k\right\rangle - \delta_k^2. \tag{1.14}$$

Mas

$$
\begin{aligned}
\delta_k &= \left\langle \mathbf{r}_k, \mathbf{r}_k\right\rangle - \left\langle \mathbf{r}_{k-1}, \mathbf{r}_k\right\rangle \\
&= \left\langle \Delta\mathbf{r}_k, \mathbf{r}_k\right\rangle \\
&= \left(\Delta\mathbf{r}_k\right)^t\mathbf{W}\mathbf{r}_k
\end{aligned}
$$

$$
\begin{aligned}
&= \left(\Delta \mathbf{r}_k\right)^t \sqrt{\mathbf{W}}\sqrt{\mathbf{W}}\mathbf{r}_k \\
&= \left(\Delta \mathbf{r}_k\right)^t \left(\sqrt{\mathbf{W}}\right)^t \mathbf{B}^{-1}\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k \\
&= \left(\mathbf{B}^{-1}\sqrt{\mathbf{W}}\Delta \mathbf{r}_k\right)^t \left(\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k\right),
\end{aligned}
$$

onde $\mathbf{B}$ é a matriz tal que $\mathbf{W}^{-1/2}\mathbf{W}\mathbf{A}\mathbf{W}^{-1/2} = \mathbf{B}^t\mathbf{B}$ é a fatoração de Cholesky da matriz s. p. d. $\mathbf{W}^{-1/2}\mathbf{W}\mathbf{A}\mathbf{W}^{-1/2}$. Esta matriz é s. p. d. porque $\mathbf{W}$ o é, e $\mathbf{A}$ é a. p. d. em relação ao produto interno definido por $\mathbf{W}$ (o produto interno em questão). Ainda

$$
\begin{aligned}
\left\langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \right\rangle &= \mathbf{r}_k^t \mathbf{W}\mathbf{A}\mathbf{r}_k \\
&= \mathbf{r}_k^t \sqrt{\mathbf{W}}\mathbf{B}^t\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k \\
&= \left(\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k\right)^t \left(\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k\right) \\
&= \left\|\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k\right\|_2^2.
\end{aligned}
$$

O índice inferior 2 refere a norma induzida pelo produto interno usual.

Semelhantemente,

$$
\left\langle \Delta \mathbf{r}_k, \mathbf{A}^{-1}\Delta \mathbf{r}_k \right\rangle = \left\|\mathbf{B}^{-1}\sqrt{\mathbf{W}}\Delta \mathbf{r}_k\right\|_2^2.
$$

Com esses últimos resultados, a (1.14) pode escrever-se

$$
d = \left\|\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k\right\|_2^2 \left\|\mathbf{B}^{-1}\sqrt{\mathbf{W}}\Delta \mathbf{r}_k\right\|_2^2 - \left[\left(\mathbf{B}^{-1}\sqrt{\mathbf{W}}\Delta \mathbf{r}_k\right)^t \left(\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k\right)\right]^2.
$$

Então, pela desigualdade de Cauchy-Schwarz relativa ao produto interno usual em $\mathbb{R}^n$, resulta que $d \geq 0$ e que $d = 0$ se e somente se $\mathbf{B}^{-1}\sqrt{\mathbf{W}}\Delta \mathbf{r}_k$ e $\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k$ são linearmente dependentes. Neste último caso, existe uma constante c tal que

$$
\mathbf{B}^{-1}\sqrt{\mathbf{W}}\Delta \mathbf{r}_k = c\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k.
$$

Multiplicando à esquerda ambos os membros dessa igualdade por $\sqrt{\mathbf{W}}\mathbf{B}^t$, vem

$$
\sqrt{\mathbf{W}}\mathbf{B}^t\mathbf{B}^{-1}\sqrt{\mathbf{W}}\Delta \mathbf{r}_k = c\sqrt{\mathbf{W}}\mathbf{B}^t\mathbf{B}\sqrt{\mathbf{W}}\mathbf{r}_k,
$$

que é o mesmo que

$$
\mathbf{W}\Delta \mathbf{r} = c\mathbf{W}\mathbf{A}\mathbf{r}_k.
$$

E multiplicando ambos os membros desta, à esquerda, por $\mathbf{A}^{-1}\mathbf{W}^{-1}$, obtemos

$$
\mathbf{A}^{-1}\Delta \mathbf{r}_k = c\mathbf{r}_k. \tag{1.15}
$$

Fazendo agora o produto interno por $\mathbf{r}_k$, resulta, usando a parte (b),

$$c\langle \mathbf{r}_k, \mathbf{r}_k \rangle = 0, \tag{1.16}$$

verdadeira se e somente se $c = 0$ ou $\langle \mathbf{r}_k, \mathbf{r}_k \rangle = 0$. O primeiro caso com (1.15) implica $\mathbf{A}^{-1}\Delta\mathbf{r}_k = \mathbf{0}$, e, portanto, $\Delta\mathbf{r}_k = \mathbf{0}$. Mas $\langle \mathbf{r}_k, \mathbf{r}_k \rangle = \langle \mathbf{r}_k, \Delta\mathbf{r}_k \rangle$. Logo, em qualquer caso, (1.16) significa que $\mathbf{r}_k = \mathbf{0}$, ou seja que já foi achada uma solução $\mathbf{x}_k$ do SELAS $\mathbf{Ax} = \mathbf{b}$.

Se $\mathbf{r}_k \neq \mathbf{0}$, então $d > 0$ (por isso, a matriz dos coeficientes do SELAS em (1.10) é não singular), e obtemos por inversão

$$\begin{bmatrix} \beta_k \\ \tilde{\alpha}_k \end{bmatrix} = \frac{1}{d}\begin{bmatrix} \langle \Delta\mathbf{r}_k, \mathbf{A}^{-1}\Delta\mathbf{r}_k \rangle & \delta_k \\ \delta_k & \langle \mathbf{r}_k, \mathbf{A}\mathbf{r}_k \rangle \end{bmatrix}\begin{bmatrix} \delta_k \\ 0 \end{bmatrix} = \frac{1}{d}\begin{bmatrix} \langle \Delta\mathbf{r}_k, \mathbf{A}^{-1}\Delta\mathbf{r}_k \rangle \delta_k \\ \delta_k^2 \end{bmatrix}.$$

As componentes do vetor no segundo membro são positivas, pois, como $\mathbf{r}_k \neq \mathbf{0}$, será $\delta_k > 0$, $\Delta\mathbf{r}_k \neq \mathbf{0}$ e $\langle \Delta\mathbf{r}_k, \mathbf{A}^{-1}\Delta\mathbf{r}_k \rangle > 0$. Lembremos que $\mathbf{A}^{-1}$ é p. d. em relação a $\langle \bullet, \bullet \rangle$. Logo $\beta_k > 0$ e $\tilde{\alpha}_k > 0$.

(d) Ponhamos $\tilde{\mathbf{r}}_{k+1} := \mathbf{A}\tilde{\mathbf{x}}_{k+1} - \mathbf{b}$ e definamos um funcional linear f: $\mathrm{R}^{2k+2} \to \mathrm{R}$ por

$$f(\xi_o, \cdots, \xi_k, \eta_o, \cdots, \eta_k) := \langle \tilde{\mathbf{r}}_{k+1}, \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \rangle.$$

Nossa tese, então, consiste em demonstrar que o mínimo do funcional f é $f(0, \cdots, 0, 1-\alpha_k, \alpha_k, 0, \cdots, 0, \beta_k) = \langle \mathbf{x}_{k+1} - \bar{\mathbf{x}}, \mathbf{A}(\mathbf{x}_{k+1} - \bar{\mathbf{x}}) \rangle$. Para ver isso, basta observar que

$$\begin{aligned} \langle \tilde{\mathbf{r}}_{k+1}, \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \rangle &= \langle \mathbf{A}\tilde{\mathbf{x}}_{k+1} - \mathbf{b}, \mathbf{A}^{-1}(\mathbf{A}\tilde{\mathbf{x}}_{k+1} - \mathbf{b}) \rangle \\ &= \langle \mathbf{A}\tilde{\mathbf{x}}_{k+1} - \mathbf{A}\bar{\mathbf{x}}, \tilde{\mathbf{x}}_{k+1} - \bar{\mathbf{x}} \rangle \\ &= \langle \tilde{\mathbf{x}}_{k+1} - \bar{\mathbf{x}}, \mathbf{A}(\tilde{\mathbf{x}}_{k+1} - \bar{\mathbf{x}}) \rangle. \end{aligned}$$

Pelas condições sobre $\mathbf{A}$, f assume um mínimo em $\mathbf{y} := (\xi_o, \cdots, \xi_k, \eta_o, \cdots, \eta_k)$ se e somente se

$$f'(\mathbf{y}) = \mathbf{0},$$

isto é, se e somente se,

$$\frac{\partial f(\mathbf{y})}{\partial \xi_j} = 0 \quad \text{e} \quad \frac{\partial f(\mathbf{y})}{\partial \eta_j} = 0, \quad \text{para } j = 0, 1, \ldots, k. \tag{1.17}$$

Mas

$$\frac{\partial f(\mathbf{y})}{\partial \xi_j} = \langle \mathbf{A}\mathbf{x}_j, \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \rangle + \langle \tilde{\mathbf{r}}_{k+1}, \mathbf{x}_j \rangle = \langle \mathbf{x}_j, \tilde{\mathbf{r}}_{k+1} \rangle + \langle \mathbf{x}_j, \tilde{\mathbf{r}}_{k+1} \rangle = 2\langle \mathbf{x}_j, \tilde{\mathbf{r}}_{k+1} \rangle,$$

e

$$\frac{\partial f(\mathbf{y})}{\partial \eta_j} = \langle \tilde{\mathbf{r}}_{k+1}, -\mathbf{r}_j \rangle + \langle -\mathbf{A}\mathbf{r}_j, \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \rangle = -2\langle \mathbf{r}_j, \tilde{\mathbf{r}}_{k+1} \rangle.$$

Então as equações à esquerda em (1.17) significam

$$\left\langle \mathbf{x}_j, \tilde{\mathbf{r}}_{k+1} \right\rangle = 0, \text{ para } j = 0, 1, \ldots, k.$$

Em particular,

$$\left\langle \mathbf{x}_j, \tilde{\mathbf{r}}_{k+1} \right\rangle = 0 \text{ e } \left\langle \mathbf{x}_{j-1}, \tilde{\mathbf{r}}_{k+1} \right\rangle = 0, \text{ para } j = 1, 2, \ldots, k.$$

Isso nos permite escrever a seguinte sucessão de igualdades equivalentes, para $j = 1, 2, \ldots, k$:

$$\left\langle \mathbf{x}_j - \mathbf{x}_{j-1}, \tilde{\mathbf{r}}_{k+1} \right\rangle = 0,$$

$$\left\langle \mathbf{x}_j - \mathbf{x}_{j-1}, \mathbf{A}\mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \right\rangle = 0,$$

$$\left\langle \mathbf{A}\mathbf{x}_j - \mathbf{A}\mathbf{x}_{j-1}, \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \right\rangle = 0,$$

$$\left\langle \left(\mathbf{A}\mathbf{x}_j - \mathbf{b}\right) - \left(\mathbf{A}\mathbf{x}_{j-1} - \mathbf{b}\right), \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \right\rangle = 0,$$

$$\left\langle \mathbf{r}_j - \mathbf{r}_{j-1}, \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \right\rangle = 0,$$

$$\left\langle \Delta\mathbf{r}_j, \mathbf{A}^{-1}\tilde{\mathbf{r}}_{k+1} \right\rangle = 0,$$

$$\left\langle \tilde{\mathbf{r}}_{k+1}, \mathbf{A}^{-1}\Delta\mathbf{r}_j \right\rangle = 0. \tag{1.18}$$

A equação (1.18) é equivalente à igualdade anterior porque $\mathbf{A}^{-1}$ é autoadjunta.

Por outro lado, as equações à direita em (1.17) traduzem-se com

$$\left\langle \tilde{\mathbf{r}}_{k+1}, \mathbf{r}_j \right\rangle = 0, \text{ para } j = 0, 1, \ldots, k. \tag{1.19}$$

Em particular

$$\left\langle \tilde{\mathbf{r}}_{k+1}, \mathbf{r}_k \right\rangle = 0 \text{ e } \left\langle \tilde{\mathbf{r}}_{k+1}, \mathbf{r}_{k-1} \right\rangle = 0. \tag{1.20}$$

Vemos que as condições (1.20) obrigam que seja

$$\xi_k = \alpha_k, \ \xi_{k-1} = 1 - \alpha_k \text{ e } \eta_k = \beta_k.$$

Mas, então, observando que as condições (1.18) e (1.19) são as mesmas que as condições (b) com $\tilde{\mathbf{r}}_{k+1}$ em lugar de $\mathbf{r}_{k+1}$, concluímos que $\tilde{\mathbf{r}}_{k+1} = \mathbf{r}_{k+1}$, isto é, $\mathbf{A}\tilde{\mathbf{x}}_{k+1} - \mathbf{b} = \mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}$, ou

$$\sum_{j=0}^{k} \xi_j \mathbf{A}\mathbf{x}_j - \sum_{j=0}^{k} \eta_j \mathbf{A}\mathbf{r}_j - \mathbf{b} = \left(1 - \alpha_k\right)\mathbf{A}\mathbf{x}_{k-1} + \alpha_k \mathbf{A}\mathbf{x}_k - \beta_k \mathbf{A}\mathbf{r}_k - \mathbf{b}.$$

Conseqüentemente $\xi_j = 0$ para $j \neq k\text{-}1, k$, e $\eta_j = 0$ para $j \neq k$. Então, de fato, f assume seu mínimo sobre $\mathbb{R}^{2k+2}$ no ponto $\tilde{\mathbf{x}}_{k+1} = \mathbf{x}_{k+1} = \alpha_k \mathbf{x}_k + (1 - \alpha_k)\mathbf{x}_{k-1} - \beta_k \mathbf{r}_k$. ∎

1.4. Complexidade Computacional. O Teorema 1.3 fornece um algoritmo exeqüível, que pode ser levemente melhorado conforme o esquema

$$
\begin{aligned}
&\text{Escolher } \mathbf{x}_0 \text{ e } \varepsilon > 0;\\
&\text{seja } \mathbf{r}_0 := \mathbf{A}\mathbf{x}_0 - \mathbf{b},\ \beta_0 := \frac{\langle \mathbf{r}_0,\ \mathbf{r}_0 \rangle}{\langle \mathbf{r}_0,\ \mathbf{A}\mathbf{r}_0 \rangle},\ \tilde{\alpha}_0 = 0;\\
&\text{testar a convergência; se } \langle \mathbf{r}_0, \mathbf{r}_0 \rangle \geq \varepsilon, \text{ continuar;}\\
&\\
&\text{para } k = 0,1,2, \ldots, \text{ fazer}\\
&\Delta\mathbf{x}_{k+1} := \tilde{\alpha}_k \Delta\mathbf{x}_k - \beta_k \mathbf{r}_k;\\
&\mathbf{x}_{k+1} := \mathbf{x}_k + \Delta\mathbf{x}_{k+1};\\
&\Delta\mathbf{r}_{k+1} := \tilde{\alpha}_k \Delta\mathbf{r}_k - \beta_k \mathbf{A}\mathbf{r}_k;\\
&\mathbf{r}_{k+1} := \mathbf{r}_k + \Delta\mathbf{r}_{k+1};\\
&\beta_{k+1}^{-1} := \frac{\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle} - \beta_k^{-1} \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle};\\
&\tilde{\alpha}_{k+1} := \beta_{k+1} \frac{\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle} - 1;\\
&\text{testar a convergência; se } \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle \geq \varepsilon, \text{ continuar.}
\end{aligned}
\tag{1.21}
$$

Sem contar as operações aritméticas, para cada k, a complexidade computacional desse algoritmo envolve

- uma multiplicação matriz-vetor, onde a matriz é **A**,
- dois produtos internos,
- oito operações vetoriais: multiplicações por escalar e adições vetoriais.

Além disso, em cada passo precisa armazenar cinco vetores: $\mathbf{x}_k$, $\Delta\mathbf{x}_k$, $\mathbf{r}_k$, $\Delta\mathbf{r}_k$, $\mathbf{A}\mathbf{r}_k$.

Em alguns casos pode ser mais eficiente calcular $\mathbf{r}_{k+1} := \mathbf{A}\mathbf{x}_{k+1} - \mathbf{b}$, ao invés de calcular $\mathbf{r}_{k+1}$ pela relação de recorrência (1.6). Isso economiza três das oito operações vetoriais, mas acrescenta uma multiplicação matriz-vetor.

1.5. Precondicionamento. A eficiência de um método iterativo, quando aplicado diretamente sobre um SELAS $\mathbf{A}\mathbf{x} = \mathbf{b}$, pode não ser a esperada, se a matriz **A** for mal condicionada. O mau condicionamento ocorre quando os autovalores de **A** se distribuem num intervalo muito amplo, ou alguns deles são muito próximos de zero. Em certos casos, multiplicando ambos os membros de $\mathbf{A}\mathbf{x} = \mathbf{b}$ por uma matriz escolhida adequadamente, a distribuição dos autovalores da nova matriz dos coeficientes é melhorada. Notemos que o SELAS resultante, dito o SELAS *precondicionado*, é equivalente ao original, no sentido de terem ambos a mesma solução. Por exemplo, se uma matriz **C** é tal que $\mathbf{C}^{-1}$ é próxima de $\mathbf{A}^{-1}$, a matriz $\mathbf{C}^{-1}\mathbf{A}$ terá seus autovalores aglomerados em torno de 1, o SELAS precondicionado

$$\mathbf{C}^{-1}\mathbf{A}\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}$$

tem a mesma solução que $\mathbf{A}\mathbf{x} = \mathbf{b}$, e o processo iterativo pode ser aplicado com vantagem sobre ele.

Não pretendemos levar adiante, nesta secção, a análise do precondicionamento, mas relacioná-lo com o produto interno. Se **C** é s. p. d. (para fins práticos, uma aproximação de **A**), podemos usar o

produto interno definido por **C**, para o processo iterativo, além de adotá-la como matriz precondicionadora. Para isso, definimos o *resíduo precondicionado*

$$\tilde{\mathbf{r}}_k := \mathbf{C}^{-1}(\mathbf{A}\mathbf{x}_k - \mathbf{b}) \tag{1.22}$$

e resolvemos o SELAS $\mathbf{B}\mathbf{x} = \mathbf{c}$, com $\mathbf{B} := \mathbf{C}^{-1}\mathbf{A}$ e $\mathbf{c} := \mathbf{C}^{-1}\mathbf{b}$. É preciso que **B** seja autoadjunta e positiva definida em relação ao produto interno $\langle \bullet, \bullet \rangle$ que **C** define. A primeira condição impõe uma restrição sobre **A**, porque

$$\begin{aligned} \mathbf{B} \text{ é autoadjunta em relação a } \langle \bullet, \bullet \rangle &\Leftrightarrow \mathbf{CB} = \mathbf{B}^t\mathbf{C} \\ &\Leftrightarrow \mathbf{C}(\mathbf{C}^{-1}\mathbf{A}) = (\mathbf{C}^{-1}\mathbf{A})^t\mathbf{C} \Leftrightarrow \mathbf{A} = \mathbf{A}^t, \end{aligned}$$

ou seja, **A** deve ser simétrica. Quanto à segunda condição ela certamente ocorre se $\mathbf{x}^t\mathbf{A}\mathbf{x} > O$. Assim, se **A** é s. p. d., **B** torna-se autoadjunta em relação a $\langle \bullet, \bullet \rangle$. Nessa situação o algoritmo (1.21), onde **A** é substituída por **B** e $\mathbf{r}_k$, pelo resíduo precondicionado (1.22), é otimizado para resolver $\mathbf{B}\mathbf{x} = \mathbf{c}$ sobre o espaço de Krylov $\mathscr{K}(\tilde{\mathbf{r}}_o, \mathbf{B})$. Observemos que, nesse caso,

$$\mu_k = \frac{\langle \tilde{\mathbf{r}}_k, \mathbf{C}^{-1}\mathbf{A}\tilde{\mathbf{r}}_k \rangle}{\delta_k} = \frac{\tilde{\mathbf{r}}_k^t \mathbf{A}\tilde{\mathbf{r}}_k}{\delta_k},$$

e

$$\delta_k = \langle \tilde{\mathbf{r}}_k, \tilde{\mathbf{r}}_k \rangle = \langle \tilde{\mathbf{r}}_k, \mathbf{C}^{-1}\mathbf{r}_k \rangle = \tilde{\mathbf{r}}_k^t \mathbf{r}_k.$$

Além disso, pelo Teorema 1.3, $\mathbf{r}_{k+1}^t \mathbf{C}^{-1}\mathbf{r}_{k-j} = \langle \tilde{\mathbf{r}}_{k+1}, \tilde{\mathbf{r}}_{k-j} \rangle = 0$, para $j = 0, 1, 2, \ldots, k$, e, sendo $\mathbf{r}_{k+1}^t \mathbf{A}^{-1}\mathbf{r}_{k+1} = \langle \tilde{\mathbf{r}}_{k+1}, \mathbf{B}^{-1}\tilde{\mathbf{r}}_{k+1} \rangle$ a parte (d) desse teorema se aplica sem alterações [4]. Segue a variante precondicionada do algoritmo (1.21).

$$\left[\begin{array}{l} \text{Encontrar uma matriz precondicionadora } \mathbf{C}; \\ \text{escolher } \mathbf{x}_o \text{ e } \varepsilon; \\ \text{seja } \mathbf{r}_o := \mathbf{A}\mathbf{x}_o - \mathbf{b},\ \tilde{\mathbf{r}}_o := \mathbf{C}^{-1}\mathbf{r}_o,\ \beta_o := \dfrac{\tilde{\mathbf{r}}_o^t \mathbf{r}_o}{\tilde{\mathbf{r}}_o^t \mathbf{A}\tilde{\mathbf{r}}_o},\ \alpha_o := 1; \\ \text{para } k = 0, 1, 2, \ldots, \text{ fazer} \\ \mathbf{x}_{k+1} := \alpha_k \mathbf{x}_k + (1-\alpha_k)\mathbf{x}_{k-1} - \beta_k \tilde{\mathbf{r}}_k; \\ \mathbf{r}_{k+1} := \alpha_k \mathbf{r}_k + (1-\alpha_k)\mathbf{r}_{k-1} - \beta_k \mathbf{A}\tilde{\mathbf{r}}_k; \\ \tilde{\mathbf{r}}_{k+1} := \mathbf{C}^{-1}\mathbf{r}_{k+1}; \\ \beta_{k+1}^{-1} := \dfrac{\tilde{\mathbf{r}}_{k+1}^t \mathbf{A}\tilde{\mathbf{r}}_{k+1}}{\tilde{\mathbf{r}}_{k+1}^t \mathbf{r}_{k+1}} - \beta_k^{-1} \dfrac{\tilde{\mathbf{r}}_{k+1}^t \mathbf{r}_{k+1}}{\tilde{\mathbf{r}}_k^t \mathbf{r}_k}; \\ \alpha_{k+1} := \dfrac{\tilde{\mathbf{r}}_{k+1}^t \mathbf{A}\tilde{\mathbf{r}}_{k+1}}{\tilde{\mathbf{r}}_{k+1}^t \mathbf{r}_{k+1}}; \\ \text{testar a convergência; se } \tilde{\mathbf{r}}_{i+1}^t \mathbf{r}_{i+1} \geq \varepsilon, \text{ continuar.} \end{array}\right. \tag{1.23}$$

A declaração " $\tilde{\mathbf{r}} := \mathbf{C}^{-1}\mathbf{r}$ " deve ser interpretada como "resolver o SELAS $\mathbf{C}\tilde{\mathbf{r}} := \mathbf{r}$ ".

2. A Forma de Recorrência de Dois Termos do Método do Gradiente Conjugado

2.1. Preliminares. A forma mais simples do MGC ocorre quando a matriz **A** do SELAS $\mathbf{Ax} = \mathbf{b}$ é s. p. d. e a recorrência do método é reduzida à forma de dois termos. A idéia de partida é minimizar o funcional $f := \mathbb{R}^n \to \mathbb{R}$, dado por

$$f(\mathbf{x}) := \frac{1}{2}\langle \mathbf{r}, \mathbf{A}^{-1}\mathbf{r}\rangle, \tag{2.1}$$

onde $\mathbf{r} := \mathbf{Ax} - \mathbf{b}$ e o produto interno é definido por uma matriz s. p. d. **W**.

É útil tratar o assunto de forma mais geral. Para isso, como no capítulo 1, supomos aqui também que **A** seja a. p. d. em relação ao produto interno em (2.1), de modo que vale

$$f(\mathbf{x}) = \frac{1}{2}\langle \mathbf{x}, \mathbf{Ax}\rangle - \langle \mathbf{b}, \mathbf{x}\rangle + \frac{1}{2}\langle \mathbf{b}, \mathbf{A}^{-1}\mathbf{b}\rangle.$$

O mínimo estrito e global de f ocorre na solução $\bar{\mathbf{x}} = \mathbf{A}^{-1}\mathbf{b}$ do SELAS $\mathbf{Ax} = \mathbf{b}$, pois, para $\mathbf{d} \in \mathbb{R}^n$, indicando com $\nabla f(\mathbf{x})$ o gradiente de f,

$$\left.\begin{aligned} \langle \nabla f(\mathbf{x}), \mathbf{d}\rangle &= \frac{1}{2}\big(\langle \mathbf{x}, \mathbf{Ad}\rangle + \langle \mathbf{d}, \mathbf{Ax}\rangle\big) - \langle \mathbf{b}, \mathbf{d}\rangle \\ &= \langle \mathbf{Ax}, \mathbf{d}\rangle + \langle -\mathbf{b}, \mathbf{d}\rangle \\ &= \langle \mathbf{Ax} - \mathbf{b}, \mathbf{d}\rangle. \end{aligned}\right\} \tag{2.2}$$

Por aí vemos que: $\langle \nabla f(\mathbf{x}), \mathbf{d}\rangle = 0$ para todo $\mathbf{d} \in \mathbb{R}^n \Leftrightarrow \mathbf{Ax} = \mathbf{b}$. O fato de o mínimo de f ser estrito e global decorre de **A** ser a. p. d. em relação ao produto interno em questão.

Outro fato que é revelado por (2.2), uma vez que vale para todo **d**, é que $\nabla f(\mathbf{x}) = \mathbf{Ax} - \mathbf{b} =: \mathbf{r}$.

2.2. Minimizações Sucessivas. O que pretendemos nesse capítulo é fazer uma conveniente escolha de n direções linearmente independentes $\mathbf{d}_0, \mathbf{d}_1, \cdots, \mathbf{d}_{n-1}$, e, por minimizações sucessivas de f, ao longo de cada uma dessas direções, construir uma seqüência $(\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1})$ tal que $f(\mathbf{x}_{n-1})$ seja o mínimo global de f (a menos de erros de arredondamento atinentes ao processo). Chamaremos a cada uma das direções $\mathbf{d}_k$ *de direção de procura*, ou *vetor de procura*, pois, para chegar ao mínimo global de f, repetindo, procuramos sucessivamente o mínimo condicionado de f sobre cada reta $\tau \mapsto \mathbf{x}_k + \tau\mathbf{d}_k$, onde $\mathbf{x}_k$ é a

aproximação da solução de $\mathbf{Ax} = \mathbf{b}$ no estágio anterior. Cada nova direção de procura $\mathbf{d}_{k+1}$ é gerada após a minimização de f na direção do vetor $\mathbf{d}_k$. Explicando melhor, sendo $\mathbf{x}_k$ o ponto de mínimo de f ao longo da direção $\mathbf{d}_{k-1}$, calculamos o ponto de mínimo, digamos τ_k, da função $\tau \mapsto f(\mathbf{x}_k + \tau \mathbf{d}_k)$, de $\mathbb{R}$ em $\mathbb{R}$, e definimos

$$\mathbf{x}_{k+1} := \mathbf{x}_k + \tau_k \mathbf{d}_k, \tag{2.3}$$

como uma nova aproximação do mínimo global de f, ou seja da solução de $\mathbf{Ax} = \mathbf{b}$.

Seja $\mathbf{r}_k := \mathbf{Ax}_k - \mathbf{b}$ e definamos h: $\mathbb{R} \to \mathbb{R}$ por

$$h(\tau) := f(\mathbf{x}_k + \tau \mathbf{d}_k) - f(\mathbf{x}_k).$$

De (2.1) obtemos

$$h(\tau) = \tau \langle \mathbf{r}_k, \mathbf{d}_k \rangle + \frac{1}{2} \tau^2 \langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \rangle.$$

Vemos que h é um funcional quadrático, que assume seu mínimo (lembremos que $\mathbf{A}$ é positiva definida em relação ao produto interno em questão) no ponto

$$\tau_k = -\frac{\langle \mathbf{r}_k, \mathbf{d}_k \rangle}{\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \rangle}. \tag{2.4}$$

Novamente, devido à positividade definida de $\mathbf{A}$, a seqüência (τ_k) em $\mathbb{R}$ está bem definida, assim como a *forma de recorrência de dois termos* (2.3).

Pela definição de $\mathbf{r}_k$ e (2.3), obtemos a *forma de recorrência de dois termos* para os resíduos,

$$\mathbf{r}_{k+1} = \mathbf{r}_k + \tau_k \mathbf{A}\mathbf{d}_k. \tag{2.5}$$

Fazendo o produto interno de ambos os membros de (2.5) por $\mathbf{d}_k$, vem

$$\langle \mathbf{r}_{k+1}, \mathbf{d}_k \rangle = \langle \mathbf{r}_k + \tau_k \mathbf{A}\mathbf{d}_k, \mathbf{d}_k \rangle = \langle \mathbf{r}_k, \mathbf{d}_k \rangle + \tau_k \langle \mathbf{A}\mathbf{d}_k, \mathbf{d}_k \rangle = 0, \tag{2.6}$$

tendo em vista o valor de τ_k, dado por (2.4), isto é, o resíduo (ou gradiente) se torna ortogonal à direção de procura, na iteração k.

Para a iteração seguinte, precisamos de nova direção de procura $\mathbf{d}_{k+1}$. Escolhemos determinar esta de maneira que

$$\langle \mathbf{d}_{k+1}, \mathbf{A}\mathbf{d}_j \rangle = 0, \; j = 0, \; 1, \; \ldots, \; k. \tag{2.7}$$

Expressamos isso, dizendo que as direções de procura ficam **A**-*ortogonais* em relação ao produto interno $\langle \bullet, \bullet \rangle$ ou *conjugadamente ortogonais*. Veremos que a condição (2.7) é equivalente a $\langle \mathbf{r}_{k+1}, \mathbf{r}_j \rangle = 0$, para $j = 0, 1, \ldots, k$.

O desenvolvimento acima comporta que a escolha das direções de procura $\mathbf{d}_k$ possa ser feita de várias maneiras. Mas há uma opção importante, de que vamos tratar a seguir.

Chamaremos ao subespaço de $\mathbb{R}^n$, gerado pelas direções de procura, $\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k$, denotado por $\mathscr{D}_k := \mathscr{G}(\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k)$, de *espaço de ordem k das direções de procura*.

2.3. Lema. *Se as direções de procura são conjugadamente ortogonais, o resíduo na iteração k+1 é ortogonal a* $\mathscr{D}_k$.

Demonstração. Basta provar que $\mathbf{r}_{k+1}$ é ortogonal às direções de procura $\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k$. A prova será por indução. Escolhidos $\mathbf{x}_o$ e $\mathbf{d}_o$, temos $\mathbf{r}_1 = \mathbf{r}_o + \tau_o \mathbf{A}\mathbf{d}_o$, conforme (2.5). Então, usando (2.4),

$$\langle \mathbf{r}_1, \mathbf{d}_o \rangle = \langle \mathbf{r}_o, \mathbf{d}_o \rangle + \tau_o \langle \mathbf{A}\mathbf{d}_o, \mathbf{d}_o \rangle = 0.$$

Suponhamos que

$$\langle \mathbf{r}_k, \mathbf{d}_j \rangle = 0, \text{ para } j = 0, 1, \ldots, k\text{-}1 \text{ e } k \geq 1. \tag{2.8}$$

Então, por (2.5)

$$\langle \mathbf{r}_{k+1}, \mathbf{d}_j \rangle = \langle \mathbf{r}_k, \mathbf{d}_j \rangle + \tau_k \langle \mathbf{A}\mathbf{d}_k, \mathbf{d}_j \rangle.$$

Aplicando agora (2.7) e (2.8), resulta $\langle \mathbf{r}_{k+1}, \mathbf{d}_j \rangle = 0$, para $j = 0, 1, \ldots, k\text{-}1$; também $\langle \mathbf{r}_{k+1}, \mathbf{d}_k \rangle = 0$ por (2.6), o que completa a prova. ∎

Na secção 2.7 veremos que a propriedade demonstrada no Lema 2.3 implica que o método calcula o ponto de mínimo $\mathbf{x}_{k+1}$ da função f, definida em (2.1), restrita a $\mathbf{r}_o + \mathscr{G}(\mathbf{A}\mathbf{r}_o, \mathbf{A}^2\mathbf{r}_o, \ldots, \mathbf{A}^{k+1}\mathbf{r}_o)$ ou, o que é o mesmo, calcula a melhor aproximação $\mathbf{x}_{k+1}$ da solução do SELAS $\mathbf{A}\mathbf{x} = \mathbf{b}$ sobre esse espaço afim.

2.4. Escolha das Direções de Procura. Para completar a exposição em curso, precisamos achar uma maneira eficiente de gerar a seqüência das direções de procura de forma que seja **A**-ortogonal em relação a esse produto interno. Com esse objetivo em mente, seja

$$\mathbf{d}_{k+1} := -\mathbf{r}_{k+1} + \beta_k \mathbf{d}_k, \quad k = 0, 1, 2, \ldots, \tag{2.9}$$

onde iniciamos a recursão pondo $\mathbf{d}_o := -\mathbf{r}_o$, o oposto do gradiente do funcional f definido em (2.1) (poderia ser outro $\mathbf{d}_o$). Para obter a **A**-ortogonalidade desejada, precisamos definir a propósito a seqüência (β_k) em (2.9).

Mas, diante da (2.9), exigir

$$\langle \mathbf{d}_{k+1}, \mathbf{A}\mathbf{d}_k \rangle = 0, \text{ para todo } k$$

é equivalente a exigir que a seqüência (β_k) seja dada por

$$\beta_k = \frac{\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{d}_k \rangle}{\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \rangle}. \tag{2.10}$$

Aqui, para que haja clareza sobre certos fatos, apresentamos o

2.4.1. Teorema. *Se* $\mathbf{d}_o := -\mathbf{r}_o$, *então,*

(a) para todo k, $\mathscr{G}(\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k) = \mathscr{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k) = \mathscr{G}(\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^k\mathbf{r}_o) = \mathscr{G}(\mathbf{d}_o, \mathbf{A}\mathbf{d}_o,$

$\ldots, \mathbf{A}^k\mathbf{d}_o$), *isto é, o* $k^{\text{ésimo}}$ *espaço de Krylov é gerado tanto pelas* $k+1$ *primeiras direções de procura quanto pelos* $k+1$ *primeiros resíduos.*

(b) para todo k, *se* $\{\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k\}$ *é conjugadamente ortogonal, o conjunto* $\{\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k\}$ *dos correspondentes resíduos é ortogonal em relação ao produto interno* $\langle \bullet, \bullet \rangle$.

Demonstração. (a) Demonstremos inicialmente a primeira igualdade, por indução. Obviamente $\mathcal{G}(\mathbf{d}_o) = \mathcal{G}(\mathbf{r}_o)$.

Suponhamos $\mathcal{G}(\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_j) = \mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_j)$. Então $\mathbf{d}_j$ é uma combinação linear de $\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_j$. Logo, qualquer que seja a seqüência (β_k), com $\beta_k \neq 0$ para todo k, por (2.9), $\mathbf{d}_{j+1}$ é uma combinação linear de $\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_j, \mathbf{r}_{j+1}$. Isso mostra que $\mathcal{G}(\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k) \subset \mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k)$. A inclusão oposta segue imediatamente de (2.9), *independentemente da escolha da seqüência* (β_k), *desde que* $\beta_k \neq 0$ *para todo* k. A primeira igualdade subsiste, pois.

Como a igualdade $\mathcal{G}(\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^k\mathbf{r}_o) = \mathcal{G}(\mathbf{d}_o, \mathbf{A}\mathbf{d}_o, \ldots, \mathbf{A}^k\mathbf{d}_o)$ é óbvia, para completar a demonstração de (a) basta mostrar, por exemplo, que $\mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k) = \mathcal{G}(\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^k\mathbf{r}_o)$, e isso será feito também por indução. Por (2.5), $\mathbf{r}_1 = \mathbf{r}_o + \tau_o\mathbf{A}\mathbf{d}_o$, donde segue $\mathcal{G}(\mathbf{r}_o, \mathbf{r}_1) = \mathcal{G}(\mathbf{r}_o, \mathbf{A}\mathbf{r}_o)$.

Suponhamos que $\mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_j) = \mathcal{G}(\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^j\mathbf{r}_o)$. Novamente por (2.5), $\mathbf{r}_{j+1} = \mathbf{r}_j + \tau_j\mathbf{A}\mathbf{d}_j$. Pela hipótese da indução, $\mathbf{r}_j$ é uma combinação linear de $\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^j\mathbf{r}_o$. Pela primeira igualdade já demonstrada, $\mathbf{d}_j \in \mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_j)$; portanto, temos $\mathbf{A}\mathbf{d}_j \in \mathcal{G}(\mathbf{A}\mathbf{r}_o, \mathbf{A}\mathbf{r}_1, \ldots, \mathbf{A}\mathbf{r}_j) = \mathcal{G}(\mathbf{A}\mathbf{r}_o, \mathbf{A}^2\mathbf{r}_o, \ldots, \mathbf{A}^{j+1}\mathbf{r}_o)$; daí decorre que $\mathbf{A}\mathbf{d}_j$ é combinação linear de $\mathbf{A}\mathbf{r}_o, \mathbf{A}^2\mathbf{r}_o, \ldots, \mathbf{A}^{j+1}\mathbf{r}_o$. Em suma, $\mathbf{r}_{j+1}$ é combinação linear de $\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^j\mathbf{r}_o, \mathbf{A}^{j+1}\mathbf{r}_o$. Com isso mostramos que $\mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_k) \subset \mathcal{G}(\mathbf{r}_o, \mathbf{A}\mathbf{r}_o, \ldots, \mathbf{A}^k\mathbf{r}_o)$.

Para mostrar a inclusão oposta, basta provar que, com base na hipótese de indução feita, $\mathbf{A}^{j+1}\mathbf{r}_o \in \mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_{j+1})$. Por essa hipótese, $\mathbf{A}^j\mathbf{r}_o \in \mathcal{G}(\mathbf{r}_o, \mathbf{r}_1, \ldots, \mathbf{r}_j)$; logo $\mathbf{A}^{j+1}\mathbf{r}_o \in \mathcal{G}(\mathbf{A}\mathbf{r}_o, \mathbf{A}\mathbf{r}_1, \ldots, \mathbf{A}\mathbf{r}_j) = \mathcal{G}(\mathbf{A}\mathbf{d}_o, \mathbf{A}\mathbf{d}_1, \ldots, \mathbf{A}\mathbf{d}_j)$. Mas, por (2.5), $\mathbf{A}\mathbf{d}_i$ é combinação linear de $\mathbf{r}_i$ e $\mathbf{r}_{i+1}$ para $i = 0, 1, 2, \ldots, j$, e a demonstração de (a) está completa.

(b) Pelo Lema 2.3, para todo $k \geq 0$, $\mathbf{r}_{k+1}$ é ortogonal a $\mathcal{G}(\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k)$; então, pela primeira igualdade em (a), decorre a ortogonalidade a provar. ▌

Agora estamos em condição de apresentar o resultado central desta secção, que consiste no

2.4.2.Teorema. *A seqüência* $(\mathbf{d}_k)$ *das direções de procura definida em* (2.9) *é* **A***-ortogonal, se definimos a seqüência* (β_k) *como em* (2.10).

Demonstração. A (2.9) leva a

$$\langle \mathbf{d}_{k+1}, \mathbf{A}\mathbf{d}_j \rangle = -\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{d}_j \rangle + \langle \beta_k\mathbf{d}_k, \mathbf{A}\mathbf{d}_j \rangle. \tag{2.11}$$

Por (2.9) e (2.10) $\mathbf{d}_o$ e $\mathbf{d}_1$ são ortogonais. Admitamos que $\{\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_k\}$ seja **A**-ortogonal. Então o último termo em (2.11) é nulo para $j = 0, 1, \ldots, k\text{-}1$. A escolha $\mathbf{d}_o := -\mathbf{r}_o$ implica, através de (2.5) e da parte (a) do Teorema 2.4.1, que

$$\mathbf{A}\mathbf{d}_j = \frac{1}{\tau_j}(\mathbf{r}_{j+1} - \mathbf{r}_j) \in \mathcal{D}_{j+1}.$$

Portanto, pelo Lema 2.3,

$$\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{d}_j \rangle = 0, \text{ para } j = 0, 1, \ldots, k\text{-}1. \tag{2.12}$$

Então, olhando para (2.11),

$$\langle \mathbf{d}_{k+1}, \mathbf{A}\mathbf{d}_j \rangle = 0, \text{ para } j = 0, 1, \ldots, k\text{-}1. \tag{2.13}$$

Conseqüentemente, com base em (2.13) e na definição (2.10) da seqüência (β_k) que figura em (2.9), a seqüência $(\mathbf{d}_k)$ das direções de procura é **A**-ortogonal. ▌

2.5. Comentários. O método iterativo não estacionário para resolver um SELAS, em que as direções de procura são determinadas por (2.9) e (2.10) é dito *método do gradiente conjugado padrão*, ou simplesmente *método do gradiente conjugado*. Há outro método para calcular o ponto de mínimo global de f definida por (2.1), sob alguns aspetos mais simples, mas menos eficiente, chamado de *método da descida mais íngreme*, para o qual $\beta_k = 0$ em (2.9). Esta última condição significa que, em cada iteração, o ponto de mínimo condicionado é buscado na direção e sentido oposto do gradiente de f. A menor eficiência vem do seguinte: no MGC, como explicaremos abaixo, o processo teoricamente termina em no máximo n iterações, ao passo que, no método da descida mais íngreme, isso não ocorre, porque algumas direções podem repetir-se. Para detalhes Vd. [6], [121].

Como já mencionamos, podemos optar por outra seqüência de direções de procura. Um possibilidade óbvia é $\mathbf{d}_k := [0 \ldots 1\ 0 \ldots 0]^t$, onde a unidade ocupa a $(k+1)^{\underline{ésima}}$ posição. Facilmente se mostra que esta opção conduz ao método iterativo de Gauss-Seidel, este sabidamente com convergência lenta. (Vd. esse método em [32] e [84]). Hestenes [68] mostrou em 1980 que essa mesma escolha pode descrever o processo da eliminação gaussiana.

2.6. Terminação do Processo. Se, para algum k, $\mathbf{d}_k = 0$, a iteração sucessora não pode ser executada, porque τ_k em (2.4) não fica definido. Vamos investigar a situação que surge, quando (2.9) produz uma direção de procura nula.

Em (2.9), substituímos $k + 1$ por k e depois fazemos o produto interno dos dois membros por $\mathbf{r}_k$, resultando

$$\langle \mathbf{d}_k, \mathbf{r}_k \rangle = -\langle \mathbf{r}_k, \mathbf{r}_k \rangle + \beta_{k-1} \langle \mathbf{d}_{k-1}, \mathbf{r}_k \rangle.$$

Mas, pelo Lema 2.3, $\langle \mathbf{d}_{k-1}, \mathbf{r}_k \rangle = 0$ e, portanto,

$$\langle \mathbf{d}_k, \mathbf{r}_k \rangle = -\langle \mathbf{r}_k, \mathbf{r}_k \rangle. \tag{2.14}$$

Se $\mathbf{d}_k = 0$, será $\langle \mathbf{r}_k, \mathbf{r}_k \rangle = 0$, o que implica $\mathbf{r}_k = \mathbf{A}\mathbf{x}_k - \mathbf{b} = \mathbf{0}$, ou seja, que $\mathbf{x}_k$ já é solução de $\mathbf{A}\mathbf{x} = \mathbf{b}$. A conclusão é que, encontrar uma direção de procura nula significa encontrar o ponto de mínimo global de f, em outras palavras, a solução do SELAS $\mathbf{A}\mathbf{x} = \mathbf{b}$, que é tudo o que procuramos, e o processo iterativo tem mesmo que ser suspenso.

Por outro lado, se $\mathbf{x}_k$ não é a solução de $\mathbf{A}\mathbf{x} = \mathbf{b}$, então $\mathbf{r}_k \neq \mathbf{0}$, pois isso acarreta $\langle \mathbf{r}_k, \mathbf{r}_k \rangle \neq 0$ e vemos, por (2.14), que $\langle \mathbf{d}_k, \mathbf{r}_k \rangle \neq 0$ e daí, por (2.4), que $\tau_k \neq 0$ (recordemos que aqui **A** é positiva definida). Concluímos que, enquanto $\mathbf{x}_k$ não é solução exata de $\mathbf{A}\mathbf{x} = \mathbf{b}$, o processo iterativo definido por (2.3), (2.4) e (2.9) continua, independentemente da escolha da seqüência (β_k).

Juntando as conclusões, temos que ou o processo será infinito, com $\mathbf{x}_k$ não coincidindo com a solução de $\mathbf{A}\mathbf{x} = \mathbf{b}$, para todo k, ou existirá um inteiro l tal que $\mathbf{x}_l$ é a solução exata de $\mathbf{A}\mathbf{x} = \mathbf{b}$, e o processo terminará na iteração $l^{\underline{ésima}}$, sendo que nenhum $\mathbf{x}_k$ é solução exata desse SELAS para $k < l$. Mas, se o cálculo da seqüência (β_k) for feito de maneira que a seqüência das direções de procura seja **A**-ortogonal,

como descrevemos na secção 2.4, o processo deverá teoricamente terminar numa iteração $l \leq n$, onde n é a ordem de **A**, pelo simples motivo de que em $\mathbb{R}^n$ os conjuntos ortogonais têm no máximo n vetores.

Vale a pena fazer o seguinte comentário prático: embora o MGC, como o definimos neste capítulo, seja um processo finito, na prática, devido aos erros de arredondamento se torna infinito; mas, mais importante que isso, mesmo na ausência hipotética de erros computacionais, uma vez que muitos SELAS que surgem no campo prático são da ordem dos milhares, não se cogitaria em executar todas as iterações até atingir a solução exata; em vez disso, tendo o MGC convergência rápida, depois de algumas iterações já é atingida aproximação suficiente; ainda, o MCG é um método poderoso, que entra quando o tamanho do SELAS $\mathbf{Ax} = \mathbf{b}$ é grande, para o qual os métodos diretos claudicam, e, como método iterativo que é, preserva os elementos nulos de **A**, tornando-se, por isso, adequado para SELAS grandes e esparsos, hoje freqüentes.

2.7. Otimização. Vimos que o resíduo $\mathbf{r}_{k+1}$ na iteração $k+1$ está em $\mathscr{K}_{k+1}$. Escrevamos a (2.9) assim

$$\mathbf{d}_j := -\mathbf{r}_j + \beta_{j-1}\mathbf{d}_{j-1},\ j = 1, 2, \ldots$$

Agora façamos o produto interno de ambos os membros por $\mathbf{r}_{k+1}$:

$$\langle \mathbf{r}_{k+1}, \mathbf{d}_j \rangle = -\langle \mathbf{r}_{k+1}, \mathbf{r}_j \rangle + \beta_{j-1}\langle \mathbf{r}_{k+1}, \mathbf{d}_{j-1} \rangle.$$

Aplicando o Lema 2.3, essa igualdade leva a que

$$\langle \mathbf{r}_{k+1}, \mathbf{r}_j \rangle = 0, \text{ para } j = 0, 1, \ldots, k. \tag{2.15}$$

Portanto, pondo

$$S_{k+1} := \mathscr{G}\left(\mathbf{A}\mathbf{r}_o, \mathbf{A}^2\mathbf{r}_o, \cdots, \mathbf{A}^{k+1}\mathbf{r}_o\right),$$

obtemos

$$\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\mathbf{v} \rangle = 0, \text{ para todo } \mathbf{v} \in S_{k+1}, \tag{2.16}$$

pois $\mathbf{v} \in S_{k+1}$ se e só se $\mathbf{A}^{-1}\mathbf{v} \in \mathscr{K}_k$. O resíduo $\mathbf{r}_{k+1} \in \mathscr{K}_{k+1}$ é um vetor do tipo $\mathbf{v} + \mathbf{r}_o$, para algum $\mathbf{v} \in S_{k+1}$. Para referir a dependência desse $\mathbf{v}$ de k, ponhamos $\mathbf{r}_{k+1} := \mathbf{v}_{k+1} + \mathbf{r}_o$. Com isso, a (2.16) pode ser escrita

$$\langle \mathbf{r}_o + \mathbf{v}_{k+1}, \mathbf{A}^{-1}\mathbf{v} \rangle = 0, \text{ para todo } \mathbf{v} \in S_{k+1}. \tag{2.17}$$

Seja $\| \bullet \|$ a norma em $\mathbb{R}^n$ induzida pelo produto interno, isto é, definida por $\|\mathbf{x}\| = \sqrt{\langle \mathbf{x}, \mathbf{A}^{-1}\mathbf{x} \rangle}$. Pela lei de Pitágoras, usando a propriedade de ortogonalidade (2.17),

$$\begin{aligned} \|\mathbf{r}_o + \mathbf{v}\|^2 &= \|(\mathbf{r}_o + \mathbf{v}_{k+1}) + (\mathbf{v}_o - \mathbf{v}_{k+1})\|^2 = \|\mathbf{r}_o + \mathbf{v}_{k+1}\|^2 + \|\mathbf{v} - \mathbf{v}_{k+1}\|^2 \\ &\geq \|\mathbf{r}_o + \mathbf{v}_{k+1}\|^2, \end{aligned}$$

o que mostra que $\|\mathbf{r}_o + \mathbf{v}\|$ é mínimo sobre $\mathbf{r}_o + S_{k+1}$ se e somente se $\mathbf{r}_o + \mathbf{v} = \mathbf{r}_o + \mathbf{v}_{k+1} = \mathbf{r}_{k+1}$, onde $\mathbf{r}_{k+1}$ é

o resíduo computado pelo MGC na etapa $k+1$. Em termos do funcional quadrático f, definido em (2.1), isso corresponde a que $\mathbf{x}_{k+1}$ é ponto de mínimo de f restrito a $\mathbf{x}_o + S_{k+1}$, ou ainda, que $\mathbf{x}_{k+1}$ é a melhor aproximação da solução de $\mathbf{Ax} = \mathbf{b}$ entre todos os vetores em $\mathbf{x}_o + S_{k+1}$.

Organizemos os resultados obtidos acima e focalizemos a questão da minimização no

2.8. Teorema. *Seja* $\mathbf{Ax} = \mathbf{b}$ *um SELAS de ordem* $n \times n$, *onde* $\mathbf{A}$ *é a. p. d. em relação a um produto interno* $\langle \bullet, \bullet \rangle$. *Seja qualquer* $\mathbf{x}_o \in \mathbb{R}^n$, $\mathbf{r}_o := \mathbf{Ax}_o - \mathbf{b}$ *e* $\mathbf{d}_o := -\mathbf{r}_o$. *Como em* (2.3), (2.5) *e* (2.9), *definamos um método iterativo por*

$$\left.\begin{aligned} \mathbf{x}_{k+1} &:= \mathbf{x}_k + \tau_k \mathbf{d}_k \\ \mathbf{r}_{k+1} &:= \mathbf{r}_k + \tau_k \mathbf{A}\mathbf{d}_k \\ \mathbf{d}_{k+1} &:= -\mathbf{r}_{k+1} + \beta_k \mathbf{d}_k \end{aligned}\right\}, \text{para } k=0,1,2,\ldots$$

onde, como em (2.4) *e* (2.10), *os* τ_k e os β_k *são calculados por*

$$\tau_k = -\frac{\langle \mathbf{r}_k, \mathbf{d}_k \rangle}{\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \rangle} \quad e \quad \beta_k = \frac{\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{d}_k \rangle}{\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \rangle}.$$

Então

(*a*) $$\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\mathbf{r}_{k+1} \rangle \leq \langle \mathbf{r}, \mathbf{A}^{-1}\mathbf{r} \rangle, \text{ para todo } \mathbf{r} \in \mathbf{r}_o + S_{k+1},$$

onde $S_{k+1} := \mathscr{G}\left(\mathbf{A}\mathbf{r}_o, \mathbf{A}^2\mathbf{r}_o, \cdots, \mathbf{A}^{k+1}\mathbf{r}_o\right)$;

(*b*) *as seguintes propriedade de ortogonalidade, ou de ortogonalidade conjugada, valem*

$$\langle \mathbf{r}_{k+1}, \mathbf{r}_j \rangle = 0, \quad \text{para } j = 0, 1, 2, \cdots, k$$
$$\langle \mathbf{r}_{k+1}, \mathbf{d}_j \rangle = 0, \quad \text{para } j = 0, 1, 2, \cdots, k$$
$$\langle \mathbf{d}_{k+1}, \mathbf{A}\mathbf{d}_j \rangle = 0, \quad \text{para } j = 0, 1, 2, \cdots, k$$
$$\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{d}_j \rangle = 0, \quad \text{para } j = 0, 1, 2, \cdots, k-1$$
$$\langle \mathbf{r}_i, \mathbf{r}_j \rangle = \langle \mathbf{r}_o, \mathbf{r}_j \rangle, \quad \text{para } i = 0, 1, 2, \cdots, j-1;$$

(*c*) *se o produto interno é o usual, isto é,* $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t\mathbf{y}$, *indicando com* $\mathbf{e}_k := \mathbf{x}_k - \bar{\mathbf{x}}$ *o* ***erro*** *na iteração k, onde* $\bar{\mathbf{x}}$ *é a solução exata de* $\mathbf{Ax} = \mathbf{b}$, *então o MGC minimiza*

$$\mathbf{r}_{k+1}^t \mathbf{A}^{-1} \mathbf{r}_{k+1} = \mathbf{e}_{k+1}^t \mathbf{A}\mathbf{e}_{k+1};$$

aqui a primeira e a terceira propriedades em (*b*) *tornam-se, respectivamente,*

$$\mathbf{r}_{k+1}^t \mathbf{r}_j = 0, \quad \text{para } j = 0, 1, 2, \cdots, k$$
$$\mathbf{d}_{k+1}^t \mathbf{A}\mathbf{d}_j = 0, \quad \text{para } j = 0, 1, 2, \cdots, k.$$

Se o produto interno é dado por $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t \mathbf{A}\mathbf{y}$, *então o MGC dá, em cada iteração, a solução dos mínimos quadrados de* $\mathbf{Ax} = \mathbf{b}$ *sobre o correspondente espaço de Krylov. Nesse caso*

$$\mathbf{r}_{k+1}^t \mathbf{A}\mathbf{r}_j = 0, \quad \textit{para } j = 0, 1, 2, \cdots, k$$
$$\left(\mathbf{A}\mathbf{d}_{k+1}\right)^t \mathbf{A}\mathbf{d}_j = 0, \quad \textit{para } j = 0, 1, 2, \cdots, k.$$

Se o produto interno é dado por $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t \mathbf{A}^{-1}\mathbf{y}$, *então o MGC minimiza a norma 2 do erro em cada iteração, isto é,* $\|\mathbf{e}_{k+1}\|_2$ *é minimizado. Nesse caso,*

$$\mathbf{r}_{k+1}^t \mathbf{A}^{-1}\mathbf{r}_j = 0, \quad \textit{para } j = 0, 1, 2, \cdots, k$$
$$\mathbf{d}_{k+1}^t \mathbf{d}_j = 0, \quad \textit{para } j = 0, 1, 2, \cdots, k.$$

Observação. Para o último tipo de produto interno, o MGC não tem validade prática, porque o cálculo de produtos internos, como $\langle \mathbf{r}_k, \mathbf{r}_k \rangle = \mathbf{r}_k^t \mathbf{A}^{-1}\mathbf{r}_k$, *requer a inversão de* $\mathbf{A}$ *o que remonta a determinar a solução* $\mathbf{A}^{-1}\mathbf{b}$ *de* $\mathbf{Ax} = \mathbf{b}$.

Demonstração. (a) Esta é a propriedade da otimização provada na secção 2.7.

(b) A prova das quatro primeiras propriedades são dadas em (2.15), Lema 2.3 e secção 2.4. Resta-nos, então, provar a última propriedade. Para isso mostremos por indução que a fórmula de recorrência (2.5) implica a existência de um polinômio q_{i-1} de grau i-1 tal que

$$\mathbf{r}_i = \mathbf{r}_o + \mathbf{A}q_{i-1}(\mathbf{A})\mathbf{r}_o, \text{ para } i = 1, 2, \ldots \tag{2.18}$$

Para $i = 1$, a (2.18) é a (2.5) com $k = 0$ e com $q_o = -\tau_o$. Suponhamos a (2.18) válida para j=1, 2, . . . , i-1. Então, partindo da (2.5), e substituindo aí $\mathbf{d}_{i-1}$ com o uso da (2.9), escrevemos

$$\mathbf{r}_i = \mathbf{r}_{i-1} + \tau_{i-1}\mathbf{A}\mathbf{d}_{i-1} = \left(\mathbf{r}_o + \mathbf{A}q_{i-2}(\mathbf{A})\mathbf{r}_o\right) - \tau_{i-1}\left(\mathbf{A}\mathbf{r}_o + \mathbf{A}q_{i-2}(\mathbf{A})\mathbf{r}_o\right) + \tau_{i-1}\beta_{i-2}\mathbf{A}\mathbf{d}_{i-2}. \tag{2.19}$$

Mas, voltando a usar a (2.5) com base na hipótese da indução, vem

$$\mathbf{A}\mathbf{d}_{i-2} = \frac{\mathbf{r}_{i-1} - \mathbf{r}_{i-2}}{\tau_{i-2}} = \frac{\left(\mathbf{r}_o + \mathbf{A}q_{i-2}(\mathbf{A})\mathbf{r}_o\right) - \left(\mathbf{r}_o + \mathbf{A}q_{i-3}(\mathbf{A})\mathbf{r}_o\right)}{\tau_{i-2}}$$
$$= \frac{1}{\tau_{i-2}}\left[\mathbf{A}\left(q_{i-2}(\mathbf{A}) + q_{i-3}(\mathbf{A})\right)\right]\mathbf{r}_o.$$

Nesta última expressão o coeficiente de $\mathbf{r}_o$ é um polinômio de grau i-1 em $\mathbf{A}$. Substituindo $\mathbf{A}\mathbf{d}_{i-2}$ em (2.19), facilmente vemos que (2.19) se reduz à (2.18).

Agora, para $i < j$, lembrando que tomamos $\mathbf{r}_o = -\mathbf{d}_o$,

$$\langle \mathbf{r}_i, \mathbf{d}_j \rangle = \langle \mathbf{r}_o + \mathbf{A}q_{i-1}(\mathbf{A})\mathbf{r}_o, \mathbf{r}_j \rangle = \langle \mathbf{r}_o, \mathbf{d}_j \rangle - \langle \mathbf{A}q_{i-1}(\mathbf{A})\mathbf{d}_o, \mathbf{d}_j \rangle.$$

Como $q_{i-1}(\mathbf{A})\mathbf{d}_o = -q_{i-1}(\mathbf{A})\mathbf{r}_o$ está em $\mathscr{D}_{i-1}$, pelo Lema 2.3, $\langle \mathbf{A}q_{i-1}(\mathbf{A})\mathbf{d}_o, \mathbf{d}_j \rangle = 0$, e a parte (b) está completamente demonstrada.

(c) Aqui tudo o que temos que fazer ainda é provar três igualdades, um para cada tipo de produto interno citado, como segue.

Para o caso do produto interno canônico,

$$\begin{aligned}\langle \mathbf{e}_{k+1}, \mathbf{A}\mathbf{e}_{k+1}\rangle &= (\mathbf{x}_{k+1}-\bar{\mathbf{x}})^t \mathbf{A}(\mathbf{x}_{k+1}-\bar{\mathbf{x}}) \\ &= (\mathbf{A}(\mathbf{x}_{k+1}-\bar{\mathbf{x}}))^t \mathbf{A}^{-1}\mathbf{A}(\mathbf{x}_{k+1}-\bar{\mathbf{x}}) \\ &= (\mathbf{A}\mathbf{x}_{k+1}-\mathbf{b})^t \mathbf{A}^{-1}(\mathbf{A}\mathbf{x}_{k+1}-\mathbf{b}) \\ &= \langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\mathbf{r}_{k+1}\rangle.\end{aligned}$$

Para o caso do produto interno definido por **A**, basta observar que

$$\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\mathbf{r}_{k+1}\rangle = \mathbf{r}_{k+1}^t \mathbf{r}_{k+1} = \|\mathbf{r}_{k+1}\|_2^2.$$

E, finalmente, para o produto interno definido por $\mathbf{A}^{-1}$, basta notar que

$$\begin{aligned}\langle \mathbf{r}_{k+1}, \mathbf{A}^{-1}\mathbf{r}_{k+1}\rangle &= \mathbf{r}_{k+1}^t \mathbf{A}^{-1}\mathbf{A}^{-1}\mathbf{r}_{k+1} = (\mathbf{A}\mathbf{x}_{k+1}-\mathbf{b})^t \mathbf{A}^{-1}\mathbf{A}^{-1}(\mathbf{A}\mathbf{x}_{k+1}-\mathbf{b}) \\ &= (\mathbf{A}\mathbf{e}_{k+1})^t \mathbf{A}^{-1}\mathbf{A}^{-1}(\mathbf{A}\mathbf{e}_{k+1}) \\ &= \mathbf{e}_{k+1}^t \mathbf{e}_{k+1} \\ &= \|\mathbf{e}_{k+1}\|_2^2. \ \blacksquare\end{aligned}$$

Observação. A Parte (a) do Teorema 2.8 mostra que, mesmo que permitamos a $\mathbf{x}_{k+1}$, ou a $\mathbf{d}_{k+1}$, ou a ambos conter termos adicionais que envolvam direções de procura anteriores a $\mathbf{d}_k$, não podemos obter melhores aproximações que as fornecidas por (2.3). A forma curta da relação de recorrência constitui uma razão da eficiência do MGC padrão. Na verdade, devido à unicidade da solução ótima, numa iteração k, ambos os algoritmos de dois e de três termos geram a mesma aproximação da solução de $\mathbf{Ax} = \mathbf{b}$, no caso de desconsiderarmos os erros de arredondamento.

A relação entre os coeficientes das relações de recorrência de dois termos (2.3) e de três termos (1.4) pode ser encontrada como segue. Substituindo em (2.3) $\mathbf{d}_{k+1}$, definido por (2.9), escrevemos

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k(-\mathbf{r}_k + \beta_{k-1}\mathbf{d}_{k-1}).$$

Agora substituímos nessa igualdade a expressão de $\mathbf{d}_{k-1}$, extraída de (2.3), para obter

$$\mathbf{x}_{k+1} = \left(1 + \frac{\tau_k}{\tau_{k-1}}\beta_{k-1}\right)\mathbf{x}_k - \frac{\tau_k}{\tau_{k-1}}\beta_{k-1}\mathbf{x}_{k-1} - \tau_k\mathbf{r}_k. \qquad (2.20)$$

Comparando (2.20) com (1.4) (por razão óbvia, em (1.4) escrevemos $\hat{\beta}_k$ em lugar de β_k), encontramos

$$\mathbf{x}_{k+1} = \alpha_k\mathbf{x}_k + (1-\alpha_k)\mathbf{x}_{k-1} - \hat{\beta}_k\mathbf{r}_k,$$

onde

$$\alpha_k = 1 + \frac{\tau_k}{\tau_{k-1}}\beta_{k-1} \quad \text{e} \quad \hat{\beta}_k = \tau_k .$$

Na próxima secção veremos que $\beta_k > 0$ e $\tau_k > 0$, quando a matriz dos coeficientes **A** é a. p. d. em relação a um produto interno $\langle \bullet, \bullet \rangle$. Isso, junto com as relações apenas estabelecidas, constitui nova prova (Vd. Teorema 1.3(c)) de que $\alpha_k > 1$ e $\hat{\beta}_k > 0$, para todo k.

2.9. Complexidade Computacional. Diversas identidades permitem variações na formulação do MGC padrão. Por exemplo, o gradiente pode ser calculado diretamente por $\mathbf{r}_k := \mathbf{A}\mathbf{x}_k - \mathbf{b}$, em vez de fazê-lo pela fórmula de recorrência (2.5). Também existem outras expressões para β_k e α_k. Com $\mathbf{d}_k = -\mathbf{r}_k + \beta_{k-1}\mathbf{d}_{k-1}$, $\mathbf{A}\mathbf{d}_k = (1/\tau_k)(\mathbf{r}_{k+1} - \mathbf{r}_k)$ e a ortogonalidade, obtemos

$$-\langle \mathbf{r}_k, \mathbf{d}_k \rangle = \langle \mathbf{r}_k, \mathbf{r}_k \rangle,$$

$$\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{d}_k \rangle = \frac{1}{\tau_k}\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} - \mathbf{r}_k \rangle = \frac{1}{\tau_k}\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle$$

e

$$\tau_k \langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \rangle = \langle \mathbf{d}_k, \mathbf{r}_{k+1} - \mathbf{r}_k \rangle = \langle -\mathbf{d}_k, \mathbf{r}_k \rangle = \langle \mathbf{r}_k, \mathbf{r}_k \rangle.$$

Assim, a (2.4) e a (2.10) tomam a forma respectiva

$$\tau_k = \frac{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}{\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \rangle}$$

e

$$\beta_k = \frac{\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle}{\langle \mathbf{r}_k, \mathbf{r}_k \rangle}.$$

A vantagem dessas fórmulas consiste em que elas dispensam o cálculo dos produtos internos $\langle \mathbf{r}_k, \mathbf{d}_k \rangle$ e $\langle \mathbf{r}_{k+1}, \mathbf{A}\mathbf{d}_k \rangle$, o que reduz consideravelmente o número de operações. Além disso o produto interno $\langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle$ pode ser utilizado para teste de parada. A implementação computacional do correspondente algoritmo, a exemplo do algoritmo (1.21), encontra-se em (2.21).

Com relação ao *armazenamento em cada iteração*, para executar o algoritmo (2.21),

- é necessário armazenar 4 vetores do $\mathbb{R}^n$: $\mathbf{h}_k$, $\mathbf{x}_k$, $\mathbf{r}_k$ e $\mathbf{d}_k$;
- o armazenamento relativo a **A** depende da esparsidade, e da estrutura de dados escolhida; normalmente, nos problemas importantes de hoje, **A** é de dimensão elevada e esparsa e a estrutura simples usual de tabela implica um armazenamento demais custoso; daí a relevância em criar capacidade específica de armazenamento barato de matrizes esparsas e de operar com elas, como já efetivamente acontece, por exemplo, com o software MATLAB.

Sem contar as operações aritméticas, para cada k, a complexidade computacional desse algoritmo envolve

- uma multiplicação do vetor $\mathbf{d}_k$ pela matriz **A**, sendo esta a operação mais custosa;
- dois produtos internos;
- seis operações vetoriais - multiplicações por escalar e adições vetoriais.

$$
\begin{cases}
\text{Escolher } \mathbf{x}_0 \text{ e } \varepsilon > 0; \\
\text{seja } \mathbf{r}_0 := \mathbf{A}\mathbf{x}_0 - \mathbf{b},\ \delta_0 := \langle \mathbf{r}_0, \mathbf{r}_0 \rangle,\ \mathbf{d}_0 := \mathbf{r}_0; \\
\text{testar a convergência; se } \langle \mathbf{r}_0, \mathbf{r}_0 \rangle \geq \varepsilon, \text{ continuar;} \\
\text{para } k = 0,\ 1,\ 2,\ \ldots,\ \text{fazer} \\
\mathbf{h}_k := \mathbf{A}\mathbf{d}_k; \\
\tau_k := \delta_k / \langle \mathbf{d}_k, \mathbf{h}_k \rangle; \\
\mathbf{x}_{k+1} := \mathbf{x}_k + \tau_k \mathbf{d}_k; \\
\mathbf{r}_{k+1} := \mathbf{r}_k + \tau_k \mathbf{h}_k;\ \delta_{k+1} := \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle; \\
\text{testar a convergência; se } \langle \mathbf{r}_{k+1}, \mathbf{r}_{k+1} \rangle \geq \varepsilon, \text{ continuar;} \\
\beta_k := \delta_{k+1} / \delta_k; \\
\mathbf{d}_{k+1} := -\mathbf{r}_{k+1} + \beta_k \mathbf{d}_k.
\end{cases}
\tag{2.21}
$$

Os dois produtos internos com as três fórmulas de recursão requerem, em cada passo, $5n$ multiplicações e $5n$ adições, onde n é a ordem de **A**. Aqui há uma clara vantagem sobre o algoritmo (1.21), que descreve a forma de recorrência de três termos, em que ocorrem em torno de $6n$ multiplicações e $6n$ adições, em cada iteração, além da presença de uma multiplicação de vetor por matriz em ambos os algoritmos.

2.10. Exemplo. Nesse exemplo fizemos duas comparações do desempenho das formas de recorrência de três e dois termos do MGC, com os respectivos programas *mgc3t.m* e *mgc2t.m* implementados na linguagem do MATLAB. A execução foi feita num microcomputador pentium com 32 megabytes de memória e 133 megahertz de velocidade. O produto interno usado foi o canônico. No SELAS $\mathbf{Ax} = \mathbf{b}$ considerado,

$$
\mathbf{A} = \text{tridiag}(-1,4,-1) \text{ e } \mathbf{b} = \mathbf{b}(i) = \begin{cases} 2, \text{ se } i \text{ é par} \\ -1, \text{ se } i \text{ é impar.} \end{cases}
$$

Usamos um vetor randômico $\mathbf{x}_0$ com componentes inteiras -2, -1, 0, 1, 2 para inicializar o processo, uma tolerância $tol = 10^{-15}$ para o erro, o número máximo de iterações $iter = 100$, e a ordem n de **A** igual a:

(a) 3000.

Resultados obtidos:

- **Forma de recorrência de três termos**:

  Número de *flops* realizadas: 2 754 181
  Tempo de execução do programa (em s): 1,32
  Número de iterações realizadas: 31
  Estimativa de erro: **6,794182415340**597 $\times 10^{-16}$

- **Forma de recorrência de dois termos**

  Número de *flops* realizadas: 2 264 983
  Tempo de execução do programa (em s): 0,83
  Número de iterações realizadas: 31
  Estimativa de erro: **6,794182415340**612 $\times 10^{-16}$

(b) 4000.

Resultados obtidos:

- **Forma de recorrência de três termos**:

  Número de *flops* realizadas: 3 672 181
  Tempo de execução do programa (em s): 1,7
  Número de iterações realizadas: 31
  Estimativa de erro: $\mathbf{7{,}575421742913410} \times 10^{-16}$

- **Forma de recorrência de dois termos**

  Número de *flops* realizadas: 3 019 965
  Tempo de execução do programa (em s): 1,43
  Número de iterações realizadas: 31
  Estimativa de erro: $\mathbf{7{,}575421742913401} \times 10^{-16}$

Análise dos Resultados. Considerando que o número de iterações e o valor do erro, que é a norma usual do resíduo na última iteração, são quase os mesmos em ambas as formas com o mesmo n, concluímos, observando o número de *flops*, que o algoritmo da forma de recorrência de três termos produziu um custo computacional maior, por iteração, do que o algoritmo da forma de recorrência de dois termos. Isso era de fato esperado. Como consequência, o tempo de execução para a primeira forma, até obter uma aproximação dentro dos parâmetros estipulados, é maior que o tempo de execução para a forma curta até obter a mesma aproximação.

Por razões óbvias, as soluções computadas **x** e **y**, em cada uma das duas simulações não são exibidas aqui. No entanto, observamos que a norma 2 da diferença **x - y** (Vd. apêndice) é muito pequena, o que nos leva a concluir que **x** e **y** praticamente coincidem. Os resultados exibidos acima encontram-se em apêndice. ▌

2.11. Precondicionamento no MGC padrão. A explanação que segue considera os aspetos computacionais que envolvem a versão precondicionada do MGC padrão. Tem, como ponto de partida, o exposto na secção 1.5. Não faremos a análise do aumento da razão de convergência devido ao precondicionamento.

Como já vimos, se a matriz precondicionadora **C** for s. p. d., podemos usar o produto interno definido por **C**, $\langle \mathbf{x}, \mathbf{y} \rangle := \mathbf{x}^t\mathbf{C}\mathbf{y}$, e considerar os *resíduos precondicionados*

$$\mathbf{h}_k := \mathbf{C}^{-1}(\mathbf{A}\mathbf{x}_k - \mathbf{b}).$$

Como, por hipótese, **A** também é s. p. d., a matriz $\mathbf{C}^{-1}\mathbf{A}$ é a. p. d. em relação ao citado produto interno. Então, substituindo **A** por $\mathbf{C}^{-1}\mathbf{A}$ no algoritmo (2.21) e no subespaço de Krylov, a implementação computacional do MGC padrão precondicionado é dada por (2.22). A declaração "$\mathbf{h}:=\mathbf{C}^{-1}\mathbf{r}$", que aparece no algoritmo, deve ser interpretada como "resolver o SELAS $\mathbf{Ch} = \mathbf{r}$".

O funcional f, definido em (2.1), aqui passa a ser dado por

$$\begin{aligned} f(x) &= \frac{1}{2}\left\langle \mathbf{h}, \left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\mathbf{h} \right\rangle \\ &= \frac{1}{2}\left\langle \mathbf{C}^{-1}\mathbf{r}, \left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\mathbf{C}^{-1}\mathbf{r} \right\rangle = \frac{1}{2}\left(\mathbf{C}^{-1}\mathbf{r}\right)^t\mathbf{C}\left(\mathbf{C}^{-1}\mathbf{A}\right)^{-1}\mathbf{C}^{-1}\mathbf{r} = \frac{1}{2}\mathbf{r}^t\left(\mathbf{C}^{-1}\right)^t\mathbf{C}\mathbf{A}^{-1}\mathbf{C}\mathbf{C}^{-1}\mathbf{r} \\ &= \frac{1}{2}\mathbf{r}^t\mathbf{A}^{-1}\mathbf{r}, \end{aligned}$$

por onde observamos que o algoritmo

$$
\begin{aligned}
&\text{Encontrar uma matriz precondicionador } \mathbf{C};\\
&\text{escolher } \mathbf{x}_0 \text{ e } \varepsilon > 0;\\
&\text{seja } \mathbf{r}_0 := \mathbf{A}\mathbf{x}_0 - \mathbf{b},\ \mathbf{h}_0 := \mathbf{C}^{-1}\mathbf{r}_0,\ \mathbf{d}_0 := -\mathbf{h}_0, \delta_0 := \mathbf{r}_0^t\mathbf{h}_0;\\
&\text{testar a convergência; se } \delta_0 \geq \varepsilon, \text{ continuar;}\\
&\text{para } k = 0,\ 1,\ 2,\ \ldots,\ \text{fazer}\\
&\mathbf{p}_k := \mathbf{A}\mathbf{d}_k;\\
&\tau_k := \delta_k / \mathbf{d}_k^t\mathbf{p}_k;\\
&\mathbf{x}_{k+1} := \mathbf{x}_k + \tau_k\mathbf{d}_k;\\
&\mathbf{r}_{k+1} := \mathbf{r}_k + \tau_k\mathbf{p}_k;\\
&\mathbf{h}_{k+1} := \mathbf{C}^{-1}\mathbf{r}_{k+1};\ \delta_{k+1} := \mathbf{r}_{k+1}^t\mathbf{h}_{k+1};\\
&\text{testar a convergência; se } \delta_{k+1} \geq \varepsilon, \text{ continuar;}\\
&\beta_k := \delta_{k+1}/\delta_k;\\
&\mathbf{d}_{k+1} := -\mathbf{h}_{k+1} + \beta_k\mathbf{d}_k.
\end{aligned}
\tag{2.22}
$$

minimiza o mesmo funcional que na versão não-precondicionada, desde que nessa versão tomemos o produto interno canônico (Teorema 2.8 (d)), mas agora sobre o espaço de Krylov

$$\mathscr{K}_k = \mathscr{G}\left(\mathbf{r}_0, \mathbf{C}^{-1}\mathbf{A}\mathbf{r}_0, \cdots, \left(\mathbf{C}^{-1}\mathbf{A}\right)^k \mathbf{r}_0\right). \tag{2.23}$$

Além disso,

$$\mathbf{d}_k^t\mathbf{A}\mathbf{d}_k = \left\langle \mathbf{d}_k, \mathbf{C}^{-1}\mathbf{A}\mathbf{d}_j \right\rangle = 0, \text{ para } j = 0,\ 1,\ \ldots,\ k.$$

Com uma escolha adequada da matriz precondicionadora **C**, no subespaço (2.23) a minimização de $f(\mathbf{x})$ pode ser muito mais rápida que o correspondente subespaço não precondicionado.

A escolha da melhor matriz precondicionadora é ainda área de pesquisa. Até agora, muitos precondicionadores, alguns bem sofisticados, tem sido desenvolvidos e, independente de como escolher um, o fato é que, para SELAS oriundos de problemas de aplicação, o MGC deve ser usado com um precondicionador.

Sem a intenção de nos aprofundar no assunto do precondicionamento, faremos uma rápida exposição sobre algumas matrizes precondicionadoras encontradas na literatura.

(a) Como já vimos, se **A** e **C** são s. p. d., a matriz $\mathbf{C}^{-1}\mathbf{A}$ é a. p. d. em relação ao produto interno definido por **C**. Resolver o SELAS $\mathbf{C}^{-1}\mathbf{A}\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}$ pelo MGC padrão requer que também $\mathbf{C}^{-1}\mathbf{A}$ seja s. p. d., o que não acontece em geral, pois $\mathbf{C}^{-1}$ pode destruir a simetria. A escolha de **C** exige, então, um cuidado a mais. Por exemplo, a escolha $\mathbf{C} := \sqrt{\mathbf{A}}$ conduz a uma matriz $\mathbf{C}^{-1}\mathbf{A} = \mathbf{C}$ s. p. d. e a um SELAS $\mathbf{C}\mathbf{x} = \mathbf{C}^{-1}\mathbf{b}$ equivalente ao SELAS original $\mathbf{A}\mathbf{x} = \mathbf{b}$, como é imediato ver, com a vantagem de que $\text{cond}_2(\mathbf{C}) = \sqrt{\text{cond}_2(\mathbf{A})}$.

(b) O problema de não obter uma matriz $\mathbf{C}^{-1}\mathbf{A}$ s. p. d. pode ser contornado usando precondicionamento bilateral: se tomamos $\mathbf{S} \approx \left(\sqrt{\mathbf{A}}\right)^{-1}$, a matriz **SAS** será s. p. d. se **S** o for, e seus autovalores se

agruparão em torno de 1, sendo, então, $\text{cond}_2(\mathbf{SAS}) \approx 1$. Resolvemos o SELAS

$$\mathbf{SASy} = \mathbf{Sb}, \text{ equivalente a } \mathbf{SAx} = \mathbf{Sb},$$

onde, portanto, $\mathbf{Sy} = \mathbf{x}$, o que monstra que a solução $\bar{\mathbf{x}}$ de $\mathbf{Ax} = \mathbf{b}$, ao final, pode ser recuperada multiplicando a solução $\bar{\mathbf{y}}$ de $\mathbf{SASy} = \mathbf{b}$ por **S**. Aparentemente, na aplicação do MGC a $\mathbf{SASy} = \mathbf{Sb}$, ocorre o acréscimo de uma multiplicação por **A** e de duas por **S**. De fato não é assim, pois, se $\hat{\mathbf{y}}_k$, $\hat{\mathbf{r}}_k$ e $\hat{\mathbf{d}}_k$ são as aproximações, os resíduos e as direções de procura para o MGC aplicado a $\mathbf{SASy} = \mathbf{Sb}$, pondo

$$\mathbf{x}_k := \mathbf{S}\hat{\mathbf{y}}_k,\ \mathbf{r}_k := \mathbf{S}^{-1}\hat{\mathbf{r}}_k,\ \mathbf{d}_k := \mathbf{S}\hat{\mathbf{d}}_k \text{ e } \mathbf{z}_k = \mathbf{S}\hat{\mathbf{r}}_k,$$

podemos executar a iteração *k* diretamente em termos de $\mathbf{x}_k$, **A** e $\mathbf{S}^2$.

(c) Sendo **A** s. p. d., admite a fatoração de Choleski $\mathbf{A} = \mathbf{LL}^t$, onde **L** é uma matriz triangular inferior única, ou $\mathbf{A} = \mathbf{LDL}^t$, onde **L** é uma matriz triangular inferior com diagonal unitária e **D**, uma matriz diagonal com todos seus elementos diagonais estritamente positivos. A fatoração de Choleski não preserva os elementos nulos de **A**. Mas, durante o processo da fatoração, no momento em que ocorre uma substituição de um zero por um elemento não nulo, podemos descartar esta substituição e manter o zero. De outra maneira: seja $\mathbf{A} = [a_{ij}]$ e $\mathbf{L} = [l_{ij}]$; se $a_{ij} \neq 0$, calculamos l_{ij}; se $a_{ij} = 0$ fazemos $l_{ij} = 0$. Obteremos

$$\mathbf{A} = \mathbf{LL}^t + \mathbf{R} \quad \text{ou} \quad \mathbf{A} = \mathbf{LDL}^t + \mathbf{R}, \tag{2.24}$$

com $\mathbf{R} \neq \mathbf{0}$. Na maioria da vezes $\|\mathbf{R}\| \approx 0$, e uma boa escolha para matriz precondicionadora é $\mathbf{C} = \mathbf{LL}^t$ ou $\mathbf{C} = \mathbf{LDL}^t$. Com a primeira escolha, resolver $\mathbf{Ch} = \mathbf{r}$ conforme algoritmo (2.22) é equivalente a resolver os dois SELAS triangulares $\mathbf{Ly} = \mathbf{r}$ e $\mathbf{L}^t\mathbf{h} = \mathbf{y}$. Com a segunda escolha, a equivalência ocorre com a resolução dos SELAS $\mathbf{Ly} = \mathbf{r}$, $\mathbf{Dz} = \mathbf{y}$ e $\mathbf{L}^t\mathbf{h} = \mathbf{y}$.

(d) A decomposição de Choleski também pode ser aplicada à matriz precondicionadora **C**, desde que seja s. p. d., por exemplo $\mathbf{C} := \mathbf{LL}^t$ em (2.24), com o fim de obter um precondicionamento bilateral. Esse caminho trará vantagem se $\text{cond}(\mathbf{C}^{-1}\mathbf{A})$ for melhor que $\text{cond}(\mathbf{A})$. Para explicar isso, suponhamos que **C** seja s. p. d. e, mudando a notação, seja $\mathbf{C} = \mathbf{LL}^t$ a fatoração de Choleski de **C**. Então a matriz $\mathbf{L}^{-1}\mathbf{AL}^{-t}$ é s. p. d. e aplicamos o MGC ao SELAS

$$\mathbf{L}^{-1}\mathbf{AL}^{-t}\mathbf{y} = \mathbf{L}^{-1}\mathbf{b}, \text{ onde } \mathbf{y} = \mathbf{L}^t\mathbf{x}.$$

Observemos que as matrizes $\mathbf{C}^{-1}\mathbf{A}$ e $\mathbf{L}^{-1}\mathbf{AL}^{-t}$ são semelhantes e, portanto, possuem os mesmos autovalores.

(e) Mediante a série de Neumann podemos obter uma aproximação arbitrária direta da inversa da matriz dos coeficientes **A**. Suponhamos que $\mathbf{A} = \mathbf{I} - \mathbf{B}$, onde **B** tem raio espectral menor que 1. Nesse caso **A** é inversível e

$$\mathbf{A}^{-1} = \sum_{k=0}^{\infty} \mathbf{B}^k.$$

A mencionada aproximação é obtida por truncamento da série.

3. O MGC para SELAS Singulares e Quase Singulares

3.1. Preliminares. No desenvolvimento que segue, salvo declaração em contrário, o produto interno será o usual. Antes de entrar no assunto do capítulo, façamos algumas observações em torno do MGC para SELAS não-simétricos. O MGC pode ser modificado para resolver SELAS $\mathbf{Ax} = \mathbf{b}$, embora $\mathbf{A}$ não seja simétrica e, conseqüentemente, não seja positiva definida. No breve histórico, exposto na Introdução, citamos algumas variações do MGC que surgiram para resolver SELAS dessa classe, mas não é nosso objetivo aqui desenvolver os métodos lá citados e sim fazer a rápida explanação que segue.

Se $\mathbf{A}$ é não-singular e não é s. p. d., podemos resolver $\mathbf{Ax} = \mathbf{b}$ aplicando o MGC à equação normal

$$\mathbf{A}^t\mathbf{Ax} = \mathbf{A}^t\mathbf{b}, \tag{3.1}$$

uma vez que $\mathbf{A}^t\mathbf{A}$ é s. p. d.. O método que resolve tal equação é o MGC *para equação normal que minimiza o resíduo* (CGNR). A razão para esse nome é que a propriedade de minimização do MGC, aplicada à (3.1), mostra que, se $\bar{\mathbf{x}}$ é a solução de $\mathbf{Ax} = \mathbf{b}$,

$$\begin{aligned}\|\mathbf{x}-\bar{\mathbf{x}}\|^2_{\mathbf{A}^t\mathbf{A}} &= (\mathbf{x}-\bar{\mathbf{x}})^t\mathbf{A}^t\mathbf{A}(\mathbf{x}-\bar{\mathbf{x}})^t \\ &= (\mathbf{Ax}-\mathbf{A}\bar{\mathbf{x}})^t(\mathbf{Ax}-\mathbf{A}\bar{\mathbf{x}}) = (\mathbf{Ax}-\mathbf{b})^t(\mathbf{Ax}-\mathbf{b}) = \|\mathbf{r}\|_2^2\end{aligned}$$

é minimizado sobre $\mathbf{x}_o+\mathscr{K}_k$ na iteração k. Alternativamente, podemos resolver

$$\mathbf{AA}^t\mathbf{y} = \mathbf{b}, \tag{3.2}$$

e depois fazer $\mathbf{x} = \mathbf{A}^t\mathbf{y}$. O MGC aplicado à (3.2) assume a designação de *gradiente conjugado para a equação normal que minimiza o erro* (CGNE), e a razão desse nome é que a propriedade acima citada, aplicada à (3.2), mostra que, se $\bar{\mathbf{y}}$ é a solução de (3.2),

$$\begin{aligned}\|\mathbf{y}-\bar{\mathbf{y}}\|^2_{\mathbf{AA}^t} &= (\mathbf{y}-\bar{\mathbf{y}})^t\mathbf{AA}^t(\mathbf{y}-\bar{\mathbf{y}}) \\ &= (\mathbf{A}^t\mathbf{y}-\mathbf{A}^t\bar{\mathbf{y}})^t(\mathbf{A}^t\mathbf{y}-\mathbf{A}^t\bar{\mathbf{y}}) = \|\mathbf{x}-\bar{\mathbf{x}}\|_2^2\end{aligned}$$

é minimizado sobre $\mathbf{y}_o+\mathscr{K}_k$ na iteração k.

A grande vantagem desses dois métodos é que toda a teoria válida para o MGC é também válida para eles. Das desvantagens, destacamos três que podem influir em menor ou maior grau, mas que devem ser consideradas.

(a) O número de condição de $\mathbf{A}^t\mathbf{A}$ (igual ao de $\mathbf{A}\mathbf{A}^t$),

$$\rho(\mathbf{A}^t\mathbf{A}) = \lambda_{max}(\mathbf{A}^t\mathbf{A})/\lambda_{min}(\mathbf{A}^t\mathbf{A}),$$

razão do maior e o menor autovalor, é o quadrado do número de condição de **A**, o que faz a convergência para (3.1) e (3.2) ser, em geral, significativamente mais lenta [121].

(b) São necessários dois produtos matriz-vetor em cada iteração no CGNR, a saber, $\mathbf{w} := \mathbf{A}^t\mathbf{A}\mathbf{d} = \mathbf{A}^t(\mathbf{A}\mathbf{d})$, assim como, semelhantemente, no CGNE. Observamos que a colocação dos parênteses no cálculo de **w** é providencial pois não é aconselhável efetuar o produto $\mathbf{A}^t\mathbf{A}$ (ou $\mathbf{A}\mathbf{A}^t$) explicitamente, pois a matriz desse produto pode ser menos esparsa que **A** [121].

(c) Somos obrigados a determinar a ação de $\mathbf{A}^t$ sobre um vetor como parte dos produtos matriz-vetor que o cálculo de **w** envolve. Esta é a desvantagem principal. Em problemas não lineares existem casos onde isso não é possível [84]. No caso de SELAS, que é nosso objeto de estudo, esta desvantagem não conta.

3.2. O MGC para SELAS singulares. Suponhamos que a matriz dos coeficientes do SELAS $\mathbf{A}\mathbf{x} = \mathbf{b}$ seja singular e simétrica positiva semidefinida, isto é,

$$\langle \mathbf{x}, \mathbf{A}\mathbf{x} \rangle \geq 0, \text{ para todo } x \in \mathbb{R}^n.$$

Sejam $\mathcal{R}(\mathbf{A})$ e $\mathcal{N}(\mathbf{A})$ o espaço coluna e o espaço nulo (núcleo) de **A**, respectivamente. Na solução de um sistema singular $\mathbf{A}\mathbf{x} = \mathbf{b}$, devemos considerar dois efeitos da singularidade:

- Consistência: o SELAS pode ser inconsistente, isto é, $\mathbf{b} \notin \mathcal{R}(\mathbf{A}) = \mathcal{N}(\mathbf{A}^t)^{\perp}$, o que significa que não há solução.
- Não-unicidade: se o SELAS é consistente, existem infinitas soluções, sendo $\hat{\mathbf{x}} + \mathcal{N}(\mathrm{A})$, o conjunto-solução, desde que $\hat{\mathbf{x}}$ seja uma solução de $\mathbf{A}\mathbf{x} = \mathbf{b}$.

No caso de $\mathbf{A}\mathbf{x} = \mathbf{b}$ ser inconsistente, podemos resolver a equação normal (3.1), ou (3.2), que traduzem SELAS consistentes. No entanto, como já foi comentado na secção 3.1, o número de condição de $\mathbf{A}^t\mathbf{A}$, ou de $\mathbf{A}\mathbf{A}^t$, que aqui é calculado por

$$\rho(\mathbf{A}^t\mathbf{A}) = \lambda_{max}(\mathbf{A}^t\mathbf{A})/\lambda^{+}_{min}(\mathbf{A}^t\mathbf{A}),$$

onde $\lambda^{+}_{min}(\mathbf{A}^t\mathbf{A})$ denota o menor autovalor positivo de $\mathbf{A}^t\mathbf{A}$, pode ser muito grande, pois é o quadrado de $\rho(\mathbf{A})$. Para contornar este problema, conforme secções 1.5 e 2.11, podemos fazer o uso de uma adequada matriz precondicionadora .

Há outra abordagem: se conhecemos uma base $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\}$ de $\mathcal{N}(\mathbf{A}^t)$, podemos projetar **b** sobre $\mathcal{N}(\mathbf{A}^t)$, obter

$$\tilde{\mathbf{b}} := \mathbf{b} - \frac{\langle \mathbf{b}, \mathbf{v}_1 \rangle}{\langle \mathbf{v}_1, \mathbf{v}_1 \rangle}\mathbf{v}_1 - \frac{\langle \mathbf{b}, \mathbf{v}_2 \rangle}{\langle \mathbf{v}_2, \mathbf{v}_2 \rangle}\mathbf{v}_2 - \cdots - \frac{\langle \mathbf{b}, \mathbf{v}_k \rangle}{\langle \mathbf{v}_k, \mathbf{v}_k \rangle}\mathbf{v}_k,$$

que é ortogonal a $\mathcal{N}(\mathbf{A}^t)$, e resolver o SELAS consistente $\mathbf{A}\mathbf{x} = \tilde{\mathbf{b}}$. Sua solução é aceita como uma solução de $\mathbf{A}\mathbf{x} = \mathbf{b}$.

Observamos que ambas as abordagens apresentadas acima transformam um SELAS inconsistente em um SELAS consistente, cujas soluções nada mais são que as soluções dos mínimos quadrados do SELAS original.

Para aprofundar este estudo, admitamos que $\mathbf{Ax} = \mathbf{b}$ seja consistente (caso contrário aplicamos previamente um dos procedimentos acima), isto é, $\mathbf{b}\in\mathscr{R}(\mathbf{A})$. Definamos $\mathbf{d}_o := -\mathbf{A}^t\mathbf{r}_o$ e $\mathbf{r}_o := \mathbf{Ax}_o - \mathbf{b}$, onde $\mathbf{x}_o$ é um vetor inicial qualquer. Resulta que $\mathbf{d}_o\in\mathscr{N}(\mathbf{A})^\perp$. Para mostrar isso, seja $\mathbf{y}\in\mathscr{N}(\mathbf{A})$, isto é, $\mathbf{Ay} = \mathbf{0}$. Então

$$\begin{aligned}\langle \mathbf{d}_o, \mathbf{y}\rangle &= -\langle \mathbf{A}^t\mathbf{r}_o, \mathbf{y}\rangle = -\langle \mathbf{A}^t\mathbf{Ax}_o - \mathbf{A}^t\mathbf{b}, \mathbf{y}\rangle \\ &= -\langle \mathbf{A}^t\mathbf{Ax}_o, \mathbf{y}\rangle - \langle \mathbf{A}^t\mathbf{b}, \mathbf{y}\rangle \\ &= -\mathbf{x}_o^t\mathbf{A}^t\mathbf{Ay} + \mathbf{b}^t\mathbf{Ay} \\ &= (\mathbf{Ax}_o)^t\mathbf{Ay} + \mathbf{b}^t\mathbf{Ay} \\ &= -\langle \mathbf{Ax}_o, \mathbf{Ay}\rangle + \langle \mathbf{b}, \mathbf{Ay}\rangle = 0.\end{aligned}$$

Sabemos que $\mathbb{R}^n = \mathscr{N}(\mathbf{A}) \oplus \mathscr{N}(\mathbf{A})^\perp$, ou seja, para todo $\mathbf{x}\in\mathbb{R}^n$, existe um único $\hat{\mathbf{x}} \in \mathscr{N}(\mathbf{A})^\perp$ e um único $\mathbf{s}\in\mathscr{N}(\mathbf{A})$ tal que $\mathbf{x} = \hat{\mathbf{x}} + \mathbf{s}$. É claro que $\mathbf{Ax} = \mathbf{0} \Leftrightarrow \mathbf{A}\hat{\mathbf{x}} = \mathbf{0}$. Tomamos então $\mathbf{x}_o = \hat{\mathbf{x}}_o + \mathbf{s}_o$, com $\hat{\mathbf{x}}_o \in \mathscr{N}(\mathbf{A})^\perp$ e $\mathbf{s}_o\in\mathscr{N}(\mathbf{A})$. Utilizando esses dados para inicializar o algoritmo (2.21), afirmamos que, para todo k, $\mathbf{x}_k$ terá a mesma componente $\mathbf{s}_o$ de $\mathbf{x}_o$ na direção de $\mathscr{N}(\mathbf{A})$. De fato, suponhamos por indução que, para um k qualquer fixo, $\mathbf{x}_k$ tenha a componente $\mathbf{s}_o$ na direção de $\mathscr{N}(\mathbf{A})$, isto é,

$$\mathbf{x}_k = \hat{\mathbf{x}}_k + \mathbf{s}_o,$$

com $\hat{\mathbf{x}}_k \in \mathscr{N}(\mathbf{A})^\perp$. Podemos escolher a direção de procura $\mathbf{d}_k$ de maneira que sua componente na direção de $\mathscr{N}(\mathbf{A})$ seja nula, ou melhor, $\mathbf{d}_k$ ortogonal ao $\mathscr{N}(\mathbf{A})$. A própria definição de $\mathbf{d}_o$ inicia essa escolha[1]. Então $\langle \mathbf{d}_k, \mathbf{Ad}_k\rangle \neq 0$. Calculemos a iteração de ordem k+1 de (2.21):

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \tau_k\mathbf{d}_k,$$

onde

$$\tau_k = \frac{\langle \mathbf{r}_k, \mathbf{r}_k\rangle}{\langle \mathbf{d}_k, \mathbf{Ad}_k\rangle}.$$

Logo

$$\mathbf{x}_{k+1} = (\hat{\mathbf{x}}_k + \mathbf{s}_o) + \tau_k\mathbf{d}_k = (\hat{\mathbf{x}}_k + \tau_k\mathbf{d}_k) + \mathbf{s}_o = \hat{\mathbf{x}}_{k+1} + \mathbf{s}_o.$$

Agora, se $\mathbf{Ax} = \mathbf{b}$ é consistente e $\mathbf{A}$ é simétrica, então $\mathbf{b}\in\mathscr{R}(\mathbf{A}) = \mathscr{N}(\mathbf{A}^t)^\perp = \mathscr{N}(\mathbf{A})^\perp$ e, nesse caso, como $\mathbf{r}_o\in\mathscr{R}(\mathbf{A})$, podemos tomar $\mathbf{d}_o := -\mathbf{r}_o$, de forma que, assim definido, $\mathbf{d}_o\in\mathscr{N}(\mathbf{A})^\perp$ e o acima exposto se aplica.

---

[1] Notemos que não há necessidade de esgotar as dimensões de $\mathbb{R}^n$, na execução do MGC, para obter teoricamente uma solução exata, mas apenas as n - dim$\mathscr{N}(\mathbf{A})$ dimensões fora do núcleo de $\mathbf{A}$, porque procuramos uma solução particular, que, somada com $\mathscr{N}(\mathbf{A})$, vai produzir o conjunto solução. Então nunca vai precisar que $\mathbf{d}_k \in \mathscr{N}(\mathbf{A})$.

3.3. Exemplo. Consideremos o SELAS $\mathbf{Ax} = \mathbf{b}$,

$$\begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 0 \\ 2 \\ 1 \end{bmatrix},$$

onde $\mathbf{A}$ é singular, simétrica positiva semidefinida com $\dim \mathcal{N}(\mathbf{A}) = 1$. O sistema é consistente pois $\mathbf{b} \in \mathcal{R}(\mathbf{A})$. Tomando $\mathbf{x}_o = [1\ 1\ 0\ 0]^t$ como vetor inicializador do algoritmo (11.25), e executando-o no MATLAB[2], obtemos, em 3 iterações, a solução

$$\tilde{\mathbf{x}} = [1/2\ \ 1\ \ 2/3\ \ 1]^t.$$

Para exibir a solução geral do SELAS devemos conhecer o $\mathcal{N}(\mathbf{A})$. Executando novamente o algoritmo (11.25), com a mesma aproximação inicial $\mathbf{x}_o$, obtemos, com apenas uma iteração, a solução de $\mathbf{Ax} = \mathbf{0}$,

$$\mathbf{s}_o = [0\ 1\ 0\ 0]^t.$$

A solução geral do SELAS pode ser escrita na forma $\mathbf{x} = \tilde{\mathbf{x}} + \mathbf{t}\mathbf{s}_o, \mathbf{t} \in \mathbb{R}$. ▌

3.4. O MGC para SELAS Quase Singulares. A denominação *quase singulares* refere-se a SELAS $\mathbf{Ax} = \mathbf{b}$ tais que $\mathbf{A}$ possui um ou mais autovalores muito próximos de zero. Nesse caso, o número de condição de $\mathbf{A}$ pode ser muito grande, provocando um mau condicionamento para o SELAS. Por questão de simplicidade, consideremos primeiramente um SELAS, cuja matriz dos coeficientes tenha apenas um autovalor λ_1 próximo de zero, com um autovetor correspondente $\mathbf{v}_1$. Supomos também que $\mathbf{A}$ seja s. p. d. Podemos escrever a solução $\tilde{\mathbf{x}}$ de $\mathbf{Ax} = \mathbf{b}$ como

$$\tilde{\mathbf{x}} = \frac{\mathbf{b}^t\mathbf{v}_1}{\lambda_1}\mathbf{v}_1 + \tilde{\mathbf{x}}_d, \tag{3.3}$$

onde

$$\tilde{\mathbf{x}}_d := \sum_{i=2}^{n} \frac{\mathbf{b}^t\mathbf{v}_i}{\lambda_i}\mathbf{v}_i,$$

e $(\lambda_i, \mathbf{v}_i)$ são autopares de $\mathbf{A}$ com os autovetores $\mathbf{v}_i$ normalizados. Observamos que isso advém do fato de existir uma base $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n\}$ ortonormal de autovetores de $\mathbb{R}^n$, onde n é a ordem de $\mathbf{A}$.

Em geral, os autovetores não são conhecidos, e o acima exposto não pode ser utilizado para calcular a solução $\tilde{\mathbf{x}}$. Ainda, nas condições estabelecidas para λ_1, a primeira componente $(\mathbf{b}^t\mathbf{v}_1/\lambda_1)\mathbf{v}_1$ do vetor solução, domina a solução, embora, em muitos casos seja de interesse do problema calcular $\tilde{\mathbf{x}}_d$ apuradamente também. O MGC aplicado diretamente sobre o sistema original, geralmente fornece pobres aproximações de $\tilde{\mathbf{x}}$, por motivos já citados. No entanto, se $\mathbf{v}_1$ é conhecido (e portanto λ_1), podemos calcular a primeira componente de (3.3) e depois calcular separadamente a segunda componente $\tilde{\mathbf{x}}_d$ da seguinte forma: multiplicando a (3.3) à esquerda por $\mathbf{A}$, resulta

---

[2] O programa utilizado é o *m-file cg.m* criado, em 1993, pelos autores de [11].

$$\mathbf{A}\tilde{\mathbf{x}} = \frac{\mathbf{b}^t\mathbf{v}_1}{\lambda_1}\mathbf{A}\mathbf{v}_1 + \mathbf{A}\tilde{\mathbf{x}}_d;\ \text{ou}\ \ \mathbf{b} = \left(\mathbf{b}^t\mathbf{v}_1\right)\mathbf{v}_1 + \mathbf{A}\tilde{\mathbf{x}}_d.$$

Aí vemos que $\tilde{\mathbf{x}}_d$ é a solução do SELAS $\mathbf{Ax} = \mathbf{c}$, onde $\mathbf{c} := \mathbf{b} - (\mathbf{b}^t\mathbf{v}_1)\mathbf{v}_1$. Essa solução pode ser obtida pelo MGC tomando-se, para vetor inicializador, um vetor ortogonal ao vetor $\mathbf{v}_1$, uma vez que $\tilde{\mathbf{x}}_d$ pertence ao subespaço gerado por $\mathbf{v}_2, \mathbf{v}_3, \ldots, \mathbf{v}_n$.

Se $\mathbf{v}_1$ não é conhecido, podemos encontrar uma aproximação dele usando, por exemplo, o método de Lanczos, que será descrito na secção 3.8. Numa primeira instância, o método da potência inverso também pode ser considerado para uma estimativa, embora precária, de $\mathbf{v}_1$. Uma outra alternativa para resolver SELAS quase singulares é utilizar o método do SELAS aumentado, explanado na secção seguinte.

Convém observar que, se k autovalores de $\mathbf{A}$ estão muito próximos de zero, expandimos a solução $\tilde{\mathbf{x}}$ de $\mathbf{Ax} = \mathbf{b}$ assim

$$\tilde{\mathbf{x}} = \frac{\mathbf{b}^t\mathbf{v}_1}{\lambda_1}\mathbf{v}_1 + \frac{\mathbf{b}^t\mathbf{v}_2}{\lambda_2}\mathbf{v}_2 + \cdots + \frac{\mathbf{b}^t\mathbf{v}_k}{\lambda_k}\mathbf{v}_k + \tilde{\mathbf{x}}_d,$$

com

$$\tilde{\mathbf{x}}_d = \sum_{i=k+1}^{n} \frac{\mathbf{b}^t\mathbf{v}_i}{\lambda_i}\mathbf{v}_i.$$

E isso implica que os autovetores $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k$ de $\mathbf{A}$ devem ser conhecidos para poder determinar uma aproximação de $\tilde{\mathbf{x}}$.

3.5. O Método do SELAS Aumentado. Nosso objetivo principal nesta secção é o Teorema 3.7, e o que precede esse teorema serve para introduzir, de maneira simplificada, suas idéias mais importantes.

Seja $\mathbf{Ax} = \mathbf{b}$ um SELAS de ordem $n \times n$ com $\mathbf{A}$ s. p. d. (em verdade, interessa aqui o caso de $\mathbf{A}$ quase singular), e seja $\mathbf{v}_1$ um autovetor de $\mathbf{A}$. Suponhamos que possamos achar um vetor $\mathbf{v} \in \mathbb{R}^n$ não-ortogonal a $\mathbf{v}_1$ em relação ao produto interno usual, de modo que $\mathbf{Av}$ não tenha componentes negativas. Muitas vezes o vetor $\mathbf{v} := [1\ 1\ \ldots\ 1]^t$ preenche essas condições. Consideremos o SELAS *aumentado* de ordem $(n+1) \times (n+1)$,

$$\begin{bmatrix} \mathbf{A} & -\mathbf{Av} \\ -\mathbf{v}^t\mathbf{A} & \mathbf{v}^t\mathbf{Av} \end{bmatrix}\begin{bmatrix} \mathbf{x} \\ y \end{bmatrix} = \begin{bmatrix} \mathbf{b} \\ -\mathbf{v}^t\mathbf{b} \end{bmatrix}, \tag{3.4}$$

ou $\tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}}$. Para o vetor $\mathbf{u} := [\mathbf{v}^t\ 1]^t \neq \mathbf{0}$,

$$\tilde{\mathbf{A}}\mathbf{u} = \begin{bmatrix} \mathbf{A} & -\mathbf{Av} \\ -\mathbf{v}^t\mathbf{A} & \mathbf{v}^t\mathbf{Av} \end{bmatrix}\begin{bmatrix} \mathbf{v} \\ 1 \end{bmatrix} = \begin{bmatrix} \mathbf{Av} - \mathbf{Av} \\ -\mathbf{v}^t\mathbf{Av} + \mathbf{v}^t\mathbf{Av} \end{bmatrix} = \mathbf{0}.$$

Isso mostra que $\tilde{\mathbf{A}}$ é singular e que $\mathbf{u} \in \mathcal{N}(\tilde{\mathbf{A}})$ pois a dimensão de $\mathcal{N}(\tilde{\mathbf{A}})$ é, então, 1. Claro, o SELAS (3.4) é consistente, uma vez que $[\mathbf{x}_o^t\ \ 0]^t$ é, evidentemente, uma solução particular dele, se $\mathbf{x}_o$ é a solução de $\mathbf{Ax} = \mathbf{b}$, e isso se mantém mesmo que $\mathbf{A}$ não seja s. p. d., desde que $\mathbf{v} \notin \mathcal{N}(\mathbf{A})$. Resulta que o conjunto

solução de (3.4) é a reta do $\mathbb{R}^{n+1}$,

$$\left\{\widetilde{\mathbf{x}} \in \mathbb{R}^{n+1} \middle| \widetilde{\mathbf{x}} = t\begin{bmatrix}\mathbf{v}\\ 1\end{bmatrix} + \begin{bmatrix}\mathbf{x}_o\\ 0\end{bmatrix}, t \in \mathbb{R}\right\}.$$

Vemos, então, que resolver o SELAS $\mathbf{Ax} = \mathbf{b}$ é equivalente a resolver o SELAS (3.4). No entanto, como veremos no Teorema 3.7, este último é mais bem condicionado que o primeiro, o que justifica sua criação, com o objetivo de obter através dele a solução do SELAS original, pelo MGC, quando $\mathbf{A}$ é s. p. d.

Pelo que se viu acima, $\mathbf{u} = [\mathbf{v}^t\ 1]^t$ é um autovetor de $\widetilde{\mathbf{A}}$ correspondente ao autovalor $\widetilde{\lambda}_o := 0$. Então, se for possível tomar o autovetor $\mathbf{v}_1$ de $\mathbf{A}$, em lugar de $\mathbf{v}$ (para isso, basta que $\mathbf{Av}_1$ não tenha componentes negativas), cria-se um autovalor extra $\widetilde{\lambda}_o$ de $\widetilde{\mathbf{A}}$; o menor autovalor λ_1 de $\mathbf{A}$ (supostamente próximo de zero), fica substituído pelo autovalor $\widetilde{\lambda}_1 := \lambda_1\left(1 + \mathbf{v}_1^t\mathbf{v}_1\right)$ de $\widetilde{\mathbf{A}}$ associado ao autovetor $\widetilde{\mathbf{v}}_1 := \begin{bmatrix}\mathbf{v}_1\\ -\mathbf{v}_1^t\mathbf{v}_1\end{bmatrix}$; e os demais autovalores $\lambda_2, \ldots, \lambda_n$ de $\mathbf{A}$, associados, respectivamente, aos autovetores $\mathbf{v}_2, \ldots, \mathbf{v}_n$, são também autovalores de $\widetilde{\mathbf{A}}$, mas associados aos autovetores $\widetilde{\mathbf{v}}_2 := \begin{bmatrix}\mathbf{v}_2\\ 0\end{bmatrix}, \cdots, \widetilde{\mathbf{v}}_n := \begin{bmatrix}\mathbf{v}_n\\ 0\end{bmatrix}$. Provemos essas afirmações:

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{v}}_1 = \begin{bmatrix}\mathbf{A} & -\mathbf{Av}_1\\ -\mathbf{v}_1^t\mathbf{A} & \mathbf{v}_1^t\mathbf{Av}_1\end{bmatrix}\begin{bmatrix}\mathbf{v}_1\\ -\mathbf{v}_1^t\mathbf{v}_1\end{bmatrix} = \begin{bmatrix}\lambda_1\mathbf{v}_1 + \lambda_1\left(\mathbf{v}_1^t\mathbf{v}_1\right)\mathbf{v}_1\\ -\lambda_1\mathbf{v}_1^t\mathbf{v}_1 - \lambda_1\left(\mathbf{v}_1^t\mathbf{v}_1\right)^2\end{bmatrix} = \begin{bmatrix}\lambda_1\left(1 + \mathbf{v}_1^t\mathbf{v}_1\right)\mathbf{v}_1\\ \lambda_1\left(1 + \mathbf{v}_1^t\mathbf{v}_1\right)\left(-\mathbf{v}_1^t\mathbf{v}_1\right)\end{bmatrix}$$

$$= \lambda_1\left(1 + \mathbf{v}_1^t\mathbf{v}_1\right)\begin{bmatrix}\mathbf{v}_1\\ -\mathbf{v}_1^t\mathbf{v}_1\end{bmatrix} = \widetilde{\lambda}_1\widetilde{\mathbf{v}}_1.$$

Para $i = 2, \ldots, n$,

$$\widetilde{\mathbf{A}}\widetilde{\mathbf{v}}_i = \begin{bmatrix}\mathbf{A} & -\mathbf{Av}_1\\ -\mathbf{v}_1^t\mathbf{A} & \mathbf{v}_1^t\mathbf{Av}_1\end{bmatrix}\begin{bmatrix}\mathbf{v}_i\\ 0\end{bmatrix} = \begin{bmatrix}\mathbf{Av}_i\\ -\mathbf{v}_1^t\mathbf{Av}_i\end{bmatrix} = \begin{bmatrix}\lambda_i\mathbf{v}_i\\ -\lambda_i\mathbf{v}_1^t\mathbf{v}_i\end{bmatrix} = \begin{bmatrix}\lambda_i\mathbf{v}_i\\ 0\end{bmatrix} = \lambda_i\widetilde{\mathbf{v}}_i.$$

Para a penúltima igualdade, usamos o fato de que os $\mathbf{v}_i$ são ortogonais a $\mathbf{v}_1$, o que decorre de supor $\mathbf{A}$ s. p. d. (ou λ_1 menor que os demais autovalores de $\mathbf{A}$) e de poder escolher, por isso, n autovetores de $\mathbf{A}$ ortogonais entre si.

3.6. Exemplo. Esta ilustração é bastante esclarecedora e interessante. Consideremos um SELAS, cuja matriz dos coeficientes é a matriz tridiagonal s. p. d. de ordem $n \times n$,

$$\mathbf{A} := \begin{bmatrix}1+a & -1 & & & \\ -1 & 2+a & \ddots & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & 2+a & -1\\ & & & -1 & 1+a\end{bmatrix},$$

onde a é um número positivo próximo de zero, e a matriz circundada

$$\widetilde{\mathbf{A}}:=\begin{bmatrix} 1+a & -1 & & & 0 & -a \\ -1 & 2+a & \ddots & & 0 & -a \\ & \ddots & \ddots & \ddots & \vdots & \vdots \\ & & \ddots & 2+a & -1 & -a \\ 0 & 0 & \cdots & -1 & 1+a & -a \\ -a & -a & \cdots & -a & -a & na \end{bmatrix}.$$

Vemos que, escolhendo $\mathbf{v}:=[1\ 1\ \ldots\ 1]^t$, $\widetilde{\mathbf{A}}$ é construída como a matriz dos coeficientes em (3.4). Sejam $\lambda_1, \lambda_2, \ldots, \lambda_n$ os autovalores de $\mathbf{A}$ e $\widetilde{\lambda}_0 = 0, \widetilde{\lambda}_1, \ldots, \widetilde{\lambda}_n$ os de $\widetilde{\mathbf{A}}$. Claro $(a, \mathbf{v})$ é um autopar de $\mathbf{A}$. Podemos pôr $\lambda_1 := a$ e $\mathbf{v}_1 := \mathbf{v}$. O autovalor $\widetilde{\lambda}_1 = \left(1+\mathbf{v}_1^t\mathbf{v}_1\right)\lambda_1 = (n+1)a$. Para $i \geq 2$, temos $\lambda_i > a$ e $\widetilde{\lambda}_i = \lambda_i$. ∎

Para o caso em que $\mathbf{A}$ tem k autovetores próximos de zero, circundamos $\mathbf{A}$ com k colunas, e k linhas correspondentes. A perturbação dos autovalores ocorre como estabelecido no teorema seguinte.

3.7. Teorema. *Seja* $\mathbf{A}$ *uma matriz de ordem* $n \times n$ *e* $\mathbf{V}$ *uma matriz de ordem* $n \times m$ *com* $m < n$. *Consideremos a matriz aumentada de ordem* $(n + m) \times (n + m)$,

$$\widetilde{\mathbf{A}}:=\begin{bmatrix} \mathbf{A} & -\mathbf{AV} \\ -\mathbf{V}^t\mathbf{A} & \mathbf{V}^t\mathbf{AV} \end{bmatrix}.$$

(*a*) $\widetilde{\mathbf{A}}$ *tem, no mínimo,* m *autovalores nulos, e tem* n *autovalores* $\widetilde{\lambda}_i$ *iguais aos de* $(\mathbf{I}+\mathbf{VV}^t)\mathbf{A}$.

(*b*) *Se* $\mathbf{A}$ *é simétrica não-singular e* $\mathbf{V} = [\alpha_1\mathbf{v}_1\ \ldots\ \alpha_m\mathbf{v}_m]$, *onde os* $\mathbf{v}_i$ *são autovetores ortonormais de* $\mathbf{A}$, *correspondentes aos autovalores* λ_i, *então os autovalores não-nulos de* $\widetilde{\mathbf{A}}$ *são* $\widetilde{\lambda}_i = \left(1+\alpha_i^2\right)\lambda_i$, i = 1, 2, . . . , *m*; e $\widetilde{\lambda}_i = \lambda_i$, i = *m*+1, . . . , *n*.

Demonstração. (a) Ponhamos $\mathbf{T}:=\begin{bmatrix} \mathbf{I}_n & -\mathbf{V} \\ \mathbf{0} & \mathbf{I}_m \end{bmatrix}$. Então $\mathbf{T}$ é inversível e

$$\mathbf{T}^t\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{T} = \begin{bmatrix} \mathbf{A} & -\mathbf{AV} \\ -\mathbf{V}^t\mathbf{A} & \mathbf{V}^t\mathbf{AV} \end{bmatrix} = \widetilde{\mathbf{A}}.$$

Ainda,

$$\mathbf{T}\left(\mathbf{T}^t\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\mathbf{T}\right)\mathbf{T}^{-1} = \mathbf{T}\mathbf{T}^t\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = \begin{bmatrix} \left(\mathbf{I}_n+\mathbf{VV}^t\right)\mathbf{A} & \mathbf{0} \\ -\mathbf{V}^t\mathbf{A} & \mathbf{0} \end{bmatrix}.$$

Daí concluímos que a última matriz é semelhante a $\widetilde{\mathbf{A}}$ e que, portanto, sendo $\mathbf{I}$ a matriz identidade de ordem $n + m$,

$$\det\left(\widetilde{\lambda}\mathbf{I}-\widetilde{\mathbf{A}}\right) = \det\begin{bmatrix} \widetilde{\lambda}\mathbf{I}_n - \left(\mathbf{I}_n+\mathbf{VV}^t\right)\mathbf{A} & \mathbf{0} \\ \mathbf{V}^t\mathbf{A} & \widetilde{\lambda}\mathbf{I}_m \end{bmatrix}.$$

Conseqüentemente

$$\det\left(\widetilde{\lambda}\mathbf{I}-\widetilde{\mathbf{A}}\right)=0 \Leftrightarrow \det\left(\widetilde{\lambda}\mathbf{I}_m\right)\cdot\det\left(\widetilde{\lambda}\mathbf{I}_n-\left(\mathbf{I}_n+\mathbf{V}\mathbf{V}^t\right)\mathbf{A}\right)=0$$
$$\Leftrightarrow \widetilde{\lambda}^m=0 \text{ ou } \det\left(\widetilde{\lambda}\mathbf{I}_n-\left(\mathbf{I}_n+\mathbf{V}\mathbf{V}^t\right)\mathbf{A}\right)=0,$$

o que mostra que zero é um autovalor de $\widetilde{\mathbf{A}}$ com multiplicidade algébrica no mínimo igual a m e que os demais autovalores de $\widetilde{\mathbf{A}}$ são autovalores de $\left(\mathbf{I}_n+\mathbf{V}\mathbf{V}^t\right)\mathbf{A}$, como também, reciprocamente, que todos os autovalores de $\left(\mathbf{I}_n+\mathbf{V}\mathbf{V}^t\right)\mathbf{A}$, em número de n, são autovalores de $\widetilde{\mathbf{A}}$.

(b) Seja $(\lambda_i, \mathbf{v}_i)$ um autopar de $\mathbf{A}$ para $i = 1, 2, \ldots, n$ e seja

$$\mathbf{v}_i := \begin{bmatrix} v_{1i} & v_{2i} & \cdots & v_{ni} \end{bmatrix}^t,\ i = 1, 2, \ldots, m.$$

Calculemos

$$\mathbf{V}^t\mathbf{v}_i=\begin{bmatrix} \alpha_1 v_{11} & \alpha_1 v_{21} & \cdots & \alpha_1 v_{n1} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_i v_{1i} & \alpha_i v_{2i} & \cdots & \alpha_i v_{ni} \\ \cdots & \cdots & \cdots & \cdots \\ \alpha_m v_{1m} & \alpha_m v_{2m} & \cdots & \alpha_m v_{nm} \end{bmatrix}\begin{bmatrix} v_{1i} \\ v_{2i} \\ \vdots \\ \vdots \\ v_{ni} \end{bmatrix}=\begin{bmatrix} \alpha_1\langle v_1, v_i\rangle \\ \vdots \\ \alpha_i\langle v_i, v_i\rangle \\ \vdots \\ \alpha_m\langle v_m, v_i\rangle \end{bmatrix}=\begin{bmatrix} 0 \\ \vdots \\ \alpha_i \\ \vdots \\ 0 \end{bmatrix},$$

e

$$\mathbf{V}\mathbf{V}^t\mathbf{v}_i=\begin{bmatrix} \alpha_1 v_{11} & \cdots & \alpha_i v_{1i} & \cdots & \alpha_m v_{1m} \\ \alpha_1 v_{21} & \cdots & \alpha_i v_{2i} & \cdots & \alpha_m v_{2m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \alpha_1 v_{n1} & \cdots & \alpha_i v_{ni} & \cdots & \alpha_m v_{nm} \end{bmatrix}\begin{bmatrix} 0 \\ \vdots \\ \alpha_i \\ \vdots \\ 0 \end{bmatrix}=\alpha_i^2\mathbf{v}_i.$$

Temos, então,

$$\left(\mathbf{I}+\mathbf{V}\mathbf{V}^t\right)\mathbf{A}\mathbf{v}_i=\lambda_i\left(\mathbf{I}+\mathbf{V}\mathbf{V}^t\right)\mathbf{v}_i=\begin{cases}\lambda_i\left(1+\alpha_i^2\right)\mathbf{v}_i, & \text{se } i=1, 2, \cdots, m \\ \lambda_i\mathbf{v}_i, & \text{se } i = m+1, \cdots, n.\end{cases}$$

Definindo agora

$$\widetilde{\lambda}_i := \begin{cases}\lambda_i\left(1+\alpha_i^2\right), & \text{se } i=1, 2, \cdots, m \\ \lambda_i, & \text{se } i = m+1, \cdots, n,\end{cases}$$

vemos que $\widetilde{\lambda}_i$ é autovalor de $\left(\mathbf{I}+\mathbf{V}\mathbf{V}^t\right)\mathbf{A}$ e, portanto, por (a), $\widetilde{\lambda}_i$ é autovalor de $\widetilde{\mathbf{A}}$ nas condições do enunciado da parte (b). ∎

Se **A** é não-singular, então $\left(\mathbf{I}+\mathbf{V}\mathbf{V}^{t}\right)\mathbf{A}$ é não-singular. Logo, pela parte (a) do Teorema 3.7, o espaço nulo de $\widetilde{\mathbf{A}}$ torna-se conhecido, pois é o autoespaço, com dimensão m, do autovalor zero. E, se **A** é s. p. d., então $\widetilde{\mathbf{A}}$ será positiva semidefinida (singular) e podemos aplicar o MGC na forma explicada na secção 3.2. Além disso, a fim de maximizar a velocidade de convergência, usamos a parte (b) do Teorema 3.7 da seguinte maneira: escolhemos os menores autovalores λ_i de **A**, $i = 1, 2, \ldots, m$, e, para correspondentes autovalores de $\widetilde{\mathbf{A}}$, escolhemos $\widetilde{\lambda}_i = \max\{\lambda_1, \lambda_2, \ldots, \lambda_m\}$, $i=1, 2, \ldots, m$; para conseguir isso, basta tomar $\alpha_i = \sqrt{\widetilde{\lambda}_i / \lambda_i - 1}$.

3.8. O Método de Lanczos para Gerar Vetores A-ortogonais. O MGC, desenvolvido no capítulo 2, gera uma seqüência $(\mathbf{d}_k)$ **A**-ortogonal de vetores de direção, isto é, vetores que satisfazem

$$\left\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_j \right\rangle = 0, \text{ para todo } j \neq k,$$

quando **A** for autoadjunta em relação ao produto interno $\langle \bullet, \bullet \rangle$ definido por uma matriz s. p. d. **W.** Aqui será apresentado outro caminho, que decorre do método de Lanczos [88], [89], mostrando de que forma esse método pode ser utilizado para estimar os autovalores extremos de uma matriz **A**. Lembremos que um de nossos objetivos neste capítulo é resolver um SELAS **Ax = b**, quando **A** é s. p. d. e quase singular, da forma indicada na secção 3.4, para o que é necessário conhecer a auto-solução $(\lambda_1, \mathbf{v}_1)$ de **A**, quando $\lambda_1 \approx 0$ e λ_1 é o menor dos autovalores.

Consideremos, então, **A** autoadjunta em relação a um produto interno $\left\langle \bullet, \bullet \right\rangle$ como o citado acima. Dado $\mathbf{d}_0 \neq \mathbf{0}$, para $k = 0, 1, \ldots$, seja

$$\mathbf{d}_{k+1} = \mathbf{A}\mathbf{d}_k - r_k \mathbf{d}_k - s_{k-1}\mathbf{d}_{k-1}, \tag{3.5}$$

onde $s_{-1} := 0$ e

$$r_k = \frac{\left\langle \mathbf{A}\mathbf{d}_k, \mathbf{A}\mathbf{d}_k \right\rangle}{\left\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \right\rangle}, \quad s_{k-1} = \frac{\left\langle \mathbf{A}\mathbf{d}_k, \mathbf{A}\mathbf{d}_{k-1} \right\rangle}{\left\langle \mathbf{d}_{k-1}, \mathbf{A}\mathbf{d}_{k-1} \right\rangle}.$$

O lema que segue mostra que o método de Lanczos, definido pela fórmula de recorrência (3.5), gera uma seqüência **A**-ortogonal $(\mathbf{d}_j)$ de vetores $\mathbf{d}_j$, e também que a expressão para s_{k-1} pode ser simplificada.

3.8.1. Lema.

(*a*) *Para todo* $k \neq j$, *temos* $\left\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_j \right\rangle = 0$.

(*b*) $s_{k-1} = \dfrac{\left\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_k \right\rangle}{\left\langle \mathbf{d}_{k-1}, \mathbf{A}\mathbf{d}_{k-1} \right\rangle}, k \geq 1.$

Demonstração. (a) Basta mostrar que, fixado arbitrariamente k, $\left\langle \mathbf{d}_k, \mathbf{A}\mathbf{d}_j \right\rangle = 0$ para todo $j < k$, o que será feito por indução. Para $k = 1$,

$$\left\langle \mathbf{d}_1, \mathbf{A}\mathbf{d}_0 \right\rangle = \left\langle \mathbf{A}\mathbf{d}_0 - r_0\mathbf{d}_0, \mathbf{A}\mathbf{d}_0 \right\rangle = \left\langle \mathbf{A}\mathbf{d}_0, \mathbf{A}\mathbf{d}_0 \right\rangle - \frac{\left\langle \mathbf{A}\mathbf{d}_0, \mathbf{A}\mathbf{d}_0 \right\rangle}{\left\langle \mathbf{d}_0, \mathbf{A}\mathbf{d}_0 \right\rangle}\left\langle \mathbf{d}_0, \mathbf{A}\mathbf{d}_0 \right\rangle = 0.$$

Suponhamos que, fixado k, tenhamos $\langle \mathbf{d}_k, \mathbf{Ad}_j \rangle = 0$ para todo $j < k$ e mostremos que $\langle \mathbf{d}_{k+1}, \mathbf{Ad}_j \rangle = 0$ para todo $j \leq k$. Para $j = k$ temos:

$$\begin{aligned}\langle \mathbf{d}_{k+1}, \mathbf{Ad}_k \rangle &= \langle \mathbf{Ad}_k, \mathbf{Ad}_k \rangle - r_k \langle \mathbf{d}_k, \mathbf{Ad}_k \rangle - s_{k-1} \langle \mathbf{d}_{k-1}, \mathbf{Ad}_k \rangle \\ &= \langle \mathbf{Ad}_k, \mathbf{Ad}_k \rangle - \frac{\langle \mathbf{Ad}_k, \mathbf{Ad}_k \rangle}{\langle \mathbf{d}_k, \mathbf{Ad}_k \rangle} \langle \mathbf{d}_k, \mathbf{Ad}_k \rangle - s_{k-1} \langle \mathbf{d}_{k-1}, \mathbf{Ad}_k \rangle \\ &= -s_{k-1} \langle \mathbf{d}_{k-1}, \mathbf{Ad}_k \rangle \\ &= -s_{k-1} \langle \mathbf{d}_k, \mathbf{Ad}_{k-1} \rangle = 0.\end{aligned}$$

Também para $j = k$ -1,

$$\begin{aligned}\langle \mathbf{d}_{k+1}, \mathbf{Ad}_{k-1} \rangle &= \langle \mathbf{Ad}_k, \mathbf{Ad}_{k-1} \rangle - r_k \langle \mathbf{d}_k, \mathbf{Ad}_{k-1} \rangle - s_{k-1} \langle \mathbf{d}_{k-1}, \mathbf{Ad}_{k-1} \rangle \\ &= \langle \mathbf{Ad}_k, \mathbf{Ad}_{k-1} \rangle - \frac{\langle \mathbf{Ad}_k, \mathbf{Ad}_{k-1} \rangle}{\langle \mathbf{d}_{k-1}, \mathbf{Ad}_{k-1} \rangle} \langle \mathbf{d}_{k-1}, \mathbf{Ad}_{k-1} \rangle = 0.\end{aligned}$$

Finalmente, para $j \leq k$-2,

$$\begin{aligned}\langle \mathbf{d}_{k+1}, \mathbf{Ad}_j \rangle &= \langle \mathbf{Ad}_k, \mathbf{Ad}_j \rangle - r_k \langle \mathbf{d}_k, \mathbf{Ad}_j \rangle - s_{k-1} \langle \mathbf{d}_{k-1}, \mathbf{Ad}_j \rangle \\ &= \langle \mathbf{Ad}_k, \mathbf{d}_{j+1} + r_j \mathbf{d}_j + s_{j-1} \mathbf{d}_{j-1} \rangle \\ &= \langle \mathbf{Ad}_k, \mathbf{d}_{j+1} \rangle + r_j \langle \mathbf{Ad}_k, \mathbf{d}_j \rangle + s_{j-1} \langle \mathbf{Ad}_k, \mathbf{d}_{j-1} \rangle \\ &= \langle \mathbf{d}_k, \mathbf{Ad}_{j+1} \rangle + r_j \langle \mathbf{d}_k, \mathbf{Ad}_j \rangle + s_{j-1} \langle \mathbf{d}_k, \mathbf{Ad}_{j-1} \rangle = 0,\end{aligned}$$

e a prova de (a) está completa.

(b) De (3.5) e da **A**-ortogonalidade dos vetores $\mathbf{d}_j$, provada em (a), vem:

$$\begin{aligned}\langle \mathbf{Ad}_k, \mathbf{Ad}_{k-1} \rangle &= \langle \mathbf{Ad}_k, \mathbf{d}_k + r_{k-1} \mathbf{d}_{k-1} + s_{k-2} \mathbf{d}_{k-2} \rangle \\ &= \langle \mathbf{Ad}_k, \mathbf{d}_k \rangle + r_{k-1} \langle \mathbf{d}_k, \mathbf{Ad}_{k-1} \rangle + s_{k-2} \langle \mathbf{d}_k, \mathbf{Ad}_{k-2} \rangle \\ &= \langle \mathbf{Ad}_k, \mathbf{d}_k \rangle,\end{aligned}$$

o que simplifica a expressão de s_{k-1}. ∎

3.8.2. Versão Precondicionada. Se precondicionamos o SELAS $\mathbf{Ax} = \mathbf{b}$, onde **A** é simétrica, na forma $\mathbf{B} := \mathbf{C}^{-1}\mathbf{A}$, com **C** s. p. d., então podemos escolher o produto interno, representado por **C**,

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t \mathbf{C} \mathbf{y},$$

obtendo

$$\langle \mathbf{x}, \mathbf{By} \rangle = \mathbf{x}^t \mathbf{Ay} = \mathbf{x}^t \mathbf{AC}^{-1} \mathbf{Cy} = \left(\mathbf{C}^{-1} \mathbf{Ax} \right)^t \mathbf{Cy} = \langle \mathbf{Bx}, \mathbf{y} \rangle,$$

isto é, **B** é autoadjunta em relação ao produto interno definido por **C**. Assim sendo, a recursão (3.5) toma a forma:

$$\mathbf{d}_{k+1} = \mathbf{C}^{-1}\mathbf{A}\mathbf{d}_k - r_k\mathbf{d}_k - s_{k-1}\mathbf{d}_{k-1}, \tag{3.6}$$

onde

$$r_k = \frac{\left(\mathbf{C}^{-1}\mathbf{A}\mathbf{d}_k\right)^t \mathbf{A}\mathbf{d}_k}{\mathbf{d}_k^t \mathbf{A}\mathbf{d}_k} \quad \text{e} \quad s_{k-1} = \frac{\mathbf{d}_k^t \mathbf{A}\mathbf{d}_k}{\mathbf{d}_{k-1}^t \mathbf{A}\mathbf{d}_{k-1}}.$$

Observamos que, nesse caso, os vetores $\mathbf{d}_j$ são **A**-ortogonais em relação ao produto interno usual, pois

$$\mathbf{d}_j^t \mathbf{A}\mathbf{d}_i = \left\langle \mathbf{d}_j, \mathbf{B}\mathbf{d}_i \right\rangle = 0, \text{ se } i \neq j.$$

3.8.3. Versão C-ortogonal. Se tomarmos, como acima, $\mathbf{B} = \mathbf{C}^{-1}\mathbf{A}$, sendo aqui **A** s. p. d. e **C** simétrica e não singular, então podemos fazer a escolha do produto interno, dado por

$$\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^t \mathbf{C}\mathbf{A}^{-1}\mathbf{C}\mathbf{y},$$

que fica bem definido, pois, usando a decomposição de Choleski $\mathbf{A} = \mathbf{L}\mathbf{L}^t$ de **A**, vemos que $\mathbf{C}\mathbf{A}^{-1}\mathbf{C} = \mathbf{C}(\mathbf{L}\,\mathbf{L}^t)^{-1}\mathbf{C} = (\mathbf{L}^{-1}\mathbf{C})^t(\mathbf{L}^{-1}\mathbf{C})$ é s. p. d. Nesse caso, então,

$$\langle \mathbf{x}, \mathbf{B}\mathbf{y} \rangle = \mathbf{x}^t \left(\mathbf{C}\mathbf{A}^{-1}\mathbf{C}\right)\mathbf{C}^{-1}\mathbf{A}\mathbf{y} = \mathbf{x}^t \mathbf{C}\mathbf{y} = \left(\mathbf{C}^{-1}\mathbf{A}\mathbf{x}\right)^t \mathbf{C}\mathbf{A}^{-1}\mathbf{C}\mathbf{y} = \langle \mathbf{B}\mathbf{x}, \mathbf{y} \rangle,$$

ou seja, **B** é autoadjunta em relação a esse produto interno. A forma da recursão é a (3.6), mas os coeficientes aqui são calculados pelas fórmulas:

$$r_k := \frac{\mathbf{d}_k^t \mathbf{A}\mathbf{d}_k}{\mathbf{d}_k^t \mathbf{C}\mathbf{d}_k}, \quad s_{k-1} := \frac{\mathbf{d}_k^t \mathbf{C}\mathbf{d}_k}{\mathbf{d}_{k-1}^t \mathbf{C}\mathbf{d}_{k-1}}.$$

Nessa versão, os vetores $\mathbf{d}_j$ formam um conjunto **C**-ortogonal em relação ao produto interno usual, pois

$$\mathbf{d}_j^t \mathbf{C}\mathbf{d}_i = \left\langle \mathbf{d}_j, \mathbf{B}\mathbf{d}_i \right\rangle = 0, \text{ se } i \neq j.$$

Observações.

- Na versão **C**-ortogonal podemos tomar $\mathbf{C} = \mathbf{I}$, a matriz identidade, em cujo caso os vetores $\mathbf{d}_j$ formam um conjunto ortogonal em relação ao produto interno usual.

- Quando **A** e **C** são ambas s. p. d., o comportamento das duas versões, a **A**-ortogonal e a **C**-ortogonal, é semelhante.

- Se **A** é indefinida (não-definida), então o cálculo de $\mathbf{d}_j$ na versão **A**-ortogonal pode ser interrompido, porque o denominador da expressão que determina os coeficientes r_k e s_{k-1} pode se anular.

- Se **C** é indefinida, então o cálculo de $\mathbf{d}_j$ na versão **C**-ortogonal pode ser interrompido, pelo mesmo motivo já citado.

- Se **C** é s. p. d. e **A** é simétrica, não necessariamente definida ou semidefinida, então a versão **C**-ortogonal (em particular, a versão de vetores de direção ortogonais, quando **C** = **I**) é aplicável. O método não é interrompido por causa de uma divisão por zero, mas emperra; em outras palavras, não gera novos vetores de direção, quando $\mathbf{Ad}_j = \mathbf{0}$, para algum j. Contudo, como foi sugerido por Faddev e Faddeeva [45], supondo que isso ocorra, tomamos um novo vetor **y** não-nulo e subtraímos dele sua projeção **C**-ortogonal sobre o subespaço gerado por $\{\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_j\}$; o vetor resultante

$$\mathbf{d}_{10} := \mathbf{y} - \sum_{i=o}^{j} c_i \mathbf{d}_i, \text{ com } c_i := \frac{\mathbf{y}^t \mathbf{C} \mathbf{d}_i}{\mathbf{d}_i^t \mathbf{C} \mathbf{d}_i},$$

é **C**-ortogonal aos vetores $\mathbf{d}_i$, para $i = 0, 1, \ldots, j$. O vetor $\mathbf{d}_{10}$ reinicializa o processo de cálculo dos vetores de direção, e os que seguem, $\mathbf{d}_{1p}$, $p = 1, 2, \ldots$, continuam sendo calculados com o método de Lanczos na versão **C**-ortogonal. Nesse caso, a seqüência de vetores de direção é

$$(\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_j, \mathbf{d}_{10}, \mathbf{d}_{11}, \ldots) \tag{3.7}$$

O Lema 3.8.3.1 prova que os vetores $\mathbf{d}_{1p}$, $p = 0, 1, \ldots$, estão no complemento **C**-ortogonal de $\{\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_j\}$, precisamente como desejamos. Se $\mathbf{Ad}_{1p} = 0$, para algum $p \geq 1$, então o procedimento acima é repetido. Dessa forma, o universo $\mathbb{R}^n$, onde ocorre o processo, é decomposto na soma direta de subespaços **C**-ortogonais, cada um deles gerado por um conjunto de vetores, originado cada vez que o método é inicializado.

3.8.3.1. Lema. *A seqüência (3.7) é* **C***-ortogonal.*

Demonstração. Pelo Lema 3.8.1 (a), a subseqüência $(\mathbf{d}_{10}, \mathbf{d}_{11}, \ldots)$ da seqüência (3.7) é **C**-ortogonal. Usaremos a indução para provar que cada vetor dessa seqüência é **C**-ortogonal ao conjunto $\{\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_j\}$.

Por construção, $\mathbf{d}_{10}$ é **C**-ortogonal a $\{\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_j\}$. Suponhamos que, fixado arbitrariamente p, tenhamos para todo $0 \leq k \leq p$,

$$\mathbf{d}_{1k}^t \mathbf{C} \mathbf{d}_i = 0, \; i = 1, 2, \ldots, j.$$

Utilizando a (3.6), essas igualdades se escrevem

$$\mathbf{d}_{1k}^t \mathbf{C}\left(\mathbf{C}^{-1}\mathbf{A}\mathbf{d}_{i-1} - r_{i-1}\mathbf{d}_{i-1} - s_{i-2}\mathbf{d}_{i-2}\right) = 0, \; i = 1, 2, \ldots, j,$$

ou

$$\mathbf{d}_{1k}^t \mathbf{A} \mathbf{d}_{i-1} = r_{i-1}\mathbf{d}_{1k}^t \mathbf{C} \mathbf{d}_{i-1} + s_{i-2}\mathbf{d}_{1k}^t \mathbf{C} \mathbf{d}_{i-2}, \; i = 1, 2, \ldots, j. \tag{3.8}$$

Também, pela ocorrência $\mathbf{Ad}_j = \mathbf{0}$,

$$\mathbf{d}_{1k}^t \mathbf{A} \mathbf{d}_j = 0. \tag{3.9}$$

Então ($k = p + 1$)

$$\begin{aligned}\mathbf{d}_{1p+1}^t\mathbf{C}\mathbf{d}_i &= \left(\mathbf{C}^{-1}\mathbf{A}\mathbf{d}_{1p} - r_{1p}\mathbf{d}_{1p} - s_{1p-1}\mathbf{d}_{1p-1}\right)^t\mathbf{C}\mathbf{d}_i \\ &= \mathbf{d}_{1p}^t\mathbf{A}\mathbf{d}_i - r_{1p}\mathbf{d}_{1p}^t\mathbf{C}\mathbf{d}_i - s_{1p-1}\mathbf{d}_{1p-1}^t\mathbf{C}\mathbf{d}_i \\ &= \mathbf{d}_{1p}^t\mathbf{A}\mathbf{d}_i = \begin{cases} 0, \text{ se } i = j, \text{ por (3.9)} \\ r_i\mathbf{d}_{1p}^t\mathbf{C}\mathbf{d}_i - s_{i-1}\mathbf{d}_{1p}^t\mathbf{C}\mathbf{d}_{i-1} = 0, \text{ se } i < j, \text{ por (3.8).} \end{cases}\end{aligned}$$

Para obter a penúltima igualdade antes da chave, usamos a hipótese da indução. ▌

3.8.4. Resolução de SELAS com Vetores A-ortogonais. Consideremos a versão **A**-ortogonal do método de Lanczos definido em (3.5). Se o conjunto de vetores $\{\mathbf{d}_o, \mathbf{d}_1, \ldots, \mathbf{d}_{n-1}\}$, gerados por (3.5), é linearmente independente, então a solução **x** do SELAS **Ax** = **b** é dada por

$$\mathbf{x} = \sum_{j=0}^{n-1}\alpha_j\mathbf{d}_j,$$

para certos α_j, onde *n* é a ordem de **A**. Usando a **A**-ortogonalidade dos vetores $\mathbf{d}_j$ e o fato de que $\langle \mathbf{d}_j, \mathbf{Ax} - \mathbf{b}\rangle = 0$, temos:

$$\begin{aligned}\left\langle \mathbf{d}_j, \mathbf{Ax} - \mathbf{b}\right\rangle &= \left\langle \mathbf{d}_j, \mathbf{A}\left(\alpha_o\mathbf{d}_o + \alpha_1\mathbf{d}_1 + \cdots + \alpha_{n-1}\mathbf{d}_{n-1}\right) - \mathbf{b}\right\rangle \\ 0 &= \alpha_o\left\langle \mathbf{d}_j, \mathbf{Ad}_o\right\rangle + \alpha_1\left\langle \mathbf{d}_j, \mathbf{Ad}_1\right\rangle + \cdots + \alpha_j\left\langle \mathbf{d}_j, \mathbf{Ad}_j\right\rangle + \cdots + \alpha_{n-1}\left\langle \mathbf{d}_j, \mathbf{Ad}_{n-1}\right\rangle - \left\langle \mathbf{d}_j, \mathbf{b}\right\rangle \\ 0 &= \alpha_j\left\langle \mathbf{d}_j, \mathbf{Ad}_j\right\rangle - \left\langle \mathbf{d}_j, \mathbf{b}\right\rangle,\end{aligned}$$

e, portanto,

$$\alpha_j = \frac{\left\langle \mathbf{d}_j, \mathbf{b}\right\rangle}{\left\langle \mathbf{d}_j, \mathbf{Ad}_j\right\rangle}.$$

3.8.4.1. Exemplo. Tomamos a versão **A**-ortogonal precondicionada (subsecção 3.8.2) para resolver o SELAS **Ax** = **b**, com

$$\mathbf{A} = \begin{bmatrix} 1 & -2 & 0 & -1 \\ -2 & 9 & 2 & 6 \\ 0 & 2 & 2 & 0 \\ -1 & 6 & 0 & 7 \end{bmatrix}, \text{ e } \mathbf{b} = \begin{bmatrix} -2 \\ 15 \\ 4 \\ 12 \end{bmatrix}.$$

O precondicionamento e o produto interno são dados pela matriz, adrede criada com o MATLAB,

$$\mathbf{C} = \begin{bmatrix} 5 & -1 & 3 & 2 \\ -1 & 3 & -1 & 2 \\ 3 & -1 & 4 & 0 \\ 2 & 2 & 0 & 8 \end{bmatrix}.$$

O SELAS a ser resolvido é $\mathbf{C}^{-1}\mathbf{Ax} = \mathbf{C}^{-1}\mathbf{b}$, onde $\mathbf{C}^{-1}\mathbf{b} = [-11/6\ 31/6\ 11/3\ 2/3]^t$, e, para o vetor que inicializa o método, escolhemos arbitrariamente, $\mathbf{d}_o := [0\ 1\ -1\ 0]^t$. Assim,

$$\mathbf{d}_o^t\mathbf{A}\mathbf{d}_o = 7,\quad \mathbf{C}^{-1}\mathbf{A}\mathbf{d}_o = [-5/6\ 13/6\ 7/6\ 5/12]^t,\quad r_o = 58/21;$$

$\mathbf{d}_1 = [-5/6\ -25/42\ 55/14\ 5/12]^t$;

$$\mathbf{d}_1^t\mathbf{A}\mathbf{d}_1 = 10972/491,\quad \mathbf{C}^{-1}\mathbf{A}\mathbf{d}_1 = [-295/252\ 415/126\ 485/144\ -1025/2016]^t,\quad r_1 = 2791/1405,$$

$s_o = 11221/3515$;

$\mathbf{d}_2 = [684/1411\ 2063/1607\ -929/747\ -3343/2502]^t$;

$$\mathbf{d}_2^t\mathbf{A}\mathbf{d}_2 = 1267/508,\quad \mathbf{C}^{-1}\mathbf{A}\mathbf{d}_2 = [-250/8109\ 621/2287\ 267/2404\ -2401/7341]^t,\quad r_2 = 1103/3658,$$

$s_1 = 471/4220$;

$\mathbf{d}_3 = [\ -383/4560\ -14/285\ 217/4560\ \ 401/13680\ ]^t$;

$$\mathbf{d}_3^t\mathbf{A}\mathbf{d}_3 = 30/27109,\ \mathbf{C}^{-1}\mathbf{A}\mathbf{d}_3 = [-25/4104\ -8/2565\ 31/10260\ 11/6840]^t,\ r_3 = 107/1425,$$

$s_2 = 59/132971$;

E, fazendo o produto interno por $\mathbf{C}\mathbf{d}_j$, com $j = 0, 1, 2, 3$, de

$$\mathbf{C}^{-1}\mathbf{A}\mathbf{x} = \sum_{j=0}^{3}\alpha_j\mathbf{C}^{-1}\mathbf{A}\mathbf{d}_j = \mathbf{C}^{-1}\mathbf{b},$$

obtemos os valores $\alpha_o = 11/7$, $\alpha_1 = 1567/2603$, $\alpha_2 = -1641/1504$ e $\alpha_3 = -2401/7341$. Assim a solução é $\mathbf{x} = \sum_{j=0}^{3}\alpha_j\mathbf{d}_j = [1\ 1\ 1\ 1]^t$. ∎

Observações.

- Quando n é grande, pode ocorrer uma perda significativa da **A**-ortogonalidade dos vetores $\mathbf{d}_j$, por causa dos erros de arredondamento. Se tentarmos achar uma aproximação para a solução **x** usando apenas alguns desses vetores, essa aproximação não será boa, a menos que o número de condição da matriz **A** seja pequeno e $\mathbf{d}_o$ esteja relacionado com o sistema, como por exemplo, $\mathbf{d}_o = \mathbf{A}\mathbf{x}_o - \mathbf{b} = \mathbf{r}_o$, onde $\mathbf{x}_o$ é uma aproximação inicial qualquer [08].

- Para uso posterior, observemos que, no Exemplo 3.8.4.1, a matriz $\mathbf{H} := \text{tridiag}(1, r_i, s_i)$,

$$\mathbf{H} = \begin{bmatrix} 58/21 & 11221/3515 & 0 & 0 \\ 1 & 2791/1405 & 471/4220 & 0 \\ 0 & 1 & 1103/3658 & 59/132971 \\ 0 & 0 & 1 & 107/1425 \end{bmatrix},$$

e $\mathbf{C}^{-1}\mathbf{A}$ têm o mesmo polinômio característico,

$$\det(\mathbf{H}-\lambda\mathbf{I}) = \lambda^4 - \frac{41}{8}\lambda^3 + \frac{575}{144}\lambda^2 - \frac{47}{72}\lambda^2 + \frac{1}{36},$$

e, como consequência, possuem os mesmos autovalores.

3.8.5. Cálculo de Auto-soluções com Vetores A-ortogonais. Outra aplicação importante da criação de vetores **A**-ortogonais de Lanczos, gerados pela fórmula de recursão (3.5), é o cálculo dos autovalores de **A**. Lembremos que a **A**-ortogonalidade de vetores não-nulos implica sua independência linear. Dessa forma, a matriz **Q**, de ordem $n \times n$, cujas colunas são os vetores $\mathbf{d}_i$, $i = 0, 1, \ldots, n-1$, isto é,

$$\mathbf{Q} := [\mathbf{d}_0\ \mathbf{d}_1 \cdots \mathbf{d}_{n-1}]$$

é não singular. Além disso, a recursão (3.5) pode ser escrita, matricialmente,

$$\mathbf{A}[\mathbf{d}_0\ \mathbf{d}_1 \cdots \mathbf{d}_{n-1}] = [\mathbf{d}_0\ \mathbf{d}_1 \cdots \mathbf{d}_{n-1}] \begin{bmatrix} r_0 & s_0 & 0 & \cdots & 0 \\ 1 & r_1 & s_1 & & 0 \\ 0 & 1 & r_2 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & s_{k-2} \\ 0 & 0 & \cdots & 1 & r_{k-1} \end{bmatrix},$$

ou, mais compactamente,

$$\mathbf{AQ} = \mathbf{QH},$$

em que $\mathbf{H} := \text{tridiag}(1, r_i, s_i)$. Assim, $\mathbf{H} = \mathbf{Q}^{-1}\mathbf{AQ}$ e **A** são semelhantes, o que nos permite afirmar que possuem os mesmos autovalores (cp. Exemplo 3.8.4.1). Dizemos que a *matriz **Q** reduz a matriz **A** a uma forma tridiagonal*.

Podem ser usadas também transformações de semelhança de Householder para reduzir **A** a uma forma tridiagonal. No entanto, se **A** é grande e esparsa, o que, de fato, estamos supondo no desenvolvimento deste trabalho, não é aconselhável tal abordagem, pois essas transformações tendem a destruir a esparsidade e resultam matrizes grandes densas, com a impossibilidade de serem manipuladas, durante o processo.

3.8.6. Forma Normalizada da Versão Precondicionada. Lembremos primeiramente que uma matriz **P** é ortogonal se e somente se $\mathbf{P}^t = \mathbf{P}^{-1}$, ou, equivalentemente, se satisfaz a condição $\mathbf{P}^t\mathbf{P} = \mathbf{I}$, onde **I** é a matriz identidade de mesma ordem que **P**.

Considerando a fórmula de recursão (3.6), que corresponde à versão precondicionada de (3.5), se os vetores $\mathbf{d}_j$ forem normalizados em relação ao produto interno definido pela matriz **C**, então a matriz **Q**, definida na subsecção 3.8.5, torna-se **C**-ortogonal, isto é, $\mathbf{Q}^t\mathbf{CQ} = \mathbf{I}$. Se a normalização ocorre em relação ao produto interno definido pela matriz **A**, a matriz **Q** torna-se **A**-ortogonal. O desenvolvimento que segue utiliza a **C**-normalização dos vetores $\mathbf{d}_j$, porque as fórmulas que advêm dessa escolha são mais simples que as dadas pela **A**-normalização dos citados vetores.

Partindo da fórmula de recursão (3.6), e **C**-normalizando os vetores $\mathbf{d}_j$, obtemos:

$$\tilde{s}_k\tilde{\mathbf{d}}_{k+1} := \mathbf{d}_{k+1} = \mathbf{C}^{-1}\mathbf{A}\tilde{\mathbf{d}}_k - \tilde{r}_k\tilde{\mathbf{d}}_k - \tilde{s}_{k-1}\tilde{\mathbf{d}}_{k-1},$$

onde

$$\tilde{\mathbf{d}}_j = \frac{\mathbf{d}_j}{\sqrt{\mathbf{d}_j^t \mathbf{C} \mathbf{d}_j}},\ \tilde{r}_k = \tilde{\mathbf{d}}_k^t \mathbf{A} \tilde{\mathbf{d}}_k \text{ e } \tilde{s}_j = \tilde{\mathbf{d}}_{j+1}^t \mathbf{C} \tilde{\mathbf{d}}_{j+1}.$$

Os vetores $\tilde{\mathbf{d}}_j$ agora satisfazem $\tilde{\mathbf{d}}_j^t \mathbf{C} \tilde{\mathbf{d}}_i = \delta_{ij}$, e a matriz $\mathbf{H} := \text{tridiag}(\tilde{s}_{i-1}, \tilde{r}_i, \tilde{s}_i)$ torna-se simétrica.

O cálculo dos autovalores λ_j de uma tridiagonal matriz **H** pode ser feito através de vários algoritmos, como, por exemplo, o algoritmo da bissecção [139]. Mais recentemente, algumas variações de algoritmos de redução cíclica têm-se tornado populares [8]. De qualquer forma, tendo a característica da esparsidade, o cálculo dos autovalores (e autovetores) da matriz tridiagonal obtida de **A** (ou $\mathbf{C}^{-1}\mathbf{A}$) por Lanczos é, em geral, mais simples que o cálculo dos mesmos pela matriz original.

Como já sabemos, os autovalores de **A** podem ser determinados calculando os autovalores de **H**. Na prática, entretanto, os elementos de **H** são perturbados por erros de arredondamento e os seus autovalores fornecem apenas aproximações dos autovalores de **A**. O que torna o método de Lanczos para o cálculo de auto-soluções particularmente interessante é que podemos, geralmente, obter ótimas estimativas de alguns dos autovalores de **A** - em particular, dos autovalores extremais - através da submatriz $\mathbf{H}_k$ da matriz **H**, obtida no $k^{\underline{ésimo}}$ passo, a saber,

$$\mathbf{H}_k = \begin{bmatrix} r_0 & s_0 & 0 & \cdots & 0 \\ 1 & r_1 & s_1 & \cdots & 0 \\ 0 & 1 & r_2 & & \vdots \\ \vdots & \vdots & & \ddots & s_{k-1} \\ 0 & 0 & \cdots & 1 & r_k \end{bmatrix},$$

para k consideravelmente menor do que n-1.

Com esse objetivo enunciaremos o teorema a seguir, que indica uma maneira de estimar o menor autovalor de **A**, conhecendo o menor autovalor de $\mathbf{H}_k$. A precisão do resultado depende do número de condição de **A** e de quão separados estão os dois menores autovalores distintos. Explicando melhor, se a matriz **A** é bem condicionada (em caso contrário, usamos precondicionamento), e se os dois menores autovalores distintos de **A** estão suficientemente distanciados um do outro, então a estimativa do menor autovalor de **A** será muito boa. O teorema contempla o caso em que o menor autovalor de **A** tem multiplicidade algébrica r-1.

3.8.7. Teorema. (Lanczos, Kaniel, Paige) *Seja* **A** *uma matriz simétrica com autovalores* $\lambda_1 = \lambda_2 = \ldots = \lambda_{r-1} < \lambda_r \le \lambda_{r+1} \le \ldots \le \lambda_n$, *para* $r \ge 2$, *e seja* $\mathcal{Z}_1$ *o subespaço gerado pelos autovetores correspondentes aos autovalores* $\lambda_1, \lambda_2, \ldots, \lambda_{r-1}$. *Além disso, seja* μ_1 *o menor autovalor de* $\mathbf{H}_k$, *para algum* $k \ge 1$. *Então, temos a seguinte estimativa:*

$$0 \le \mu_1 - \lambda_1 \le (\lambda_n - \lambda_1) \left[\frac{\tan \phi_1}{\mathcal{T}_k\left(\dfrac{\kappa_r \lambda + 1 - 2\lambda_1/\lambda_r}{\kappa_r - 1} \right)} \right]^2, \tag{3.10}$$

onde $\kappa_r := \lambda_n / \lambda_r$, ϕ_1 *é o ângulo entre o vetor* $\mathbf{d}_0$ *e* $\mathcal{Z}_1$, *isto é,*

$$\cos\phi_1 = \max_{\mathbf{z}_1 \in \mathcal{Z}_1} \frac{|\langle \mathbf{d}_0, \mathbf{z}_1 \rangle|}{\left(\langle \mathbf{d}_0, \mathbf{d}_0 \rangle \langle \mathbf{z}_1, \mathbf{z}_1 \rangle \right)^{1/2}},$$

e $\mathcal{T}_k$ é o polinômio de Chebyshev de grau k.

Demonstração. A demonstração desse teorema pode ser encontrada em [81], [88] e [102].

Observação. De maneira semelhante podemos encontrar uma estimativa para o maior autovalor μ_{k+1} de $\mathbf{H}_k$ [8].

De (3.10) obtemos:

$$0 \le \mu_1 \le \lambda_1 + (\lambda_n - \lambda_1)\left[\frac{\tan\phi_1}{\mathcal{T}_k\left(\frac{\kappa_r\lambda + 1 - 2\lambda_1/\lambda_r}{\kappa_r - 1}\right)}\right]^2.$$

Dessa forma, se a matriz **A** é bem condicionada (em caso contrário usamos um precondicionador de boa precisão), a estimativa será bem apurada, mesmo que o valor de k seja escolhido bem pequeno. De fato, se o número de condição κ de **A** não depender da ordem n de **A**, e se, além disso, a distância entre λ_r e λ_1 é uniforme em n, isto é, $\lambda_r - \lambda_1 \le \delta\lambda_1$, onde $\delta > 0$ não depende de n, e $\tan\phi_1 \le \tau$, onde τ é uma constante que igualmente não depende de n, então podemos escolher k independente de n também. Por exemplo, suponhamos que queiramos calcular μ_1 com uma exatidão relativa ε tal que

$$0 \le \mu_1 - \lambda_1 \le \varepsilon\lambda_1.$$

Então, observando que $\kappa_r \le \kappa$, o Teorema 3.8.7 mostra que é suficiente escolher k de modo que

$$\frac{\mu_1 - \lambda_1}{\lambda_1} \le (\kappa - 1)\left[\frac{\tau}{\mathcal{T}_k\left(\frac{\kappa + \lambda_r/\lambda_1 - 2}{\kappa - 1}\right)}\right]^2 \le \varepsilon.$$

Manipulando a segunda desigualdade acima, concluímos que precisa encontrar o menor valor de k para o qual

$$\mathcal{T}_k\left(\frac{\kappa + \lambda_r/\lambda_1 - 2}{\kappa - 1}\right) \ge \tau\left(\frac{\kappa - 1}{\varepsilon}\right)^{1/2},$$

onde $\tau := \tan\phi_1$. Mas um k nessa condição é menor que o menor k tal que

$$\mathcal{T}_k\left(1 + \frac{\delta}{\kappa - 1}\right) \ge \tau\left(\frac{\kappa - 1}{\varepsilon}\right)^{1/2},$$

e esse k não cresce com n porque supusemos que δ, ρ e τ não dependem de n.

Vemos também que, para $k \le n$-1, a seguinte igualdade matricial é verdadeira:

$$\mathbf{AQ}_k = \mathbf{Q}_k\mathbf{H}_k + [\mathbf{0}\ \mathbf{0}\ \ldots\ \mathbf{0}\ \mathbf{d}_{k+1}], \tag{3.11}$$

para $\mathbf{Q}_k := [\mathbf{d}_0\ \mathbf{d}_1\ \ldots\ \mathbf{d}_k]$ de ordem $n \times (k+1)$.

O cálculo dos autovalores μ_j de uma matriz tridiagonal $\mathbf{H}_k$ pode ser feito, como já dito anteriormente, através de vários algoritmos. Denotando com $\mathbf{w}_j$ os autovetores de $\mathbf{H}_k$, correspondentes aos autovalores μ_j, podemos determinar os autovetores $\mathbf{v}_j$ de $\mathbf{A}$ por

$$\mathbf{v}_j = \mathbf{Q}_k\mathbf{w}_j,$$

porque, se $\mathbf{H}_k\mathbf{w}_j = \mu_j\mathbf{w}_j$, então (3.11) mostra que

$$\mathbf{A}\mathbf{v}_j = \mathbf{A}\mathbf{Q}_k\mathbf{w}_j = \mathbf{Q}_k\mathbf{H}_k\mathbf{w}_j + w_{jk+1}\mathbf{d}_{k+1} = \mu_k\mathbf{v}_j + w_{jk+1}\mathbf{d}_{k+1},$$

onde w_{jk+1} denota a $(k+1)^{\underline{\acute{e}sima}}$ componente do vetor $\mathbf{w}_j$.

Conclusão

Como vimos nos capítulos 1 e 2, as duas formas de recorrência do MGC apresentadas, a de três termos e a de dois termos, são válidas quando a matriz **A** do SELAS $\mathbf{Ax} = \mathbf{b}$ é a. p. d. em relação ao produto interno usado no algoritmo de cada forma. Esse fato, que generaliza a construção dos algoritmos do método, embora teoricamente correto, não é diretamente utilizado na prática, porque requer que encontremos uma matriz s. p. d. **W** tal que **WA** seja s. p. d. e $\mathbf{WA} = \mathbf{A}^t\mathbf{W}$, para definir o produto interno, o que é muito difícil, mesmo que a ordem n de **A** seja pequena. A matriz **A** dos coeficientes de grande parte dos SELAS oriundos de problemas de aplicação é s. p. d., e, nesse caso, que é uma particularização do caso geral que apresentamos, tomando o produto interno usual, os algoritmos do método podem ser utilizados para a busca da solução. Mas, e aqui está uma primeira utilidade da generalidade de nossa exposição, sendo **A** s. p. d., podemo-nos valer do produto interno definido pela própria **A** ou por sua inversa. Particularizando ainda mais, o método fornece excelentes resultados quando **A** é esparsa e o SELAS é bem condicionado, como no Exemplo 2.10. Nem cogitamos na aplicação do método quando **A** é grande e não-esparsa, devido ao enorme número de operações envolvidas no processo. A questão do mau condicionamento do sistema pode ser contornada com o uso de uma matriz precondicionadora **C**, s. p. d., que transforma o sistema original num equivalente no sentido de terem ambos a mesma solução, e, mais importante, que define o produto interno nos algoritmos do método. O fato é que pode existir uma estreita relação entre a matriz precondicionadora e o produto interno definido para resolver o SELAS precondicionado, ou seja, a matriz que precondiciona pode definir também o produto interno, e este é um segundo ponto que empresta importância à generalidade em relação ao produto interno do MGC, conforme o desenvolvimento deste trabalho.

O algoritmo (1.21) da forma de recorrência de três termos do MGC envolve teoricamente, como vimos na secção 2.9, um número maior de operações que o algoritmo (2.21) da forma de recorrência de dois termos. Para compararmos o custo computacional das duas formas, devemos implementar os algoritmos numa mesma seqüência lógica, em uma linguagem escolhida, e executar os mesmos numa única máquina. Com esse objetivo e seguindo esses critérios, implementamos um algoritmo para cada uma das formas, na linguagem do MATLAB, que aplicamos no Exemplo 2.10. O resultado foi o esperado, ou seja, o número de operações geradas com a forma de recorrência de três termos foi maior que o número da forma curta.

Não é por acaso que os programas, encontrados na literatura para resolver SELAS pelo MGC, são implementações do algoritmo da forma de recorrência de dois termos. Na verdade esses programas supõem a matriz do SELAS s. p. d. e usam precondicionamento. Com isso o custo computacional é menor e é contornado o problema do mau condicionamento do sistema. Quando este é bem condicionado, escolhemos como matriz precondicionadora a matriz identidade e o produto interno no algoritmo do método é o usual.

Queremos ainda acrescentar que o que apresentamos sobre o MGC justifica o fato dele ser hoje tão utilizado pela comunidade científica. Acreditamos que a popularidade dele deve-se a muitos fatores, dentre os quais citamos: tem uma propriedade de otimização relevante, o que geralmente implica convergência em muito menos iterações que as correspondentes à propriedade da terminação finita; a rapidez da convergência pode ser bastante melhorada com o emprego de técnicas de precondicionamento; o método é livre de parâmetros (isso não ocorre, por exemplo, no método SOR), o que o torna simples de ser aplicado; a forma de recorrência de dois termos torna o tempo de execução e os

requisitos de armazenagem aceitáveis; o método pode ser aplicado a SELAS não simétricos, singulares e quase singulares. Nesses últimos, depois que certas transformações são aplicadas ao sistema.

Referências Bibliográficas

[01] Altman, M., *On the Convergence of the Conjugate Gradient method for Non-Bounded Linear Operators in Hilbert Space*. In Approximation Methods in Funcional Analysis, Lecture Notes, 1959, California Institute of Technology, pp. 33-36.

[02] Antosiewicz, H., Rheinboldt, C., *Numerical Analysis and Funtional Analysis*. In Survey of Numerical Analysis, ed. John Todd, McGraw-Hill, 1962, N. Y., pp. 485-517 (Ch. 14).

[03] Arnoldi, W., *The Principle of Minimized Iterations in the Solution the Matrix Eingenvalue Problem*. Quart. Of Appl. Math. 9, 1951, pp. 17-29.

[04] Axelsson, O., *A Generalized SSOR Method*. BIT 12, 1972, pp. 443-467.

[05] Axelsson, O., *A Class of Iterative Methods for Finite Element Equations*. Computer methods in Applied Mechanics and Engineering 9, 1976, pp. 123-137.

[06] Axelsson, O., Barker, A., *Finite Element Solution of Boundary Value Problems*. Theory and Computation, Academic Press, Orlando, Fl., 1984.

[07] Axelsson, O., Vassilevski, P., *A Block Box Generalized Conjugate Gradient Solver with Inner Iterations and Variable-Step Preconditioning*. SIAM J. Matrix Anal. Appl. 12, 1991, pp. 625-644.

[08] Axelsson, O., *Iterative Solutions Methods*. Cambridge University Press, N. Y., 1994.

[09] Bank, R., Chan, T., *Na Analysis of the Composite Step Bi-Conjugate Gradient Method*. Numerisch Mathematik 66, 1993, pp. 295-319.

[10] Bank, R., Chan, T., *A Composite Step Bi-Conjugate Gradient Algorithm for Nonsymmetric Linear Systems*. Numer. Alg., 1994, pp. 1-16.

[11] Barret, R., Berry, M., Chan, T., Demmel, J., Donato, J., Dongarra, J., Eijkhout, V., Pozo, R., Romine, C., Van der Vorst, H., *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, 1994.

[12] Bartels, R., Daniel, J., *A Conjugate Gradient Approach to Nonlinear Elliptic Boundary Value Problems in Irregular Regions*. In Conference on the Numerical Solution of Differential Equations, Dundee, 1973, ed, G. ª Watson, Springer Verlog, 1974, New York.

[13] Bjorch, A., Elfving, T., *Accelerated Projection Methods for Computing Pseudo-Inverse Solutions of System of Linear Equations*. BIT19, 1979, pp. 145-163.

[14] Bothner-By, A., Naar-Colin, C., *The Proton Magnetic Resonance Spectra of 2,3 - Disubstituted n-Butones*. J. of the ACS84, 1962, pp. 743-747.

[15] Branley, R., Sameh, A., *Row Projection Methods for Large Nonsymmetric Linear Systems*. SIAM J. Sci. Sctist. Comp. 13, 1992, pp. 168-193.

[16] Brezinski, C., Sadak, H., *Avoiding Breakdown in the CGS Algorithm*. Num. Alg. 1, 1991, pp. 199-206.

[17] Brezinski, C., Zaglia, M., Sadak, H., *Avoiding Breakdown and Near Breakdown in Lanczos Type Algorithms*. Num. Alg. 1, 1991, pp. 261-284.

[18] Broyden, C., *A Class of Methods for Solving Non Linear Simultaneous Equations*. Math. Of Comp. 19, 1965, pp. 577-593.

[19] Cai, X., Widlund, O., *Multiplicative Schwarz Algorithms for some Nonsymmetric and Indefinite Problems*. SIAM J. Numer. Anal. 30, 1993, pp. 936-952.

[20] Chan, T., Gallopoulos, E., Simoncini, V., Szeto, T., Tong, C., *A Quasi-Minimal Residual Variant of the Bi-CGSTAB Algorithm for Nonsymmetric Systems*. SIAM J. Sci. Comp. 15, 1994, pp. 338-347.

[21] Chan, T., Mathew, T., Shao, J., *Efficient Variants of the Vertex Domain Decomposition Algorithm*. SIAM J. Sci. Comp. 15, 1994, pp. 1349-1374.

[22] Chandra, R., Einsenstot, S., Schultz, M., *Conjugate Gradient Methods for Partial Defferential Equations*. In Advances in Computer methods for Partial Differential Equations, ed. R. Vichnevetsky, AICA, Rutgers University, 1975, New Brunswick, New Jersey, pp. 60-64.

[23] Chang, F., Wing, O., *Multilayer RC Distributed Networks*. IEEE Trans. on Circuit Theory CT-17, 1970, pp. 32-40.

[24] Cline, A., Golub, G., Platzman, G., *Calculation of Normal Modes of Oceans Using A Lanczos Method*. In Sparse matrix Computation, ed. James R. Bunch and Donald J. Rose, Academic Press, 1976, N. Y., pp. 409-426.

[25] Concus, P., Golub, G., *A Generalized Conjugate Gradient Method for Non-Symmetric Systems of Linear Equations*. In Computing Methods in Applied Sciences and Engineering, 1976, ed. R. Glowinski and J. L. Lions, Springer-Verlag, N. Y., pp. 56-65.

[26] Concus, P., Golub, G., O'Leary, D., *A Generalized Conjugate Gradient Method for the Numerical Solution of Elliptic Partial Differential Equations*. In Sparse Matrix Computation, ed. James R. Bunch and Donald J. Rose, Academic Press, 1976, N. Y., pp. 309-332.

[27] Cullum, J., Donath, W., *A Block Generalization of the Simmetric S-Step Lanczos Algoritm*. IBM T. J. Watson Research Center Report RC4845, 1974a, Yorktown Heigths, N. Y.

[28] Cullum, J., Donath, W., *A Block Lanczos Algorithm for Computing the q Algebraically Largest Eingenvalues and a Corresponding Eingenspace for Large, Sparse Symmetric Matrices*. In Proc. 1974 IEEE Conference on Decision and Control, 1974b, IEEE Press, N. Y., pp. 505-509.

[29] Cunha, C., *Métodos Numéricos para as Engenharias e Ciências Aplicadas*. Editora da Unicamp, São paulo, 1993.

[30] Curtiss, J., *A Generalization of the Method of Conjugate Gradients for Solving Systems of Linear Algebraic Equations*. Math. Tables and Aids to Comp. 8, 1954, pp. 189-193.

[31] Daniel, J., *The Conjugate Gradient Methods for the Solutions of a Class of Nonlinear Operator Equations*. SIAM J. Numer. Anal. 4, 1967[a], pp. 10-26.

[32] Datta, B., *Numerical Linear Algebra and Applications*. Brooks/Cole Publishing Company, USA, 1995.

[33] De, S., Davies, A., *Convergence of Adaptive Equaliser for Data Transmission*. Eletronics Letters 6, 1970, pp. 858-861.

[34] Demmel, J., Hesth, M., Van der Vorst, H., *Parallel Numerical Linear Algebra*. In Acta Numerico, Vol. 2, Cambridge Press, New York, 1993.

[35] Dodson, E., Isaacs, N., Rollett, J., *A Method of Fitting Satisfactory Models to Sets of Atomic Positions in Protein Structure Refinaments*. Acta Cryst. A32, 1976, pp. 311-315.

[36] Dongarra, J., Duff, I., Sorensen, D., Van der Vorst, H., *Solving Linear Systems on Vector and Shared Memory Computers*. SIAM, Philadelphia, PA, 1991.

[37] Dongarra, J., Van der Vorst, H., *Performancew of Varioures Computers using Standards Sparse Linear Equations Solving Techniques*. In Computer Benchmarks, eds. J. Dongarra and W. Gentzsch, Elsevier Scince Publishers B. V., N. Y., 1993, pp. 177-188.

[38] Douglas, Jr., Dupont, J. and T., *Preconditioned Conjugate Gradient Iteration Applied to Galerkin Methods for a Mildly-Nonlinear Dirichlet Problem*. In Sparse Matrix Computations, ed. James R. Bunch and Donald J. Rose, Academic Press, 1976, N. Y., pp. 333-348.

[39] Elman, H., *Iterative Methods for Large, Sparse, Nonsymmetric Systems of Linear Equations*. Ph. D. Thesis, Yale Universyti, New Haven, CT, 1982.

[40] Emília, D., Bodvarsson, G., *More on the Direct Interpretation of Magnetic Anomalies*. Earth and Planetary Science Letters 8, 1970, pp. 320-321.

[41] Engeli, M., Ginsburg, T., Rutishauser, H., Stiefel, E., *Refeined Iterative Methods for Computation of the Solution and the Eingenvalue of SelfAdjoint Boundary Value Problems*. Birkhauser Verlag, 1959, Basel/Stuttgart.

[42] Eu, B., *Method of Moments in Collision Theory*. J. Chem. Phys. 48, 1968, pp. 5611-5622.

[43] Evans, D., *The Use of Pre-conditioning in Iterative Methods for Solving Linear Equations with Symmetric Positive Definite Matrices*. J. Inst. Maths. Applics. 4, 1968, pp. 295-314.

[44] Evans, D., *The Analysis and Application of Sparse Matrix Algorithms in the Finite Elements Method*. In the Mathematics of Finite Elements and Applications, ed. J. R. Whiterman, Academic Press, 1973, N. Y., pp. 427-447.

[45] Faddeev, D., Faddeeva, V., *Computational Methods of Linear Algebra*, W. H. Freeman and Co., San Francisco, California, 1963.

[46] Fletcher, R., *Conjugate Gradient Methods for Indefinite Systems*. In Numerical Analysis Dundee, ed. G. Watson, Berlin, 1975, Spring-Verlag, New York, 1976, pp. 73-89.

[47] Fletcher, R. Powell, M., *A Rapidly Convergent Descent Method for Minimization*. Computer j. 6, 1963, pp. 163-168.

[48] Fletcher, R., Reeves, C., *Functions Minimization by Conjugate Gradients*. Computer J. 7, 1964, pp. 149-154.

[49] Forsyte, G., Hestenes, M., Rosser, J., *Iterative Methods for Solving Linear Equations*. Bull. Amer. Math. Soc. 57, 1951, p. 480.

[50] Fox, L., Huskey, H. D., Wilkinson, J. H., *Notes on the Solution of Algebraic Linear Simultaneous Equations*. Quart. J. of Mech. and Appl. Math. 1, 1948, pp. 149-173.

[51] Frank, W., *Solution of Linear Systems by Richardson's Methods*. J. Assoc. Comp. Mach. 7, 1960, pp. 274-286.

[52] Freund, R., *A Transpose-Free Quasi-Minimum Residual Algorithm for Non-Hermitian Linear Systems*. SIAM J. Sci. Comp. 14, 1993, pp. 470-482.

[53] Freund, R., Gutknecht, M., Nachtigal, N., *Na Implementation of the Look-Ahead Lanczos Algorithm for Non-Hermitian Matrices*. SIAM J. Sci. Comp. 14, 1993, pp. 137-158.

[54] Freund, R., Nachtigal, N., *QMR: A Quasi-Minimal Residual Method for Non-Hermitian Linear Systems*. Num. Math. 60, 1991, pp. 315-339.

[55] Freund, R., Nachtigal, N., *Na Implementation of the QMR Method based on coupled Two-Term Recurrences*. SIAM, J. Sci. Statist. Comp. 15, 1994, pp. 313-337.

[56] Freund, R., Szeto, T., *A Quasi-Minimal Residual Squared Algorithm for Non-Hermitian Linear Systems*. Tech. Rep. CAM Report 92-19, UCLA Dept. of Math., 1992.

[57] Garibotti, C., Villani, M., *Continuation in the Coupling Constant for the Total K and T Matrices*. Il Nuovo Cimento 59, 1969, pp. 107-123.

[58] Golub, G., O'Leary, D., *Some History of the Conjugate Gradient and Lanczos Algorithms: 1948-1976*. SIAM Review 31, No. 1, 1989, pp. 50-102.

[59] Gutknecht, M., *The Unsummetric Lanczos Algorithmos and their Relations to Paide Approximation, Continued Fractions and the QD Algorithm*. In Proceeding of the Copper Mountain Conference on Iterative Methods, 1990.

[60] Gutknecht, M., *A Completed Theory of the Unsymmetric Lanczos Process and Related Algorithms, part I*. SIAM J. Matrix Anal. Appl. 13, 1992, pp. 594-639.

[61] Gutknecht, M., *Variants of Bi-CGSTAB for Matrices with Complex Spectrum*. SIAM J. Sci. Comp. 14, 1993, pp. 1020-1033.

[62] Gutknecht, M., *A Completed Theory of the Unsymmetric Lanczos Process and Related Algorithms, part II*. SIAM J. Matrix Anal. Appl. 15, 1994, pp. 15-58.

[63] Harms, E., *A Modified Method of Moments Approach to the Solution of Scattering Equations*. Nuclear Physics A222, 1974, pp. 125-139.

[64] Hausman, Jr., Bloom, S., Bender, C., *A New Technique for Describing the Electronic States of Atoms and Molecules – The Vector Method*. Chem. Phys. Letters 32, 1975, pp. 483-488.

[65] Haydock, R., Heine, V., Kelley, M., *Eletronic Structure Based on the Local Atomic Environment for Tight-Binding Bands: II*. J. Phys. C: Solid State Physics 8, 1975, pp. 2591-2605.

[66] Hestenes, M., *Iterative Methods for Solving Linear Equations*. NAML Report 52-9, National Bureau of Standards, 1951, Los Angeles, California.

[67] Hestenes, M., *Iterative Computational Methods*. Comunications on Pure and Applied Mathematics 8, 1955, pp. 85-96.

[68] Hestenes, M., *Multiplier and Gradient Methods*. J. of Optimizations Theory and Apllications 4, 1969, pp.303-320.

[69] Hestenes, M., *Conjugate Direction Methods in Optimization*. Berlin: Springer-Verlag, 1980.

[70] Hestenes, M., Karush, W., *Solutions of* $\boldsymbol{Ax} = \lambda \boldsymbol{Bx}$. J. Res. Nat. Bur. Standards 49, 1951 (b), pp. 471-478.

[71] Hestenes, M., Stein, M., *The Solution of Linear Equations by Minimization*. NAML Report 52-45, December 12, 1951, National Bureau of Standards, Los Angeles, California.

[72] Hestenes, M., Stiefel, E., *Methods of Conjugate Gradients for Solving Linear Systems*. J. Res, Nat. Bur. Standards 49, 1952, pp. 409-436.

[73] Hill, D., *Experiments in Computational Matrix Algebra*. Random House, Inc. New York, 1988.

[74] Horwitz, L., Sarachik, P., *Davidon's Method in Hilbert Space*. SIAM J. Appl. Math. 16, 1968, pp. 676-695.

[75] Humes, A., Melo, I., Yoshida, L., Martiyns, W., *Noções de Cálculo Numérico*. Ed. McGraw-Hill, São Paulo, 1994.

[76] Ibarra, R., Vallieres, M., Feng, D., *Extended Basis Shell-Model Study of Two-Neutron Transfer Reactions*. Nuclear Physisc A241, 1975, pp. 386-406.

[77] Johnsons, O., Micchelli, C., Paul, G., *Polynomial Preconditioning for Conjugate Gradient Method*. SIAM J. Numer. Anal. 20, 1983, pp. 362-376.

[78] Joubert, W., *Lanczos Methods for the Solution of Nonsymmetric Systems of Linear Equations*. SIAM J. Matrix Anal. Appl. 13, 1992, pp. 926-943.

[79] Kahan, W., Parlett, B., *How far Should you go with the Lanczos Process?*. In Sparse Matrix Computation, 1976, ed. James R. Bunch and Donald J. Rose, Academic Press, N. Y., pp. 131-144.

[80] Kamoshida, M., Kani, K., Sato, K., Okada, T., *Hest Transfer Analysis of Beam-Lead Trasistor Chip*. IEEE Trans. on Electron. Devices ED-17, 1970, pp. 863-870.

[81] Kaniel, S., *Estimates for Some Computational Techniques in Linear Algebra*. Math. of Comp. 95, 1966, pp. 369-378.

[82] Kaplan, T., Gray, L., *Elemantary Excitations in Random Substitutional Allays*. Physical Review B14, 1976, pp. 3462-3470.

[83] Kawamura, K., Volz, R., *On the Convergence of the Conjugate Gradient Method in Hilbert Space*. IEEE Trans. on Auto. Control AC-14, 1969, pp. 296-297.

[84] Kelley, C., *Iterative Methods for Linear e Nonlinear Equations*. SIAM, Philadelphia, 1995.

[85] Kelley, C., Suresh, R., *GMRES and Integral Operators*. SIAM J. Sci. Comput. 17, 1996, pp. 217-226.

[86] Kobayashi, H., *Iterative Synthesis Methods for a Sismic Array Processor*. IEEE Trans. on Geoscience Eletronics GE-8, 1970, pp. 169-178.

[87] Kratochvil, A., *La Méthode des Gradients Conjugents pour les Equations Non Linéaires dos L'Espace de Banach*. Commentationes Mathematical Universitalis Carolinae 9, 1968, pp. 659-676.

[88] Lanczos, C., *An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators*. J. Res. Nat. Bur. Standards 45,1950, pp.255-282.

[89] Lanczos, C., *Solution of Systems of Linear Equations by Minimized Iterations*. J. Res. Nat. Bur. Standards 49, 1952, pp. 33-53.

[90] Lanczos, C., *Iterative Solution of Large-Scale Linear Systems*. J. Soc. Industr. Appl. Math. 6, 1958, pp. 91-109.

[91] Livesley, R., *The Analysis of Large Structural Systems*. Computer J. 3, 1960, pp. 34-39.

[92] Luenberger, D., *Hyberbolic Pair in the Method of Conjugate Gradients*. SIAM J. Appl. Math. 17, 1969, pp. 1263-1267.

[93] Marshall, Jr., Thomas, G., *Synthesis of RCL Ladder Networks by Matrix Tridisgonalization*. IEEE Trans. Circuit Theory CT-16, 1969, pp. 39-46.

[94] Mathews, J., *Numerical Methods for Mathematics, Science and Engineering*. Prentice-Hall, Inc., A Simon & Schuster Company, Englewood Clips, new jersey 07632, USA, 1992.

[95] Meier-Yang, U., *Preconditioned Conjugate Gradient-Like Methods for Nonsymmetric Linear Systems*. Tech. Rep., CSRD, University of Illinois, Urbana, IL, April 1992.

[96] Meijerink, J., Van der Vorst, H., *Na Iterative Solution Method for Linear Systems of which the Coeficient Matrix is a Symmetric M-matrix*. Mathematics of Computation 31, 1977, pp. 148-162.

[97] Miele, A., Huang, H., Heideman, J., *Sequential Gradient-Restoration Algorithm for the Minimization of Constrained Functions .. Ordinary and Conjugate Gradient Versions*. J. of Optimization Theory and Applications 4, 1969, pp. 213-243.

[98] Nachtingal, N., Reddy, S., Trefethen, L., *How fast are Nonsymmetric Matrix Iterations?*. SIAM J. Matrix Anal. Appls. 13, 1992, pp. 778-795.

[99] Nachtingal, N., Reichel, L., Trefethen, L., *A Hybrid GMRES Algorithm for Nonsymmetric Matrix Iterations*. SIAM J. Sci. Statist. Comp. 13, 1992, pp. 796-825.

[100] Nashed, M., *On General Iterative Methods for the Solutions of a Class of Nonlinear Operator Equations*. Math. of Comp. 19, 1965, pp. 14-24.

[101] Pagurek, B., Woodside, C., *The Conjugate Gradient Method for Optimal Control Problems with Bounded Variables*. Automatica 4, 1968, pp. 337-349.

[102] Paige, C., *Computational Variantes of the Lanczos Method for the Eingenproblem*. J. Inst. Maths. Applics. 10, 1972, pp. 373-381.

[103] Paige, C., *Error Analysis of the Lanczos Algorithm for Tridiagonalizing a Symmetric Matrix*. J. Inst. Maths. Applics. 18, 1976, pp. 341-349.

[104] Paige, C., Saunders, M., *Solution of Sparse Indefinide Systems of Linear Equations*. SIAM J. Numer. Anal. 12, 1975, pp. 617-629.

[105] Paige, C., Saunders, M., *LSQR: An Algorithm for Sparse Linear Equations and Sparse Least Squares*. ACM Trans. Math. Soft. 8, 1982, pp. 43-71.

[106] Paige, C., Parlett, B., Van der Vorst, H., *Approximate Solutions and Eigenvalue Bounds from Krylov Subspaces*. Numer. Lin. Alg. Appls. 29, 1995, pp. 115-134.

[107] Parlett, B., Taylor, D., Liu, Z., *A Look-Ahead Lanczos Algorithm for Unsymmetric Matrices*. Mathematics of Computation 44, 1985, pp. 105-124.

[108] Pitha, J., Norman, R., *Na Evaluation of Mathematical Functions to fit Infrared Band Envelopes*. Canadian J. of Chemistry 45, 1967, pp. 2347-2352.

[109] Platzman, G., *Normal Modes of the Atlantic and Indian Oceans*. J. of Physical Oceanography 5, 1975, pp. 201-221.

[110] Polak, E., Ribiere, G., *Note Sur la Convergence de Methods de Directions Conjugees*. R. Francaise d'Informatique et de Recherche Operationnele 3, 1969, pp. 35-43.

[111] Polyak, B., *The Conjugate Gradient Method in Extremal Problems*. USSR Comp. Math. and Math. Phys. 9, No. 4, 1969, pp. 94-112.

[112] Powell, M., *An Iterative Method for Finding Stationary Values of a Function of Several Variables*. Computer J. 5, 1962, pp. 147-151.

[113] Powers, W., *A Crude-Search Davidon-Type Technique with Applications to Shuttle Optimization*. J. Spacecraft 10, 1973, pp. 710-715.

[114] Reid, J., *On the Method of Conjugate Gradients for the Solution of Large Sparse Systems of Linear Equation*. In Large Sparse Sets of Linear Equations, Academic Press, 1971, N. Y., pp. 231-254.

[115] Rosser, J., Lanczos, C., Hestenes, M., Karush, W., *Separation of Close Eingenvalues of a Real Symmetric Matrix*. J. Res. Nat. Bur. Standards 47, 1951, pp. 291-297.

[116] Saad, Y., *A Flexible Inner-Outer Preconditioned GMRES Algorithm*. SIAM J. Sci. Comp. 14, 1993, pp. 461-469.

[117] Saad, Y., *Practical use of Polynomial Preconditioning for the Conjugate Gradient Method*. SIAM J. Sci. Statist. Comp. 6, 1985, pp. 865-881.

[118] Saad, Y., Schultz, M., *GMRES a Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems*. SIAM J. Sci. Statist. Comp. 7, 1986, pp. 856-869.

[119] Sebe, T., Nachamkin, J., *Varational Buildup of Nuclear Shell Model Bases*. Annals of Physics 51, 1969, pp. 100-123.

[120] Shah, B., Buehler, R., Kempthorne, O., *Some Algorithms for Minimizing a Function of Several Variables*. J. Soc. Industr. Appl. Math. 12, 1964, pp. 74-92.

[121] Shewchuk, J., *An Introduction to the Conjugate Gradient Method Without the Agonizing Pain*. School of computer Science Carnegie Mellon University Pittsburgh, PA 15213, 1994.

[122] Sinnott, Jr., J. F., Luenberger, D., *Solution of Optimal Control Problems by the Method of Conjugate Gradients*. Joint Automatic Control Conferenca, Preprints of Papers, Lewis Winner, 1967, New York, pp. 566-574.

[123] Sleijpen, G., Fokkema, D., *Bi-CGSTAB(l) for Linear Equations involving Unsymmetric Matrices with Complex Spectrum*. Elec. Trans. Numer. Anal. 1, 1993, pp. 11-32.

[124] Sonneveld, P., *CGS, A Fast Lanczos-Type solver for Nonsymmetric Linear Systems*. SIAM j. Sci. Statisi. Comp. 10, 1989, pp. 36-52.

[125] Stein, M., *Gradient Methods in the Solution of Systems of Linear Equations*. J. Res. Nat. Bur. Standards 48, 1952, pp. 407-413.

[126] Steiefel, E., *Über einige Methoden der Relaxationsrechnung*. Zeitschrift fur angewandte Mathematik und Physik 3, 1952, pp. 1-33.

[127] Stiefel, E., *Recent Developments in Relaxation Techniques*. In Proceedings of the Sixth Symposium in Applied Mathematics, 1953, Mc Graw-Hill, New York, pp. 59-72.

[128] Stiefel, E., *Kernel Polynomials in Linear Algebra and Their Numerical Applications*. In Further Contributions to the Solution of Simultaneous linear Equations and the Detreminations of Eingenvalues, Appl. Math. 49, 1958, Nat. Bur. Standards, U. S. Government Printing Office, Washington, D. C., pp. 1-22.

[129] Strang, G., *Introduction to Linear Algebra*. Wellesley-Cambridge Press, Wellesley, USA, 1993.

[130] Takahasi, H., Natori, M., *Eingenvalue Problem of Large Sparse Matrices*. Rep. Compt. Centre, 1971-72, Univ. Tokyo 4, pp. 129-148.

[131] Tong, C., *A Comparative Study of Preconditioned Lanczos Methods for Nonsymmetric Linear Systems*. Tech. Rep. SAND91-8240, Sandia Nat. Lab., Livermore, CA, 1992.

[132] Underwood, R., *An Iterative Block Lanczos method for the Solution of Large Sparse Symmetric Eigenproblems*. Ph. D. Dissertation, Stanford University Computer Science Dept. Report STAN-CS-75-496, 1975, Stanford, California.

[133] Van der Vorst, H. *Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems*. SIAM J. Sci. Statist. Comp. 13, 1992, pp. 631-644.

[134] Van der Vorst, H., Vuik, C., *GMRESR: A Family of Nested GMRES Methods*. Numer. Lin. Alg. Appl. 1, 1994, pp. 369-386.

[135] Vassilevski, P., *Preconditioning Nonsymmetric and Indefinide Finite Element Matrices*. J. Numer. Alg. Appl. 1, 1992, pp. 59-76.

[136] Wang, R., Treitel, S., *The Determination of Digital Wiener Filters by Means of Gradient Methods*. Geophysics 38, 1973, pp. 310-326.

[137] Weaver, Jr., Yoshida, W., D., *The Eingenvalue Problem for Banded Matrices*. Computers and Structures 1, 1971, pp. 651-664.

[138] Whitehead, R., *A Numerical Approach to Nuclear Shell-Model Calculations*. Nuclear Physics A182, 1972, pp. 290-300.

[139] Wilkinson, J., *The Algebraic Eigenvalue Problem*, Oxford, Clarendon Press,1961.

[140] Zoutendijk, G., *Methods of Feasible Directions*. Elsevier, Amsterdam, 1960.

Apêndice

Solução do Exemplo 2.10. Esse exemplo foi resolvido no *software* MATLAB, com o auxílio dos programas *mgc3t.m* e *mgc2t.m*, de nossa autoria.

(a) $n = 3000$

```
»i=1:3000;
»a=sparse(i,i,4);
»j=1:2999;
»d=sparse(j,j+1,-1);d=[d;zeros(1,3000)];
»c=sparse(j+1,j,-1);c=[c zeros(3000,1)];
»for k=1:1500
b(2*k)=2;
b(2*k-1)=-1;
end
»b=b';
»xo=randint(3000,1,2,1);
»format long;iter=100;tol=1e-15;

»[x1,n1,err1,suc]=mgc2t(A,b,xo,iter,tol);
elapsed_time =
   0.83000000000000
flops=
    2264983
n1 =
   31
err1 =
 6.794182415340612e-016
suc =
    0

» [x2,n2,err2,suc]=mgc3t(A,b,xo,iter,tol);
elapsed_time =
   1.32000000000000
f lops=
    2754181
n2 =
   31
err2 =
 6.794182415340597e-016
suc =
    0
```

```
» normdif=norm(x2-x1)
normdif =
  1.2064012032041O2e-014
```

(b) n = 4000;

```
»i=1:4000;
»a=sparse(i,i,4);
»j=1:3999;
»d=sparse(j,j+1,-1);d=[d;zeros(1,4000)];
»c=sparse(j+1,j,-1);c=[c zeros(4000,1)];
»for k=1:2000
b(2*k)=2;
b(2*k-1)=-1;
end
»b=b';
»xo=randint(4000,1,2,1);
»format long;iter=100;tol=1e-15;

» [x1,n1,err1,suc]=mcg2t(A,b,xo,iter,tol);
elapsed_time =
  1.43000000000000
flops =
    3019965
n1 =
   31
err1 =
   7.575421742913401e-016
suc =
    0

»[x2,n2,err2,suc]=mgc3t(A,b,xo,iter,tol);
elapsed_time =
  1.70000000000000
f lops=
    3672181
n2 =
   31
err2 =
   7.575421742913410e-016
suc =
    0

» normdif=norm(x1-x2)
normdif=
   1.435376190864308e-014
```

Programas construídos, no MATLAB, para ilustrar o Exemplo 2.10.

(a) *M-file mgc2t.m*

```
function [x,n,err,suc]=mgc2t(A,b,x,iter,tol)

%Esse algoritmo implementa a forma de recorrência de dois termos do MGC,
%para resolver SELAS Ax = b.
%       Entrada         A, matriz s. p. d. do SELAS;
%                       b, vetor coluna dos termos independentes;
%                       x, aproximação inicial;
%                       iter, número máximo de iterações;
%                       tol, tolerância para o erro.
%       Saída           x, solução obtida;
%                       n, número de iterações realizadas;
%                       err, estimativa de erro;
%                       suc, estimativa do sucesso.
%       suc=1, a tolerância foi atingida;
%       suc=0, não ocorreu a convergência para o número de iterações;
%                       elapsed_time, tempo gasto, em segundos, na execução do programa;
%                       f, número de flops realizadas.
%Vânia M. P. Slaviero. Novembro de 1997.


tic
flops(0);
suc = 1;
n=0;
r=A*x-b;d=-r;err=norm(r);
while (n<=iter)&(err>tol)
  n=n+1;h=A*d;
  p=(r'*r)/(d'*h);
  x=x+p*d;
  r=r+p*h;
  k=(r'*h)/(d'*h);
  d=-r+k*d;
  err=norm(r);
end
if( err > tol )suc = 0; end
f=flops
toc
```

(b) *M-file mgc3t.m*

```
function [x,n,err,suc]=mgc3t(A,b,x,iter,tol)

%Esse algoritmo implementa a forma de recorrência de três termos do MGC,
%para resolver SELAS Ax = b.
%       Entrada         A, matriz s. p. d. do SELAS;
%                       b, vetor coluna dos termos independentes;
%                       x, aproximação inicial;
%                       iter, número máximo de iterações;
```

```
%                       tol, tolerância para o erro.
%       Saída           x, solução obtida;
%                       n, número de iterações realizadas;
%                       err, estimativa de erro;
%                       suc, estimativa do sucesso.
%       suc=1, a tolerância foi atingida;
%       suc=0, não ocorreu a convergência para o número de iterações;
%                       elapsed_time, tempo gasto, em segundos, na execução do programa;
%                       f, número de flops realizadas.
%Vânia M. P. Slaviero. Novembro de 1997.

tic
flops(0);
suc = 1;
n=0;
r=A*x-b;ar=A*r;beta=r'*r/(r'*ar);alfa=1;err=norm(r);
p=0;q=0;
while (n<=iter)&(err>tol)
  xvelho=x;rvelho=r;n=n+1;
  x=alfa*x+(1-alfa)*p-beta*r;
  r=alfa*r+(1-alfa)*q-beta*ar;
  ar=A*r;rr=r'*r;p=xvelho;q=rvelho;
  beta=((r'*ar/rr)-(rr/(q'*q))*(beta\1))\1;
  alfa=beta*r'*ar/rr;
  err=norm(r);
end
if( err > tol )suc = 0; end
f=flops
toc
```

Programa construído, no MATLAB, para ilustrar o Exemplo 3.8.4.1.

M-file lanczos.m

```
function [x,esc,d,R,S]=lanczos(A,C,b,do)

%Resolve um SELAS Ax = b pelo método de Lanczos na versão A-ortogonal,
%com precondicionamento.
%       Entrada         A, matriz do SELAS;
%                       C, matriz precondicionadora;
%                       b, vetor coluna dos termos independentes;
%                       do, vetor coluna inicializador do processo.
%       Saída           x, solução do SELAS;
%                       esc, vetor coluna formado pelos escalares alfa;
%                       d, matriz cujas colunas são formadas pelos vetores de Lanczos;
%                       R, vetor linha formado pelos escalares r;
%                       S, vetor linha formado pelos escalares s.
%Os escalares r e s são componentes da matriz T=tridiag(1,r(i),s(i)).
%Vânia M. P. Slaviero. Novembro de 1997.
```

```
[n,n]=size(A);
d=[do];
ad=A*d(:,1);matr=C\A;
a=(d(:,1))'*b/((d(:,1))'*ad);r=(matr*d(:,1))'*ad/((d(:,1))'*ad);s=0;p=0;R=[r];S=[s];
esc=[a];x=[a*(d(:,1))'];
for i=2:n
  d(:,i)=matr*d(:,i-1)-r*d(:,i-1)-s*p;
  p=d(:,i-1);
  advelho=ad;
  ad=A*d(:,i);
  a(i)=(d(:,i))'*b/((d(:,i))'*ad);
  esc=[esc;a(i)];
  r=(matr*d(:,i))'*ad/((d(:,i))'*ad);
  R=[R r];
  s=(d(:,i))'*ad/((d(:,i-1))'*advelho);
  S=[S s];
  x=[x;a(i)*(d(:,i))'];
end
x=sum(x);
```