

Atividade para casa #2

Soluções de Sistemas lineares

Contando operações no algoritmo de eliminação Gaussiana

May 3, 2025

1 Contextualização

Nesta parte do curso entraremos em um assunto fascinante e fundamental no campo dos métodos numéricos: sistemas lineares. Na verdade, o objetivo aqui é desenvolver esquemas numéricos voltados à solução de sistemas de equações lineares. Já vimos brevemente neste curso como podemos representar um sistema de equações algébricas lineares de n variáveis na forma de um sistema passível de organização numa estrutura matricial do tipo $[A]_{n,n}\{x\}_n = \{b\}_n$ em que os subíndices indicam a ordem de cada matriz ou vetor. Uma forma geral de representarmos um sistema linear pode ser dada por

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}. \quad (1)$$

Nessa estrutura da equação (1) dizemos que \mathbf{A} é a matriz dos coeficientes, \mathbf{x} é o vetor solução e \mathbf{b} é um vetor associado a termos de fonte. Estruturas matemáticas com essa característica aparecem em diversos problemas do mundo real. Toda simulação computacional desenvolvida para modelagem de problemas de engenharia tem dentro de seu código fonte alguma chamada para uma função, subrotina ou biblioteca cuja única função dentro desse código consiste em resolver um sistema linear. Podemos dizer que sistemas lineares são uma espécie de *blocos construtores* no campo dos métodos numéricos, uma vez que costumam aparecer em tópicos como otimização, regressão e soluções de equações diferenciais ordinárias e parciais.

O objetivo desta parte do curso é se aprofundar em métodos desenvolvidos justamente com essa finalidade: resolver sistemas de equações do tipo expresso em (1) com o objetivo de determinar $\{x\}$ a partir do conhecimento de $[A]$ e $\{b\}$. Mas antes de entrarmos nesse tópico, vamos definir brevemente algumas questões básicas para falarmos sobre terminologia associada a alguns tipos especiais de matrizes quadradas.

O primeiro tipo de matriz especial é a **matriz simétrica**, que ocorre quando $a_{ij} = a_{ji}$ para todos os índices $i = [1, n]$ e $j = [1, n]$ de uma matriz quadrada $[A]$ de ordem n . A relação entre as componentes a_{ij} e a_{ji} pode ser também representada como $[A] = [A]^T$, em que $[A]^T$ é a matriz transposta de $[A]$, obtida a partir da troca de linhas por colunas. Matrizes simétricas são muito frequentes e surgem em diversos problemas físicos. Por exemplo, o tensor de tensões de um fluido Newtoniano, tem suas componentes alocadas na forma de uma matriz 3×3 que no espaço 3D é sempre simétrica.

Um outro tipo de matriz especial é a matriz diagonal, que ocorre sempre que todos os termos não nulos se localizam na sua diagonal principal e o restante dos termos é zero. Essa matriz pode ser representada visualmente por

$$[A] = \begin{pmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_{nn} \end{pmatrix}. \quad (2)$$

Um tipo especial de matriz diagonal é a matriz identidade $[I]$ que ocorre quando $a_{ii} = 1$ e $a_{ij} = 0$ para $i \neq j$. Uma forma de representar as componentes da matriz identidade é dada por $a_{ij} = \delta_{ij}$, em que δ_{ij} é o operador delta de Kronecker, um operador matemático muito importante que vale $\delta_{ij} = 0 \rightarrow i \neq j$ e $\delta_{ij} = 1 \rightarrow i = j$. Quando o **produto matricial** entre duas matrizes $[A]$ e $[B]$ resulta na matriz identidade $[I]$ dizemos que a matriz $[B]$ é a matriz inversa de $[A]$, ou simplesmente $[B] = [A]^{-1}$. Note que se conhecemos por exemplo a matriz inversa de $[A]$, podemos obter rapidamente a solução $\{x\}$ do nosso sistema linear por meio da seguinte conta

$$[A]^{-1}[A]\{x\} = [A]^{-1}\{b\} \longrightarrow [I]\{x\} = \{x\} = [A]^{-1}\{b\}. \quad (3)$$

Apesar de parecer simples resolver um sistema linear por meio do uso da matriz inversa, o processo de inversão matricial não é simples e muito menos barato computacionalmente. A inversão matricial, apesar de poder ser utilizada como direcionamento para a solução de um sistema linear envolve um algoritmo computacionalmente custoso. Um outro tipo importante de matriz especial são as matrizes triangulares, que podem ser divididas entre matrizes triangulares superior $[A]^\Delta$ e inferior $[A]_\nabla$, ambas dadas por

$$[A]^\Delta = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \quad \text{e} \quad [A]_\nabla = \begin{pmatrix} a_{11} & 0 & 0 & 0 \\ a_{21} & a_{22} & 0 & 0 \\ a_{31} & a_{32} & a_{33} & 0 \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}. \quad (4)$$

Matrizes triangulares são a chave por trás de técnicas muito poderosas e inteligentes de solução de sistemas lineares, conhecidas como técnicas de eliminação, como veremos mais para frente. Um outro tipo de matriz especial importante no nosso estudo são as matrizes de banda, nas quais todos os elementos não nulos se concentram numa faixa centrada em torno da diagonal principal. Essas matrizes podem ser representadas por

$$[A] = \begin{pmatrix} a_{11} & a_{12} & 0 & 0 \\ a_{21} & a_{22} & a_{23} & 0 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & 0 & a_{43} & a_{44} \end{pmatrix}. \quad (5)$$

No caso do exemplo da matriz da equação (5), temos 3 elementos nas regiões mais largas da faixa, nesse caso dizemos que essa banda possui 3 elementos e que a matriz é uma matriz de largura de banda igual a 3, o que a classifica como uma matriz tri-diagonal. Finalmente, quando a maior parte dos elementos de uma matriz é zero, dizemos simplesmente que esta matriz é esparsa.

Pronto! Agora já temos o que precisamos para avançar na discussão de como buscar formas de resolvermos esses sistemas lineares. Vamos então começar com um sistema pequeno de ordem 2 para entendermos alguns pontos. Considere então o sistema dado por

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2. \end{aligned} \quad (6)$$

Podemos começar isolando x_2 da primeira equação expressa no sistema (6) para obter

$$x_2 = \frac{b_1}{a_{12}} - \left(\frac{a_{11}}{a_{12}} \right) x_1 \quad (7)$$

e em seguida isolar x_2 da segunda equação do sistema (6) para obter

$$x_2 = \frac{b_2}{a_{22}} - \left(\frac{a_{21}}{a_{22}} \right) x_1. \quad (8)$$

Note que as equações (7) e (8) representam duas equações de retas e envolvem todos os coeficientes da matriz $[A]$ e do vetor $\{b\}$. Se plotarmos essas duas retas, a solução $\{x_1, x_2\}$ do sistema que estamos buscando é simplesmente o ponto em que as duas retas se cruzam, como ilustrado na figura (1). A técnica de isolar uma variável em função da outra, plotar em um gráfico e verificar o ponto em que essas soluções se cruzam é conhecida como método gráfico. Esse método é simples e fácil

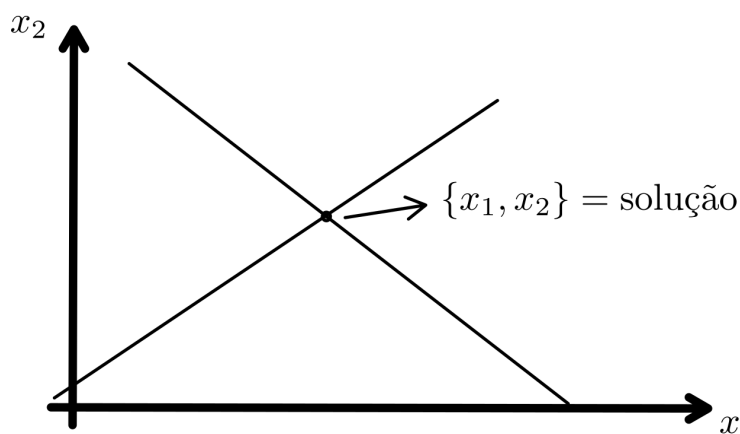


Figure 1: Exemplo de uso do método gráfico para resolver um sistema linear de ordem 2

de ser aplicado a sistemas de ordem 2. Para um sistema de ordem 3, teríamos uma função de 2 variáveis $x_2 = f(x_1, x_3)$ ao isolar x_2 de qualquer uma das 3 equações que compõe o sistema. Nesse caso, ao invés de uma reta teríamos um plano. O ponto de cruzamento desses 3 planos seria a solução desse sistema de ordem 3. Uma ideia dessa imagem é representada na figura (2).

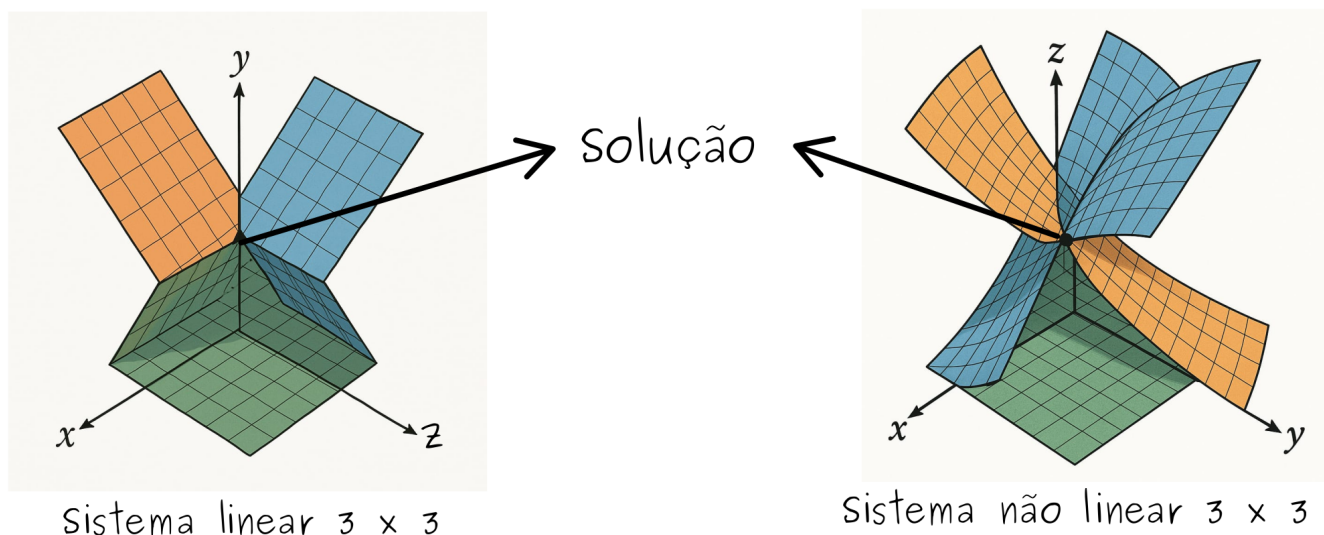


Figure 2: Aplicação do método gráfico para resolver sistemas lineares e não-lineares de ordem 3

Evidentemente, quando a ordem do sistema é $n > 3$, o método gráfico deixa de ser uma alternativa viável e precisamos então de uma solução mais robusta.

1.1 Determinantes e a regra de Cramer

Uma alternativa consiste na solução do sistema por meio da regra de Cramer, uma técnica antiga que usa do conceito de determinantes. Podemos definir a princípio o determinante de uma matriz como um único número, que carrega consigo características muito representativas da estrutura dos números organizados nessa matriz. Para uma matriz quadrada de ordem 3, esse determinante é definido como

$$D = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}(a_{22}a_{33} - a_{23}a_{32}) + a_{12}(a_{31}a_{23} - a_{33}a_{21}) + a_{13}(a_{21}a_{32} - a_{22}a_{31}). \quad (9)$$

Na equação (9) os termos entre parênteses são na verdade determinantes de sub-matrizes 2×2 localizadas no interior da matriz principal de ordem 3×3 . Uma questão importante aqui é como o custo computacional do cálculo dessa propriedade cresce com o aumento da ordem do sistema. Se para o cálculo do determinante de uma matriz de ordem 3, precisamos calcular 3 determinantes de uma matriz de ordem 2, a mesma lógica se aplica quando vamos subindo a ordem. Calcular o determinante de uma matriz de ordem 4 equivalente a calcular 4 determinantes de uma matriz de ordem 3, que por sua vez consiste em calcular 3 determinantes de uma matriz de ordem 2. Dessa forma, o custo computacional relacionado ao cálculo do determinante de uma matriz de ordem n é $\mathcal{O}(n!)$. O que para os nossos padrões é considerado absurdo do ponto de vista de custo computacional. Entretanto, para sistemas menores, ainda podemos utilizar o cálculo de determinantes para resolver o sistema por meio da regra de Cramer, que para um sistema 3×3 pode ser resumida por meio das equações expressas em (10).

$$x_1 = \frac{1}{D} \begin{vmatrix} b_1 & a_{12} & a_{13} \\ b_2 & a_{22} & a_{23} \\ b_3 & a_{32} & a_{33} \end{vmatrix}, \quad x_2 = \frac{1}{D} \begin{vmatrix} a_{11} & b_1 & a_{13} \\ a_{12} & b_2 & a_{23} \\ a_{31} & b_2 & a_{33} \end{vmatrix}, \quad x_3 = \frac{1}{D} \begin{vmatrix} a_{11} & a_{12} & b_1 \\ a_{21} & a_{22} & b_2 \\ a_{31} & a_{32} & b_3 \end{vmatrix}. \quad (10)$$

Note que o processo de cálculo dos elementos do vetor solução consiste em dividir o determinante da matriz formada pela substituição do vetor $\{b\}$ na coluna correspondente ao número do elemento em $\{x\}$ que desejamos calcular para em seguida dividir esse determinante por D , que nada mais é do que o determinante da matriz $[A]$, também representado por $D = \det[A]$. Uma desvantagem clara da regra de Cramer é o fato de que precisamos calcular um determinante novo para cada elemento do vetor solução \mathbf{x} que estamos buscando, o que fornece um custo computacional total $\mathcal{O}(n \times n!)$.

Esse alto custo computacional demandará a busca de soluções mais inteligentes e viáveis, mas antes de partirmos nessa empreitada, precisamos falar um pouco mais sobre determinantes. Afinal de contas, que informação crucial sobre a estrutura de uma dada matriz se esconde no valor de seu determinante? Para respondermos essa questão, voltemos ao exemplo simples de um sistema 2×2 . Dessa vez, considere o sistema de equações a seguir

$$3x_1 - 5x_2 = 13, \quad (11)$$

$$1.5x_1 - 2.5x_2 = 6.5. \quad (12)$$

Note que essas duas equações são na verdade a mesma equação. A equação (12) foi obtida simplesmente pela divisão da equação (11) por 2. Nesse sentido, a equação (12) não traz nenhuma informação nova com relação às informações trazidas pela equação (11). Para esse caso, se pensarmos na ideia do método gráfico teríamos duas equações sobrepostas e os pontos de encontro entre as retas seriam infinitos. De outra maneira, como as duas equações são na verdade a mesma equação, para qualquer valor real de x_1 teríamos um par correspondente de x_2 , ou seja, o sistema

tem infinitas soluções. Agora vamos olhar para o determinante da matriz dos coeficientes desse mesmo exemplo:

$$D = \begin{vmatrix} 3 & -5 \\ 1.5 & -2.5 \end{vmatrix} = 3 \times (-2.5) - (-5) \times 1.5 = -7.5 + 7.5 = 0. \quad (13)$$

O fato do determinante desse sistema específico ter dado 0 é compatível com a regra de Cramer, uma vez que esse sistema é redundante (mal posto) e consequentemente não possui uma única solução, é de se esperar que a regra de Cramer não seja aplicável a esse tipo de sistema, já que não podemos ter em nenhum cenário o determinante nulo no denominador das expressões utilizadas para calcular a raiz por esse método. Sistemas lineares nos quais a matriz dos coeficientes possui determinante nulo são chamados de *singulares*. Um outro tipo de sistema singular é ilustrado a seguir:

$$4x_1 - 2x_2 = 20, \quad (14)$$

$$4x_1 - 2x_2 = 13. \quad (15)$$

Para esse último exemplo, a equação (15) não é redundante com relação à equação (14), uma vez que o lado direito de ambas as equações, ou seja, o vetor $\{b\}$ é diferente. Entretanto, essa diferença entre b_1 e b_2 não possui qualquer impacto na matriz dos coeficientes, de tal sorte que para esse exemplo $\det[A] = 0$. Aqui não se trata mais de infinitas soluções, mas simplesmente de duas retas com o mesmo coeficiente angular que nunca se cruzam e portanto não existe solução. Finalmente, um outro tipo de cenário é quando o determinante não é exatamente nulo, mas muito pequeno. Nesse caso, se pensarmos no método gráfico, teríamos duas retas que são muito próximas, de tal sorte que pode ser difícil detectar um ponto em que ambas se cruzam, como ilustrado na figura (3). Nesse caso, quando $\det[A] \ll 1$, dizemos simplesmente que o sistema é mal condicionado. Nesse caso, erros de arredondamento podem levar a problemas de divergência na busca por uma solução numérica do problema. Em ambos os casos, sistemas singulares e mal-condicionados são um problema a ser resolvido.

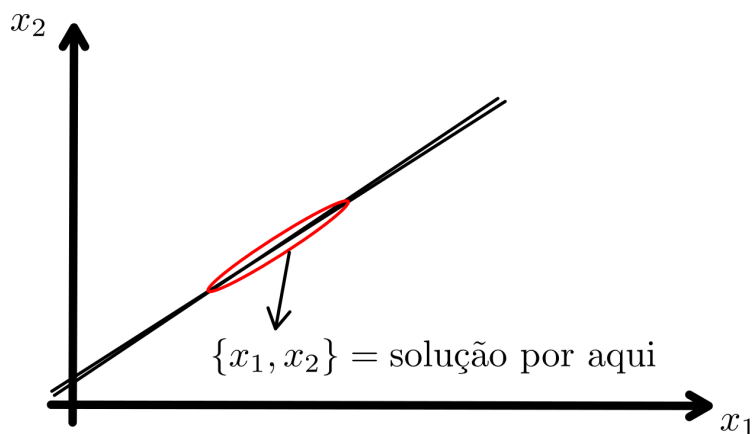


Figure 3: Exemplo de como um sistema mal condicionado se comportaria pelo método gráfico

1.2 A eliminação de variáveis

Essa discussão que tivemos sobre determinantes e a regra de Cramer serviu para abrir um pouco a nossa cabeça com relação a conceitos importantes no campo da busca por métodos numéricos voltados à solução de sistemas lineares. Entretanto, como vimos até o momento, enquanto o método

gráfico é impraticável para sistemas de ordem $n > 3$, o custo computacional da regra de Cramer é $\mathcal{O}(n \times n!)$, o que torna a aplicação do mesmo impraticável para sistemas de alta ordem.

Uma solução viável para o problema consiste na construção de métodos baseados na estratégia de eliminação de variáveis. Tais métodos são chamados de métodos de eliminação. Para introduzirmos o conceito central por trás da técnica de eliminação de variáveis voltemos a nosso sistema 2×2 :

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 &= b_1 \\ a_{21}x_1 + a_{22}x_2 &= b_2. \end{aligned} \tag{16}$$

Considere multiplicar a primeira equação em (16) por a_{21} :

$$a_{21}a_{11}x_1 + a_{21}a_{12}x_2 = a_{21}b_1. \tag{17}$$

Agora, multiplicamos a segunda equação em (16) por a_{11} :

$$a_{11}a_{21}x_1 + a_{11}a_{22}x_2 = a_{11}b_2. \tag{18}$$

Agora subtraímos (17) de (18) para eliminar x_1 do resultado final:

$$a_{21}a_{12}x_2 - a_{11}a_{22}x_2 = a_{21}b_1 - a_{11}b_2. \tag{19}$$

Finalmente, isolamos x_2 de (19) para obter

$$x_2 = \frac{a_{21}b_1 - a_{11}b_2}{a_{21}a_{12} - a_{11}a_{22}}, \tag{20}$$

Finalmente, substituímos x_2 obtido em (20) em qualquer uma das equações originais que compõe o sistema e isolamos x_1 para finalizar a solução do sistema. Para sistemas lineares de ordem $n > 2$ podemos aplicar um algoritmo semelhante para resolver o sistema por meio de uma série de manipulações dessa natureza que se organizam na forma de um método conhecido como eliminação Gaussiana.

1.3 A eliminação Gaussiana ingênua

A ideia do algoritmo de eliminação Gaussiana consiste em dividir a estratégia de solução em duas etapas. Na etapa 1, chamada de *eliminação progressiva*, transformamos o sistema linear num sistema triangular (superior ou inferior). Na etapa 2, que chamamos de *substituição regressiva*, partimos de baixo (ou de cima, dependendo de como optamos por fazer a transformação na etapa 1), e por meio de uma série de substituições regressivas vamos estimando cada valor de $\{x\}$. A ideia fica mais clara com imagens. A transformação de um sistema linear do tipo $[A]\{x\} = \{b\}$ em um sistema triangular superior na etapa de eliminação leva a um sistema do tipo:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ 0 & a'_{22} & a'_{23} & a'_{24} \\ 0 & 0 & a''_{33} & a''_{34} \\ 0 & 0 & 0 & a'''_{44} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} b_1 \\ b'_2 \\ b''_3 \\ b'''_4 \end{pmatrix}. \tag{21}$$

Note que aqui tanto os termos da matriz $[A]$ quanto do vetor $\{b\}$ sofrem alterações ao longo do processo de transformação, porém a solução $\{x\}$ do sistema modificado é a mesma do sistema original. Essa transformação do sistema original em um sistema triangular sem alteração da solução é a

grande ideia por trás do método da eliminação Gaussiana. Uma vez que tenhamos triangularizado o sistema, podemos começar resolvendo ele de baixo para cima para obter:

$$x_4 = \frac{b_4'''}{a_{44}'''}, \quad (22)$$

$$x_3 = (b_3'' - a_{34}''x_4)/a_{33}'', \quad (23)$$

$$x_2 = (b_2' - a_{23}'x_3 - a_{24}'x_4)/a_{22}', \quad (24)$$

$$x_1 = (b_1 - a_{12}x_2 - a_{13}x_3 - a_{14}x_4)/a_{11}. \quad (25)$$

Essa última etapa é o que chamamos de substituição regressiva. Mais para frente voltaremos à questão da palavra *ingênua* no título dessa seção. Por enquanto, o que estamos buscando para entender melhor como realizar essas duas etapas na prática, precisamos construir o algoritmo por meio de um exemplo. Considere então um sistema linear geral de ordem n , expresso por

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \quad (26)$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n = b_2 \quad (27)$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n = b_n \quad (28)$$

Para esse sistema geral, apresentaremos a seguir uma espécie de receita de bolo para a construção do algoritmo de eliminação Gaussiana.

Passo a passo do algoritmo de Eliminação Gaussiana

1. Multiplique (26) por a_{21}/a_{11} :

$$a_{21}x_1 + \left(\frac{a_{21}}{a_{11}}\right)a_{12}x_2 + \left(\frac{a_{21}}{a_{11}}\right)a_{13}x_3 + \cdots + \left(\frac{a_{21}}{a_{11}}\right)a_{1n}x_n = \left(\frac{a_{21}}{a_{11}}\right)b_1; \quad (29)$$

2. Subtraia (30) de (27):

$$\left(a_{22} - \frac{a_{21}a_{12}}{a_{11}}\right)x_2 + \cdots + \left(a_{2n} - \frac{a_{21}a_{1n}}{a_{11}}\right)x_n = \left(b_2 - \frac{a_{21}b_1}{a_{11}}\right), \quad (30)$$

que pode ser reescrita como

$$a'_{22}x_2 + \cdots + a'_{2n}x_n = b'_2; \quad (31)$$

3. Aplica-se o mesmo procedimento para as linhas subsequentes. Por exemplo, multiplicando (26) por a_{31}/a_{11} e subtraindo o resultado da terceira linha do sistema, eliminamos o coeficiente vinculado à x_1 dessa linha também. Nesse procedimento, a equação (26) é chamada de equação pivô e o termo a_{11} é denominado *coeficiente ou elemento pivô*. Aplicando esse procedimento para as demais linhas da matriz nos leva a uma estrutura do tipo:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1 \quad (32)$$

$$a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n = b'_2 \quad (33)$$

$$a'_{32}x_2 + a'_{33}x_3 + \cdots + a'_{3n}x_n = b'_3 \quad (34)$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$a'_{n2}x_2 + a'_{n3}x_3 + \cdots + a'_{nn}x_n = b'_n \quad (35)$$

4. Precisamos agora eliminar x_2 da equação (34) em diante. Para isso, multiplicamos a equação (33) pelo fator a'_{32}/a'_{22} e subtraímos o resultado de (34), transformando então a equação (34) em:

$$a''_{33}x_3 + \cdots + a''_{3n}x_n = b''_3, \quad (36)$$

aplicando esse passo para todas as linhas abaixo da equação (34), transformamos nosso sistema em

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

$$a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n = b'_2$$

$$a''_{33}x_3 + \cdots + a''_{3n}x_n = b''_3$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$a'_{n3}x_3 + \cdots + a'_{nn}x_n = b'_n$$

5. Continuamos o processo para irmos eliminando x_3 a partir da linha 4 em diante e assim sucessivamente, até que nosso sistema tenha se tornado um sistema triangular superior do tipo

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n = b_1$$

$$a'_{22}x_2 + a'_{23}x_3 + \cdots + a'_{2n}x_n = b'_2$$

$$a''_{33}x_3 + \cdots + a''_{3n}x_n = b''_3$$

$$\vdots \quad \quad \quad \vdots \quad \quad \quad \vdots$$

$$a^{(n-1)}_{nn}x_n = b^{(n-1)}_n.$$

Nessa simbologia o número de aspas representa o número de modificações pelas quais aquela linha passou durante o processo de eliminação. O expoente $(n-1)$ nos coeficientes da última linha do sistema indica que essa linha passou por $n-1$ modificações ao longo do caminho de triangularização do sistema.

6. O último passo para resolvermos nosso sistema, consiste na aplicação da etapa de substituição regressiva. Para isso, começamos calculando x_n com

$$x_n = \frac{b^{(n-1)}_n}{a^{(n-1)}_{nn}}; \quad (37)$$

7. Em seguida, iteramos dentro do seguinte laço:

$$x_i = \frac{b^{(i-1)}_i - \sum_{j=i+1}^n a^{(i-1)}_{ij}x_j}{a^{(i-1)}_{ii}} \rightarrow i = n-1, n-2, \dots, 1. \quad (38)$$

1.4 Contando o número de operações do algoritmo de eliminação Gaussiana

Como vimos até o momento, o custo computacional associado à solução de sistemas lineares por métodos que envolvem o cálculo de determinantes é muito alto e portanto precisamos buscar caminhos alternativos para tal missão. Apresentamos o algoritmo de eliminação Gaussiana como uma proposta que até esse momento vem sendo vendida como viável para tal finalidade. Entretanto, precisamos avaliar efetivamente o custo computacional desse novo algoritmo. Portanto, temos que contar o número de operações de ponto flutuante (FLOPS) que o método da eliminação Gaussiana demanda.

Antes de entrarmos no detalhe dessa contagem, precisamos definir algumas relações que irão facilitar o processo de contagem. Essas relações encontram-se definidas abaixo.

$$\sum_{i=1}^m cf(i) = c \sum_{i=1}^m f(i) \quad (39)$$

$$\sum_{i=1}^m [f(i) + g(i)] = \sum_{i=1}^m f(i) + \sum_{i=1}^m g(i) \quad (40)$$

$$\sum_{i=1}^m 1 = 1 + 1 + 1 + \cdots + 1 = m \quad (41)$$

$$\sum_{i=k}^m 1 = m - k + 1 \quad (42)$$

$$\sum_{i=1}^m i = 1 + 2 + 3 + \cdots + m = \frac{m(m+1)}{2} = \frac{m^2}{2} + \mathcal{O}(m) \quad (43)$$

$$\sum_{i=1}^m i^2 = 1 + 2^2 + 3^2 + \cdots + m^2 = \frac{m(m+1)(2m+1)}{6} = \frac{m^3}{3} + \mathcal{O}(m^2) \quad (44)$$

Das relações definidas pelas equações (39-44), as que merecem maiores comentários são as equações (43) e (44). A equação (43) nada mais é do que a clássica fórmula, proposta inclusive por Gauss, para a soma dos termos de uma progressão aritmética. A equação (44) merece uma discussão um pouco mais aprofundada para ser adequadamente compreendida. Note que $\sum_{i=1}^m 1 = \sum_{i=1}^m i^0 = m$, enquanto $\sum_{i=1}^m i \sim m^2$, dessa forma, podemos propor que $\sum_{i=1}^m i^2 \sim m^3$. Uma forma geral de construção dessa relação se dá por meio de um polinômio de terceiro grau com a seguinte fórmula geral:

$$\sum_{i=1}^m i^2 = Am^3 + Bm^2 + Cm + D, \quad (45)$$

Considerando valores de $m = 0$, $m = 1$, $m = 2$ e $m = 3$, obtemos:

$$\sum_{i=1}^0 i^2 \rightarrow D = 0,$$

$$\sum_{i=1}^1 i^2 \rightarrow A + B + C = 1,$$

$$\sum_{i=1}^2 i^2 \rightarrow 8A + 4B + 2C = 5,$$

$$\sum_{i=1}^3 i^2 \rightarrow 27A + 9B + 3C = 14.$$

Resolvendo o sistema linear 3×3 acima obtemos a fórmula fechada da equação (44). É interessante notar que até na dedução de uma das relações que usaremos para contar o número de operações do algoritmo de eliminação Gaussiana acabamos tendo que resolver um pequeno sistema linear.

Para facilitar nossa contagem de operações no algoritmo, considere o pseudocódigo em FORTRAN expresso nas figuras (4) e (5). Essas imagens representam respectivamente as etapas 1 e 2 do nosso algoritmo.

```
! Etapa de eliminação progressiva

do k=1, n-1 ! linha anterior
  do i=k+1, n ! linha atual
    fator=a(i,k)/a(k,k)
    do j = k+1, n ! colunas
      a(i,j) = a(i,j) - fator*a(k,j)
    end do
    b(i) = b(i) - fator*b(k)
  end do
end do
```

Figure 4: Pseudocódigo em FORTRAN para a implementação da etapa de eliminação progressiva no algoritmo de eliminação Gaussiana

```
! Etapa de substituição regressiva

x(n) = b(n)/a(n,n) ! começando de baixo

do i=n-1, 1 ! subindo de baixo pra cima
  soma=b(i)
  do j = i+1, n
    soma = soma - a(i,j)*x(j)
  end do
  x(i) = b(i)/a(i,i)
end do
```

Figure 5: Pseudocódigo em FORTRAN para a implementação da etapa de substituição regressiva no algoritmo de eliminação Gaussiana

Para realizarmos a contagem de operações, vamos organizar nosso raciocínio na forma de tópicos.

- Na primeira passagem pelo primeiro laço em k na etapa de eliminação, temos $k = 1$, logo o laço do meio em i varia nessa primeira passagem de 2 até n ;
- O número de iterações no laço do meio em i é portanto

$$\sum_{i=2}^n 1 = n - 2 + 1 = n - 1;$$

- Para cada uma dessas iterações temos:

1. 1 divisão para o cálculo do fator;
 2. 1 multiplicação e 1 subtração para o cálculo de $a(i, j)$, que é realizada $n - 1$ vezes no laço em j ;
 3. 1 multiplicação e 1 subtração para o cálculo de $b(i)$;
- Dessa forma, para $k = 1$ temos $n + 1$ multiplicações ou divisões e n somas ou subtrações para o laço em i ;
 - O total de multiplicações ou divisões para $k = 1$ é portanto $(n - 1) \times [2 + (n - 1)] = (n - 1) \times (n + 1)$;
 - O total de operações de soma ou subtração, ainda para $k = 1$ é dado por: $(n - 1) \times [1 + (n - 1)] = (n - 1) \times n$;
 - Podemos então estender esse raciocínio para as próximas iterações em k por meio da tabela (1):

laço em k	laço em i	FLOPS _(+/-)	FLOPS _(x/÷)
1	$2 \rightarrow n$	$(n - 1) \times n$	$(n - 1) \times (n + 1)$
2	$3 \rightarrow n$	$(n - 2) \times (n - 1)$	$(n - 2) \times n$
\vdots	\vdots	\vdots	\vdots
k	$k + 1 \rightarrow n$	$(n - k) \times (n - k + 1)$	$(n - k) \times (n - k + 2)$
\vdots	\vdots	\vdots	\vdots
$n - 1$	$n \rightarrow n$	1×2	1×3

Table 1: Contagem de operações em cada laço da etapa de eliminação do algoritmo de eliminação Gaussiana

- O total de operações de ponto flutuante para soma ou subtração para todos os laços na etapa de eliminação progressiva é dado por:

$$\text{FLOPS}_{(+/-)} = \sum_{k=1}^{n-1} (n - k)(n - k + 1) = \sum_{k=1}^{n-1} [n(n + 1) - k(2n + 1) + k^2],$$

aplicando as propriedades definidas no início dessa seção, temos:

$$\text{FLOPS}_{(+/-)} = n(n + 1) \sum_{k=1}^{n-1} 1 - (2n + 1) \sum_{k=1}^{n-1} k + \sum_{k=1}^{n-1} k^2,$$

desenvolvendo a expressão acima, temos:

$$\text{FLOPS}_{(+/-)} = n \times (n + 1) \times (n - 1) - (2n + 1) \times \frac{(n - 1) \times n}{2} + \frac{(n - 1) \times n \times (2n - 1)}{6} \rightarrow \frac{n^3}{3} + \mathcal{O}(n).$$

Provamos então que o custo computacional relacionado às operações de soma ou subtração durante a etapa de eliminação progressiva do algoritmo de eliminação Gaussiana é da ordem $\mathcal{O}(n)$, o que é muito melhor que o custo $\mathcal{O}(n \times n!)$ associado à regra de Cramer. É claro que este ainda não é o custo total do algoritmo, uma vez que precisamos computar também o número de operações de divisão e/ou multiplicação e todas as operações relacionadas à etapa de substituição regressiva.

2 Enunciado da tarefa

1. Mostre que o número de operações de ponto flutuante associadas à multiplicação e/ou divisão da etapa de eliminação progressiva é:

$$\text{FLOPS}_{(\times/\div)} = \frac{n^3}{2} + \mathcal{O}(n^2);$$

2. Em seguida, mostre que o número total de operações de ponto flutuante para a etapa de eliminação progressiva, considerando as parcelas de soma/subtração e divisão/multiplicação é dado por:

$$\text{FLOPS} = \frac{2n^3}{3} + \mathcal{O}(n^2);$$

3. Mostre agora que o número total de operações de ponto flutuante para a etapa de substituição regressiva é dado por:

$$\text{FLOPS} = n^2 + \mathcal{O}(n);$$

4. Finalmente, monte uma tabela contendo 6 colunas, mostrando as seguintes informações:

- Coluna 1: n , variando de 5 até 50000 com os seguintes passos:

$$n = [5, 10, 50, 100, 500, 1000, 5000, 10000, 50000];$$

- Coluna 2: número total de operações de ponto flutuante relacionadas à etapa de eliminação progressiva;
- Coluna 3: número total de operações de ponto flutuante relacionadas à etapa de substituição regressiva;
- Coluna 4: número total de operações de ponto flutuante do algoritmo, somando-se as duas etapas;
- Coluna 5: $2n^3/3$;
- Coluna 6: Percentual de FLOPS relacionados à etapa de eliminação progressiva com relação aos FLOPS totais do algoritmo;

O que você pode concluir a partir da sua tabela com relação ao peso de cada etapa no custo computacional do algoritmo de eliminação Gaussiana?

3 Referências bibliográficas

1. S. C. Chapra, R. P. Canale. “Métodos Numéricos para Engenharia.” McGrawHill, 5a edição (2008): 1-825.