TRIANGULARIZAÇÃO SEM PIVOTEAMENTO: UM (mau) EXEMPLO *

Carlos A. de Moura – UFF †
Niterói, RJ

Abstract

Com este exemplo – de grande simplicidade – mostra-se dramaticamente a necessidade de pivoteamento, na implementação computacional do processo de triangularização de Gauss para a solução de sistemas lineares de equações algébricas.

Palavras-chave: Álgebra Linear Computacional; escalonamento; triangularização de Gauss; sistemas lineares de equações algébricas

*Notas de aula: Cálculo Numérico/Introdução aos Métodos Numéricos

†Professor Visitante
http://www.ic.uff.br/∼demoura
demoura@ic.uff.br
fax: (55-21) 717 08 12; fone: (55-21) 717 09 70
s_mail: Instituto de Computação - UFF
Rua Passo da Pátria, 156
24210-240 Niterói, RJ Brasil

1 Introdução

O algoritmo de triangularização de Gauss, também conhecido como *escalonamento de matrizes*, aparenta, quando exposto num curso de Métodos Numéricos [5], não trazer nenhuma abordagem diferente daquela já conhecida no contexto tradicional de Álgebra Linear [1], até ser discutida a questão do pivoteamento. O exemplo aqui apresentado/aprofundado consta já do clássico [3].
Enfatizamos a importância de se dispor de um sistema computacional confiável para resolver sistemas lineares de equações algébricas, um problema ingenuamente classificado como simples, fácil, mas que se tem de resolver em pelo menos uma passagem de talvez 90% dos projetos em Computação Científica.

2 A reta de vírgula flutuante

No que segue, estaremos simulando (imitando) a forma como os programas de um computador (*software*) tratam as operações algébricas com os números reais. Estes necessitam ser aproximados, diferentemente da representação exata dos inteiros, ou mesmo dos números com "vírgula fixa" nas caixas registradoras de lojas comerciais.

Trabalharemos com a reta $\Re$ de vírgula[1] flutuante especificada pelos seguintes dados:

- base 10,
- mantissa com dois dígitos,
- limitantes M e N para os expoentes (arbitrários, os valores não influem no que desejamos mostrar).

Em outras palavras, qualquer número real r será representado nessa reta $\Re$ na forma

$$r \cong \pm a,b \times 10^j,$$

onde os inteiros a , b e j satisfazem

$$0 \leq \begin{array}{c} a \\ b \end{array} \leq 9 \, ,$$

$$-M \ < j \leq \ N \, ,$$

com M, N inteiros positivos. Esta notação significa aproximar r pelo valor

$$\rho = \pm[a \times 10^j + b \times 10^{j-1}].$$

Observe que $\Re$ é um conjunto finito, contém apenas uma quantidade finita de números – ou de pontos, a representação gráfica daqueles. E que esses pontos não se distribuem de forma homogênea, não são equidistantes como na régua ou nas trenas a que estamos habituados. De fato, por só dispormos de dois dígitos em $\Re$:

[1] *ponto*, na literatura de língua inglesa

- entre 10 e 100, apenas os inteiros estão (todos) presentes, nada entre eles constando da representação,
  $\mathbf{10} = 1,0 \times 10^1$, $\mathbf{11} = 1,1 \times 10^1$, $\ldots$, $\mathbf{99} = 9,9 \times 10^1$, $\mathbf{100} = 1,0 \times 10^2$;
- de 100 a 1000 apenas estão representados os inteiros múltiplos de 10, i.e.,
  $\mathbf{100} = 1,0 \times 10^2$, $\mathbf{110} = 1,1 \times 10^2$, $\ldots$, $\mathbf{990} = 9,9 \times 10^2$, $\mathbf{1000} = 1,0 \times 10^3$;
- entre 1000 e 10000 há uma ainda maior rarefação, temos apenas os múltiplos de 100,
  $\mathbf{1000} = 1,0 \times 10^3$, $\mathbf{1100} = 1,1 \times 10^3$, $\ldots, \mathbf{9900} = 9,9 \times 10^3$, $\mathbf{10000} = 1,0 \times 10^4$;

e assim sucessivamente, entre duas potências (positivas) de 10 – porque foi esse número tomado como base, neste caso.
Em contrapartida, ao nos aproximarmos da origem, temos:

- no intervalo [1, 10] , uma precisão de décimos,
  $\mathbf{1} = 1,0 \times 10^0$, $\mathbf{1,1} = 1,1 \times 10^0$, $\ldots$, $\mathbf{9,9} = 9,9 \times 10^0$, $\mathbf{10} = 1,0 \times 10^1$;
- de 0,1 a 1 , uma precisão de centésimos,
  $\mathbf{0,1} = 1,0 \times 10^{-1}$, $\mathbf{0,11} = 1,1 \times 10^{-1}$, $\ldots$, $\mathbf{0,99} = 9,9 \times 10^{-1}$, $\mathbf{1} = 1,0 \times 10^0$;
- entre 0,01 e 0,1 , uma precisão de milésimos,
  $\mathbf{0,01} = 1,0 \times 10^{-2}$, $\mathbf{0,011} = 1,1 \times 10^{-2}$, $\ldots$, $\mathbf{0,099} = 9,9 \times 10^{-2}$, $\mathbf{0,1} = 1,0 \times 10^{-1}$.

Podemos pensar que a reta $\Re$ de vírgula flutuante é a representação da reta real $\mathbb{R}$ na *escala logarítmica*, a qual nos é provavelmente familiar das aulas de laboratório de Física. Observemos ainda que, apesar da "não-homogeneidade" da reta $\Re$, ela contém um parâmetro homogêneo: os erros absolutos na representação de um número real podem ser muito elevados, como mostram as representações nos exemplos acima. Mas o erro relativo está sempre limitado por $0,5\%$, ou melhor,

$$\frac{|\rho - r\,|}{|\rho|} \leq \frac{0,5}{10} = 0,05.$$

3 Um sistema linear de aparência inocente

Consideremos o sistema linear de equações algébricas

$$\left[\begin{array}{rcrcr} x & - & 100y & = & -99 \\ 100x & - & y & = & 99 \end{array}\right. \tag{1}$$

cuja (única) solução é

$$\boxed{x \;=\; y \;=\; 1}$$

A aplicação direta do algoritmo de triangularização de Gauss, cf. [2], requer apenas um passo, por termos uma matriz 2×2. Nesse passo, é "zerado" o (único) coeficiente localizado abaixo da diagonal, para o que se efetuam as operações algébricas descritas a seguir, sempre

restritos à reta $\Re$.
A primeira linha da matriz dos coeficientes A – e do segundo membro –, multiplicada por $-a_{21}/a_{11}$, é adicionada à segunda linha:

$$\left.\begin{array}{ll} 100 - \frac{100}{1} \times 1 & = 0 \\ -1 - \frac{100}{1} \times (-100) = -0,1 \times 10^1 + 0,1 \times 10^5 & = 10^4 \\ 99 - \frac{100}{1} \times (-99) = 0,0099 \times 10^4 + 0,99 \times 10^4 & = 9,9 \times 10^3 \end{array}\right] . \qquad (2)$$

Assim, obtemos no lugar da segunda equação de (1)

$$10^4 y = 9,9 \times 10^3 \qquad \Rightarrow \qquad 10y = 9,9 \quad \Rightarrow \qquad y = 9,9 \times 10^{-1},$$

enquanto a primeira, que não se altera, fornece, ao substituirmos este valor de y ,

$$x - 10^2 \times (9,9 \times 10^{-1}) = -99 \;\Rightarrow\; x - 99 = -99 \quad \Rightarrow \quad x = 0 \quad .$$

O resultado que obtivemos já ultrapassa o nível da imprecisão: é **inexato**.
Foi o pequeno número de dígitos da reta de vírgula flutuante que usamos o que causou um tal erro catastrófico?
De fato, a representação dos dados (coeficientes e termo conhecido) foi feita de forma exata, os erros surgiram apenas na representação dos resultados das operações algébricas. Mas se estivéssemos trabalhando com quatro dígitos – o que equivale, neste caso, a lançar mão da chamada *precisão dupla* – teríamos uma resposta exata para (1), usando este mesmo algoritmo. Mas não é difícil chegar a um exemplo equivalente ao dado por (1), mas para essa reta $\Re$ de quatro dígitos, ou mesmo para qualquer outra "mais precisa", obtendo sempre uma resposta completamente **inexata**.
É a matriz quase singular?
Não, pois o determinante de A é da ordem de 10^4 , que não é um valor pequeno relativamente aos dados do problema (nem à sua resposta).
Para responder a essa pergunta de uma forma mais geométrica, olhamos para a transformação linear associada à matriz A . Podemos pensar nela como "quase" uma reflexão relativamente ao eixo $0x$, seguida de uma homotetia de razão 100, pois temos, para os vetores $\mathbf{i}$, $\mathbf{j}$:

$$\mathbf{i} = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \rightarrow \quad \begin{pmatrix} 100 \\ 1 \end{pmatrix} ,$$

$$\mathbf{j} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \quad \rightarrow \quad \begin{pmatrix} -1 \\ -100 \end{pmatrix} .$$

Assim, os vetores unitários da base canônica $\mathbf{i}$, $\mathbf{j}$ podem ser pensados "quase" como autovetores, com autovalores 100 e -100, respectivamente. A imagem desta base pela transformação A nos dá uma base "quase" ortogonal, o que afasta a hipótese de ser A mal condicionada (pois nesse caso teríamos as imagens de $\mathbf{i}$ e $\mathbf{j}$ aproximadamente colineares!).
Este raciocínio também indica que o número de condicionamento (cf. [4]) de A é próximo

de 1, nos deixando definitivamente convencidos de que a falha cabe ao algoritmo, não aos dados.
Vamos reescrever o sistema (1) na forma

$$\left[\begin{array}{rcrcr} 100x & - & y & = & 99 \\ x & - & 100y & = & -99 \end{array}\right] . \tag{3}$$

Agora, o procedimento acima (de triangularização de Gauss) – trocando, é claro, $-\frac{100}{1}$ por $-\frac{1}{100}$, – conduz a

$$\left.\begin{array}{ll} 1 - \frac{1}{100} \times 100 & = 0 \\ -100 - \frac{1}{100} \times (-1) = -1 \times 10^2 + 10^{-2} & = -100 \\ -99 - \frac{1}{100} \times (99) = -9,9 \times 10^1 - 0,099 \times 10^1 & = -99 \end{array}\right] . \tag{4}$$

Desta vez, a segunda equação de (3) se transforma em

$$-100y = -99 \qquad \Rightarrow \qquad y = 0,99$$

e a primeira, que – também desta vez – não se altera, nos dá

$$100x - 0,99 = 99 \;\; \Rightarrow \;\; x = \frac{99}{100} \qquad \Rightarrow \qquad x = 0,99 \;\; .$$

Chegamos então a um erro de 1% com relação à solução exata do sistema. (Mas convém observar que na realidade temos um erro relativo de $0,01\%$, comparando os dados de *saída* ao maior valor dos dados de *entrada*.)

4 Conclusão

Se na primeira equação de (1) fosse nulo o coeficiente de x , ou seja, se tivéssemos $a_{11} = 0$, o procedimento que conduzimos – esquema de triangularização de Gauss (sem pivoteamento) – seria bloqueado, já que esse coeficiente comparece no denominador (multiplicando a segunda linha do sistema). O algoritmo de triangularização de Gauss, para matrizes de ordem arbitrária, quando encontra um elemento na diagonal (*pivot*) que seja nulo, efetua uma troca de linhas, de forma a poder calcular a divisão que define os multiplicadores correspondentes a esse passo.
Assim, ao trocarmos a posição das linhas, passando de (1) para (3), estávamos obedecendo à seguinte *regra de ouro do Cálculo Numérico*: "se um certo valor, calculado exatamente, traz dificuldades, ou não é factível, valores próximos a ele também vão trazer dificuldades, em cálculos aproximados". O *pivot* da primeira linha no sistema (1) não é nulo, mas é relativamente muito pequeno, em termos da ordem de precisão com que trabalhamos.
A técnica chamada de *pivoteamento* consiste em escolher justamente o maior elemento possível para ocupar o papel de *pivot*, ou selecionando entre aqueles da mesma linha (*pivoteamento parcial*), ou entre todos que ainda estão sujeitos a modificações (*pivoteamento total*).

Este procedimento, como ocorreu com o exemplo exposto, tornará o valor dos multiplicadores associados a esse passo o maior possível, de forma que os erros introduzidos com as aproximações – originadas da reta de vírgula flutuante sendo utilizada – serão minimizados. Trata-se de um procedimento **absolutamente indispensável**, principalmente quando lidamos com matrizes de ordem elevada, e com valores sobre a grandeza dos quais não temos controle, o que em geral ocorre no caso de problemas práticos: os dados de entrada de um dado sistema são gerados em outra parte do programa, sem nenhum acesso direto da nossa parte.
Conseqüências inesperadas, algumas lamentáveis, de erros originados no (mau) tratamento numérico de diferentes problemas em aplicações reais – tecnologia, ciência, bolsa de valores, e até eleições – podem ser consultadas em

`http://www.ic.uff.br/~demoura/2-99/dados.htm#links`

5 Exercícios

- Construir exemplos equivalentes ao que discutimos, para retas de vírgula flutuante $\Re$, mas com número arbitrário de dígitos.
- Justificar com rigor as passagens acima onde a expressão "quase" foi tomada como suficiente para os argumentos.
- Observe que, nas operações com a reta $\Re$ efetuadas acima, foi sempre usado o **cancelamento** – "*chopping* ", em inglês; deduzir que o mesmo desastre ocorreria, com dados ligeiramente modificados, se usássemos o **arredondamento**, ou "*rounding*".

References

[1] J.E. Boldrini, S.R.I. Costa, V.L.F.F. Ribeiro, and H. G. Wetzler. *Álgebra Linear*. Ed. Harbra, S. Paulo, 1980.

[2] C.A. de Moura. *Apontamentos de Álgebra Linear Computacional*. Un. de Brasília, Monografias de Matemática – Ensino e Extensão, Vol. 2, Brasília, DF, 1984. 131+vi pp.

[3] G.E. Forsythe and C.B. Moller. *Computer Solutions of Linear Algebraic Systems*. Prentice Hall, Engl. Cliffs, NJ, EUA, 1967.

[4] G. Golub and J.M. Ortega. *Scientific Computing – an Introduction with Parallel Computing*. Academic Press, San Diego, CA, EUA, 1993.

[5] Márcia A. G. Ruggiero and Vera L. da R. Lopes. *Cálculo Numérico – Aspectos Numéricos e Computacionais*. McGraw Hill, S. Paulo, SP, 1998. 2^a. Edição.