UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Trabalho 1
Decomposição LU

Vinicius de Freitas Reis

Cálculo Numérico I

São Carlos - SP
Outubro/2008

1 Introdução

Vários sistemas do mundo real podem ser resolvidos através da análise linear, como: determinação do potencial em redes elétricas, cálculo da tensão na estrutura metálica da construção civil, cálculo da razão de escoamento num sistema hidráulico com derivações. O problema matemático, em todos estes casos, se reduz ao problema de resolver um sistema de equações simultâneas.

Uma equação é dita linear se cada termo que a compõe contém não mais do que uma variável e cada variável aparece na primeira potência. A solução de um conjunto de equações é muito mais difícil quando as equações são não lineares. Entretanto, a maioria das aplicações envolve apenas equações lineares, o que nos motiva o seu estudo.

Consideremos n equações lineares com n incógnitas. Vamos nos referir a elas como *Sistema Linear de Ordem* n. De forma geral, um sistema linear é descrito como:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ a_{31}x_1 + a_{32}x_2 + \cdots + a_{3n}x_n & = & b_3 \\ \dots\dots\dots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{cases}$$

Ou, na forma matricial, como:

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \dots & \dots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{pmatrix} \Leftrightarrow Ax = b$$

Definição 1. Um sistema linear de ordem n é triangular inferior, se tiver a forma:

$$\begin{cases} a_{11}x_1 & = & b_1 \\ a_{21}x_1 + a_{22}x_2 & = & b_2 \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 & = & b_3 \\ \dots\dots\dots & & \vdots \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n & = & b_n \end{cases}$$

Definição 2. Um sistema linear de ordem n é triangular superior, se tiver a forma:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n & = & b_1 \\ a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n & = & b_2 \\ a_{33}x_3 + \cdots + a_{3n}x_n & = & b_3 \\ \dots\dots & = & \vdots \\ a_{nn}x_n & = & b_n \end{cases}$$

A solução de um sistema triangular inferior é obtida por substituição direta, isto é, determinamos o valor de x_1 na primeira equação, substituímos esse valor na segunda equação e determinamos o valor de x_2 e assim por diante. Da mesma forma podemos proceder para sistemas triangulares superiores. Algebricamente, temos:

$$\begin{cases} x_1 & = & \dfrac{b_1}{a_{11}} \\ \\ x_i & = & \left(b_i - \displaystyle\sum_{j=1}^{i-1} a_{ij} x_j \right) / a_{ii} \end{cases} \tag{1}$$

Ora, se conseguirmos um método que transforme um sistema qualquer em um sistema triangular, teremos uma forma eficiente de resolução de qualquer sistema.

2 A Decomposição LU

O método da decomposição LU baseia-se na decomposição da matriz de um sistema qualquer em duas matrizes, sendo que uma é triangular inferior (L) e outra é triangular superior (U). O teorema a seguir nos dá as condições de aplicabilidade do método.

Teorema LU. Seja $A = (a_{ij})$ uma matriz quadrada de ordem n e A_k os menores principais, constituídos das k primeiras linhas e k primeiras colunas de A. Se $det(A_k) \neq 0 \ \forall k \in 0, 1, 2, \ldots, n-1$, então existe uma única matriz triangular inferior $L = (\ell_{ij})$, com $\ell_{11} = \ell_{12} = \ldots = \ell nn = 1$, e uma única matriz triangular superior U tal que $LU = A$.

Satisfeitas as condições da decomposição, a resolução do sistema torna-se simples. Se $A = LU$ e $Ax = b$, então $LUx = b$. Esse último sistema pode ser resolvido em duas etapas: $Ux = y$ e $Ly = b$. Como L e U são matrizes triangulares, a resolução destes últimos é extremamente simples e, a partir deles, temos o vetor x, solução inicialmente procurada.

3 O Algoritmo

Já temos as condições em que o método pode ser aplicado e também a forma, a partir das matrizes L e U, de resolver o sistema. Só nos resta discutir um algoritmo capaz de realizar a decomposição.

Podemos calcular $L = (\ell_{ij})$ e $U = (u_{ij})$ simplesmente aplicando a definição de produto e igualdade de matrizes, ou seja, impondo que a matriz A seja igual a LU. A partir dessa constatação, nota-se que:

$$
\begin{cases} u_{ij} & = a_{ij} - \displaystyle\sum_{k=1}^{i-1} \ell_{ik} u_{kj}, & i \leq j, \\ & \\ \ell_{ij} & = \left(a_{ij} - \displaystyle\sum_{k=1}^{j-1} \ell_{ik} u_{kj} \right) / u_{jj}, & i > j. \end{cases} \tag{2}
$$

Portanto, basta um algoritmo que percorra a matriz A, atualizando os valores correspondentes de L e U de acordo com as equações 2. Feito isso, temos a decomposição da matriz em LU e resta apenas resolver os sistemas triangulares $Ly = b$ e $Ux = y$, o que pode ser feito através das equações 1.

Bibliografia

Franco, Neide Bertoldi - Cálculo Numérico - São Paulo: Pearson Prentice Hall, 2006.

4 Implementação

O algoritmo foi implementado na linguagem C++, no compilador gcc 4.1.2 e foi testado em ambiente linux (apesar de ser independente de plataforma). A interface para entrada dos dados foi extremamente minimalista, visando simplificar o processo e aumentar a interoperabilidade do projeto. Um arquivo com os dados de entrada deve ser criado, no formato a seguir:

```
n

a11 a12 a13 ... a1n
a21 a22 a23 ... a2n
... ... ...     ...
an1 an2 an3 ... ann

b1
b2
...
bn
```

Onde n é a dimensão da matriz dos coeficientes, aij são os elementos da matriz dos coeficientes e bi são os termos independentes do sistema. Na verdade, estes dados são lidos da entrada padrão (teclado), mas a recomendação é que, para maior comodidade, utilize-se o redirecionamento da entrada para um arquivo (como indicado no exemplo).

5 Exemplo

Um arquivo de exemplo de entrada (entrada.txt) foi incluído, visando exemplificar o formato de entrada e o modo de utilização do programa. Compilação:

```
~ $ g++ -o lu lu.cpp
```

Para rodar o programa, em Linux:

```
~ $./lu < entrada.txt > saida.txt
```

Em Windows:

```
C:\> lu.exe < entrada.txt > saida.txt
```

Após o término da execução, o arquivo saida.txt deverá conter os resultados da resolução (matrizes L, U e vetor x).