1 - Introdução

Uma grande gama de problemas pode ser resolvida através da análise linear, tais como, determinação do potencial em redes elétricas, cálculo da tensão na estrutura metálica, cálculo da razão de escoamento num sistema hidráulico com derivações, previsão da concentração de reagentes sujeitos à reações químicas simultâneas. O problema matemático em todos estes casos se reduz ao problema de resolver um sistema de equações. Neste trabalho, como proposto, apenas serão considerados métodos para a resolução de sistemas de equações lineares.

Neste trabalho apresentaremos a resolução do problema proposto pelo método de decomposição LU com pivotamento parcial.

O algoritmo aqui apresentado foi implementado em linguagem C/C++ e compilado com o "GNU GCC Compiler" junto a ferramenta Code::Blocks. As entradas do algoritmo devem ser respeitadas conforme indicado no inicio do algoritmo.

2 – Método de decomposição LU

Primeiramente vejamos o teorema da decomposição LU

Teorema 1 - *Seja A uma matriz quadrada de ordem n, e A_k o menor principal, constituído das k primeiras linhas e k primeiras colunas de A. Assumimos que $det(A_k) \neq 0$ para $k = 1, 2, \ldots, n - 1$. Então existe uma única matriz L triangular inferior com diagonal unitária e uma única matriz U triangular superior tal que A = LU.*

Observação: Note que det(A) = det(LU) = det(L)*det(U) = det(U).

Note que L possui inversa pois det(L) = 1 ≠ 0. Chamaremos a inversa de L de M

Veja que assumirmos $\det(A_k) \neq 0$ para k = 1, 2, . . . , n – 1 é uma restrição muito restritiva. Podemos reformular este teorema da seguinte maneira:

Teorema 2 - *Seja A uma matriz quadrada de ordem n. Assumimos que det(A) ≠ 0. Então existe uma única matriz L triangular inferior com diagonal unitária, uma única matriz U triangular superior e uma matriz P de permutação tal que PA = LU.*

Este segundo teorema, além de eliminar restrições na execução do método, ainda reduz os erros na solução numérica do problema. Veremos nesta seção como se aplica um método derivado do teorema 2.

Decomposição LU

Primeiramente vamos definir o que é um pivô

Seja: $A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$ ao fazer um passo da eliminação de gauss nesta matriz, temos que $A^1 = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^1 & \cdots & a_{2n}^1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^1 & \cdots & a_{nn}^1 \end{pmatrix}$ neste caso a_{11} foi o pivô.

No k-ésimo passo teríamos:

$$A^k = \begin{pmatrix} a_{11} & a_{12} & & \cdots & & a_{1n} \\ 0 & a_{22}^1 & & \cdots & & a_{2n}^1 \\ \vdots & \vdots & \ddots & & & \vdots \\ 0 & 0 & \cdots & a_{k+1,k+1}^k & \cdots & a_{k+1,n}^k \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 0 & a_{n2}^1 & \cdots & a_{n,k+1}^k & \cdots & a_{nn}^k \end{pmatrix}$$

Em que $a_{k-1,k-1}^k$ seria o pivô do k-ésimo passo. Ou seja, o pivô é o elemento da diagonal que é utilizado para zerar os elementos de sua coluna abaixo dele.

Utilizaremos aqui a expressão pivotamento de linhas para nos referirmos a troca de duas linhas mudando assim o pivô.

Com a noção de pivô fica fácil definirmos a eliminação de gauss com pivotamento parcial.

Abaixo descrevemos como obter os elementos da matriz L e da matriz U. Durante o processo cometemos alguns abusos de notação para facilitar o entendimento.

Devemos calcular os elementos das linhas de U e os elementos da colunas de L na seguinte ordem:

1ª linha de U: Faça inicialmente o pivotamento de linhas na matriz MA que inicialmente é igual a A de forma que o modulo de a_{11} seja maior que modulo a_{k1}, k=1,...,n (sempre que pivotamos linhas da matriz A pivotamos as linhas do vetor b). Um ponto que devemos deixar claro aqui é que na pratica a matriz U é calculada sobre a matriz A portanto durante todo processo a seguir mostrado, não é necessário o calculo da matriz MA pois MA=U. Manteremos índices de A como a_{ij} e de U como u_{ij} para facilitar a compreensão, porem vale lembrar que no final do processo $a_{ij} = u_{ij}$.

Agora, fazendo o produto da 1ª linha de L por todas as colunas de U e igualando com os elementos da 1ª linha de A, obtemos ,

$$
\begin{gathered}
1 * u_{11} = a_{11} \Rightarrow u_{11} = a_{11} \\
1 * u_{12} = a_{12} \Rightarrow u_{12} = a_{12} \\
\vdots \\
1 * u_{1n} = a_{1n} \Rightarrow u_{1n} = a_{1n}
\end{gathered}
$$

$$\Longrightarrow u_{1j} = a_{1j} \, , j = 1, \dots, n$$

1ª coluna de L: Fazendo o produto de todas as linhas de L, (da 2ª até a nª), pela 1ª coluna de U e igualando com os elementos da 1ª coluna de A, (abaixo da diagonal principal), obtemos,

$$
\begin{gathered}
l_{21} * u_{11} = a_{21} \Rightarrow l_{21} = \frac{a_{21}}{u_{11}} \\
l_{31} * u_{11} = a_{31} \Rightarrow l_{31} = \frac{a_{31}}{u_{11}} \\
\vdots \\
l_{n1} * u_{11} = a_{n1} \Rightarrow l_{n1} = \frac{a_{n1}}{u_{11}}
\end{gathered}
$$

$$\Longrightarrow l_{j1} = \frac{a_{j1}}{u_{11}} \, , j = 2, \dots, n$$

2ª linha de U: Faça novamente o pivotamento de linhas na matriz MA de forma que a_{22} seja maior que a_{k2}, k=2, ..., n (Lembrando que M é a inversa de L). Agora, Fazendo o produto da 2ª linha de L por todas as colunas de U, (da 2ª até a nª), e igualando com os elementos da 2ª linha de A, (da diagonal principal em diante), obtemos,

$$
\begin{gathered}
l_{21} * u_{12} + u_{22} = a_{22} \Rightarrow u_{22} = a_{22} - l_{21} * u_{12} \\
l_{21} * u_{13} + u_{23} = a_{23} \Rightarrow u_{23} = a_{23} - l_{21} * u_{13} \\
\vdots \\
l_{21} * u_{1n} + u_{2n} = a_{2n} \Rightarrow u_{2n} = a_{2n} - l_{21} * u_{1n}
\end{gathered}
$$

$$\Longrightarrow u_{2j} = a_{2j} - l_{21} * u_{1j} \, , j = 2, \dots, n$$

2ª coluna de L: Fazendo o produto de todas as linhas de L (da 3ª até a nª) pela 2ª coluna de U e igualando com os elementos da 2ª coluna de A, (abaixo da diagonal principal), obtemos,

$$l_{31} * u_{12} + l_{32} * u_{22} = a_{32} \Rightarrow l_{32} = \frac{a_{32} - l_{31} * u_{12}}{u_{22}}$$
$$l_{41} * u_{12} + l_{42} * u_{22} = a_{42} \Rightarrow l_{42} = \frac{a_{42} - l_{41} * u_{12}}{u_{22}}$$
$$\vdots$$
$$l_{n1} * u_{12} + l_{n2} * u_{22} = a_{n2} \Rightarrow l_{n2} = \frac{a_{n2} - l_{n1} * u_{12}}{u_{22}}$$
$$\Longrightarrow l_{j2} = \frac{a_{j2} - l_{j1} * u_{12}}{u_{22}}, j = 3, \dots, n$$

Se continuarmos calculando 3ª linha de U, 3ª coluna de L, 4ª linha de U, 4ª coluna de L, etc..., teremos as fórmulas gerais:

$$\begin{cases} u_{ij} = a_{ij} - \sum_{k=1}^{i-1} l_{ik} u_{kj} & i \leq j \\ u_{ij} = 0 & i < j \\ l_{ij} = \dfrac{\left(a_{ij} - \sum_{k=1}^{j-1} l_{ik} u_{kj}\right)}{u_{jj}} & i > j \\ l_{ij} = 1 & i = j \\ l_{ij} = 0 & i < j \end{cases}$$

Lembrando que a cada passo devemos fazer o pivoteamento de linhas da matriz MA e que para não haver a necessidade do calculo da matriz M (que não é complexo) calculamos a matriz U sobre a própria matriz A. Ou seja, vamos atualizando a Matriz A a cada passo afim de que se torne triangular superior.

Resolução de sistema triangular inferior

Considere o sistema abaixo

$$Ax = b \Longrightarrow \begin{pmatrix} a_{11} & & \cdots & 0 \\ a_{21} & a_{22} & & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$$

Podemos obter a solução para este problema fazendo

$$x_1 = \frac{b_1}{a_{11}}$$

$$x_2 = \frac{b_2 - a_{21}x_1}{a_{22}}$$

Não é difícil ver que:

$$x_j = \frac{b_j - \sum_{i=1}^{j-1} a_{ji}x_i}{a_{jj}} \text{ para } j = 1, ..., n$$

O procedimento análogo pode ser feito para a obtenção da solução de sistemas cuja a matriz é triangular superior.

3 – Algoritmos

Algoritmo 1: Decomposição LU com pivotamento

```
Para k = 1 até n-1
      l={j | mod(a[k][j]) >= mod(a[k][i]), i=k até n}
      para j = k até n
            aux = a[k][j];
            a[k][j] = a[k][l];
            a[k][l] = aux;
      fim para
      para i = k+1 até n
            m = a[i][k]/a[k][k];
            para j = k+1 até n
                  a[i][j] = a[i][j] – m*a[k][j]
            fim para
      fim para
fim para
```

Algoritmo 2: Sistema triangular inferior

```
Para i = 1 até n
      x[i]=b[i]
      Para j=1 até i
            x[i] = (x[i]-a[i][j]*x[j])/a[i][i];
      fim para
fim para
```

Algoritmo 3: Sistema triangular superior

```
Para i = n até 1
      x[i]=b[i]
      Para j=n até i
            x[i] = (x[i]-a[i][j]*x[j])/a[i][i];
      fim para
fim para
```

4 - Resolução do problema

Dado um problema da forma Ax = b. Seja P uma matriz de permutação, e então note que:

Ax = b ➔ PAx = Pb

Fazendo a decomposição PA=LU como no algoritmo 1 obtemos,

PAx = LUx = Pb

Note agora que fazendo Ux = y temos um que resolver um sistema da forma Ly=Pb que é triangular inferior e podemos resolve-lo pelo algoritmo 2 obtendo assim o vetor y.

Para encontrar x basta resolvermos o sistema Ux=y, como é um sistema triangular superior pode ser resolvido pelo algoritmo 3.

Portanto para o problema proposto realizamos os cálculos desta mesma maneira.

A patir da matriz A abaixo

2.000	1.000	7.000	4.000	-3.000	-1.000	4.000	4.000	7.000	0.000
4.000	2.000	2.000	3.000	-2.000	0.000	3.000	3.000	4.000	1.000
3.000	4.000	4.000	2.000	1.000	-2.000	1.000	1.000	9.000	-3.000
9.000	3.000	5.000	1.000	0.000	5.000	6.000	-5.000	-3.000	4.000
2.000	0.000	7.000	0.000	-5.000	7.000	1.000	0.000	1.000	6.000
1.000	9.000	8.000	0.000	3.000	9.000	9.000	0.000	0.000	5.000
4.000	1.000	9.000	0.000	4.000	3.000	7.000	-4.000	1.000	3.000
6.000	3.000	1.000	1.000	6.000	8.000	3.000	3.000	0.000	2.000
6.000	5.000	0.000	-7.000	7.000	-7.000	6.000	2.000	-6.000	1.000
1.000	6.000	3.000	4.000	8.000	3.000	-5.000	0.000	-6.000	0.000

Obtivemos as matrizes L e U abaixo representadas com o vetor b permutado,

```
A matriz L eh:
1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
0.444   1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
0.333   0.391   1.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000
0.222   0.043   1.330   1.000   0.000   0.000   0.000   0.000   0.000   0.000
0.222   -0.087  1.479   0.059   1.000   0.000   0.000   0.000   0.000   0.000
0.111   0.217   0.147   0.754   0.712   1.000   0.000   0.000   0.000   0.000
0.444   -0.043  0.838   0.532   1.040   -1.914  1.000   0.000   0.000   0.000
0.667   0.130   0.049   0.251   1.724   -2.217  0.765   1.000   0.000   0.000
0.667   0.391   -0.446  -1.764  -1.149  -0.538  0.196   -0.535  1.000   0.000
0.111   0.739   -0.250  0.988   0.403   1.498   -0.406  0.862   -0.679  1.000

A matriz U eh:
9.000   3.000   5.000   1.000   0.000   5.000   6.000   -5.000  -3.000  4.000
0.000   7.667   5.778   -0.444  3.000   6.778   6.333   2.222   1.333   3.222
0.000   0.000   5.072   -0.159  2.826   -1.319  2.522   -3.203  1.478   0.406
0.000   0.000   0.000   4.009   -6.890  -0.651  -0.964  9.276   5.642   -1.569
0.000   0.000   0.000   0.000   4.486   4.467   -9.456  5.496   -7.736  -1.117
0.000   0.000   0.000   0.000   0.000   3.473   8.050   -7.368  1.083   2.774
0.000   0.000   0.000   0.000   0.000   0.000   27.253  -17.754 1.271   6.329
0.000   0.000   0.000   0.000   0.000   0.000   0.000   -8.360  19.098  1.521
0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   22.758  -9.727
0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   0.000   -4.230
```

O vetor b permutado (Pb) fica:

```
108.00
90.50
139.00
86.00
31.00
61.00
-43.50
45.00
52.20
66.50
```

Resolvendo o sistema Ly = b obtemos o valor de y, fazendo Ux = y obtemos a solução x do problema proposto, abaixo relacionamos y e x respectivamente:

108.00	16.19
42.50	-77.91
86.37	15.42
-54.76	-74.05
-113.83	6.66
149.49	17.62
271.64	14.22
296.83	29.41
-39.10	23.68
-251.38	59.42

Assim obtemos a solução do problema proposto.

5 – Conclusão

Resolução de sistemas lineares é muito utilizada em diversas áreas da matemática, engenharia, etc. Assim se faz necessário o uso de técnicas avançadas para resolução destes sistemas. Neste contexto a técnica de decomposição LU com pivotamento parcial torna-se uma poderosa ferramenta para a resolução destes problemas sendo um método rápido e que reduz erros gerados pelo computador em sua aplicação. A técnica foi aplicada ao problema proposto onde tivemos êxito em sua aplicação, obtendo um resultado confiável.