Projeto 1

Gabriel Belém Barbosa	RA: 234672
Diogo Batalha	RA: 169955
Felipe Kenzo Obata Piton	RA: 171108
Felipe Purini Policastro	RA: 215760

18 de Abril de 2021

Conteúdo

1 1.

O algoritmo foi implementado diretamente na linguagem Matlab. Para implementar um método de substituição regressiva de resolver um sistema linear $Ax = b$ com uma matriz A triangular superior quadrada nxn, primeiro se obtém o último termo da solução, $x_n = b_n/DIAG(n)$, que é diretamente obtível. Após isso, trabalhando na n-ésima coluna de A, para todos as entradas a_{in} do ENV, subtrai-se o produto de tal entrada com x_n da entrada b_i, ou seja com o índice da linha dessa entrada coincidindo com o índice de b. Para acessar corretamente as entradas em ENV, faz-se um for em p no intervalo [ENVcol(n) :ENVcol($n+1$) -1], que expressam os índices de ENV que contêm elementos da n-ésima coluna. Repare que em Matlab, exceto se especificado, o for não é iterado caso o segundo termo do intervalo seja menor que o primeiro, o que ocorre quando ENVcol(n) =ENVcol($n+1$), o que convenientemente coincide com nossa estrutura de dados, pois nesse caso é inferido que a coluna em questão não tem elementos em ENV.

Para associar a entrada ENV(p) ao seu respectivo b usa-se ENVlin(p), que retorna a linha na matriz A do elemento p de ENV. Após interados todos os p's, o mesmo processo é aplicado para a coluna $n-1$, para assim obter x_{n-1} que agora se resume a dividir b_{n-1} pelo elemento da diagonal principal, percorrer os elementos da coluna $n-1$ do envelope e subtrair o produto entre eles e x_{n-1} dos b's adequados, usando a indexação acima descrita. Para generalizar esse processo, cria-se um for que percorre as colunas de n a 1 porém em j neste caso.

Assim, dado um elemento j da diagonal principal (no passo j), todos os possíveis elementos à direita dele, intervalo $[j+1:n]$, foram obrigatoriamente eliminados por conta do processo acima descrito à essa altura.

```
for j=n:-1:1 #começa pela coluna n/x(n)/b(n)
    xj=b(j)/DIAG(j)
    x=[xj;x]
    for p=ENVcol(j):ENVcol(j+1)-1
        b(ENVlin(p))-=ENV(p)*xj
    end
end
```

2 2.

$$A = \begin{bmatrix} * & & & & & & \\ & * & & & * & & \\ & & * & & & & \\ & * & & * & & * & \\ & & & & * & & \\ & * & & & & * & * \\ & & & & & & * \end{bmatrix}$$

DIAG =

a_{11}	a_{22}	$\dots$	a_{77}

Seguindo as definições para os envelopes, temos para parte superior por colunas:

$\mathrm{ENV}_{\mathrm{sup}} =$

a_{25}	a_{35}	a_{45}	a_{46}	a_{56}	a_{67}	

$\mathrm{ENVcol}_{\mathrm{sup}} =$

1	1	1	1	1	4	6	7

Sendo $a_{35} = a_{45} = a_{56} = 0$.

$\mathrm{ENVlin}_{\mathrm{sup}} =$

2	3	4	4	5	6	

E para a parte inferior por linhas:

$\mathrm{ENV}_{\mathrm{inf}} =$

a_{42}	a_{43}	a_{62}	a_{63}	a_{64}	a_{65}	

$\mathrm{ENVlin}_{\mathrm{inf}} =$

1	1	1	1	3	3	7	7

Sendo $a_{43} = a_{63} = a_{64} = a_{65} = 0$.

$\mathrm{ENVcol}_{\mathrm{inf}} =$

2	3	2	3	4	5	

3 3.

Realizando a fatoração pelo método da eliminação gaussiana para obter $A \in \mathbb{R}^{n\times n}$ como produto LU, define-se $A^0 = A$ e L_p tal que $A^p = L_p A^{p-1}$, $1 \leq p \leq n-1$, onde A^p é a matriz A^{p-1} operada de tal forma a eliminar os termos abaixo da diagonal na p-ésima coluna, ou seja, L_p é a matriz de operações elementares que leva à eliminação dos termos da parte triangular inferior da matriz na p-ésima coluna. Dessa forma:

$$A = A^0 = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1} A^{n-1}$$

Sendo, então, $L = L_1^{-1} L_2^{-1} \dots L_{n-1}^{-1}$ e $U = A^{n-1}$. A^{n-1} é trangular superior pela forma que foi obtida (eliminação de gauss por colunas), porém L é triangular inferior, como provaremos à seguir.

Pela definição L_p acima, para eliminar os termos abaixo da diagonal na p-ésima coluna basta que se some à linha i de interesse a linha p vezes uma constante

$$c_{i,p} = -\frac{a^{p-1}_{i,p}}{a^{p-1}_{p,p}}$$

Dessa forma, temos:

$$A^p = \left[\begin{array}{cccc} a^{p-1}_{1,1} & \cdots & \cdots & a^{p-1}_{1,n} \\ & \ddots & & \vdots \\ & & a^{p-1}_{p,p} & \vdots \\ \mathbf{0} & & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} & \underbrace{\begin{bmatrix} a^p_{p+1,p+1} & \cdots & a^p_{p+1,n} \\ \vdots & \ddots & \vdots \\ a^p_{n,n-1} & \cdots & a^p_{n,n} \end{bmatrix}}_{A^p_{colateral}} \end{array}\right]$$

$$= \overbrace{\begin{bmatrix} 1 & & & & \\ & \ddots & & \mathbf{0} & \\ & & 1 & & \\ & & c_{p+1,p} & \ddots & \\ & & \vdots & & \\ & & c_{n,p} & & 1 \end{bmatrix}}^{L_p} \overbrace{\begin{bmatrix} a^{p-1}_{1,1} & \cdots & \cdots & \cdots & a^{p-1}_{1,n} \\ & \ddots & & & \vdots \\ & & a^{p-1}_{p,p} & & \vdots \\ \mathbf{0} & & a^{p-1}_{p+1,p} & \ddots & \vdots \\ & & \vdots & & \\ & & a^{p-1}_{n,p} & \cdots & a^{p-1}_{n,n} \end{bmatrix}}^{A^{p-1}}$$

Fica claro, pelo esquema acima, que L_{p+1} só desviará da matriz identidade na sua $(p+1)$-ésima coluna, composta de constantes da mesma forma que L_p. Analisemos L, produtório dos inversos de todas as matrizes L_p. Primeiramente, calcular as inversas de L_p é um caso de multiplicar as entradas fora da diagonal por -1. Em seguida, como as matrizes L_p^{-1} diferem da identidade somente por constantes que só aparecem em suas respectivas colunas p abaixo da diagonal principal e em mais nenhum lugar, o produtório entre elas tem o efeito de compor uma matriz com cada coluna p sendo exatamente a coluna p de L_p^{-1}. Portanto, segue que $l_{i,p} = -c_{i,p}$ (nos interessa que a esparcidade da p-ésima coluna de L é determinada pela esparcidade de L_p em sua p-ésima coluna). Se convencer deste último fato exige somente um pouco de resolução mecânica, e na literatura em geral L já é apresentada nessa forma, diretamente.

Repare também que $c_{i,p} = 0 \Leftrightarrow a^{p-1}_{i,p} = 0$. Sendo assim, os zeros de determinada linha de A entre a primeira coluna de A e o primeiro elemento não nulo daquela linha (se é que existe algum zero) são carregados para L imediatamente, estando então fora dos envelopes de ambas as matrizes. Então o primeiro elemento não nulo desta linha de A à esquerda da diagonal, se existir, criará um c e um $l \neq 0$ naquela mesma entrada em L, que analogamente à A entrará no envelope de L para aquela linha (o que garante que o envelope não expande). Mesmo que a operação de uma dada L_p zere uma entrada na submatriz $A^p_{colateral}$ (submatriz esta que é o limite de atuação de L_p, claramente vista acima) previamente não nula em A^{p-1} e abaixo da diagonal principal (entrada esta, definimos, $a^p_{i,j}$, com $n > i > j > p$), a entrada $l_{i,p}$ à esquerda desse possível ponto de contenção em L já não poderá ser nula (pois o $l_{i,p} = -c_{i,p} \neq 0$, para começo de conversa), e então tudo no intervalo $[p, i-1]$ entrará no envelope de L, zerado ou não $-c_{i,j} = l_{i,j}$ dentro dele (o que pode nem vir a ocorrer, se j não vem imediatamente depois de p no processo, porém isso é irrelevante). Portanto, o envelope também não diminui, terminando a prova.

Provar que a estrutura de envelope de U por colunas é equivalente à da parte superior de A por colunas é mais direto e visual, visto que no esquema acima fica claro que A^p difere de A^{p-1} em decorrência da eliminação na coluna p somente das colunas $p+1$ até n (final da matriz) da linha $p+1$ para baixo, simbolizado na representação de A^p acima pelo supra-índice p dos elementos de $A^p_{colateral}$ (submatriz que compreende a região possivelmente diferente) em contrapartida aos supra-índices $p-1$ herdados de A^{p-1} fora dessa região.

Precisamos focar somente nas colunas após $p+1$ e antes de n, pois essas são as quais são possivelmente afetadas também acima da diagonal principal, como fica claro no esquema. Assumindo que uma operação foi efetuada numa linha i no intervalo de interesse durante a obtenção de A^p (visto que se nenhuma operação ocorrer não nos importa uma vez que a linha em A^p será igual a de A^{p-1}), uma entrada de A^p, $a^p_{i,j}$, $p+1 < i, j < n$, só divergirá de sua correspondente em A^{p-1}, $a^{p-1}_{i,j}$, se e somente se $a^{p-1}_{p,j} \neq 0$ ($a^{p-1}_{p,j} \neq 0 \Leftrightarrow a^p_{i,j} \neq a^{p-1}_{i,j}$, $p+1 < i, j < n$). Isto é dizer que uma operação da linha p em i só afetará o elemento na coluna j da linha i ($a^{p-1}_{i,j}$) se o elemento acima dele na linha p ($a^{p-1}_{p,j}$) for não nulo. Como isso é verdade para cada passo do processo, o efeito é carregado, o que implica que não ocorrerão casos de uma entrada nula em U ocorrer sem que uma entrada acima dela seja não nula em A (i.e., um envelope não pode ser expandido), e pela contrapositiva, a primeira entrada não nula de uma coluna de A não pode ser nulificada durante o processo de obter U (um envelope não pode diminuir de tamanho). Segue, então, o que queríamos provar: as estruturas dos envelopes de U e da parte superior de A por colunas são equivalentes.

4 4.

Na passagem de uma matriz A^{p-1} para A^p na estratégia de pivoteamento parcial, multiplica-se antes A^{p-1} por uma matriz que permuta a linha p de A^{p-1} com a linha abaixo dela cujo elemento na coluna p é o maior em módulo, se necessário, garantindo, então $|a_{p,p}^{p-1}| \geq |a_{i,p}^{p-1}|$, $p \leq i \leq n$. Após isso L_p é aplicada sobre a A^{p-1} permutada normalmente. O processo pode ser condensado na forma $PA = LU$, onde PA é uma matriz que obedece a relação anteriormente descrita para os módulos das entradas de determinada coluna abaixo da diagonal e seus respectivos elementos. PA é bem comportada no sentido em que não elimina ou duplica linhas de A, uma vez que escolhida uma linha por possuir o pivô abaixo da diagonal principal para uma coluna e esta ser fixada de tal forma a colocar tal pivô na diagonal principal, tal linha não será mais candidata para futuros pivoteamentos nos próximos passos, assim P preserva a não singularidade de A, já que tais operações só mudam o sinal do determinante. Tal fato resulta em um teorema de que toda matriz $A \in \mathbb{R}^{n\times n}$ não singular pode ser fatorada na forma $PA = LU$, na qual L é triangular inferior com diagonal unitária e em que U é triangular superior.

5 5.

Sabemos, como provado no item 3, que a estrutura de envelope de L (por linhas) é igual à da parte triangular inferior de A e que o mesmo vale para U e a triangular superior de A, por colunas. De forma similar, provamos no item 4 que esse resultado também se aplica à PA, matriz A após a aplicação de pivoteamento parcial. Assim, vamos propor um possível algoritmo para a fatoração LU de PA e sua consequente resolução, reaproveitando, para economizar memória, as estruturas de PA superior e inferior para U e L, respectivamente. A DIAG_L não foi feita pois é unitária e, portanto, desnecessária por não guardar informação verdadeiramente, enquanto DIAG_U será gravada na própria DIAG de PA, pelo mesmo motivo anterior. Definimos para fins de explicar a solução:

- **ENV$_{sup}$:**envelope da parte triangular superior de PA por colunas
- **ENVcol$_{sup}$:**ENVcol de ENV$_{sup}$
- **ENVlin$_{sup}$:**ENVlin de ENV$_{sup}$
- **DIAG:**envelope dos elementos da diagonal de PA
- **ENV$_{inf}$:**envelope da parte triangular inferior de PA por linhas
- **ENVlin$_{inf}$:**ENVlin de ENV$_{inf}$

- **ENVcol$_{inf}$:**ENVcol de ENV$_{inf}$
- **n:**tamanho de A

Seguindo o processo descrito no item 3 para a decomposição LU, usando nomenclatura condizente para as constantes envolvidas, no primeiro laço em j, percorrendo o intervalo $[1 : n-1]$, define-se o elemento da diagonal que estamos trabalhando na fatoração. O próximo laço, em p, no intervalo de 1 ao tamanho ENVcol$_{inf}$, percorre esta última à procura de elementos iguais a j, ou seja, os elementos de ENV$_{inf}$ associados à esses p's estão na coluna do elemento da diagonal que estamos trabalhando. Fazemos isso pois esses são o elementos à serem eliminados nesse passo. Se tais elementos, ENV$_{inf}(p)$, são não nulos (para economizar operações), dividimo-los pelo elemento na posição j de DIAG e armazenamos o negativo desse valor em c para futuras contas. Após isso, substituímos o valor na posição p do envelope inferior por $-c$ (relembrando que iremos construir L e U com base nos vetores atuais). Definimos então uma variável auxiliar $linha$ como 2 e, no terceiro laço, enquanto ENVlin$_{inf}$ no índice $linha$+1 for menor ou igual a p, adiciona-se 1 ao valor de $linha$ para que possamos obter a $linha$ do elemento ENV$_{inf}(p)$. Repare que $linha$ pode começar em 2, uma vez que o primeiro e segundo elementos de ENVlin$_{inf}$ são sempre 1 (por definição, no caso do primeiro elemento, e por falta de opção, no caso do segundo), e portanto são sempre menores ou iguais à qualquer p. Pensando em termos da decomposição, só eliminamos elementos abaixo da primeira linha ($linha$ 2 em diante), e então começar com $linha$ igual à 2 implica que, pela comparação estabelecida no loop, o mínimo valor de $linha$ possível será o próprio 2, caso ENVlin$_{inf}(3) > p$, o que coincide com o método.

No quarto laço, em x, de 1 até o comprimento de ENVlin$_{sup}$, buscamos agora elementos na mesma linha do elemento da diagonal que estamos trabalhando, de forma análoga ao que fizemos para as colunas da parte inferior acima. Fazemos isso pois esses são os elementos que serão multiplicados por c e somados às linhas que estamos eliminando. Note que quando mudamos de triangular inferior para o superior, ENVcol da inferior age de forma equivalente à ENVlin da superior, assim por dizer, e a mesma relação vale para ENVlin da inferior e ENVcol da superior. Novamente, só elementos não nulos importam. Achado tal elemento, um processo análogo ao usado para determinar a variável $linha$ é empregado para obter a variável $coluna$, coluna de tal elemento da parte superior que mora na linha j.

Finalmente, fazemos uma sequência de condicionais com o intuito de determinar se o elemento ENVsup(x) estará atuando acima, abaixo ou na própria diagonal, dependendo das variáveis $linha$ e $coluna$ previamente encontradas. Caso $linha > coluna$, a soma será na parte inferior, $coluna - j$ (distância horizontal de ENVsup(x) até o elemento da diagonal do passo) elementos após p (índice do elemento que foi eliminado, que mora na coluna j). Caso $coluna > linha$, a soma será na parte superior, $linha - j$ (distância vertical de ENVinf(p) até o elemento da diagonal do passo) elementos após x (índice do elemento que vai ser somado, que mora na linha j). O último caso, portanto, é $linha = coluna$, no qual a soma cairá na prórpria diagonal, mais especificamente em DIAG$(linha)$ ou DIAG$(coluna)$ (obviamente não faz diferença qual dos dois).

O algoritmo generelizado final para a decomposição LU é da forma:

```
Para j de 1 até (n-1) faça
   Para p de 1 até (comprimento de ENVcolinf) faça
      Se ENVcolinf(p) for igual a j e ENVinf(p) for diferente de 0 faça
         c = -ENVinf(p)/DIAG(j)
         ENVinf(p) = -c
         linha = 2
         Enquanto ENVlininf(linha+1) for menor ou igual a p
            linha = linha + 1
         fim[Enquanto]
         Para x de 1 até (comprimento de ENVlinsup) faça
            Se ENVlinsup(x) for igual a j e ENVsup(x) for diferente de 0 faça
               coluna = 2
               Enquanto ENVcolsup(coluna+1) for menor ou igual a x faça
                  coluna = coluna + 1
               fim[enquanto]
               Se linha for maior do que coluna faça
                  ENVinf(p+coluna-j) = ENVinf(p+coluna-j) + c*ENVsup(x)
               Senão, se linha for menor do que coluna faça
                  ENVinf(x+linha-j) = ENVinf(x+linha-j) + c*ENVsup(x)
               Senão faça
                  DIAG(coluna) = DIAG(coluna) + c*ENVsup(x)
               fim[Se]
            fim[Se]
         fim[x]
      fim[Se]
   fim[p]
fim[j]
```

Sendo assim, podemos seguir para a resolução do sistema linear de fato, que pode ser quebrada na resolução de dois sistemas, $Ly = Pb$ e $Ux = y$, para obter $PAx = Pb$. Começamos por $Ly = Pb$.

Primeiramente definimos um laço para j no intervalo $[1 : n]$, que percorrerá as linhas de L e montará o vetor y, que foi inicializado vazio. No caso estamos fazendo substituição progressiva ao invés de regressiva (que foi feita no item 1).

Primeiramente se obtém o termo da solução $y_j = Pb(j)/DIAG_L(j)$, processo o qual garantiremos ser direto através da eliminação dos termos à esquerda de $DIAG_L(j)$ mais à frente (para $j = 1$ esse problema não existe). Como definido anteriormente que DIAG_L é unitária, podemos simplesmente adicionar os elementos de Pb no índice j ao vetor y nesse passo, sem a necessidade da divisão.

No segundo laço, para p, percorremos o intervalo de 1 até o comprimento de ENVcol_{inf} à procura de elementos deste último iguais a j como no algoritmo anterior, ou seja, os elementos de ENV_{inf} associados a esses p's estão na coluna do elemento da diagonal em que estamos trabalhando. Caso o elemento $\mathrm{ENV}_{inf}(p)$ no índice p seja não nulo (novamente para economizar operações), definimos uma variável auxiliar $linha$ começando em 2, e, com o processo de obtenção de $linha$ idêntico ao do algoritmo anterior, encontramos a linha do elemento da coluna j com o propósito de determinar de qual elemento de Pb ($Pb(linha)$) ele multiplicado por y_j será subtraído. Terminado isso, efetivamente garantimos que no próximo passo, $j+1$, poderá se obter y_{j+1} diretamente. O algoritmo geral final para $Ly = Pb$ é da forma:

```
Para j de 1 até n faça
   yj = Pb(j)
   adicionar yj a y
   Para p de 1 até (comprimento de ENVcolinf) faça
      Se ENVcolinf(p) for igual a j e ENVinf for diferente de 0 faça
         linha = 2
         Enquanto ENVlininf(linha+1) for menor ou igual a p faça
            linha = linha + 1
         fim[Enquanto]
         Pb(linha) = Pb(linha) - ENVinf(p)*yj
      fim[Se]
   fim[p]
fim[j]
```

Prosseguimos então para $Ux = y$, sendo U triangular superior com envelope por colunas. Note que esse processo foi resolvido e explicado em detalhes no item 1, para $Ux = b$. A única diferença da forma generalizada é a presença de uma condicional para que o elemento j de ENVcol_{sup} seja diferente do elemento seguinte. Isso nos garante, pela definição de envelope, que há ao menos um elemento na coluna j da matriz U, algo que anteriormente era redundante para MATLAB, uma vez que loops com intervalos $[e,d]$ com $d < e$ são desconsiderados. O algoritmo generelizado final para $Ux = y$ é da forma:

```
Para j de n até 1 decrescentemente
```

```
   xj = y(j)/DIAG(j)
   Adicionar xj ao inicio de x
   Se ENVcolsup(j) é diferente de ENVcolsup(j+1)
      Para p de ENVcolsup(j) até ENVcolsup(j+1)-1 faça
         y(ENVlinsup(p)) = y(ENVlinsup(p)) - ENVsup(p)*xj
      fim[p]
   fim[Se]
fim[j]
```

6 6.

Das informações do problema

$$Pb = [0,0,10,0,0,0,0,0,15,0,20,0,0]$$

A dimensão de PA, $n = 13$, e PA, após ser estruturada na forma de envelope, é

- **ENV**$_{sup}$ **=**[-1,0,0,α,0,0,α,-1,0,0,0,-1,0,0,-α,0,α,-1,0,0,0,-α,0,0,0,-1,0,0]
- **ENVcol**$_{sup}$ **=**[1,1,1,1,4,8,12,12,15,18,22,22,26,29]
- **ENVlin**$_{sup}$ **=**[1,2,3,1,2,3,4,2,3,4,5,5,6,7,6,7,8,6,7,8,9,8,9,10,11,10,11,12]
- **DIAG =**[α,1,1,0,0,1,1,1,α,1,1,α,1]
- **ENV**$_{inf}$ **=**[α,0,1,1,α,α,0,1,0,α,0,1,-α]
- **ENVlin**$_{inf}$ **=**[1,1,1,1,4,5,6,6,6,10,10,10,13,14]
- **ENVcol**$_{inf}$ **=**[1,2,3,4,5,5,6,7,8,9,10,11,12]

Onde $\alpha = \frac{1}{\sqrt{2}}$. Utilizando a implementação desenvolvida no item 5, cujo algoritmo em MATLAB está nos anexos, obtemos a solução para $PAx = Pb$

$x = [-28.2843, -30, 10, -30, 14.1421, -30, 0, -30, 7.0711, -25, 20, -35.3553, -25]^T$

Ou

$$x = [-20\sqrt{2}, -30, 10, -30, 10\sqrt{2}, -30, 0, -30, 5\sqrt{2}, -25, 20, -25\sqrt{2}, -25]^T$$

7 Referências

- **C. B. Moler**, Numerical Computing with MATLAB, Philadelphia: SIAM, 2004.
- **D. S. Watkins**, Fundamentals of Matrix Computations, New Jersey: John Wiley & Sons, 2 ed., 2002.