

Curso: Métodos Numéricos II
Professor: Creto Augusto Vidal
Semestre: 2021.1
Aula # 20

1. Objetivo: Métodos que usam transformações de similaridade.

Nesta Aula, vamos apresentar um método que usa o conceito de transformação de similaridade: o Método de **Householder**. Dois outros métodos de transformação de similaridade, o de **Jacobi** e o Método **QR**, serão apresentados nas aulas #21 e #22.

2. Transformações de Similaridade

2.1 Definição.

Dadas uma matriz **A** e uma matriz **P** que tenha uma inversa \mathbf{P}^{-1} , podemos construir uma nova matriz a partir do **produto** dessas três matrizes e chamá-la de $\bar{\mathbf{A}}$, isto é,

$$(1) \quad \bar{\mathbf{A}} = \mathbf{P}^{-1}\mathbf{A}\mathbf{P}.$$

Agora suponha que **conseguimos achar um par (autovalor, autovetor correspondente) da matriz $\bar{\mathbf{A}}$** , de maneira que a seguinte equação seja satisfeita

$$(2) \quad \bar{\mathbf{A}}\mathbf{v}_i = \lambda_i\mathbf{v}_i.$$

Vamos substituir $\bar{\mathbf{A}}$, como definido na equação (1), na equação (2). Assim temos

$$(3) \quad (\mathbf{P}^{-1}\mathbf{A}\mathbf{P})\mathbf{v}_i = \lambda_i\mathbf{v}_i.$$

Vamos multiplicar os dois lados da equação (3) pela matriz **P** e usar as propriedades associativas da multiplicação de matrizes, isto é,

$$(4) \quad \mathbf{P}(\mathbf{P}^{-1}\mathbf{A}\mathbf{P})\mathbf{v}_i = \lambda_i\mathbf{P}\mathbf{v}_i \Rightarrow (\mathbf{P}\mathbf{P}^{-1})\mathbf{A}(\mathbf{P}\mathbf{v}_i) = \lambda_i(\mathbf{P}\mathbf{v}_i).$$

Note que, na equação (4), há dois produtos nas associações marcadas:

$$(5) \quad \mathbf{P}\mathbf{P}^{-1} = \mathbf{I}, \text{ e}$$

$$(6) \quad \mathbf{P}\mathbf{v}_i = \mathbf{x}_i,$$

onde o primeiro resulta na matriz identidade (veja equação (5)) e o segundo resulta em um vetor pois é um produto Matriz-Vetor. Assim, a equação (4) pode ser reescrita como

$$(7) \quad \mathbf{I}\mathbf{A}\mathbf{x}_i = \lambda_i\mathbf{x}_i \Rightarrow \mathbf{A}\mathbf{x}_i = \lambda_i\mathbf{x}_i.$$

Note as relações entre as equações (2) e (7):

- 1) Em (2) nós **resolvemos o problema de autovalores e autovetores da matriz $\bar{\mathbf{A}}$** que foi construída a partir das matrizes **A** e **P** dadas.
- 2) Porém, em (7), vemos que **o autovalor encontrado em (2) também é autovalor de A**.
- 3) Além disso, **o autovetor \mathbf{x}_i da matriz A associado ao autovalor λ_i é obtido simplesmente multiplicando o autovetor \mathbf{v}_i de $\bar{\mathbf{A}}$ encontrado em (2) pela matriz **P** dada, isto é, $\mathbf{P}\mathbf{v}_i = \mathbf{x}_i$ (veja equação (6)).**

Nossas conclusões finais para esta subseção são:

Em uma transformação de similaridade

$$\bar{A} = P^{-1}AP.$$

- 1) Para qualquer valor $i = 1, 2, \dots, n$, λ_i é autovalor tanto de \bar{A} quanto de A , portanto, **os espectros das duas matrizes são os mesmos**, isto é, $\lambda(A) \equiv \lambda(\bar{A})$;
- 2) Os **autovetores x_i** de A são obtidos a partir dos autovetores v_i de \bar{A} , **$x_i = Pv_i$** .

2.2 Escolha da matriz P .

O único inconveniente na transformação de similaridade $\bar{A} = P^{-1}AP$ é que temos de fornecer uma matriz P que tenha inversa e, pior ainda, temos que calcular sua inversa P^{-1} .

A boa notícia é que, se P for uma matriz ortogonal, então $P^{-1} = P^T$.

Relembrando: Uma matriz P é **ortogonal** se:

- 1) todas as suas **colunas são vetores unitários**, isto é, têm comprimentos iguais a 1, e,
- 2) todas as suas **colunas são vetores ortogonais entre si**, i. e., $col_j(P) \cdot col_k(P) = 0, j \neq k$

A **escolha** de uma matriz **P ortogonal** facilita a transformação de similaridade **$\bar{A} = P^TAP$** , mas **não garante** que o problema de autovalores da matriz transformada \bar{A} **seja mais fácil de resolver** do que o da matriz original A .

3. Métodos de Transformações de Similaridade

Nesta seção, vamos apresentar dois métodos principais e um método auxiliar que usam transformações de similaridade. O primeiro é o método de **Householder**, o segundo o de **Jacobi** e o terceiro o **QR**.

Todos os métodos têm por objetivo definir matrizes P de maneira que os autovalores e autovetores da matriz transformada \bar{A} daquele método sejam mais fáceis de obter do que o da matriz original.

3.1 O Método de Householder.

O objetivo deste método é fazer uma sequência de transformações de similaridade até que a matriz transformada final tenha uma estrutura simples chamada matriz tridiagonal (isso acontece porque estamos lidando com matrizes **A simétricas**). Antes de discutirmos as propriedades da matriz de Householder, $P \equiv H$, vamos entender como essas matrizes vão ser aplicadas para chegar na matriz \bar{A} final.

$$(8) \quad \bar{A} = H_{n-2}^T \left(\underbrace{H_{n-3}^T \cdots \left(\underbrace{H_2^T \left(\underbrace{H_1^T A H_1}_{\text{Passo 1}} \right) H_2}_{\text{Passo 2}} \right) \cdots H_{n-3}}_{\text{Passo (n-3)}} \right) H_{n-2}.$$

Passo (n-2)

A equação (8) dá uma visão macroscópica de como o algoritmo de Householder funciona. Vamos escrever este algoritmo em alto nível e depois apresentaremos os detalhes que faltam.

3.1.1 Método de Householder em forma algorítmica

O algoritmo abaixo não se preocupa com otimização, mas sim com a clareza.

```
(MatrizSimétrica, MatrizSimétrica) metodoDeHouseholder (MatrizSimétrica A, int n)
MatrizSimétrica H, Hi, Ai,  $\bar{\mathbf{A}}$ ;
// Inicializar matrizes
H  $\leftarrow$  I; // Esta matriz vai ser passada para o método de Jacobi ou QR
           // para recuperar os autovetores da matriz original.
Ai-1  $\leftarrow$  A;
Para i = 1 ... (n-2) faça
    // Construção da matriz de Householder do passo i
    Hi  $\leftarrow$  matrizHouseholderBaseadaNaCol_iDaMatrizDoPassoAnterior(Ai-1, i);

    // Transformação de similaridade do passo i
    Ai  $\leftarrow$  HiT Ai-1 Hi;

    // Salvar para o próximo passo.
    Ai-1  $\leftarrow$  Ai;

    // Acumular o produto das matrizes de Householder como H  $\leftarrow$  I H1 H2 ... Hn-3 Hn-2
    H  $\leftarrow$  H Hi;
End Para
// No final do loop, a matriz Ai já está no formato de uma matriz tridiagonal e já poderia ser
// retornada sem a cópia abaixo.
 $\bar{\mathbf{A}}$   $\leftarrow$  Ai;
return ( $\bar{\mathbf{A}}$ , H);
```

Neste momento, você consegue ver que o algoritmo é a reprodução exata da equação (8). Porém, alguns pontos precisam ser esclarecidos:

- 1) Por que o loop vai de 1 até n-2?
- 2) Como é o método matrizHouseholderBaseadaNaCol_iDaMatrizDoPassoAnterior(...)?
- 3) Para que serve a matriz acumulada **H**?

Na subseção 3.1.2, vamos esclarecer o ponto 1). Na subseção 3.1.3, vamos esclarecer o ponto 2). Finalmente, na seção 3.1.4, vamos esclarecer o ponto 3).

3.1.2 Estrutura da matriz tridiagonal \bar{A}

O método de Householder aplica uma sequência de transformações de similaridade para que a matriz original seja transformada em uma matriz de estrutura mais simples. A matriz tridiagonal tem a seguinte forma:

$$(9) \quad \bar{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n-2} & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n-2} & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ a_{n-2,1} & a_{n-2,2} & \cdots & a_{n-2,n-2} & a_{n-2,n-1} & a_{n-2,n} \\ a_{n-1,1} & a_{n-1,2} & \cdots & a_{n-1,n-2} & a_{n-1,n-1} & a_{n-1,n} \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n-2} & a_{n,n-1} & a_{n,n} \end{bmatrix}$$

$$= \begin{bmatrix} a_1 & b_1 & 0 & \cdots & 0 & 0 & 0 \\ b_1 & a_2 & b_2 & \cdots & 0 & 0 & 0 \\ 0 & b_2 & a_3 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & a_{n-2} & b_{n-2} & 0 \\ 0 & 0 & 0 & \cdots & b_{n-2} & a_{n-1} & b_{n-1} \\ 0 & 0 & 0 & \cdots & 0 & b_{n-1} & a_n \end{bmatrix}$$

A **diagonal principal** tem n elementos e está representada pela cor **verde**. As **duas subdiagonais paralelas** têm $n-1$ elemento e estão representadas pela cor **vermelha**.

Note que, em todas as colunas, exceto a última, há apenas 1 elemento **b** abaixo do elemento da diagonal principal e, **abaixo do elemento b**, todos os elementos são 0. Portanto, para transformar a matriz original para este formato, precisamos zerar os elementos abaixo dos **bs** em cada coluna. Porém, abaixo dos **bs** **só há elementos as serem zerados nas primeiras n-2** colunas. Como a matriz é simétrica, o que fizemos para as colunas acontecerá igualmente nas linhas.

Cada passo do loop tem o objetivo de deixar apenas um elemento diferente de zero abaixo do elemento da diagonal. Por isso, o loop vai da coluna 1 até a coluna n-2.

3.1.3 Construção da matriz de Householder H_i

Vamos repetir a equação (8) aqui e observar que a matriz H_i atua na matriz resultante dos parênteses mais internos, ou seja, na matriz resultante dos $(i-1)$ passos anteriores.

$$(8) \quad \bar{A} = H_{n-2}^T \left(H_{n-3}^T \cdots \left(H_2^T \underbrace{\left(H_1^T A H_1 \right)}_{\text{Passo 1}} H_2 \right) \cdots H_{n-3} \right) H_{n-2}$$

$\underbrace{\hspace{10em}}_{\text{Passo (n-3)}}$
 $\underbrace{\hspace{15em}}_{\text{Passo (n-2)}}$

Assim, quando aplicarmos a operação do passo i do algoritmo, isto é,

$$(10) \quad A_i \leftarrow H_i^T A_{i-1} H_i,$$

a estrutura tridiagonal, já conseguida nas primeiras (i-1) colunas e (i-1) linhas, deve ser preservada. Para que isso aconteça, a estrutura da matriz \mathbf{H}_i é composta de quatro blocos como

$$(11) \quad \mathbf{H}_i = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{H}_{II,II} \end{bmatrix}.$$

Essa estrutura é levada em conta no método

MatrizSimétrica **matrizHouseholderBaseadaNaCol_iDaMatrizDoPassoAnterior(...)**.

3.1.3.1 Matiz de reflexão de Householder

A matriz de Householder é uma matriz de reflexão. Isso significa que, dadas as informações de um plano (espelho) que passa pela origem do sistema de coordenadas, a matriz de Householder aplicada ao vetor posição de um ponto D produz o vetor posição do ponto D' que é a imagem do ponto D em relação ao espelho. Assim, dado o vetor unitário \mathbf{n} perpendicular ao plano do espelho, a matriz de Householder é escrita como

$$(12) \quad \mathbf{H} = \mathbf{I} - 2\mathbf{n}\mathbf{n}^T.$$

No nosso caso, \mathbf{n} não é dado, mas calculado. Vamos ver os dois problemas possíveis que levam à construção da matriz \mathbf{H} :

Problema 1. Usado em computação gráfica.

Dados o ponto D e o vetor \mathbf{n} normal (perpendicular) ao espelho, construir $\mathbf{H} = \mathbf{I} - 2\mathbf{n}\mathbf{n}^T$ para obter a imagem D' de D em relação ao espelho.

Problema 2. Nosso problema aqui.

Dado um vetor \mathbf{w} (considerado como vetor posição de algum ponto D), definimos o vetor \mathbf{w}' (considerado como vetor posição da imagem D' do ponto D com relação a um espelho) e achamos o vetor \mathbf{n} do espelho que produziria a imagem \mathbf{w}' especificada.

Vamos fazer algumas observações que nos ajudarão a resolver o problema 2.

Observação 1. O vetor paralelo ao segmento de reta que liga D' a D é perpendicular ao espelho mas não tem comprimento unitário. Esse vetor pode ser escrito como

$$(13) \quad \mathbf{N} = \mathbf{w} - \mathbf{w}' \Rightarrow \mathbf{n} = \frac{\mathbf{N}}{\|\mathbf{N}\|}$$

onde

$$(14) \quad \|\mathbf{N}\| = \sqrt{\mathbf{N} \cdot \mathbf{N}}.$$

Observação 2. Como o espelho passa pela origem, a distância da origem ao ponto D e a distância da origem ao ponto D' são iguais. Isso significa que

$$(15) \quad \|\mathbf{w}'\| = \|\mathbf{w}\|.$$

Observação 3. Se pusermos a imagem D' sobre o eixo de coordenadas cujo vetor unitário é \mathbf{e}_j , já que a equação (19) dá o comprimento do vetor \mathbf{w}' , podemos escrever o vetor \mathbf{w}' como

$$(16) \quad \mathbf{w}' = \|\mathbf{w}\|\mathbf{e}_j.$$

Solução do Problema 2.

1. Calcular o comprimento de \mathbf{w} substituindo \mathbf{N} por \mathbf{w} na equação (14);
2. Calcular \mathbf{w}' usando a equação (16);
3. Calcular \mathbf{N} e \mathbf{n} usando as equações (13) e (14);
4. Construir a matriz \mathbf{H} usando a equação (12).

Exemplo do efeito de H sobre a estrutura de uma matriz

Suponha que os dois primeiros passos do algoritmo já foram executados e a matriz obtida seja

$$A_2 = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 4 & 3 & 0 & 0 \\ 0 & 3 & 5 & 2 & 1 \\ 0 & 0 & 2 & 6 & 3 \\ 0 & 0 & 1 & 3 & 8 \end{bmatrix}$$

O próximo passo é construir a matriz H_3 a partir das informações da coluna 3 da matriz A_2 . A matriz H_3 deve preservar as duas primeiras linhas e colunas e transformar a coluna 3 e a linha 3 de forma que o vetor abaixo da diagonal da coluna 3 faça o papel de \mathbf{w} . Assim

$$\mathbf{w} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{pmatrix}$$

i) Calcular o comprimento de \mathbf{w} substituindo \mathbf{N} por \mathbf{w} na equação (14)

$$\|\mathbf{w}\| = \sqrt{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{pmatrix}} = \sqrt{4 + 1} = \sqrt{5}$$

ii) Calcular \mathbf{w}' usando a equação (16)

$$\mathbf{w}' = \sqrt{5} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sqrt{5} \\ 0 \end{pmatrix}$$

iii) Calcular \mathbf{N} e \mathbf{n} usando as equações (13) e (14)

$$\mathbf{N} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \\ 1 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 0 \\ \sqrt{5} \\ 0 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 - \sqrt{5} \\ 1 \end{pmatrix}$$

$$\|\mathbf{N}\| = \sqrt{(2 - \sqrt{5})^2 + 1}$$

$$\mathbf{n} = \frac{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 - \sqrt{5} \\ 1 \end{pmatrix}}{\sqrt{(2 - \sqrt{5})^2 + 1}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\sqrt{\frac{1}{2} - \frac{1}{\sqrt{5}}} \\ \sqrt{\frac{1}{2} + \frac{1}{\sqrt{5}}} \end{pmatrix}$$

Construir a matriz \mathbf{H}_3 usando a equação (12).

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} - 2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\sqrt{\frac{1}{2} - \frac{1}{\sqrt{5}}} \\ \sqrt{\frac{1}{2} + \frac{1}{\sqrt{5}}} \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ -\sqrt{\frac{1}{2} - \frac{1}{\sqrt{5}}} \\ \sqrt{\frac{1}{2} + \frac{1}{\sqrt{5}}} \end{pmatrix}^T$$

$$\mathbf{H}_3 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \\ 0 & 0 & 0 & \frac{1}{\sqrt{5}} & -\frac{2}{\sqrt{5}} \end{bmatrix}$$

Vamos aplicar essa matriz para obter a matriz

$$\mathbf{A}_3 = \mathbf{H}_3 \mathbf{A}_2 \mathbf{H}_3 = \begin{bmatrix} 3 & 1 & 0 & 0 & 0 \\ 1 & 4 & 3 & 0 & 0 \\ 0 & 3 & 5 & 2.23607 & 0 \\ 0 & 0 & 2.23607 & 8.8 & -2.6 \\ 0 & 0 & 0 & -2.6 & 5.2 \end{bmatrix}.$$

Numa matriz 5 x 5, $\mathbf{A}_3 = \bar{\mathbf{A}}$

3.1.3.2 Mostrar que a matriz de Householder é simétrica

Se \mathbf{H} for simétrica, ela tem de ser igual à sua transposta. Assim,

$$(17) \quad \mathbf{H}^T = (\mathbf{I} - 2\mathbf{n}\mathbf{n}^T)^T = (\mathbf{I})^T - 2(\mathbf{n}\mathbf{n}^T)^T = \mathbf{I} - 2(\mathbf{n}^T)^T(\mathbf{n})^T = \mathbf{I} - 2\mathbf{n}\mathbf{n}^T = \mathbf{H}$$

3.1.3.3 Mostrar que a matriz de Householder é ortogonal

Se a matriz for ortogonal sua transposta é igual à sua inversa. Porém, com $\mathbf{H}^T = \mathbf{H}$, basta mostrar que

$$\mathbf{H} \cdot \mathbf{H} = \mathbf{I}.$$

Assim,

$$\begin{aligned} \mathbf{H} \cdot \mathbf{H} &= (\mathbf{I} - 2\mathbf{n}\mathbf{n}^T)(\mathbf{I} - 2\mathbf{n}\mathbf{n}^T) \\ &= \mathbf{I} \cdot \mathbf{I} - 2\mathbf{I}\mathbf{n}\mathbf{n}^T - 2\mathbf{n}\mathbf{n}^T\mathbf{I} + 4\mathbf{n}\mathbf{n}^T\mathbf{n}\mathbf{n}^T \\ &= \mathbf{I} - 4\mathbf{n}\mathbf{n}^T + 4\mathbf{n}(\mathbf{n}^T\mathbf{n})\mathbf{n}^T \\ &= \mathbf{I} - 4\mathbf{n}\mathbf{n}^T + 4\mathbf{n}(\mathbf{I})\mathbf{n}^T \\ &= \mathbf{I} - 4\mathbf{n}\mathbf{n}^T + 4\mathbf{n}\mathbf{n}^T \\ \mathbf{H} \cdot \mathbf{H} &= \mathbf{I}. \end{aligned}$$

Logo, a matriz \mathbf{H} é sua própria transposta e sua própria inversa. Portanto, \mathbf{H} é ortogonal.

Vamos descrever o algoritmo de construção da matriz \mathbf{H}_i .

3.1.3.4 Algoritmo de construção da matriz de Householder

```
MatrizSimétrica metodoDeHouseholder (MatrizSimétrica A, int i)
MatrizSimétrica I;
Vetor w, w', N, n, e;
// Inicializar vetores
w ← 0; // Vetor nulo com n elementos
w' ← 0; // Vetor nulo com n elementos
e ← 0; // Vetor nulo com n elementos
// Copiar os elementos abaixo da diagonal da coluna i da matriz A
// para as respectivas posições no vetor w, isto é, da posição i+1 até o final.
w(i + 1:n) ← A((i + 1:n), i);
// Calcular o comprimento do vetor w
Lw ← ||w||;
// Copiar Lw na posição i+1 do vetor w'
w'(i + 1) ← Lw;
// Calcular o vetor N
N ← w – w';
// Normalizar o vetor N
n ←  $\frac{\mathbf{N}}{\|\mathbf{N}\|}$ ;
// Montar a matriz de Householder
H ← I – 2nnT;
return (H);
```

Observação: Cuidado com os índices na implementação.

Tarefa #13:

Com a matriz **A** abaixo, faça o que se pede:

$$\mathbf{A} = \begin{bmatrix} 40 & 8 & 4 & 2 & 1 \\ 8 & 30 & 12 & 6 & 2 \\ 4 & 12 & 20 & 1 & 2 \\ 2 & 6 & 1 & 25 & 4 \\ 1 & 2 & 2 & 4 & 5 \end{bmatrix}$$

- 1) Implemente o método de Householder e aplique-o sobre **A** para encontrar
 - i. a matriz tridiagonal $\bar{\mathbf{A}}$
 - ii. a matriz acumulada $\mathbf{H} = \mathbf{H}_1\mathbf{H}_2\mathbf{H}_3$
- 3) Use os métodos da potência para encontrar os autovalores e autovetores da matriz $\bar{\mathbf{A}}$.
- 4) Usando a matriz **H** e os autovetores da matriz $\bar{\mathbf{A}}$ encontre os autovetores da matriz **A**.
- 5) Encontre os autovalores da matriz **A**.