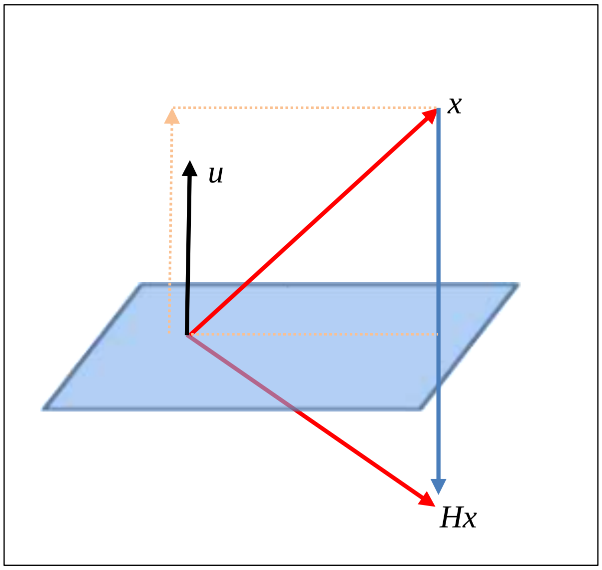
TRANFORMACIÓN DE HOUSEHOLDER

Definiremos la matriz de Householder por $H = I - \frac{2uu^t}{u^t u}$

Interpretación geométrica

Dados dos vectores u, x no nulos(representados como matrices columna) calcularemos la reflexión de x con respecto a un plano P que tiene a u como vector normal a esta reflexión la denotaremos Hx



Hallamos la proyección ortogonal de x sobre u

$$Proy_u x = \frac{ux}{\|u\|^2} u := u\frac{u^t x}{u^t u}$$

Es decir $Hx = x + (-2Proy_u x)$

$$= x - 2u\frac{u^t x}{u^t u}$$

$$:= \left(I - 2\frac{uu^t}{u^t u}\right) x$$

EJEMPLO: Sea $u = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad x = \begin{bmatrix} 5 \\ 7 \\ 3 \end{bmatrix}$

$$u^t u = 1 \qquad uu^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad -\frac{2}{u^t u} uu^t = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix}$$

$$H = I - \frac{2uu^t}{u^t u} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

$$Hx = \left(I - \frac{2uu^t}{u^t u}\right) x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} x$$

$$Hx = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 5 \\ 7 \\ 3 \end{bmatrix} = \begin{bmatrix} 5 \\ 7 \\ -3 \end{bmatrix}$$

Si tomamos dos vectores paralelos

$$u=\begin{bmatrix}5\\7\\3\end{bmatrix}\quad x=\begin{bmatrix}5\\7\\3\end{bmatrix}$$

$$u^tu=83\quad uu^t=\begin{bmatrix}25&35&15\\35&49&21\\15&21&9\end{bmatrix}\qquad -\frac{2}{u^tu}u^tu=\begin{bmatrix}-0.6024&-0.8434&-0.3614\\-0.8434&-1.1807&-0.5060\\-0.3614&-0.5060&-0.2169\end{bmatrix}$$

$$H=I-\frac{2uu^t}{u^tu}=\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\end{bmatrix}-\begin{bmatrix}-0.6024&-0.8434&-0.3614\\-0.8434&-1.1807&-0.5060\\-0.3614&-0.5060&-0.2169\end{bmatrix}=\begin{bmatrix}0.3976&-0.8434&-0.3614\\-0.8434&-0.1807&-0.5060\\-0.3614&-0.5060&0.7831\end{bmatrix}$$

$$Hx=\left(I-\frac{2uu^t}{u^tu}\right)x=\begin{bmatrix}0.3976&-0.8434&-0.3614\\-0.8434&-0.1807&-0.5060\\-0.3614&-0.5060&0.7831\end{bmatrix}x$$

$$Hx=\begin{bmatrix}0.3976&-0.8434&-0.3614\\-0.8434&-0.1807&-0.5060\\-0.3614&-0.5060&0.7831\end{bmatrix}\begin{bmatrix}5\\7\\3\end{bmatrix}=\begin{bmatrix}-5\\-7\\-3\end{bmatrix}$$

Dado un vector no nulo $x\neq e_1$ existe una matriz de Householder H tal que Hx es un múltiplo de e_1

Definiendo $H=I-\frac{2uu^t}{u^tu}$ con $u=x+sign(x_1)\|x\|e_1$

EJEMPLO: Sea $x=\begin{bmatrix}2\\4\\7\end{bmatrix}$ entonces $u=\begin{bmatrix}2\\4\\7\end{bmatrix}$+(+)(8.3066$\begin{bmatrix}1\\0\\0\end{bmatrix}=\begin{bmatrix}10.3066\\4\\7\end{bmatrix}$

$$Hx=\left(I-\frac{2uu^t}{u^tu}\right)x=\begin{bmatrix}-0.2408&-0.4815&-0.8427\\-0.4815&0.8131&-0.3271\\-0.8427&-0.3271&0.4277\end{bmatrix}\begin{bmatrix}2\\4\\7\end{bmatrix}=\begin{bmatrix}-8.3066\\0\\0\end{bmatrix}$$

MATRICES HOUSEHOLDER Y LA FACTORIZACION QR

Dada una matriz A de orden n existen una matriz ortogonal Q y una matriz triangular R tal que A=QR

La matriz Q se puede hallar como $Q=H_1H_2\cdots H_{n-1}$ donde cada H_i es una matriz de Householder para mostrar que la factorización QR de A puede obtenerse usando matrices de Householder seguiremos los siguientes pasos

Paso 1: construir una matriz de Householder H_1 tal que H_1A tenga ceros debajo de la entrada (1,1) en la primera columna

$$H_1A = \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & & \vdots \\ 0 & * & \cdots & * \end{bmatrix}$$

Para ello construiremos H_1 como, $H_1 = I_n - \frac{2u_n u_n{}^t}{u_n{}^t u_n}$ tal que

$$H_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Asi definiremos H_1A, tal que $A^{(1)} = H_1A$

Paso 2: Construimos la matriz de Householder H_2 tal que $H_2A^{(1)}$ tiene ceros debajo de la entrada (2,2) en la segunda columna de tal forma que los ceros creados en la primera etapa no son destruidos

$$A^{(2)} = H_2A^{(1)} = \begin{bmatrix} * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ 0 & 0 & * & \cdots & * \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & * & \cdots & * \end{bmatrix}$$

Donde H_2 se define como

$$H_2 = \begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & & & \\ 0 & & \widehat{H}_2 & \\ \vdots & & & \\ 0 & & & \end{bmatrix}$$

Donde $\widehat{H}_2 = I_{n-1} - \frac{2u_{n-1}u_{n-1}{}^t}{u_{n-1}{}^t u_{n-1}}$ es de orden n-1 tal que

$$\widehat{H}_2 \begin{bmatrix} a_{22} \\ a_{32} \\ \vdots \\ a_{n2} \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Paso k: Construimos la matriz de Householder

$$\widehat{H}_k = I_{n-(k-1)} - \frac{2u_{n-(k-1)}u_{n-(k-1)}{}^t}{u_{n-(k-1)}{}^t u_{n-(k-1)}} = I_{n-k+1} - \frac{2u_{n-k+1}u_{n-k+1}{}^t}{u_{n-k+1}{}^t u_{n-k+1}}$$

De orden n-k+1 tal que

$$\widehat{H}_k \begin{bmatrix} a_{kk} \\ a_{k+1,k} \\ \vdots \\ a_{nk} \end{bmatrix} = \begin{bmatrix} * \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Definiendo H_k como

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \widehat{H}_k \end{bmatrix}$$

Asi $A^{(k)} = H_k A^{(k-1)}$ para $k = n-1, \dots, 2$ luego de n-1 etapas la matriz $A^{(n-1)}$ será una matriz triangular R, por tanto

$$R = A^{(n-1)} = H_{n-1}A^{(n-2)} = H_{n-1}H_{n-2}A^{(n-3)} = \cdots = H_{n-1}H_{n-2}\cdots H_2 H_1 A$$

Denotamos por $Q^t = H_{n-1}H_{n-2}\cdots H_2 H_1$, como cada matriz Hk es ortogonal Q^t es ortogonal por tanto obtenemos que

$$R = Q^t A \qquad A = QR$$

La matriz $Q = H_1^t H_2^t \cdots H_{n-1}^t$ es también ortogonal

El algoritmo requiere de aproximadamente $\frac{2n^3}{3}$ operaciones para calcular la matriz triangular R, en efecto:

La construcción de $\widehat{H}_k$ requiere cerca de 2(n-k) operaciones mientras que $A^{(k)} = H_k A^{(k-1)}$ requiere $2(n-k)^2$ operaciones así el número total de operaciones está dado por

$$2\sum_{k=1}^{n-1}[(n-k)^2 + (n-k)] = 2[(n-1)^2 + (n-2)^2 + \cdots + 1^2] + 2[(n-1) + (n-2) + \cdots + 1]$$

$$= 2\frac{n(n-1)(2n-1)}{6} + 2\frac{n(n-1)}{2}$$

$$\approx \frac{2n^3}{3}$$

EJEMPLO: Hallar la descomposición QR de la matriz

A =

4	6	5	8
4	6	1	2
3	7	4	2
1	4	7	8

Usamos el programa creado de nombre **DescQR** y el programa **matrizHC**

```
function [Q,R,QR]=DesQR(A)
n=length(A);Ai=A;Q=eye(n);
for i=1:n-1
   j=0;x=[];
   for k=i:n
      j=j+1;x(j)=Ai(k,i);
   end
   xi=x';
   [Hi,U]=matrizHC(x');Hsigma=Hi*xi;
   for m=length(Hi)+1:n
   z=cat(2,zeros(length(Hi),1),Hi);I=eye(m);
   Hi=cat(1,I(1,:),z);
   end
   Hi
   Ai=Hi*Ai,Q=Q*Hi;
   R=Q'*A;
end
Q;R;QR=Q*R;
```

```
function [Hi,U]=matrizHC(x)
n=length(x);I=eye(n);e1=I(1,:)';
if x(1)~=0 sig=sign(x(1));
else sig=1;
end
u=x+sig*norm(x)*e1;
Hi=eye(n) -(2/(u'*u))*(u*u');
U=u;
```

```
>> [Q,R,QR]=DesQR(A)
Hi =

   -0.6172   -0.6172   -0.4629   -0.1543
   -0.6172    0.7644   -0.1767   -0.0589
   -0.4629   -0.1767    0.8675   -0.0442
   -0.1543   -0.0589   -0.0442    0.9853
```

Ai =

-6.4807	-11.2641	-6.6350	-8.3324
0.0000	-0.5889	-3.4405	-4.2333
-0.0000	2.0583	0.6696	-2.6750
0.0000	2.3528	5.8899	6.4417

Hi =

1.0000	0	0	0
0	-0.1851	0.6471	0.7396
0	0.6471	0.6467	-0.4038
0	0.7396	-0.4038	0.5384

Ai =

-6.4807	-11.2641	-6.6350	-8.3324
-0.0000	3.1810	5.4265	3.8173
-0.0000	0.0000	-4.1716	-7.0704
0.0000	0.0000	0.3561	1.4174

Hi =

1.0000	0	0	0
0	1.0000	0	0
0	0	-0.9964	0.0850
0	0	0.0850	0.9964

Ai =

-6.4807	-11.2641	-6.6350	-8.3324
-0.0000	3.1810	5.4265	3.8173
0.0000	-0.0000	4.1868	7.1654
0.0000	0.0000	0	0.8110

Q =

-0.6172	-0.2994	0.6041	-0.4055
-0.6172	-0.2994	-0.3512	0.6372
-0.4629	0.5614	-0.5058	-0.4634
-0.1543	0.7111	0.5058	0.4634

R =

-6.4807	-11.2641	-6.6350	-8.3324
-0.0000	3.1810	5.4265	3.8173
-0.0000	-0.0000	4.1868	7.1654
-0.0000	-0.0000	0	0.8110

QR =

4.0000	6.0000	5.0000	8.0000
4.0000	6.0000	1.0000	2.0000
3.0000	7.0000	4.0000	2.0000
1.0000	4.0000	7.0000	8.0000

LA DESCOMPOSICION QR PARA LA DETERMINACIÓN DE AUTOVALORES

Tomaremos la sucesión de matrices $(A_n)_{n\epsilon N}$ que converja a una matriz triangular en cuya diagonal aparecen los autovalores de la matriz A.

Tomamos inicialmente $A_0 = A$, hallaremos la descomposición QR de $A_n = Q_n R_n$ mediante matrices de Householder y $A_{n+1} = R_n Q_n$

$$\begin{gathered} A_0 = A = Q_0 R_0 \\ A_1 = R_0 Q_0 \\ A_2 = R_1 Q_1 \\ \vdots \\ A_{n+1} = R_n Q_n \end{gathered}$$

EJEMPLO: Hallar los autovalores de la matriz simétrica

A =

20	14	3	23	19
14	26	7	22	20
3	7	5	7	7
23	22	7	31	24
19	20	7	24	24

Usaremos el programa **autoQR**

```
function M_triangular=autoQR(A)
Ai=A;n=length(A);e=1;
while e>0.1
[Q,R,QR]=DesQR(Ai);
A0=Ai;
Ai=R*Q;
e=norm(Ai-A0,1);
end
autovalores=diag(Ai)
M_triangular=Ai;
```

```
>> M_triangular=autoQR(A)

Ai =

   87.0502   11.7636    2.5336    0.0972   -0.0109
   11.7636   12.2687    0.7581   -0.3155   -0.0005
    2.5336    0.7581    3.6865   -0.3672   -0.0064
    0.0972   -0.3155   -0.3672    2.9863    0.0162
   -0.0109   -0.0005   -0.0064    0.0162    0.0083

Ai =

   88.9111    1.4336    0.1034    0.0032    0.0000
    1.4336   10.5190    0.1476   -0.0912   -0.0000
    0.1034    0.1476    3.6519   -0.2861    0.0000
    0.0032   -0.0912   -0.2861    2.9097   -0.0000
    0.0000   -0.0000    0.0000   -0.0000    0.0082

Ai =

   88.9371    0.1694    0.0042    0.0001   -0.0000
    0.1694   10.4970    0.0553   -0.0250    0.0000
    0.0042    0.0553    3.6887   -0.2235   -0.0000
    0.0001   -0.0250   -0.2235    2.8690    0.0000
   -0.0000    0.0000   -0.0000    0.0000    0.0082
```

```
Ai =

   88.9375    0.0200    0.0002    0.0000    0.0000
    0.0200   10.4971    0.0204   -0.0068   -0.0000
    0.0002    0.0204    3.7122   -0.1723    0.0000
    0.0000   -0.0068   -0.1723    2.8450   -0.0000
    0.0000   -0.0000    0.0000   -0.0000    0.0082

Ai =

   88.9375    0.0024    0.0000    0.0000    0.0000
    0.0024   10.4971    0.0074   -0.0018   -0.0000
    0.0000    0.0074    3.7263   -0.1314   -0.0000
    0.0000   -0.0018   -0.1314    2.8309    0.0000
   -0.0000    0.0000   -0.0000    0.0000    0.0082

autovalores =

   88.9375
   10.4971
    3.7263
    2.8309
    0.0082

M_triangular =

   88.9375    0.0024    0.0000    0.0000    0.0000
    0.0024   10.4971    0.0074   -0.0018   -0.0000
    0.0000    0.0074    3.7263   -0.1314   -0.0000
    0.0000   -0.0018   -0.1314    2.8309    0.0000
   -0.0000    0.0000   -0.0000    0.0000    0.0082
```

Verificamos el resultado usando la instrucción **eig**

```
>> eig(A)

ans =

    0.0082
    2.8120
    3.7451
   10.4971
   88.9375
```

TRANSFORMACION A LA FORMA DE HESSENBERG SUPERIOR USANDO LA TRANFORMACION DE HOUSEHOLDER

Sea A una matriz de orden n siempre puede ser trasformada a una matriz Hessenberg superior por semejanza ortogonal

$$PAP^t = H_u$$

Donde H_u es la matriz de Hessenberg superior

$$H_u = \begin{bmatrix} * & * & \cdots & * & * \\ * & * & \cdots & * & * \\ 0 & * & \cdots & * & * \\ 0 & 0 & \ddots & * & * \\ \vdots & \vdots & \ddots & * & \vdots \\ 0 & 0 & \cdots & 0 & * \end{bmatrix}$$

$$A^{(1)} = H_1 A H_1^t$$

$$A^{(2)} = H_2 H_1 A H_1^t H_2^t$$

Luego de n-2 iteraciones

$$H_u = \underbrace{H_{n-2} H_{n-3} \cdots H_2 H_1}_{P} A \underbrace{H_1^t H_2^t \cdots H_{n-3}^t H_{n-2}^t}_{P^t} = PAP^t$$

Donde

$$H_k = \begin{bmatrix} I_k & 0 \\ 0 & \widehat{H}_k \end{bmatrix}$$

EJEMPLO: Hallar la matriz de Hessenberg de la matriz

A =

1	3	0	5	3
6	1	6	4	9
7	7	3	7	2
2	8	5	8	7
5	7	9	5	7

Ejecutamos el programa **DesHS**

```
function [P,Hu,PxPt]=DesHS(A)
n=length(A);Ai=A;P=eye(n);
for i=1:n-2
    j=0;x=[];
    for k=i+1:n
        j=j+1;x(j)=Ai(k,i);
    end
    xi=x';
   [Hi,U]=matrizHC(x');
    for m=length(Hi)+1:n
    z=cat(2,zeros(length(Hi),1),Hi);I=eye(m);
    Hi=cat(1,I(1,:),z);
    end
    Hi,Ai=Hi*Ai*Hi',P=Hi*P;
end
Hu=P*A*P';PxPt=P*P';
```

```
>> [P,Hu,PxPt]=DesHS(A)

Hi =

    1.0000         0         0         0         0
         0   -0.5620   -0.6556   -0.1873   -0.4683
         0   -0.6556    0.7248   -0.0786   -0.1966
         0   -0.1873   -0.0786    0.9775   -0.0562
         0   -0.4683   -0.1966   -0.0562    0.8596

Ai =

    1.0000   -4.0273   -2.9496    4.1572    0.8931
  -10.6771   19.5877    1.8990   -7.1374   -2.1089
    0.0000    4.0439   -3.2369    1.0350   -5.8882
    0.0000   -9.0668   -3.5910    4.3503   -0.1174
         0   -5.3401    0.2518   -0.4873   -1.7011
```

```
Hi =

    1.0000         0         0         0         0
         0    1.0000         0         0         0
         0         0   -0.3587    0.8043    0.4737
         0         0    0.8043    0.5239   -0.2804
         0         0    0.4737   -0.2804    0.8348

Ai =

    1.0000   -4.0273    4.8249   -0.4449   -1.8174
  -10.6771   19.5877   -7.4209   -1.6205    1.1404
    0.0000  -11.2728    3.4809   -0.1155   -0.5621
    0.0000    0.0000    2.2001   -0.7507   -6.6756
    0.0000    0.0000   -2.7791    0.3215   -3.3179

Hi =

    1.0000         0         0         0         0
         0    1.0000         0         0         0
         0         0    1.0000         0         0
         0         0         0   -0.6207    0.7841
         0         0         0    0.7841    0.6207

Ai =

    1.0000   -4.0273    4.8249   -1.1488   -1.4769
  -10.6771   19.5877   -7.4209    1.9000   -0.5627
    0.0000  -11.2728    3.4809   -0.3690   -0.4394
   -0.0000    0.0000   -3.5445    0.7635    1.5202
    0.0000    0.0000   -0.0000   -5.4770   -4.8320

P =

    1.0000         0         0         0         0
         0   -0.5620   -0.6556   -0.1873   -0.4683
         0   -0.1373   -0.4164    0.7878    0.4325
         0   -0.2021   -0.2127   -0.5693    0.7680
         0   -0.7903    0.5929    0.1419    0.0614

Hu =
```

```
    1.0000   -4.0273    4.8249   -1.1488   -1.4769
  -10.6771   19.5877   -7.4209    1.9000   -0.5627
         0  -11.2728    3.4809   -0.3690   -0.4394
         0    0.0000   -3.5445    0.7635    1.5202
    0.0000   -0.0000    0.0000   -5.4770   -4.8320
```

PxPt =

```
    1.0000         0         0         0         0
         0    1.0000    0.0000    0.0000   -0.0000
         0    0.0000    1.0000   -0.0000   -0.0000
         0    0.0000   -0.0000    1.0000    0.0000
         0   -0.0000   -0.0000    0.0000    1.0000
```

METODO DE LA POTENCIA

```
function  PS_RAY(A,e)
n=length(A);
z=ones(n,1);w=zeros(n,1);
m=0;
while norm(z-w)>e
    m=m+1;
    w=z;z=A*w;
    for i=1:n
        if abs(z(i))==max(abs(z))
            break
        end
    end
    z=z/z(i);
end
c1=(z'*A*z)/(z'*z);error=norm(A*z-c1*z);
fprintf('n= %g iteraciones  error=%f\n',m,error)
fprintf('maximo autovalor  =%f \n',c1)
```

EJEMPLO: Hallar el autovalor máximo de la matriz

```
A =

    86    40    61
    40    46    55
    61    55    77
```

Usando el algoritmo anterior

```
>> PS_RAY(A,0.01)
n= 3 iteraciones  error=0.073142
maximo autovalor  =176.962179
```

verificamos la respuesta usan la instrucción **eig** del Matlab

```
>> eig(A)

ans =

    3.6866
   28.3512
  176.9622
```

METODO DE LA POTENCIA INVERSA

```
function  PI_RAY(A,e)
n=length(A);
z=ones(n,1);w=zeros(n,1);
n=0;
while norm(z-w)>e
    n=n+1;
    w=z;
    z=A\w;
    for i=1:length(A)
        if abs(z(i))==max(abs(z))
            break
        end
    end
    z=z/z(i);
end
c1=(z'*A*z)/(z'*z);error=norm(A*z-c1*z);
fprintf('n= %g iteraciones  error=%f\n',n,error)
fprintf('minimo autovalor  =%f\n',c1)
```

EJEMPLO: Hallar el autovalor mínimo de la matriz

```
A =

    86    40    61
    40    46    55
    61    55    77

>> PI_RAY(A,0.01)
n= 4 iteraciones  error=0.006044
minimo autovalor  =3.686598
```

verificamos la respuesta usan la instrucción **eig** del Matlab

```
>> eig(A)

ans =

    3.6866
   28.3512
  176.9622
```