Programación: descomposición QR completa usando reflexiones de Householder

Objetivos. Programar una función que realice el algoritmo de descomposición QR usando reflexiones de Householder.

Requisitos. Reflexión ortogonal respecto a un hipersubespacio, construcción de un vector de Householder.

Se recomienda resolver estos ejercicios antes de la clase práctica.

1. Reflexión ortogonal respecto a un hipersubespacio (repaso). Dado un vector $a \in \mathbb{R} \setminus \{\mathbf{0}_n\}$, denotamos por H_a a la matriz de reflexión ortogonal respecto al hipersubespacio $\{a\}^\perp$. Recuerde la fórmula para la matriz H_a:

$$H_a =$$

2. Aplicar la reflexión ortogonal simultáneamente a varias columnas. Supongamos que $a \in \mathbb{R} \setminus \{\mathbf{0}_n\}$ y $X \in \mathcal{M}_{n\times m}(\mathbb{R})$. Aplicamos H_a a todas las columnas de X:

$$H_a X = X - \frac{2}{\|a\|^2}\big(aa^\top\big)X = X - \frac{2}{\|a\|^2}a(a^\top X).$$

Determine cuál de las dos fórmulas escritas requiere menos operaciones aritméticas con entradas de matrices y vectores (en otras palabras, menos flops).

3. Construcción de un vector de Householder (repaso). Dado un vector $f \in \mathbb{R}^n$ tal que $f \neq \|f\|e_1$, encuentre un vector $a \in \mathbb{R}^n$ tal que $H_a f = \|f\|e_1$:

$$a = \underbrace{\qquad\qquad}_{?}.$$

Escriba una función `householdervector` que calcule este vector.

4. Descomposición QR completa con reflexiones de Householder, el caso de tres columnas.

```
function [Q, R] = qrh3columns(A),
   [n, m] = size(A); R = A; Q = eye(n);
   # step 1
   a = householdervector(R(1 : n, 1)); coef = 2 / (norm(a) ^ 2);
   R(1 : n, 1 : 3) -= coef * a * (a' * R(1 : n, 1 : 3));
   Q(:, 1 : n) -= coef * (Q(:, 1 : n) * a) * a';
   # step 2
```

```
   a = householdervector(R(2 : n, 2)); coef = 2 / (norm(a) ^ 2);
   R(2 : n, 2 : 3) -= coef * a * (a' * R(2 : n, 2 : 3));
   Q(:, 2 : n) -= coef * (Q(:, 2 : n) * a) * a';
   # step 3
   a = householdervector(R(3 : n, 3)); coef = 2 / (norm(a) ^ 2);
   R(3 : n, 3 : 3) -= coef * a * (a' * R(3 : n, 3 : 3));
   Q(:, 3 : n) -= coef * (Q(:, 3 : n) * a) * a';
end
```

Prueba:

```
function [] = testqrh3columns(),
   n = 5; m = 3;
   A = rand(n, m);
   [Q, R] = qrh3columns(A);
   display(Q);
   display(R);
   display(norm(Q' * Q - eye(n)));
   display(norm(Q * R - A));
end
```

5. Problema: descomposición QR con reflexiones de Householder. Generalice las funciones de ejercicios anteriores al caso de un número arbitrario de columnas.

> Entrada: $A \in \mathcal{M}_{n\times m}(\mathbb{R})$, donde $n \geq m$ y $\mathrm{r}(A) = m$.
>
> Salida: matrices $Q \in \mathcal{M}_n(\mathbb{R})$ y $R \in \mathcal{M}_{n\times m}(\mathbb{R})$ tales que $A = QR$, $Q^\top Q = I_n$ y $R_{j,k} = 0$ para cualesquiera j, k tales que $j > k$.

6. Pruebas con matrices pequeñas. Haga pruebas pequeñas de la función programada (con matrices de 3 columnas, de 4 columnas, de 5 columnas) para verificar que la función es correcta.

7. Pruebas con matrices grandes. Haga pruebas con matrices aleatorias de tamaños grandes ($n = 200$, $n = 400$, o más grandes, $m = n$ o $m = n/2$), y analice cómo depende de n el tiempo de ejecución.

8. Tarea optativa. Revise libros y encuentre sugerencias sobre el cálculo del vector de Householder y la realización del método QR con reflexiones de Householder. Intente de disminuir los errores de redondeo en la función programada.