Programación: descomposición QR reducida usando el método modificado de Gram–Schmidt

Objetivos. Programar una función que realice el algoritmo de descomposición QR usando el método modificado de Gram–Schmidt.

Requisitos. Operaciones con vectores (operaciones lineales y el producto punto), ciclos, la idea del método modificado de Gram–Schmidt.

1. Descomposición QR de una matriz de tres columnas usando el método modificado de Gram–Schmidt.

```
function [Q, R] = QRMGS3(A),
   [n, m] = size(A); # suponemos que m = 3
   Q = A; R = eye(m);
   # first step
   R(1, 1) = norm(Q(:, 1));
   Q(:, 1) = Q(:, 1) / R(1, 1);
   R(1, 2) = Q(:, 1)' * Q(:, 2);
   Q(:, 2) = Q(:, 2) - R(1, 2) * Q(:, 1);
   R(1, 3) = Q(:, 1)' * Q(:, 3);
   Q(:, 3) = Q(:, 3) - R(1, 3) * Q(:, 1);
   # segundo paso
   R(2, 2) = norm(Q(:, 2));
   Q(:, 2) = Q(:, 2) / R(2, 2);
   R(2, 3) = Q(:, 2)' * Q(:, 3);
   Q(:, 3) = Q(:, 3) - R(2, 3) * Q(:, 2);
   # tercer paso
   R(3, 3) = norm(Q(:, 3));
   Q(:, 3) = Q(:, 3) / R(3, 3);
end
```

Prueba:

```
function [] = testQR3(),
   n = 5; m = 3;
   A = randn(n, m);
   [Q, R] = QRMGS3(A);
   display(Q);
   display(R);
   display(norm(Q' * Q - eye(m)));
   display(norm(Q * R - A));
   display(norm(R - triu(R)));
end
```

2. Descomposición QR de una matriz de cuatro columnas usando el método modificado de Gram–Schmidt. Convierta la función QRMGS3 en una función QRMGS4 que trabaje con una matriz de cuatro columnas. Utilice operaciones matriciales para trabajar con varias columnas simultaneamente. Por ejemplo, en vez de

```
R(1, 2) = Q(:, 1)' * Q(:, 2);
R(1, 3) = Q(:, 1)' * Q(:, 3);
R(1, 4) = Q(:, 1)' * Q(:, 4);
```

se puede hacer

```
R(1, 2 : 4) = Q(:, 1)' * Q(:, 2 : 4);
```

Haga comprobaciones para la función QRMGS4.

3. Problema: descomposición QR con el método modificado de Gram–Schmidt.

Entrada: $A \in \mathcal{M}_{n\times m}(\mathbb{R})$, donde $n \geq m$ y $\mathrm{r}(A) = m$.

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

```
function [Q, R] = QRGSM(A),
   [n, m] = size(A);
   Q = A; R = eye(m);
   for p = 1 : m,
      R(p, p) = ???;
      ???;
      R(p, ??? : ???) = ???;
      Q(:, ??? : ???) = ???;
   end
end
```

4. Pruebas con matrices pequeñas. Hacer pruebas pequeñas de la función programada con matrices de 3 columnas, de 4 columnas, de 5 columnas, para verificar que la función regresa resultados correctos.

5. Pruebas con matrices grandes. Hacer pruebas con matrices aleatorias de tamaños grandes ($m = 512$, $m = 1024$ o más grandes, $n = m$ o $n = 2m$), y analizar cómo depende de m el tiempo de ejecución.