UNIVERSIDAD DE COSTA RICA
FACULTAD DE CIENCIAS
ESCUELA DE MATEMÁTICA

MA-0501 Análisis Numérico I
TAREA 3

Fecha de entrega: Viernes 08 de noviembre, 11:59pm.
Instrucciones:

- La entrega de la tarea debe realizarse en la plataforma de Mediación Virtual, antes de la fecha y hora establecida.
- Debe subir dos archivos:
  - Un archivo en formato pdf con la resolución de los ejercicios y la discusión de los resultados. El documento debe llamarse `ApellidosNombre.pdf`
  - Un archivo `MATLAB` (`ApellidosNombre.m` o `ApellidosNombre.mlx`) con todos los códigos implementados, debidamente documentados y listos para ser ejecutados.
- Recuerde que *una imagen dice más que mil palabras*. Al mostrar resultados, presente gráficos claros debidamente rotulados, con escalas y leyendas adecuadas.
- El trabajo se puede realizar individualmente o en parejas. Si aplica, debe citar cualquier colaboración o referencia utilizada. En caso de ser en parejas, incluir en el nombre de los archivos `Apellido1Nombre1_Apellido2Nombre2.m`. Basta subir una entrega por pareja.
- Se evaluará sobre un total de 100 puntos (de 115 posibles); en caso de obtener un mayor puntaje, dichos puntos no serán acumulables ni transferibles.

Desarrollo

1. `[MATLAB]` Considere los siguientes tres algoritmos para obtener la factorización QR de una matriz A:

Data: Matriz $A \in \mathbb{R}^{m\times n}$
Result: Matriz unitaria $Q \in \mathbb{R}^{m\times n}$ y triangular superior $R \in \mathbb{R}^{n\times n}$ tales que $A = QR$
for $j = 1 : n$ **do**
  $\boldsymbol{v}_j = \boldsymbol{a}_j$;
  for $i = 1 : j-1$ **do**
    $r_{ij} = \boldsymbol{q}_i^T \boldsymbol{a}_j$;
    $\boldsymbol{v}_j = \boldsymbol{v}_j - r_{ij}\boldsymbol{q}_i$;
  end
  $r_{jj} = \|\boldsymbol{v}_j\|_2$;
  $\boldsymbol{q}_j = \boldsymbol{v}_j / r_{jj}$;
end

Algoritmo 1: Ortogonalización de Gram-Schmidt (inestable)

Data: Matriz $A \in \mathbb{R}^{m\times n}$
Result: Matriz unitaria $Q \in \mathbb{R}^{m\times n}$ y triangular superior $R \in \mathbb{R}^{n\times n}$ tales que $A = QR$
$V = A$;
for $i = 1 : n$ **do**
  $r_{ii} = \|\boldsymbol{v}_i\|_2$;
  $\boldsymbol{q}_i = \boldsymbol{v}_i / r_{ii}$;
  for $j = i+1 : n$ **do**
    $r_{ij} = \boldsymbol{q}_i^T \boldsymbol{v}_j$;
    $\boldsymbol{v}_j = \boldsymbol{v}_j - r_{ij}\boldsymbol{q}_i$;
  end
end

Algoritmo 2: Ortogonalización de Gram-Schmidt (estable)

Data: Matriz $A \in \mathbb{R}^{m\times n}$
Result: Matriz unitaria $Q \in \mathbb{R}^{m\times m}$ y triangular superior $R \in \mathbb{R}^{m\times n}$ tales que $A = QR$
$R = A$;
$M = I$;
for $j = 1 : n$ **do**
  $\boldsymbol{x} = R_{j:m,j}$ % vector de valores entre filas j y m de la columna j;
  $\boldsymbol{v}_j = \operatorname{sign}(x_1)\|\boldsymbol{x}\|_2\boldsymbol{e}_1 + \boldsymbol{x}$;
  $\boldsymbol{v}_j = \boldsymbol{v}_j / \|\boldsymbol{v}_j\|_2$;
  $R_{j:m,j:n} = R_{j:m,j:n} - 2\boldsymbol{v}_j\left(\boldsymbol{v}_j^T R_{j:m,j:n}\right)$;
  $M_{j:m,:} = M_{j:m,:} - 2\boldsymbol{v}_j\left(\boldsymbol{v}_j^T M_{j:m,:}\right)$;
end
$Q = M^T$;

Algoritmo 3: Triangularización de Householder

a) (5 puntos) Implemente una función `[Q,R] = qr1(A)` que implemente el Algoritmo 1.

b) (5 puntos) Implemente una función `[Q,R] = qr2(A)` que implemente el Algoritmo 2.

c) (5 puntos) Implemente una función `[Q,R] = qr3(A)` que implemente el Algoritmo 3.

d) (4 puntos) Tome $m = n = 20$. Genere una matriz aleatoria `A = rand(m)`. Calcule las tres factorizaciones QR con las tres funciones `qr1`, `qr2`, `qr3`. Para cada caso, calcule $\|A - QR\|_2$, $\|QQ^T - I\|_2$. ¿Se cumple que $A = QR$? ¿Se cumple que Q es ortogonal? Determine si algún algoritmo es *mejor* que otro.

e) (4 puntos) Tome ahora $m = n = 20$. Defina `A = hilb(m)` como la matriz de Hilbert de tamaño $m \times m$. Repita el inciso anterior. Determine si algún algoritmo es *mejor* que otro. Compare con el algoritmo `qr` de `MATLAB`.

2. [`MATLAB`] En esta pregunta se deben usar los comandos de factorización propios de `MATLAB` (`qr(A)`, `lu(A)`, `svd(A)`, `chol(A)`). Defina `m=12`, `x = ones(m,1)`, `A = hilb(m)`, `b = A*x`.

Buscamos resolver el sistema de ecuaciones $Ax = b$ mediante estas factorizaciones. Note que la solución exacta es $x = [1, \ldots, 1]^T$ y denotamos por $\hat{x}$ la solución aproximada numéricamente.

a) (2 puntos) Calcule el número de condición de A. ¿Qué nos indica este número? Sugerencia: utilice el comando `cond(A)`.

b) (3 puntos) Calcule la factorización $PA = LU$ y resuelva el sistema[1]. Determine $\|x - \hat{x}\|_2$.

c) (3 puntos) Calcule la factorización $A = QR$ y resuelva el sistema. Determine $\|x - \hat{x}\|_2$.

d) (3 puntos) Calcule la factorización $A = LL^T$ y resuelva el sistema. Determine $\|x - \hat{x}\|_2$.

e) (3 puntos) Calcule la factorización $A = USV^T$ y resuelva el sistema. Determine $\|x - \hat{x}\|_2$.

f) (1 punto) Determine si algún método brinda alguna solución aceptable en términos del error.

g) (3 puntos) Utilizando la factorización en valores singulares de A, obtenga la aproximación de rango $\nu = 9$ dada por $A_\nu = \sum_{i=1}^{\nu} \sigma_i u_i v_i^T$. Resuelva el sistema $A_\nu x = b$ y determine $\|x - \hat{x}\|_2$. Justifique sus resultados.

3. Para graficar una función en dos variables $z = f(x, y)$, se puede generar una malla (matriz) de puntos `xx,yy` en las cuales se evalúa f. De esta forma, `zz = f(xx,yy)` es una matriz del mismo tamaño de `xx` y `yy`.

a) (2 puntos) [`MATLAB`] Defina los vectores

`x = linspace(0,1,100), y = linspace(0,2,200).`

La malla de puntos equidistantes en el dominio $D = [0, 1] \times [0, 2]$ se genera mediante el comando `[xx,yy]=meshgrid(x,y)` (note que `xx` y `yy` son matrices). Grafique la función

$$f(x, y) = \sin(2\pi(x + y)) \sin(\pi(x - y))$$

en esta malla del dominio D. Sugerencia: utilice el comando `surf(xx,yy,f(xx,yy))` y revise opciones de visualización como `view`, `colorbar`, `shading`.

b) (4 puntos) [`MATLAB`] Defina `zz = f(xx,yy)`, donde f es la función del inciso anterior. Calcule la descomposición en valores singulares de `zz`. Mediante el comando `subplot`, grafique en una misma ventana las mejores aproximaciones de rango 1,2,3 y 4. En una nueva ventana, grafique el error absoluto para cada una de las cuatro aproximaciones. ¿Qué observa?

c) (2 puntos) [`MATLAB`] ¿Cuáles son los valores singulares y el rango de `zz`?

[1] Para resolver sistemas triangulares superiores o inferiores $L\boldsymbol{y} = \boldsymbol{b}$ o $U\boldsymbol{z} = \boldsymbol{b}$, basta ejecutar `y=L\b` o `z=U\b`, pues el comando `\` chequea si las matrices son triangulares y realiza sustitución hacia adelante o atrás. De esta forma, realmente no se está invirtiendo ninguna matriz.

d) (4 puntos) Demuestre (a mano) que la matriz tiene el rango y los valores singulares correspondientes. Sugerencia: recuerde fórmulas para $\sin(x \pm y)$.

4. Sea

$$p(x) = c_0 + c_1 x + \ldots + c_{n-1}x^{n-1}, \; x \in [0,1].$$

Considere el problema de minimizar

$$E(c_0, \ldots, c_{n-1}) := \|f - p\|_2^2 = \int_0^1 |f(x) - p(x)|^2 \, dx \to \text{mín}\,.$$

(a) (5 puntos) Obtenga las derivadas parciales $\partial E(c_0, \ldots, c_{n-1})/\partial c_k$ $(k = 0, \ldots, n-1)$. Para obtener el punto crítico, exprese el resultado como un sistema de n ecuaciones con n incógnitas, $M\boldsymbol{c} = \boldsymbol{b}$. ¿Qué problema tiene este sistema?

(b) (5 puntos) [`MATLAB`] Resuelva el sistema correspondiente para $n = 10$, $f(x) = \cos(4\pi x)$. Dibuje en un mismo gráfico f y p. ¿Cuál es el valor de $\|f - p\|_2^2$? Sugerencia: puede utilizar el comando `integral(f,0,1)`.

(c) (5 puntos) [`MATLAB`] Para obtener una mejor aproximación, considere la base de polinomios ortogonales de Legendre $\{\phi_j\}_{j\geq 0}$. La proyección de f sobre el espacio de polinomios de grado $n-1$ viene dada por

$$p(x) = \beta_0 + \beta_1\phi_1(x) + \ldots \beta_{n-1}\phi_{n-1}(x),$$

donde

$$\beta_k = \frac{\langle f, \phi_k\rangle}{\langle \phi_k, \phi_k\rangle}.$$

Calcule p para $n = 10$, $f(x) = \cos(4\pi x)$. Compare con los resultados del apartado anterior. Dibuje en la misma gráfica la nueva aproximación y calcule la 2-norma del error.

5. Si $\boldsymbol{u}, \boldsymbol{v} \in \mathbb{R}^m$, la matriz $A = I + \boldsymbol{u}\boldsymbol{v}^T$ es llamada *perturbación de rango uno de la identidad.*

(a) (4 puntos) Muestre que si A es invertible, entonces su inversa tiene la forma $A^{-1} = I + \alpha\boldsymbol{u}\boldsymbol{v}^T$ con $\alpha \in \mathbb{R}$. Encuentre el valor de α explícitamente.

(b) (4 puntos) ¿Para qué valores de $\boldsymbol{u}$ y $\boldsymbol{v}$ es A singular? En este caso, determine el núcleo de A.

6. (6 puntos) Dada $A = U\Sigma V^T \in \mathbb{R}^{m\times n}$ una descomposición en valores singulares de A, defina su pseudoinversa como $A^+ = V\Sigma^+ U^T$, donde Σ^+ se obtiene de Σ al invertir sus entradas no nulas; esto es$(\Sigma^+)_{ii} = 1/\sigma_{ii}$ si $\sigma_{ii} > 0$. Demuestre que si $\boldsymbol{x}$ está en el espacio fila de A y $\boldsymbol{y} = A\boldsymbol{x}$ entonces $\boldsymbol{x} = A^+\boldsymbol{y}$.

7. Considere un conjunto de mediciones $\{(x_i, y_i)\}_{i=1}^m$. En este ejercicio deseamos resolver el problema de mínimos cuadrados, donde la función buscada $f(x, \boldsymbol{c})$ no depende linealmente de los coeficientes $\boldsymbol{c} = [c_1, \ldots, c_n]^T \in \mathbb{R}^n$ (en clase analizaremos el caso donde f es un polinomio). De esta forma, deseamos hallar $\boldsymbol{c}$ tal que

$$\sum_{i=1}^{m} |f(x_i, \boldsymbol{c}) - y_i|^2 \to \text{mín}\,.$$

Para x_i fijo y f con derivadas continuas (con respecto a $c_1, \ldots, c_n$), considere la aproximación de primer orden con centro $\boldsymbol{c}^{(k)}$ dada por

$$f(x_i, \boldsymbol{c}) \approx f(x_i, \boldsymbol{c}^{(k)}) + J_f(x_i, \boldsymbol{c}^{(k)})(\boldsymbol{c} - \boldsymbol{c}^{(k)}),$$

donde

$$J_f(x_i, \boldsymbol{c}) = \nabla_{\boldsymbol{c}} f(x_i, \boldsymbol{c}) \in \mathbb{R}^{1\times n}$$

es el Jacobiano de f con derivadas parciales $\dfrac{\partial f}{\partial c_j}$ evaluado en x_i.

De esta forma, deseamos que

$$J_f(x_i, \boldsymbol{c}^{(k)})(\boldsymbol{c} - \boldsymbol{c}^{(k)}) \approx y_i - f(x_i, \boldsymbol{c}^{(k)}) \;\forall\; i = 1, \ldots, m. \tag{1}$$

Defina la matriz $J(\boldsymbol{c}^{(k)}) \in \mathbb{R}^{m\times n}$ donde su fila i viene dada por $J_f(x_i, \boldsymbol{c}^{(k)})$. Defina además

$$\boldsymbol{y} := [y_1, \ldots, y_m]^T \in \mathbb{R}^m,$$

$$\boldsymbol{g}(\boldsymbol{c}) := [f(x_1, \boldsymbol{c}), \ldots, f(x_m, \boldsymbol{c})]^T \in \mathbb{R}^m.$$

Podemos escribir así las ecuaciones dadas en (1) de forma matricial. Si sabemos que $\boldsymbol{c}^{(k)}$ es cercano a $\boldsymbol{c}$, buscamos encontrar $\boldsymbol{c}$ tal que

$$J(\boldsymbol{c}^{(k)})(\boldsymbol{c} - \boldsymbol{c}^{(k)}) \approx \boldsymbol{y} - \boldsymbol{g}(\boldsymbol{c}^{(k)}). \tag{2}$$

Para tal efecto, consideramos el siguiente algoritmo:

- Considere $\boldsymbol{c}^{(0)}$ dado.
- Para $k = 0, 1, 2, \ldots$
  - Obtenga $\Delta\boldsymbol{c}^{(k)}$ tal que $\|J(\boldsymbol{c}^{(k)})\Delta\boldsymbol{c}^{(k)} - (\boldsymbol{y} - \boldsymbol{g}(\boldsymbol{c}^{(k)}))\|_2 \to \text{mín}$ (por ejemplo, mediante el uso de las ecuaciones normales).
  - Defina $\boldsymbol{c}^{(k+1)} = \boldsymbol{c}^{(k)} + \Delta\boldsymbol{c}^{(k)}$.

En este ejercicio asumimos que f tiene la forma

$$f(x, \boldsymbol{c}) = c_1 e^{-c_2 x} \sin(c_3 x + c_4), x \in [0, 10],$$

donde $\boldsymbol{c} = [c_1, c_2, c_3, c_4]^T \in \mathbb{R}^4$ es el vector de coeficientes que debemos de encontrar (los cuales corresponden a la amplitud, el decaimiento, el periodo y la fase de f, respectivamente). Para obtener cotas del error, asumiremos que la solución exacta viene dada por el vector

$$\boldsymbol{c}_{\text{exact}} = [1, 1/2, 2, 0]^T.$$

(a) (1 punto) Utilice el comando `load data` para cargar los datos guardados en el archivo `data.dat` suministrado. Este archivo incluye los valores $\{(x_i, y_i)\}_{i=1}^m$ para $m = 100$, guardados en dos vectores $\boldsymbol{x}$ y $\boldsymbol{y}$.

(b) (5 puntos) Implemente un programa que ejecute el algoritmo descrito. Para detener la iteración, utilice la condición $\|\Delta \boldsymbol{c}^{(k)}\|_\infty < 10^{-8}$.

(c) (4 puntos) Corra su programa para $\boldsymbol{c}_0 = [1.1; 0.4; 2.1; 0.2]$. Reporte el número de iteraciones requeridas. Grafique los nodos $\{x_i, y_i\}_{i=1}^m$, la función exacta $f(x, \boldsymbol{c}_{\text{exact}})$ y la función aproximada $f(x, \boldsymbol{c}^{(k)})$. Grafique además $\|\Delta \boldsymbol{c}^{(k)}\|_\infty$ en función de k. ¿Qué puede deducir sobre la velocidad de convergencia?

(d) (2 puntos) Para el valor inicial del inciso anterior, grafique el número de condición $\kappa(J(\boldsymbol{c}^{(k)}))$ en función de k. ¿Qué nos indica este gráfico? Sugerencia: utilice la función `cond(J)`.

(e) (2 puntos) Ahora considere $\boldsymbol{c}_0 = [1, 1, 1, 1]^T$. ¿Qué ocurre en este caso?

8. Considere un conjunto de $m + 1$ puntos $\{(x_i, y_i)\}_{i=0}^m$, donde

$$0 = x_0 < x_1 < \ldots < x_{m-1} < x_m = 1$$

es una partición del intervalo $[0, 1]$ (no necesariamente con puntos equidistantes). En particular asumiremos que $y_0 = y_m$ pues consideraremos funciones periódicas. Deseamos construir una función $s_2 : \mathbb{R} \to \mathbb{R}$ tal que:

- s_2 es periódica con periodo 1; esto es, $s_2(x + 1) = s_2(x) \ \forall \ x \in \mathbb{R}$.
- $s_2 \in C^1(\mathbb{R})$; esto es, s_2 tiene derivada continua.
- s_2 interpola los valores del conjunto de puntos; esto es, $s_2(x_i) = y_i \ \forall \ i = 0, 1, \ldots, m$.
- la restricción de s_2 a cada subintervalo $[x_i, x_{i+1}]$ $(i = 0, 1, \ldots, m - 1)$ es un polinomio cuadrático.

Es claro que basta construir s_2 en el intervalo $[0, 1]$ al ser periódica.

(a) (7 puntos) Escriba un sistema de ecuaciones que permita determinar $s_2 : [0,1] \to \mathbb{R}$ (Sugerencia: puede seguir la construcción de los splines cúbicos naturales del Capítulo 7; en este caso defina $\sigma_i = s_2'(x_i),\ i = 0, 1, \ldots, m$ y genere un sistema para encontrar los σ_i. Otra idea es escribir la restricción de $s_2(x)$ en cada subintervalo $[x_i, x_{i+1}]$ de la forma $s_2(x) = \alpha_i + \sigma_i(x - x_i) + \beta_i(x - x_i)^2$. Recuerde además que s_2 debe ser periódica).

(b) [`MATLAB`] (4 puntos) Fije $m = 15$ y tome $f(x) := cos(2\pi x)$. Defina $\boldsymbol{x} = [x_0, \ldots, x_m]^T$ utilizando valores aleatorios con la función `rand` y calcule $y_i = f(x_i)\ \forall\ i = 0, 1, \ldots, m$. Construya s_2 para este conjunto de datos. Grafique f, s_2 y los puntos $\{(x_i, y_i)\}_{i=0}^{m}$ en un mismo gráfico. Grafique además el error absoluto $|f(x) - s_2(x)|$ y estime $\|f - s_2\|_{L^\infty([0,1])}$.

(c) [`MATLAB`] (3 puntos) Repita el ejercicio anterior para $\boldsymbol{x} =$ `linspace(0,1,m+1)'`. Compare con los resultados del inciso anterior y comente sus resultados.

(d) [`MATLAB`] (6 puntos) Uno puede aproximar $f'(x)$ mediante $s_2'(x)$. Utilizando s_2 tal como se calculó en el inciso anterior, escriba un programa que calcule $s_2'(x)$. Grafique f' y s_2' en el mismo gráfico. Grafique además el error absoluto $|f'(x) - s_2'(x)|$ y estime $\|f' - s_2'\|_{L^\infty([0,1])}$. Note que s_2' es diferenciable a trozos. ¿Cómo podría aproximar f' por una función s de manera tal que s'' sea continua?