

EJERCICIO VARIOS
SCT - JU.24.06.21

1. Sean $P, Q \in \mathbb{C}^{k \times k}$ matrices unitarias, las cuales pueden ser expresadas en función de sus filas o columnas de la siguiente forma:

$$P = \left[\begin{array}{c|c|c|c|c|c} p_1 & p_2 & \dots & p_{k-1} & p_k \end{array} \right] = \left[\begin{array}{c} \hline \hat{p}_1 \\ \hline \hat{p}_2 \\ \hline \vdots \\ \hline \hat{p}_{k-1} \\ \hline \hat{p}_k \end{array} \right], \quad Q = \left[\begin{array}{c|c|c|c|c|c} q_1 & q_2 & \dots & q_{k-1} & q_k \end{array} \right] = \left[\begin{array}{c} \hline \hat{q}_1 \\ \hline \hat{q}_2 \\ \hline \vdots \\ \hline \hat{q}_{k-1} \\ \hline \hat{q}_k \end{array} \right]$$

Además, considere $A = PQ$ y $x \in \mathbb{C}^k$.

- (a) Demuestre que la matriz A es unitaria.
- (b) Demuestre que $A^{-1} = \sum_{i=1}^k \hat{q}_i^* p_i^*$.
- (c) Demuestre que $\|Ax\|_2^2 = \|x\|_2^2$.
- (d) Demuestre que los vectores pertenecientes a $\text{range}(A)$ son linealmente independientes. *Hint: Las columnas de una matriz unitaria son ortonormales entre sí.*

2. El primer algoritmo que tradicionalmente se discute cuando uno estudia interpolación polinomial en una variable utiliza la matriz de Vandermonde y rápidamente uno llega a la conclusión que la matriz es mal condicionada, por lo cual es poco recomendable trabajar directamente con esta matriz utilizando aritmética de punto flotante. El segundo algoritmo que uno tradicionalmente estudia es la Interpolación de Lagrange y luego el algoritmo de Diferencias Divididas de Newton, ambos algoritmos son capaces de encontrar el polinomio interpolador en cuestión.

Considere el problema de interpolar 3 puntos: (x_1, y_1) , (x_2, y_2) y (x_3, y_3) , donde $x_1 \neq x_2$, $x_1 \neq x_3$ y $x_2 \neq x_3$. Si uno utiliza la matriz de Vandermonde uno obtiene el siguiente sistema de ecuaciones lineales:

$$\underbrace{\begin{pmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ 1 & x_3 & x_3^2 \end{pmatrix}}_{V_3} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}. \quad (1)$$

Donde a, b y c , son los coeficientes del polinomio interpolador $p_3(x) = a + bx + cx^2$. Un dato importante sobre la matriz de Vandermonde cuando interpola n puntos es que su determinante puede ser expresado de la siguiente forma:

$$\det(V_n) = \prod_{1 \leq i < j \leq n} (x_j - x_i).$$

Lo que indica que el determinante no es nulo siempre y cuando $x_i \neq x_j \forall i \neq j$. En nuestro caso particular nosotros tenemos $\det(V_3) = \prod_{1 \leq j < i \leq 3} (x_j - x_i) \neq 0$. Ahora, dado que sabemos que V_3 no es singular, podemos estudiar la inversa de esta matriz, i.e. V_3^{-1} .

- (a) Construya la interpolación de Lagrange o de Diferencias de Newton para obtener el polinomio interpolador de (x_1, y_1) , (x_2, y_2) y (x_3, y_3) .
- (b) Explique como puede usted obtener V_3^{-1} a partir de las interpolaciones anteriormente encontradas.
- (c) Obtenga V_3^{-1} utilizando el procedimiento descrito anteriormente. Note que V_3^{-1} satisface la siguiente ecuación:

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} = \underbrace{\begin{pmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{pmatrix}}_{V_3^{-1}} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}. \quad (2)$$

Donde $(V_3^{-1})_{ij} = b_{ij}$ para $i, j = 1, 2, 3$, i.e. b_{ij} son los coeficientes de la matriz V_3^{-1} .

3. Una de las funciones más interesantes dentro del mundo de la Computación Científica es la función Exponencial, $\exp(x)$, y se vuelve aún más interesante cuando su argumento es un número complejo $z = x + iy$, donde $i^2 = -1$, $x \in \mathbb{R}$ y $y \in \mathbb{R}$. Por simplicidad considere solo el dominio $\Omega = \{z = x + iy \mid |x| \leq 1, |y| \leq 1, x \in \mathbb{R} \text{ and } y \in \mathbb{R}\}$.
- (a) Obtenga la parte real e imaginaria de $\exp(z)$, donde $z = x + iy$. Es decir, obtenga $\text{Re}(\exp(z))$ y $\text{Im}(\exp(z))$, respectivamente. *Hint: Do you recall the Euler's formula $\exp(i\theta) = \cos(\theta) + i \sin(\theta)$? It may be helpful, but please read carefully before you may apply it.*
- (b) Explique como se puede utilizar interpolación polinomial para poder obtener la parte real e imaginaria de $\exp(z)$. Es decir, para cada $z = x + iy$ usted debe entregar el valor de $\text{Re}(\exp(z))$ y $\text{Im}(\exp(z))$ para $z \in \Omega$. Por simplicidad considere x y y es un input.
Nota : No construya el interpolador, solo explique como lo construirá. *Hint: Recall you only know how to do 1D interpolation, so please use it wisely.*
- (c) Explique claramente como puede asegurar que cada *componente* de su algoritmo de interpolación tendrá un error menor a ε . Determine todas las cotas que necesite para obtener todo el puntaje.

4. Dada una secuencia de n datos $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$, con $y_i = f(x_i)$, $x_1 < x_2 < \dots < x_n$ y $y_1 < y_2 < \dots < y_n$; se desea determinar un cierto \hat{x} tal que $f(\hat{x}) = c$, donde $\hat{x} \in [x_1, x_n]$ y $c \in [y_1, y_n]$ es un valor conocido. Un posible enfoque para resolver este problema consiste en invertir los roles de las variables x e y , tal que se obtenga un polinomio interpolador $Q(y)$ para la función inversa, es decir, $f^{-1}(y) \approx Q(y)$. De esta manera, al evaluar el polinomio interpolador en $y = c$, esto es, $Q(c)$, se puede obtener el valor de \hat{x} . A este enfoque se le conoce como **interpolación inversa** y suele ser exitoso si la función inversa $f^{-1}(y)$ es apta para ser aproximada localmente por un polinomio.

A continuación, considere los intervalos locales $x \in [-19.500000, 33.131593]$ e $y \in [-1.750000, 2.705000]$ para los cuales se han obtenido los polinomios interpoladores $P(x)$ y $Q(y)$ para una cierta función $f(x)$ y su inversa $f^{-1}(y)$, respectivamente, con n puntos equiespaciados. La Tabla 1 muestra el máximo error de interpolación al variar el número de puntos en cada caso. Considere $\text{MaxError}(f(x)) = \max_{x \in [x_1, x_n]} |f(x) - P(x)|$ y $\text{MaxError}(f^{-1}(y)) = \max_{y \in [y_1, y_n]} |f^{-1}(y) - Q(y)|$. La Tabla 2 muestra una data de puntos en intervalo local $y \in [-1.750000, 2.705000]$.

- (a) ¿Cuál es la menor cantidad de puntos usados para construir el polinomio interpolador $P(x)$, tal que $f(x)$ es aproximada con un menor error máximo?
- (b) ¿Cuál es la menor cantidad de puntos usados para construir el polinomio interpolador $Q(y)$, tal que $f^{-1}(y)$ es aproximada con un menor error máximo?
- (c) La Tabla 3 muestra los ceros de $f(x)$ obtenidos al utilizar un algoritmo de búsqueda de ceros y el polinomio interpolador $P(x)$ con n puntos equiespaciados. Además, se incluye el error absoluto respectivo. ¿Es posible utilizar el método de interpolación inversa, usando el polinomio interpolador $Q(y)$ para obtener un cerro de $f(x)$ con menor error absoluto que el reportado por la Tabla 3? Si su respuesta es afirmativa, encuentre el polinomio interpolador $Q(y)$ y reporte el error absoluto obtenido. En caso contrario, explique por qué no es posible. *Hint: un cerro de $f(x)$ es $r = -5.5$.*

n	MaxError($f(x)$)	MaxError($f^{-1}(y)$)
2	1.465429920804	17.310510000000
3	1.006442913015	8.507976750000
4	1.290437889011	0.000000000000
5	0.821769831965	0.000000000000
6	0.796895617300	0.000000000000
7	0.925798808746	0.000000000000
8	0.779068033901	0.000000000000
9	1.312915859328	0.000000000000
10	2.044329057632	0.000000000000
11	2.142430331012	0.000000000000
12	0.449573648686	0.000000000000
13	5.697822805297	0.000000000000
14	7.934193173311	0.000000000000
15	6.960626259215	0.000000000000
16	9.302263510621	0.000000000000
17	29.585399634093	0.000000000000
18	35.757803468460	0.000000000000
19	20.356129370770	0.000000000000
20	84.566878879464	0.000000000000
21	167.568545492845	0.000000000001
22	184.522773707511	0.000000000001
23	17.551988717740	0.000000000003
24	660.324622194300	0.000000000007
25	1041.445306622219	0.000000000013
26	936.589735569821	0.000000000007
27	907.083303921799	0.000000000024
28	4693.906195774704	0.000000000071
29	6306.065311134063	0.000000000027
30	3913.598643557723	0.000000000043
31	12235.110195776335	0.000000001212
32	31762.902169689569	0.000000000706
33	36533.060198583604	0.000000001655
34	6937.031285148519	0.000000006243
35	111575.485303357593	0.000000001125
36	206812.219595668925	0.000000005219
37	195836.443265721755	0.000000020460
38	106312.394475715279	0.000000011808
39	865560.434166750056	0.000000064978
40	1293737.065086921444	0.000000056089
41	898158.157771163853	0.000000400396

Tabla 1: Máximo error al interpolar la función $f(x)$ y su inversa $f^{-1}(y)$ con n puntos equiespaciados.

i	y_i	x_i
1	-1.750000	-19.500000
2	-1.304500	-12.095148
3	-1.255000	-11.495825
4	-1.193125	-10.801838
5	-1.113571	-9.995337
6	-1.007500	-9.060366
7	-0.940000	-8.542968
8	-0.859000	-7.995620
9	-0.795357	-7.617928
10	-0.760000	-7.426752
11	-0.636250	-6.853783
12	-0.477143	-6.308234
13	-0.413500	-6.140393
14	-0.265000	-5.837332
15	-0.130000	-5.642844
16	-0.079375	-5.583525
17	0.032000	-5.468446
18	0.159286	-5.345317
19	0.230000	-5.272116
20	0.477500	-4.918757
21	0.680000	-4.422336
22	0.725000	-4.275656
23	0.795714	-4.013237
24	0.923000	-3.430304
25	1.034375	-2.787170
26	1.113929	-2.242079
27	1.220000	-1.392504
28	1.368500	0.057936
29	1.432143	0.781373
30	1.490000	1.495848
31	1.591250	2.883545
32	1.715000	4.832789
33	1.814000	6.606984
34	1.962500	9.653566
35	2.068571	12.131851
36	2.148125	14.165697
37	2.210000	15.855672
38	2.259500	17.277862
39	2.300000	18.489000
40	2.705000	33.131593

Tabla 2: Data de puntos en el intervalo local $y \in [-1.750000, 2.705000]$.

n	r_a	$e_i = r_a - r $
2	-4.733958018000	0.564970314827
3	-4.899870546725	0.487367259998
4	-4.161251742856	0.757183749154
5	-5.166765458162	0.319148264006
6	-5.498188190039	0.001813442320
7	-5.004673283810	0.429262585405
8	-4.959717800266	0.455224847925

Tabla 3: Ceros del polinomio interpolador $P(x)$, construido con n puntos equiespaciados.

5. El precio del dólar ha estado fluctuando bastante este último tiempo. Este tipo de fluctuaciones ha sido visto como una interesante posibilidad de inversión por expertos del área de *High Frequency Trading* de finanzas. Sin embargo, para poder rentabilizar su negocio ellos necesitan saber si el dólar subirá o bajará en el *futuro inmediato*, ya que en ambos casos ellos ganan, pero la estrategia es diferente. Para poder responder a su pregunta, ellos han propuesto utilizar un algoritmo autoregresivo de segundo orden para estimar el valor futuro de dólar. Este algoritmo consiste en estimar el precio del dolar de un instante como una combinación lineal del precio del dolar en 2 instantes inmediatamente anteriores. En términos matemáticos esto significa lo siguiente:

$$X^{(n+1)} = \alpha_0 X^{(n)} + \alpha_1 X^{(n-1)},$$

donde $X^{(n)}$ corresponde al precio del dólar en el tiempo n . Considere la Tabla 2 para el precio del dólar en los *tiempos* discretos indicados.

n	$X^{(n)}$
0	700
1	705
2	703
3	690
4	695

Tabla 4: Precio del dolar

Por simplicidad solo nos referiremos a instantes discretos y equiespaciados en el tiempo.

- (i) Explique cómo determinaría las constantes α_0 y α_1 , y encuentrelas. *Hint: You should get a rectangular matrix of 3×2 and use an algorithm for that.*
- (ii) Utilice la aproximación con los α 's estimados en (i) para estimar el valor del dolar en el *tiempo* $n = 5$.
- (iii) Considerando que el valor del dólar verdadero para $n = 5$ fue 685. ¿Considera buena su estimación realizada en (ii)? Justifique. Recuerde que la empresa gana si sabe si el dólar subirá o bajará solamente.

6. Construya un algoritmo basado en la descomposición QR que determine si una matriz cuadrada es singular o no, usted debe escribir explícitamente su algoritmo.

7. Considere la siguiente función vectorial $\mathbf{F}(\mathbf{x}) = \underset{\mathbf{y}}{\operatorname{argmin}} \|A B \mathbf{x} - B \mathbf{y}\|_2$, donde $A \in \mathbb{R}^{n \times n}$ y densa; $B \in \mathbb{R}^{n \times m}$, *sparse* y es *full rank*, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$ y $m < n$.

(a) Construya un algoritmo que pueda evaluar $\mathbf{F}(\mathbf{x})$ para cualquier vector \mathbf{x} .

8. Sea $A \in \mathbb{R}^{m \times n}$ con $m > n$ y $\mathbf{b} \in \mathbb{R}^m$. Considere el siguiente sistema de ecuaciones lineales sobre-determinado:

$$A \mathbf{x} = \mathbf{b}. \quad (3)$$

El procedimiento tradicional para encontrar $\bar{\mathbf{x}}$ es por medio de minimizar el error cuadrático de (3), es decir,

$$\bar{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \|\mathbf{b} - A \mathbf{x}\|.$$

El cual puede ser obtenido a través de las ecuaciones normales,

$$A^* A \bar{\mathbf{x}} = A^* \mathbf{b},$$

donde $*$ corresponde al operador de trasposición y conjugación. Este método es muy utilizado en *Machine Learning*, pero una variante es quizás más utilizada, i.e. la regularización de Tikhonov. La cual consiste en resolver el siguiente sistema de ecuaciones lineales,

$$(A^* A + \delta I) \bar{\mathbf{x}}_\delta = A^* \mathbf{b},$$

donde I es la matriz identidad de orden n y $0 < \delta \in \mathbb{R}$. Ahora, considerando la descomposición QR-reducida de $A = \hat{Q} \hat{R}$, se propone la siguiente regularización,

$$(A^* A + \lambda \hat{R}) \hat{\mathbf{x}}_\lambda = A^* \mathbf{b}. \quad (4)$$

- (a) Considerando que ya se tiene acceso a la descomposición QR-reducida de A , proponga un algoritmo $\mathcal{O}(n^2)$ que obtenga $\hat{\mathbf{x}}_\lambda$.

9. Recordando a los ilustres profesores Jørgen Pedersen Gram y Erhard Schmidt, los cuales desarrollaron el proceso de ortonormalización de vectores en un espacio vectorial equipado con un producto interno. Conocido como la descomposición QR, en particular la descomposición QR reducida de una matriz $A \in \mathbb{R}^{m \times n}$ con $m \geq n \geq 1$, genera la matriz ortonormal $Q \in \mathbb{R}^{m \times n}$, i.e. sus columnas son ortonormales, y la matriz $\hat{R} \in \mathbb{R}^{n \times n}$ es triangular superior, como se muestra a continuación:

$$A = \left(\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array} \right) = \underbrace{\left(\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array} \right)}_{\hat{Q}} \underbrace{\begin{pmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & r_{nn} \end{pmatrix}}_{\hat{R}}$$

Sin embargo, en uno de los posibles futuros de la humanidad, se ha decretado que no se podrán utilizar matrices triangulares superiores en los procesos de ortonormalización. Lo cual deja a la humanidad de forma inmediata sin acceso a resolver problemas de mínimos cuadrados por medio de la clásica factorización QR...

Esto genera el inicio de la revolución científica liderada por los estudiantes de Computación Científica, donde su más fuerte habilidad es la construcción de algoritmos sofisticados que velen por el continuo avance de la Ciencia y la Ingeniería considerando la restricción definida, i.e. no utilizar matrices triangulares superiores.

Para resolver este problema, se propone construir la siguiente factorización matricial:

$$A = \left(\begin{array}{c|c|c|c} \mathbf{a}_1 & \mathbf{a}_2 & \cdots & \mathbf{a}_n \end{array} \right) = \underbrace{\left(\begin{array}{c|c|c|c} \mathbf{t}_1 & \mathbf{t}_2 & \cdots & \mathbf{t}_n \end{array} \right)}_{\hat{T}} \underbrace{\begin{pmatrix} u_{11} & 0 & 0 & \cdots & 0 \\ u_{21} & u_{22} & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ u_{n1} & u_{n2} & \cdots & \cdots & u_{nn} \end{pmatrix}}_{\hat{U}}$$

- (a) **[25 puntos]** Construya un algoritmo que determine la factorización TU propuesta, i.e. el input del algoritmo es la matriz A y retorna la matriz \hat{T} donde sus columnas son ortonormales y la matriz \hat{U} que es triangular inferior.

10. Considere que usted tiene acceso a la matriz unitaria $Q \in \mathbb{R}^{n \times n}$ y el coeficiente $\lambda \neq 0$. Ahora, se le solicita resolver el siguiente sistema de ecuaciones lineales,

$$(Q + \lambda I) \mathbf{x} = \mathbf{b},$$

donde $\mathbf{b} \in \mathbb{R}^n$, $\mathbf{x} \in \mathbb{R}^n$ y I es la matriz identidad de dimensiones $n \times n$. Lamentablemente no es posible utilizar la factorización matricial PALU dado que se necesita ir obteniendo una aproximación de la solución prontamente y se espera que la calidad de la solución vaya mejorando en el tiempo, en resumen, debemos usar un algoritmo iterativo. Además de los algoritmos iterativos clásicos que tenemos a nuestra disposición, que son el método de Jacobi y Gauss-Seidel, se proponen las siguientes iteraciones de punto fijo en alta dimensión:

$$\mathbf{x}^{(k+1)} = Q^T (\mathbf{b} - \lambda \mathbf{x}^{(n)}), \quad (5)$$

$$\mathbf{x}^{(k+1)} = \frac{1}{\lambda} (\mathbf{b} - Q^T \mathbf{x}^{(n)}). \quad (6)$$

Considere la siguiente definición de variables:

```
1 import numpy as np
2 np.random.seed(0)
3 n = 2000
4 Q, _ = np.linalg.qr(np.random.rand(n,n))
5 b = np.random.rand(n)
```

- Determine para que valores de λ se asegura convergencia del método de Jacobi.
- Determine para que valores de λ se asegura convergencia del método de Gauss-Seidel.
- Implemente el algoritmo inducido por la ecuación (5), al cual llamaremos Alg. 1.
- Implemente el algoritmo inducido por la ecuación (6), al cual llamaremos Alg. 2.
- Para $\lambda = 0.1$, determine, entre el Alg. 1, Alg. 2, el método de Jacobi y el método de Gauss-Seidel, cual de los 4 algoritmos converge primero a la solución considerando un error relativo del residuo de $1e - 5$.
- Para $\lambda = 1$, determine, entre el Alg. 1, Alg. 2, el método de Jacobi y el método de Gauss-Seidel, cual de los 4 algoritmos converge primero a la solución considerando un error relativo del residuo de $1e - 5$.
- Para $\lambda = 1000$, determine, entre el Alg. 1, Alg. 2, el método de Jacobi y el método de Gauss-Seidel, cual de los 4 algoritmos converge primero a la solución considerando un error relativo del residuo de $1e - 5$.