

Hoja1

April 9, 2020

1 Ejercicios teóricos

1.1 Números complejos

- (1) (*Golan p. 14*) Sea y un numero complejo que satisface $|y| < 1$. Encontrar el conjunto de todos los numeros complejos z tales que $|z - y| \leq |1 - \bar{y}z|$.
- (2) (*Golan p. 14*) Demostrar que $|z + 1| \leq |z + 1|^2 + |z|$ para todo $z \in \mathbb{C}$.
- (3) (*Golan p. 14*) Dado $z \in \mathbb{C}$, encontrar $w \in \mathbb{C}$ tal que $w^2 = z$.
- (4) (*Golan p. 14*) Para cualesquiera $z_1, z_2 \in \mathbb{C}$, demostrar que

$$|z_1|^2 + |z_2|^2 - z_1\bar{z}_2 - \bar{z}_1z_2 = |z_1 - z_2|^2.$$

- (5) (*Golan p. 14*) Encontrar el conjunto de todos los numeros complejos $z = a + bi$ que satisfacen la relacion dada. Notar que ese conjunto puede ser vacio, o puede ser todo el plano complejo. Justificar en cada caso.
 - (i) $z^2 = \frac{1}{2}(1 + i\sqrt{3})$.
 - (ii) $\sqrt{2}|z| \geq |a| + |b|$.
 - (iii) $|z| + z = 2 + i$.
 - (iv) $z^4 = 2 - \sqrt{12}i$.
 - (v) $z^4 = -4$.

1.2 Matrices

- (1) Dada $A \in \mathbb{R}^{n \times n}$, decimos que X es la inversa de A si verifica que: $AX = XA = I_n$. Notamos $A^{-1} = X$.
 - (i) Si una matriz $A \in \mathbb{R}^{n \times n}$ es simetrica e invertible, entonces A^{-1} tambien lo es.
 - (ii) La transpuesta y la inversa de una matriz satisfacen (mientras que las inversas esten definidas):
 - (ii.a) $(AB)^T = B^T A^T$
 - (ii.b) $(AB)^{-1} = B^{-1} A^{-1}$.

1.3 Símbolo de Landau y costo computacional

- (1) Probar que para todo $a \in \mathbb{R}$, $f(x) = x + a$ es una $\mathcal{O}(x^2)$.
- (2) Observar que el simbolo de Landau *no conmuta*, es $f(x) = \mathcal{O}(g(x))$ no implica $g(x) = \mathcal{O}(f(x))$.
- (3) Observar que el simbolo de Landau es reflexivo: $f = \mathcal{O}(f)$.
- (4) Sean $E \subseteq \mathbb{R}$, $f_1 : E \rightarrow \mathbb{R}$, $g_1 : E \rightarrow \mathbb{R}$, $f_2 : E \rightarrow \mathbb{R}$, $g_2 : E \rightarrow \mathbb{R}$ funciones y sea $k \in \mathbb{R}$ un real. Probar las siguientes propiedades:
 - (i) Si $f_1 = \mathcal{O}(g_1)$ y $g_1 = \mathcal{O}(g_2)$, entonces $f_1 = \mathcal{O}(g_2)$.
 - (ii) Si $f_1 = \mathcal{O}(g_1)$ y $g_1 = \mathcal{O}(g_2)$, entonces $f_1 f_2 = \mathcal{O}(g_1 g_2)$.
 - (iii) $f_2 \mathcal{O}(g_1) = \mathcal{O}(f_2 g_1)$ (igualdad entre conjuntos).
 - (iv) Si $f_1 = \mathcal{O}(g_1)$ y $f_2 = \mathcal{O}(g_2)$, entonces $f_1 + f_2 = \mathcal{O}(|g_1| + |g_2|)$.
 - (v) Si $k \neq 0$ entonces $\mathcal{O}(g_1) = \mathcal{O}(k g_1)$ (igualdad entre conjuntos).
 - (vi) Si $f_1 = \mathcal{O}(g_1)$ entonces $k f_1 = \mathcal{O}(g_1)$.
- (5) Sea $A \in \mathbb{R}^{n \times n}$ una matriz tridiagonal.
 - (i) La matriz A puede ser almacenada, utilizando su estructura particular, en espacio $\mathcal{O}(n)$ en vez de $\mathcal{O}(n^2)$.
 - (ii) El producto matriz-vector tiene una complejidad lineal $\mathcal{O}(n)$.
- (6) Sea $\{p_n\}$ la sucesion definida por $p_n = \sum_{k=1}^n \frac{1}{k}$. Probar que la sucesion diverge aun cuando el limite de las diferencias $p_n - p_{n-1}$ para $n \rightarrow \infty$ tiende a cero.
- (7) Sea $f(x) = (x - 1)^{10}$, $p = 1$, $p_n = 1 + 1/n$. Probar que $|f(p_n)| < 10^{-3}$ siempre que $n > 1$, pero que $|p - p_n| < 10^{-3}$ requiere que $n > 1000$.

2 Ejercicios para resolver por computadora

2.1 Básicos de Julia

- (1) Tipos de matrices en Julia.
 - (i) Construir, en Julia, algunos ejemplos de los tipos de matrices vistos en clase.
 - (ii) Cargar el paquete `LinearAlgebra` y e investigar los siguientes tipos: `Diagonal`, `Bidiagonal`, `Tridiagonal`, `SymTridiagonal`, `UpperTriangular`, `LowerTriangular`.
 - (iii) Generar las matrices especiales de la parte 1, pero usando los tipos de la parte (ii).
- (2) Números y matrices aleatorias
 - (i) Investigar los comandos `rand`, `randn` para generar vectores y matrices aleatorias.
 - (ii) Investigar los comandos `sprand`, `sprandn` para generar vectores y matrices esparsas aleatorios.
 - (iii) Visualizar las matrices generadas anteriormente con el comando `spy` de la libreria `PyPlot`.
- (3) Productos de matrices y vectores
 - (i) Escribir una funcion que recibe una matriz y un vector y devuelve el producto entre A y x .

- (ii) Observar que Julia ya “sabe” calcular Ax .
- (4) Escribir una funcion que recibe una matriz y devuelve la matriz transpuesta. Comparar con `transpose(A)`.

Nota: en Julia, `transpose(A)` una representación *lazy* de la matriz traspuesta, ver el articulo de wikipedia [evaluacion perezosa](#). Para “materializar” la transpuesta, hacer `copy(transpose(A))`.

- (5) Escribir una funcion que determina si una matriz dada es simétrica o no.

2.2 Algoritmos elementales

En todos los problemas siguientes, deberá:

- (i) Escribir el pseudo-código que resuelve el problema.
- (ii) Implementarlo en Julia.

Haber razonado y escrito el pseudo-código *antes* de largarse a escribir el programa es **muy importante!**

- (1) Dado un vector $x \in \mathbb{R}^n$, nos interesa calcular su suma,

$$s = \sum_{i=1}^n x_i.$$

- (2) Escribir una función que calcula el producto escalar entre dos vectores de \mathbb{R}^n .
- (3) Escribir una función que calcula el triángulo de Pascal hasta un orden dado.
- (4) Implementar un algoritmo que devuelve el maximo de una lista de numeros.
- (5) Implementar una [busqueda binaria](#) en una lista de numeros enteros.
- (6) Implementar el metodo de [busqueda binaria o biseccion](#) para hallar una raiz x de una funcion dada f que se asumaria continua en el intervalo cerrado $[a, b] \in \mathbb{R}$.