EJERCICIO EN CLASES
SCIENTIFIC COMPUTING TEAM - SEMANA - 06

1. Considere el siguiente sistema de ecuaciones lineales de dimensión $2\,n \times 2\,n$:

$$\begin{pmatrix} A & \alpha\,B \\ C & D \end{pmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}, \tag{1}$$

del cual sabemos que es posible obtener las factorizaciones LU de cada una de las sub-matrices A, B, C, y D, las cuales son:

$$\begin{aligned} A &= L_A\,U_A, \\ B &= L_A\,U_B, \\ C &= L_C\,U_A, \\ D &= L_D\,U_B. \end{aligned}$$

Además sabemos que $\alpha \neq 1$ y que las sub-matrices tienen dimensión de $n \times n$.

(a) ✍ Determine el número de operaciones elementales que se necesitan realizar en función de n para resolver el sistema de ecuaciones lineales definido en la ecuación (1) si se utilizara PALU. Es decir, resolver el sistema de ecuaciones lineales de dimensión $2\,n \times 2\,n$.

(b) ✍ Proponga un algoritmo que obtenga $\mathbf{x}_1$ y $\mathbf{x}_2$ utilizando las factorizaciones LU de las matrices A, B, C, y D. Hint: I suggest you to think how you can handle this problem if all the sub-matrices are the identity matrix, how would you deal with this?

(c) ✍ Determine el número de operaciones elementales del algoritmo propuesto incluyendo las operaciones elementales requeridas para obtener las factorizaciones LU de las sub-matrices A, B, C, y D.

(d) ✍ ¿Qué ocurre si $\alpha = 1$ con su propuesta de algoritmo?

(e) ⌨ Implemente su algoritmo.

(f) ⌨ Compare los tiempos de ejecución del algoritmo original y su propuesta de algoritmo para $n \in \{100, 1000\}$. ¿Es consistente con el número de operaciones elementales obtenido? Para la realización de este experimento debe construir aletoriamente las matrices L_A, L_C, L_D, U_A, y U_B. Recuerde que las matrices "L" deben tener 1 en la diagonal y que las matrices "U" no pueden tener 0 en la diagonal.

(g) ✍ ¿Cuánto más rápido es su propuesta de algoritmo que haber resuelto el problema original con PALU?

2. Uno de los primeros algoritmos iterativos que se construyeron en el curso fue el algoritmo para sacar la raíz cuadrada de un número real, sin embargo es mucho más interesante trabajar con números complejos. Sabemos que todo número complejo es posible descomponerlo en su parte real e imaginaria, es decir si $z \in \mathbb{C}$, entonces se puede escribir de la siguiente forma $z = x + \mathrm{i}\,y$, donde $\mathrm{i}^2 = -1$, $a \in \mathbb{R}$ y $b \in \mathbb{R}$. Lo interesante es que en la práctica uno puede manipular un número complejo solo con álgebra real, esto se logra trabajando con cada componente de forma independiente.

(a) ✍ Proponga un algoritmo basado en el método de Newton que obtenga la raíz **cúbica** de un número complejo $w_0 = a_0 + \mathrm{i}\,b_0$ solo considerando álgebra real, particularmente double precision, y que solo puede aplicar las 4 operaciones elementales, es decir: $+$, $-$, $*$, y $\div$. Su algoritmo debe recibir como entrada la parte real y imaginaria, es decir a_0 y b_0, y retornar la parte real y imaginaria de la raíz cúbica, es decir debe retornar x e y de $z = x + \mathrm{i}\,y$, tal que se cumpla que $z^3 = w_0$. Considerar que el error absoluto debe ser menor o igual a $1e-5$. Su función debe tener la siguiente firma:

```
solverCubicComplexRoot(a0,b0):
# Your code
return x, y
```

(b) ⌨ Obtenga la raíz cúbica de $w_0 = 9.2 - 2.1\,\mathrm{i}$, considerando el siguiente "datos inicial" para su algoritmo iterativo: $z_0 = 2 - 0.01\,\mathrm{i}$. ¿Existen más raíces?

3. Considere el siguiente sistema de ecuaciones lineales $A\,\mathbf{x} = \mathbf{b}$, donde $A \in \mathbb{C}^{n\times n}$ y $\mathbf{b} \in \mathbb{C}^n$, por lo tango $\mathbf{x} \in \mathbb{C}^n$. Es decir, tenemos matrices y vectores donde sus elementos pueden ser número complejos. El uso de un algoritmo tradicional exige que debamos ser capaces de representar y manipular números complejos, sin embargo, esto aún no lo hemos estudiado. Una forma de manejar este tipo de situaciones es descomponer nuestro sistema de ecuaciones lineales en su parte real e imaginaria de la siguiente forma, $(A_r + \mathrm{i}\,A_i)\,(\mathbf{x}_r + \mathrm{i}\,\mathbf{x}_i) = \mathbf{b}_r + \mathrm{i}\,\mathbf{b}_i$, donde el subscript r denota la parte real, el subscript i denota la parte imaginaria y $\mathrm{i}^2 = -1$.

(a) Construya un nuevo sistema de ecuaciones lineales en el cual no se requiera manipulación explicita de números complejos. Describa claramente su <u>nueva</u> matriz, el vector de incógnitas y el lado derecho. <u>Hint: Real is real and imaginary is imaginary!</u>

(b) Considerando ahora que A_r es una matriz diagonal, y que $|(A_r)_{k,k}| > \sum_{j=1}^{n} |(A_i)_{k,j}|$ donde $k = 1 : n$, proponga un algoritmo iterativo que asegure convergencia para el sistema de ecuaciones lineales propuesto en la parte **(a)**. Usted debe demostrar que el algoritmo propuesto convergerá. Notese que en la desigualdad anterior, la matriz que está al lado izquierdo es diferente a la matriz que está al lado derecho.