

Tarea 2, Fecha de entrega: Antes del lunes 29 de Agosto.

i) Leer de la página 7 a la 23 de http://www.cimat.mx/~alram/met_num/clases/clase03b.pdf. **Revisar, compilar, ejecutar y entender** todos los ejemplos de la clase03, verificar que se entiende el código con respecto a lo que se menciona en la clase03b. Mandar las salidas de los programas en un archivo de texto.

ii) Programar el método de sustitución-hacia-atrás y sustitución-hacia-adelante como dos funciones que resuelve sistemas lineales con estructura triangular superior y triangular inferior, respectivamente.

iii) Implementar en C/C++ el método de Eliminación Gaussiana **CON PIVOTEO PARCIAL ESCALADO** y juntarlo con la sustitución-hacia-atrás del punto anterior para resolver Sistemas de Ecuaciones Lineales (SEL). Usa el programa para resolver los SELs que se especifican abajo. Hacerlo usando precisión `double` completa y usando solo 4 dígitos para los valores (usar la función de la tarea anterior). Reportar que tan diferentes son las soluciones, es decir reportar la norma 2 del vector diferencia de las soluciones $\|x_1 - x_2\|_2$.

$$\begin{aligned} \text{(a)} \quad & \frac{1}{4}x_1 + \frac{1}{5}x_2 + \frac{1}{6}x_3 = 9, \\ & \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 = 8, \\ & \frac{1}{2}x_1 + x_2 + 2x_3 = 8. \end{aligned}$$

$$\begin{aligned} \text{(b)} \quad & 3.333x_1 + 15920x_2 - 10.333x_3 = 15913, \\ & 2.222x_1 + 16.71x_2 + 9.612x_3 = 28.544, \\ & 1.5611x_1 + 5.1791x_2 + 1.6852x_3 = 8.4254. \end{aligned}$$

$$\begin{aligned} \text{(c)} \quad & x_1 + \frac{1}{2}x_2 + \frac{1}{3}x_3 + \frac{1}{4}x_4 = \frac{1}{6}, \\ & \frac{1}{2}x_1 + \frac{1}{3}x_2 + \frac{1}{4}x_3 + \frac{1}{5}x_4 = \frac{1}{7}, \\ & \frac{1}{3}x_1 + \frac{1}{4}x_2 + \frac{1}{5}x_3 + \frac{1}{6}x_4 = \frac{1}{8}, \\ & \frac{1}{4}x_1 + \frac{1}{5}x_2 + \frac{1}{6}x_3 + \frac{1}{7}x_4 = \frac{1}{9}. \end{aligned}$$

$$\begin{aligned} \text{(d)} \quad & 2x_1 + x_2 - x_3 + x_4 - 3x_5 = 7, \\ & x_1 + 2x_3 - x_4 + x_5 = 2, \\ & -2x_2 - x_3 + x_4 - x_5 = -5, \\ & 3x_1 + x_2 - 4x_3 + 5x_5 = 6, \\ & x_1 - x_2 - x_3 - x_4 + x_5 = 3. \end{aligned}$$

Para procesar las matrices del SEL, estas deben de estar guardadas en memoria como un solo vector continuo como se explicó en la clase03. Para leer las matrices del SEL, estas deben de estar guardadas en disco en **formato binario** como se explica en http://www.cimat.mx/~alram/met_num/clases/clase03b.pdf (es decir que primero deben de crear

sus archivos binarios). Se recomienda utilizar el código de los ejemplos de la clase03, ahí están prácticamente todas las funciones que necesitan.

iv) Escribir programas en C/C++ que (usando también el almacenamiento en memoria del punto anterior) verifiquen experimentalmente algunas propiedades de álgebra matricial mencionadas en http://www.cimat.mx/~alram/met_num/clases/repasoAL.pdf . Las operaciones entre matrices y vectores (+, -, *, etc.) deben de ser funciones. Verificar experimentalmente:

- Una propiedad de la página 2 del pdf.
- El producto matriz por vector de la página 6 del pdf.
- Dos propiedades de las matrices que tienen inversa de la página 10 del pdf. Proponer las matrices y sus inversas, para ello pueden usar matlab, wolfram-mathematica, buscar en internet, etc.

(posteriormente en las siguientes clases usaremos archivo binarios), un ejemplo de una matriz guardada **en formato de texto** de 3x4 está en la liga http://www.cimat.mx/~alram/met_num/clases/mat_texto_3x4.txt.

El formato del archivo para este caso es:

```
n m
a11 a12 . . . a1m
.
.
.
an1 an2 . . . anm
```

Donde n es el numero de renglones y m el numero de columnas.

Una función que lee las matrices **en modo texto** es la siguiente (asegúrate que entiendes cada parte de código de la función). El siguiente código usa la función de la Clase03 en la pagina web del curso.

```
// lee la matriz en el archivo fileName, regresa por la izquierda la
matriz y regresa en los apuntadores nr y nc el numero de renglones y el
numero de columnas de la matriz respectivamente.
```

```
double ** readMatrix(const char* fileName, int *nr, int *nc){
    FILE* fp = fopen(fileName, "rt");
    if (!fp)
        return NULL;

    fscanf(fp, "%d %d", nr, nc);

    double ** mat = createMatrix(*nr, *nc);

    for (int i=0; i<*nr; i++) {
        for (int j=0; j<*nc; j++) {
            double val;
            fscanf(fp, "%f", &val);
            mat[i][j] = val;
        }
    }

    return mat;
}
```

Que se usa como:

```
int nr,nc;  
double ** matr = readMatrix("/ruta/mat_texto_3x4.txt",&nr,&nc);  
printMatrix(matr, nr, nc );
```