**Universidad Simón Bolívar**

**Laboratorio de Cálculo Numérico**

**Período**: Sept- Dic

**Nombre:** Luis Alejandro Vieira Zambrano

**Carnet**: 07-41651

**Laboratorio #3**

**Funciones:** LU.m – LUR.m – backsub.m – forsub.m

**Pregunta 1:**

Script: Lab03P1

**Los resultados se muestran en el Script.**

Al comparar los errores relativos notamos que la solucion de LU tiene un error mayor que el comando de matlab, esto se debe a que los metodos como LU o Gauss son mas inexactos ya que son aproximaciones sin embargo suelen ser un poco mas rapido, y tambien el LU sin pivoteo trae mayores problemas cuando el pivote no es el mayor.

Los resultados difieren por muy poco, pero en el error relativo se observa que LU tuvo un error mayor con respecto a la solucion exacta.

Con el nuevo b, notamos que el problema esta muy mal condicionado, con una muy pequena perturbacion , un error relativo muy bajo, la nuevca solucion tiene un error relativo de 1 y esto es bastante comparado con las soluciones anteriores.

Al ver el numero de condicion de la matriz A esta es alta, 3.220998840821112e+05, muy lejano de 1, y por ende con pequenas perturbacion en los parametros de entrada los resultados cambiaran por mucho.

**Pregunta 2:**

Script: Lab03P2

La solucion con b es:

x1 =

0.006583963241436

0.017819918297615

0.029183554661252

0.040677807534815

0.052305714511559

La solucion con c es:

x2 =

0.006583963241436

0.017819918297615

0.029183554661252

0.040677807534815

0.052305714511559

La solucion con d es:

x3 =

0.114354702221662

-0.042948668564855

0.082051331435145

0.070557078561581

0.093812892515070

La manera en que programe mi funcion LU, el back y el forward me permitieron facilmente sin cambiar codigo hallar la solucion de y'\*A = c