Informe Laboratorio 1

Presentado a: Prof. Hernán Darío Vargas Cardona, PhD

Estudiantes:
Nicolás Ibagón Rivera
Santiago Uribe Pastás

Pontificia Universidad Javeriana
Computación Científica
Santiago de Cali, Colombia
Septiembre de 2020

Abstract- En el siguiente documento se presenta un informe acerca de la implementación de sistemas punto flotante y de sistemas de ecuaciones lineales en lenguaje de programación Python. Entre las implementaciones se encuentran Gauss y Gauss-Jordan para sistema de ecuaciones lineales. Posteriormente se presentarán los métodos utilizados para la realización de dichas implementaciones junto con los resultados y su respectivo análisis.

Introducción

En el laboratorio se realizaron dos implementaciones correspondientes a la unidad 1 y 2, siendo estos sistemas de punto flotante (normalizado) y sistemas de ecuaciones lineales respectivamente.

- Sistemas de punto flotante:

  Para llevar a cabo una representación de los números reales en un ordenador se ve necesario el tener una manera de codificar la coma raíz, para ello se debe de descomponer el número en dos partes, siendo:

$$x = \pm \left(d_0 + \frac{d_1}{\beta} + \frac{d_2}{\beta^2} + ... + \frac{d_{t-1}}{\beta^{t-1}} \right) \beta^e,$$

  donde $0 \leq d_i \leq \beta - 1$, $i = 0, ..., t - 1$, $L \leq e \leq U$.
  La parte en paréntesis se conoce como la mantisa (m) y e es el exponente.

  - Una mantisa: Se le conoce como coeficiente o significado. Es la cantidad de dígitos significativos que posee un número.
  - Un exponente: Indica dónde se coloca el punto decimal o binario en relación al inicio de la mantisa.

- Sistema de ecuaciones lineales:

  Un sistema de ecuaciones lineales mxn consta de m ecuaciones y n incógnitas. Dicho sistema puede llegar a no tener solución y de tenerla puede poseer una única solución o soluciones infinitas. En este laboratorio nos vamos a centrar en aquellos que tienen solución única.

$$\begin{aligned} 2x_1 + 4x_2 - 2x_3 &= 2, \\ 4x_1 + 9x_2 - 3x_3 &= 8, \\ -2x_1 - 3x_2 + 7x_3 &= 10, \end{aligned}$$

Fig1. Ejemplo de un sistema de ecuaciones lineales 3x3

  Para la solución de los sistemas de ecuaciones lineales existen diversos métodos, en este laboratorio se presentarán los que son resueltos con Gauss y Gauss-Jordan.

Materiales y Métodos

Para la realización del laboratorio se hizo uso del lenguaje de programación Python junto con librerías que facilitaron en algunos aspectos la implementación del código, como lo son numpy y matplotlib. Además se hizo uso de las diapositivas y videos de clase para solucionar dudas acerca de teoría y métodos de resolución.

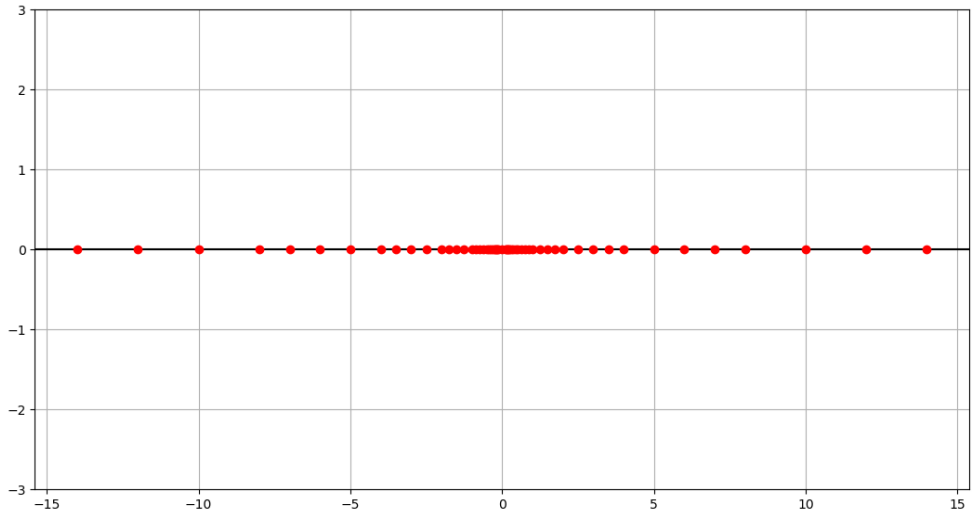
Para la implementación de el sistema punto flotante, al tener un β fijo ($\beta=2, sistema\,binario$), se hizo uso del método de la tabla de verdad para facilitar el cálculo de la mantisa. Esta tabla de verdad contiene $2\square^{t-1}$filas y $t-1$columnas (pues en un sistema binario el dígito d_0 siempre es 1). Para cada fila de la tabla se procede de la siguiente manera; si se tiene un 1 en la fila i-ésima se procede a hacer la operación $1/\beta^j$ (donde $j\in[0,t-1]$, o en otras palabras j hace referencia a la columna respectiva) y se va acumulando el valor para dicha fila; este valor corresponde a la mantisa de cada $x_i$del sistema punto flotante; y el anterior valor multiplicado por su respectivo β^e donde $e\in[L,U]$. Después de hecho todos esos cálculos se agrega el número tanto positivo como negativo al sistema (lo anterior corresponde al $\pm$ de la ecuación). Las demás propiedades del sistema (N, UFL, OFL) se calcularon con las fórmulas vistas en clase.

Para la implementación de Gauss en sistemas de ecuaciones lineales como se nombro anteriormente se centrara la atención en sistemas de ecuaciones lineales *nxn* con única solución. Primeramente con el determinante se revisa de entrada si la matriz A tiene solución o no; de ser así se procede de la siguiente forma. Se escoge el pivote el cual se encuentra en la diagonal de la matriz ($A[i][j] con\, i=j$) esto con la intención de "aniquilar" lo que está en la columna debajo del pivote. Dicha "aniquilación" se realiza con lo llamado matrices de eliminación elemental. Estas matrices, llamemoslas E, son una matriz identidad de tamaño *nxn* con la variación de que en la columna que se quiere "aniquilar" se coloca un valor para que esta "aniquilación" ocurra al multiplicar la matriz $E\cdot A$. Dicho valor se calcula con la siguiente fórmula: $E[i][j]=A[i][j]/-pivote$. En este punto surge la duda de qué sucede si $pivote=0$, la anterior fórmula sería indefinida. Por lo que para este problema lo que se hace es permutar entre filas hacia abajo (pues hacia arriba el sistema ya está "procesado") hasta que $pivote\neq 0$, si esto no ocurre el sistema no tendrá solución. Una vez creada la matriz E se procede a multiplicar en ambos lados del sistema $E\cdot A\,x=E\cdot b$ (para que el sistema no se altere). Lo anterior se hace para todos los pivotes, una vez terminado ese proceso el resultado será una matriz triangular superior. Ya con esta matriz triangular superior solo resta realizar sustitución sucesiva hacia atrás para encontrar la solución al sistema $Ax=b$.

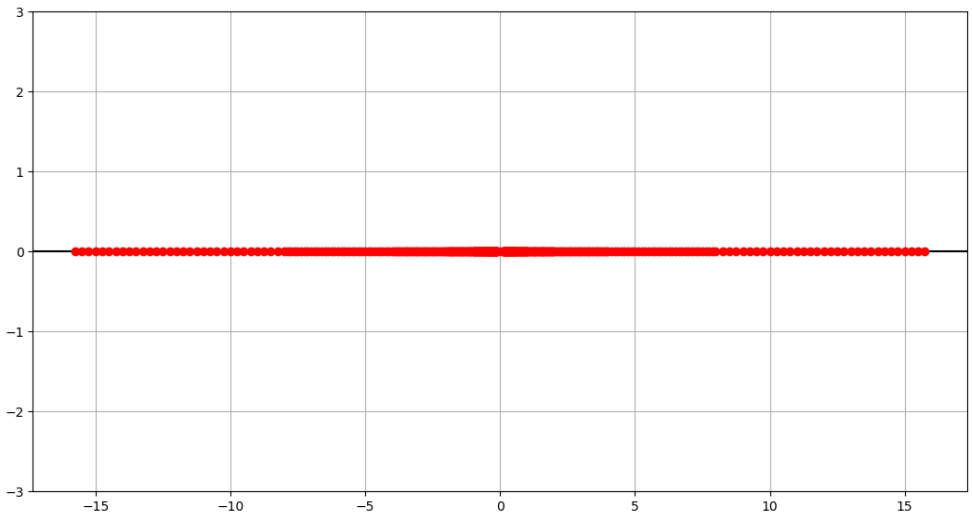
Para la implementación del Gauss-Jordan, como es sabido la idea de este es quedar con una matriz identidad y el vector b con la solución. Para esto se hace lo anteriormente mencionado en el proceso Gauss para tener una matriz triangular superior. Una vez obtenida la matriz A en forma de triangular superior, se procede a llevar a cabo el mismo proceso pero de manera inversa para llevar a la matriz A a una matriz triangular inferior, para así obtener una matriz diagonal D donde $D[i][j]=0\,si\,i\neq j$. Obtenida ya esta matriz D se procede a dividir cada valor del vector b con su respectivo valor en la matriz diagonal D ($b[i]/D[i][i]$) y el valor de esta operación se guarda en el vector solución x.

Resultados y Análisis

- **Sistema punto flotante:** Se corrió el código implementado para 2 simulaciones diferentes. La gráfica 1 corresponde a los parámetros $\beta=2, t=3, L=-3, U=3$ y la gráfica 2 corresponde a los parámetros $\beta=2, t=6, L=-3, U=3$



Gráfica 1. Simulación 1



Gráfica 2. Simulación 2

Para $t=3$, las propiedades del sistema punto flotante son: $N=57$, $UFL=0{,}125$ y $OFL=14{,}0$
Para $t=6$, las propiedades del sistema punto flotante son: $N=449$, $UFL=0{,}125$ y $OFL=15{,}75$

Como se puede observar en la gráfica y fácilmente en las propiedades, a mayor tamaño de t (el cual es la precisión del sistema) hay mayor cantidad de números en el sistema y gráficamente se ve más como una línea recta debido a la cantidad de

puntos que hay y la pequeña distancia que hay entre estos puntos.

- **Sistema de ecuaciones lineales:** Se corrió el código para distintos casos (diferentes n). Para revisar que el código diera la respuesta correcta se probó con sistemas 2x2, 3x3, 4x4 y 5x5. Para probar eficiencia se generaron matrices random de orden grande para fijarse en cuánto tiempo tarda el código en dar una respuesta. Los valores se presentan en la siguiente tabla:

orden sistema \ método	Gauss	Gauss-Jordan
$n=3$	0.0 seg	0.0 seg
$n=10$	0.0 seg	0.0 seg
$n=50$	0.00598096847534 seg	0.00997376441955 seg
$n=100$	0.04767274856567 seg	0.07975339889526 seg
$n=200$	0.36490631103515 seg	0.74600076675415 seg
$n=300$	1.93779850006103 seg	3.54087853431701 seg
$n=400$	4.67171359062194 seg	9.52365446090698 seg
$n=500$	11.1165103912353 seg	21.5273771286010 seg

Tabla 1. Toma de tiempos para diferentes valores de n en Gauss y Gauss-Jordan

Como vemos en la tabla de tiempos el método de Gauss es más eficiente respecto a tamaños grandes de n, esto se debe a que para el Gauss únicamente hay que volver la matriz A en una matriz triangular superior y posteriormente aplicar sustitución sucesiva hacia atrás. En cambio, para el método Gauss-Jordan hay que volver la matriz A en una matriz diagonal, por lo que hay que hacer el procedimiento de volver matriz triangular superior y después matriz triangular inferior (o viceversa) y al final hacer las n divisiones para encontrar cada valor solución del vector x, por lo que todo este procedimiento toma más iteraciones que el Gauss. Por ende, en la práctica se sugiere usar la implementación de Gauss.

Conclusiones

La utilización del lenguaje de programación python y las librerías de matplotlib y numpy junto con su previo aprendizaje facilitó en gran medida las implementaciones de sistemas de punto flotante y sistema de ecuaciones lineales, al mismo tiempo la implementación de las funciones mencionadas anteriormente se hicieron en base a las diapositivas vistas en clase. Algunas de las funciones las cuales facilitaron el laboratorio son:

- $np.linalg.det(A)$donde A es una matriz : Esta función permite el cálculo de la determinante.

- $np.matmul(A,B)$donde A y B son matrices : Esta función permite el cálculo de la multiplicación de dos matrices.

- $np.random.rand(n,n)$ donde n es el tamaño de la matriz: Esta función permite generar matrices de tamaño n con números aleatorios.

Para la toma de tiempos se debe tener en cuenta que el lenguaje python es de tipo interpretado, esto significa que el código fuente es traducido por un intérprete a un lenguaje el cual sea entendible para la máquina paso a paso. De igual modo se debe de considerar que el procesador en el cual se ejecutaron los códigos se encuentran dadas por un Intel Core i5 de 7ma generación.

Referencias

https://matplotlib.org/

https://numpy.org/

http://puntoflotante.org/formats/fp/

http://orga.blog.unq.edu.ar/wp-content/uploads/sites/5/2018/05/orga_apuntePtoFlotante.pdf