Prueba N°1 Algoritmos Numéricos

Bastián Loyola (16 horas) *Departamento de Ingeniería Informática*
Universidad de Santiago de Chile, Santiago, Chile
bastian.loyola@usach.cl

***Resumen*—Este documento describe la estructura y contenido a considerar en la construcción de un informe de laboratorio para el curso de Estructura de Computadores. El escrito debe ser construido empleando LaTeX y no debe superar las 4 páginas de extensión, incluyendo: texto, fórmulas, algoritmos, tablas, imágenes y referencias. El documento comienza con un ENCABEZADO que contiene un TÍTULO con el número y nombre de la actividad, indicando entre paréntesis el número de horas que cada integrante destinó al desarrollo de esta. Posteriormente señala el nombre de los autores del documento y sus correos electrónicos. La sección de RESUMEN corresponde a un párrafo que NO debe tener más de 250 palabras y hace referencia al contexto de trabajo, indica el método aplicado, el resultado obtenido y la principal contribución. Debe cautivar al lector e invitarlo a leer el documento. Al esta sección se debe incluir 3 a 5 palabras o conceptos claves asociados a la actividad.**

***Palabras claves*—palabra clave 1, palabra clave 2, palabra clave 3 y palabra clave 4.**

I. INTRODUCCIÓN

Durante el curso de algoritmos numéricos, se logro estudiar distintos algoritmos para la resolución de problemas matemáticos, una clase a destacar son aquellos algoritmos para la resolución de sistemas de ecuaciones, ya que estos se pueden aplicar para estudiar fenómenos físicos como la difusión de fotones de luz, de está forma considerando el fenomeno fisico mencionado se tendrá como objetivo implementar y evaluar tres algoritmos para la resolución de sistemas de ecuaciones, estos algoritmos serán evaluados respecto a sus errores minimos obtenidos, el tiempo empleado, los costos operacionales y finalmente las soluciones como tal de los sistemas, esperando claras diferencias en los tres primeros y siendo similar o idénticos en cuanto a los resultados todo esto mediante MatLab.

II. ANTECEDENTES

II-A. Marco Teórico

II-A1. Matlab: Es una plataforma de programación y calculo numérico, que incluye un entorno de desarrollo integrado (IDE)[1], debido a sus funciones para el cálculo y matemáticas, es ideal para diseñar e implementar algoritmos numéricos para resolución de problemas matemáticos

II-A2. Algoritmo numérico: Un algoritmo es según la RAE: Çonjunto ordenado y finito de operaciones que permite hallar la solución de un problema"[2] y numérico como: "Perteneciente o relativo a los números"[3]; entonces un algoritmo numérico debería ser aquel conjunto de operaciones relacionadas con los números que nos permite hallar la solución de un problema matemático.

II-B. Ecuaciones

A continuación se verán las ecuaciones utilizadas

II-B1. Difusión de fotones de luz: Se debe resolver los sistemas de ecuaciones $A\Phi$= b, que se obtienen de la formulación variacional de la ecuación de disfusión de fotones de luz, ecuación 1.

$$-k(r)\phi(r)+u_a(r)\phi(r)+\frac{1}{c}*\frac{\delta\phi(r)}{\delta t}=q_0(r) \tag{1}$$

II-C. Soluciones conocidas de sistemas de Ecuaciones

Si se tiene el sistema de ecuaciones Ax = b, existen formas de encontrar la soluciones si A cumple ciertas condiciones.

II-C1. A es diagonal: Las soluciones están dadas por 2

$$x_i=\frac{b_i}{a_{ii}} \qquad i=1,...,n \tag{2}$$

II-C2. A es ortogonal: Las soluciones están dadas por 3

$$x=A^{-1}b=A^tb \tag{3}$$

II-C3. A es triangular superior: Las soluciones están dadas por la sustitución regresiva a continuación su algoritmo 1

Algoritmo 1 Sustitución Regresiva

Input: A Matriz triangular superior , b vector.
Output: x Solución del sistema de ecuaciones
$[M,N]\leftarrow size(A)$
$tam\leftarrow length(b)$
if $M!=N||tam!=M$ **then**
  $Return-1$
end if
$x\leftarrow[0,0,0,...,0],delargotam$
$x[tam]\leftarrow b[tam]/A[tam,tam]$
for $k=tam-1:1;k--$ **do**
  $suma\leftarrow(x[k+1:tam])'*(A[k,k+1:tam])'$
  $x[k]\leftarrow(b[k]-suma)/A[k,k]$
end for
$Return(x)$

II-C4. A es triangular inferior: Las soluciones están dadas por la sustitución progresiva a continuación su algoritmo 2

II-D. Métodos

A continuación de describirán los metodos empleados para encontrar las soluciones de los sistemas de ecuaciones

Algoritmo 2 Sustitución Progresiva

Input: A Matriz triangular inferior , b vector.
Output: x Solución del sistema de ecuaciones
$[M, N] \leftarrow size(A)$
$tam \leftarrow length(b)$
if $M! = N||tam! = M$ **then**
  $Return - 1$
end if
$x \leftarrow [0, 0, 0, ..., 0], delargotam$
$x[1] \leftarrow b[1]/A[1, 1]$
for $k = 2 : tam; k + +$ **do**
  $suma \leftarrow (x[1 : k - 1])' * (A[k, 1 : k - 1])'$
  $x[k] \leftarrow (b[k] - suma)/A[k, k]$
end for
$Return(x)$

II-D1. Método directo: LU: Este método consiste en descomponer la matriz A, en dos matrices a las cuales se les conoce el resultado como las que vimos en el sector II-C, esté método factoriza A en dos matrices L que es una triangular inferior y U que es una triangular superior, por ende se tiene Ax=b $\rightarrow$ LUx=b, donde se debe emplear sustitución progresiva para resolver Lz=b, al obtener los valores de z se debe realizar la sustitución regresiva para desarrollar Ux=z, el algoritmo para factorizar A en LU es el siguiente llamado Metodo de Crout 3

II-D2. Método iterativo: Gauss-Seidel: Este método al igual que otros métodos iterativos, consta con un tolerancia y una cantidad máxima de iteraciones, definidas por uno mismo, para definir las soluciones se hace uso de una solución inicial parcial, con la cual se irá trabajando y acercando cada vez al valor final, dada la ecuación 4, s indicará la iteración en la que se encuentran, si está llega a ser el valor máximo asignado se detendrá pese a no haber encontrado la solución final, esto puede provocar errores si no sé define un buen numero de iteraciones, por otro lado tenemos la tolerancia que está mientras sea menor al error encontrado se seguirá iterando.

$$x_i^{(s+1)} = \frac{b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(s)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(s)}}{a_{ii}} \qquad i = 1, ..., n \tag{4}$$

II-D3. Método Normal: HouseHolder: Este método consiste en descomponer la matriz A, en dos matrices a las cuales se les conoce el resultado como las que vimos en el sector II-C, esté método factoriza A en dos matrices Q que es una ortogonal y R que es una triangular superior, por ende se tiene Ax=b $\rightarrow$ QRx=b, donde Q al ser ortogonal se tiene que emplear Qz=b $\rightarrow$ z=Qt̂b, al obtener los valores de z se debe realizar la sustitución regresiva para desarrollar Ux = z, el algoritmo para factorizar A en QR es el siguiente 4, esté contará con dos algoritmos houseCero quese encargará de crear un vector de 0 usando una matriz HouseHolder y houseMult que realizará la postmultiplicación de una matriz con una matriz houseHolder, información extraída desde [4].

Algoritmo 3 Método de Crout

Input: A Matriz cuadrada
Output: L Matriz triangular inferior, U Matriz triangular superior
if $nargin < 5$ **then**
  return(-1)
else
  $[cols, filas] \leftarrow size(A)$
  if $cols \neq filas$ **then**
    return(-1)
  else
    $L \leftarrow zeros(cols)$
    $U \leftarrow zeros(cols)$
    $U[1, 1] \leftarrow 1$
    $L[1, 1] \leftarrow A[1, 1]$
    for $i = 2 : cols$ **do**
      $U[i, i] \leftarrow 1$
      $L[i, 1] \leftarrow A[i, 1]$
      $U[1, i] \leftarrow A[1, i]/L[1, 1]$
    end for
    for $j = 2 : cols$ **do**
      for $i = j : cols$ **do**
        $sumal \leftarrow 0$
        $sumau \leftarrow 0$
        for $k = 1 : j - 1$ **do**
          if $(L[i, k] \neq 0)\&\&(U[k, j] \neq 0)$ **then**
            $sumal \leftarrow sumal + L[i, k] * U[k, j]$
          end if
          if $(L[j, k] \neq 0)\&\&(U[k, i] \neq 0)$ **then**
            $sumau \leftarrow sumau + L[j, k] * U[k, i]$
          end if
        end for
        $L[i, j] \leftarrow A[i, j] - sumal$
        if $(j < cols)\&\&(i > j)$ **then**
          $U[j, i] = (A[j, i] - sumau)/L[j, j]$
        end if
      end for
    end for
  end if
end if
$Return[L, U]$

III. Materiales y métodos

III-A. Materiales

Los materiales utilizados para creación y desarrollo del presente documento: -Software LaTeX, para la creación del presente documento
-El ambiente donde fue realizado el documento y la simulación fue una computadora con sistema operativo Windows 10 de 64 bits.
-IDE Matlab, para la implementación y prueba de los algoritmos.

Algoritmo 4 House Holder

Input: A Matriz cuadrada
Output: Q matriz ortogonal, R matriz triangular superior
$[M, N] \leftarrow size(A)$
$S = min(n, m-1)$
$Q = eye(m, m)$
for $k = 1 : S$ **do**
  $[x, sigma] \leftarrow housecero(A(k:m, k))$
  $Q[1:m, k:m] \leftarrow housemult(Q[1:m, k:m], x)$
  $A[k, k] \leftarrow sigma;$
  $s1 \leftarrow size(x)$
  $A[k+1:m, k] \leftarrow x[2:s1]$
  $v[k] \leftarrow x[1]$
  $beta \leftarrow 2/(x' * x)$
  for $j = k+1 : n$ **do**
    $s \leftarrow 0$
    $s \leftarrow s + x[1:m-k+1]' * A[k:m, j]$
    $s \leftarrow beta * s$
    $A[k:m, j] \leftarrow A[k:m, j] - s * x[1:m-k+1]$
  end for
end for
$R \leftarrow triu(A)$
$Return(x)$

III-B. Métodos

El procedimiento seguido, fue primero obtener el sistema de ecuaciones para eso, se cargaron unos archivos proporcionados por el profesor, que contenia un archivo A1089 que contenia una constante para todos los sistemas y un archivo b1089, que contenia un vector el cual giraremos para así obtener sistemas de ecuaciones distintos por cada rotaccón, este proceso se repetirá 100 veces a traves de un ciclo, ya teniendo los distintos sistemas Ax = b, toca obtener su solución para ello se emplearon 3 algoritmos uno de método directo, uno iterativo y uno normal(ortogonal), estos son respectivamente el Metodo de Crout para obtener una factorización LU (3), el metodo de Gauss-Seidel (4) y el metodo houseHolder para obtener una factorización QR (4).
Los tres métodos fueron implementados dos veces, la primera vez para calcular los tiempos de ejecucción y la segunda para hacer los cálculos de costos operacionales, para comparar dichos métodos se realizarán un gráfico de barras donde estarán los valores de los errores mínimos, tiempos de ejecución y costos operacionales, por otro lado se realizará un diagrama de caja para revisar los resultados de cada uno de los metodos, se espera que estos sean iguales o similares ya que están resolviendo los mismos sistemas de ecuaciones. Tanto el metodo de Crout (Directo) como el de HouseHolder (Normal) realizarán la factorización fuera del ciclo para obtener los 100 sistemas de ecuaciones distintos, dado que la factorización depende de A el cual no cambiará para ningún sistema de ecuación y así ahorrar recursos, dado que los sistemas contarán con 1089 variables se espera que ocupen altos tiempos de ejecución y costos operacionales; por otro lado el metodo de Gauss-Seidel (Iterativo) contemplará un nivel de tolerancia y cantidad de iteración maxima que afectará directamente a los resultados, se recomienda usar un bajo nivel de tolerancia y una gran cantidad de iteraciones.

IV. Resultados

A continuación se describirán los resultados obtenidos, se puede apreciar los tiempos de ejecución son lo más destacable son los tiempos de ejecucción dado que el metodo normal aumenta considerablemente la ejecucion de todo, esto se puede apreciar en la imagen 1 en los tiempos del método normal, son realmente elevados en comparación a los otros, gran parte de este tiempo se destina a calcular las matrices Q y R.
Por otro lado hablando de los errores mínimos encontrados, el método directo es aquel que tiene error más grande, respecto a costos operacionales se puede apreciar nuevamente en la imagen 1 que el método iterativo es aquel que posee mayor costo, esto sin embargo pudo ser incluso mayor dado que las iteraciones máximas definidas impiden que el método siga, esto afecta considerablemente en dos cosas más el tiempo de ejecución que también debió ser mayor y las soluciones encontradas, las media de las soluciones puede ser apreciado en el diagrama de caja 2, donde el método iterativo presenta un valor demasiado distinto al método directo y normal, siendo que estos debiesen ser iguales o similares, esto sucede debido al limite de iteraciones provocando que los valores obtenidos sean incorrectos.

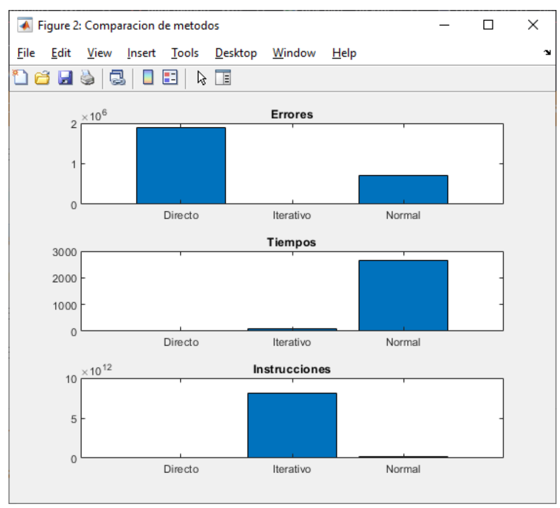

Figura 1: Comparación de métodos

V. Conclusiones

Como se puede apreciar, cada uno de los métodos tiene sus propias desventajas frente al resto, el método LU (directo) siendo de los más rápido, pero aquel que presenta el mayor error; el método householde (normal), presenta bajo costo operacional y error, pero en temas de tiempo es el peor y por ultimo tenemos al método de Gauss-Seidel (iterativo) es aquel más sencillo de implementar y dadas las condiciones puntuales de tolerancia e iteraciones maximas, logró conseguir el más bajo de los errores, sin embargo su costo operacional y la media de sus soluciones demuestran que no es el mejor método en la resolucion de sistemas de ecuaciones, ya que si bien aumentando el numero de

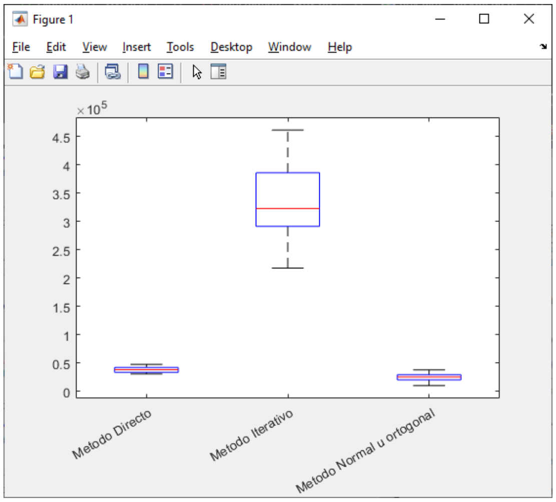

Figura 2: Diagrama de caja de las soluciones

iteraciones mejorará el valor de las soluciones, tanto como tiempos de ejecucion y costos operacionales se elevarán mucho.

Referencias

[1] Mathworks, "¿qué es matlab?" 7/12/2021. [Online]. Available: https://la.mathworks.com/products/matlab.html

[2] . REAL ACADEMIA ESPAÑOLA, "Algoritmo," *Diccionario de la lengua española, 23.ª ed., [versión 23.4 en línea].*, 7/12/2021.

[3] ——, "Numérico," *Diccionario de la lengua española, 23.ª ed., [versión 23.4 en línea].*, 7/12/2021.

[4] O. Rojas, "Unidad iii: Sistemas de ecuaciones ii," 19/12/2021. [Online]. Available: https://uvirtual.usach.cl/moodle/pluginfile.php/424627/mod_resource/content/2/SE-II.pdf

Contribución

Conceptualización, M.V.C. y J.G.; Metodología, M.V.C. y J.G.G.; software, M.V.C.; validación, J.G.G.; análisis, M.V.C.; investigación, M.V.C. ; recursos, J.G.G; preparación de datos, M.V.C. y J.G.G.; escritura de borrador, M.V.C. y J.G.G.; revisión de la escritura y formalización, M.V.C.; formato y visualización, J.G.G; supervisión, M.V.C.; Todos los autores y autoras han leído y están de acuerdo con la entrega de este documento, así como el trato de principios éticos.