Tarea 3-Métodos numéricos

1st Daniel Vallejo Aldana
Maestría en Ciencias de la Computación
Centro de Investigación en Matemáticas
daniel.vallejo@cimat.mx

***Resumen*—El presente trabajo consta de dos partes principales, la primera de ellas se refiere a la aproximación del cálculo de las derivadas de una función f mediante diferencias finitas. La segunda parte se refiere a métodos de factorización de matrices cuadradas de $n \times n$ para solución de sistemas de ecuaciones. En este trabajo se abordarán los métodos de eliminación gaussiana, factorización LU, método de Doolittle y descomposición LDL, así como casos especiales como sustitución hacia adelante y hacia atrás para matrices triangulares y también se tratará el caso especial en donde la matriz cuadrada es una matriz diagonal.**

***Index Terms*—Diferencias finitas, Cálculo de derivadas, Sistemas de ecuaciones, Factorización de matrices**

I. INTRODUCCIÓN

I-A. *Diferencias finitas*

El método de aproximación del valor de la derivada de una función f, en un punto $a \in \mathbb{R}$ mediante diferencias finitas, resulta conveniente cuando se desconoce una expresión analítica de la derivada de la función. No obstante, dicho método depende en gran medida del tamaño de paso h que se le asigne, de acuerdo a [1], el error de estimación del valor de la derivada en un punto a es de $\frac{M|h|}{2}$ siendo M una cota para el valor de la derivada. En este trabajo se abordará de manera superficial la forma de utilizar el método de diferencias finitas para poder encontrar los máximos y los mínimos de una función f.

I-B. *Factorización de matrices y solución de sistemas de ecuaciones*

Cuando se tiene un sistema de la forma $Ax = b$ donde A es una matriz cuadrada de dimensión $n \times n$, y x, b son vectores de tamaño n, existen diversos métodos de solución del sistema dependiendo de las condiciones que satisfaga la matriz A. En el presente trabajo consideraremos los métodos de factorización de Doolittle y factorización LU. En ambos métodos se desea descomponer a la matriz A en un producto de matrices tales que $A = LU$, con L una matriz triangular inferior y U una matriz triangular superior. Dentro de dicha factorización se requieren dos casos especiales de solución al sistema de ecuaciones que constan en los métodos de sustitución hacia adelante y sustitución hacia atrás que aplica para matrices triangulares inferiores y triangulares superiores respectivamente, dichos métodos también serán descritos en el presente trabajo. Así mismo consideraremos una extensión al método de factorización LU, donde se agrega una matriz diagonal D de tal forma que ahora A se puede descomponer como el producto de las matrices $A = LDU$. A medida de comparación se implementó el método tradicional de eliminación Gaussiana con pivoteo parcial para comparar los tiempos de ejecución de los diferentes métodos.

II. MÉTODO/ALGORITMO

II-A. *Diferencias finitas*

De acuerdo a la definición formal de la derivada de una función f sea $f : \mathbb{R} \to \mathbb{R}$ una función continua a, entonces decimos que f es derivable en a si

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

existe.

En el método de diferencias finitas se utiliza dicha definición para hacer una estimación numérica de la derivada de una función en un punto a. Para esto necesitamos un tamaño de paso h para poder evaluar la función f en los puntos $f(x+h)$ y $f(x-h)$ según sea el caso. Las expresiones para las derivadas de primer y segundo orden de una función f en un punto x con tamaño de paso h son las siguientes

$$\begin{aligned} f'(x) &= \frac{f(x+h) - f(x)}{h} \\ f''(x) &= \frac{f'(x+h) - f'(x-h)}{h} \\ &= \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} \end{aligned}$$

En este caso se tomó $h = \frac{x}{1000} + 0{,}005$ obtenido mediante resultados experimentales. Los pseudocódigos de la estimación de las derivadas de una función f en un punto x para orden 1 y 2 se muestran a continuación.

Algorithm 1 Diferencias finitas de primer orden

Require: f, x, h
Ensure: $f'(x)$
  Evaluar $f(x+h)$,$f(x)$
  return $\frac{f(x+h)-f(x)}{h}$

Algorithm 2 Diferencias finitas de segundo orden

Require: f, x, h
Ensure: $f''(x)$
  Evaluar $f(x+h)$,$f(x)$,$f(x-h)$
  return $\frac{f(x+h)-2f(x)+f(x-h)}{h^2}$

II-B. *Factorización de Matrices y Solución de Sistemas de Ecuaciones*

II-B1. Solución de sistemas de ecuaciones para matrices diagonales: El primer caso a considerar es cuando la matriz A es una matriz diagonal, es decir, que las entradas $a_{i,j}$ de la matriz son 0, cuando $i \neq j$. Para que el sistema de ecuaciones de una matriz diagonal tenga solución se requiere que $a_{i,i} \neq 0 \forall i \in \{1, \dots, n\}$. Para toda entrada i se tiene que

$$x_i = \frac{b_i}{a_{i,i}}$$

El algoritmo de solución de un sistema de ecuaciones para una matriz diagonal se muestra a continuación. Para denotar la entrada de un vector usaremos la notación x_i, mientras que para denotar el vector completo usaremos **x**. Las matrices se declaran con letra mayúscula.

Algorithm 3 SOLVEDIAGONAL

Require: A matriz diagonal, **b**
Ensure: **x** tal que A **x** = **b**
  Inicializar **x**
  for i from 1 to n **do**
    $x_i = \frac{b_i}{a_{i,i}}$
  end for
  return **x**

II-B2. Solución de sistemas de ecuaciones para matrices triangulares inferiores: Consideremos ahora una matriz A, triangular inferior es decir que las entradas $a_{i,j}$ de la matriz son 0 para $j > i$. En este caso también tenemos que asegurar que las entradas en la diagonal de la matriz sean diferentes de 0 para que el sistema pueda tener solución. El paso de actualización para el vector x_i se muestra a continuación.

$$x_i = \frac{b_i - \sum_{j=1}^{i-1} a_{i,j} x_j}{a_{i,i}}$$

El algoritmo correspondiente para resolver un sistema de ecuaciones para una matriz triangular inferior es el siguiente

Algorithm 4 FORWARD-SUBSTITUTION

Require: A triangular inferior, **b**
Ensure: **x** tal que A**x** = **b**
  Inicializar **x**
  for i in 1 to n **do**
    $tmp = 0$
    for j tal que $j < i$ **do**
      $tmp = tmp + a_{i,j} x_i$
    end for
    $x_i = \frac{b_i - tmp}{a_{i,i}}$
  end for
  return **x**

II-B3. Solución de sistemas de ecuaciones para matrices triangulares superiores: Si A es una matriz de $n \times n$ triangular superior, entonces las entradas $a_{i,j}$ de la matriz son 0 cuando $j < i$. En ese caso, la primera entrada que se actualiza es la n-ésima entrada x_i y el paso de actualización es el siguiente

$$x_i = \frac{b_i - \sum_{j=i+1}^{n} a_{i,j} x_j}{a_{i,i}}$$

Notemos que para x_n queda solamente la expresión de la forma

$$x_n = \frac{b_n}{a_{n,n}}$$

El pseudocódigo de dicho método se muestra a continuación

Algorithm 5 BACKWARD-SUBSTITUTION

Require: A triangular superior, **b**
Ensure: **x** tal que A**x** = **b**
  Inicializar **x**
  for i from n to 1 **do**
    $tmp = 0$
    for $j > i$ **do**
      $tmp = tmp + a_{i,j} x_j$
    end for
    $x_i = \frac{b_i - tmp}{a_{i,i}}$
  end for
  return **x**

II-B4. Solución de sistemas de ecuaciones para matrices generales con eliminación Gaussiana y pivoteo parcial: El método de eliminación Gaussiana se basa en convertir una matriz general A en una matriz triangular superior mediante operaciones renglón, estas operaciones son substraer un renglón de otro y multiplicar dicho renglón por un escalar α. Así mismo, el pivoteo parcial permite intercambiar renglones, dicha operación no cambia el valor de las soluciones.

En el pivoteo parcial, si estamos procesando el renglón i, buscamos en los renglones faltantes el vector tal que

$a_{j,i} > a_{i,i}$ e intercambiamos dichos renglones al igual que los valores correspondientes b_j y b_i.

Si estamos procesando el reglón i, la actualización para la entrada de alguno de los renglones $k \in \{i+1, \ldots, n\}$ es la siguiente

$$a_{k,j} = a_{k,j} - \frac{a_{k,i}}{a_{i,i}}$$

Para que este método funcione se requiere que la matriz A sea una matriz de rango completo.

El pseudocódigo del método de eliminación Gaussiana sin pivoteo parcial es el siguiente

Algorithm 6 ELIMINACION-GAUSSIANA

Require: A matriz general, **b**
Ensure: **x** tal que $A\mathbf{x} = \mathbf{b}$

```
Inicializar x
for i from 1 to n do
  for j from i+1 to n do
    value = a_{j,i} / a_{i,i}
    for k in i+i to m do
      a_{j,k} = a_{j,k} - value * a_{i,j}
    end for
    b_j = b_j - b_i * value
  end for
end for
x=BACKWARD-SUBSTITUTION(A,b)
return x
```

II-B5. Factorización LU y solución de sistemas de ecuaciones: El método de factorización LU, consiste en descomponer una matriz general A como el producto de dos matrices LU, en donde L es triangular superior y U es triangular superior. Deben de satisfacer además que alguna de ellas debe de tener $1's$ en su diagonal. En este trabajo supondremos que L es la matriz que tiene unos en la diagonal. Las entradas de la primera columna de U y del primer renglón de L, están dadas de la siguiente manera

$$u_{1,j} = \frac{a_{j,1}}{l_{1,1}}$$
$$l_{j,1} = \frac{a_{j,1}}{u_{1,1}}$$

Donde $u_{1,1} = a_{1,1}$

Luego, para las entradas de la diagonal tenemos el siguiente paso de actualización

$$u_{i,i} = a_{i,i} - \sum_{k=1}^{i-1} l_{i,k} u_{k,i}$$

Para las entradas fuera de la diagonal tenemos las siguientes actualizaciones

$$u_{i,j} = \frac{1}{l_{i,i}} \left[a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} u_{k,j} \right]$$
$$l_{i,j} = \frac{1}{u_{i,i}} \left[a_{j,i} - \sum_{k=1}^{i-1} l_{j,k} u_{k,i} \right]$$

Una vez que se tienen las matrices L y U, procedemos con los métodos de sustitución hacia adelante y hacia atrás para poder encontrar las soluciones **x**. El proceso de solución es el siguiente

$L\mathbf{y} = \mathbf{b}$ con sustitucion hacia adelante
$Q\mathbf{x} = \mathbf{y}$ con sustitucion hacia atrás

El pseudocódigo de la factorización LU se muestra a continuación.

Algorithm 7 LU-FACTORIZATION

Require: A
Ensure: L,U

```
Inicializar L y U del del mismo tamaño de A
u_{1,1} = a_{1,1}
for i from 1 to n do
  l_{i,i} = 1
end for
for j from 2 to n do
  u_{1,j} = a_{1,j} / l_{1,1}
  l_{j,1} = a_{j,1} / u_{1,1}
end for
for i from 2 to n-1 do
  u_{i,i} = a_{i,i} - \sum_{k=1}^{i-1} l_{i,k} u_{k,i}
  for j from i+1 to n do
    u_{i,j} = 1/l_{i,i} [a_{i,j} - \sum_{k=1}^{i-1} l_{i,k} u_{k,j}]
    l_{i,j} = 1/u_{i,i} [a_{j,i} - \sum_{k=1}^{i-1} l_{j,k} u_{k,i}]
  end for
end for
u_{n,n} = a_{n,n} - \sum_{k=1}^{n-1} l_{n,k} u_{k,n}
return L y U
```

Dentro del algoritmo LU existe una variante en donde queremos encontrar una matriz diagonal D de tal forma que podemos descomponer A como el producto de las tres matrices $A = LDU$, para esto necesitamos convertir la matriz U en producto de dos matrices DU donde ahora U tiene unos en su diagonal, la actualización de las entradas de D y de U son las siguientes

$$d_{i,i} = u_{i,i}$$
$$u_{i,j} = \frac{u_{i,j}}{d_{i,i}}$$

Para extraer estas dos matrices mostramos a continuación el pseudocódigo.

Algorithm 8 FACTORIZAR U EN DU

Require: U
Ensure: D,U
  Inicializar D una matriz de 0's
  for i from 1 to n **do**
    $d_{i,i} = u_{i,i}$
    for j from i+1 to n **do**
      $u_{i,j} = \frac{u_{i,j}}{d_{i,i}}$
    end for
  end for
  return D, U

II-B6. Factorización de matrices usando el método de Doolittle: El método de Doolittle es una variante de la factorización LU, donde se quiere evitar el proceso de la eliminación Gaussiana. Las actualizaciones de las entradas de la matriz L y U se muestran a continuación.

$$u_{i,k} = a_{i,k} - \sum_{j=1}^{i-1} l_{i,j} u_{j,k}$$
$$l_{k,i} = \frac{a_{k,i} - \sum_{j=1}^{i-1} l_{k,j} u_{j,i}}{u_{i,i}}$$

Notemos que, nuevamente L, tiene que tener $1's$ en las diagonales. El pseudocódigo para el algoritmo de factorización usando el método de Doolittle es el siguiente

---

Algorithm 9 DOOLITTLE-FACTORIZATION

Require: A
Ensure: L, U
  Inicializar L y U tal que L tiene 1's en la diagonal
  for i from 1 to n **do**
    for k from i to m **do**
      $u_{i,k} = a_{i,k} - \sum_{j=1}^{i-1} l_{i,j} u_{j,k}$
    end for
    for k from i to m **do**
      if k ==i **then**
        $l_{k,i} = 1$
      else
        $l_{k,i} = \frac{a_{k,i} - \sum_{j=1}^{i-1} l_{k,j} u_{j,i}}{u_{i,i}}$
      end if
    end for
  end for
  return L,U

III. RESULTADOS

III-A. Estimación de máximos y mínimos por diferencias finitas

Se consideró la función $f(x) = x^4 + 1$ como función de prueba para la estimación del valor de la derivada en un intervalo $[a, b]$ usando el método de diferencias finitas, para este caso se consideró que el valor de la derivada en el punto x es cercano a 0 cuando $|f'(x)| < | \times 10^{-8}$ El punto encontrado por el algoritmo fue $x = 0$, donde además $f''(x) > 0$ por lo que corresponde a un mínimo de f, en la Figura 1 se muestran las gráficas de las derivadas aproximadas por el método de diferencias finitas.

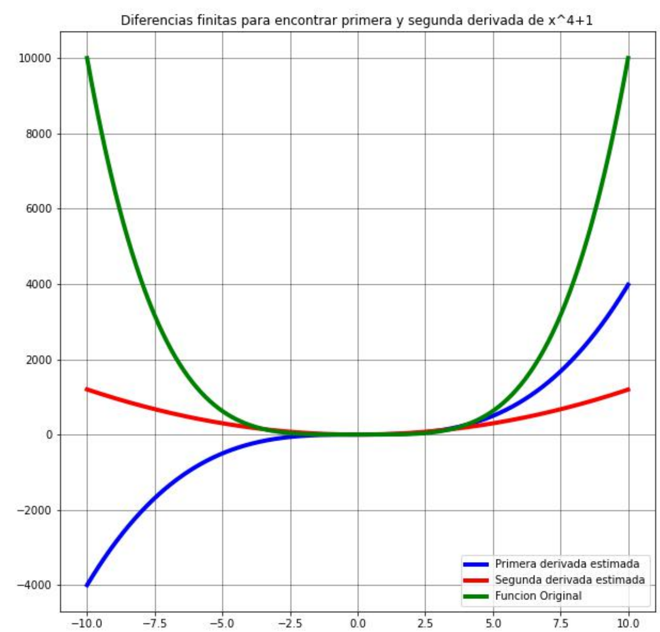

Figura 1. Estimaciones de las derivadas en un intervalo, $[a, b]$ la línea verde es la función original, la azul corresponde a la primera derivada y la roja a la segunda derivada

III-B. Solución de sistemas de ecuaciones con factorización de matrices

En la Tabla I comparamos los métodos de Doolittle, Factorización LU, Eliminación Gaussiana y Factorización LDL en cuanto a tiempo computacional y error de precisión de la forma $\|A\mathbf{x} - \mathbf{b}\|_2$

Cuadro comparativo de métodos		
Método	**Tiempo de ejecución (segs)**	$\|A\mathbf{x}-\mathbf{b}\|_2$
Eliminación Gaussiana	**268.911**	$1{,}76 \times 10^{-9}$
Factorización LU	461.54	$1{,}072 \times 10^{-9}$
Factorización LDL	463.192	$6{,}78 \times 10^{-8}$
Método de Doolittle	424.527	$2{,}414 \times 10^{-9}$

Cuadro I

COMPARATIVA DE LOS DIFERENTES MÉTODOS IMPLEMENTADOS PARA RESOLVER SISTEMAS DE ECUACIONES EN UNA MATRIZ COMPLETA A

Al ser métodos exactos, todos los errores deben de ser de magnitud muy pequeña, lo cual vemos que se cumple, en este caso al no haber necesidad de llevar a cabo alguna operación de pivoteo el método de eliminación gaussiana obtuvo mejores resultados en cuanto a tiempo de ejecución respecto a los demás métodos comparados.

IV. CONCLUSIONES

Con base en los resultados obtenidos en el presente trabajo, concluimos que el método de diferencias finitas para estimación de derivadas muestra buenos resultados cuando se desconoce la expresión analítica de la derivada de una función f. Así mismo, se concluye que los métodos de factorización de matrices son métodos útiles para resolver sistemas de ecuaciones de la forma $Ax = b$, sin embargo, en su mayoría tienen costos computacionales altos, especialmente al trabajar con matrices muy grandes.

REFERENCIAS

[1] Richard L Burden, J Douglas Faires, and Annette M Burden. *Numerical analysis*. Cengage learning, 2015.