PROYECTO FINAL
Métodos Numéricos.

7/Diciembre/2018

Zepeda Montiel Antonio Aldair - NUA: 461770
Ramos Soto Michel Abraham - NUA: 768936

1 Introducción

Los lenguajes de programación representan un papel muy importante en el desarrollo y evolución de la ingeniería de múltiples campos. La implementación de modelos matemáticos en c y matlab, permiten a los estudiantes e investigadores alcanzar metas y resultados que permiten la optimización de diversos procesos.

El análisis, procesamiento de información, y obtención de resultados, resultan ser tareas muy laboriosas y a veces complicadas para un ingeniero, llegando al punto de que es tanta la información obtenida que es casi imposible procesarla con una unica persona bajo su estudio. Por ello ¿, es de vital importancia que los lenguajes de programación optimicen y faciliten dichas tareas al ser humano, en especial, a los ingenieros para sus multiples aplicaciones.

Como muestra importante de ello, se han desarrollado diferentes algoritmos capaces de realizar tareas de aproximación, análisis y optimización mediante el uso de diversos métodos numéricos, los cuales desarrollaremos en el presente documento.

2 Objetivos

Implementar y desarrollar los diferentes algoritmos aplicados durante el curso para la solución de métodos numéricos en dos lenguajes de programación diferentes.

3 Procedimiento

Como punto inicial, se muestran de manera general los diferentes métodos utilizados en el desarrollo de este proyecto:

- **Método de bisección.**

  Este es uno de los metodos mas sencillos y de facil intuicion para resolver ecuaciones en una variable, tambien conocido como Metodo de Intervalo Medio. Se basa en el teorema del valor intermedio (TVI), el cual establece que toda funcion continua f en un intervalo cerrado [a,b] toma todos los valores que se hallan entre f(a) y f(b). Esto es que todo valor entre f(a) y f(b) es la imagen de al menos un valor en el intervalo [a,b]. En caso de que f(a) y f(b) tengan signos opuestos, el valor cero sera un valor intermedio entre f(j) y f(e), por lo que con certeza existe un p en [a,b] que cumple f(p)=0. De esta forma, se asegura la existencia de al menos una solucion de la ecuacion f(a)=0.

- **Método de falsa posición.**

  El método de la falsa posición pretende conjugar la seguridad del método de la bisección con la rapidez del método de la secante. Este método, como en el método de la bisección, parte de dos puntos que rodean a la raíz f(x) = 0, es decir, dos puntos x0 y x1 tales que f(x0)f(x1) menor que 0. La siguiente aproximación, x2, se calcula como la intersección con el eje X de la recta que une ambos puntos. La asignación del nuevo intervalo de búsqueda se realiza como en el método de la bisección: entre ambos intervalos, [x0,x2] y [x2,x1], se toma aquel que cumpla f(x)f(x2) menor que 0.

- **Método de la secante.**

  Este método necesitará dos aproximaciones iniciales de la raíz para poder inducir una pendiente inicial.

  El método se basa en obtener la ecuación de la recta que pasa por los puntos (xn -1, f(xn -1)) y (xn, f(xn)). A dicha recta se le llama secante por cortar la gráfica de la función. Como ejemplo, se toman los puntos iniciales x0 y x1, se construye una línea por los puntos (x0, f(x0)) y (x1, f(x1)). Posteriormente se escoge como siguiente elemento de la relación de recurrencia, xn+1, la intersección de la recta secante con el eje de abscisas obteniendo la fórmula, y un nuevo valor. Seguimos este proceso, hasta llegar a un nivel suficientemente alto de precisión (una diferencia lo suficientemente pequeñas entre xn y xn-1).

- **Método de Ridder.**

  En el análisis numérico, el método de Ridders es un algoritmo de búsqueda de raíces basado en el método de posición falsa y el uso de una función exponencial para aproximar sucesivamente una raíz de una función continua f (x). El método se debe a C. Ridders.

El método de Ridders es más simple que el método de Muller o el método de Brent, pero con un rendimiento similar. La siguiente fórmula converge cuadráticamente cuando la función se comporta bien, lo que implica que el número de dígitos significativos adicionales encontrados en cada paso se duplica aproximadamente; pero la función debe evaluarse dos veces para cada paso, por lo que el orden general de convergencia del método es raíz de dos. Si la función no se comporta bien, la raíz permanece entre corchetes y la longitud del intervalo de paréntesis al menos a la mitad en cada iteración, por lo que se garantiza la convergencia.

- **Método de Newton-Raphson.**

  El método de Newton-Raphson es un método abierto, en el sentido de que no está garantizada su convergencia global. La única manera de alcanzar la convergencia es seleccionar un valor inicial lo suficientemente cercano a la raíz buscada. Así, se ha de comenzar la iteración con un valor razonablemente cercano al cero (denominado punto de arranque o valor supuesto). La relativa cercanía del punto inicial a la raíz depende mucho de la naturaleza de la propia función; si ésta presenta múltiples puntos de inflexión o pendientes grandes en el entorno de la raíz, entonces las probabilidades de que el algoritmo diverja aumentan, lo cual exige seleccionar un valor supuesto cercano a la raíz. Una vez que se ha hecho esto, el método linealiza la función por la recta tangente en ese valor supuesto. La abscisa en el origen de dicha recta será, según el método, una mejor aproximación de la raíz que el valor anterior. Se realizarán sucesivas iteraciones hasta que el método haya convergido lo suficiente.

- **Método de Muller.**

  Este método utilizado para encontrar raíces de ecuaciones con raíces múltiples, y consiste en obtener los coeficientes de la parábola que pasa por tres puntos elegidos. Dichos coeficientes son sustituidos en la formula cuadrática para obtener el valor donde la parábola intersecta al eje X; es decir, la raíz estimada. La aproximación se puede facilitar, si se escribe la ecuación de la parábola en una forma conveniente.

  Una de las mayores ventajas de este método, es que al trabajar con la formula cuadrática es posible localizar tanto raíces reales, como raíces complejas.

- **Método de Bairstow.**

En análisis numérico, el método de Bairstow es un algoritmo eficiente de búsqueda de las raíces de un polinomio real de grado arbitrario. Es un método iterativo, basado en el método de Müller y de Newton Raphson.
El algoritmo de Bairstow tiene orden de convergencia cuadrático como el método de Newton, excepto en el caso de que el polinomio tenga factores cuadráticos de multiplicidad superior a 1, pudiendo ser el orden de convergencia menor.

- **Eliminasión por método de Gauss.**

  En forma general este método propone la eliminación progresiva de variables en el sistema de ecuaciones, hasta tener sólo una ecuación con una incógnita. Una vez resuelta esta, se procede por sustitución regresiva hasta obtener los valores de todas las variables. El método de solución será simplificar las ecuaciones, de tal modo que las soluciones se puedan identificar con facilidad.

- **Descomposición LU.**

  El método de descomposición LU para la solución de sistemas de ecuaciones lineales debe su nombre a que se basa en la descomposición de la matriz original de coeficientes (A) en el producto de dos matrices (L y U).

  Esto es: A = LU
  Donde:

  - L - Matriz triangular inferior
  - U - Matriz triangular superior con todos los elementos de la diagonal principal iguales a 1.

- **Método de Gauss-Seidel.**

  En análisis numérico el método de Gauss-Seidel es un método iterativo utilizado para resolver sistemas de ecuaciones lineales.
  Aunque este método puede aplicarse a cualquier sistema de ecuaciones lineales que produzca una matriz (cuadrada, naturalmente pues para que exista solución única, el sistema debe tener tantas ecuaciones como incógnitas) de coeficientes con los elementos de su diagonal no-nulos, la convergencia del método solo se garantiza si la matriz es diagonalmente dominante o si es simétrica y, a la vez, definida positiva.

- **Interpolación.**

  En el subcampo matemático del análisis numérico, se denomina interpolación a la obtención de nuevos puntos partiendo del conocimiento de un conjunto discreto de puntos. En estos metodos, es frecuente disponer de un cierto número de puntos obtenidos por muestreo o a partir de un

experimento y pretender construir una función que los ajuste.
Si tenemos una función cuyo cálculo resulta costoso, podemos partir de un cierto número de sus valores e interpolar dichos datos construyendo una función más simple. En general, por supuesto, no obtendremos los mismos valores evaluando la función obtenida que si evaluamos la función original, si bien dependiendo de las características del problema y del método de interpolación usado la ganancia en eficiencia puede compensar el error cometido.

- **Regresión por mínimos cuadrados.**

  Mínimos cuadrados es una técnica de análisis numérico enmarcada dentro de la optimización matemática, en la que, dados un conjunto de pares ordenados —variable independiente, variable dependiente— y una familia de funciones, se intenta encontrar la función continua, dentro de dicha familia, que mejor se aproxime a los datos (un "mejor ajuste"), de acuerdo con el criterio de mínimo error cuadrático.
  En su forma más simple, intenta minimizar la suma de cuadrados de las diferencias en las ordenadas (llamadas residuos) entre los puntos generados por la función elegida y los correspondientes valores en los datos. Específicamente, se llama mínimos cuadrados promedio (LMS) cuando el número de datos medidos es 1 y se usa el método de descenso por gradiente para minimizar el residuo cuadrado. Se puede demostrar que LMS minimiza el residuo cuadrado esperado, con el mínimo de operaciones (por iteración), pero requiere un gran número de iteraciones para converger.

4 Resultados

Para los resultados de los métodos aplicados previamente, presentamos las siguientes ejecuciones de los métodos implementados y sus resultados obtenidos:

- **Método de bisección.**

```
-----------------------------------------------------------------------
Ciclo          a               b               Xn              Error
-----------------------------------------------------------------------
  0        0.0000          5.0000          2.5000          1.0000La raíz
  1        0.0000          5.0000          2.5000          0.0000
-----------------------------------------------------------------------

 La raíz de la ecuación es: 2.500000
 Se encontró en 1 iteraciones
```

- **Método de falsa posición.**

```
------------------------------------------------------------------------
Ciclo          a              b              Xn             Error
------------------------------------------------------------------------
 0        0.0000         5.0000         0.1333         1.0000
 1        0.0000         0.1333         0.1333        81.1111
 2        0.7059         0.1333         0.7059        45.2012
 3        0.7059         0.4861         0.4861         6.7818
 4        0.7059         0.5215         0.5215         0.5582
 5        0.7059         0.5244         0.5244         0.0454
 6        0.7059         0.5247         0.5247         0.0037
 7        0.7059         0.5247         0.5247         0.0003
------------------------------------------------------------------------

 La raíz de la ecuación es: 0.524695
 Se encontró en 7 iteraciones
```

- **Método de la secante.**

```
----------------------------------------------------------------------------
Ciclo           a                b                Xn                Error
----------------------------------------------------------------------------
 0         0.0000           5.0000           0.0000             1.#INF
 1         5.0000           0.0000           0.0000            -1.#IND
----------------------------------------------------------------------------

 La raíz de la ecuación es: 0.000000
 Se encontró en 1 iteraciones
```

- **Método de Ridder.**

```
Ridder

Usaremos la forma: Ax^B+Cx^D+E dentro de un rango [Xo,Xi]
Ingrese su valor de A:
5
Ingrese el valor de B:
2
Ingrese el valor de C:
5
Ingrese el valor de D:
4
Ingrese el valor de E:
0
Ingrese su valor inicial de Xo:
0
Ingrese su valor inicial de Xi:
5


a2=223.750      a1=-468.750     a0=0.000


La ecuación 5.0x^2.0+5.0x^4.0+0.0 tiene raíz: 2.0950

Desea regresar al menu principal?
Sí, presione '1'        No presione '2'
```

- **Método de Newton-Raphson.**

```
--------------------------------------------------------
Ciclo          a              Xn              Error
--------------------------------------------------------
  0         5.0000          1.0000          0.0000
  1         8.3805         17.0319          0.0000
  2        10.1009          8.6276          0.0000
  3         9.2986          1.3454          0.0000
  4         9.4255          0.0071          0.0000
  5         9.4248          0.0000          0.0000
--------------------------------------------------------

 La raíz de la ecuación es: 9.424778
 Se encontró en 5 iteraciones
```

- **Método de Bairstow.**

```
Bairstow

Introduzca el grado del polinomo:
3
Introduzca los coeficientes del polinomio (El primer coef
Introduzca el coeficiente de x^2:
5
Introduzca el coeficiente de x^1:
3
Introduzca el coeficiente de x^0:
5
Introduzca el valor inicial de r:
0
Introduzca el valor inicial de s:
1
Introduzca el valor del error de convergencia:
0.002
Introduzca numero maximo de iteraciones:
100

Su polinomio es: 1.000X^45.000X^33.000X^25.000X^1Raices:
X0=-0.1923 1.0000iX1=-0.1923 -1.0000iX2=-5.000000


Desea regresar al menu principal?
Sí, presione '1'        No presione '2'
```

- **Eliminasión por método de Gauss.**

```
Su sistema de ecuaciones ingresado es:
5.000X3+3.000X2+2.000X1+=5.000
7.000X3+8.000X2+4.000X1+=5.000
4.000X3+2.000X2+1.000X1+=5.000

La matriz ingresada es:

5.000   3.000   2.000

7.000   8.000   4.000

4.000   2.000   1.000


La matriz identidad es:
1.000   0.000   0.000

0.000   1.000   0.000

-0.000  -0.000  1.000

El valor de las incognitas es :
X1=1.667

X2=-0.000

X3=-1.667
```

- **Descomposición LU.**

```
Su sistema de ecuaciones ingresado es:
2.000X3+5.000X2+2.000X1+=5.000
3.000X3+5.000X2+2.000X1+=5.000
5.000X3+3.000X2+5.000X1+=5.000

La matriz ingresada es:

2.000   5.000   2.000

3.000   5.000   2.000

5.000   3.000   5.000

El valor de las incognitas es :
X1=0.000
X2=0.789
X3=0.526
```

- **Método de Gauss-Seidel.**

```
Su sistema de ecuaciones ingresado es:
2.000X3+5.000X2+2.000X1+=5.000
3.000X3+6.000X2+3.000X1+=5.000
2.000X3+5.000X2+2.000X1+=5.000

------------------------------------------------
Iteración 0:
(1.0018,0.0000)(2.5000,0.0000)(-0.4167,0.0000)
Iteración 1:
(1.0018,0.0000)(3.5417,0.0000)(-0.9375,0.0000)
Resultados:
X0=1.0018
X1=3.5417
X2=-0.9375

Se realizó en 2 iteraciones.
```

- **Regresión por mínimos cuadrados.**

```
Regresión por mínimos cuadrados

Cuántos puntos tiene? (mínimo 2,máximo 20).
2
Ingrese el valor de X1:
3
Ingrese el valor de Y1:
4
Ingrese el valor de X2:
5
Ingrese el valor de Y2:
3

Usted ingresó los siguientes pares de números:

1) [5.000,3.000]
2) [-0.000,-0.000]      La pendiente ajustada por mínim

El punto de corte por mínimos cuadrados es b=5.500000


Desea regresar al menu principal?
Sí, presione '1'        No presione '2'
```

5 Tabla comparativa

Finalmente, una vez obtenidos los resultados de los valores evaluados la simulación, podemos apreciar variación entre los diferentes valores obtenidos y de los diferentes métodos proporcionados a mayor profundidad mediante los valores de las tablas comparativas que se presentarán a continuación:

Determinacion de Raices Algebraicas

Ecuacion	Xi	Xf	Error Perm.	Ciclos	Biseccion	Falsa Posicion	Ridder	NR	Secante	Muller	Bairstow
Sin(x)	0	5	0.002	1000	2.5	23,212	239,912	94,247	95,234	97,541	-1,000
5x^2+5x-4	0	5	0.002	1000	0.312495	0.29932	0.30123	0.5246	0.60232	0-7564	0.1923

Como podemos apreciar, la variacion entre algunos de los metodos implementados puede presentar diferentes cambios entre los valores obtenidos de los calculos, todo esto debido a las metodologías que utiliza en cada uno de ellos, sin embargo, podemos ver que se presentan ciertas similitudes entre los valores de los metodos de la secante, NR y Muller, esto debido a que utilizan metodologías que comparten similitudes en sus argumentos y procedimientos.

Solucion de sistemas de ecuaciones algebraicas					
Ecuacion	Xi	Xf	Error Perm.	Eliminacion de Gauss	
5x^2+5x-4	0	5	0.002	1,667	
7x^2+8x-1	0	5	0.002	0.000	
4x^2+1x-1	0	5	0.002	1,666	
Ecuacion	Xi	Xf	Error Perm.	Descomposicion LU	
5x^2+5x-4	0	5	0.002	0.000	
7x^2+8x-1	0	5	0.002	0.789	
4x^2+1x-1	0	5	0.002	0.526	
Ecuacion	Xi	Xf	Error Perm.	Matrices y el método de Gauss-Seidel	
5x^2+5x-4	0	5	0.002	1	
7x^2+8x-1	0	5	0.002	4	
4x^2+1x-1	0	5	0.002	0	

En el caso de los metodos de la tabla anterior, los resultados pueden variar entre los metodos presentados, sin embargo, dichos valores encontrados pueden obtener una cercanía bastante considerable, lo cual puede implicar un acercamiento muy proximo al resultado, a diferencia de los metodos anteriores debido a su poca presición en el cálculo de sus estimaciones.

6 Conclusiones

El desarrollo de diversos métodos numéricos en los lenguajes de programación proporcionados, es una herramienta que no sólo muestra resultados en el campo de la programación, sino que nos permite expandir sus aplicaciones a procesos de mecánica, óptica, reconocimiento de imagenes, en los cuales se tenía un previo conocimiento de su funcionamiento, pero que al combinarlos con dichas metodologías en programación, nos permite obtener resultados todavía más precisos y acertados.

Todo el desarrollo de este proyecto, ha sido la recuperación de los diversos métodos vistos en clase, y nos posibilitan evidenciar la calidad y el proceso de desarrollo que nosotros como estudiantes hemos podido alcanzar, sin dejar de considerar que no existe un límite para el uso del conocimiento adquirido en este curso.

7 Bibliografía

References

[1] Métodos Numéricos para ingeniería. - http://disi.unal.edu.co/ lctorress/MetNum/LiMetNu2.pdf

[2] Programación y métodos numéricos. - http://ocw.upm.es/matematica-aplicada/programacion-y-metodos-numericos

[3] Análisis Numérico. - https://es.wikipedia.org/wiki/Análisis-numérico

[4] Conceptos y operaciones básicas - http://matlabinformatica.wikidot.com/conceptos-y-operaciones-basicas