Universidad Corporativa Coppel | Programas Académicos

Ingeniería en Desarrollo de Software

Actividad | 3 | Método de Jacobi y Gauss-Seidel

Nombre del curso

Ingeniería en Desarrollo de Software

academiglobal

TUTOR: Miguel Angel Rodriguez Vega.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 10 de Noviembre del 2024

Índice

Introducción

Ya estamos en la última actividad de nuestra materia de Métodos numéricos, y, como es de costumbre tendremos que resolver los ejercicios correspondientes de esta sección, en donde veremos nuevas formas de resolver las ecuaciones que se nos pueden presentar en las diversas formas de problemas. En el software de RStudio se programará el método de Bisección utilizando los ciclos, comandos y la sintaxis correcta para resolver el ejercicio, después de ello, de acuerdo al método de "Jacobi" resolver un sistema de ecuaciones lineal despejando la "x" para obtener una ecuación distinta lista para ser ejecutada con el método anterior dicho, cuando se saquen las capturas del proceso y cuando se llegue al resultado final se pasará a mostrar el siguiente de la lista que es el de Gauss-Seidel, como ya se mencionó se realizaran los mismos pasos para mostrar todo el proceso en el que se ejecuta el método.

Descripción

El método de Aitken es un método de aceleración de la convergencia. Alexander Aitken fue quien introdujo ese método en 1926. Su forma primitiva era conocida por Kōwa Seki a finales del siglo XVII y fue encontrado en la rectificación del círculo, es decir, el cálculo de pi. Este método es muy útil para acelerar la convergencia de una sucesión que converge linealmente, este método es de orden "2" que no necesita cálculo de la derivada. Este método depende totalmente de métodos anteriores como el de bisección o punto fijo para el cálculo de sus propias iteraciones, ya que este sirve solamente en acelerar uno de los procesos. El método Jacobi es un método iterativo, usado para resolver sistemas de ecuaciones lineales del tipo "Ax=b", este método consiste en usar fórmulas como iteración de punto fijo, la base de este método consiste en contribuir una sucesión convergente definida iterativamente. Este método nos ayuda a resolver sistemas de ecuaciones lineales donde el número de ecuaciones es grande y cada ecuación tiene pocas variables.

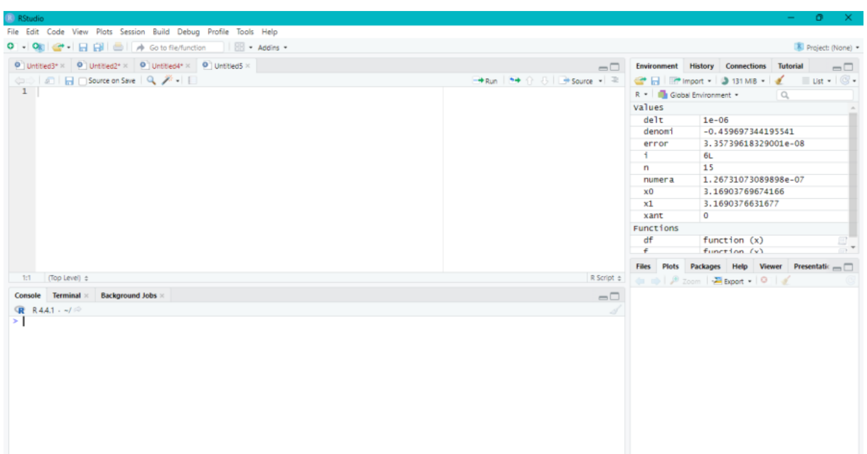
Justificación

En esta actividad en importante mencionar que los métodos que se ven son para que nosotros como aprendices de esta materia nos familiaricemos con ellos, los conozcamos, los estudiemos y nos familiaricemos con todos los tópicos que se ven en todas las actividades de esta materia. Todo tema cuando es nuevo puede resultar tedioso o aburrido en primera instancia, pero si se le da la importancia necesaria para que nos profundicemos en ello resulta que en realidad puede ser muy interesante, puede resultar divertido y en muchas ocasiones hasta un reto para uno mismo, el método de “Jacobi” nos enseña de una manera sencilla y amigable el cómo trabajar con las ecuaciones al igual que el de “Gauss-Seidel”, son métodos que se tienen que conocer para que estemos en sintonía con todo ello y que a futuro podamos ponerlo en práctica para resolver algún problema de la vida diaria que se pueda presentar.

Desarrollo

Método de Bisección:

Para ello abrimos el software de RStudio:



Colocamos una variable junto con la función que se quiere resolver:

```
# Método de Bisección
polinomio = function(x){
  f= x - 2*x^3-8*x^2+10*x-15
  f
}
```

Generaremos un valor de “a”, de “b”, una tolerancia y “n” cantidad de números:

```
Raiz_bisección= function (a, b, tol, n){
```

Definimos variables:

```
  # Paso 1
  i= 1
  Fa= polinomio(a)
```

Utilizamos un proceso:

```
 # Paso 2
 while(i<=n){
```

Definimos nuevamente los procesos para que estos se ejecuten de acuerdo a lo que se quiere llegar:

```
# Paso 3
p= a + (b-a)/2
Fp= polinomio(p)
print(c(i,p))
```

Vamos por el paso “4”:

```
 # Paso 4
 if(Fp==0 | (b-a)/2 < tol){
   return(p)
 }
```

El paso “5”:

```
 # Paso 5
 i=i+1
```

El “6”:

```
  # Paso 6
  if(Fp*Fa>0){
    a=p;
    Fa=Fp
  }else{
    b=p
  }
}
```

Y por último el paso número “7”:

```
  # Paso 7
  return(paste( "El  metodo fallo luego de ", n, "iteraciones."))

}
```

Si nosotros seleccionamos toda la función nos arrojara en la consola el proceso:

```
+     # Paso 6
+     if(Fp*Fa>0){
+       a=p;
+       Fa=Fp
+     }else{
+       b=p
+     }
+   }
+
+   # Paso 7
+   return(paste( "El  metodo fallo luego de ", n, "iteraciones."))
+
+ }
>
```

Mencionamos los números en los diferentes campos del comando:

```
> Raiz_bisección(0, 1, 0.000001, 10)
```

Con las iteraciones que mencionamos, por ejemplo, se dijo que a las “10” iteraciones el método se detendría, en este caso paso lo siguiente:

```
[1] 1.0 0.5
[1] 2.00 0.75
[1] 3.000 0.875
[1] 4.0000 0.9375
[1] 5.00000 0.96875
[1] 6.000000 0.984375
[1] 7.0000000 0.9921875
[1] 8.0000000 0.9960938
[1] 9.0000000 0.9980469
[1] 10.0000000  0.9990234
[1] "El  metodo fallo luego de  10 iteraciones."
```

Con esto podemos observar que el comando sirve y se ejecuta en las diferentes maneras en las que se escribió el código, por ejemplo, el número de decimales, entre otros.

Interpretación de resultados:

Método de Jacobi:

El método de "Jacobi" lo realizaremos en una plantilla de Excel, para ello nos dirigiremos a ver la ecuación que nos pide la actividad:

$$\begin{cases} 3x - y - z = 1 \\ -x + 3y + z = 3 \\ 2x + y + 4z = 7 \end{cases}$$

Después de ello despejaremos la "x" para tener una ecuación diferente y que nos ayudara a resolver el problema por el método de "Jacobi", la primera "x" queda de esta manera:

$$x = \frac{+y + z + 1}{3}$$

La segunda "x" que en este caso sería la "y" quedaría de la siguiente manera:

$$y = \frac{+x - z + 3}{3}$$

La tercera "x" que sería la "z" quedaría de la siguiente manera:

$$z = \frac{-2x - y + 7}{4}$$

Y con esto tendríamos el método de "Jacobi" que es el siguiente:

$$x°1 = 0;\ x°2 = 0;\ x°3 = 0;$$

Nos dirigimos a nuestra tabla de Excel para plasmar esas ecuaciones en la plantilla:

Creamos nuestra plantilla en donde organizaremos lo necesario para tener una coherencia:

Metodo de Jacobi	$x°1 = 0; x°2 = 0; x°3 = 0;$

$$x = \frac{+y+z+1}{3}$$

$$y = \frac{+x-z+3}{3}$$

$$z = \frac{-2x-y+7}{4}$$

Acomodamos nuestra tabla con las iteraciones, las variables y el error de las variables:

Iteraciones	X	Y	Z	Error X	Error Y	Error Z

Colocamos las primeras "x" con el valor de "0" que nos marca el Método de "Jacobi":

Iteraciones	X	Y	Z
0	0	0	0

Resolvemos el primer despeje, en el que "x" es igual a "y" más "z" más "1" entre "3":

Iteraciones	X	Y	Z
0	0	0	0
1	=(G5+H5+1)/3		1.75

Y nos da el siguiente resultado:

0.33333333

Ahora pasamos con el segundo despeje, en el que “y” es igual a “x” menos “z” más “3” entre “3”:

X	Y	Z
0	0	0
0.33333333	=(F5-H5+3)/3	

Lo que nos da como resultado lo siguiente:

1

Por último, pasamos al despeje final, en el que “z” es igual a “-2x” menos “y” más “7” entre “4”:

X	Y	Z	Error X
0	0	0	
0.33333333	1	=(-2*F5-G5+7)/4	

Lo que nos da como resultado lo siguiente:

1.75

Ahora sacamos el valor absoluto del número en el error seleccionado, que en este caso sería el siguiente:

X	Y	Z	Error X	Err
0	0	0		
0.33333333	1	1.75	=ABS((F6-F5)/F6)	

Simplemente arrastramos la fórmula para que nos ejecute el comando en el error de “y” y “z”:

Error X	Error Y	Error Z
1	1	1

Ahora arrastramos una casilla hacia abajo, si el error disminuye estamos en lo correcto, y así hasta llegar el error que queremos:

0.33333333	1	1.75	1	1	1
1.25	0.52777778	1.33333333	0.733333333	0.89473684	0.3125

Continuamos bajando:

0.33333333	1	1.75	1	1	1
1.25	0.52777778	1.33333333	0.733333333	0.89473684	0.3125
0.9537037	0.97222222	0.99305556	0.310679612	0.45714286	0.34265734

Seguimos bajando:

0.33333333	1	1.75	1	1	1
1.25	0.52777778	1.33333333	0.733333333	0.89473684	0.3125
0.9537037	0.97222222	0.99305556	0.310679612	0.45714286	0.34265734
0.98842593	0.98688272	1.03009259	0.035128806	0.01485536	0.03595506

Observamos que el error sigue disminuyendo:

1	0.33333333	1	1.75	1	1	1
2	1.25	0.52777778	1.33333333	0.733333333	0.89473684	0.3125
3	0.9537037	0.97222222	0.99305556	0.310679612	0.45714286	0.34265734
4	0.98842593	0.98688272	1.03009259	0.035128806	0.01485536	0.03595506
5	1.00565844	0.98611111	1.00906636	0.01713555	0.00078247	0.02083732

Bajamos hasta la casilla número "10":

1	0.33333333	1	1.75	1	1	1
2	1.25	0.52777778	1.33333333	0.733333333	0.89473684	0.3125
3	0.9537037	0.97222222	0.99305556	0.310679612	0.45714286	0.34265734
4	0.98842593	0.98688272	1.03009259	0.035128806	0.01485536	0.03595506
5	1.00565844	0.98611111	1.00906636	0.01713555	0.00078247	0.02083732
6	0.99839249	0.99886403	1.000643	0.007277645	0.01276742	0.00841794
7	0.99983568	0.99924983	1.00108775	0.001443424	0.00038609	0.00044426
8	1.00011253	0.99958264	1.0002697	0.000276818	0.00033295	0.00081782
9	0.99995078	0.99994761	1.00004808	0.000161751	0.00036498	0.00022162
10	0.99999856	0.99996757	1.00003771	4.77788E-05	1.9962E-05	1.0369E-05

Seguimos bajando, en la iteración número "13" obtenemos el error que queríamos en los tres errores:

13	1.00000001	0.99999874	1.00000129	1.416E-06	1.0929E-06	1.0903E-06

Podemos seguir bajando para tratar de llegar al cero en el error, en la iteración número “36” se llega al cero en el error:

36	1	1	1	0	0	0

$$x = \frac{+y+z+1}{3}$$

$$y = \frac{+x-z+3}{3}$$

$$z = \frac{-2x-y+7}{4}$$

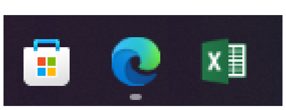
Metodo de Jacobi

$x°1 = 0; x°2 = 0; x°3 = 0;$

Iteraciones	X	Y	Z	Error X	Error Y	Error Z
0	0	0	0			
1	0.333333333	1	1.75	1	1	1
2	1.25	0.5277778	1.3333333	0.733333333	0.894736842	0.3125
3	0.953703704	0.9722222	0.9930556	0.310679612	0.457142857	0.342657343
4	0.988425926	0.9868827	1.0300926	0.035128806	0.014855356	0.035955056
5	1.005658436	0.9861111	1.0090664	0.01713555	0.000782473	0.020837316
6	0.99839249	0.998864	1.000643	0.007277645	0.012767418	0.008417941
7	0.999835677	0.9992498	1.0010877	0.001443424	0.000386092	0.000444261
8	1.000112526	0.9995826	1.0002697	0.000276818	0.000332953	0.000817824
9	0.999950782	0.9999476	1.0000481	0.000161751	0.000364983	0.000221617
10	0.999998561	0.9999676	1.0000377	4.77788E-05	1.99622E-05	1.0369E-05
11	1.000001759	0.999987	1.0000088	3.19737E-06	1.9383E-05	2.88795E-05
12	0.999998593	0.9999976	1.0000024	3.16569E-06	1.06924E-05	6.44436E-06
13	1.000000009	0.9999987	1.0000013	1.416E-06	1.0929E-06	1.09025E-06
14	1.00000001	0.9999996	1.0000003	8.80505E-10	8.3542E-07	9.81226E-07
15	0.999999961	0.9999999	1.0000001	4.86021E-08	3.27369E-07	2.09295E-07
16	1.000000001	1	1	3.93579E-08	5.35644E-08	5.75412E-08
17	0.999999999	1	1	1.3256E-09	3.22997E-08	3.30701E-08
18	0.999999999	1	1	2.56784E-10	1.05815E-08	7.41212E-09
19	1	1	1	1.05645E-09	2.38511E-09	2.51698E-09
20	1	1	1	4.39553E-11	1.19114E-09	1.1245E-09
21	1	1	1	2.22131E-11	3.60183E-10	2.75808E-10
22	1	1	1	2.81248E-11	9.93405E-11	1.01152E-10
23	1	1	1	6.03961E-13	4.30924E-11	3.88976E-11
24	1	1	1	1.39833E-12	1.27646E-11	1.0471E-11
25	1	1	1	7.64611E-13	3.95639E-12	3.89044E-12
26	1	1	1	2.19824E-14	1.55165E-12	1.37135E-12
27	1	1	1	6.01741E-14	4.64517E-13	3.99014E-13
28	1	1	1	2.17604E-14	1.531E-13	1.46105E-13
29	1	1	1	2.33147E-15	5.59552E-14	4.92939E-14
30	1	1	1	2.22045E-15	1.70974E-14	1.5099E-14
31	1	1	1	5.55112E-16	5.77316E-15	5.32907E-15
32	1	1	1	2.22045E-16	1.9984E-15	1.77636E-15
33	1	1	1	1.11022E-16	6.66134E-16	4.44089E-16
34	1	1	1	0	2.22045E-16	4.44089E-16
35	1	1	1	0	1.11022E-16	0
36	1	1	1	0	0	0

Método de Gauss-Seidel:

Para ello seleccionamos el software de Excel:



Tomaremos la plantilla que teníamos del método anterior:

$$x = \frac{+y+z+1}{3}$$

$$y = \frac{+x-z+3}{3}$$

$$z = \frac{-2x-y+7}{4}$$

Metodo de Jacobi

$x°1 = 0; x°2 = 0; x°3 = 0;$

Iteraciones	X	Y	Z	Error X	Error Y	Error Z
0	0	0	0			
1						
2						
3						
4						
5						
6						
7						
8						
9						
10						
11						
12						
13						
14						
15						
16						

La primera "x" se coloca de la misma manera en la que "x" es igual a "y" más "z" más "1" entre "3":

X	Y	Z
0	0	0
=(G5+H5+1)/3		

Para sacar el valor de "y" tenemos que "y" es igual a "x" menos "z" más "3" entre "3", pero esta vez tomaremos el valor de "x" que nos dio anteriormente:

Iteraciones	X	Y	Z
0	0	0	0
1	0.3333333	=(F6-H5+3)/3	

Y hacemos lo mismo con "z":

X	Y	Z	Error
0	0	0	
0.3333333	1.1111111	=(-2*F6-G6+7)/4	

Después obtenemos el valor absoluto:

X	Y	Z	Error X	E
0	0	0		
0.3333333	1.1111111	1.3055556	=((F6-F5)/F6)	

Que en este caso nos tiene que dar “1”:

1	1	1

Ahora simplemente arrastramos las iteraciones para fijarnos que los números disminuyan:

1	0.3333333	1.1111111	1.3055556	1	1	1
2	1.1388889	0.9444444	0.9444444	0.7073171	0.1764706	0.3823529

Seguimos practicando las iteraciones:

3	0.962963	1.0061728	1.0169753	0.1826923	0.0613497	0.0713202
4	1.007716	0.9969136	0.9969136	0.0444104	0.0092879	0.0201238
5	0.9979424	1.0003429	1.0009431	0.0097938	0.0034282	0.0040257

Y bajamos hasta la iteración que nos dice que está en el error que buscamos:

6	1.0004287	0.9998285	0.9998285	0.0024852	0.0005145	0.0011147
7	0.9998857	1.0000191	1.0000524	0.000543	0.0001905	0.0002238
8	1.0000238	0.9999905	0.9999905	0.0001381	2.858E-05	6.192E-05
9	0.9999936	1.0000011	1.0000029	3.017E-05	1.058E-05	1.244E-05
10	1.0000013	0.9999995	0.9999995	7.674E-06	1.588E-06	3.44E-06

Podemos seguir si queremos llegar hasta el cero en nuestro problema:

11	0.9999996	1.0000001	1.0000002	1.676E-06	5.88E-07	6.909E-07
12	1.0000001	1	1	4.263E-07	8.82E-08	1.911E-07
13	1	1	1	9.31E-08	3.267E-08	3.838E-08
14	1	1	1	2.368E-08	4.9E-09	1.062E-08
15	1	1	1	5.172E-09	1.815E-09	2.132E-09
16	1	1	1	1.316E-09	2.722E-10	5.898E-10
17	1	1	1	2.874E-10	1.008E-10	1.185E-10
18	1	1	1	7.31E-11	1.512E-11	3.277E-11
19	1	1	1	1.596E-11	5.602E-12	6.582E-12
20	1	1	1	4.061E-12	8.403E-13	1.82E-12
21	1	1	1	8.867E-13	3.112E-13	3.655E-13
22	1	1	1	2.255E-13	4.663E-14	1.01E-13
23	1	1	1	4.918E-14	1.732E-14	2.021E-14
24	1	1	1	1.232E-14	2.665E-15	5.44E-15
25	1	1	1	2.554E-15	8.882E-16	9.992E-16
26	1	1	1	5.551E-16	0	2.22E-16
27	1	1	1	0	0	0

$$x = \frac{+y + z + 1}{3}$$

$$y = \frac{+x - z + 3}{3}$$

$$z = \frac{-2x - y + 7}{4}$$

Metodo de Gauss-Seidel

$x°1 = 0; x°2 = 0; x°3 = 0;$

Iteraciones	X	Y	Z	Error X	Error Y	Error Z
0	0	0	0			
1	0.3333333	1.1111111	1.3055556	1	1	1
2	1.1388889	0.9444444	0.9444444	0.7073171	0.1764706	0.3823529
3	0.962963	1.0061728	1.0169753	0.1826923	0.0613497	0.0713202
4	1.007716	0.9969136	0.9969136	0.0444104	0.0092879	0.0201238
5	0.9979424	1.0003429	1.0009431	0.0097938	0.0034282	0.0040257
6	1.0004287	0.9998285	0.9998285	0.0024852	0.0005145	0.0011147
7	0.9998857	1.0000191	1.0000524	0.000543	0.0001905	0.0002238
8	1.0000238	0.9999905	0.9999905	0.0001381	2.858E-05	6.192E-05
9	0.9999936	1.0000011	1.0000029	3.017E-05	1.058E-05	1.244E-05
10	1.0000013	0.9999995	0.9999995	7.674E-06	1.588E-06	3.44E-06
11	0.9999996	1.0000001	1.0000002	1.676E-06	5.88E-07	6.909E-07
12	1.0000001	1	1	4.263E-07	8.82E-08	1.911E-07
13	1	1	1	9.31E-08	3.267E-08	3.838E-08
14	1	1	1	2.368E-08	4.9E-09	1.062E-08
15	1	1	1	5.172E-09	1.815E-09	2.132E-09
16	1	1	1	1.316E-09	2.722E-10	5.898E-10
17	1	1	1	2.874E-10	1.008E-10	1.185E-10
18	1	1	1	7.31E-11	1.512E-11	3.277E-11
19	1	1	1	1.596E-11	5.602E-12	6.582E-12
20	1	1	1	4.061E-12	8.403E-13	1.82E-12
21	1	1	1	8.867E-13	3.112E-13	3.655E-13
22	1	1	1	2.255E-13	4.663E-14	1.01E-13
23	1	1	1	4.918E-14	1.732E-14	2.021E-14
24	1	1	1	1.232E-14	2.665E-15	5.44E-15
25	1	1	1	2.554E-15	8.882E-16	9.992E-16
26	1	1	1	5.551E-16	0	2.22E-16
27	1	1	1	0	0	0

¿Cuál método resulto más fácil de utilizar?:

En lo personal el método de “Gauss- “Seidel” fue más rápido y sencillo de utilizar ya que el llegar a al error permitido se hace más rápido de lo habitual, ahorrando desde las “5” iteraciones hasta casi más de “10”.

¿Cuál método es más eficiente?:

El método que me resulto mejor fue el de “Gauss-Seidel”, es un método que nos ahorra pasos a la hora de resolver una ecuación y llegar al error permitido.

Conclusión

En la vida diaria nos enfrentamos problemas en los que nosotros tenemos que usar nuestra inteligencia, resolución y ejecución de situaciones que nos pongan a prueba y todo esto es para eso, hacen que nuestra mente se despeje y sea más fácil llegar a un resultado de las maneras más rápidas y eficientes sin toparnos con demasiados problemas. En esta actividad aprendí que hay métodos que no necesitan procesos tan complejos, pero no por ello sean más eficientes, el método de “newton” requiere de derivadas que puede ser complejo a simple vista pero que es uno de los métodos más eficaces y rápidos que podemos encontrar a diferencia del método de “Jacobi” por ejemplo que no requiere de las derivadas, pero si de despejes en las ecuaciones para llegar a un número determinado de iteraciones que obviamente es mayor al método de “Newton”, es la regla de oro, entre más complicado más eficaz y entre menos complicado es menos eficaz.

Link para GitHub:

https://github.com/Leyzu-Ing/M-todos-Num-ricos.git