



Ingeniería en Seguridad Informática y Redes

Ingeniería en Sistemas Computacionales

Algoritmos de Solución Numérica

Método de Jacobi con Formato

IS727272 - Marco Ricardo Cordero Hernández

SI727576 - Edgar Guzmán Claustro

Jal., 24 de octubre de 2023

Índice

Contexto.....	1
Resultados.....	2
Contenidos del archivo de texto.....	2
Código manejador.....	2
Salida.....	3
Conclusiones.....	4
Bibliografía.....	5

Contexto

El método de Jacobi es un algoritmo numérico utilizado para encontrar aproximaciones sucesivas a las soluciones de un sistema de ecuaciones lineales, especialmente cuando el sistema es grande y disperso. El método tiene la ventaja de ser fácil de implementar y es especialmente útil para sistemas de ecuaciones lineales grandes y dispersas. Sin embargo, su convergencia puede ser lenta en ciertos casos, y existen otros métodos más eficientes para resolver sistemas de ecuaciones lineales.

Dentro de este ejercicio presentado, se le dio formato a la salida del algoritmo de Jacobi hecho en Matlab. Dicha salida fue configurada para asemejarse al ejercicio realizado en clase utilizando la herramienta de ofimática “Excel”. La importancia de mostrar correctamente los datos y de manera legible para el usuario, marca la diferencia entre la utilidad de un algoritmo y uno que no. Ya que ningún algoritmo tiene utilidad, si los resultados no son comprendidos por el usuario.

Resultados

Se modificó el código original proporcionado para mostrar cada iteración en formato de tabla, de forma que todo el proceso detrás del método se muestra puntualmente con el fin de lograr un comportamiento como el que podría verse en la realización del mismo a través de hojas de cálculo. Para la implementación actual, se ha limitado el número de variables en el sistema a sólo 27 de ellas, cada una correspondiente a cada letra del abecedario inglés.

Adicional a ello, se documentaron varias partes del programa provisto, dado que muchas partes de este eran confusas y no se comprendía en su totalidad lo que realizaba en cada segmento.

Código

```
% Restablecer entorno
clear, clc

% Definición de argumentos de entrada
dimension = input('Escriba la dimensión de la matriz(2x2 = 2, 3x3 = 3): ');
ecuaciones = input('Escriba los coeficientes de las ecuaciones ya ordenadas sin resultados [a1 a2 an ; b1 b2 bn ; n1 n2 nn]: ');
resultados = input('Escriba los resultados de las ecuaciones ya ordenadas [r1 ; r2 ; rn]: ');
errorPermitido = input('Escriba el error en decimales: ');

% Variables auxiliares
abc = 'abcdefghijklmnopqrstuvwxyz'; % Identificadores de variables
error = 1; % Error inicial
cnt = 0; % Auxiliar para iteraciones

% Imprimir encabezado de tabla e inicializar resultados
fprintf('n\t');
for i=1:dimension
    valores(i)=0;
    fprintf('%s\t', abc(i));
end
fprintf('Error\n');

while error >= errorPermitido
    for i=1:dimension
        aux = 0; % Auxiliar del método
        for j=1:dimension
            % Si no es parte de la diagonal, sustituir valor
            if j ~= i
                aux = aux + ecuaciones(i,j) * valores(j);
            end
        end

        % Sustituir valores de iteración anterior
        x_Nueva(i) = (resultados(i,1) - aux)/(ecuaciones(i,i));
        Error_Ind(i) = abs((x_Nueva(i) - valores(i))/x_Nueva(i));
    end

    % Actualizar error y mostrar resultados de iteración
    error = sum(Error_Ind(1:dimension))/dimension;
    fprintf('%d\t', cnt);
    for i=1:dimension
        valores(i) = x_Nueva(i);
        fprintf('%-.5f\t', valores(i));
    end
end
```

```

    cnt = cnt + 1;
    fprintf('\t%.5f\n', error);
end

```

Resultado

```

Escriba la dimensión de la matriz(2x2 = 2, 3x3 = 3): 3
Escriba los coeficientes de las ecuaciones ya ordenadas sin resultados [a1 a2 an ; b1 b2 bn ; n1 n2 nn]: [5 -2 1;-1 -7 3;2 -1 8]
Escriba los resultados de las ecuaciones ya ordenadas [r1 ; r2 ; rn]: [3;-2;1]
Escriba el error en decimales: 0.00001

```

n	a	b	c	Error
0	0.60000	0.28571	0.12500	1.00000
1	0.68929	0.25357	0.01071	3.64099
2	0.69929	0.19184	-0.01562	0.67394
3	0.67986	0.17912	-0.02584	0.16498
4	0.67682	0.17752	-0.02257	0.05275
5	0.67552	0.17935	-0.02201	0.01253
6	0.67614	0.17978	-0.02146	0.00968
7	0.67620	0.17992	-0.02156	0.00189
8	0.67628	0.17987	-0.02156	0.00019
9	0.67626	0.17986	-0.02159	0.00044
10	0.67626	0.17985	-0.02158	0.00008
11	0.67626	0.17986	-0.02158	0.00003
12	0.67626	0.17986	-0.02158	0.00002
13	0.67626	0.17986	-0.02158	0.00000

Conclusiones

Guzmán Claustro, Edgar

Se presentó la solución tratando de asemejarse a lo visto en excel. Tratando de proveer un resultado corto, pero preciso. Sin duda se tuvo que realizar un análisis del algoritmo para entender qué variable necesitaba imprimir, practicando otra vez la comprensión y análisis del algoritmo, además de practicar una corrida de escritorio mental para predecir la salida y proceso de ejecución. Se convirtió en un proceso un poco más complicado de lo que esperaba.

Cordero Hernández, Marco R.

La realización de este ejercicio ha resultado un poco más complejas que las vistas anteriormente, incluso existiendo el reto de la comprensión de un código completamente desconocido con el objetivo de añadir mejoras. Este proceso es muy similar en contextos de desarrollo de alta dinamicidad en cuestión de los requerimientos presentes, en donde se proporciona una base de código previamente ajena al desarrollador y tiene que comprenderse a tal grado de poder ser capaces de explicarlo.

Más allá de la significancia con la que cuenta el método revisado, la realización de este documento para atender el requerimiento de la tarea asignada fue un desafío nuevo y curioso que nos ha brindado la oportunidad de explorar un campo adicional de los algoritmos.

Bibliografia

The MathWorks Inc. (2023). MATLAB version: 9.14.0 (R2023a), Natick, Massachusetts: The MathWorks Inc. <https://www.mathworks.com>.