ITESO

Universidad Jesuita de Guadalajara

Ingeniería en Seguridad Informática y Redes

Ingeniería en Sistemas Computacionales

Algoritmos de Solución Numérica

<u>Método de Euler</u>

IS727272 - Marco Ricardo Cordero Hernández

SI727576 - Edgar Guzmán Claustro

Jal., 07 de noviembre de 2023

Índice

Contexto

La línea del aprendizaje cursada hasta el momento ha hecho posible la revisión e implementación de múltiples métodos numéricos, los cuales tienen el propósito de hacer un proceso analítico de forma más sencilla con la herramienta ya por defecto, Matlab (The MathWorks Inc., 2023).

En esta ocasión, se realizará la revisión de un método para la resolución de ecuaciones diferenciales. Este tipo de ecuaciones pertenece al estudio analítico de las matemáticas, algo que puede resultar vano, sin embargo, el campo de estudio no sería tan amplio si no existiera una aplicación; por tanto, las ecuaciones diferenciales sirven para describir fenómenos reales a través de modelos matemáticos en campos tales como la biología, mecánica, medicina, e incluso política. Definidas de manera concisa, las ecuaciones diferenciales son aquellas en donde aparece una función desconocida y una o más de sus derivadas (Universidad de Jaén, s.f.).

El campo de análisis y estudio de las ecuaciones diferenciales usualmente tiene una parte introductoria en donde no se emplean más que álgebra para la resolución de sistemas de ecuaciones, sin embargo, en esta ocasión, se pretende hacer uso del software ya mencionado y un nuevo (en el sentido introductorio) método numérico: el método de Euler. Definido como un método *iterativo*, este procedimiento consiste en encontrar la solución de una ecuación diferencial de *primer orden* a través de un conjunto de valores iniciales conocidos dentro de otro rango de valores (Facultad de Ingeniería UNMdP, 2015). Quizás una parte que podría resaltar de lo anterior es aquello del órden; este término es debido a que las ecuaciones diferenciales no se limitan a un solo tipo, sino que existen múltiples categorías como ordinarias, parciales, de primer hasta n orden, lineales y no lineales (Universidad de Guanajuato, 2022).

La premisa del método es el conocimiento de los siguientes valores:

$$\frac{dy}{dx} = f(x, y); x_0 = a; x_1 = b$$

El cálculo de un paso en un intervalo:

$$h = \frac{x_f - x_0}{n} \text{ donde n es el número de iteraciones que se desean realizar}$$

Y un número finito de nuevas aproximaciones dadas por:

$$y_{n+1} = y_n + h * f(x_n, y_n);\ x_n = x_0 + n * h$$

De forma que se calcularán nuevas aproximaciones con cada nueva iteración hasta que se haya cumplido la cantidad establecida. Como se puede intuir, realizar esta serie de pasos de forma repetida puede llegar a ser tedioso, sin mencionar la muy probable introducción del error humano dentro de los resultados, por lo tanto, a continuación de se propone una implementación de este útil método a través de automatización programable.

Resultados

Código

Dentro de dicha implementación en Matlab, el método numérico de Euler fue implementado y automatizado con simples sentencias y en pocas líneas de código. Para que este script se ejecute con éxito, es necesaria la intervención del usuario final, quien será la entidad que definirá los datos iniciales para el método, es decir, la función de la ecuación diferencial con la que se realizará el método, el primer punto $x0$, el segundo punto $x1$, la condición inicial $y(x0)$ y el número de pasos a iterar.
Prosiguiendo con las fórmulas del método, estas se evalúan dentro del ciclo iterativo finito, que recorre de 0 a n, donde n es el número de iteraciones dadas por el usuario al iniciar el programa.

```
% Introducción y valores iniciales
fprintf('\n \tRESOLUCIÓN DE ECUACIONES DIFERENCIALES POR MEDIO MÉTODO DE EULER\n')
f = input('\nIngrese la ecuación diferencial de la forma: dy/dx=f(x,y) -> ','s');
x0 = input('\nIngrese el primer punto x0: ');
x1 = input('\nIngrese el segundo punto x1: ');
y0 = input('\nIngrese la condición inicial y(x0): ');
n = input('\nIngrese el número de pasos n: ');

% Calcular h e inicializar ambas variable
h = (x1 - x0)/n;
y = y0;
x = x0;

% Imprimir encabezado de tabla
fprintf('\nIteración\t\tx\t\ty\n');

% Iterar sobre número de pasos definidos
for i = 0:n
   y = y + h*eval(f);
   x = x + h;
   fprintf("\t%d\t\t%f\t%f\n",i,x,y);
end

fprintf('\n El punto aproximado y(x1) es = %10.6f\n', y);
```

Salida

El programa en primera instancia le solicitará al usuario los parámetro de entrada:

```
        RESOLUCIÓN DE ECUACIONES DIFERENCIALES POR MEDIO MÉTODO DE EULER

Ingrese la ecuación diferencial de la forma: dy/dx=f(x,y) -> sin(x)-log(y)

Ingrese el primer punto x0: 0.13

Ingrese el segundo punto x1: 0.1325

Ingrese la condición inicial y(x0): 0.32

Ingrese el número de pasos n: 4
```

Después, se desplegará una tabla con los resultados de cada iteración del método y posteriormente el resultado final:

Iteración	x	y
0	0.130625	0.320793
1	0.131250	0.321585
2	0.131875	0.322376
3	0.132500	0.323166
4	0.133125	0.323954

El punto aproximado y(x1) es = 0.323954

Conclusiones

Guzmán Claustro, Edgar

Dentro de todo este ámbito de programación, me es sumamente claro que siempre necesitamos realizar por lo menos una corrida de escritorio para tener una idea clara de cómo será el comportamiento del programa. En este escenario donde implementamos el método numérico de Euler, se realizó una simple implementación del método usando excel, de esta forma la manera de implementarla en código será más sencillo.

Al momento de codificar dentro de matlab, me di cuenta que es muy sencillo ver los resultados para el usuario, ahorrando mucho tiempo de cálculos.

Cordero Hernández, Marco R.

Al desarrollar la solución para atender al método actual se pudo percatar cómo este mismo cuenta con un nivel de simplicidad considerable, lo cual se agradece bastante porque hasta el momento se ha tenido la oportunidad de ver implementaciones complejas y ocasionalmente ofuscadas. Revisando la prueba del método de Euler, no es necesario contar con un gran trasfondo matemático o un nivel de razonamiento lógico para comprenderla, por ende, su implementación es aún más sencillo, algo que pudiera parecer intuitivo pero no siempre es de este modo.

Con menos de 20 líneas de código se ha logrado el objetivo puntual propuesto de forma exacta y sucinta, lo cual no requirió un desgaste mental significativo, algo que en demasiadas ocasiones se pasa por alto, terminando en desempeños pobres, y por ende, en resultados deficientes.

Bibliografía

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

Universidad de Jaén. (s.f.). *ECUACIONES DIFERENCIALES*. Recuperado de http://matema.ujaen.es/jnavas/web_modelos_empresa/archivos/archivos%20pdf/teoria/teoria%20continuo/teoria%20continuo%20tema1.pdf.

Facultad de Ingeniería UNMdP. (2015). *Método de Euler*. Recuperado de http://www3.fi.mdp.edu.ar/metodos/apuntes/euler%20-%20Rodrigo.pdf.

Universidad de Guanajuato. (2022). *Clase digital 1. Ecuaciones diferenciales en general*. Recuperado de https://blogs.ugto.mx/rea/clase-digital-1-ecuaciones-diferenciales-en-general/.