Asignatura	Datos del alumno		Fecha
Métodos numéricos avanzados en ingeniería.	**Apellidos:** Balsells Orellana		09/03/2021
	Nombre: Jorge A.		

Laboratorio: Resolución de PCU con el método de Jacobi.

El problema consiste en encontrar la solución aproximada del problema de contorno lineal dado por la ecuación 1

$$y(x)'' + \frac{1}{x}y(x)' + y(x) = \ln(x), x \in [1,2], y(1) = 0, y(2)' = \frac{1}{2} \tag{1}$$

cuya solución exacta es la ecuación 2

$$y(x) = \ln(x) \tag{2}$$

Para la solución aproximada, se debe considerar que tomando $h = \frac{b-a}{10}$ se establece un mallado en el intervalo $[a, b]$ de 11 puntos, siendo primero a y el último b.

1. Transformación del problema

Utilizando las aproximaciones derivadas 7, transforma el problema de frontera en un sistema lineal, que denotaremos $Ax = b$, de 10 ecuaciones con 10 incógnitas.

$$y(x)' \approx \frac{y(x+h) - y(x-h)}{2h}, y(x)'' \approx \frac{y(x+h) - 2y(x) + y(x-h)}{h^2} \tag{3}$$

1.1. Transformación del problema por medio de diferencias finitas

$$\begin{gathered}
\tfrac{y(x+h)-2y(x)+y(x-h)}{h^2} + \tfrac{1}{x}[\tfrac{y(x+h-y(x-h))}{2h}] + y(x) \approx \ln(x) \\
y(x+h) - 2y(x) + y(x-h) + \tfrac{h}{2xh}[y(x+h) - y(x-h)] + h^2 y(x) \approx h^2 \ln(x) \\
(1 + \tfrac{h}{2x})y(x+h) + (h^2 - 2)y(x) + (1 - \tfrac{h}{2x})y(x-h) \approx h^2 \ln(x) \\
\mathbf{(1 + \tfrac{h}{2x_i})y_{i+1} + (h^2 - 2)y_i + (1 - \tfrac{h}{2x_i})y_{i-1} \approx h^2 \ln(x_i)}
\end{gathered} \tag{4}$$

Las ecuaciones 4 hacen referencia a la simplificación llevada a cabo para obtener la ecuación en la que se validarán los valores de las condiciones iniciales para $y(1)$, para $y(2)'$ y para valores desde i hasta N.

$$
\begin{gathered}
(1+\tfrac{h}{2x_i})y_{i+1}+(h^2-2)y_i+(1-\tfrac{h}{2x_i})\alpha \approx h^2\ln(x_i) \\
(1+\tfrac{h}{2x_i})y_{i+1}+(h^2-2)y_i \approx h^2\ln(x_i)-(1-\tfrac{h}{2x_i})\alpha \\
\mathbf{(1+\tfrac{h}{2x_0})y_1+(h^2-2)y_0 \approx h^2\ln(x_0)}
\end{gathered} \tag{5}
$$

En la ecuación 5, evaluamos la condición inicial $y(1)=0=\alpha$.

$$
(h^2-2)y_i+(1-\tfrac{h}{2x_i})y_{i-1} \approx h^2\ln(x_i)-(1+\tfrac{h}{2x_i})y_{i+1}\beta \tag{6}
$$

En la ecuación 6, evaluamos la condición inicial $y(2)'=\frac{1}{2}=\beta$. Dado que la condición inicial no es Dirichlet, se debe evaluar en la primera derivada de la misma manera en que se ha trabajado el proceso anterior, tal y como se muestra en la ecuación 7.

$$
\frac{1}{2}\approx\frac{y_{i+1}-y_{i-1}}{2h}\longrightarrow\frac{1}{2}\approx\frac{\beta-y_{i-1}}{2h}\longrightarrow\beta\approx h+y_{i-1} \tag{7}
$$

Si despejamos β como se hace en la ecuación 7 y la evaluamos en 6, nos quedará y_{i+1} en términos de y_{i-1}, lo cual es útil y necesario a la vez, ya que en la última posición ya no tenemos el valor $i+1$. El resultado de la evaluación se obtiene en la ecuación 8.

$$
\begin{gathered}
(h^2-2)y_i+(1-\tfrac{h}{2x_i})y_{i-1} \approx h^2\ln(x_i)-(h+y_{i-1}+\tfrac{h^2}{2x_i}+\tfrac{h}{2x_i}y_{i-1}) \\
(h^2-2)y_i+(1-\tfrac{h}{2x_i})y_{i-1} \approx h^2\ln(x_i)-h-\tfrac{h^2}{2x_i}-(1+\tfrac{h}{2x_i})y_{i-1} \\
\mathbf{(h^2-2)y_n+2y_{n-1} \approx h(h\ln(x_n)-\tfrac{h}{2x_n}-1)}
\end{gathered} \tag{8}
$$

Teniendo estas ecuaciones listas se puede crear una matriz de coeficientes generada a través de la cantidad de ecuaciones e incógnitas que necesitemos. En este problema se necesita obtener una matriz de 10 ecuaciones con 10 incognitas, motivo por el cual, al determinar el intervalo $h=$

$(b-a)/N$ tiene en el denominador solamente la N, y no $N+1$, ya que siempre existe un punto más a la cantidad de subintervalos generados, y la cantidad de subintervalos en realidad serán los intervalos dónde destacarán las ecuaciones.

1.2. Matriz de ecuaciones

Una matriz de coeficientes de un sistema de ecuaciones lineales también se conoce como matriz aumentada. Es una matriz que contiene en cada una de las columnas, los coeficientes correspondientes a una variable de un sistema de ecuaciones. La matriz Aumentada generada por las ecuaciones obtenidas, se visualiza a continuación en la ecuación 9.

$$\begin{bmatrix} (h^2-2) & (1+\frac{h}{2x_0}) & 0 & 0 & \cdots & 0 & 0 & 0 & 0 \\ (1-\frac{h}{2x_1}) & (h^2-2) & (1+\frac{h}{2x_1}) & 0 & \cdots & 0 & 0 & 0 & 0 \\ 0 & (1-\frac{h}{2x_2}) & (h^2-2) & (1+\frac{h}{2x_2}) & \cdots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & (1-\frac{h}{2x_{n-2}}) & (h^2-2) & (1+\frac{h}{2x_{n-2}}) & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & (1-\frac{h}{2x_{n-1}}) & (h^2-2) & (1+\frac{h}{2x_{n-1}}) \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & 2 & (h^2-2) \end{bmatrix} \tag{9}$$

Un vector de términos independientes contiene los resultados de los valores constantes de un sistema de ecuaciones, este debe de tener en filas, la misma longitud de la matriz de coeficientes. El vector de términos independientes generado por las ecuaciones obtenidas, se visualiza a continuación en la ecuación 10

$$\begin{bmatrix} h^2\ln(x_0) \\ h^2\ln(x1) \\ h^2\ln(x_2) \\ h^2\ln(x_3) \\ \vdots \\ h^2\ln(x_{n-3}) \\ h^2\ln(x_{n-2}) \\ h^2\ln(x_{n-1}) \\ h^2 ln(x_n) + (\frac{h}{2x_n} - 1)2h\beta \end{bmatrix} \tag{10}$$

2. Método de Jacobi

El método de Jacobi es un método iterativo utilizado para resolver problemas de sistemas ecuaciones lineales de la forma $Ax = b$. Apuntes vistos en clase [1]. Programa del método de Jacobi para la resolución del problema anterior. Para que este método funcione, debe converger. Para que un método converja, debe ser consistente y estable, si la matriz A es estrictamente diagonal dominante. En algunos casos, para verificar la convergencia del método se calcula el radio espectral $\rho(D^{-1}R) < 1$ En este caso podemos guiarnos a través de la cantidad de iteraciones que el método ha tenido, y a través del comportamiento del error. Método implementado a traves de apuntes en clase [1].
Para generar la matriz A necesaria en el método de Jacobi, utilizamos las 3 diagonales generadas por diferencias finitas, y creamos una matriz sumando las 3 diagonales en la posición que le corresponde a cada una, considerando $A = diagonal(D_p, P_0) + diagonal(D_i, P_i) + diagonal(D_s, P_s)$. Al sumarlas no se genera conflicto ya que A inicialmente es una matriz de ceros siendo Dp la diagonal principal, Di la diagonal inferior, Ds la diagonal superior, Pi la posicion de la diagonal inferior igual a menos uno, Ps la posicion de la diagonal superior igual a uno, y P0 la posicion cero.

```
%[x,y,iter,v] = df('p','r',1,2,10,0,1/2,1e-7,5000);
function [x,y,iter,v] = df(p,r,a,b,N,alfa,beta,tol,maxiter)

h = (b-a)/(N); %generacion de intervalos
x = a:h:b;
X = [a+h:h:b];

px = feval(p,X); %validaciones de funciones
qx = -1.*ones(1,length(X)); %Q es constante, no requiere funcion.
rx = feval(r,X);

dp = 2+(h^2).*qx; %generacion de diagonales
ds = -1+(h/2).*px(1:end-1);
di = -1-(h/2).*px(2:end);
di(end) = -2;

d = -(h^2).*rx; %generacion de vector de terminos independientes
d(1) = d(1) + (1+(h./2).*px(1)).*alfa;
d(end) = d(end) + (1-(h/2).*px(end)).*2.*h.*beta; %Beta es el valor de la ...
    derivada, por lo tanto se recalcula.
d = d(:);

mat = zeros(length(x)); %Generacion de matriz con diagonales
mat = diag(dp) + diag(ds,1) + diag(di,-1);
init = zeros(length(X),1);

[y,iter,v] = jacobi(mat,d,init,tol,maxiter); %Jacobi
y = [alfa ; y]; %Agregando terminos conocidos
x = x(:);
end

function [x,iter,v] = jacobi(A,b,x0,tol,maxiter)
    L = tril(A,-1);
```

```
    U = triu(A,1);
    D = diag(A);
    iD = diag(1./D);

    iter = 1;
    incre = tol+1;
    v = [];

    while iter<maxiter && incre>tol
        x = -iD*(L+U)*x0+iD*b;
        incre = norm(x-x0,inf); %incre = norm(b-a*x);
        v = [v,incre];
        iter = iter+1;
        x0 = x;
    end

    if incre>tol
        disp('se necesitan mas iteraciones')
    end
end

function y = p(x)
    y = -1./x;
end

%Funcion Q es constante, no se requiere funcion.
%Solamente se requiere inicializar un vector con el valoir de la cte.

function y = r(x)
    y = log(x);
end
```

Listing 1: Script en Matlab para el método de Jacobi con Diferencias Finitas.

3. Implementación de parámetros

Implementación de la matriz A y del vector b. Utiliza como criterio de parada $\|y^{(k+1)} - y^{k}\| < 10^{-7}$ y como estimación inicial $y^{(0)} = [00 \ldots 0]$. Indica el numero de iteraciones necesario y el valor del residuo en la última iteración.

Se alcanzó la tolerancia buscada después de **2631 iteraciones**, dando como resultado desde un error de centécimas a un error muy bajo mientras la función se volvía convergente. La implementación se visualiza en el código descrito anteriormente, donde se hace un conjunto de lo visto en Diferencias Finitas y método de Jacobi. Luego de la implementación, el resultado de la solución exacta Vs el resultado de la solución aproximada, se muestra en la figura 1. A simple vista es muy poco notoria la diferencia por la escala, ya que desde el punto inicial, la función se visualiza muy cercana a la solución exacta.

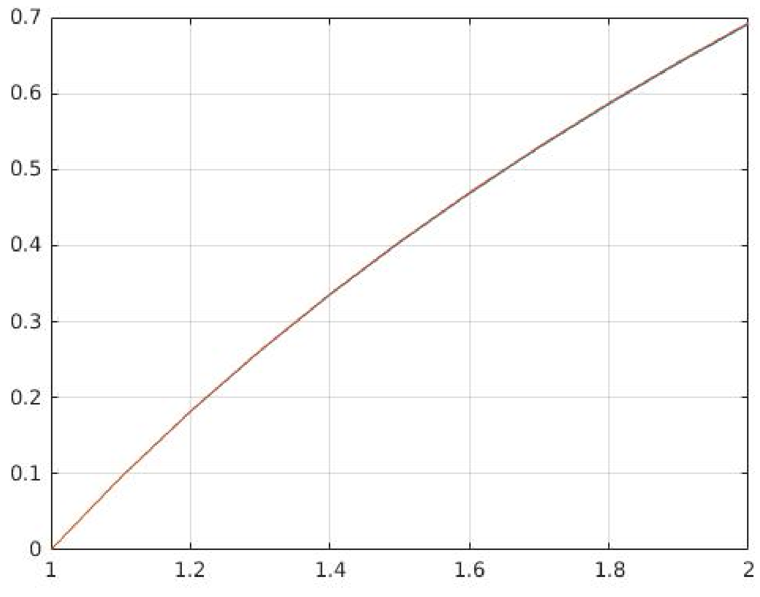

Figura 1: Representación gráfica de la función aproximada vs función exacta.

4. Error

Proporciona una tabla con el error exacto cometido en cada uno de los puntos $x = a : h : b$.

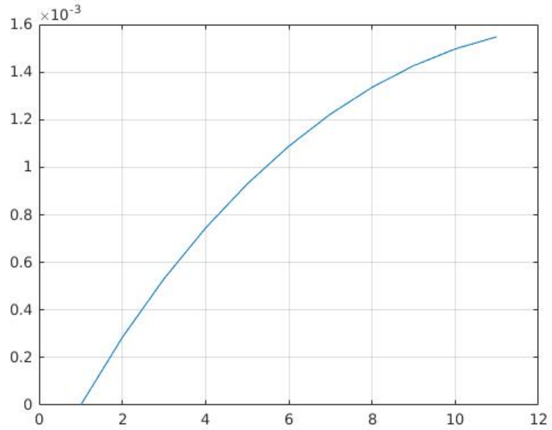

Figura 2: Representación gráfica del error en el problema resuelto.

x	y	f(x)	error
1	0	0	0
1.1	0.095025	0.095310	0.000284
1.2	0.181790	0.182321	0.000530
1.3	0.261619	0.262364	0.000744
1.4	0.335542	0.336472	0.000929
1.5	0.404376	0.405465	0.001088
1.6	0.468780	0.470003	0.001223
1.7	0.529292	0.530628	0.001335
1.8	0.586359	0.587786	0.001426
1.9	0.640356	0.641853	0.001497
2	0.691598	0.693147	0.001548

5. Anexo

El problema fue resuelto a partir de las derivadas, sin embargo se utilizó el problema desarrollado en clase. Para poder utilizar el problema dado en clase, se necesita llevar la función a la forma $y(x)'' = P(x)y(x)' + Q(x)y(x) + R(x)$, respetando signos. en este caso, $P(x)$ obtuvo valor de $\frac{-1}{x}$, $Q(x)$ valor de -1 quedando constante como un vector con el mismo valor en todos sus puntos, y $R(x)$ con valor de $\ln(x)$.

En la figura 3, se muestra el valor de la tolerancia dada por $\left|\left|y^{(k+1)} - y^{(k)}\right|\right| < 1e^{-7}$. Se demuestra que el método converge según disminuye este valor.

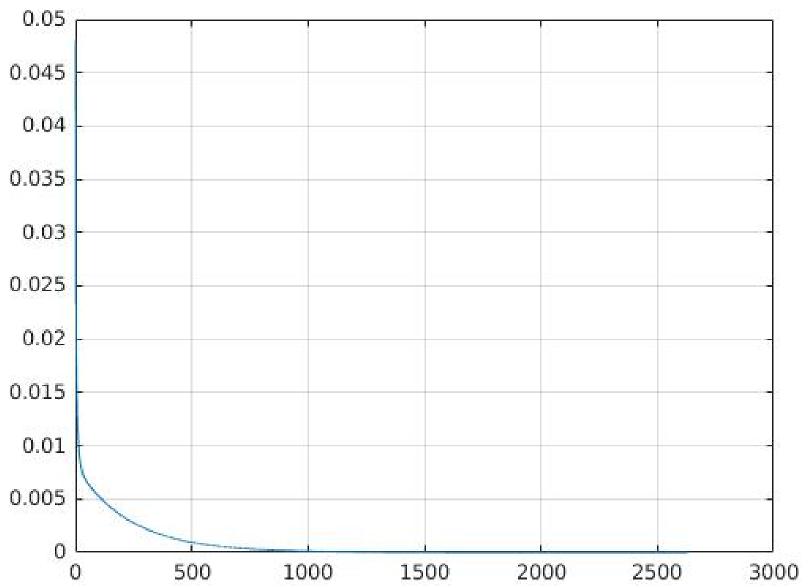

Figura 3: Representación gráfica de la convergencia.

Referencias

[1] Juan R. Torregrosa Alicia Cordero, Neus Garrido. Apuntes de clase de métodos numéricos avanzados en ingeniería., 2020.