PRÁCTICA CALIFICADA 5

1) USO DEL MÉTODO RKF

```
clearvars
clear all
close all
clc
w = @(x,u) [u(2);4*exp(x)-2*u(1)];
w0 = [2;0];
x0 = 0; xf = 2; h = 0.01;
net = 6;
m = 6;
A = [0 0 0 0 0 0;
    1/5 0 0 0 0 0;
    3/40 9/40 0 0 0 0;
    3/10 -9/10 6/5 0 0 0;
    -11/54 5/2 -70/27 35/27 0 0;
    1631/55296 175/512 575/13824 -44275/110592 253/4096 0]
```

```
A = 6×6
         0         0         0         0         0         0
    0.2000         0         0         0         0         0
    0.0750    0.2250         0         0         0         0
    0.3000   -0.9000    1.2000         0         0         0
   -0.2037    2.5000   -2.5926    1.2963         0         0
    0.0295    0.3418    0.0416   -0.4003    0.0618         0
```

```
bi = [37/378 0 250/621 125/594 0 512/1771]
```

```
bi = 1×6
    0.0979         0    0.4026    0.2104         0    0.2891
```

```
bj = [2825/27648 0 18575/48384 13525/55296 277/14336 1/4]
```

```
bj = 1×6
    0.1022         0    0.3839    0.2446    0.0193    0.2500
```

```
c = [0;1/5;3/10;3/5;1;7/8]
```

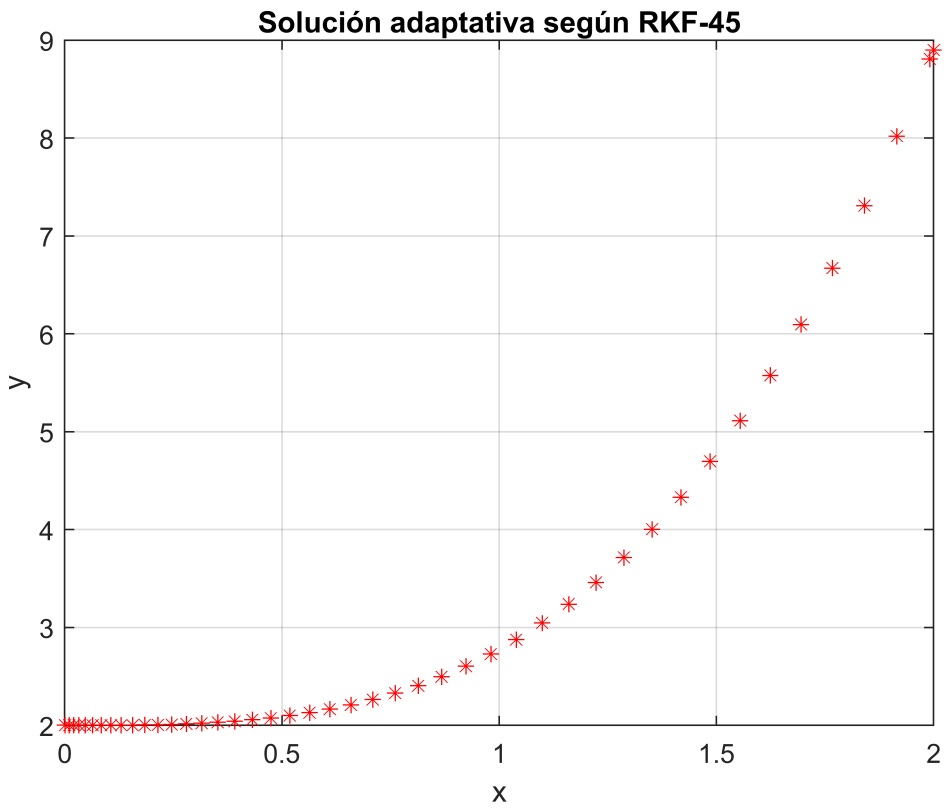
```
c = 6×1
         0
    0.2000
    0.3000
    0.6000
    1.0000
    0.8750
```

```
n = 4;
[xRKF,yRKF] = RKFGeneral(A,bi,bj,c,x0,xf,h,w,w0,n,net);
plot(xRKF,yRKF,'r*')
```

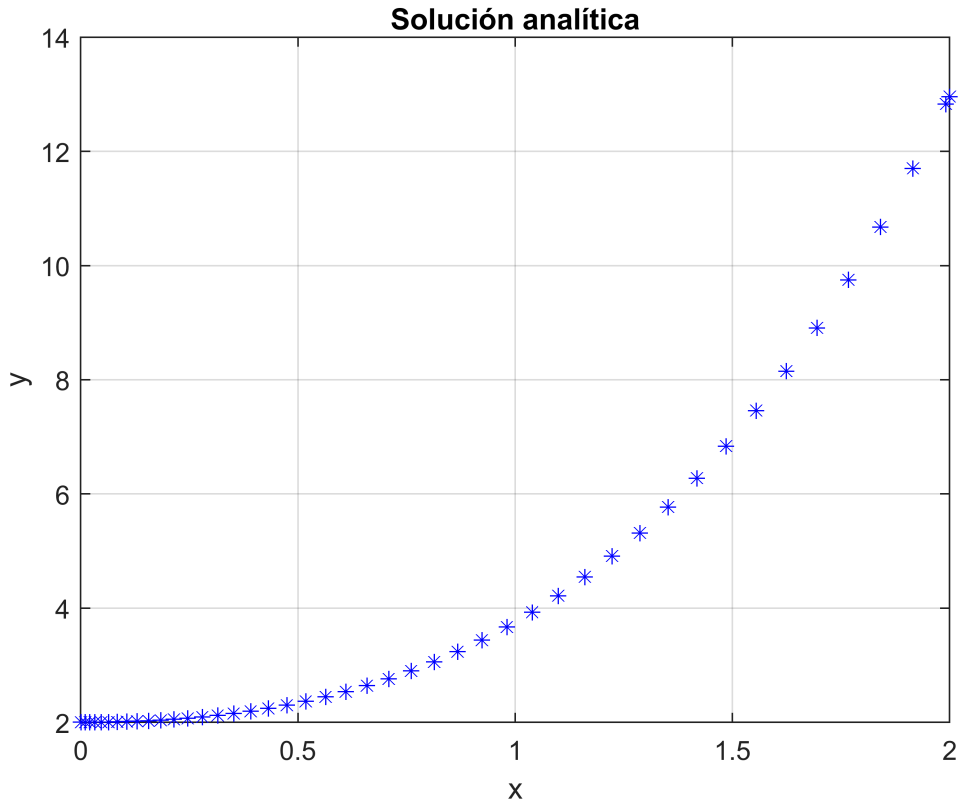
```
Warning: MATLAB has disabled some advanced graphics rendering features by switching to software OpenGL. For
 more information, click here.
```

```
xlabel('x')
ylabel('y')
grid on
```

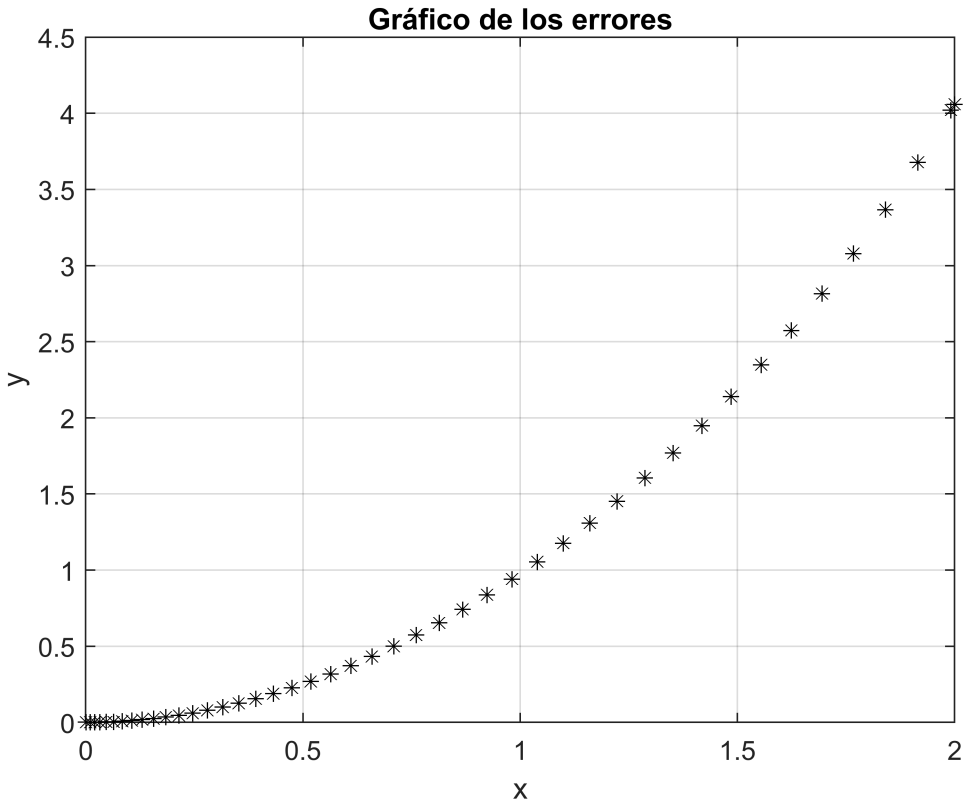
```
title('Solución adaptativa según RKF-45')
```




```
yanal = @(x) 2*(exp(x)-sin(x));
plot(xRKF,yanal(xRKF),'b*')
xlabel('x')
ylabel('y')
grid on
title('Solución analítica')
```




```
plot(xRKF,yanal(xRKF)-yRKF,'k*')
xlabel('x')
ylabel('y')
grid on
title('Gráfico de los errores')
```



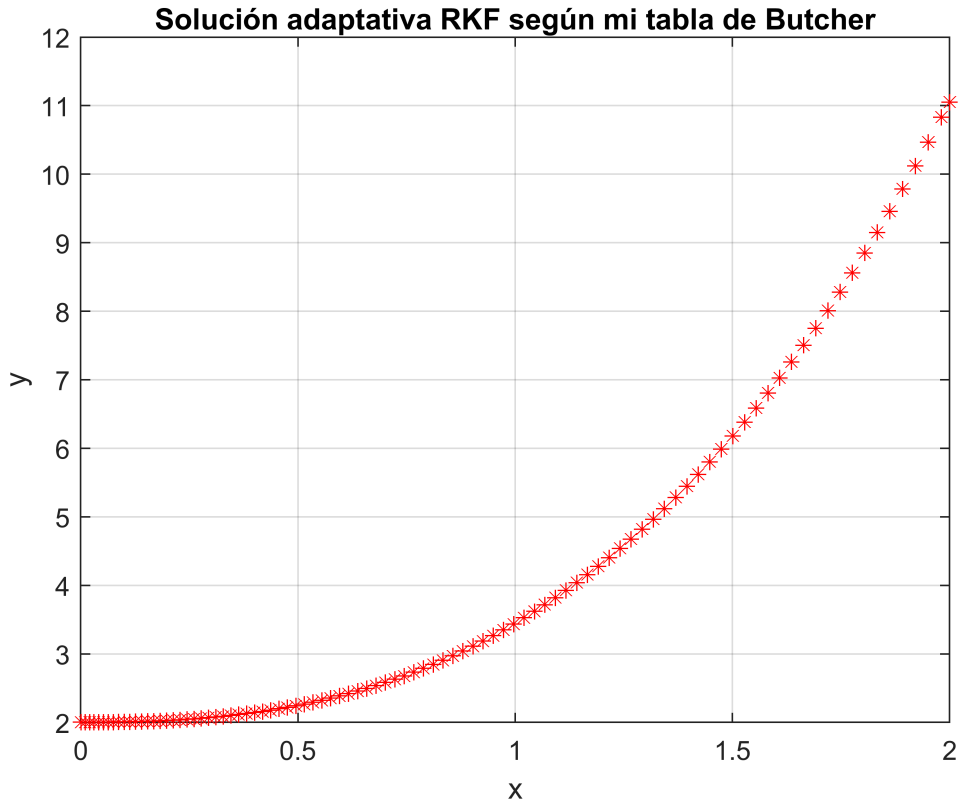
2) MÉTODO ADAPTATIVO DESARROLLADO POR MI

```
clearvars
clear all
close all
clc
w = @(x,u) [u(2);4*exp(x)-2*u(1)];
w0 = [2;0];
x0 = 0; xf = 2; h = 0.01;
net = 6;
m = 6;
[A,bi,bj,c] = TablaButcher(net,m)
```

```
A = 6×6
     0     0     0     0     0     0
     6     0     0     0     0     0
     3     5     0     0     0     0
     1     3     6     0     0     0
     5     6     4     1     0     0
     6     6     5     5     5     0
bi = 1×6
    0.2174    0.2609    0.0435    0.2609    0.1739    0.0435
bj = 1×6
    0.0800    0.1600    0.2400    0.2400    0.0400    0.2400
c = 6×1
     0
     6
     8
    10
    16
```

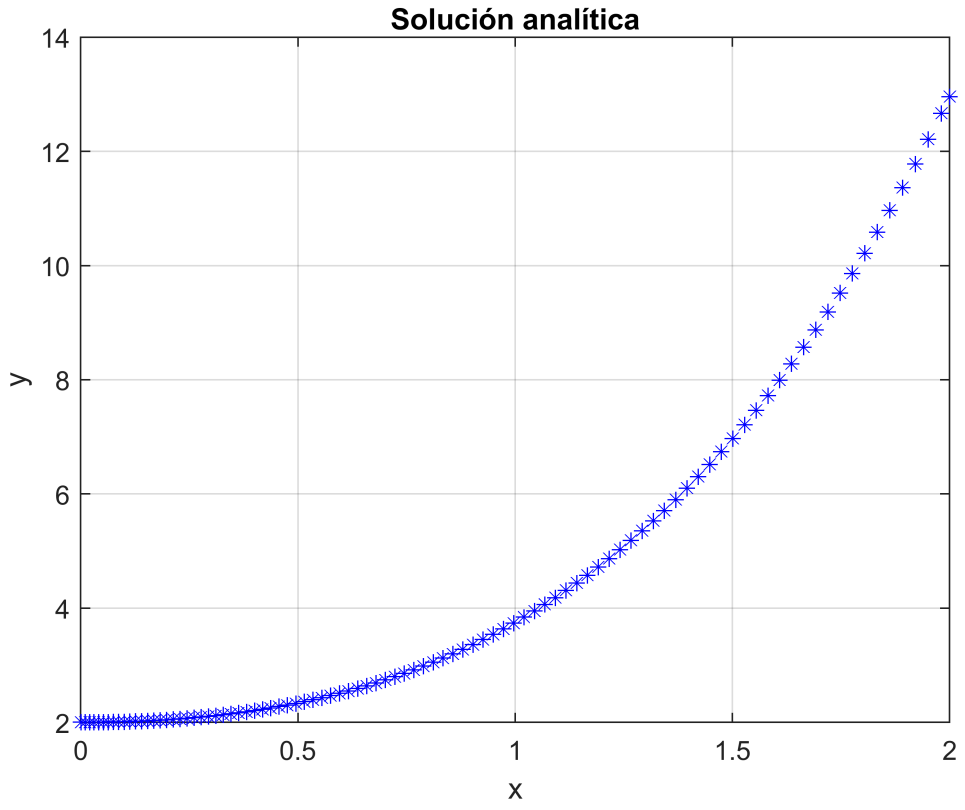
27

```
n = 4;
[xButcher,yButcher] = RKFGeneral(A,bi,bj,c,x0,xf,h,w,w0,n,net);
plot(xButcher,yButcher,'r*')
xlabel('x')
ylabel('y')
grid on
title('Solución adaptativa RKF según mi tabla de Butcher')
```

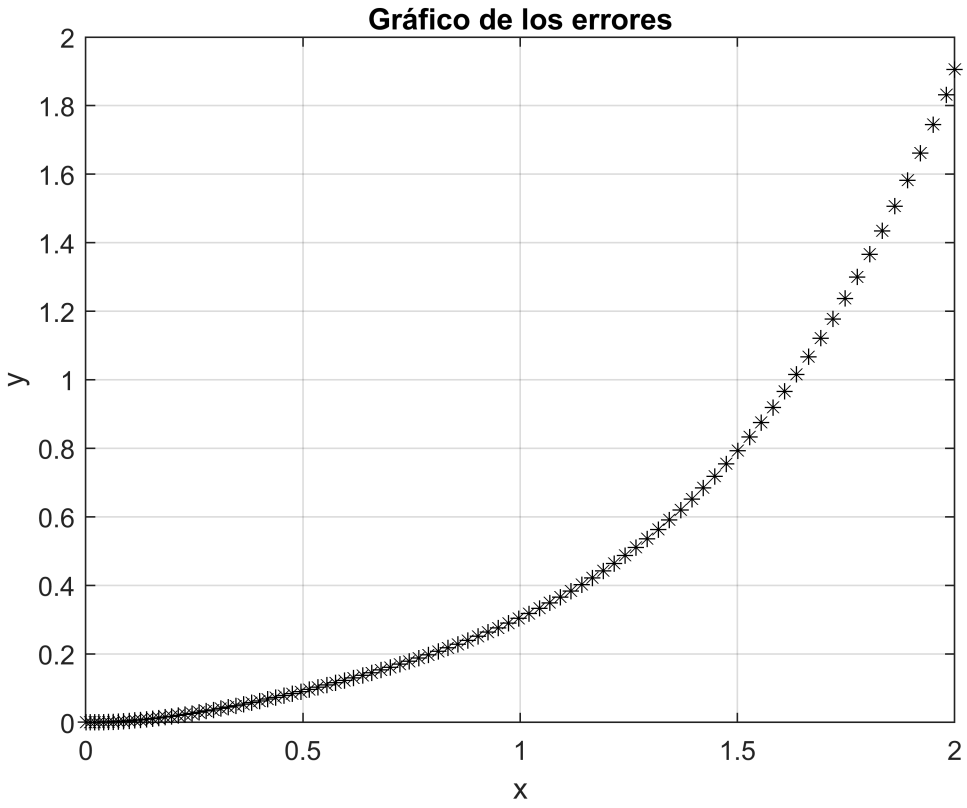



```
yanal = @(x) 2*(exp(x)-sin(x));
```

```
plot(xButcher,yanal(xButcher),'b*')
xlabel('x')
ylabel('y')
grid on
title('Solución analítica')
```




```
plot(xButcher,yanal(xButcher)-yButcher,'k*')
xlabel('x')
ylabel('y')
grid on
title('Gráfico de los errores')
```



4) USO DEL MÉTODO DE DISPARO LINEAL

```
clearvars
clear all
close all
clc
```

Datos del problema:

```
L = 50; T = 300; w = 50; E = 1.2*10^7; I = 4;
a = 0; b = L;
alpha = 0;
beta = 0;
```

Dominio de integración:

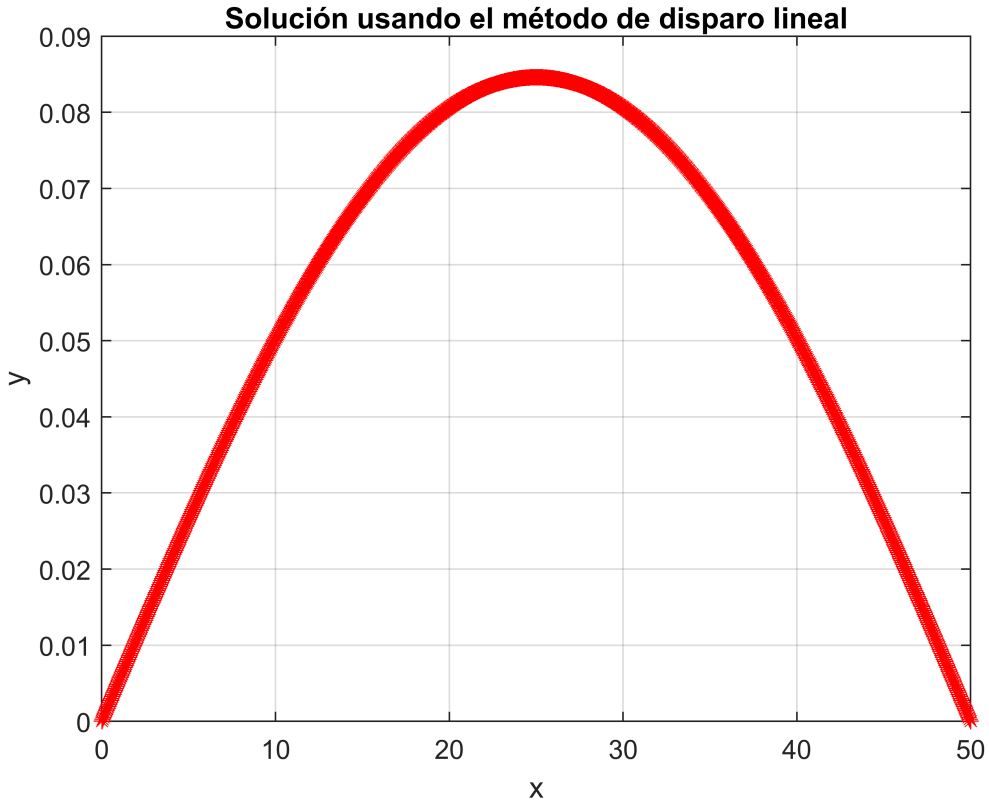
```
x0 = a;
xf = b;
h = 0.05;
```

SISTEMA 1:

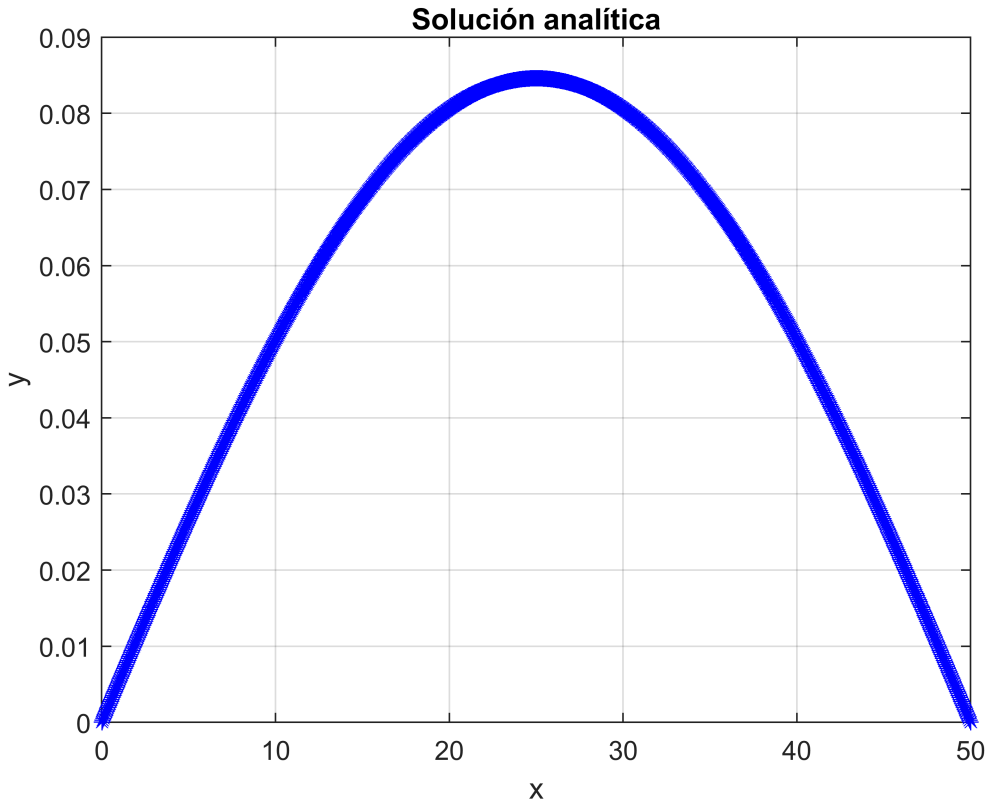
```
w1 = @(x,v) [v(2);(T/(E*I))*v(1)+(w*x*(x-L)/(2*E*I))];
v0w1 = [alpha; 0];
[x1SL,w1SL] = RK4(w1,x0,xf,v0w1,h);
```

SISTEMA 2:

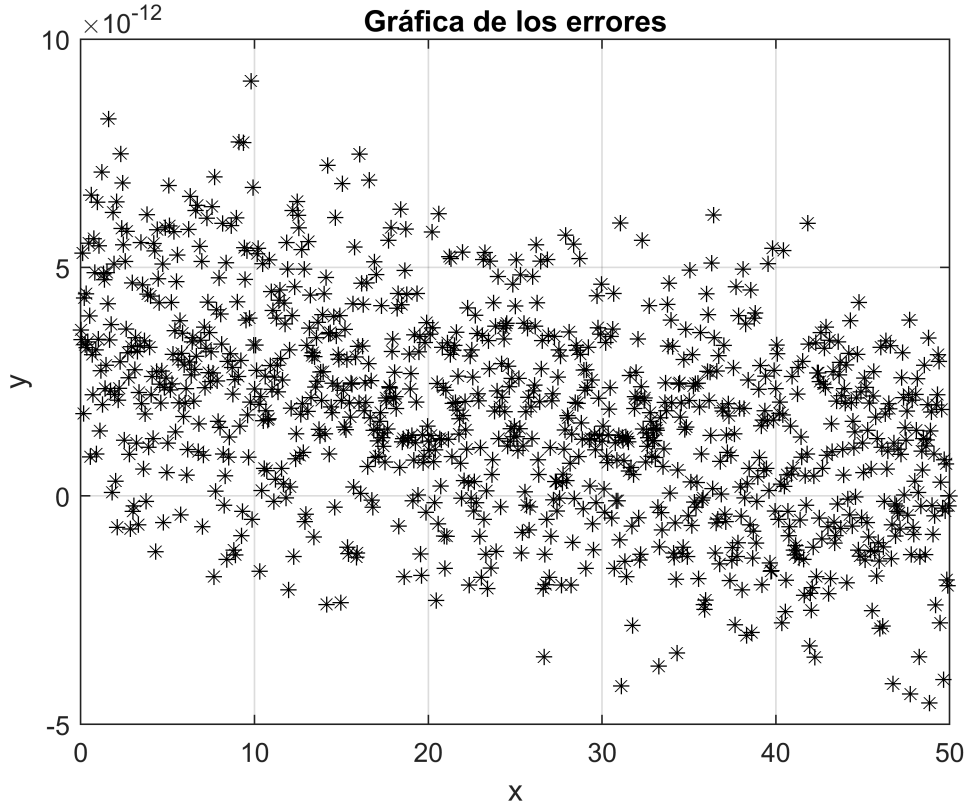
```
w2 = @(x,v) [v(2);(T/(E*I))*v(1)];
v0w2 = [0; 1];
[x2SL,w2SL] = RK4(w2,x0,xf,v0w2,h);
y = w1SL(1,:)+((beta-w1SL(1,end))./(w2SL(1,end))).*w2SL(1,:);
x = x1SL;
plot(x,y,'*r')
xlabel('x')
ylabel('y')
grid on
title('Solución usando el método de disparo lineal')
```




```
q = T/(E*I);
r = w/(2*E*I);
s = L;
ypar = @(x) -r*x.^2/q+r*s*x/q-2*r/q^2;
M = [1 1;exp(sqrt(q)*L) exp(-sqrt(q)*L)];
C = [2*r/q^2;-ypar(L)];
A = M\C;
yanal = @(x) A(1)*exp(sqrt(q)*x)+A(2)*exp(-sqrt(q)*x)+ypar(x);
plot(x,yanal(x),'b*')
xlabel('x')
ylabel('y')
grid on
title('Solución analítica')
```




```
plot(x,yanal(x)-y,'k*')
xlabel('x')
ylabel('y')
grid on
title('Gráfica de los errores')
```




```
function [A,bi,bj,c] = TablaButcher(n,m)
    A = zeros(n);
    c = zeros(n,1);
    irand = randi([1,m],1,n);
    jrand = randi([1,m],1,n);
    bi = irand./sum(irand);
    bj = jrand./sum(jrand);
    for i = 2:n
        A(i,1:i-1) = randi([1,m],1,i-1);
    end
    for i = 1:n
        c(i) = sum(A(i,:));
    end
end
function [x,y] = RKFGeneral(A,bi,bj,c,x0,xf,h,w,w0,n,net)
    x(1) = x0; wi(:,1) = w0; wj(:,1) = w0;
    stop = 0; i = 1;
    while(1)
        x(i+1) = x(i) + h;
        if x(i+1)-xf > 0
            x(i+1) = xf;
            h = xf - x(i);
            stop = 1;
        end
        K = KGeneral(A,c,w,wi(:,i),x(i),h,net);
        dwi(:,i) = h*(sum((K.*bi)'))'; wi(:,i+1) = wi(:,i)+ dwi(:,i);
        dwj(:,i) = h*(sum((K.*bj)'))'; wj(:,i+1) = wj(:,i)+ dwj(:,i);
```

```
        if i>1 & stop == 0
            epre = dwj(1,i-1)-dwi(1,i-1);
            enew = dwj(1,i)-dwi(1,i);
            if epre <= enew
                a = 1/(n+1);
            else
                a = 1/n;
            end
            h = h*abs(enew/epre)^a;
        elseif stop == 1
            break;
        end
        i = i + 1;
    end
    y = wi(1,:);
end
function K = KGeneral(A,c,w,wl,xj,h,net)
    K = zeros(length(wl),net);
    for i = 1:net
        if i == 1
            K(:,i) = w(xj,wl);
        else
            w1l = wl(1,1) + h*sum(K(1,:).*A(i,:));
            w2l = wl(2,1) + h*sum(K(2,:).*A(i,:));
            w12l = [w1l;w2l];
            K(:,i) = w(xj+c(i)*h,w12l);
        end
    end
end
function [x,wj] = RK4(w,x0,xf,w0,h)
    x = [x0:h:xf];
    n = (xf-x0)/h;
    wj(:,1) = w0;
    for i = 1:n
        k1 = w(x(i),wj(:,i));
        k2 = w(x(i)+(h/2),wj(:,i)+(h/2)*k1);
        k3 = w(x(i)+(h/2),wj(:,i)+(h/2)*k2);
        k4 = w(x(i)+h,wj(:,i)+h*k3);
        wj(:,i+1) = wj(:,i)+(h/6)*(k1+2*k2+2*k3+k4);
    end
end
```