Apellidos, Nombre: García Martínez, Máximo
Apellidos, Nombre:

TAREA2 (Ampliación Métodos Adaptativos)

Adjuntad vuestro código para adapta23.m

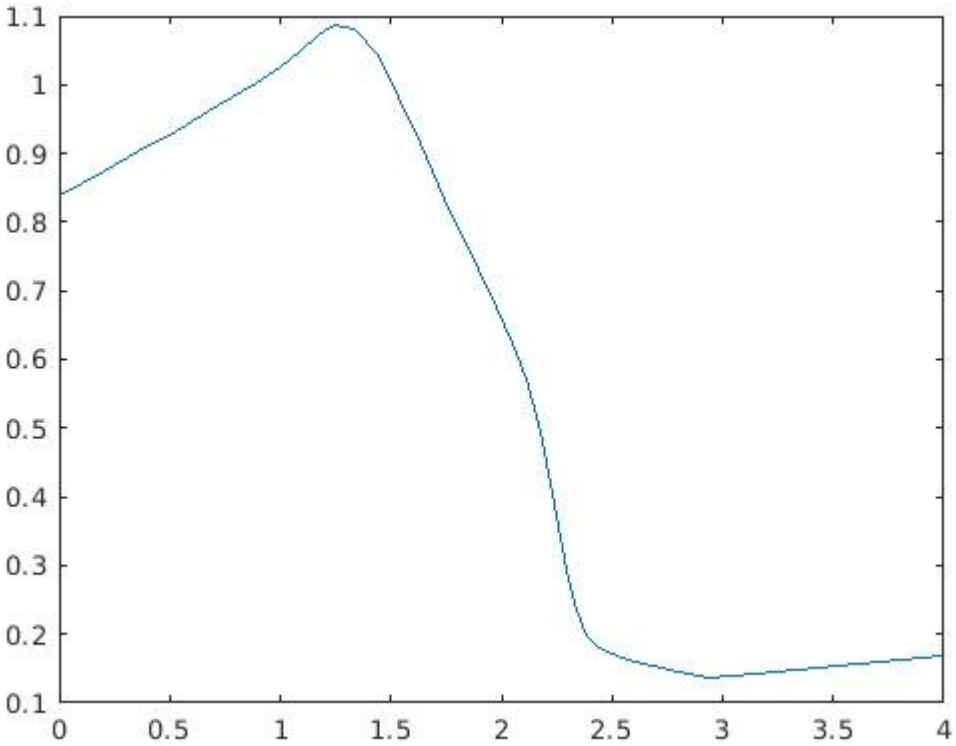
```
function [T, S]=adapta23(fun,Tspan,y0,tol)
   %Metodo adaptativo basado en Heun + Euler
   % fun   -> funcion con la ec. diferencial
   % Tspan -> intervalo [a,b] donde resolver
   % y0    -> vector columna de condiciones iniciales
   % tol   -> cota del error relativo (p.e 1e-3 o 1e-5)

   % Ptos inicial y final del intervalo a resolver
   T0 = Tspan(1);
   Tfin = Tspan(2);

   % Inicializar el primer valor para h
   h = (Tfin - T0) / 20;

   % Inicializar las salidas T y S con sus valores iniciales
   T = T0;
   S = y0;

   while(T(end)<Tfin)
      % Parto del ultimo punto conocido de la solucïï¿½n
      t=T(end); y=S(:,end);

      % Calcular K1, K2, K3
      K1 = fun(t, y);

      delta = h/2;
      K2 = fun(t + delta, y + delta * K1);

      delta = 3*h/4;
      K3 = fun(t+delta, y+delta*K2);

      % Estimacion orden 3
      s3 = y + (h/9) * (2*K1 + 3*K2 + 4*K3);
      K4 = fun(t+h, s3);

      % Estimacion orden 2
      s2 = y +(h/24)*(7*K1 + 6*K2 + 8*K3 + 3*K4);

      % Calcular estimacion del error E
      E = norm(abs(s3 - s2));

       % Calcular ratio r entre E y tol
       ratio = E / tol;

       if ratio<=1   % Paso aceptado, aï¿½adir los nuevos valores a T y S
          T = [T t+h]; S=[S s3];
       end
       % Actualizar h usando la formula optima con un factor adicional 0.8
       % Comprobad que no nos salimos del intervalo en el siguiente salto
       % Recortad h si se da el caso.
       h = (h / ratio^(1/3)) * 0.8;
```

```
        if T(end)+h > Tfin
            h = Tfin - T(end);
        end
    end
return
```

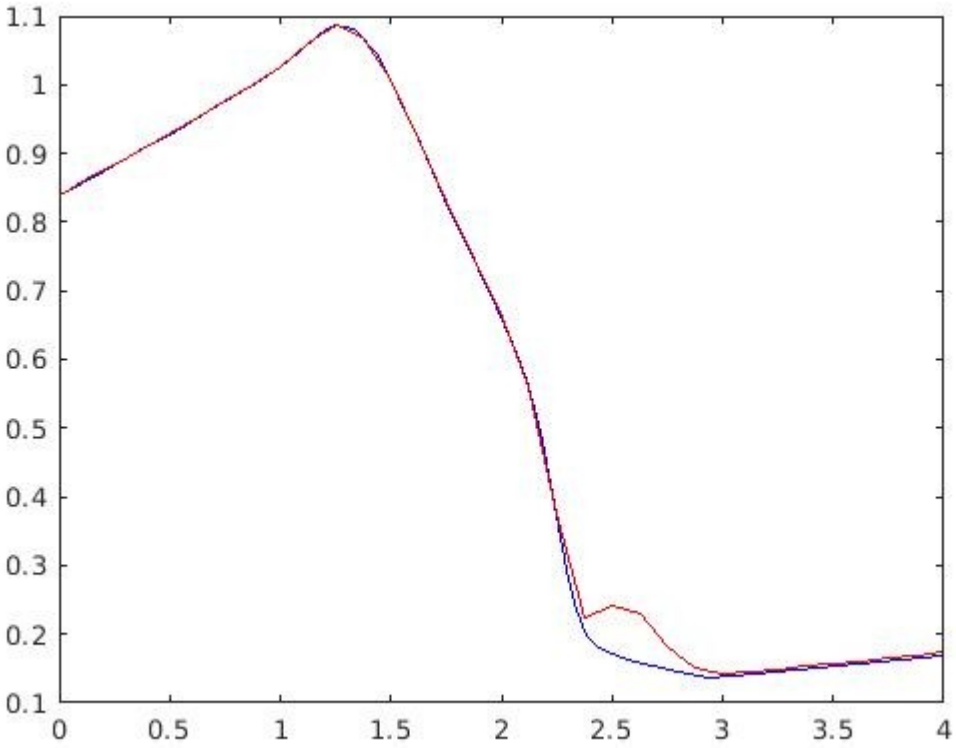
¿Cuántos puntos necesita ahora adapta23 para cubrir el intervalo comparados con los que usó adapta12?

En el algoritmo de adapta12, el tamaño de T (número de puntos), era 8. Usando el método adaptativo 2 + 3 obtenemos que T tiene 34 puntos.

Adjuntar la figura.



Adjuntad nueva gráfica.



La función roja es rk4 y la azul adapta23.

Comparando ambas gráficas, ¿qué solución era la más exacta en el primer caso, adapta23 o rk4 con h=0.125? ¿Cómo es esto posible si usan el mismo número de pasos y rk4 es un método de orden más alto?
La solución más exacta es la de adapta23 porque su h es variable en función de la zona que esté evaluando. Por otro lado rk4 siempre es constante y por lo tanto puede que tenga errores como se puede observar entre 2.5 y 3.

Verificación con una ecuación de 2º orden con solución exacta conocida.

Adjuntad su código.

```
function yp=fun(t,y)
   yp = ((-2*y(2)) - 5*y(1));
return
```

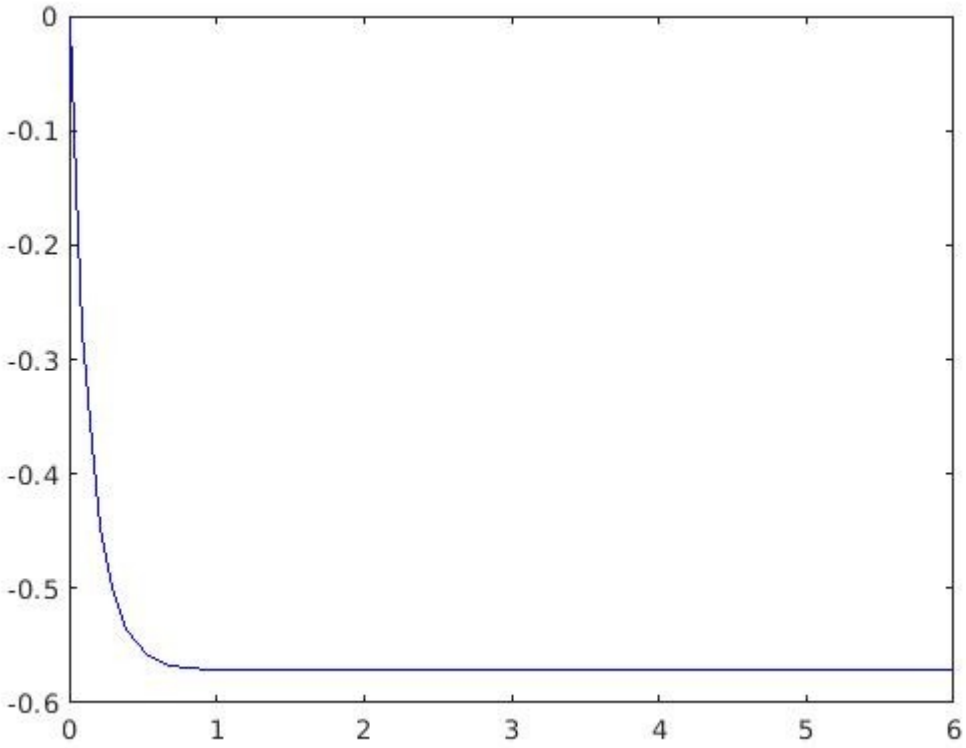
Resolver esta ecuación usando el método adapta23 en el intervalo [0, 6] especificando una tolerancia de 10^-4. Adjuntad el código usado para resolverla y una gráfica de la solución y(t) obtenida en función del tiempo T. ¿Cuántos puntos usa adapta23 para cubrir el intervalo?
Script:

```
intervalo = [0 6];
y0 = [0 2]';
tol = 10e-4;
[T1,S1] = adapta23(@fun, intervalo, y0, tol);
plot(T1,S1(1,:),'b');
```

Adapta23:

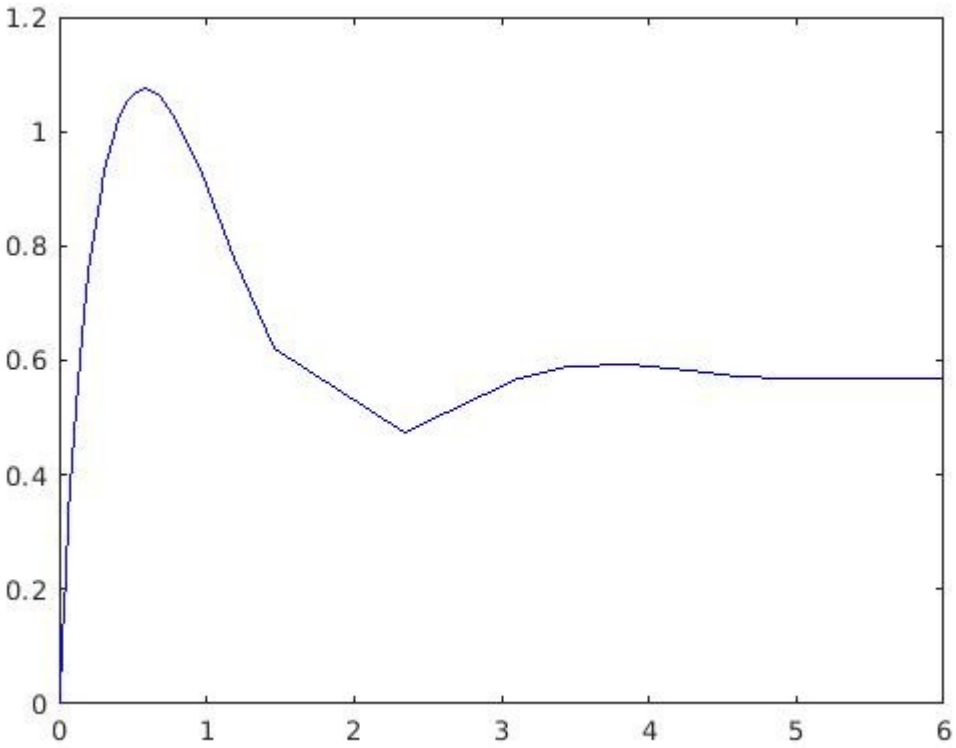
```
function [T, S]=adapta23(fun,Tspan,y0,tol)
   %Metodo adaptativo basado en Heun + Euler
   % fun   -> funcion con la ec. diferencial
```

```
% Tspan -> intervalo [a,b] donde resolver
% y0    -> vector columna de condiciones iniciales
% tol   -> cota del error relativo (p.e 1e-3 o 1e-5)


% Ptos inicial y final del intervalo a resolver
T0 = Tspan(1);
Tfin = Tspan(2);

% Inicializar el primer valor para h
h = (Tfin - T0) / 20;

% Inicializar las salidas T y S con sus valores iniciales
T = T0;
S = y0;

while(T(end)<Tfin)

   % Parto del ultimo punto conocido de la solucïï¿½n
   t=T(end); y=S(:,end);

   % Calcular K1, K2, K3
   K1 = fun(t, y);

   delta = h/2;
   K2 = fun(t + delta, y + delta * K1);

   delta = 3*h/4;
   K3 = fun(t+delta, y+delta*K2);

   % Estimacion orden 3
   s3 = y + (h/9) * (2*K1 + 3*K2 + 4*K3);
   K4 = fun(t+h, s3);

   % Estimacion orden 2
   s2 = y +(h/24)*(7*K1 + 6*K2 + 8*K3 + 3*K4);


   % Calcular estimacion del error E
   E = norm(abs(s2-s3));

    % Calcular ratio r entre E y tol
    ratio = E / tol;

    if ratio<=1   % Paso aceptado, aï¿½adir los nuevos valores a T y S
       T = [T t+h]; S=[S s3];
    end
    % Actualizar h usando la formula optima con un factor adicional 0.8
    % Comprobad que no nos salimos del intervalo en el siguiente salto
    % Recortad h si se da el caso.
    h = (h / ratio^(1/3)) * 0.8;

    if T(end)+h > Tfin
       h = Tfin - T(end);
    end
  end
return
```



En total se usan 21 puntos.

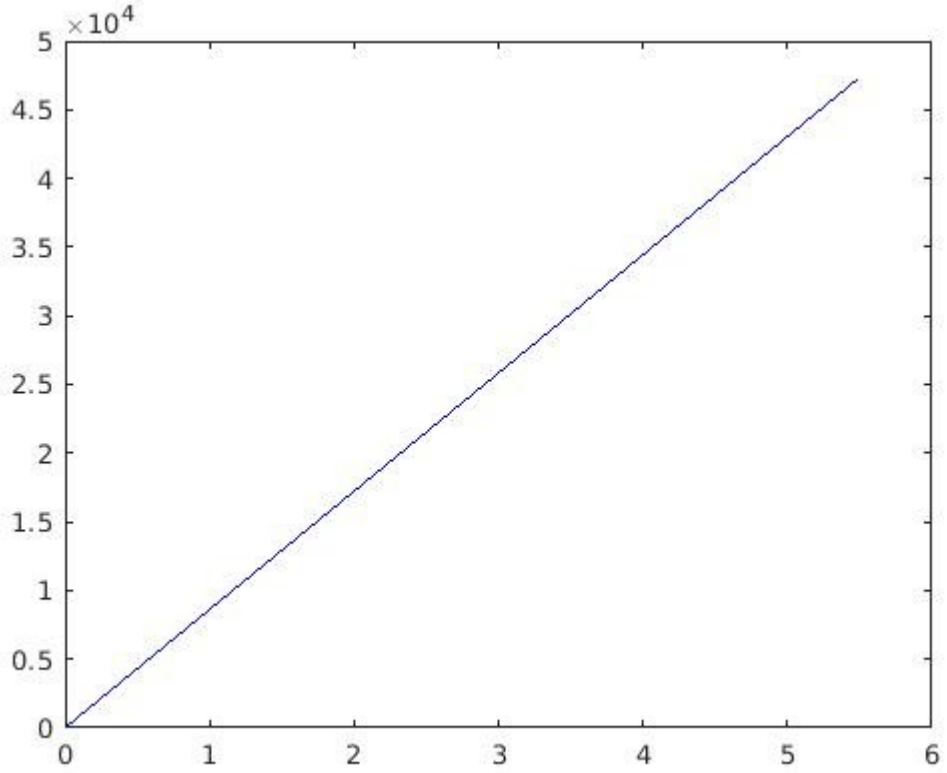
Adjuntad la gráfica del error absoluto entre la solución numérica de adapta23 y la real.
¿Cuál es el error máximo cometido por la solución numérica en el intervalo [0,6]?
¿Cumple más o menos la solución numérica las especificaciones (Eabs<~10^-4)?



El error máximo que hay en de alrededor 1.1. No cumple cunome con el error especificado.

Vamos ahora a usar rk4 para resolver la misma ecuación diferencial en el intervalo [0,6].
¿Qué paso h debéis usar para que la solución de rk4 tenga el mismo número de puntos que la obtenida por adapta23? Justificar
La h que hay que usar será 33/6 = 5.5.

Resolver con rk4 usando el h que acabáis de deducir. Hacer una gráfica del error de la solución numérica obtenida. Adjuntad la gráfica. ¿Cuál es el máximo error cometido por el método rk4? ¿Es ahora el error mayor o menor que el conseguido por adapta23?
¿Por qué creéis que pasa ahora esto?



El error es lineal. El error cometido es $4.7 * 10^4$.

¿Cuántos pasos da ode23 para cubrir el intervalo?
Usa el mismo número que adapta23.