

UNL - FICH

CÁLCULO NUMÉRICO - ENTREGABLE 2

CENTIS MATEO

Enunciado

La energía térmica total de un dispositivo está dada por la expresión $E(t) = ((t + \frac{1}{3})^3 + \frac{1}{3})e^{-t}$ para cada instante de tiempo t .

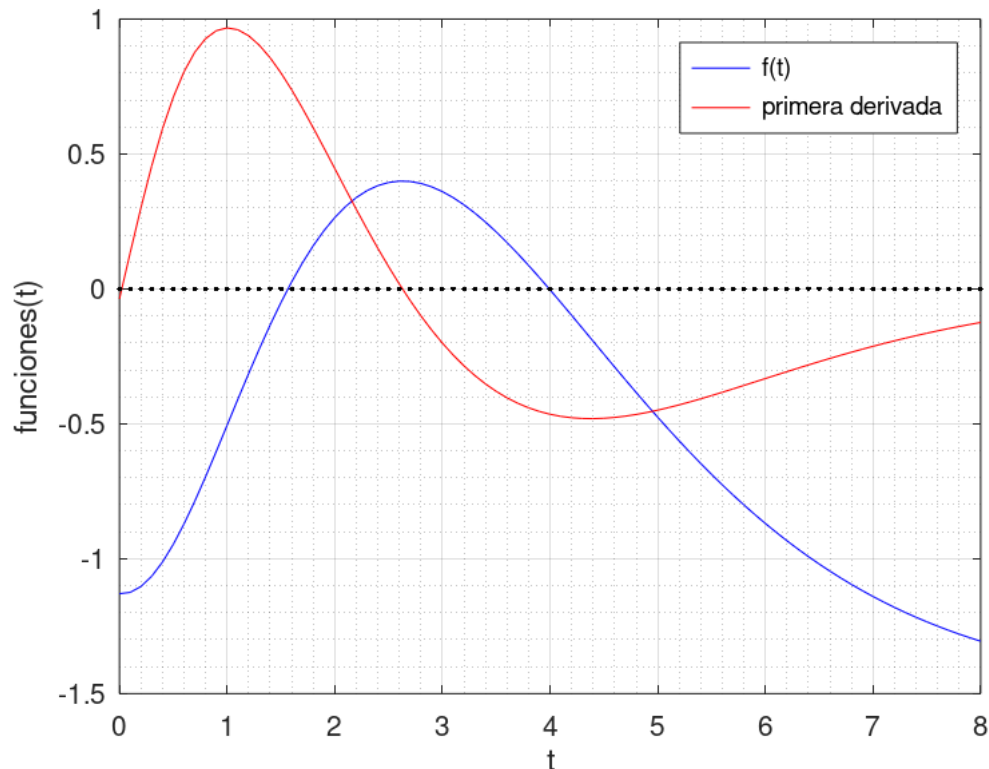
- a. Determinar los instantes de tiempo en los que la energía del dispositivo es igual a 1.5, con 5 dígitos de precisión.
- b. Determinar la máxima energía del sistema y en qué tiempo ocurre, utilizando la misma precisión que en ítem anterior.
- c. Determinar el instante de tiempo en dónde se da la máxima tasa de crecimiento instantánea de la energía respetando la precisión anterior.

Justificar cuál método eligió para resolver cada inciso, destacando las ventajas del mismo.

Resolución

(a)

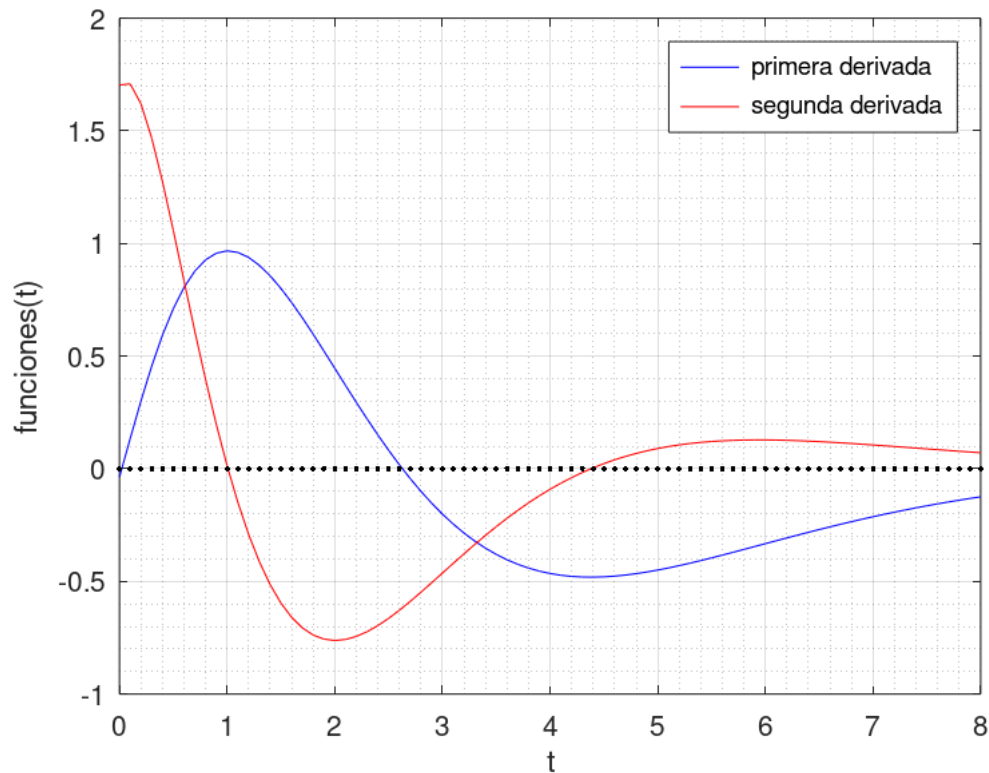
Se debe realizar la búsqueda de los instantes de tiempo dónde $E(t)$ sea igual a 1.5, es decir, $E(t) = 1,5$. Para esto se creó una nueva función $f(t) = E(t) - 1,5$. Y se la igualó a cero para poder utilizar los métodos vistos en la materia. Al realizar la gráfica de la función es posible observar que existen dos raíces entre los intervalos $[1,2]$ y $[3.5,4.5]$.



Para decidir que método utilizar se realizó la derivada de la función $f(t)$, y se encontró $f'(t) = -e^{-t}((t + \frac{1}{3})^3 + \frac{1}{3}) + 3e^{-t}(t + \frac{1}{3})^2$. Se decidió usar el método de Newton-Raphson para encontrar las raíces, esto se debe a que tenemos una buena aproximación inicial y podemos asegurar su convergencia, ya que su derivada es distinta de cero entre los intervalos buscado y además, $f(p) = 0$ y $f'(p) \neq 0$ en ambos p buscados, por lo que son raíces simples y por ende el orden de convergencia del método va a ser cuadrático. Utilizando la tolerancia en valor relativo para detener las iteraciones, se llegó a las raíces $x_{New11} = 1,5649$ y $x_{New12} = 3,9924$ en 4 iteraciones.

(b)

Para determinar la máxima energía del sistema, se buscó el punto dónde la derivada de la función se anule, es decir, su raíz. Se graficó la función para localizar el intervalo dónde se encuentra el punto buscado y se puede notar como este se encuentra en el intervalo $[2,3]$.

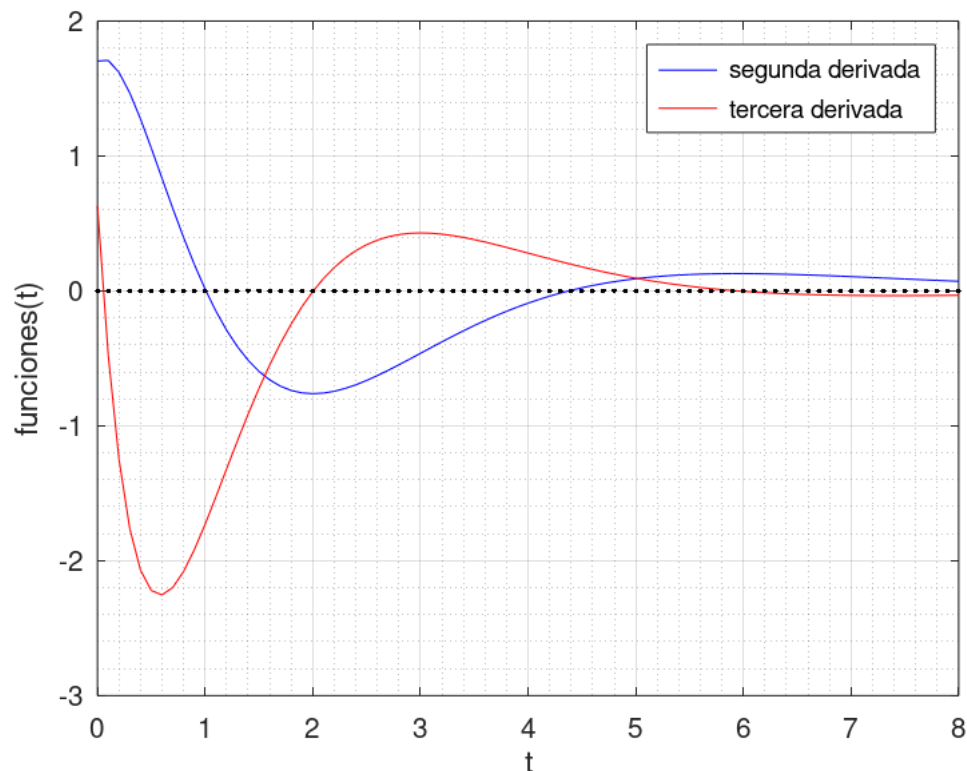


Se derivó nuevamente la función para obtener $f''(t) = \frac{e^{-t}(27t^3 - 135t^2 + 63t + 46)}{27}$ y utilizando un razonamiento similar al usado en el inciso anterior se puede observar que cumple con las condiciones para aplicar el método de Newton-Raphson. Aplicando el método con aproximación inicial $x_{02} = 2,5$ y utilizando mismos criterios para detener las iteraciones se obtuvo la solución $x_{New_2} = 2,6287$ en 3 iteraciones.

(c)

Se desea determinar el instante de tiempo en dónde se de la máxima tasa de crecimiento instantánea de la energía, es decir, encontrar el valor dónde la derivada de la función es máxima. Esto es equivalente a buscar las raíces de la segunda derivada de la función, ya que los máximos de $f'(t)$ son los puntos dónde cambia la concavidad de la $f(t)$ original, es decir, los puntos p tales que $f''(p) = 0$.

Se procedió de forma análoga al inciso anterior y se empezó por graficar $f''(t)$. Al examinarla se puede ver a simple vista que existe una raíz en el intervalo $[4,5]$.



Para encontrar la raíz se utilizó una combinación de los métodos de la bisección y de Newton-Raphson. En primer lugar, se buscó la raíz por bisección en el intervalo $[1,2]$ con una tolerancia mayor $tol_{bis} = 1e - 3$, para luego utilizar Newton-Raphson desde el punto retornado por bisección, con una tolerancia menor. Procediendo de esta manera, se encontró por bisección el punto $x_{Bis} = 1,0083$ luego de 11 iteraciones y utilizándolo en Newton-Raphson se encontró como resultado $x_{New3} = 1,0079$ luego de 2 iteraciones, teniendo como máxima tasa de decrecimiento $f'(x_{New3}) = 0,9674$.

ANEXO

Listing 1: newton.m

```
function [x h1 i] = newton(f,df,x0,kmax,tol)
i=1;
while i < kmax
    x=x0-f(x0)/df(x0);
    h1 = abs(x-x0); %Cambiar a necesidad de error
    h2 = abs(x-x0)/abs(x);
    h3 = abs(f(x));
    if h1 < tol
        return;
    endif
    i++;
    x0=x;
endwhile
disp('No converge en maximas iteraciones')
endfunction
```

Listing 2: biseccion.m

```
function [p h2 i] = biseccion(f,a,b,kmax,tol)
    %a y b son [a,b]
    %h es la convergencia<
i=1;
while i < kmax
    p = a + (b-a)/2;
    h1=f(p); %Elegir que convergencia usar
    h2=abs(b-a);
    h3=abs(b-a)/2;
    if h2<=tol
        return
    endif
    i++;
    if f(b)*f(p) < 0
        a = p;
    else
        b = p;
    endif
endwhile
display('Iteraciones maximas alcanzadas')
endfunction
```

Listing 3: G4Ej7.m

```
E=@(t)((t+1/3).^ (3)+1/3).*e.^(-t); %Ecuacion de energia
tol=1e-5;
kmax=1000;
%funciones y sus derivadas
f=@(t)((t+1/3).^ (3)+1/3).*e.^(-t)-1.5; %funcion de (a)
df=@(t) -e.^(-t).*((t+1/3).^ (3)+1/3)+3*e.^(-t).*(t+1/3).^ (2);
ddf=@(t)(e.^(-t).*(27*t.^3-135.*t.^2+63.*t+46))/27;
ddd=@(t)(1/27).*(-333.*t.*e.^(-t)+
216.*t.^2.*e.^(-t)-27.*t.^3.*e.^(-t)+17.*e.^(-t));
t=[0:0.1:8];
```

```

Et = E(t);
dft = df(t);
ddft=ddf(t);
dddf = dddf(t);
%-----
%(a) Determinar instantes de tiempo en los que la energia es
%igual a 1.5, con 5 digitos de precision
%Grafica de f(t)
ft=f(t);
figure(1)
plot(t,ft,'b');
hold on
plot(t,dft,'r');
hold on
plot(t,0,'k')
set(gca,'XTick',0:1:100)
grid on
grid minor
xlabel('t')
ylabel('funciones(t)')
legend('f(t)','primera derivada')
hold off
%solucion por Newton-Raphson
%intervalo 1
x011=1;
[xNew11 hNew11 itNew11] = newton(f,df,x011,kmax,tol)
%intervalo 2
x012=3.5;
[xNew12 hNew12 itNew12] = newton(f,df,x012,kmax,tol)
%-----
%(b) Determinar max energia del sistema y en que t ocurre, utilizando
%misma tol
%Grafica de df(t)
figure(2)
plot(t,dft,'b');
hold on
plot(t,ddft,'r');
hold on
plot(t,0,'k');
set(gca,'XTick',0:1:100)
grid on
grid minor
xlabel('t')
ylabel('funciones(t)')
legend('primera derivada','segunda derivada')
hold off
%solucion por Newton-Raphson
x02=2.5;
[xNew2 hNew2 itNew2] = newton(df,ddf,x02,kmax,tol)
%-----
%(c) Determinar t en donde se da la max tasa de
%crecimiento instantanea de la energia con misma tol
%Grafica de ddf(t)
figure(3)
plot(t,dddf,'b');
hold on
plot(t,dddf,'r');
hold on
plot(t,0,'k');
set(gca,'XTick',0:1:100)
grid on

```

```
grid minor
xlabel('t')
ylabel("funciones(t)")
legend('segunda derivada','tercera derivada')
hold off
%busqueda por biseccion
tolBis = 1e-3;
[xBis hBis itBis] = biseccion(ddf,1,2,kmax,tolBis)
%solucion por newton-raphson
[xNew3 hNew3 itNew3] = newton(ddf,dddf,xBis,kmax,tol)
```