Trabajo Práctico nº2:

Solución de ecuaciones no lineales

CONSIGNA: La energía térmica total de un dispositivo está dada por la expresión $E(t) = \left(\left(t + \frac{1}{3}\right)^3 + \frac{1}{3}\right) e^{-t}$, 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 donde se da la máxima tasa de crecimiento instantánea de la energía respetando la precisión anterior.

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

RESOLUCIÓN:

Para los tres incisos se decidió utilizar el método de Newton-Raphson, ya que este método es el más eficiente para intervalos pequeños como en el de la función a analizar en este trabajo. Gracias a la posibilidad de graficar las funciones y sus derivadas (las cuales mantienen cierta regularidad por tratarse del producto de una función exponencial y un polinomio), y el dominio reducido de estas (tienden a cero para $t > 10$ y, debido a que esta variable corresponde al tiempo, no puede tomar valores negativos y $t \geq 0$), se pueden hacer estimaciones iniciales cercanas al punto, y se verifica que la derivada es distinta de 0 cuando la función es 0 para cada nueva expresión, escenario óptimo para la convergencia cuadrática del método elegido. Además, se verá que para resolver un inciso se deberá calcular una derivada, la cual se utilizará en el siguiente inciso y así sucesivamente; luego, la desventaja de tener que calcular la derivada de la función no tiene tanto peso en este caso.

Para obtener los instantes de tiempo requeridos por la **consigna a),** se debieron obtener los valores de t tales que $E(t) = 1.5$. Despejando, se obtiene $E(t) - 1.5 = 0$, por lo que la solución a nuestro problema son las raíces de esta nueva expresión.

El método de Newton-Raphson se implementó en una función que devuelve el punto buscado, un vector de residuos, la cantidad de iteraciones y el tiempo de ejecución. En cuanto a los argumentos de entrada, se precisaron la función, su derivada, el punto de aproximación, el máximo de iteraciones y la tolerancia.

Como primer paso, se debió obtener la derivada de E(t), $\frac{dE(t)}{t} = e^{-t}\left({}^{1}\!/_{3} + 3 \cdot \left(t + {}^{1}\!/_{3}\right)^2 - \left(t + {}^{1}\!/_{3}\right)^3\right)$. En el código, se declaró con el nombre de variable dE.

A continuación, se graficaron tanto E(t) como E(t) – 1.5 y su derivada dE(t) (la cual coincide en ambos casos ya que difieren en una constante). Ver Figura 1.

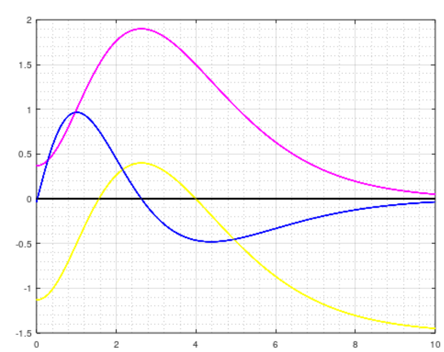

Figura 1: Gráfica de E(t) (curva rosada), E(t) – 1.5 (curva amarilla), dE(t) (curva azul). También se graficó y = 0 (recta negra) para resaltar el eje x.

A partir de esta gráfica, se pudo observar que la función E(t) – 1.5 cuenta con dos raíces, por lo que el método debe ser aplicado dos veces, con distintos puntos de aproximación. Se tomó un $p_0 = 2$ para el primer instante y $p_0 = 4$ para el segundo, al ser los números enteros más cercanos a los ceros de la función, como es observable en la intersección de la curva amarilla y el eje x (esto es, la función y = 0).

En cuanto a la tolerancia precisada para el método, si bien el enunciado no lo especifica de forma explícita, se piden 5 dígitos de precisión. Tomando este requerimiento, es posible pensar en una tolerancia de 10^{-6} ya que se tendrá un entero en las soluciones. Por último, el argumento de iteraciones máximas se estableció en 50, aunque se sabe que el método convergerá rápidamente al tener una aproximación muy cercana de donde se encuentran las raíces por el dominio reducido de la función y sus derivadas.

Con dichos argumentos, se llamó a la función newtonRaphson.m y se obtuvo:

```
INCISO A:
El instante de tiempo (1) en el que la energía del dispositivo es igual a 1.5 es:
pA = 1.5649
El número de iteraciones realizadas para determinarlo:
itA = 5

El instante de tiempo (2) en el que la energía del dispositivo es igual a 1.5 es:
pA = 3.9924
El número de iteraciones realizadas para determinarlo:
itA = 3
```

Los instantes de tiempo en los que la energía del dispositivo es igual a 1.5, con 5 dígitos de precisión, son t = 1.5649 y t = 3.9924. El cálculo del primer instante tomó 5 iteraciones, mientras que el segundo tomó 3 iteraciones.

En cuanto al **inciso b),** para obtener los extremos locales de la función E(t), por el criterio de la primera derivada, se deben obtener las raíces de su primera derivada dE(t). Debido a que las raíces pueden encontrarse en un punto mínimo o máximo de la función sin derivar, primero es necesario observar la gráfica en detalle.

Tras calcular la segunda derivada de E(t), $\frac{d_2E(t)}{t^2} = e^{-t}\left({}^{1}\!/_{3} + 6 \cdot \left(t + {}^{1}\!/_{3}\right) - 6 \cdot \left(t + {}^{1}\!/_{3}\right)^2 + \left(t + {}^{1}\!/_{3}\right)^3\right)$ (nombre de variable d2E en el script), se graficaron las funciones E(t), dE(t) y d2E(t) para poder encontrar un punto de aproximación adecuado (ver Figura 2). Observando la intersección entre la recta y = 0 y dEt cerca de la zona en la que E(t) tiene su máximo, se logró determinar un punto aproximado $p_0 = 3$.

La tolerancia se mantiene respecto al inciso anterior, lo mismo con las iteraciones máximas, y se llama a la función newtonRaphson.m con estos argumentos, obteniéndose los siguientes resultados:

```
INCISO B:
El tiempo en el que se da la máxima energía es:
pB = 2.6287
El número de iteraciones realizadas para determinarlo:
itB = 4
La máxima energía es:
ans = 1.8997
```

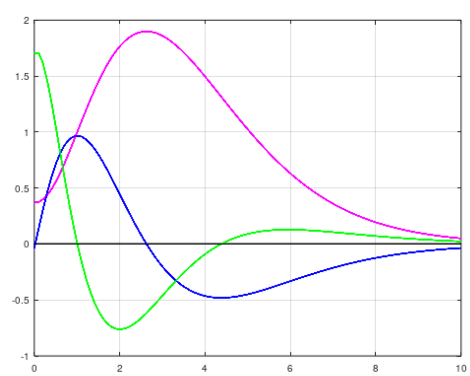

Figura 2: Gráfica de E(t) (curva rosada), dE(t) (curva azul), d2E(t) (curva verde) e y = 0 (recta negra).

Del método Newton-Raphson (el cual tomó 4 iteraciones) obtenemos el tiempo en el que se da la máxima energía, pB = 2.6287 tomando 5 dígitos de precisión. Para obtener la máxima energía, evaluamos la función E(t) en t = pB = 2.6287 y obtenemos nuestra respuesta: la máxima energía térmica del dispositivo es 1.8997 (unidad desconocida de energía).

En **el inciso c)** se precisó obtener el máximo de la tasa de crecimiento. Para lograr esto, se debió encontrar el máximo de la función derivada dE(t), la cual representa las pendientes de las rectas tangentes de los puntos de E(t). Luego, maximizar esta función es obtener la mayor pendiente, es decir, en qué instante de tiempo la función crece más. Esto es posible obteniendo las raíces de la segunda derivada d2E(t), que ya fue calculada en el inciso anterior.

Para aplicar Newton-Raphson, se debió calcular una derivada más. La tercera derivada de E(t) es $\frac{d_3E(t)}{t^3} = e^{-t}\left(6 - {}^1/_3 - 18 \cdot \left(t + {}^1/_3\right) + 9 \cdot \left(t + {}^1/_3\right)^2 - \left(t + {}^1/_3\right)^3\right)$. Se define d3E en el script.

El punto de aproximación se pudo obtener observando la gráfica, viendo que el máximo de la primera derivada (curva azul) se da en un valor cercano a t = 1, mejor observado en la intersección de la segunda derivada (curva verde) con el eje x (ver Figura 3). Luego, p0 = 1, y se mantienen los valores de tolerancia e iteraciones máximas de los incisos previos. Obtenidos los argumentos necesarios, se llamó a la función y se obtuvieron los siguientes resultados:

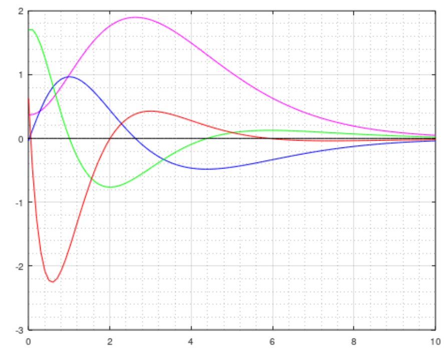

Figura 3: Gráfica de E(t) (curva rosada), dE(t) (curva azul), d2E(t) (curva verde), d3E(t) (curva roja) e y = 0 (recta negra).

```
INCISO C:
El instante de tiempo donde se da la máxima tasa de crecimiento instantánea es:
pC = 1.0079
El número de iteraciones realizadas para determinarlo:
itC = 3
```

Luego, el instante de tiempo donde se da la máxima tasa de crecimiento instantánea de la energía, con cinco dígitos de precisión, es t = 1.0079. Este cálculo se logró en 3 iteraciones.

FUNCIONES Y SCRIPTS UTILIZADOS

Script TP2.m:

```
# INCISO A:
disp('INCISO A:');
E = @(t) ((t+1/3).^3+1/3).*e.^(-t);
# E(t) = 1.5 => E(t) - 1.5 = 0:
Ea = @(t) ((t+1/3).^3+1/3).*e.^(-t) - 1.5;
dE = @(t) (e.^(-t).*(3.*(t+1/3).^2-(t+1/3).^3-1/3));

# grafico E(t) y su primera derivada junto a y = 0
figure(1);
t = linspace(0,10,100);
plot(t, E(t), '-m');
hold on;
plot(t, Ea(t), '-y');
plot(t, dE(t), '-b');
y = @(t) t.*0;
grid on;
grid minor;
plot(t,y(t),'-k');

# para determinar p0 uso aproximaciones de las raíces
# que, tras ver en la gráfica, se acercan a 2 y a 4.
p0 = 2;
tol = 10^-6;
maxit = 50;
[pA, r_hA, itA, tA] = newtonRaphson(Ea, dE, p0, maxit, tol);
disp('El instante de tiempo (1) en el que la energía del dispositivo es igual a 1.5 es:');
pA
disp('El número de iteraciones realizadas para determinarlo:');
itA

p0 = 4;
disp('');
[pA, r_hA, itA, tA] = newtonRaphson(Ea, dE, p0, maxit, tol);
disp('El instante de tiempo (2) en el que la energía del dispositivo es igual a 1.5 es:');
pA
disp('El número de iteraciones realizadas para determinarlo:');
itA

# INCISO B:
disp('');
disp('INCISO B:');
d2E = @(t) e.^(-t).*((t+1/3).^3-6.*(t+1/3).^2+6.*(t+1/3)+1/3);

# grafico E(t), su primera y segunda derivada junto a y = 0.
figure(2);
t = linspace(0,10,100);
plot(t, E(t), '-m');
hold on;
plot(t, dE(t), '-b');
plot(t, d2E(t), '-g');
y = @(t) t.*0;
grid on;
grid minor;
plot(t,y(t),'-k');

# tras observer la gráfica, aproximo:
p0=3;
[pB, r_hB, itB, tB] = newtonRaphson(dE, d2E, p0, maxit, tol);
disp('El tiempo en el que se da la máxima energía es:');
pB
disp('El número de iteraciones realizadas para determinarlo:');
itB
disp('La máxima energía es:');
maxE=E(pB)
```

```
# INCISO C:
disp('');
disp('INCISO C:');
d3E = @(t) e.^(-t).*(6-1/3-18.*(t+1/3)+9.*(t+1/3).^2-(t+1/3).^3);

# grafico E(t), su primera, segunda y tercera derivada junto a y = 0.
figure(3);
t = linspace(0,10,100);
plot(t, E(t), '-m');
hold on;
plot(t, dE(t), '-b');
plot(t, d2E(t), '-g');
plot(t, d3E(t), '-r');
y = @(t) t.*0;
grid on;
grid minor;
plot(t,y(t),'-k');

# tras observer la gráfica, aproximo:
p0=1;
disp('El instante de tiempo donde se da la máxima tasa de crecimiento instantánea es:');
[pC, r_hC, itC, tC] = newtonRaphson(d2E, d3E, p0, maxit, tol);
pC
disp('El número de iteraciones realizadas para determinarlo:');
itC
```

Función newtonRaphson.m:

```
function [p, r_h, it, t] = newtonRaphson(f, df, p0, maxit, tol)
  tic();
  it = 1;
  fp=f(p0);
  while it < maxit
    p = p0 - (fp/df(p0));
    fp = f(p);
    r_h(it)=norm(((p-p0)),inf);
    if r_h(it) < (tol*abs(p))
      break;
    endif
    it = it + 1;
    p0 = p;
  endwhile

  t = toc();
endfunction
```