EL2 - 2022_2

Problema 1 - Interpolación (variante)

Editar | Accione

Resumen del problema

La ecuación de Michaelis - Menten aparece en cinética química y tiene la forma:

$$v(x) = \frac{ax}{b+x}$$

Determine los coeficientes a y b de tal forma que la ecuación anterior sea una aproximación por mínimos cuadrados a los datos obtenidos en un experimento para medir la velocidad v de una reacción catalizada por enzimas a varias concentraciones x.

Datos obtenidos en un experimento de Michaelis - Menten:

x	0.197	0.139	0.068	0.0427	0.027	0.015	0.009	0.008
v	21.5	21	19	16.5	14.5	11	8.5	7

Confiamos en que usted es un estudiante muy estudioso y entiende que el objetivo en este ejercicio es minimizar la suma de cuadrados S dada por:

$$S = \sum_{i=0}^{7}\left(v_i - \frac{ax_i}{b+x_i}\right)^2$$

donde x_i y v_i con $i = 0, 1, \ldots, 7$ son los puntos datos obtenidos. Sucede en este caso, que al querer encontrar las derivadas parciales $\frac{\partial S}{\partial a}$ y $\frac{\partial S}{\partial b}$ e igualarlas a 0, resultan ecuaciones no lineales para a y b (ojo que usted podría resolver esas ecuaciones no lineales con los métodos aprendidos: bisección, Newton - Raphson, punto fijo, entre otros). Sin embargo, sus profesores desean que resuelvan el ejercicio en el menor tiempo posible. Por ello, se les da como sugerencia minimizar la función P en lugar de S, donde:

$$P = \sum_{i=0}^{7}(v_i(b+x_i) - ax_i)^2$$

Se le pide a usted, efectuar lo siguiente:

1. Derivar parcialmente la función P respecto a ciertas variables (usted debe identificar cuales son) para poder plantear el sistema de ecuaciones que posteriormente permita encontrar los valores de a y b.
2. Resolver el sistema planteado en 1, en la variable **a** almacenar a y en la variable **b**, almacenar el valor de b.
3. En la variable **aprox1** almacenar v(0.13) con la aproximación de mínimos cuadrados obtenida en el ítem anterior.

```
format long

syms vi xi ai bi
P = sum((vi*(bi+xi)-ai*xi)^2)
```

P = $(\mathrm{vi}\ (\mathrm{bi} + \xi) - \mathrm{ai}\ \xi)^2$

```
dP_da = diff(P,ai)
```

dP_da = $-2\,\xi\ (\mathrm{vi}\ (\mathrm{bi} + \xi) - \mathrm{ai}\ \xi)$

```
dP_db = diff(P,bi)
```

dP_db = $2\,\mathrm{vi}\,(\mathrm{vi}\,(\mathrm{bi}+\xi)-\mathrm{ai}\,\xi)$

```
% dP_da = 0
% sum(-2*xi*(vi*(bi + xi) - ai*xi)) = 0
% sum(xi*(vi*(bi + xi) - ai*xi)) = 0
% sum(xi*(vi*bi + vi*xi - ai*xi)) = 0
% sum(xi*vi*bi) + sum(vi*xi^2) - sum(ai*xi^2) = 0
% bi*sum(xi*vi) + sum(vi*xi^2) - ai*sum(xi^2) = 0
% sum(vi*xi^2) = ai*sum(xi^2) - bi*sum(xi*vi)

% dP_db = 0
% 2*vi*(vi*(bi + xi) - ai*xi) = 0
% sum(bi*vi^2) + sum(xi*vi^2) - sum(ai*xi*vi) = 0
% bi*sum(vi^2) + sum(xi*vi^2) - ai*sum(xi*vi) = 0
% sum(xi*vi^2) = ai*sum(xi*vi) - bi*sum(vi^2)


% sum(vi*xi^2) = ai*sum(xi^2) - bi*sum(xi*vi)
% sum(xi*vi^2) = ai*sum(xi*vi) - bi*sum(vi^2)

x = [0.197 0.139 0.068 0.0427 0.027 0.015 0.009 0.008]
```

```
x = 1×8
   0.197000000000000   0.139000000000000   0.068000000000000   0.042700000000000 ···
```

```
v = [21.5 21 19 16.5 14.5 11 8.5 7]
```

```
v = 1×8
  21.500000000000000  21.000000000000000  19.000000000000000  16.500000000000000 ···
```

```
sumvx2 = sum(v.*(x.^2))
```

```
sumvx2 =
   1.372256785000000
```

```
sumxv2 = sum(x.*(v.^2))
```

```
sumxv2 =
     1.970693250000000e+02
```

```
% Vector de Terminos Independientes
c = [sumvx2; sumxv2]
```

```
c = 2×1
10^2 ×
   0.013722567850000
   1.970693250000000
```

```
sumx2 = sum(x.^2)
```

```
sumx2 =
```

```
   0.065676290000000
```

```
sumxv = sum(x.*v)
```

```
sumxv =
   9.840049999999998
```

```
sumv2 = sum(v.^2)
```

```
sumv2 =
        1989
```

```
A = [sumx2 -sumxv; sumxv -sumv2]
```

```
A = 2×2
10^3 ×
   0.000065676290000  -0.009840050000000
   0.009840050000000  -1.989000000000000
```

```
x = A\c
```

```
x = 2×1
  23.377620834367885
   0.016574978829171
```

```
a = x(1)
```

```
a =
  23.377620834367885
```

```
b = x(2)
```

```
b =
   0.016574978829171
```

```
v = @(x) a*x/(b+x)
```

```
v = function_handle with value:
    @(x)a*x/(b+x)
```

```
aprox1 = v(0.13)
```

```
aprox1 =
  20.734034776902774
```

Método de extrapolación de Richardson

Un procedimiento basado en la extrapolación de Richardson utiliza dos estimaciones de la derivada para calcular una tercera aproximación más exacta.
Una aproximación de la derivada se escribirá como:

$$D \approx \frac{4}{3}D(h_2) - \frac{1}{3}D(h_1)$$

cuando $h_2 = h_1/2$.
Se conoce que la trayectoria de una partícula está dada por:

$$x(t) = \arctan(e^t - 1{,}5t - 0{,}5)$$

a) Hallar la aproximación de la rapidez en $t = 0{,}2$ usando la aproximación de la derivada con 4 puntos con $h = 0{,}01$ y almacenarlo en la variable $vel1$.

b) Repetir el paso anterior con $h = 0{,}005$ y almacenar la aproximación en la variable $vel2$.

c) Use el método de extrapolación de Richardson para obtener una mejor aproximación y almacenelo en la variable $vel3$.

d) Hallar la aproximación de la rapidez en $t = 1$ usando el método de extrapolación de Richardson con $h = 0{,}01$ y almacene su respuesta en la variable $vel4$.

e) Si la partícula tiene velocidad positiva en $t = 0{,}2$ defina la variable $signo1$ con valor 1 y si es negativa defina la variable $signo1$ con valor -1.

f) Si la partícula tiene velocidad positiva en $t = 1$ defina la variable $signo2$ con valor 1 y si es negativa defina la variable $signo2$ con valor -1.

```
format long

fa=@(t) atan(exp(t)-1.5*t-0.5)
```

```
fa = function_handle with value:
    @(t)atan(exp(t)-1.5*t-0.5)
```

```
% Item a
t0 = 0.2
```

```
t0 =
   0.200000000000000
```

```
h = 0.01
```

```
h =
   0.010000000000000
```

```
vel1 = (fa(t0-2*h)-8*fa(t0-h)+8*fa(t0+h)-fa(t0+2*h))/(12*h)
```

```
vel1 =
  -0.236584497450491
```

```
% Item b
h = 0.005
```

```
h =
   0.005000000000000
```

```
vel2 = (fa(t0-2*h)-8*fa(t0-h)+8*fa(t0+h)-fa(t0+2*h))/(12*h)
```

```
vel2 =
  -0.236584499045728
```

```
% Item c
vel3 = (4/3)*vel2 - (1/3)*vel1
```

```
vel3 =
  -0.236584499577473
```

```
% Item d
% Derivada de 4 puntos
t0 = 1
```

```
t0 =
     1
```

```
h = 0.01
```

```
h =
   0.010000000000000
```

```
vel4 = (fa(t0-2*h)-8*fa(t0-h)+8*fa(t0+h)-fa(t0+2*h))/(12*h)
```

```
vel4 =
   0.803653715461970
```

```
% Item e
signo1 = 1
```

```
signo1 =
     1
```

```
if (vel3 < 0)
    signo1 = -1
end
```

```
signo1 =
    -1
```

```
% Item d
signo2 = 1
```

```
signo2 =
     1
```

```
if (vel4 < 0)
    signo2 = -1
end
```

Problema 3 - Diferencias finitas

Editar | Acciones

Resumen del problema

Resolver la siguiente ecuación diferencial mediante el método de diferencias finitas:

$$\begin{cases} y'' = 4y \\ y(0) = 1 \\ y(1) = 3 \end{cases}$$

Se le pide hacer lo siguiente:

1. Resolver la EDO de forma exacta (se sabe que esta viene dada por ($y(t) = c_1 e^{2t} + c_2 e^{-2t}$ donde c_1 y c_2 son constantes a determinar) y almacenar el valor exacto de y(0.75) en la variable **primaprox**.
2. Resolver la EDO por el método de diferencias finitas y almacenar en un vector **v** columna las aproximaciones y(0), y(0.25), y(0.5), y(0.75), y(1) en ese orden.
3. En la variable err almacenar el error relativo con el que el método de diferencias finitas aproxima a y(0.75) en la forma exacta.

```
format long

% Item 1

syms y(x)
D2y = diff(y,2)
```

D2y(x) =

$$\frac{\partial^2}{\partial x^2}\, y(x)$$

```
y(x) = dsolve(D2y==4*(y),y(0)==1,y(1)==3)
```

y(x) =

$$\frac{e^{2x}\,(3\,e^2-1)}{e^4-1} - \frac{e^{-2x}\,(3\,e^2-e^4)}{e^4-1}$$

```
yu = matlabFunction(y(x))
```

```
yu = function_handle with value:
    @(x)-(exp(x.*-2.0).*(exp(2.0).*3.0-exp(4.0)))./(exp(4.0)-1.0)+(exp(x.*2.0).*(exp(2.0).*3.0-1.0))./(exp(4.0)-1.0)
```

```
primaprox = yu(0.75)
```

```
primaprox = 
   1.904935093677754
```

```
% Item 2

[c,v]=dif_fin([0 1],[1 3],3)
```

```
c = 3×1
   1.024943310657596
   1.306122448979592
   1.913832199546485
v = 1×5
   1.000000000000000   1.018116209707681   1.296108547327770   1.904935093677754 ···
```

```
v = v'
```

```
v = 5×1
   1.000000000000000
   1.018116209707681
   1.296108547327770
   1.904935093677754
   3.000000000000000
```

```
% Item 3

err = abs(primaprox - v(4)) / abs(primaprox)
```

```
err = 
     0
```

Función de Cobb-Douglas

Resumen del problema

Se conoce que la función de producción de un cierto bien es del tipo Cobb-Douglas, es decir, tiene la forma

$$F(K, L) = 4K^{\alpha}L^{\beta}$$

y se tienen los siguientes datos:

K	10	15	20	20
L	12	18	12	15
F	44,6240	66,9360	58,8818	67,3173

Usando ajuste por mínimos cuadrados:

a) Halle el valor de α y almacénelo en la variable *alpha_cb*.

b) Halle el valor de β y almacénelo en la variable *beta_cb*.

c) Estime la producción que se obtendrá para $K = 12$, $L = 25$, y almacénelo en la variable $Faprox$.

Se sabe además que K y L satisfacen la restricción presupuestaria:

$$2K + 3L = 45$$

Use el método de la bisección con 10 iteraciones para hallar el valor de L para obtener una producción de $F(K, L) = 31{,}0369$ unidades. Usar el intervalo inicial [4,8].

d) De como respuesta el valor de L almacenándolo en la variable L.

e) De como respuesta el valor de K, a partir del valor de L hallado en el ítem anterior, almacene su respuesta en la variable K.

```
format long

Ki = [10 15 20 20]
```

```
Ki = 1×4
    10    15    20    20
```

```
Li = [12 18 12 15]
```

```
Li = 1×4
    12    18    12    15
```

```
Fi=[44.6240 66.9360 58.8818 67.3173]
```

```
Fi = 1×4
  44.624000000000002  66.936000000000007  58.881799999999998  67.317300000000003
```

```
% F = 4*(K^a)*(L^b)
% F/4 = (K^a)*(L^b)
% ln ---> log
% ln(F/4) = ln((K^a)*(L^b))
% (ln(F) - ln(4)) = ln(K^a)+ln(L^b)
% (ln(F) - ln(4))/ln(L) = (ln(K^a)+ln(L^b))/Ln(L)
% (ln(F) - ln(4))/ln(L) = a*Ln(K)/Ln(L)+b*Ln(L))/Ln(L)
% (ln(F) - ln(4))/ln(L) = a*Ln(K)/Ln(L) + b

Y = (log(Fi)-log(4))./log(Li)
```

```
Y = 1×4
   0.970651138935549   0.974768235323528   1.082228857053871   1.042492823544221
```

```
X = log(Ki)./log(Li)
```

```
X = 1×4
   0.926628408029127   0.936921070344715   1.205571353680257   1.106232178537418
```

```
p = polyfit(X,Y,1)
```

```
p = 1×2
   0.400001719588163   0.599998167683237
```

```
alpha_cb = p(1)
```

```
alpha_cb = 
   0.400001719588163
```

```
beta_cb = p(2)
```

```
beta_cb = 
   0.599998167683237
```

```
f1 = @(K,L) 4*(K^alpha_cb)*(L^beta_cb)
```

```
f1 = function_handle with value:
    @(K,L)4*(K^alpha_cb)*(L^beta_cb)
```

```
Faprox = f1(12,25)
```

```
Faprox = 
  74.558264307736792
```

```
g = @(L) 4*((45/2-3*L/2)^alpha_cb)*(L^beta_cb) - 31.0369
```

```
g = function_handle with value:
    @(L)4*((45/2-3*L/2)^alpha_cb)*(L^beta_cb)-31.0369
```

```
[z,L]=biseccion(g,4,8,10)
```

```
z = 11×7
   4.000000000000000   8.000000000000000   6.000000000000000  -2.834208669292035 ···
   4.000000000000000   6.000000000000000   5.000000000000000  -2.834208669292035
   4.000000000000000   5.000000000000000   4.500000000000000  -2.834208669292035
   4.500000000000000   5.000000000000000   4.750000000000000  -1.327041404300708
   4.750000000000000   5.000000000000000   4.875000000000000  -0.641839805625878
   4.875000000000000   5.000000000000000   4.937500000000000  -0.315567354899031
   4.937500000000000   5.000000000000000   4.968750000000000  -0.156431981208506
   4.968750000000000   5.000000000000000   4.984375000000000  -0.077855162866562
   4.984375000000000   5.000000000000000   4.992187500000000  -0.038813350595415
   4.992187500000000   5.000000000000000   4.996093750000000  -0.019353956565041
      ⋮
L = 
   4.998046875000000
```

```
K=45/2-3*L/2
```

```
K = 
  15.002929687500000
```

Viento extremo

Editar | Accione

En ingeniería eléctrica se aborda el problema del cálculo de la potencia disponible para una turbina eólica. Este cálculo se basa en la densidad del aire, el área de barrido de las palas de la turbina (imagínese un gran círculo formado por las palas giratorias) y la velocidad del viento.

Un análisis de frecuencias del viento nos permite conocer la forma de distribución de los datos y obtener su porcentaje de probabilidad de ocurrencia. La distribución de Weibull (Weibull, 1951) es una distribución típicamente utilizada en meteorología, específicamente en el análisis de velocidad del viento. Su expresión matemática se muestra a continuación

$$f(v) = \left(\frac{k}{c}\right)\left(\frac{v}{c}\right)^{k-1} e^{-\left(\frac{v}{c}\right)^k}$$

Donde:

$f(v)$: Función de densidad de probabilidad de Weibull

k: Factor de forma (adimensional)

c: Factor de escala (m/s)

v: Velocidad del viento (m/s)

Para nuestro estudio la variable en cuestión es la velocidad del viento, por lo que los valores que toma $f(v)$ indican la probabilidad de observar cada velocidad del viento. El parámetro k representa el rango de variación de la velocidad del viento durante un periodo de tiempo, mientras que el parámetro c tiene unidades de m/s y está relacionado con la media de la velocidad del viento.

Sea $k = 2.1939$, $c = 3.348$ m/s.

Realice los siguientes pasos.

- Aproxime la integral $\int_0^x f(v)dv$ para x = 5, mediante el método de cuadratura de Gauss con 20 puntos y asigne el resultado a la variable **pv5**.
- El objetivo es hallar el valor de la velocidad del viento v_{95} (conocido como indicador de viento fuerte) tal que:

$\int_0^{v_{95}} f(v)dv = 0.95$. Use el método de Newton-Raphson, con una tolerancia igual a 0.000001, para hallar el indicador de viento fuerte. Asigne el resultado a la variable **v95**.

Sugerencia: Use las funciones trabajadas en el curso y/o adapte el método de Newton-Raphson teniendo en cuenta que:

$$\frac{d}{dx}\int_0^x f(v)dv = f(x)$$

para todo $x > 0$.

```
format long

pv5 = F(5)
```

```
pv5 =
   0.910247298613594
```

```
Tol=0.000001; x0=5;
error=1; z=[x0 error];
```

```
while error > Tol
    x1 = x0-(F(x0)-0.95)/f(x0);
    error=abs(x1 - x0)/abs(x1);
    z=[z; x1 error];
    x0=x1;
end

v95 = z(end,1)
```

```
v95 =
   5.520528163337224
```

Funciones

Biseccion

```
function [z,vaprox]=biseccion(f,a,b,Maxiter)
c=(a+b)/2;
error=(b-a)/2;
z=[a b c f(a) f(b) f(c) error];
for k=1:Maxiter
    if f(a)*f(c)<0
        b=c;
    else
        a=c;
    end
    c=(a+b)/2;
    error=(b-a)/2;
    z=[z;a b c f(a) f(b) f(c) error];
end
vaprox=c;
end
```

Diferencias Finitas

```
function [c,v] = dif_fin(inter,rv,n)
a = inter(1); b = inter(2); ya = rv(1); yb = rv(2);
sol = ones(n,1); h = (b-a)/(n+1);
alfa = -4*h*h-2;
M = spdiags([sol alfa*sol sol],-1:1,n,n);
d = zeros(n,1); d(1)=-ya; d(n)=-yb;
t = 0:h:b;

e = exp(2)-exp(-2);
v = (3-exp(-2))*exp(2*t)/e+(exp(2)-3)*exp(-2*t)/e;

c = M\d;

end
```

Gauss

```
function p = f(v)
c=3.348;
k=2.1939;
p = (k/c)*(v/c)^(k-1)*exp(-(v/c)^k);
end

function I = F(x)
I = gaussnp(@f,0,x,20);
end

function G = gaussnp(f,a,b,n)

syms x
le = legendreP(n,x);

dle = matlabFunction(diff(le,x));
roots = double(vpasolve(le==0));

pesos = (2./((1-roots.^2).*dle(roots).^2));

syms t
x=((b-a)*t+(a+b))/2;
F=matlabFunction(f(x));

G=((b-a)/2)*sum(pesos.*(F(roots)));

end
```