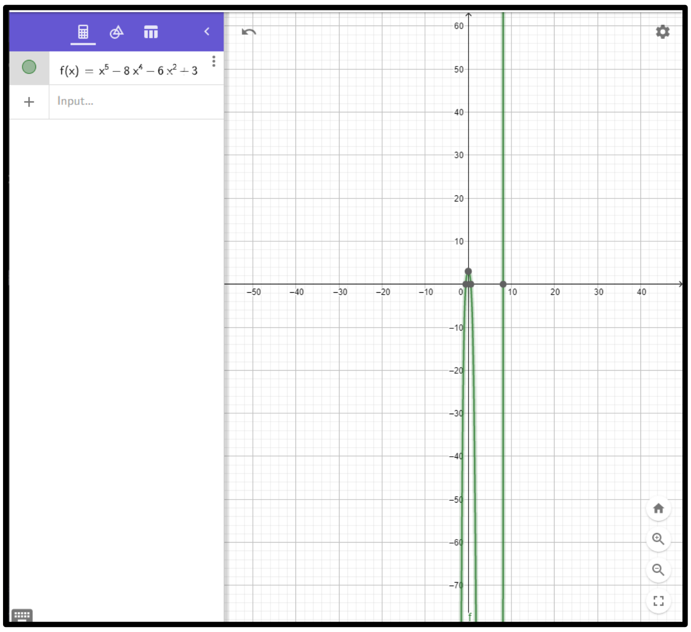
3η εργασία μαθήματος

«Ειδικά θέματα επιχειρησιακής έρευνας»

Αντώνιος Ελευθέριος Καρναβάς, ΑΜ:Π18063

- <u>Άσκηση 1</u>

Εφόσον ο ΑΜ είναι Π18063 η f της πρώτης άσκησης θα έχει τη μορφή `f(x) = x^5 - 8*x^4 - 6*x^2 + 3`. Η γραφική της παράσταση είναι η παρακάτω:



Ερώτημα α)

Κώδικας matlab:

```
%% Bisection search method
% AM=P18063 -->  1*x^5 - 8*x^4 + 0*x^3 - 6*x^2 + 3

% Equation: f(x) = x^5 - 8*x^4 - 6*x^2 + 3
% Derivative: F=f'(x) = 5*x^4-32*x^3-12*x

F =@(x) (1*x.^5)-(8*x.^4)-(6*x^2)+3; % given function
f=@(x) (5*x.^4)-(32*x.^3)-(12*x); % derivative
e = 10^(-6);
a = -10;
b = 10;

while ( b-a>=0||(abs(f(a))>=e && abs(f(b))>=e) )
    c=(a+b)/2;
    if (f(c)==0)
        break;
    elseif (f(a)*f(c)< 0)
        b=c;
    else
        a=c;
    end
end
fprintf("Using the bisection search method, the local max value of f(x)= x^5 - 8*x^4 - 6*x^2 + 3 in [-10,10] with an e=10^-6 is the point: ("+c+","+F(c)+")")
```

Ερώτημα β)

Κώδικας matlab:

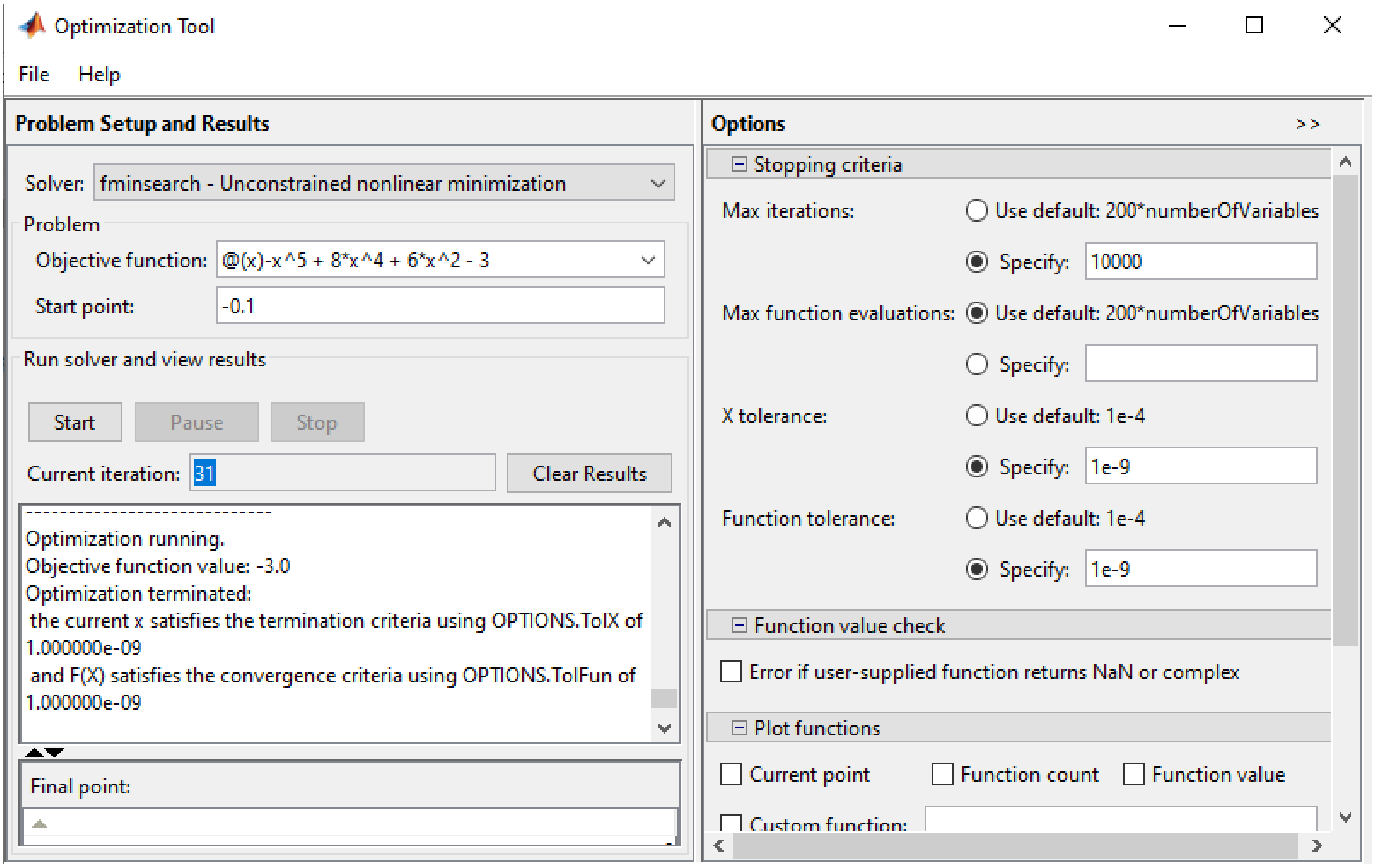
```
%% Newton - Raphson method
% AM=P18063 -->  1*x^5 - 8*x^4 + 0*x^3 - 6*x^2 + 3

% Equation: f(x) = x^5 - 8*x^4 - 6*x^2 + 3
% Derivative: F=f'(x) = 5*x^4-32*x^3-12*x

f= (5*x.^4)-(32*x.^3)-(12*x); % given function
g=diff(f); % derivative
e = 10^(-6);

while (true)
    x0 = input('Enter an intial approximation x0:');
    if (x0~=0)
        break;
    end
    fprintf('x0 must not be equal to 0');
    fprintf('\n');
end

for i=1:100 %100 iterations is enough
    f0=vpa(subs(f,x,x0)); %value of function at x0
    f0_der=vpa(subs(g,x,x0)); %value of function derivative at x0
    y=x0-f0/f0_der; % The Formula
    err=abs(y-x0);
    if err<e %checking the error at each iteration
        break;
    end
    x0=y;
end
y = y - rem(y,10^-6); %Displaying upto required decimal places
F =@(x) (1*x.^5)-(8*x.^4)-(6*x^2)+3;
fprintf("Using the Newton - Raphson method after %d iterations that, the local max
value of f(x)= x^5 - 8*x^4 - 6*x^2 + 3 with an e=10^-6 is the point: (%f,%f)",i,y,F(y))
```

Ερώτημα γ)

Για να μετατρέψω το πρόβλημα εύρεσης μεγίστου (μεγιστοποίησης) σε εύρεση ελαχίστου (ελαχιστοποίησης) καθώς χρησιμοποιώ την fminsearch του matlab (μέθοδος ελαχιστοποίησης, απλώς πολλαπλασίασα την f με το -1).

Παρακάτω είναι screenshot του optimization tool του matlab με starting point to 0,1 καθώς από τη γραφική παράσταση βλέπω ότι είναι αρκετά κοντά στο μέγιστο δηλαδή στο x=0.

Για μικρότερες τιμές από -16 και για μεγαλύτερες από 6 επιστρέφει μια τιμή που δείχνει ότι απειρίζεται και επιπλέον φτάνει τη max τιμή των iterations. Αυτό συμβαίνει γιατί λογικά εντοπίζει κάποια άνοδο και επιλέγει να μην πάει προς την άνοδο καθώς ψάχνει για ελάχιστα (κανονικά μέγιστο χωρίς τον πολλαπλασιασμό με το -1).



- Άσκηση 2

```
%% Max of f(x)

%f(x,y)=-6y-x^2+3xy-y^2
disp(" ")

syms x y
f=-6*y-x^2+3*x*y-y^2;

disp("My function according to my AM=18063: f="+string(f))
disp(" ")

eqn=jacobian(f);
disp("Derivatives found with jacobian() function: "+string(eqn(1))+" ,
"+string(eqn(2)))
disp(" ")

hess=hessian(f);
disp("Hessian matrix:")
disp(hess)
disp(" ")

[A,B] = equationsToMatrix([eqn(1), eqn(2)], [x, y]);
X = linsolve(A,B);
x=X(1);
y=X(2);

disp("Essential condition in order for a extremum to exist is the 2 derivatives of the
jacobian table to have a common x0 that: f'1(x0)=0 and f'2(x0)=0. These coordinates are
the following: ("+string(x)+","+string(y)+")")
disp("Checking if the above is a local extremum...")
disp("------------------------------Eigenvalues Method------------------------------")
disp("The eigenvalues of the hessian matrix are ")
```

```
disp(eig(hess))
disp("Because the 2 eigenvalues have different sign (-5,+ 1) the point under
investigation is a saddle point")
disp(" ")
disp("------------------------------Sylvester's Method------------------------------")
disp("Leading principal minors: ")
disp(det(hess(1,1)))
disp(det(hess))
disp("Principal minors: ")
disp("First rank")
disp(det(hess(1,1)))
disp(det(hess(1,2)))
disp(det(hess(2,1)))
disp(det(hess(2,2)))
disp("Second rank")
disp(det(hess))
disp("Leading principal minors are not all positive numbers neither they change sign
based on the following rule: (-1)^k*Dk>0 where k is the rank")
disp("Principal minors are not all non-negative neither they change sign based on the
following rule: (-1)^k*Dk>=0 where k is the rank")
disp("We can see that the none of the 4 conditions of Sylvester's criteria is met, so
we confirm that we are dealing with a saddle point")
```

- Άσκηση 3

```
%% Lagrange method

%Min f(x,y)=x^2 + 18y^2
%g(x,y) = 18x + 18y - 18 ----> g(x,y) = x + y - 1
%L(x,y,?)=f(x,y)+?*g(x,y)

syms x y l

f=x^2 + 18*y^2;
g= x + y - 1;

L=f+ l*g;

disp("My function according to my AM=18063: f="+string(f))
disp("My constraint according to my AM=18063: g="+string(g))
disp("My Lagrange function: L="+string(L))
disp(" ")

eqnx=diff(L,x);
eqny=diff(L,y);
eqnl=diff(L,l);
disp("Derivatives dL/dx, dL/dy, dL/dl: "+string(eqnx)+" , "+string(eqny)+" ,
"+string(eqnl))
disp(" ")

hess=hessian(L);
disp("Hessian matrix:")
disp(hess)
disp(" ")

disp("Leading principal minors to be examined: n-m=2-1=1 where n is the number of
variables(x,y) and m the number of constraints")
disp("The determinant is:"+string(det(hess))+"so it follows the rule (-1)^m, where m=1.
So the following point (x,y) is the local minimum")

[A,B] = equationsToMatrix([eqnx, eqny,eqnl], [x, y,l]);
X = linsolve(A,B);
```

```
x=X(1);
y=X(2);
l=X(3);
disp(x)
disp(y)
```