Universitatea Tehnica a Moldovei

Facultatea Calculatoare, Informatica si Microelectronica

Catedra Tehnologii Informationale

**RAPORT**

despre lucrarea de laborator nr. 2

la disciplina Metode si modele de calcul

Tema: Multimi convexe. Functii convexe. Rezolvarea problemelor fara restrictii.

A efectuat: st. gr. TI-173 Heghea Nicolae

A verificat: Ghetmancenco S.

**Cuprins**

[1. Notiuni teoretice 3](#_Toc526804949)

[2. Multimi convexe. Determinarea trosonului 4](#_Toc526804950)

[3. Functii convexe. Matricea Hessiana. 6](#_Toc526804951)

[3.1 Problema 9. (pag 11). Pe foaie. 6](#_Toc526804952)

[3.1 Problema 9. (pag 11). La calculator 7](#_Toc526804953)

[4. Metode de directii conjugate. Algoritmul lui Hestenes – Stiefel. 10](#_Toc526804954)

[5. Rezolvarea grafica a poblemei de programare liniara. 11](#_Toc526804955)

[6. Codul sursa 12](#_Toc526804956)

[6.1 Determinarea matricei Hessiane 12](#_Toc526804957)

[6.2 Metoda Grafica 14](#_Toc526804958)

[7. Concluzia 16](#_Toc526804959)

# Notiuni teoretice

O multime M, se numeste ***multine convexa*** daca, luand oricare 2 puncte ale sale , segmentul de dreapta ce le unste apartine multimii date.

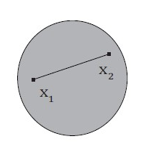
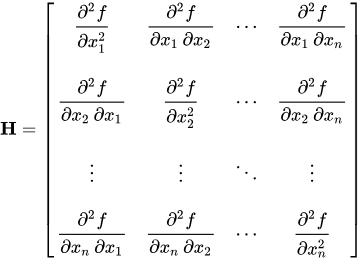


Fig. 1.1. Multime convexa.

***Trosonul*** reprezinta intersectia unui numar finit de semispatii, si este o multime convexa :

Matricea de dimensiune



se numeste ***matrice Hesse*** a functiei sau ***matricea Hessiana***. Se noteaza si .

# Multimi convexe. Determinarea trosonului

Problema 6. (pag. 6).

Conditia :

Sa se determine punctele extreme ale trosonului ***T***. definit de sistemul de inecuatii :

Rezolvare :

Observam ca trosonul este creat de 5 virfuri, care apar la intersectia liniilor:

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

Cu ajutorul programului facut in aplicatia MatLab. Pun aceste functii pe axele XOY.

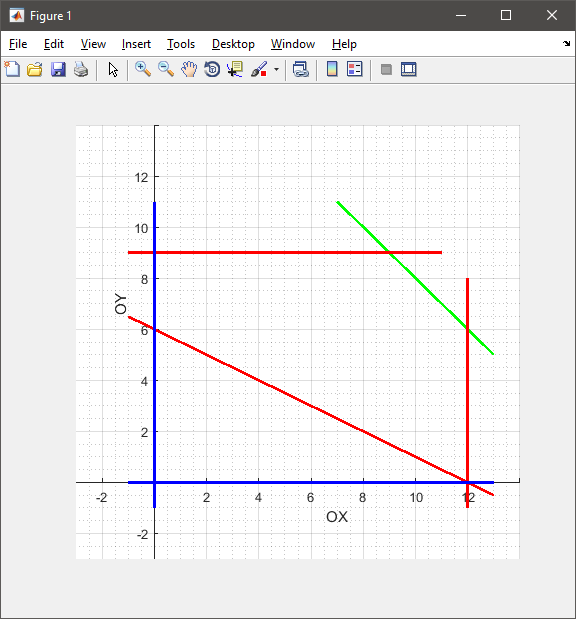


Fig. 2.1. Graficul functiilor.

# Functii convexe. Matricea Hessiana.

## Problema 9. (pag 11). Pe foaie.

P.9.1. :

Sa presupunem ca , avem :

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Rezulta ca functiile nu au Hessiana.

P.9.2. :

|  |  |  |
| --- | --- | --- |
|  |  |  |

Rezulta ca nu este continua.

P.9.3. :

|  |  |
| --- | --- |
|  |  |
|  |

In functia are punct de minim, si este functie convexa.

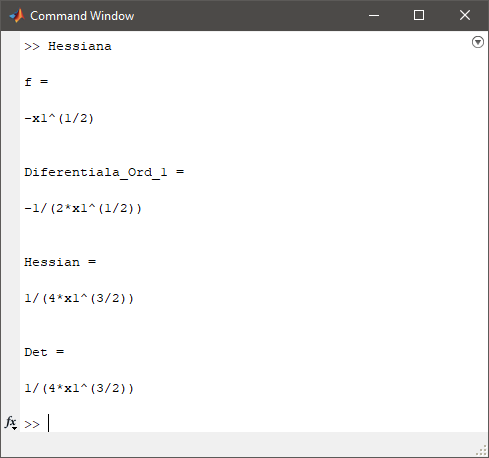
P.9.4. :

|  |  |  |
| --- | --- | --- |
|  | |  |
|  | |  |
|  |  |  |
|  |  |  |
|  | | |

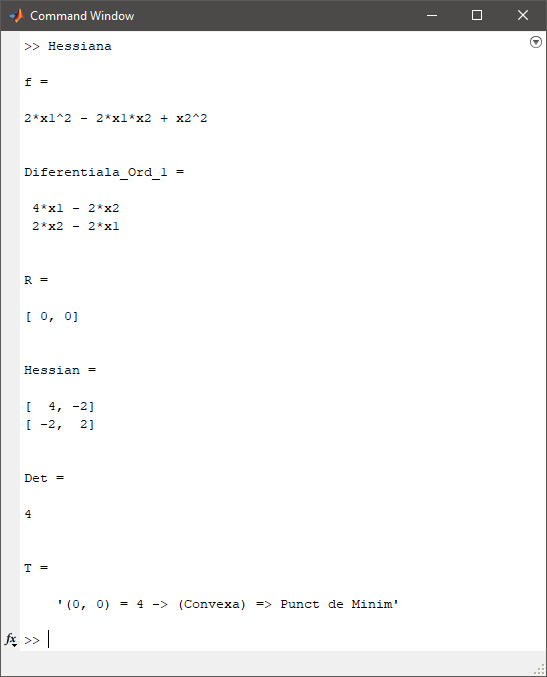
In functia are punct de minim, si este functie convexa.

## Problema 9. (pag 11). La calculator

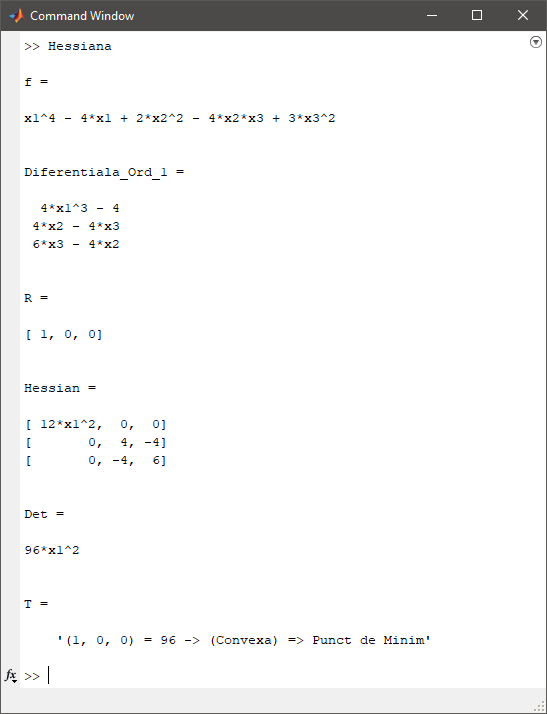
P.9.2. :



P.9.3. :



P.9.4. :



# Metode de directii conjugate. Algoritmul lui Hestenes – Stiefel.

Se da functia :

Punctele alese abitrar sunt .

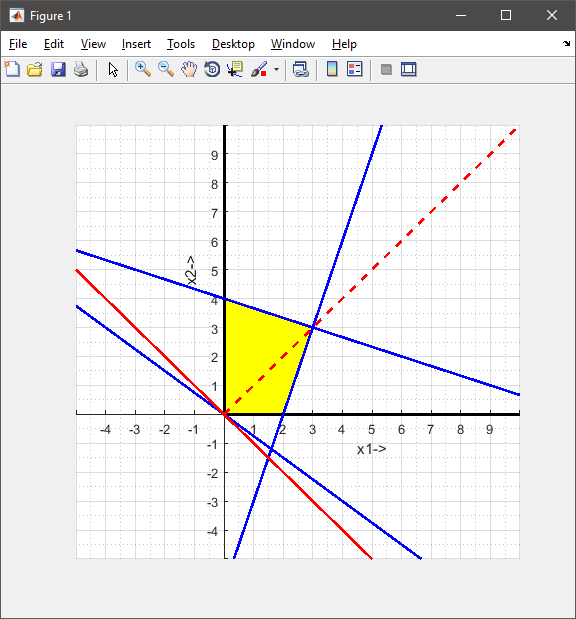
|  |  |  |  |
| --- | --- | --- | --- |
| Pas 1.1: | Pas 1.2: | Pas 1.3: |  |
| Pas 1.4: | |  |  |
| Pas 1.5:    = | | |  |
|  | | |  |

# Rezolvarea grafica a poblemei de programare liniara.

Problema 1. (pag 28)

P.1.

Aceasta problema am rezolvato in MatLab.



# Codul sursa

## Determinarea matricei Hessiane

|  |
| --- |
| function Hessiana  %#ok<\*NOPRT>%#ok<\*AGROW>%#ok<\*NBRAK>  %%%%%%%%%%-Exemple test-%%%%%%%%%%%%%%%%%%%%%%%%  %%%-Ex 3  % n = 2;  % x = sym('x',[1 n]);  % f = x(1)^3 -12\*x(1)\*x(2) + 8\*x(2)^3;    %%%-Ex 4  % n = 2;  % x = sym('x',[1 n]);  % f = x(1)^4 + x(1)^2\*x(2)^2 - 2\*x(1)^2;    %%%%%%%%%%-Pe Acasa-%%%%%%%%%%%%%%%%%%%%%%%%  %%%-Exemplul 9.2.pag.11.  n = 1;  x = sym('x',[1 n]);  f = -(x(1)^(1/2));    %%%-Exemplul 9.3.pag.11.  % n = 2;  % x = sym('x',[1 n]);  % f = 2\*x(1)^2 + x(2)^2 - 2\*x(1)\*x(2);    %%%-Exemplul 9.4.pag.11.  % n = 3;  % x = sym('x',[1 n]);  % f = x(1)^4 + 2\*x(2)^2 + 3\*x(3)^2 - 4\*x(1) - 4\*x(2)\*x(3);    Diferentiala\_Ord\_1 = diff(f, x(1));  for i = 2:n  Diferentiala\_Ord\_1 = [Diferentiala\_Ord\_1, diff(f, x(i))];  end  Diferentiala\_Ord\_1 = Diferentiala\_Ord\_1.';  Hessian = hessian(f,x);  Det = det(Hessian);    r = solve(Diferentiala\_Ord\_1, x, 'Real', true);  rs = size(r);  if rs > 0  r = struct2cell(r);  R = sym('q',[1 rs]);    R = cell2sym(r(1));  for i = 2:n  R = [R, cell2sym(r(i))];  end    nr = size(R);  nr = nr(1,1);  p = zeros( nr , 1 );    for i = 1:nr  s = Det;  for j = 1:n  s = subs(s, x(j), R(i,j));  end  p(i) = s;  end    T = repmat({''}, nr, 1);  for i = 1:nr  q = [ '(' ];  for j = 1:n  q = [q, char(R(i,j))];  if j < n  q = [q, ', '];  end  end  q = [q, ') = ', num2str(p(i)), ' -> ' ];  if p(i) <= 0  q = [q '(Concava) => Punct de Maxim' ];  else  q = [q '(Convexa) => Punct de Minim' ];  end  T(i) = {q};  end  end    %%%%-Afisarea-%%%%%%%%%%%%  f    Diferentiala\_Ord\_1    if rs > 0  R  end    Hessian    Det    if rs > 0  T  end  end |

## Metoda Grafica

|  |
| --- |
| function MetodaGrafica  [yx, xy, z] = exemplul\_1;    deseana(yx, xy, z);  end |

|  |
| --- |
| function [yx, xy, z] = exemplul\_1  setFigureProprietes;  syms x;    %cu x2  yx = [  (12 - x)/3  3\*x - 6  (-3\*x)/4  ];    %fara x2  xy = [  ];    z = [  (-x)  ];    x = [-5 10];  y = [-5 10];  axis([x y])    x = [0 0 3 2];  y = [0 4 3 0];  fill(x,y,'y');  end |

|  |
| --- |
| function deseana (yx, xy, z)  cond\_nenegativ    fx(yx);  fy(xy);  Z(z);  end |

|  |
| --- |
| function cond\_nenegativ  syms x;  int = [0 100];  oxy = 0\*x;    fplot(oxy, int, '-k', 'Linewidth', 2); hold on;  fplot(oxy, x, int, '-k', 'Linewidth', 2); hold on;  end |

|  |
| --- |
| function fx(y)  int = [-100 100];  n = size(y);  n = n(1,1);    for i = 1:n  fplot(y(i), int, '-b', 'Linewidth', 2); hold on;  end  end |

|  |
| --- |
| function fy(y)  int = [-100 100];  syms x;  n = size(y);  n = n(1,1);    for i = 1:n  fplot(y(i), x, int, '-b', 'Linewidth', 2); hold on;  end  end |

|  |
| --- |
| function Z(y)  syms x;  if size(y) > 0  fplot(y(1), [-100 100], '-r', 'Linewidth', 2); hold on;  fplot(-y(1), [0 100], '--r', 'Linewidth', 2); hold on;  end  end |

|  |
| --- |
| function setFigureProprietes  fig = figure(1);  set(fig,'units','points','position',[400,125,430,400]);    x = [-5 15];  y = [-5 15];    set(gca, 'xtick', x(1):1:x(2));  set(gca, 'ytick', y(1):1:y(2));  hold on    ax = gca;  ax.XAxisLocation = 'origin';  ax.YAxisLocation = 'origin';    xlabel('x1->');  ylabel('x2->');  grid on  grid minor  hold on  end |

# Concluzia

Aceste metode de rezolvare a problemelor, mai ajutat sa inteleg asa fel de probleme sub un al unchi. Acesti algoritmi si metode de rezolvare au un spectru larg de tipuri de probleme. Algoritmii si metodele la gasirea solutiei optime.