Rešene izpitne naloge

Optimizacijske Metode 2020/21

David Rubin

29. oktober 2020

Kazalo

1 Uvod

Dokument predstavlja zapisnik dela, ki se je izvajalo na vajah predmeta. Poleg tega dokumenta so dostopne tudi podmape, v katerih se nahajajo `m file`, ki vsebujejo nekatere definicije za izvedbo ukazov, ki pripeljejo do ustreznih rešitev.

2 Splošni napotki vaj

V kolikor uporabljaš simbolične funkcije, je potrebno spremenljivke inicializirati s `syms`. V kolikor bi si rad izrisal neko funkcijo, to storiš najlažje tako, da si definiraš simbolične spremenljivke s `syms`, definiraš neko funkcijo in kličeš izris s `ezmesh` (ali `fmesh`). Malo je treba tudi ugibati pri definiciji intervala, na katerem je izris, da dobiš dobro predstavitev. Glej kodo 1

Listing 1: Splošna uporaba Matlab na 1. vajah

```
% Inicializiraj spremenljivke za simbolicne funkcije
syms x y

% Definiraj neko funkcijo
f = x^4 - 16*x^2 - 5*x + y^4 - 16*y^2 - 5*y;

% Odvajaj parcialni odvod 1. stopnje po 'x'
diff(f, x, 1);

% jacobian -> v dveh stopnjah za pridobitev matrike
%jacobian(jacobian(f));

% Vstavi vrednost 3 v funkcijo in ostane samo spremenljivka 'y'
simpleF = subs(f,x,3);
% Vstavi se vrednost v 'y' in ostane nam neka vrednost
subs(simpleF, 4);

% roots izracuna nicle - tukaj imamo kompleksne nicle (f ne seka X osi)
roots([3 4 29 4 2 9])

% Izrisi graf na intervalu -5 5
fmesh(f, [-5, 5]);
```

Še nekaj koristnih napotkov pri uporabi funkcij:

- `fminsearch(@(`x`) function(`x`),` x_0`)` ... Uporabimo, ko iščemo minimume funkcij. `@(`x`)` nam pove, da je x neodvisna spremenljivka. `function(`x`)` je definicija neke funkcije (v večini primerov skozi ta dokument, bo ta funkcija shranjena v ločenem m-file). x_0 nam določa, kje bo postopek pričel, torej našo začetno točko. Priporočljivo je, da se uporabijo različne točke, saj se lahko nekatere *ujamejo* v lokalne minimume, mi pa si želimo poiskati globalnega. Dovolj dobro je, da vzamemo kar nekaj naključno generiranih točk, si shranimo rešitve za vsak klic in jih med sabo primerjamo. Če so vse vrednosti enake, potem to tipično pomeni, da je algoritem našel globalni minimum v vseh primerih, rešitev je potem katerakoli izmed pridobljenih. V kolikor niso vse rešitve enake, pa vzamemo najmanjšo vrednost, saj je tam globalni minimum. Za iskanje maksimumov lahko uporabimo isto funkcijo in ji spremenimo le parametre: `fminsearch(@(`x`) -function(`x`),` x_0`)` (dodali smo -). Paziti je potrebno, da ima tudi rezultat potem spremenjen (negativen) predznak.

- `fmincon(@(`x`) function(`x`),` x_0, A, b, A_{eq}, b_{eq}, `LB, UB, NONLCON)` ... pa je funkcija, kjer iščemo minimume ob podanih omejitvah. Enako kot prej sta defirnirani `function(`x`)` in x_0, dodajo pa se še naslednji členi. Matrika A predstavlja omejitve, pri katerih gre za relacijo $\leq$. V matriko A zapišemo števila na levi strani pred spremenljivkami, paziti pa je potrebno, da če katera spremenljivka ni označena v omejitvi jo v matriki predstavimo s členom 0. Vektor b predstavlja desno stran omejitev (torej vrednosti ki so na desni strani relacije *leq*). V kolikor je relacija $\geq$ je potrebno vse člene ustrezno pomnožiti z -1. Matrika A_{eq} in vektor b_{eq} predstavljata omejitve kjer je relacija $=$. `LB` in `UB` predstavljata spodnjo (*Lower Bound*) in zgornjo mejo (*Upper Bound*). `NONLCON` pa uporabimo kadar omejitve niso linearne (*non linear constraints)*. V tem primeru se tipično definirajo v ločenem *m-file*. Glej primer na nalogi 3.1.3 ali nalogi 3.5.2.

Poglejmo si primer iskanja minimuma za funkcijo `bana` definirano v ločenem *m-file*:

```
function y = bana(x)
    % Elemente v vektorjah dosegas s vektor(INDEX) ... index je seveda 1..N
    y = x(1)^4 - 16*x(1)^2 - 5*x(1) + x(2)^4 - 16*x(2)^2 - 5*x(2);
end
```

Uporabimo klic in določimo nek interval (pri tem lahko za enostavne funkcije pomaga izris):

```
% Funkcija vraca tocko x, vrednost minimuma fval, in zastavico flag (katero
   opazujemo da je 1 - pomeni da je najdena ustrezna resitev)
[x, fval, flag] = fminsearch(@(x)bana(x), [-3, -3])
% Vrne:
%       x =[-2.7468   -2.7468]
%       fval = -100.1178
%       flag = 1
% Imamo se en lokalni minimum, ce ponovimo iskanje v drugi tocki
[x, fval, flag] = fminsearch(@(x)bana(x), [-3, 3])
% Vrne:
%       x =[-2.7468   2.9035]
%       fval = -128.3912
%       flag = 1
```

3 Izpitne naloge

V nadaljevanju sledijo izpitne naloge in postopki za pridobitev rešitev.

3.1 Izpit 23. maj 2008

3.1.1 Naloga 1.)

Navodila:
Minimiziraj funkcijo:

$$f(x) = (x_1 - 0.5)^2 (x_1 + 1)^2 + (x_2 + 1)^2 (x_2 - 1) \tag{1}$$

Ali ima funkcija samo en lokalni minimum oziroma maksimum? Če jih ima morda več, jih poišči! Kolikšna je vrednost funkcije v točki minimuma. Pomagaš si lahko z risanjem funkcije.

Rešitev:
Funkcijo si lahko izrišemo s pomočjo ukazov

```
syms x y
f = (x - 0.5)^2 * (x + 1)^2 + (y + 1)^2 * (y - 1)^2;
fmesh(f, [-3, 3])
```

V novem `maj2008_1.m` si definiramo funkcijo:

```
function y = maj2008_1(x)
    y = (x(1) - 0.5)^2 * (x(1) + 1)^2 + (x(2) + 1)^2 * (x(2) - 1)^2;
end
```

Potem lahko napišemo skripto (oziroma preverimo v zanki) iskanje minimuma iz večih točk:

```
resitve = [];  % vnaprej pripravljen vektor resitev
for i=1:100  % 100 iteracij
        random_x0 = 20 * rand(1:2) - 10  % nakljucna zacetna tocka iz intervala [-10, 10]
        [x, fval, flag] = fminsearch(@(x)maj2008_1(x), random_x0)
        if flag == 1
                resitve = [resitve; [x fval]]  % shrani si dobre resitve
        end
end
b = uniquetol(resitve, 1e-6, 'byrows', true)  % izloci duplikatne resitve
% Vrne v stilu:
% b =
%    -1.00   -1.00   0
%    -1.00    1.00   0
%     0.50    1.00   0
%     0.50   -1.00   0
```

Te unikatne rešitve so potem vrednosti za naše minimume. Prva dva stolpca sta vrednosti x_1 in x_2, tretji stolpec pa vrednost funkcije v tej točki $f(x_1, x_2)$. Za iskanje maksimuma vidimo na grafu da je samo 1 in se nahaja v bližini točke $[0, 0]$, zato uporabimo le 1 klic:

```
[x, fval, flag] = fminsearch(@(x)-maj2008_1(x), [0, 0])
% Vrne v stilu:
% x = [-0.2500    0.0000]
% fval = -1.3164
% flag = 1
```

Kar pomeni da je točka $[-0.25, 0]$ lokalni maksimum z vrednostjo -1.3164.

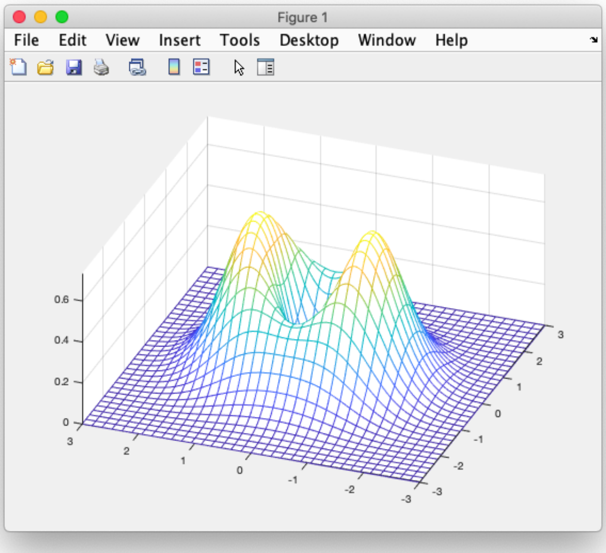
3.1.2 Naloga 2.)

Navodila:
Maksimiziraj funkcijo in izračunaj njeno vrednost v točki maksimuma:

$$f(x) = (x_1^2 + 2x_2^2)e^{-(x_1^2+x_2^2)} \tag{2}$$

Rešitev:
Podobno kot v nalogi 3.1.1 si najprej narišemo funkcijo, in potem poskušamo identificirati število in postopek iskanja maksimumov:

```
syms x y
f = (x^2 + 2*y^2) * exp( -(x^2 + y^2) );
fmesh(f, [-3, 3])
```



Slika 1: Izris funkcije 2 na intervalu $[-3, 3]$.

Pri izrisu (glej sliko 1) vidimo, da ima funkcija 2 vrhova in to v bližini točk $[0, -1]$ in $[0, 1]$. Funkcijo definiramo v ločenem *m-file* (`maj2008_2.m`):

```
function y = maj2008_2(x)
    y = (x(1)^2 + 2*x(2)^2) * exp( -(x(1)^2 +x(2)^2) );
end
```

... in uporabimo informacijo iz izrisa v naslednjem klicu:

```
[x, fval, flag] = fminsearch(@(x)-maj2008_2(x), [0, 1])
% Vrne v stilu:
% x =   [0     1]
% fval = -0.7358
% flag = 1
[x, fval, flag] = fminsearch(@(x)-maj2008_2(x), [0, -1])
% Vrne v stilu:
% x =   [0    -1]
```

```
% fval = -0.7358
% flag = 1
```

Točke, ki smo jih ocenili na grafu, so očitno maksimumi. Treba pa je paziti, saj je vrednost maksimumov v `fval` obratna - pravina vrednost obeh maksimumov je torej $f_{max} = 0.7358$.

3.1.3 Naloga 3.)

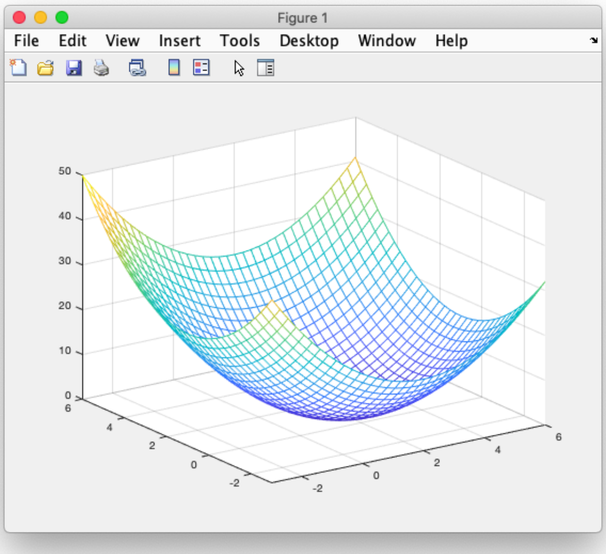
Navodila:
Določi minimum funkcije in njeno vrednost v točki minimuma:

$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \tag{3}$$

ob naslednjih omejitvah

$$\begin{aligned} -x_1 - x_2 + 2 &\geq 0 \\ -x_1^2 + x_2 &\geq 0 \end{aligned} \tag{4}$$

Rešitev:
Najprej si izrišimo funkcijo, da dobimo vpogled s čem imamo opravka (glej sliko 2). Vidimo, da imamo 1 globalni minimum nekje relativno blizu točke $[0, 0]$.



Slika 2: Izris funkcije 3 na intervalu $[-3, 6]$.

V ločenem *m-file* si sedaj definiramo funkcijo (`maj2008_3.m`):

```
% Definicija funckije ki jo minimiziramo
function y = maj2008_3(x)
    y = (x(1) - 2)^2 + (x(2)-1)^2;
end

%
```

... in omejitve (`maj2008_3_con.m`):

```
% Omejitve za funkcijo
function [c, ceq] = maj2008_3_con(x)
    c = [x(1)+x(2)-2; x(1)^2-x(2)]; % <- ze pomnozeno z -1
    ceq = [];
end
```

Omejitve so tipa $\geq$, zato smo jih pomnožili s -1 preden smo jih vstavili v spremenljvko `c`. Sedaj lahko poženemo iskanje minimuma v bližini $[0, 0]$:

```
[x, fval, flag] = fmincon(@(x)maj2008_3(x), [0, 0], [], [], [], [], [], [], @(x
    )maj2008_3_con(x))
% Vrne v stilu:
% x =  [1.000   -1.000]
% fval = 1.0000
% flag = 1
```

kar nam vrne rešitev $f_{min} = 1.000$ v točki $[1, -1]$.

3.1.4 Naloga 4.)

Navodila:
Poišči vse ničle enačbe:

$$f(x) = x^5 - 6x^4 - 92x^3 + 402x^2 + 91x - 396 \tag{5}$$

Rešitev:
Uporabimo funkcijo `roots` in ji podamo koeficiente za vsako stopnjo x:

```
roots([1 -6 -92 402 91 -396])
% Vrne v stilu:
% ans =
%     -9.0000
%     11.0000
%      4.0000
%     -1.0000
%      1.0000
```

Rešitev (ničle) so torej pri $x \in \{-9, -1, 1, 4, 11\}$

3.1.5 Naloga 5.)

Navodila:
Letalska družba kupuje gorivo za letala pri treh različnih prodajalcih. Družba potrebuje v naslednjem mesecu na vsakem od treh letališč, kjer pristaja, naslednje količine goriva: 100.000l na letališču 1, 180.000l na letališču 2 ter 350.000l na letališču 3. Gorivo prodajajo trije prodajalci, njihovo ceno goriva na posameznem letališču podaja naslednja tabela (cene so v centih na liter):

Vsak prodajalec pa ima na voljo omejene količine goriva, ki ga skupno lahko dostavi v posameznem mesecu. Te količine so 320.000l prvi prodajalec, 270.000l drugi ter 190.000l tretji prodajalec. Določi pravilo za nakup goriva letalske družbe, ki bo zadovoljilo njihovim potrebam na vsakem od letališč ter bo ekonomsko čimbolj ugodno.

	Letališče 1	Letališče 2	Letališče 3
Prodajalec 1	92	89	90
Prodajalec 2	91	91	95
Prodajalec 3	87	90	92

Rešitev:
Najprej si zastavimo spremenljivke x_i, kjer $i \in \{1,2,3,4,5,6,7,8,9\}$, pri čemer je x_1 količina goriva kupljenega na letališču 1 pri prodajalcu 1, x_2 je količina goriva kupljenega na letališču 2 pri prodajalcu 1, ... in x_9 je količina goriva kupljenega na letališču 3 pri prodajalcu 3. Iz tega sledijo omejitve:

$$\begin{aligned} x_1 + x_2 + x_3 &\geq 320.000 \\ x_4 + x_5 + x_6 &\geq 270.000 \\ x_7 + x_8 + x_9 &\geq 190.000 \\ x_1 + x_4 + x_7 &= 100.000 \\ x_2 + x_5 + x_8 &= 180.000 \\ x_3 + x_6 + x_9 &= 350.000 \\ x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 &\geq 0 \end{aligned} \tag{6}$$

... in funkcija, ki jo minimiziramo:

$$f(x) = 92x_1 + 89x_2 + 90x_3 + 91x_4 + 91x_5 + 95x_6 + 87x_7 + 90x_8 + 92x_9 \tag{7}$$

Vključili smo tudi omejitev nenegativnosti, saj smatramo, da letalska družba ne želi preprodajati goriva iz enega letališča na drugo (torej kupiti negativno količino). Problem lahko rešimo s funkcijo `intlinprog`:

```
% Koeficienti funkcije, ki jo minimiziramo
f = [92 89 90 91 91 95 87 90 92];

% Matrika in vektor, kjer je relacija <=
A = [1 1 1 0 0 0 0 0 0; 0 0 0 1 1 1 0 0 0; 0 0 0 0 0 0 1 1 1];
b = [320000 270000 190000];

% Matrika in vektor, kjer je relacija =
Aeq = [1 0 0 1 0 0 1 0 0; 0 1 0 0 1 0 0 1 0; 0 0 1 0 0 1 0 0 1];
beq = [100000 180000 350000];

% X so kolicine kupljenega goriva, fval pa strosek
% (zeros(1, 9) - LB, zelis nenegativne resitve)
[x, fval] = intlinprog(f, [], A, b, Aeq, beq, zeros(1, 9));
% Vrne v stilu:
% x =
%           0
%           0
%      320000
%           0
%      120000
%           0
%      100000
```

```
%          60000
%          30000
%
% fval = 56580000
```

Iz tega lahko razberemo, da je najugodneje na letališu 1 kupiti 100.000l pri prodajalcu 3, na letališču 2 120.000l pri prodajalcu 2 in 60.000l pri prodajalcu 3, na letališču 3 pa 320.000l pri prodajalcu 1 in 30.000l pri prodajalcu 3. Skupno bomo porabili 565.800,00 €.

3.2 Izpit 11. junij 2008

3.2.1 Naloga 1.)

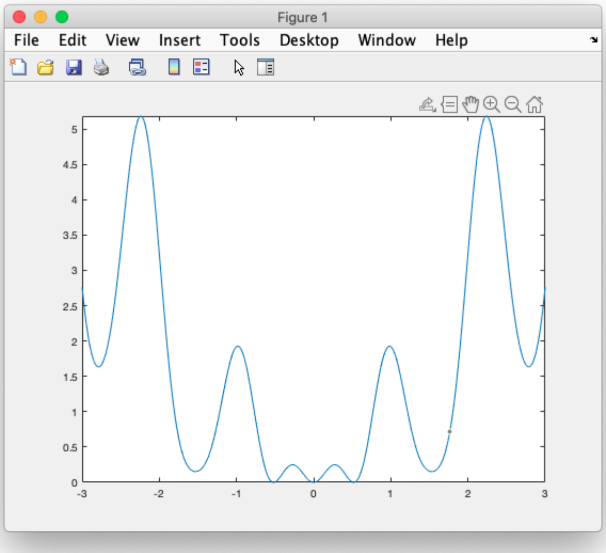
Navodila:

Poišči vse lokalne ekstreme funkcije na intervalu $[-3,3]$ in njene vrednosti v teh točkah:

$$f(x) = \frac{1}{2}(\sin(5x) - x)^2 \tag{8}$$

Rešitev:

Izrišimo funkcijo:



Slika 3: Izris funkcije 8 na intervalu $[-3,3]$.

Vidimo, da imamo 7 lokalnih minimumov in 6 lokalnih maksimumov. **TODO**

3.2.2 Naloga 2.)

Navodila:

Reši naslednji sistem enačb:

$$\begin{aligned} \sin(x+y) &= 0 \\ \cos(x-y) &= 0 \end{aligned} \tag{9}$$

Rešitev:

V novem *m-file* si definiramo enačbi (`junij2008_2.m`):

```
function y = junij2008_2(x)
    % Novo vrstico oznacimo s ';'
    y = [sin(x(1) + x(2)); cos(x(1) - x(2))];
end
```

... in nato kličemo funkcijo `fsolve`:

```
% Poskusimo najprej s [0, 0]
x = fsolve(@(x)junij2008_2(x), [0, 0])
% Kar vrne, da ni resitve ...
% Premaknimo se v negativni del
xL = fsolve(@(x)junij2008_2(x), [-.5, -.5])
% ... in v pozitivni
xR = fsolve(@(x)junij2008_2(x), [.5, .5])
% In dobimo najmanjsi resitvi (ostale so periodicne)
% xL =  [0.7854    -0.7854]
% xR =  [-0.7854    0.7854]
```

Rešitev so torej točke $x_1 = \frac{\pi}{4}$, $y_1 = -\frac{\pi}{4}$ in $x_2 = -\frac{\pi}{4}$, $y_2 = \frac{\pi}{4}$. Rešitve se periodično ponavljajo vsake $\frac{\pi}{2}$.

3.2.3 Naloga 3.)

Navodila:

Določi minimum funkcije in njeno vrednost v točki minimuma:

$$f(x) = x_1^2 + 5x_2^2 + 10x_3^2 - 4x_1x_2 + 6x_1x_3 - 12x_2x_3 - 2x_1 + 10x_2 - 5x_3 \tag{10}$$

ob naslednjih omejitvah:

$$\begin{aligned} x_1 + 2x_2 + x_3 &\geq 4 \\ x_1, x_2, x_3 &\geq 0 \end{aligned} \tag{11}$$

Rešitev:

V novem *m-file* si definiramo funkcijo (`junij2008_3.m`):

```
function y = junij2008_3(x)
    y = x(1)^2 + 5*x(2)^2 + 10*x(3)^2 - 4*x(1)*x(2) + 6*x(1)*x(3) - 12*x(2)*x
        (3) - 2*x(1) + 10*x(2) - 5*x(3);
end
```

In kličemo funkcijo `fmincon`:

```
[x, fval] = fmincon(@(x)junij2008_3(x), [1 1 1], [-1 -2 -1], -4, [], [], [0 0
    0])
% Vrne v stilu:
% x = [2.9412    0.5294    0.0000]
```

```
% fval = 3.2353
```

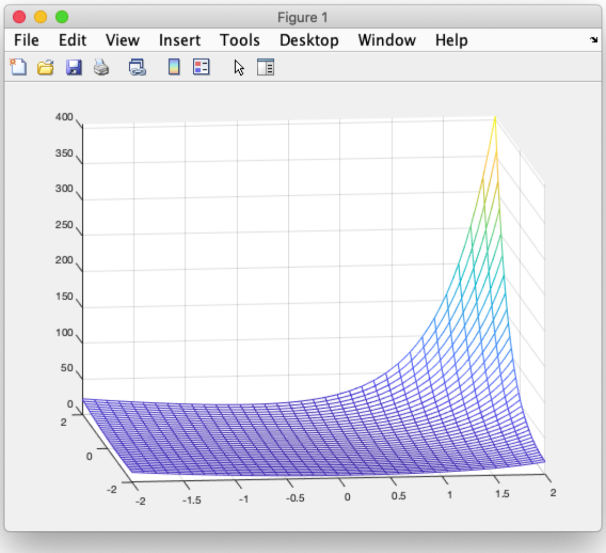
Začnemo v naključni točki $[1, 1, 1]$, omejitve (matrika A in vektor b) morajo upoštevati relacijo $\leq$, zato so vrednosti obratne, A_{eq} in b_{eq} pustimo prazne, nastavimo pa še spodnjo mejo z $[0, 0, 0]$. Rešitev je torej točka $x_{min} = [2.9412, 0.5294, 0.0000]$, v kateri imamo vrednost $f(x_{min}) = 3.2353$

3.2.4 Naloga 4.)

Navodila:
Poišči minimum funkcije in njeno funkcijsko vrednost v točki minimuma:

$$f(x) = 2x_1^2 - x_1 x_2 + x_2^2 - 3x_1 + e^{2x_1 + x_2} \tag{12}$$

Rešitev:
Ker imamo le 2 spremenljivki, si izrišimo funkcijo (glej sliko 4).



Slika 4: Izris funkcije 12 na intervalu $[-2, 2]$.

Vidimo, da je nekje v bližini točke $[0, 0]$ morda minimum. Deifinirajmo nov *m-file* (`junij2008_4.m`):

```
function y = junij2008_4(x)
    y = 2 * x(1)^2 - x(1) * x(2) + x(2)^2 - 3*x(1) + exp(2*x(1) + x(2));
end
```

... in kličemo funkcijo `fminsearch`:

```
[x, fval, flag] = fminsearch(@(x)junij2008_4(x), [0, 0])
% Vrne v stilu:
% x = [0.1737   -0.3915]
% fval = 0.7174
% flag = 1
```

... ki nam poda našo rešitev: $x_{min} = [0.1737, -0.3915]$, $f(x_{min}) = 0.7174$.

3.2.5 Naloga 5.)

Navodilo:

Trgovina z malimi živalmi je ugotovila, da potrebuje hrček najmanj 70 enot beljakovin, 100 enot ogljikovih hidratov ter 20 enot maščob dnevno. Trgovina ima na zalogi 6 različnih vrst hrane za hrčke z naslednjimi lastnostmi: Kakšno razmerje posameznih vrst hrane bo mešanica hrane

Hrana	Beljakovin/dozo	Ogljikovih hidratov/dozo	Maščob / dozo	Cena / dozo
A	20	50	4	2
B	30	30	9	3
C	40	20	11	5
D	40	25	10	6
E	45	50	9	8
F	30	20	10	8

za hrčka, ki bo zadovoljila njegove dnevne potrebe in bo cenovno najbolj ugodna za trgovino? Napiši sistem enačb in ga reši z MATLAB-om!

textbfRešitev:

Najprej določimo spremenljivke: x_N, kar pomeni koliko doz hrane N bomo kupili (N predstavlja vrsto hrane, torej A, B, C ... F). Funkcija ki jo minimiziramo je cena:

$$f(x) = 2x_1 + 3x_2 + 5x_3 + 6x_4 + 8x_5 + 8x_6 \tag{13}$$

omejitve pa so sledeče:

$$\begin{aligned} 20x_1 + 30x_2 + 40x_3 + 40x_4 + 45x_5 + 30x_6 \geq 70 \\ 50x_1 + 30x_2 + 20x_3 + 25x_4 + 50x_5 + 20x_6 \geq 100 \\ 4x_1 + 9x_2 + 11x_3 + 10x_4 + 9x_5 + 10x_6 \geq 20 \\ x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{aligned} \tag{14}$$

Enačno lahko rešimo s pomočjo funkcije `intlinprog`:

```
% Koeficienti funkcije
f = [ 2 3 5 6 8 8];
% Matrika A, obrnemo predznak
A = [-20 -30 -40 -40 -45 -30; -50 -30 -20 -25 -50 -20; -4 -9 -11 -10 -9 -10];
% Vektor b (desna stran omejitev) tudi obrnemo predznak
b = [-70 -100 -20];
% Aeq in beq sta prazna, prav tako ne potrebujemo celostevilskih resitev,
   nastavimo pa se spodnjo mejo
[x, fval] = intlinprog(f, [], A, b , [], [], zeros(1, 6))
% Vrne v stilu:
% x = 0.9091
%     1.8182
%          0
%          0
%          0
%          0
```

```
% fval = 7.2727
```

Ugotovimo, da je cenovno najbolj ugodno kupiti 0.9091 doz hrane A in 1.8182 doz hrane B za skupno ceno 7.2727.

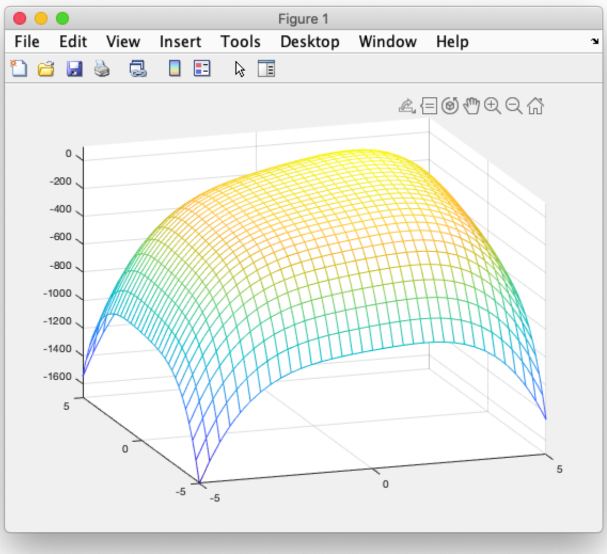
3.3 Izpit 3. september 2008

3.3.1 Naloga 1.)

Navodila:
Poišči vse lokalne minimume in maksimume funkcije na intervalu ter njene vrednosti v teh točkah. Pomagaš si lahko z risanjem funkcije. Ali obstaja globalni maksimum?

$$f(x) = 3x_1x_2 + 40x_1 + 30x_2 - 4x_1^2 - x_1^4 - 3x_2^2 - x_2^4 \tag{15}$$

Rešitev:
Najprej si narišimo funkcijo (slika 5).



Slika 5: Izris funkcije 15 na intervalu $[-5, 5]$.

Iz grafa je razvidno, da imamo opravka z 1 globalnim maksimumom, ki ga poiščemo s `fminsearch`. Deifiniramo si funkcijo v ločenem *m-file* (`september2008_1.m`):

```
function y = september2008_1(x)
    y = 3*x(1)*x(2) + 40*x(1) + 30*x(2) - 4*x(1)^2 - x(1)^4 - 3*x(2)^2 - x(2)
        ^4;
end
```

... in pokličemo funkcijo v okolici $[-3, -3]$:

```
[x, fval] = fminsearch(@(x)-september2008_1(x), [-3, -3])
% Vrne v stilu:
% x = [1.9548     1.8380]
```

```
% fval = -92.6766
```

Ne smemo pozabiti, ker smo obrnili predznak za maksimizacijo, je obrnjen tudi predznak vrednosti. Pravilna rešitev je torej pri $x_{max} = [1.9548, 1.8380]$ z vrednostjo $f(x_{max}) = 92.6766$.

3.3.2 Naloga 2.)

Navodila:
Maksimiziraj funkcijo in izračunaj njeno vrednost v točki maksimuma:

$$f(x) = -(x_1 - x_2)^2 - (x_3 - 1)^2 - 1 - 0.02(x_1^5 + x_2^5 + x_3^5 - 16)^2 \quad (16)$$

Rešitev:
Funkcija ima preveč dimenzij za risanje, zato kar poskusimo z nekaj sreče. Definiramo si nov *m-file* (`september2008_2.m`:

```
function y = september2008_2(x)
    y = -(x(1) - x(2))^2 - (x(3) - 1)^2 - 1 - 0.02*(x(1)^5 + x(2)^5 + x(3)^5 -
        16)^2;
end
```

... in kličemo funkcijo `fminsearch` (z obratno vrednostjo ker maksimiziramo):

```
% Seznam vseh resitev
s = [];
for i=1:100
    randX0 = 20 * rand(1, 3) - 10; % nakljucna zacetna tocka
    [x, fval, flag] = fminsearch(@(x)-september2008_2(x), randX0);
    if flag == 1
        % Dodaj pravilno resitev v seznam vseh
        s = [s; [x fval]];
    end
end
ss = sortrows(s, 4); % Sortiraj resitve po velikosti
ss(1:10,:) % izpisi prvih 10 vrstic
% Vrne 10 identicnih vrstic
% ans =
%     1.4963    1.4963    1.0000    1.0000
```

Ker smo v tem primeru iskali maksimum funkcije, lahko rezultat sortiramo kar naraščujoče, saj je potrebno spremeniti predznak in je najmanjša vrednost enaka največi. Vidimo, da je prvih 10 najdenih točk identičnih, torej lahko vzamemo maksimum $x_{max} = [1.4963, 1.4963, 1.0000]$ in vrednost $f_{max} = -1$ (ne pozabimo na $-$).

3.3.3 Naloga 3.)

Navodila:
Določi minimum funkcije in njeno vrednost v točki minimuma:

$$f(x) = 2x_1 + x_2^3 + x_3^2 \quad (17)$$

ob naslednjih omejitvah:

$$x_1^2 + 2x_2^2 + x_3^2 \geq 4x_1, x_2, x_3 \geq 0 \quad (18)$$

Rešitev:

V ločenih *m-file* si definiramo funkcijo (`september2008_3.m`):

```
function y = september2008_3(x)
    y = 2*x(1) + x(2)^3 + x(3)^2;
end
```

... in omejitve (`september2008_3_con.m`):

```
function [c, ceq] = september2008_3_con(x)
    % Cleni imajo spremenjen predznak, prestavilo pa smo se 4 iz leve na desno
        stran (ostane pozitivna)
    c = [-x(1)^2 -2*x(2)^2 - x(3)^2 + 4];
    ceq = [];
end
```

... in kličemo funkcijo `fmincon`:

```
[x, fval, flag] = fmincon(@(x)september2008_3(x), [1 1 1], [], [], [], [],
    zeros(1, 3), [], @(x)september2008_3_con(x))
% Vrne v stilu:
%  x = [0.0000    1.3333    0.6667]
%  fval = 2.8148
%  flag = 1
```

Matrike A, b, A_{eq}, B_{eq} smo pustili prazne, dodali smo spodnjo mejo ničel, zgornjo mejo nedefinirano in podali omejitve v ločenem *m-file*. Pridobljena rešitev je očitno veljavna (`flag = 1`): $x_{min} = [0.0000, 1.3333, 0.6667]$, vrednost pa je $f(x_{min}) = 2.8148$.

3.4 Izpit 19. Januar 2012

3.4.1 Naloga 3.)

Narisemo si skico. (Slika na namizju) Imamo dva podobna trikotnika, narisemo si kote in uporabimo kotne funkcije da racunamo l1 + l2 = l. Funkcijo racunamo kot minimum ??? Ker ce racunamo maksimum dobimo resitev neskoncno. Resitev naloge je f = 7.621m, $\alpha = 0.7482RAD$

3.4.2 Naloga 4.)

Startamo v 1 in hocemo prit v 9. Povezave so usmerjene, lahko gres le v doloceno smer (ni negativnih povezav). Nastavit je treba neke enacbe. Lahko imamo spremenljivke vozlisca, druga moznost pa je da so spremenljivke povezave. Tu je logicno da je spremenljivka povezave. Iz vsakega vozlisca se lahko odlocimo da gremo po eni poti ali ne. Torej 1 -¿ 2 povezava (recimo ji x(1)) pove ali smo sli po poti (=1) ali ne (=0).

Nastavit moramo nek sistem enacb. Spremenljivke bomo oznacevali z $x_{vhod,izhod}$. Ker v x1 zacnemo je prva omejitev: $x_12 + x_13 + x_14 = 1$ // ¡- tocno po eni izmed poti mormo Podnobna enacba dobimo za zadnje vozlisce $x_79 + x_69 + x_89 = 1$ // ¡- tocno po eni povezavi moramo prit v sink

Podobne enacbe moramo spisat za vas vmesna vozlisca. Recimo za vozlisce st. 3: vsota prihodnih mora bit enaka vsoti izhodnih (ce smo prisli v neko vozlisce potem moramo iz njega tudi oditi) $x_25 + x_35 = x_56 + x_57 + x_{58}$

treba je prestet vse puscice pa prestevilcit spremenljivke -¿ x1, x2, x3, x4 ... x16 A, b sta prazni -¿ vse omejitve so tipa je enako Aeq ima 16 stolpcev in 9 vrstic -¿ pri 7 so desne strani 0, pri source pa sink pa je desna stran 1

3.5 Izpit 10. februar 2014

Izpit je profesor pokazal na vajah.

3.5.1 Naloga 1.)

Navodila:
Poišči vsaj dvajset različnih globalnih minimumov funkcije in njihovo vrednost (vse so enake) v teh točkah.

$$f(x) = \left(\sum_{i=1}^{5} i\cos((i+1)x_1 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_2 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_3 + i)\right)\left(\sum_{i=1}^{5} i\cos((i+1)x_4 + i)\right) \tag{19}$$

in omejitvijo spremenljivk $-10 \leq x_i \leq 10$.

POMOČ: Ker bo ročno iskanje minimumov z različnimi starnimi točkami preveč zamudno, si lahko pomagaš s kratkim programčkom za iskanje najoljših rešitev! Po elektronski pošti oddaj tudi vse izdelane programe in dobljene rešitve!

Rešitev:
Definiramo si funkcijo v novem *m-file* (`februar2014_1_script.m`):

```
function y = februar2014_1(x)
    sum1 = 0;
    sum2 = 0;
    sum3 = 0;
    sum4 = 0;
    for i=1:5
        sum1 = sum1 + i * cos((i+1) * x(1) + i);
        sum2 = sum2 + i * cos((i+1) * x(2) + i);
        sum2 = sum2 + i * cos((i+1) * x(3) + i);
        sum2 = sum2 + i * cos((i+1) * x(4) + i);
    end
    % Vrni zmnozek vsot
    y = sum1 * sum2 * sum3 * sum4;
end
```

... in pomožno skripto, ki požene iskanje nad naključno generiranim vhodom. Meje naključnega generiranja postavimo v omejitev, ki je omenjena v navodilih.

```
resitve = []; % Seznam najdenih resitev
for i=1:100
    % Pozene iskanje s nakljucnim vektorjem velikosti 1x4 (uposteva omejitev -10 <= x(i) <= 10)
    [x, fval, flag] = fminsearch(@(x)februar2014_1(x), 20*rand(1,4)-10);
    if flag == 1 % Vkljuci samo dobre resitve
        resitve = [resitve; [x fval]];
    end
end
% resitve =
```

```
%     -4.5601    -5.3684     7.9907     8.1739          0
%      2.0729    -2.6953     1.9718     3.3697          0
%      7.8913    -8.2533     0.7802    -1.4310          0
%      2.3431     1.1775    -5.4830    -7.9096          0
```

Po končanem klicu skripte se nam v spremenljivko `resitve` shranijo vse veljavne vrednosti, v našem primeru je to 100 vrstic (število ponovitev zanke).

3.5.2 Naloga 2.)

Navodila:
Poišči maksimum[1] podane funkcije in njeno vrednost v tej točki ob navedenih omejitvah za $n = 2, 3, 4, 5$.

$$f(x) = \left(\sqrt{n}\right)^n \prod_{i=1}^{n} x_i \tag{20}$$

$$\begin{gathered} \sum_{i=1}^{n} x_i^2 = 1 \\ 0 \leq x_i \leq 1 \quad i = 1, ..., n \end{gathered} \tag{21}$$

Rešitev:
Kreiramo si nov *m-file* z definicijo funkcije (`februar2014_2.m`):

```
function y = februar2014_2(x)
    [~, n] = size(x); % Uporabi velikost vektorja za drug parameter
    y = sqrt(n)^n * prod(x); % Naredi produkt vseh elementov tega vektorja
end
```

... datoteko z omejitvami (`februar2014_2_con.m`):

```
function [c, ceq] = februar2014_2_con(x)
    % . pomeni da delamo operacijo nad vsakim posameznih elementon (ne
       kvadriramo vektorsko ampak po elementah)
    c = [];
    ceq = sum(x.^2)-1;
end
```

... in vse skupaj poženemo v nekem skriptu

```
resitve = [];
n = 3; % Vrednost n iz navodil
for i=1:20 % Veckrat poisci resitev
    [x, fval, flag] = fmincon(@(x)-februar2014_2(x), rand(1,n), [], [], [], [],
        zeros(1,n), ones(1,n), @(x)februar2014_2_con(x));
    if flag == 1
       % Pomnozi resitev z obratno vrednost (maksimum) in koren od n na n
       resitve = [resitve; [x -fval]];
    end
end
% (vse vrstice enake):
% resitve = 0.5774    0.5774    0.5774    1.0000
```

V skriptu upoštevamo omejitev z naključnim generiranjem med 0 in 1, prvo pa smo eksplicitno zapisali v omejitvah (`ceq`). V skriptu spreminjamo n vrednost in dobimo rezultate zanje.

[1]V originalnem izpitu je bilo navodilo iskanje minimuma, na vajah je profesor omenil, da je ta problem trivialen in je popravil navodila, da se išče maksimum

3.5.3 Naloga 3.)

Navodila:

Iz lesene krogle s polmerom 10cm želimo s pomočjo obrezovanja izdelati prisekani stožec, ki bo imel čim večjo možno prostornino (volumen). Koliko bosta znašala polmera osnovnih ploskev takega prisekanega stožca in koliko njegova višina, da bo volumen dobljenega prisekanega stožca maksimalen? Poskusi rešiti nalogo še za minimalni volumen prisekanega stožca, če želimo, da je vsota obeh polmerov osnovnih ploskev v prisekanem stožcu večja kot 12cm.

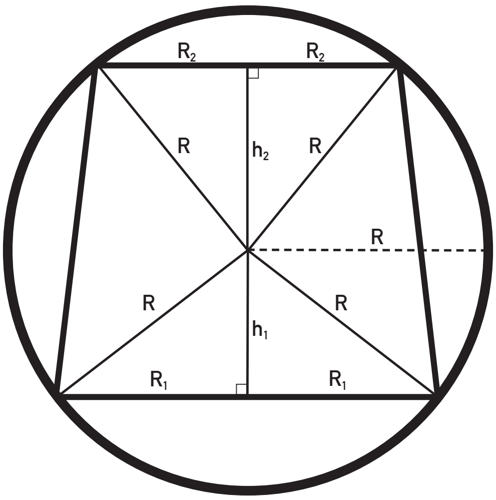
POMOČ: Volumen prisekanega stožca se izračuna po enačbi:

$$V = \frac{\pi h}{3}\left(R_1^2 + R_2^2 + R_1 R_2\right),$$

kjer sta R_1 in R_2 polmera osnovnik ploskev prisekanega stožca, h pa njegova višina.

Rešitev:

Za takšne geometrijske naloge si je najbolje narisati skico (glej sliko 6). Iz skice lahko opazimo,



Slika 6: Skica z oznakami stranic za pogled iz strani na prisekani stožec vstavljen v kroglo.

da lahko stožec razstavimo na pravokotne trikotnike ($\triangle R_1 R h_1$ in $\triangle R_2 R h_2$). Ker je polmer krogle (R) podan, bodo naše spremenljivke očitno (R_1, R_2, h_1 in h_2). Omejitve lahko zapišemo s pomočjo Pitagorovega izreka:

$$\begin{aligned} R_1^2 + h_1^2 &= R^2 \\ R_2^2 + h_2^2 &= R^2 \qquad (22) \\ R_1 + R_2 &> 12\text{cm} \; * \end{aligned}$$

To zadnjo omejitev uporabimo le v drugem delu naloge (iskanje najmanjšega stožca, kjer še to drži) in jo v prvem delu izpustimo. Uporabili bomo funkcijo `fmincon`, zato si definiramo nov *m-file* za definicijo funkcije (`februar2014_3.m`):

```
function y = februar2014_3(x)
    % Enacba za volumen, kjer je:
    % x1 ... h1
    % x2 ... h2
    % x3 ... R1
    % x4 ... R2
    y = pi * (x(1) + x(2)) / 3 * (x(3)^2 + x(4)^2 + x(3)*x(4));
end
```

... in ločenega za definicijo omejitev (`februar2014_3_con.m`):

```
function [c, ceq] = februar2014_3_con(x)
    c = [12 - x(3) - x(4)]; % omejitev tipa manjse ali enako imamo v drugem delu naloge (za prvi del pustimo prazno)
    ceq = [x(3)^2 + x(1)^2 - 100; x(4)^2 + x(2)^2 - 100]; % omejitev tipa enako iz prvega dela (maks. stozec)
end
```

Paziti je treba, da prvotne omejitve 22 v Matlabu pretvorimo v ustrezno obliko (vse morajo biti tipa $\leq$ in na desni strani želimo 0). Kar sledi je še iskanje maksimuma (prvi del naloge):

```
[x, fval, flag] = fmincon(@(x)-februar2014\_3(x), [8 7 9 8], [], [], [], [], zeros(1,4), [10 10 10 10], @(x)februar2014\_3\_con(x))
% Vrne:
% x = 5.7735    5.7735    8.1650    8.1650
% fval = -2.4184e+03
% flag = 1
```

Ne smemo pozabiti na minus pri definiciji funkcije, začetno točko izberemo naključno, A, b, A_{eq} in b_{eq} pustimo prazne, za spodnjo omejitev postavimo 0, za zgornjo lahko dodamo desetice (stožec ne sme presegati krogle), vstavimo pa še omejitve c in c_{eq}. Rešitev je potemtakem $h_1 = h_2 = 5.7735$ in $R_1 = R_2 = 8.1650$, kar pomeni da največji prisekan stožec je očitno valj.

Da rešimo drugi del naloge (najmanjši volumen ob omejitvi skupnega premera ploskev stožca), pustimo omejitev 3 v c in zopet poženemo `fmincon`:

```
[x, fval, flag] = fmincon(@(x)februar2014_3(x), [5 5 9 1], [], [], [], [], zeros(1,4), [10 10 10 10], @(x)februar2014_3_con(x))
% Vrne:
% x = 0.0000    9.7980   10.0000    2.0000
% fval = 1.2723e+03
% flag = 1
```

Klic je enak kot prej, razlika je da tokrat izpustimo minus in malce preuredimo začetno točko (da ne presežemo Matlabovih omejitev pri iskanju). Dobimo še drugo rešitev: $h_1 = 0$, $h_2 = 9.798$, $R_1 = 10$ in $R_2 = 2$.

3.5.4 Naloga 4.)

gle jslika na namizju. Izdelki so nedeljivi -¿ pomeni da nastavis integer solutions. Modificiran knapsack problem (izdelki se lahko ponavljajo). Napisati si je treba enacbe (4 spremenljivke, koliko izdelkov P1 = x1, ...). Cena $13x_1 + 27x_2 + 41x_3 + 55x_4$ - maksimiziras Omejitve .... 21 ton + celan stevila (intlinprog, inlincon -¿ [1 2 3 4] ... vse morajo bit celostevilcne)

f= [13 27 41 55] A = [1 2 3 4] b = 21

[x, fval, flag] = intlinprog(-f, 1:4, A,b, [], [], zeros(1,4)) x =

1.0000 0 0 5.0000

fval =

-288

flag =

1