

Rešene izpitne naloge

Optimizacijske Metode 2020/21

David Rubin

26. oktober 2020

Kazalo

1	Uvod	2
2	Splošni napotki vaj	2
3	Izpitne naloge	3
3.1	Izpit 23. maj 2008	4
3.1.1	Naloga 1.)	4
3.1.2	Naloga 2.)	5
3.1.3	Naloga 3.)	6
3.1.4	Naloga 4.)	7
3.1.5	Naloga 5.)	7
3.2	Izpit 11. junij 2008	9
3.2.1	Naloga 1.)	9
3.2.2	Naloga 2.)	10
3.2.3	Naloga 3.)	10
3.2.4	Naloga 4.)	11
3.2.5	Naloga 5.)	12
3.3	Izpit 3. september 2008	13
3.3.1	Naloga 1.)	13
3.3.2	Naloga 2.)	14
3.3.3	Naloga 3.)	14
3.4	Izpit 19. Januar 2012	15
3.4.1	Naloga 3.)	15
3.4.2	Naloga 4.)	15
3.5	Izpit 10. februar 2014	16
3.5.1	Naloga 1.)	16
3.5.2	Naloga 2.)	17
3.5.3	Naloga 3.)	18
3.5.4	Naloga 4.)	18

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

```
1 % Inicializiraj spremenljivke za symbolicne funkcije
2 syms x y
3
4 % Definiraj neko funkcijo
5 f = x^4 - 16*x^2 - 5*x + y^4 - 16*y^2 - 5*y;
6
7 % Odvajaj parcialni odvod 1. stopnje po 'x'
8 diff(f, x, 1);
9
10 % jacobian -> v dveh stopnjah za pridobitev matrike
11 %jacobian(jacobian(f));
12
13 % Vstavi vrednost 3 v funkcijo in ostane samo spremenljivka 'y'
14 simpleF = subs(f,x,3);
15 % Vstavi se vrednost v 'y' in ostane nam neka vrednost
16 subs(simpleF, 4);
17
18 % roots izracuna nicle - tukaj imamo kompleksne nicle (f ne seka X osi)
19 roots([3 4 29 4 2 9])
20
21 % Izrisi graf na intervalu -5 5
22 fmesh(f, [-5, 5]);
```

Še nekaj koristnih napotkov pri uporabi funkcij:

- `fminsearch(@(x) function(x), x0)` ... 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), x0)` (dodali smo -). Paziti je potrebno, da ima tudi rezultat potem spremenjen (negativen) predznak.

- `fmincon(@(x) function(x), x0, A, b, Aeq, beq, LB, UB, NONLCON)` ... pa je funkcija, kjer išemo minimume ob podanih omejitvah. Enako kot prej sta definirani `function(x)` in `x0`, 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*:

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

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

```
1 % Funkcija vraca točko x, vrednost minimuma fval, in zastavico flag (katero
   % opazujemo da je 1 - pomeni da je najdena ustrezna resitev)
2 [x, fval, flag] = fminsearch(@(x)bana(x), [-3, -3])
3 % Vrne:
4 %     x = [-2.7468    -2.7468]
5 %     fval = -100.1178
6 %     flag = 1
7 % Imamo se en lokalni minimum, ce ponovimo iskanje v drugi točki
8 [x, fval, flag] = fminsearch(@(x)bana(x), [-3, 3])
9 % Vrne:
10 %     x = [-2.7468    2.9035]
11 %     fval = -128.3912
12 %     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:

$$f(x) = (x_1 - 0.5)^2(x_1 + 1)^2 + (x_2 + 1)^2(x_2 - 1) \quad (1)$$

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

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

V novem maj2008_1.m si definiramo funkcijo:

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

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

```
1 resitve = []; % vnaprej pripravljen vektor resitev
2 for i=1:100 % 100 iteracij
3     random_x0 = 20 * rand(1,2) - 10 % naključna začetna točka iz intervala
4         [-10, 10]
5     [x, fval, flag] = fminsearch(@(x)maj2008_1(x), random_x0)
6     if flag == 1
7         resitve = [resitve; [x fval]] % shrani si dobre resitve
8     end
9 end
10 b = uniquetol(resitve, 1e-6, 'byrows', true) % izloči duplikatne resitve
11 % Vrne v stilu:
12 % b =
13 %     -1.00    -1.00     0
14 %     -1.00     1.00     0
15 %      0.50     1.00     0
16 %      0.50    -1.00     0
```

Te unikatne rešitve so potem vrednosti za naše minime. 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:

```
1 [x, fval, flag] = fminsearch(@(x)-maj2008_1(x), [0, 0])
2 % Vrne v stilu:
3 % x = [-0.2500    0.0000]
4 % fval = -1.3164
5 % flag = 1
```

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

3.1.2 Naloga 2.)

Navodila:

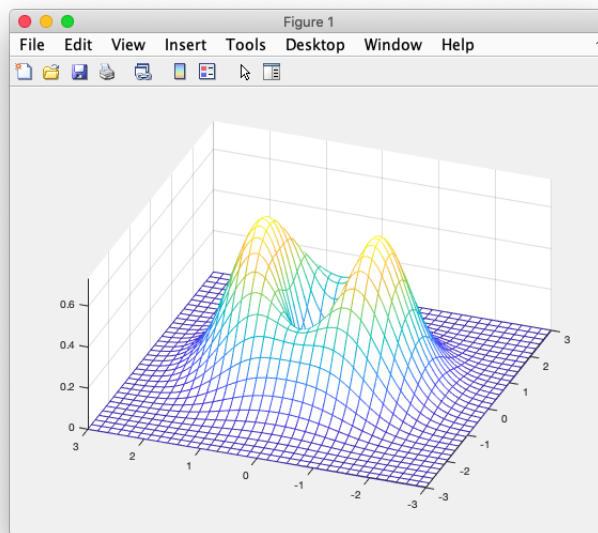
Maksimiziraj funkcijo in izračunaj njeno vrednost v točki maksimuma:

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

Rešitev:

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

```
1 syms x y
2 f = (x^2 + 2*y^2) * exp( -(x^2 + y^2) );
3 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):

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

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

```
1 [x, fval, flag] = fminsearch(@(x)-maj2008_2(x), [0, 1])
2 % Vrne v stilu:
3 % x = [0 1]
4 % fval = -0.7358
5 % flag = 1
6 [x, fval, flag] = fminsearch(@(x)-maj2008_2(x), [0, -1])
7 % Vrne v stilu:
8 % x = [0 -1]
```

```

9 % fval = -0.7358
10 % 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:

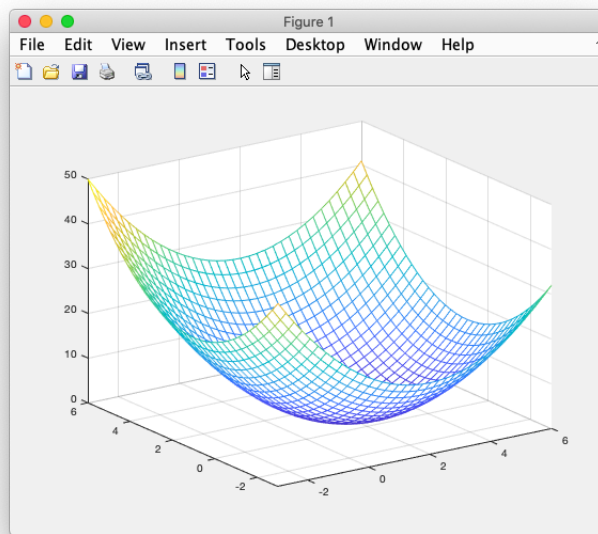
$$f(x) = (x_1 - 2)^2 + (x_2 - 1)^2 \quad (3)$$

ob naslednjih omejitvah

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

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`):

```

1 % Definicija funkcije ki jo minimiziramo
2 function y = maj2008_3(x)
3     y = (x(1) - 2)^2 + (x(2) - 1)^2;
4 end
5
6 %

```

... in omejitve (maj2008_3_con.m):

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

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

```
1 [x, fval, flag] = fmincon(@(x)maj2008_3(x), [0, 0], [], [], [], [], [], [], @(x)
    )maj2008_3_con(x))
2 % Vrne v stilu:
3 % x = [1.000 -1.000]
4 % fval = 1.0000
5 % flag = 1
```

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

3.1.4 Naloga 4.)

Navodila:

Poišči vse ničle enačbe:

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

Rešitev:

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

```
1 roots([1 -6 -92 402 91 -396])
2 % Vrne v stilu:
3 % ans =
4 %     -9.0000
5 %     11.0000
6 %     4.0000
7 %     -1.0000
8 %     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`:

```

1 % Koeficienti funkcije, ki jo minimiziramo
2 f = [92 89 90 91 91 95 87 90 92];
3
4 % Matrika in vektor, kjer je relacija <=
5 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];
6 b = [320000 270000 190000];
7
8 % Matrika in vektor, kjer je relacija =
9 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];
10 beq = [100000 180000 350000];
11
12 % X so količine kupljenega goriva, fval pa strošek
13 % (zeros(1, 9) - LB, zelis nenegativne resitve)
14 [x, fval] = intlinprog(f, [], A, b, Aeq, beq, zeros(1, 9));
15 % Vrne v stilu:
16 % x =
17 %          0
18 %          0
19 %      320000
20 %          0
21 %      120000
22 %          0
23 %      100000

```



```

24 %      60000
25 %      30000
26 %
27 % fval = 56580000

```

Iz tega lahko razberemo, da je najugodnejše 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.)

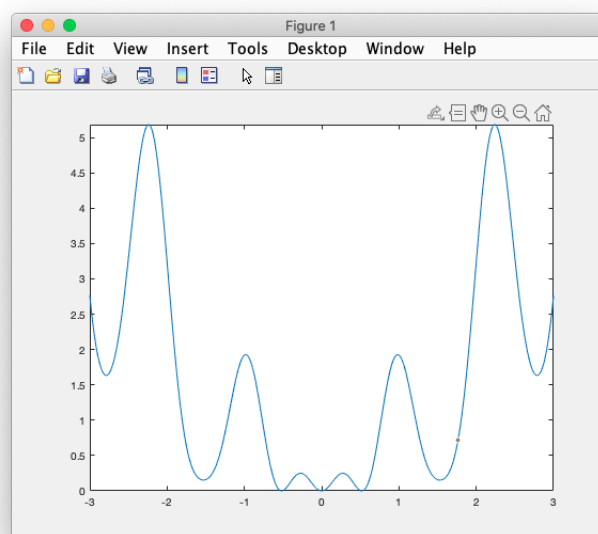
Navodila:

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

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

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):

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

... in nato kličemo funkcijo `fsolve`:

```
1 % Poskusimo najprej s [0, 0]
2 x = fsolve(@x)junij2008_2(x), [0, 0])
3 % Kar vrne, da ni resitve ...
4 % Premaknimo se v negativni del
5 xL = fsolve(@x)junij2008_2(x), [-.5, -.5])
6 % ... in v pozitivni
7 xR = fsolve(@x)junij2008_2(x), [.5, .5])
8 % In dobimo najmanjsi resitvi (ostale so periodične)
9 % xL = [0.7854 -0.7854]
10 % 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):

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

In kličemo funkcijo `fmincon`:

```
1 [x, fval] = fmincon(@x)junij2008_3(x), [1 1 1], [-1 -2 -1], -4, [], [], [0 0
2     0])
3 % Vrne v stilu:
4 % x = [2.9412 0.5294 0.0000]
```

```
4 % 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.)

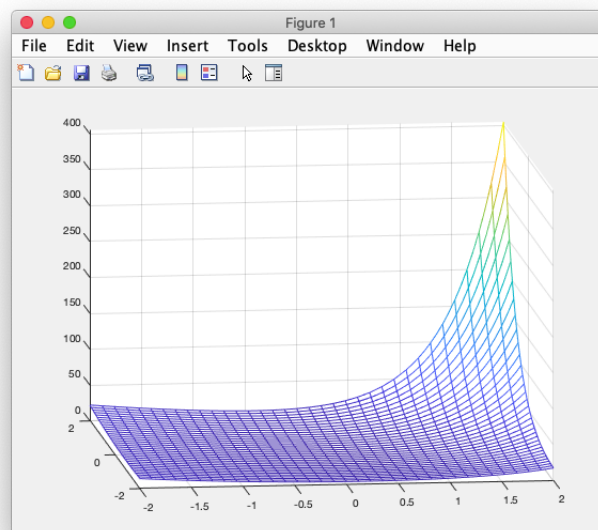
Navodila:

Poišči minimum funkcije in njeno funkcijsko vrednost v točki minimuma:

$$f(x) = 2x_1^2 - x_1x_2 + x_2^2 - 3x_1 + e^{2x_1+x_2} \quad (12)$$

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. Definirajmo nov *m-file* (*juni2008_4.m*):

```
1 function y = juni2008_4(x)
2     y = 2 * x(1)^2 - x(1) * x(2) + x(2)^2 - 3*x(1) + exp(2*x(1) + x(2));
3 end
```

... in kličemo funkcijo `fminsearch`:

```
1 [x, fval, flag] = fminsearch(@juni2008_4(x), [0, 0])
2 % Vrne v stilu:
3 % x = [0.1737    -0.3915]
4 % fval = 0.7174
5 % 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:

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

omejitve pa so sledeče:

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

Enačno lahko rešimo s pomočjo funkcije `intlinprog`:

```
1 % Koeficienti funkcije
2 f = [ 2 3 5 6 8 8];
3 % Matrika A, obrnemo predznak
4 A = [-20 -30 -40 -40 -45 -30; -50 -30 -20 -25 -50 -20; -4 -9 -11 -10 -9 -10];
5 % Vektor b (desna stran omejitev) tudi obrnemo predznak
6 b = [-70 -100 -20];
7 % Aeq in beq sta prazna, prav tako ne potrebujemo celostevilskih resitev,
  nastavimo pa se spodnjo mejo
8 [x, fval] = intlinprog(f, [], A, b, [], [], zeros(1, 6))
9 % Vrne v stilu:
10 % x = 0.9091
11 %      1.8182
12 %      0
13 %      0
14 %      0
15 %      0
```

```
16 % 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.)

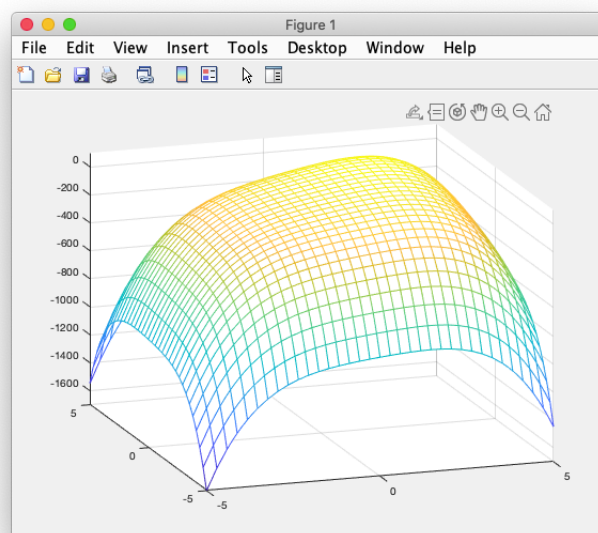
Navodila:

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?

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

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`. Definiramo si funkcijo v ločenem *m-file* (`september2008_1.m`):

```
1 function y = september2008_1(x)
2     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)
3     ^4;
4 end
```

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

```
1 [x, fval] = fminsearch(@september2008_1, [-3, -3])
2 % Vrne v stilu:
3 % x = [1.9548    1.8380]
```

```
4 % 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):

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

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

```
1 % Seznam vseh resitev
2 s = [];
3 for i=1:100
4     randX0 = 20 * rand(1, 3) - 10; % naključna zacetna točka
5     [x, fval, flag] = fminsearch(@(x)-september2008_2(x), randX0);
6     if flag == 1
7         % Dodaj pravilno resitev v seznam vseh
8         s = [s; [x fval]];
9     end
10 end
11 ss = sortrows(s, 4); % Sortiraj resitve po velikosti
12 ss(1:10,:) % izpisi prvih 10 vrstic
13 % Vrne 10 identicnih vrstic
14 % ans =
15 %      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`):

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

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

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

... in kličemo funkcijo `fmincon`:

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

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 ($\text{flag} = 1$): $x_{min} = [0.0000, 1.3333, 0.6667]$, 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 $l_1 + l_2 = l$. 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 vozlišca, druga možnost pa je da so spremenljivke povezave. Tu je logično da je spremenljivka povezave. Iz vsakega vozlišca 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: $x_12 + x_13 + x_14 = 1$ // i- točno po eni izmed poti mormo Podnobna enacba dobimo za zadnje vozlišce $x_79 + x_69 + x_89 = 1$ // i- točno po eni povezavi moramo prit v sink

Podobne enacbe moramo spisat za vas vmesna vozlišca. Recimo za vozlišce st. 3: vsota prihodnih mora bit enaka vsoti izhodnih (ce smo prisli v neko vozlišce potem moramo iz njega tudi oditi) $x_25 + x_35 = x_56 + x_57 + x_58$

treba je prestet vse puscice pa prestevilciti spremenljivke -i x1, x2, x3, x4 ... x16 A, b sta prazni -i vse omejitve so tipa je enako Aeq ima 16 stolpcev in 9 vrstic -i 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.

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

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 najboljših rešitev! Po elektronski pošti oddaj tudi vse izdelane programe in dobljene rešitve!

Rešitev:

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

```
1 function y = februar2014_1(x)
2     sum1 = 0;
3     sum2 = 0;
4     sum3 = 0;
5     sum4 = 0;
6     for i=1:5
7         sum1 = sum1 + i * cos((i+1) * x(1) + i);
8         sum2 = sum2 + i * cos((i+1) * x(2) + i);
9         sum2 = sum2 + i * cos((i+1) * x(3) + i);
10        sum2 = sum2 + i * cos((i+1) * x(4) + i);
11    end
12    % Vrni zmnozek vsot
13    y = sum1 * sum2 * sum3 * sum4;
14 end
```

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

```
1 resitve = []; % Seznam najdenih resitev
2 for i=1:100
3     % Požene iskanje s naključnim vektorjem velikosti 1x4 (uposteva omejitev
4     % -10 <= x(i) <= 10)
5     [x, fval, flag] = fminsearch(@(x)februar2014_1(x), 20*rand(1,4)-10);
6     if flag == 1 % Vkljuci samo dobre resitve
7         resitve = [resitve; [x fval]];
8     end
9 end
10 % resitve =
```



```

10 %      -4.5601      -5.3684      7.9907      8.1739      0
11 %      2.0729      -2.6953      1.9718      3.3697      0
12 %      7.8913      -8.2533      0.7802     -1.4310      0
13 %      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 maksimum¹ podane funkcije in njeno vrednost v tej točki ob navedenih omejitvah za $n = 2, 3, 4, 5$.

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

$$\sum_{i=1}^n x_i^2 = 1 \quad (21)$$

$$0 \leq x_i \leq 1 \quad i = 1, \dots, n$$

Rešitev:

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

```

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

```

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

```

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

```

... in vse skupaj poženemo v nekem skriptu

```

1 resitve = [];
2 n = 3; % Vrednost n iz navodil
3 for i=1:20 % Veckrat poišči resitev
4     [x, fval, flag] = fmincon(@(x)-februar2014_2(x), rand(1,n), [], [], [], [],
5     % zeros(1,n), ones(1,n), @(x)februar2014_2_con(x));
6     if flag == 1
7         % Pomnoži resitev z obratno vrednost (maksimum) in koren od n na n
8         resitve = [resitve; [x -fval]];
9     end
10 end
11 % (vse vrstice enake):
12 % resitve = 0.5774      0.5774      0.5774      1.0000

```

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

¹V 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.)

Geometrijska naloga, prisekan stožec ima špic odrezan. Skica: krog v njega narises trapez, po tem si nastavis enacbe za omejitev (stranski ris) slika februar2014_3 + skica na namizju R ... polmer krogle (dolocen =10cm) R1 ... sponjo presekan polmer kroznice R2 ... zgoraj prisekan polmer kroznice (naceloma R2 j R1) x1 .. h1 x2 .. h2 x3 .. R1 x4 .. R2

enacba za funkcijo je enacba za volumen Dobis pravokotne trikotnike R,h2,R2 in R,h1,R1
 $R1^2 + h1^2 = R^2$ $R2^2 + h2^2 = R^2$

2 file februar2014_3.m in februar2014_3_con.m

```
[x, fval, flag] = fmincon(@(x)-februar2014_3(x), [8 7 9 8], [], [], [], [], zeros(1,4), [10 10 10 10], @(x)februar2014_3_con(x))
```

Resitev: bojda je valj, ni blo navodil tako da ni dodatnih omejitev treba pisat x =

5.7735 5.7735 8.1650 8.1650

fval =

-2.4184e+03

flag =

1

Ostane se drugi del: vsota obeh polmerov je vecja ko 12 -i dodas v februar2014_3_con -x(3) - x(4) + 12”v c

3.5.4 Naloga 4.)

gle jslika na namizju. Izdelki so nedeljivi -i 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 -i [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