

Informatička podrška optimizaciji nelinearnih funkcija

Rajić, Damjan

Master's thesis / Diplomski rad

2010

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:235:302086>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-13**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

DIPLOMSKI RAD

DAMJAN RAJIĆ

Zagreb, 2010.godina

Sveučilište u Zagrebu
Fakultet strojarstva i brodogradnje

DIPLOMSKI RAD

INFORMATIČKA PODRŠKA OPTIMIZACIJI NELINEARNIH FUNKCIJA

Voditelj rada:

prof.dr.sc.Nedeljko Štefanić

Damjan Rajić

Zagreb, 2010.godina

SAŽETAK RADA

U prvom dijelu diplomskog rada dana je teoretska podloga formiranja matematičkog modela nelinearnog programiranja. Nakon toga slijedi treće poglavlje u kome su dani opisi softverskih paketa za optimizaciju nelinearnih funkcija, s uključenom primjenom kroz primjere rješavanja optimizacijskih problema korištenjem softvera Maple, Lingo, JMP i Matlab.

Četvrto poglavlje donosi optimizaciju u programu Maple kroz postupak formiranja matematičkog modela i nalaženje optimuma funkcije cilja, te prikaz podloga za korištenje softvera.

U nastavku se nalazi provedba faktorskog i centralno kompozitnog plana pokusa u procesu obodnog brušenja ravne površine, te optimizacija dvaju faktora brzina glavnog i pomoćnog gibanja obzirom na odzive, odnosno postignute parametre kvalitete obrađene površine R_a i P_c .

Šesto poglavlje provodi optimizaciju u softveru Matlab, gdje su kroz tehnološki problem dani opisi i podloge za korištenje i izvršavanje nelinearnog programiranja testiranjem softverskog paketa.

Sadržaj

SAŽETAK RADA	4
Popis slika	8
Popis tablica	10
Popis oznaka i mjernih jedinica fizikalnih veličina	11
Izjava studenta	12
1. UVOD.....	13
2. MATEMATIČKI MODEL NELINEARNOG PROGRAMIRANJA	15
2.1. Formiranje matematičkom modela nelinearnog programiranja.....	17
2.2. Ekstremi funkcije.....	19
2.2.1. Lokalni i globalni minimum	20
2.2.2. Lokalni i globalni maksimum.....	21
2.3. Ispitivanje prirode ekstrema	21
2.3.1. Gradijent vektor ∇f	22
2.3.2. Hesseova matrica	23
3. SOFTVERSKI PAKETI ZA OPTIMIZACIJU NELINEARNIH FUNKCIJA	25
3.1. Računalna optimizacija	25
3.2. Softverski paket: Maple™	26
3.2.1. Modularnost i podrška Maple-a	28
3.2.1.1. Osnove sintakse programskog jezika Maple	29
3.2.1.2. Naredbe unutar radnog lista	31
3.3. Softverski paket: LINGO 12.0	32

3.3.1. Glavne karakteristike LINGO-a.....	33
3.3.2. Integrirani rješavači (solveri)	34
3.3.2.1. <i>Linearni solveri</i>	34
3.3.2.2. <i>Nelinearni solveri</i>	35
3.3.3. Nelinearni solver za globalni optimum - Global Solver	36
3.3.3.1. <i>Konveksnost funkcije</i>	36
3.3.3.2. <i>Optimiranje nekonveksne funkcije cilja</i>	37
3.3. Softverski paket: JMP®8.....	40
3.3.1. Modul za planiranje pokusa.....	41
3.3.2. Pokusi odzivnih površina.....	42
4. OPTIMIZACIJA U MAPLE-u	43
4.1. Opis primjera za testiranje softvera Maple	43
4.2. Izvod matematičkog modela.....	45
4.2.1. Definiranje matematičkog modela nelinearnog programiranja.....	46
4.3. Postavljanje i rješavanje primjera	47
4.3.1. Optimalno rješenje i interpretacija.....	50
4.4. Grafički prikaz funkcije cilja.....	50
5. OPTIMIZACIJA PARAMETARA BRUŠENJA KORIŠTENJEM JMP-a	51
5.1. Priprema za faktorsko planiranje pokusa	53
5.1.1. Faktori i provedba centriranja i ujednačavanja mjere.....	53
5.2. Planiranje i provođenje faktorskog pokusa	54
5.2.1. Izračun i značenje koeficijenata modela.....	56
5.2.2. Utvrđivanje adekvatnosti dobivenih linearnih modela	59
5.3. Priprema za centralno kompozitno planiranje pokusa.....	61
5.3.1. Smještaj aksijalnih (zvjezdastih) točaka.....	62

5.4. Centralno kompozitni pokus.....	64
5.4.1. Nelinearni modeli Ra i Pc i njihovi koeficijenti.....	66
5.4.2. Analiza rezultata	67
5.4.3. Optimizacija traženih faktora x_1 i x_2	69
5.4.3.1. <i>Zaključak</i>	72
6. OPTIMIZACIJA U MATLAB-u	73
6.1. Nelinearna optimizacija s ograničenjima.....	73
6.1.1. Postavljanje problema za provođenje optimizacije.....	74
6.1.2. Provođenje optimizacije	75
6.1.2.1. <i>Način 1. – Korištenje grafičkog korisničkog sučelja</i>	75
6.1.2.2. <i>Način 2. – Korištenje naredbene funkcije</i>	77
6.1.3. Analiza rješenja i grafički prikaz funkcije cilja	78
6.2. Usporedba rezultata prema [21] i [22]	80
7. ZAKLJUČAK	85
8. LITERATURA	86

Popis slika

Slika 1: Primjer optimuma/maksimuma nelinearne funkcije cilja (paraboloid)	13
Slika 2: Prikaz pojedinih koraka prilikom provedbe operacijskih istraživanja, [2].....	14
Slika 3: Proces modeliranja, [4].....	15
Slika 4: Struktura matematičkog modela.....	16
Slika 5: Minimum $f(x)$ odgovara maksimumu $-f(x)$, [3]	16
Slika 6: Optimalni vektori x^* funkcija $cf(x)$ ili $c+f(x)$ odgovaraju i funkciji $f(x)$	17
Slika 7: Primjer različitih uzročnika ograničenja	18
Slika 8: Primjer različitih ciljeva minimalizacije.....	20
Slika 9: Primjer različitih ciljeva maksimalizacije	21
Slika 10: Ekstremi kod funkcije s jednom varijablom, [3]	22
Slika 11: Početni prozor unutar Maple-a	26
Slika 12: Primjer matematičke notacije unutar Maple-a	27
Slika 13: Primjer postavljenog modela NLP-a i njegove optimizacije	30
Slika 14: Glavni startni okvir LINGO-a u windowsima.....	32
Slika 15: Prikaz postojećih solvera unutar LINGO-a	34
Slika 16: Konveksna funkcija na intervalu.....	36
Slika 17: Nekonveksan skup i epigraf konveksne funkcije.....	37
Slika 18: Prozori postavljenog modela i generiranje izvješća o modelu.....	38
Slika 19: Graf funkcije $f(x)=\cos(2/x)+\sin(2\cdot x)$	39
Slika 20: JMP-ov polazni okvir i opcije unutar modula DOE	40
Slika 21: Glave etape provođenja pokusa unutar JMP-a	41
Slika 22: Istjecanje mlaza fluida iz otvora cijevi	43
Slika 23: Putanja čestice fluida.....	44
Slika 24: Prikaz unesenog matematičkog modela u Maple	47
Slika 25: Unos naredbe za optimizaciju Maximize.....	48
Slika 26: Optimizacijski asistent	49
Slika 27: Rješenje dobiveno pomoću Optimizacijskog asistenta	49
Slika 28: Graf putanje čestice fluida dobiven pomoću funkcije <i>plot</i>	50
Slika 29: Obodno brušenja ravne površine, istosmjernim uzdužnim načinom	51

Slika 30: Definiranje parametra P_c , odnosno broja vrhova elementa profila, [18]	54
Slika 31: Potpuni faktorski 2^2 plan pokusa	55
Slika 32: Interakcijski dijagrami za modele Ra i Pc	58
Slika 33: Konturni dijagram linearnog modela Ra	58
Slika 34: Usporedba odziva dobivenih modelom i očitanih iz kontrolne točke	60
Slika 35: Domena centralnog kompozitnog plana pokusa s pripadajućim točkama, [17]	62
Slika 36: Postavljanje odziva i faktora u izbornik JMP-a	64
Slika 37: JMP-ov izbornik za izradu matematičkog modela željenih odziva	65
Slika 38: Iznosi koeficijenata nelinearnih modela Ra i Pc dobivenih u JMP-u	66
Slika 39: Prikaz razlika stvarnih i odzivnih vrijednosti za model Ra i Pc	67
Slika 40: Prikaz kvalitete modela Ra i Pc kroz koeficijente determinacije	68
Slika 41: Interakcijski dijagrami za nelinearne modele Ra i Pc	68
Slika 42: Odzivne površine nelinearnih modela i konturni dijagram	69
Slika 43: Postavljanje vrijednosti za pojedinačne funkcije poželjnosti modela Ra i Pc	70
Slika 44: Optimalni faktori dobiveni pomoću globalne funkcije poželjnosti	71
Slika 45: Izgled postavljenog korisničkog grafičkog sučelja za optimizaciju u Matlab-u	76
Slika 46: Konturni dijagram uz konstantan iznos dubine rezanja od 1,3 mm	79
Slika 47: Prostorna krivulja za fiksnu vrijednost dubine rezanja od 1,3 mm	80
Slika 48: Konturni dijagram funkcije Usporedba1(x_1 , x_2) uz fiksni $x_3=14,8$	81
Slika 49: Odzivna površina funkcije Usporedba2(x_1 , x_2) uz fiksni $x_3=0,05$	83
Slika 50: Odzivna površina funkcije funkcijaz (x_1 , x_2)	84

Popis tablica

Tablica 1: Prikaz teorema o ekstremima za jednovarijabilnu optimizaciju	19
Tablica 2: Zaključci na temelju testova 1. i 2. derivacije.....	24
Tablica 3: Primjer osnovnih modula Maple-a	28
Tablica 4: Pet osnovnih pravila Maple-ove sintakse, [9.]	29
Tablica 5: Prikaz osnovnih pojmova vezanih uz naredbe u Maple-u.....	31
Tablica 6: Elementi općeg postuliranog matematičkog modela	42
Tablica 7: Zadani podaci u primjeru istjecanja fluida	44
Tablica 8: Formulirani model nelinearnog programiranja.....	47
Tablica 9: Domena faktorskog plana pokusa	53
Tablica 10: Odzivi traženih matematičkih modela.....	53
Tablica 11: Rezultati faktorskog pokusa	55
Tablica 12: Vrste točaka iz centralnog kompozitnog plana pokusa.....	61
Tablica 13: Vrijednost α kao funkcija n_f , n_α i n_0 za kriterij približne ortogonalnosti.....	63
Tablica 14: Centralni kompozitni plan pokusa i rezultati za oba odziva uneseni u JMP	65
Tablica 15: Usporedba odziva modela i stvarnih vrijednosti dobivenih pokusom	67
Tablica 16: Vrijednosti optimuma funkcija R_a i P_c unutar raspona kodiranih faktora -1 i 1 ...	72
Tablica 17: Najbolji mogući kompromis kombinacije traženih faktora	72
Tablica 18: Dozvoljeni intervali varijabli za pretraživanje optimalnog rješenja, [20]	74
Tablica 19: Prikaz dobivenog rješenja optimizacijom	77

Popis oznaka i mjernih jedinica fizikalnih veličina

Oznaka	Mjerna jedinica	Opis
RSE		pokusi odzivnih površina, (response surface experiments)
v_g	m/s	tangencijalna brzina
v_p	m/min	brzina pomicanja
R_a	μm	parametar hrapavosti profila površine, odnosno amplitudni parametar
P_c	vrh/cm	uzdužni parametar, odnosno broj vrhova elemenata profila (eng. peak count) (broj elemenata vrhova profila) / (dužina ocjenjivanja)
V_0		fizikalna centralna vrijednost
CCD		centralni kompozitni plan (central composite design)
n_f		zbroj pokusa iz faktorskog dijela
n_α		zbroj pokusa iz aksijalnog (zvjezdastog) dijela
n_0		zbroj pokusa iz centralnog dijela
n		ukupan zbroj izvođenja pokusa
DOM		optimizacije metodologijom poželjnosti (desirability optimization methodology)
α		udaljenost od centra pokusne domene
d_i		pojedinačne vrijednosti funkcija poželjnosti
GUI		grafičko korisničko sučelje (graphical user interface)
v_c	m/min	brzina rezanja
f_z	mm/zubu	posmak
a_p	mm	dubina rezanja

Izjava studenta

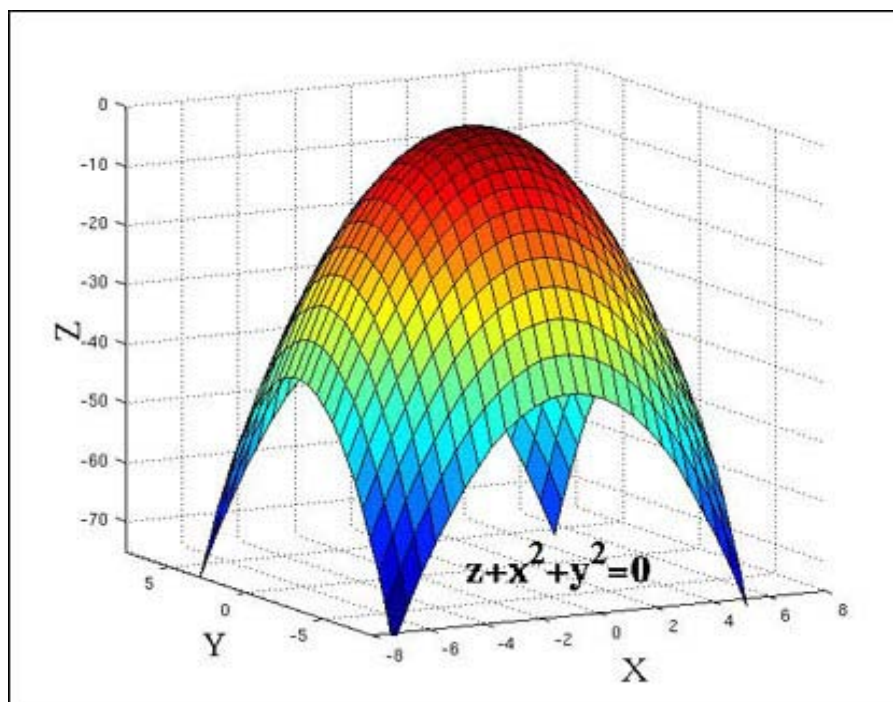
Izjavljujem da sam diplomski rad pod naslovom „Informatička podrška optimizaciji nelinearnih funkcija“ izradio samostalno služeći se navedenom literaturom, uz nadzor mentora prof. dr. sc. Nedeljka Štefanića.

Damjan Rajić

1. UVOD

Optimizacija je ostvarivanje cilja, odnosno najboljih rješenja od mogućih alternativa korištenjem egzaktnih (matematičkih) metoda kroz stvaranje optimizacijskog modela. Cilj optimizacijskog modela je u najčešćim slučajevima minimalizacija gubitaka ili maksimizacija dobiti, a primjena je raznolika kroz različita područja tehnike, poslovanja i prirodnih znanosti. To se postiže traženjem ekstrema realne funkcije sistematskim odabirom vrijednosti realne ili cjelobrojne varijable, odnosno vektora nepoznatih vrijednosti (komponentni vektor), čije su vrijednosti ograničene domenom funkcije cilja, tj. ograničenjima. Kada je funkcija cilja i/ili ograničenja nelinearna, takav skup postupaka se naziva **nelinearno programiranje (NP)**.

U pripremi pretvorbe funkcionalnog dijela/sustava u optimalni pomažu nam formalni (*dogovorom utvrđeni*) postupci operacijskih istraživanja. Operacijska istraživanja su znanstvena disciplina koja primjenom kvantitativnih i kvalitativnih metoda proučava probleme koji se javljaju pri izvođenju neke operacije i iznalazi načine najboljeg rješenja takve operacije, a jedna od tih metoda je nelinearno programiranje.

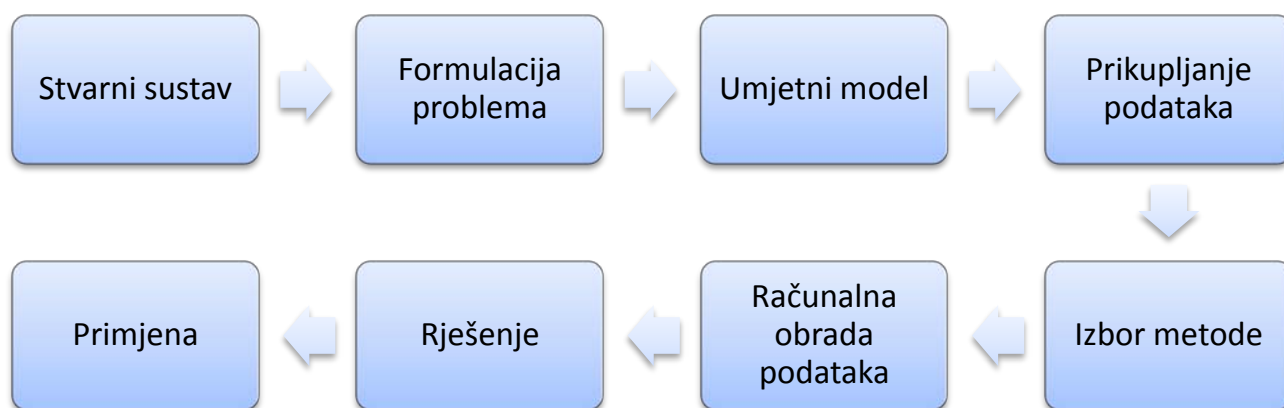


Slika 1: Primjer optimuma/maksimuma nelinearne funkcije cilja (paraboloid)

Postupak operacijskih istraživanja, a time i nelinearnog programiranja mogao bi se svesti na sljedeće korake, [2]:

- 1) Postavljanje granica sustava koji se istražuje (formulacija problema),
- 2) Konstrukcija modela,
- 3) Modelsko računanje,
- 4) Primjena.

Svaki korak ovog postupka predstavlja kompleks pojedinih postupaka, a shematski prikaz između pojedinih koraka prikazan je slikom 2.



Slika 2: Prikaz pojedinih koraka prilikom provedbe operacijskih istraživanja, [2]

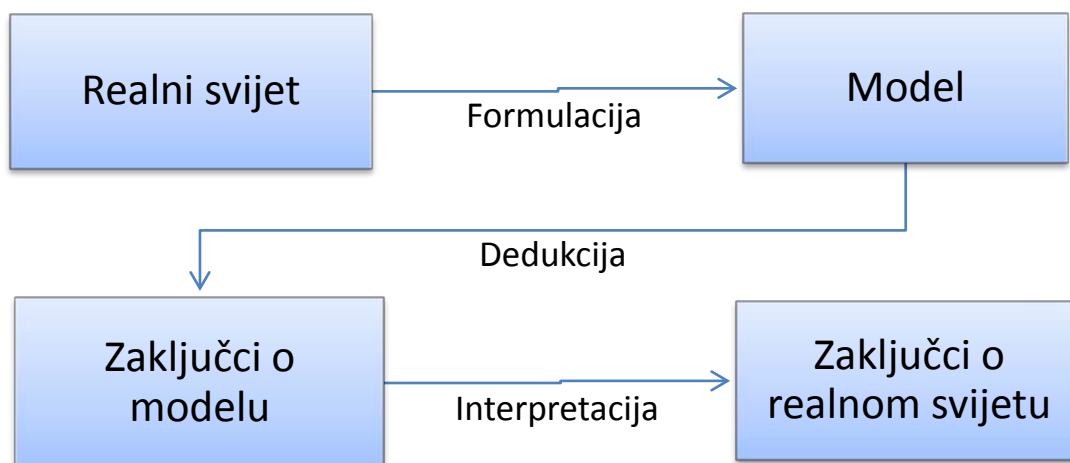
Da bi se mogla obaviti kvantitativna analiza nekog sustava iz realnog svijeta, nužno je izraditi *matematički model* tog sustava. Postupak modeliranja dosta je osjetljiv i zahtjeva određeno iskustvo, a možda i intuiciju; jer gotovo i nema zbivanja u realnom svijetu koja bi se mogla potpuno točno prenijeti u matematički model. Veze i zavisnosti u realnom svijetu najčešće su nelinearne i nedeterminističke, a također i podaci o realnom svijetu nisu nikada apsolutno točni.

U okviru ovog rada biti će prikazan matematički model nelinearnog programiranja, njegovo nastajanje i rješavanje optimizacijskih problema u softverskim paketima.

2. MATEMATIČKI MODEL NELINEARNOG PROGRAMIRANJA

Analizom stanja u realnom svijetu dobiju se informacije za formulaciju modela. Model je idealizirani opis stvarnosti, što znači da je stvarnost uzeta u obzir u odnosu na značajne činjenice koje se odnose na problem. Problem koji zahtjeva odziv je matematički model plus pitanje koje se postavlja u stvarnom problemu. Odabranim postupcima nad modelom dobije se odziv, odnosno zaključci o ponašanju modela. Te zaključke treba interpretirati da bi bili upotrebiti u realnom svijetu, a onda ih treba usporediti sa zaključcima do kojih se dolazi analizom realnog svijeta, ali bez korištenja modela. Tek se tada mogu formulirati elementi za donošenje stvarne odluke.

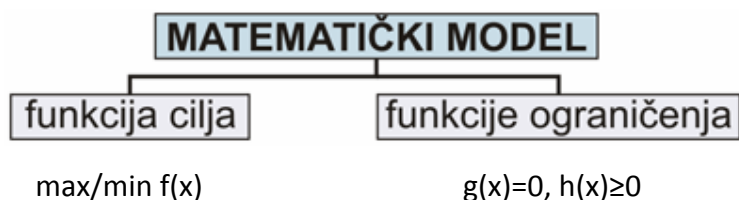
Rješenje koje se postiže pomoću matematičkog modela i korištenjem računala, tj. softvera, je najprije formalni odgovor na pitanje koje se postavilo preko modela. U matematičkim modelima do rješenja se dolazi dobro definiranim metodama optimalizacije. Zadnja faza postupka operacijskih istraživanja je primjena rezultata modela na realni sustav.



Slika 3: Proces modeliranja, [4]

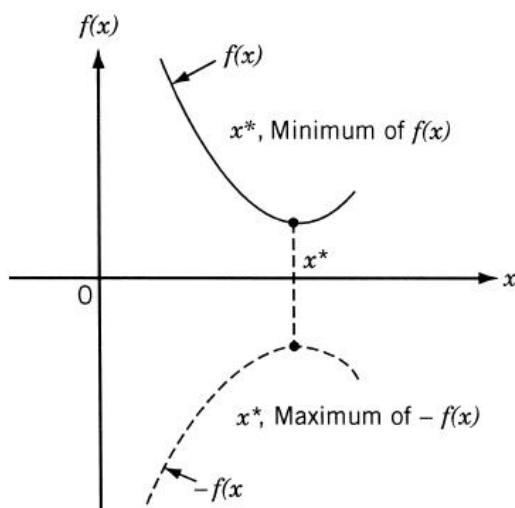
Mnogi problemi koji se pojavljuju u praksi kao i u znanosti, prevedeni na matematički oblik (model) sastoje se u određivanju ekstrema (maksimuma ili minimuma) neke funkcije $f(x)$ uz zadana ograničenja $g_i(x)$. Pritom funkcija čiji se ekstrem želi odrediti obično ima određeno značenje, kao npr. profit, troškovi i dr., dok ograničenja, najčešće u obliku

nejednadžbi ili jednadžbi, predstavljaju tehnološke, tržišne i dr. uvjete za zadani problem. To znači da se između mnogih varijanata koje mogu biti rješenja određenog problema nastoji pronaći najbolja, optimalna po nekom kriteriju.



Slika 4: Struktura matematičkog modela

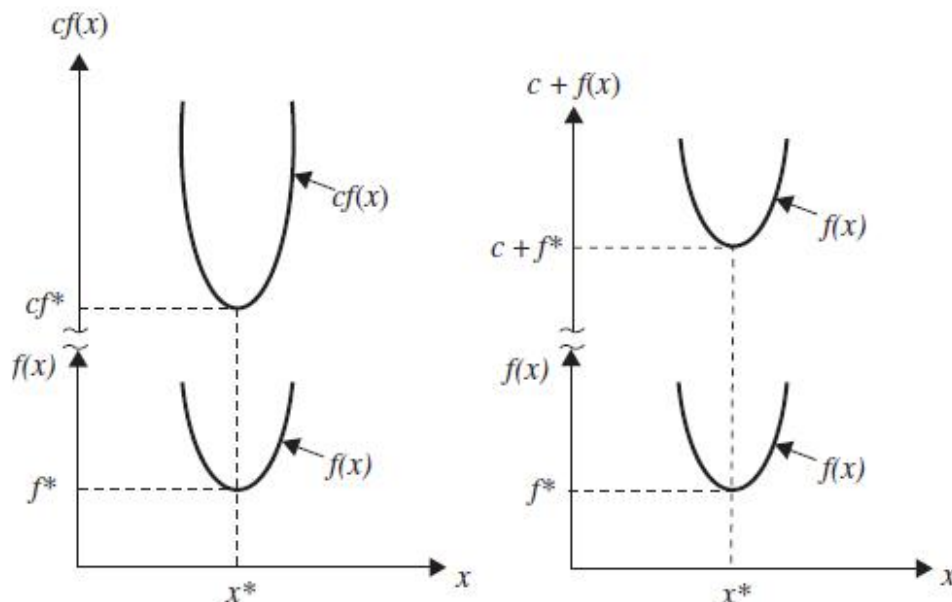
Na slici 5 je prikazana posebnost optimizacije gdje je vektor, odnosno točka x^* minimum funkcije $f(x)$, a ujedno i maksimum funkcije $-f(x)$. Zato se bez gubljenja na univerzalnosti može tražiti samo minimum, budući da se maksimum može naći traženjem minimuma negativne funkcije. Svojstvo: $\max f(x) = -\min[-f(x)], x \in S$.



Slika 5: Minimum $f(x)$ odgovara maksimumu $-f(x)$, [3]

Također, korištenjem sljedećih operacija na funkciju cilja $f(x)$ neće se promijeniti optimalni vektor, što je vidljivo na slici 6:

1. Množenje ili dijeljenje funkcije $f(x)$ s pozitivnom konstantom c .
2. Zbrajanje ili oduzimanje pozitivne konstante c sa ili od funkcije $f(x)$.



Slika 6: Optimalni vektori x^* funkcija $cf(x)$ ili $c+f(x)$ odgovaraju i funkciji $f(x)$

2.1. Formiranje matematičkom modela nelinearnog programiranja

Problem optimiranja odnosno nelinearnog matematičkog programiranja sastoji se u određivanju ekstremne vrijednosti (max/min) funkcije cilja na nekom skupu S koji predstavlja domenu funkcije, a može se definirati npr. $f(x): S \mapsto \mathbb{R}$, gdje je funkcija $f(x)$ od n varijabli x_1, x_2, \dots, x_n definirana na skupu S iz euklidskog prostora \mathbb{R}^n . Funkcija $f(x)$ ima zadatak preslikati domenu S u kodomenu \mathbb{R} , dok je S podskup skupa \mathbb{R}^n .

Za provedbu optimiranja potrebno je naći vektor $\mathbf{x} = [x_1, x_2, \dots, x_n]^T$, gdje T označava transpoziciju, koji minimizira funkciju cilja $f(\mathbf{x})$ uz ograničenja $\mathbf{x} \in S$.

Odrediti vektor promjenjivih varijabli $\mathbf{x} = \begin{Bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{Bmatrix}$ koji minimizira $f(\mathbf{x})$,

uz objedinjena ograničenja u skupu S , odnosno $\mathbf{x} \in S$.

Ako je funkcija cilja podložna ograničenjima oni se mogu prikazati u ovom obliku:

$$\left. \begin{array}{l} g_i(\mathbf{x}) = 0, \quad i = 1, 2, \dots, m \\ h_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, p \\ \mathbf{x} \in G \end{array} \right\} \in S \text{ (skup ukupnih ograničenja)}$$

Ograničenja $g_i(\mathbf{x})$ i $h_j(\mathbf{x})$ su realne funkcije čija se preslikavanja domene na kodomenu može prikazati na sljedeći način: $g_i: \mathbb{R}^n \mapsto \mathbb{R}, i=1,2,\dots,m$, $h_j: \mathbb{R}^n \mapsto \mathbb{R}, j=1,2,\dots,p$, što znači da vektor promjenjivih varijabli \mathbf{x} može biti n -dimenzionalan, dok je G podskup skupa \mathbb{R}^n , odnosno $G \subseteq \mathbb{R}^n$.

Funkcije ograničenja $g_i(\mathbf{x})$ su u obliku jednadžbi, a $h_j(\mathbf{x})$ nejednadžbi, dok $\mathbf{x} \in G$ može biti npr. ograničenje nenegativnosti na varijable, pa je skup $G = \{\mathbf{x} | \mathbf{x} \in \mathbb{R}^n, \mathbf{x} \geq 0\}$. Tada se svaka točka iz skupa ukupnih ograničenja S može definirati kao objedinjenje svih ograničenja: $S = \{\mathbf{x} | g(\mathbf{x}) = 0, h(\mathbf{x}) \leq 0, \mathbf{x} \in G\}$, gdje su $g: \mathbb{R}^n \mapsto \mathbb{R}^m$ i $h: \mathbb{R}^n \mapsto \mathbb{R}^p$. Tako je svaka točka iz skupa S moguće rješenje programiranja, a njeni uzročnici mogu biti različiti kao npr. na slici 7.



Slika 7: Primjer različitih uzročnika ograničenja

Nelinearno programiranje (NP) se pojavljuje u slučajevima kada je funkcija cilja $f(\mathbf{x})$ nelinearna ili/i su u ograničenjima neke ili sve funkcije $g_i(\mathbf{x})$ i $h_j(\mathbf{x})$ nelinearne. Obzirom na broj varijabli koje se pojavljuju u vektoru promjenjivih varijabli \mathbf{x} , NP se može podijeliti prema [6] na ono s jednom i više varijabli, a s obzirom na ograničenja na NP bez ograničenja gdje je $G \in \mathbb{R}^n$, a minimum se traži na cijelom prostoru \mathbb{R}^n i ono sa ograničenjima.

2.2. Ekstremi funkcije

Ekstremi funkcije mogu biti lokalni i globalni, a za njihovo ispitivanje se koriste Fermatov teorem i teorem o monotonosti, [6]. Oni pokazuju da kada funkcija poprima u točki $c \in (a,b) \subseteq D$ svoju najmanju ili najveću vrijednost na intervalu (a,b) , a ako 1. derivacija postoji ona je $f'(c)=0$. Također, teorem o monotonosti pokazuje da predznak derivacije označava da li funkcija raste (pozitivan) ili pada (negativan) na nekom intervalu (a,b) , u slučaju kada je funkcija cilja definirana jednom varijablom, tj. kada je domena element euklidskog prostora u dvije dimenzije \mathbb{R}^2 . Sljedeća tablica 1. donosi tvrdnje o ekstremima za optimizacije s jednom varijablom.

Tablica 1: Prikaz teorema o ekstremima za jednovarijabilnu optimizaciju

Teoremi o ekstremima
Funkcija $f(x)$ je rastuća na intervalu (a,b) ako i samo ako je $f'(x) \geq 0$ za svaki $x \in (a,b)$.
Funkcija $f(x)$ je padajuća na intervalu (a,b) ako i samo ako je $f'(x) \leq 0$ za svaki $x \in (a,b)$.
Kada je funkcija $f(x)$ kontinuirana na zatvorenom i ograničenom intervalu (a,b) , tada $f(x)$ ima globalni optimum (min i max) na tom intervalu.
Kada $f(x)$ ima lokalni optimum u x_0 i $f(x)$ ima prvu derivaciju na malom intervalu oko x_0 , tada je $f'(x_0)=0$.
Ako je $f(x)$ dvostruko derivabilna na malom intervalu oko x_0 , a $f'(x_0)=0$ i $f''(x_0)>0$, tada $f(x)$ ima lokalni minimum u x_0 . Suprotno, kada je $f'(x_0)=0$ i $f''(x_0)<0$, tada funkcija $f(x)$ ima lokalni maksimum u x_0 .

U definiciji lokalnih ekstrema te u iskazima i dokazima teorema o ekstremima se koristi ε - okoline. Okolina točke c je interval $(x-\varepsilon, x+\varepsilon)$ pri čemu je $\varepsilon > 0$. Kod lokalnih ekstrema se traži da je vrijednost funkcije u točki ekstrema strogo najmanja ili najveća na nekoj okolini. S druge strane, definicija globalnih ekstrema dozvoljava da se globalni ekstrem nalazi u više točaka pa čak i na nekom intervalu.

U slučaju kada je funkcija cilja definirana s više varijabli, tj. kada je domena element euklidskog prostora u više dimenzije \mathbb{R}^n i kada se obavlja multivarijabilna optimizacija, tada je ε ($\varepsilon > 0$) okolina od vektora $\hat{\mathbf{X}}$. Odnosno, ε je skup vektora $\vec{\mathbf{X}}$ tako da odgovara sljedeće:

$$(\vec{\mathbf{X}} - \hat{\mathbf{X}})^T \cdot (\vec{\mathbf{X}} - \hat{\mathbf{X}}) \equiv (x_1 - \hat{x}_1)^2 + (x_2 - \hat{x}_2)^2 + \dots + (x_n - \hat{x}_n)^2 \leq \varepsilon^2.$$

U geometrijskoj terminologiji, okolina ε oko $\hat{\mathbf{X}}$ je unutrašnjost i granica od n -dimenzionalne sfere promjera ε centriranog oko $\hat{\mathbf{X}}$.

2.2.1. Lokalni i globalni minimum

Funkcija $f(x)$ ima globalni minimum $f(c)$ u točki $c \in D$ ako i samo ako je $f(x) \geq f(c)$ za svaki $x \in D$.

Funkcija $f(x)$ ima lokalni minimum $f(c)$ u točki $c \in D$ ako postoji ε -okolina točke c takva da je $f(x)$ neprekidna na toj okolini i pri tome vrijedi $f(x) > f(c)$ za svaki $x \in (c-\varepsilon, c) \cup (c, c+\varepsilon)$.

U općem slučaju funkcija cilja $f(\vec{\mathbf{X}})$ ima lokalni minimum u $\hat{\mathbf{X}}$ ako postoji okolina ε oko $\hat{\mathbf{X}}$ takva da je $f(\vec{\mathbf{X}}) \geq f(\hat{\mathbf{X}})$ za sve vektore $\vec{\mathbf{X}}$ u okolini ε u kojoj je funkcija definirana. Ako je uvjet zadovoljen za sve pozitivne ε (do neograničenih veličina), tada $(\vec{\mathbf{X}})$ ima u $\hat{\mathbf{X}}$ globalni minimum.



Slika 8: Primjer različitih ciljeva minimalizacije

2.2.2. Lokalni i globalni maksimum

Funkcija $f(x)$ ima globalni maksimum $f(c)$ u točki $c \in D$ ako i samo ako je $f(x) \leq f(c)$ za svaki $x \in D$.

Funkcija $f(x)$ ima lokalni maksimum $f(c)$ u točki $c \in D$ ako postoji ε -okolina točke c takva da je $f(x)$ neprekidna na toj okolini i pri tome vrijedi $f(x) < f(c)$ za svaki $x \in (c-\varepsilon, c) \cup (c, c+\varepsilon)$.

U općem slučaju funkcija cilja $f(\vec{X})$ ima lokalni maksimum u \hat{X} ako postoji okolina ε oko \hat{X} takva da je $f(\vec{X}) \leq f(\hat{X})$ za sve vektore \vec{X} u okolini ε u kojoj je funkcija definirana. Ako je uvjet zadovoljen za sve pozitivne ε (do neograničenih veličina), tada (\vec{X}) ima u \hat{X} globalni maksimum.



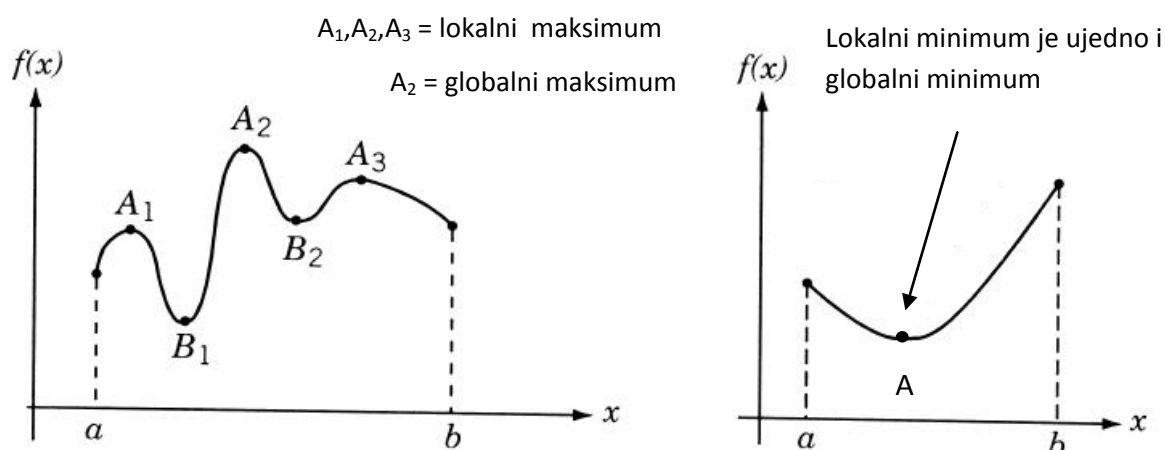
Slika 9: Primjer različitih ciljeva maksimalizacije

2.3. Ispitivanje prirode ekstrema

Za ispitivanje prirode ekstrema prilikom optimiranja nelinearnih funkcija, odnosno stacionarnog polja $f(x)$ gdje je funkcija $f(x)$ od n varijabli x_1, x_2, \dots, x_n definirana na skupu S iz euklidskog prostora \mathbb{R}^n , koristi se test prve i druge derivacije. Ekstremi funkcije, tj. točke kod kojih je prva derivacija jednaka nuli, nazivaju se stacionarne točke ili općenito kritične točke za višedimenzionalna stacionarna polja koje mogu biti stacionarne, odnosno točke koje nisu derivabilne. Zato je svaka stacionarna točka ujedno i kritična, dok obratno može i ne mora vrijediti.

Test prve derivacije služi za određivanje da li je kritična točka funkcije lokalni maksimum, minimum ili nijedno, dok u kombinaciji sa informacijama o konkavnosti, točkama infleksije (pregiba) i asimptotama funkcije služi za dobivanje grafa funkcije.

Kod multivarijabilnih funkcija prilikom testa druge derivacije kojim se još može ispitati konkavnost funkcije kao i točke infleksije, koristi se Hesseova matrica. Prilikom ispitivanja najbržeg smjera porasta funkcije stacionarnog polja $f(x)$ koristi se vektorsko polje, odnosno gradijent vektor ∇f .



Slika 10: Ekstremi kod funkcije s jednom varijablom, [3]

2.3.1. Gradijent vektor ∇f

Gradijent vektor ∇f , odnosno gradijent je vektorsko polje koje pokazuje smjer najvećeg porasta skalarnog polja $f(x)$. On je povezan sa skalarnim poljem $f(x_1, x_2, \dots, x_n)$ na način da su određene prve parcijalne derivacije funkcije $f(x)$:

$$\nabla f \equiv \left[\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right]^T$$

Gradijent se označava sa ∇f , $\vec{\nabla} f$ ili $\text{grad}(f)$, gdje je ∇ nabra simbol. Zapis, odnosno notacija $\nabla f|_{\hat{\mathbf{X}}}$ označava vrijednost gradijenta u točki $\hat{\mathbf{X}}$. Za mali odmak od $\hat{\mathbf{X}}$ u različitim smjerovima, smjer maksimalnog porasta funkcije $f(\mathbf{X})$ je u smjeru vektora $\nabla f|_{\hat{\mathbf{X}}}$.

Na primjer kod $f(x_1, x_2, x_3) = 3x_1^2 \cdot x_2 - x_2^2 x_3^3$, u točki $\hat{\mathbf{X}} = [1, 2, 3]^T$, gradijent je slijedeći:

$$\nabla f = \begin{bmatrix} 6x_1x_2 \\ 3x_1^2 - 2x_2x_3^3 \\ -3x_2^2x_3^2 \end{bmatrix}, \text{ odnosno } \nabla f|_{\hat{\mathbf{X}}} = \begin{bmatrix} 6(1)(2) \\ 3(1)^2 - 2(2)(3)^3 \\ -3(2)^2(3)^2 \end{bmatrix} = \begin{bmatrix} 12 \\ -105 \\ -108 \end{bmatrix}$$

Tako je u točki $\hat{\mathbf{X}} = [1, 2, 3]^T$, funkcija najbrže raste u smjeru $[12, -105, -108]^T$, dok veličina (magnituda) gradijenta koja se može dobiti kao drugi korijen zbroja kvadrata njegovih komponenta u pravokutnome koordinatnom sustavu, označava brzinu porasta u dobivenom smjeru.

2.3.2. Hesseova matrica

Hesseova matrica je kvadratna matrica derivacija drugog reda funkcije $f(\mathbf{x})$, a označava se prema [3] oznakom \mathbf{H}_f , dok se prema [5] koristi znak nabra pa je oznaka za Hesseovu matricu $\nabla^2 f$. Ona se koristi za opisivanje lokalne zakrivljenosti kod funkcija sa više varijabli, tj. kod multivarijabilnih funkcija prilikom testa druge derivacije.

Test druge derivacije se može primijeniti samo kod ne-degenerativne kritične točke \mathbf{X} ($\nabla f|_{\mathbf{x}} = 0$), a ona je definirana tako da je determinanta Hesseove matrice u točki \mathbf{X} , ako ona nema degeneraciju, različita od nule ($\mathbf{H}_f \neq 0$), a u suprotnom kritična točka ima degeneraciju i nije primjenjiva za test druge derivacije.

Hesseova matrica funkcije $f(x_1, x_2, x_3)$ koja ima dvostruku parcijalnu derivaciju je definirana na slijedeći način:

$$\mathbf{H}_f \equiv \left[\frac{\partial^2 f}{\partial x_i \partial x_j} \right], \quad (i, j = 1, 2, \dots, n)$$

Kod testa druge derivacije kada je Hesseova matrica pozitivno definitna u točki \hat{X} , tada funkcija $f(x)$ zauzima u točki \hat{X} lokalni minimum. Kada je Hesseova matrica negativno definitna u točki \hat{X} , tada funkcija $f(x)$ zauzima u točki \hat{X} lokalni maksimum.

Simetrična matrica A ($n \times n$), za koju vrijedi $A = A^T$, je pozitivno definitna ukoliko za bilo koji netrivialni n -dimenzionalni vektor X ($X \neq 0$) vrijedi $X \cdot A \cdot X^T > 0$, dok je u suprotnom simetrična matrica A ($n \times n$), za koju vrijedi $A = A^T$, negativno definitna ukoliko za bilo koji netrivialni n -dimenzionalni vektor X ($X \neq 0$) vrijedi $X \cdot A \cdot X^T < 0$.

Za bolje definiranje pozitivnih i negativnih definitnih matrica se može koristiti determinantom simetrične ($n \times n$) matrice $A \equiv [a_{ij}]$:

$$A_1 \equiv +|a_{11}| \quad A_2 \equiv - \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} \quad A_3 \equiv - \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} \quad \dots \quad A_n \equiv (-1)^{n-1} \det A$$

pa je A negativno definitna u točki \hat{X} , ako i samo ako su A_1, A_2, \dots, A_n negativni; dok je A negativno semidefinitna ako su samo A_1, A_2, \dots, A_r ($r < n$) negativni, a preostali A -ovi su svi nule. Također vrijedi obratno, A je pozitivno definitna u točki \hat{X} , ako i samo ako su A_1, A_2, \dots, A_n pozitivni; dok je A pozitivno semidefinitna ako su samo A_1, A_2, \dots, A_r ($r < n$) pozitivni, a preostali A -ovi su svi nule.

Tablica 2: Zaključci na temelju testova 1. i 2. derivacije

Teoremi o ekstremima
Kada je funkcija $f(x)$ kontinuirana na zatvorenom i ograničenom intervalu (a,b) , tada $f(x)$ ima globalni optimum (min i max) na tom intervalu.
Kada funkcija $f(x)$ ima lokalni minimum (ili lokalni maksimum) u točki \hat{X} , i ako ∇f postoji u okruženju ε oko točke \hat{X} , tada je $\nabla f _{\hat{X}} = 0$
Kada funkcija $f(x)$ ima dvostruku parcijalnu derivaciju u ε okruženju točke \hat{X} , $\nabla f _{\hat{X}} = 0$ i $H_f _{\hat{X}}$ je negativno definitna, tada funkcija $f(x)$ ima lokalni minimum u \hat{X} .

3. SOFTVERSKI PAKETI ZA OPTIMIZACIJU NELINEARNIH FUNKCIJA

Budući da na tržištu postoji veliki broj softverskih paketa za rješavanje problema nelinearnog programiranja, a mnogi od njih obuhvaćaju probleme linearnog (LP), kvadratnog (QP), geometrijskog, dinamičkog, cjelobrojnog i ostalog matematičkog programiranja.

Također, uz aplikativne softverske pakete za krajnje korisnike od kojih će neki biti i opisani unutar ovog poglavlja, na internetu su dostupni i manji besplatni softveri, programske skripte, algoritmi, solveri, programske datoteke pisane u nekim od programskih jezika kao npr. C++, Object Pascal/Delphi, Java, itd., [7.] koje sadrže jezični kod i podatke pa se mogu modularno povezivati sa nezavisnim programima. Pomoću njih se mogu pisati i mijenjati postojeći algoritmi za rješavanje optimizacijskih problema. Dok su ti besplatni (free-for-research) softveri otvorenog koda, tako da se mogu mijenjati, neki dolaze u izvršnim (executable) verzijama.

U daljnjem tekstu su opisani aplikativni komercijalni softverski paketi koji koriste predodređene algoritme ovisno o vrsti optimizacijskog problema, odnosno to su optimizacijski sustavi koji koriste svoj programski jezik (sintaksu) za modeliranje i postavljanje matematičkog modela nelinearnog programiranja.

3.1. Računalna optimizacija

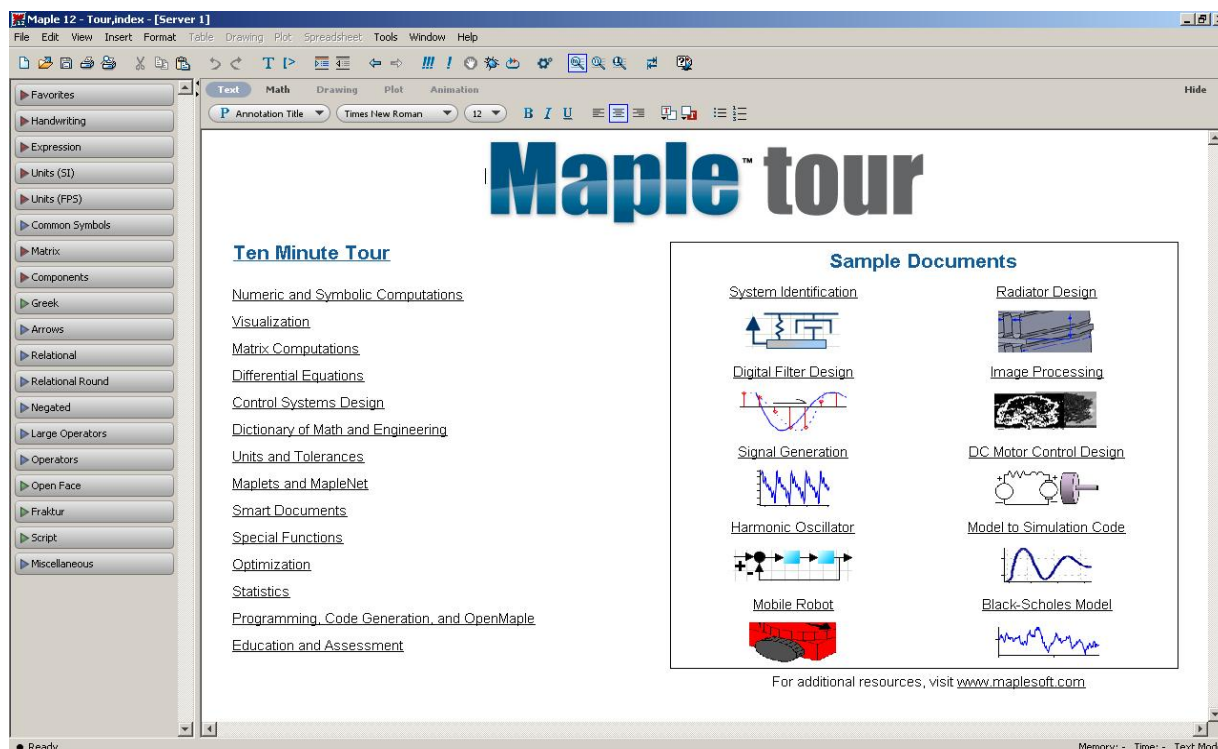
Računalna optimizacija se obavlja pomoću razvijenih algoritama koji su implementirani u različite programske jezike, a dostupni su u obliku programskih datoteka kao npr. spremište Netlib (www.netlib.org), Decision Tree for Optimization Software (<http://plato.asu.edu/sub/pns.html>), Guide to Available Mathematical Software, (<http://gams.nist.gov/>) i dr.

Komercijalni proizvodi implementiraju različite numeričke algoritme uključujući datoteke IMSL (International Mathematics and Statistics Library) i NAG (Numerical Algorithms Group), dok je besplatna alternativa GSL (GNU Scientific Library).

U popis poznatih aplikativnih softverskih paketa za računalnu optimizaciju potrebno je uključiti MATLAB, Maple, S-PLUS, LabVIEW, LINGO, Evolver, a od besplatnih alternativa FreeMat, Scilab, GNU Octave. Također, u popis se mogu uključiti i računalni algebarski sustavi kao što je Mathematica i spreadsheet (proračunska tablica) softveri namjenjeni za jednostavnije probleme npr. Evolver.

3.2. Softverski paket: Maple™

Maple je multifunkcionalni komercijalni softverski paket razvijen kao računalni algebarski sustav napisan pomoću programskog jezika C, Java, i Maple, a može se mjeriti s konkurentskim softverom Mathematica. Počeci Maple-a sežu u 1980. godinu, gdje je na Sveučilištu Waterloo u Kanadi nastao, a komercijalizirao se unutar kompanije Waterloo Maple Inc. (<http://www.maplesoft.com/products/maple/>) sa zaštitnim znakom Maplesoft 1988. godine, [10.]



Slika 11: Početni prozor unutar Maple-a

Unošenje matematičkih zapisa od strane korisnika se odvija pomoću tradicionalne matematičke notacije, kao npr. naredba za rješavanje linearne diferencijalne jednačbe sa

zadanim početnim uvjetima $dsolve\left(\left\{\frac{d^2}{dx^2}y(x) - 3y(x) = x, y(0) = 0, D(y)(0) = 2\right\}, y(x)\right)$.

The screenshot shows the Maple 12 interface with the following commands and results:

```

> restart;
> sqrt(2)=evalf(sqrt(2),21);
                                 $\sqrt{2} = 1.41421356237309504880$  (1)
> solve(3*x^2+b*x=7,x);
                                 $-\frac{1}{6}b + \frac{1}{6}\sqrt{b^2 + 84}, -\frac{1}{6}b - \frac{1}{6}\sqrt{b^2 + 84}$  (2)
> f:= x -> tan(x)*sqrt(x);
> D(f)(x);
                                 $(1 + \tan(x)^2)\sqrt{x} + \frac{1}{2}\frac{\tan(x)}{\sqrt{x}}$  (3)
> value (Int (sin(x)^2, x));
                                 $-\frac{1}{2}\sin(x)\cos(x) + \frac{1}{2}x$  (4)
> int (sin(x)^2, x = 0..Pi/2);
                                 $\frac{1}{4}\pi$  (5)
> DGL:= diff (y(x),x,x) - 3*y(x) = x;
> dsolve ({DGL, y(0)=1, D(y)(0)=2}, y(x));
                                 $y(x) = e^{\sqrt{3}x} \left( \frac{7}{18}\sqrt{3} + \frac{1}{2} \right) + e^{-\sqrt{3}x} \left( \frac{1}{2} - \frac{7}{18}\sqrt{3} \right) - \frac{1}{3}x$  (6)
> with (geom3d);
> sphere (K, [point (M, 2, 3, -2), 3/4]);
> Equation (K, [x, y, z]);
                                 $\frac{263}{16} + x^2 + y^2 + z^2 - 4x - 6y + 4z = 0$  (7)

```

Slika 12: Primjer matematičke notacije unutar Maple-a

Maple ima obuhvatnu podršku za numeričko računanje, simboličko računanje i vizualizaciju, a uključuje dinamički tip programskog jezika imperativnog stila. Također ima sučelje za ostale programske jezike kao što su: C, Fortran, Java, MATLAB i Visual Basic, a podržava i Excel.

Arhitektura Maple-a se temelji na malim jezgrama, napisanim u C-u, koje omogućuju Maple jeziku. Velik dio funkcionalnosti je ostvaren pomoću različitih programskih datoteka koje dolaze iz različitih izvora, a numerički izračuni se izvršavaju pomoću programskih datoteka NAG Numerical Libraries, ATLAS (Automatically Tuned Linear Algebra Software) i GMP (GNU Multiple-Precision Library).

3.2.1. Modularnost i podrška Maple-a

Maple koristi kombinaciju naprednog programskog jezika, numeričke matematike, manipulacije simbolima i vrhunske grafike. Zbog svoje raznolikosti, ustrojio se kao računalni algebarski sustav od izbora za mnoge korisnike računala uključujući komercijalne i sveučilišne znanstvenike, inženjere, matematičare, predavače i studente. Također je zbog svoje specifične prirode i složenosti potrebno poznavati posebnosti njegove programske sintakse da bi obavili potrebni izračuni.

Uz različite module vidljive u tablici 3., optimizacijski modul pruža podršku kroz brze algoritme za rješavanje numeričkih optimizacijskih problema linearnog, kvadratnog, nelinearnog i globalnog optimiranja.

Tablica 3: Primjer osnovnih modula Maple-a

Moduli i podrška Maple-a					
MATEMATIKA	Algebra	Osnovna matematika	Proračuni	Pretvorbe	Diferencijalne jednačbe
Diferencijalna geometrija	Diskretna matematika	Procjena	Faktorizacija i rješavanje jednačbi	Financijska matematika	Geometrija
Linearna algebra	Numerička matematika	Vektorska algebra	Optimizacija	Statistika	Tenzorska analiza
Specijalne funkcije	FIZIKA	PROGRAMIRANJE	STUDENSKI MODUL	INŽENJERSKI MODUL	GRAFIKA

Na primjer, modul za linearno programiranje uključuje algoritme za rješavanje problema sa stotinama varijabli i ograničenja. Tipovi problema koje uključuje su neprekidno, cjelobrojno, mješovito cjelobrojno i binarno-linearne programe.

Modul za nelinearno programiranje općenito daje lokalno rješenje, osim za jednovarijatne probleme gdje je dostupan globalni solver.

3.2.1.1. Osnove sintakse programskog jezika Maple

Za potpuno korištenje softverskog paketa Maple potrebno je poznavati njegovu programsku sintaksu, ali je za osnovne izračune dovoljno znati osnovne naredbe i njihove kombinacije. U tablici 4. su navedena osnovna pravila sintakse, tj. njih pet, koja su dovoljna za osnovno korištenje.

Tablica 4: Pet osnovnih pravila Maple-ove sintakse, [9.]

Osnove Maple-ove sintakse
1. Argumenti svih funkcija (onih stvorenih i onih definiranih od strane korisnika) su dani u zagradama (...). Uglate zagrade [...] se koriste za grupiranje operacija: vektori, matrice i popisi su dani u uglatim zagradama.
2. Oznaka točka-zarez (;) ili dvotočka (:) mora biti uključena na kraj svake naredbe. Maple ne ispisuje rezultat kada je dvotočka (:) na kraju naredbe. Nikada se korisnički objekt ne imenuje s istim imenom kao onaj predefiniрани od softvera.
3. Množenje se označuje s oznakom *. Zato upis $2*x*y$ označava $2xy$, koji se ne upisuje kao $2xy$.
4. Za potencije se koristi oznaka ^. Upis $(8*x^3)^{1/3}$ se koristi kod rješavanja slučaja $(8x^3)^{1/3} = 8^{1/3}(x)^{3^{1/3}} = 2x$, umjesto oznake $8*x^{1/3}$ što bi označavalo $8x/3$.
5. Maple slijedi redosljed izvođenja operacija. Zato upis $(1+x)^{1/x}$ označava $\frac{(1+x)^1}{x}$, a upis $(1+x)^{(1/x)}$ označava $(1+x)^{1/x}$. Slično, upis x^3*x označava $x^3 \cdot x = x^4$, dok izraz $x^{(3*x)}$ označava x^{3x} .

Na slici br. 13 je prikazan radni prozor Maple-a sa upisanom funkcijom cilja, odnosno nelinearnom funkcijom sa 4 varijable (w,v,x,y) i ograničenjima u obliku jedne jednadžbe i jedne nejednadžbe nakon izvršenja upisanog modela. Za optimizaciju je korišten algoritam NLPsolve za nelinearne funkcije, oblika `NLPsolve(obj, constr, bd, opts)` za unošenje u algebarskom obliku. Oznaka *obj* se odnosi na algebarski upisanu funkciju cilja s imenom upisanim prilikom modeliranja. Oznaka *constr* se odnosi na grupu zadanih ograničenja i

opcijski se unosi, dok se *bd* i *opts* unose po potrebi, odnose se na zadani raspon varijabli i na dopunska ograničenja. Plavom bojom su nakon izvršene optimizacije prikazane vrijednosti funkcije cilja i varijabli w, v, x, y . Minimalna vrijednost funkcije cilja je $\sim 1,279$ i dobije se koristeći algoritam *NLPSolve* bez posebne naznake za npr. maksimizaciju. Operator za dodjeljivanje $:=$ u maple-ovoj algebri označava dodjeljivanje imena varijablama.

$$\begin{aligned}
 &> \text{FunkcijaCilja} := w^3 \cdot (v-w)^2 + (w-x-1)^2 + (x-y-2)^2 + (y-z-3)^2 \\
 &\quad \text{FunkcijaCilja} := w^3 (v-w)^2 + (w-x-1)^2 + (x-y-2)^2 + (y-z-3)^2 \quad (1) \\
 &> \text{Ograničenje} := \{w+x+y+z \leq 5, 3z+2v-3=0\} \\
 &\quad \text{Ograničenje} := \{3z+2v-3=0, w+x+y+z \leq 5\} \quad (2) \\
 &> \text{Optimization[NLPSolve]}(\text{FunkcijaCilja}, \text{Ograničenje}) \\
 &[1.27981807009781324, [v=3.69259842437645468, w=3.66959192262816990, x \\
 &= 2.25129088086626838, y=0.540849479423198498, z=-1.46173228291763646]] \quad (3)
 \end{aligned}$$

Slika 13: Primjer postavljenog modela NLP-a i njegove optimizacije

Maple ima korisnički fleksibilno i prilagodljivo okruženje, tako da uz grupu naredbi i programski jezik ima „point-and click“ module. Naredbe su smještene u maple-ovoj bazi koja se sastoji od dvije grupe: glavna baza naredbi i dodatni paketi.

1. Glavna baza se sastoji od najčešće korištenih naredbi.
2. Dodatni paketi se sastoje od različitih specijaliziranih naredbi za obavljanje različitih proračuna, statistiku i dr.


Način unošenja naredbi i matematičkih simbola u radni okvir se odvija pomoću tekstovne 1-D Math (Text mode) i matematičke 2-D Math (Math mode) notacije. Matematička notacija je jednostavnija za unošenje i kod početnika ne zahtjeva poznavanje posebne sintakse. Simboli se odabiru iz postojećih paleta simbola, umanjuju mogućnost nastajanja pogrešaka kod upisa sintakse, a između 1-D Math i 2-D Math notacije je osigurana pretvorba na radnom okviru između jedne u drugu.

3.2.1.2. Naredbe unutar radnog lista

Radni okvir Maple-a se može koristiti kao radni list ili dokumentni list. Kod otvaranja novog radnog lista u gornjem lijevom kutu se prikazuje oznaka **>** koja označava Maple-ov odzivni znak za izvršavanje upisane naredbe. Naredbe se upisuju u nastavku tog odzivnog znaka pomoću tekstovne 1-D Math (Text mode) i matematičke 2-D Math (Math mode) notacije i izvršavaju.

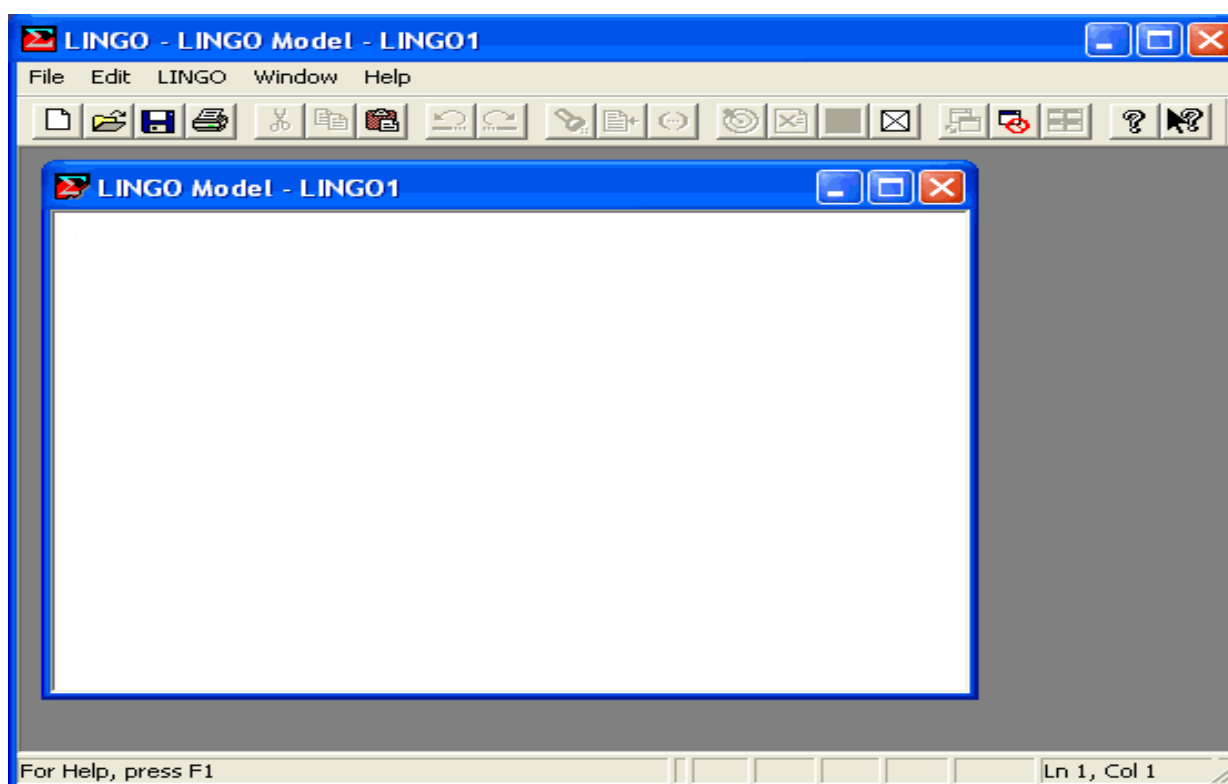
U nastavku se nalazi tablica 5. u kojoj su prikazani osnovni pojmovi vezani uz naredbe u Maple-u i njihov opis.

Tablica 5: Prikaz osnovnih pojmova vezanih uz naredbe u Maple-u

Osnovna upotreba	Opis
Maple-ov odzivni znak	>
Izvršavanje cijelog radnog lista (svih naredbi)	<ul style="list-style-type: none"> izvodi se pomoću ikone !!! na alatnoj traci pomoću izbornika uređivanje selektiranjem Execute>Worksheet
Izvršavanje jedne naredbe	Izvršava se stavljanjem kursora unutar bilo kojeg područja za unos koje se odnosi na naredbu i pritiskom Enter
Izvršavanje selektiranog područja	<ul style="list-style-type: none"> izvodi se pomoću ikone ! na alatnoj traci pomoću izbornika uređivanje selektiranjem Execute>Selection
Točka-zarez	Odvajanje naredbi. Završavanjem 1-D i 2-D naredbi sa točkom-zarez ; prikazuje se odziv. Sve 1-D naredbe moraju završiti za ; ili :
Dvotočka	Odvajanje naredbi. Završavanjem 1-D i 2-D naredbi sa točkom-zarez : ne prikazuje se odziv. Sve 1-D naredbe moraju završiti za ; ili :
restart;	Naredba restart služi za čišćenje Maple-ove unutarnje memorije. Također se nalazi u alatnoj traci u obliku ikone 

3.3. Softverski paket: LINGO 12.0

LINGO je opsežan softverski alat proizveden od američke tvrtke LINDO SYSTEMS INC sa sjedištem u Chicago-u. Namijenjen je za postavljanje i rješavanje linearnih, nelinearnih (konveksnih i nekonveksnih/globalnih), kvadratnih, stohastičkih i cjelobrojnih optimizacijskih modela. LINGO uključuje kompletno integrirani paket sa jezikom za opisivanje optimizacijskih modela, okruženje sa odlikama za postavljanje i oblikovanje problema, kao i ugrađene rješavače (solvere).



Slika 14: Glavni startni okvir LINGO-a u windowsima

Uz LINGO 12.0 od strane LINDO SYSTEMS INC na tržištu postoji softver pod imenom LINDO API 6.1 koji služi za kreiranje vlastitih optimizacijskih aplikacija. To se postiže povezivanjem LINDO solvera sa vlastitim aplikacijama i programima. Također omogućuje stohastičko programsko sučelje, statističko uzorkovanje, povezivanje sa MatLAB-om i izradu vlastitih WEB aplikacija.

Glavna svrha računalnog programa LINGO je omogućiti korisniku brzo unošenje formulacije matematičkog modela, njegovo rješavanje, procjenu točnosti formulacije modela na temelju rješenja, brzu promjenu i izmjene u postavljenom modelu i ponavljanje procesa. Također omogućuje postavljanje i rješavanje modela na temelju ugrađenih predložaka, [11.].

3.3.1. Glavne karakteristike LINGO-a

Jednostavno postavljanje modela

Jedna od glavnih odlika softvera LINGO je njegov matematički programski jezik zbog toga što omogućuje zapisivanje problema u prirodnom obliku vrlo sličnom standardnoj matematičkoj notaciji. Rezultat toga je smanjenje vremena kod izrade modela i preglednost modela linearnog, nelinearnog i cjelobrojnog problema napisanog u čitljivoj intuitivnoj formi. Zato su modeli jednostavni za postavljanje, razumijevanje i održavanje. LINGO-v jezik za modeliranje uključuje i datoteku matematičkih, statističkih i financijskih funkcija.

Prikladno korištenje podataka

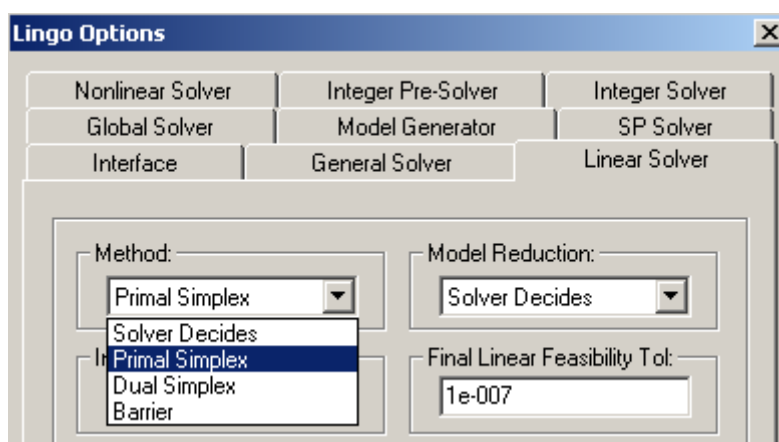
Mogućnost izrade modela koji koriste podatke direktno iz baza podataka ili proračunskih tablica (npr. Exel) omogućuje skraćanje vremena i veću jednostavnost u manipuliranju s podacima. Također moguće je izlazne podatke prebaciti u razne forme proračunskih tablica. Npr. sljedeći izraz u LINGO jeziku:

```
TROŠKOVI, KAPACITET = @OLE('KARAKTERISTIKE.XLS');
```

omogućuje prihvaćanje podataka kolone TROŠKOVI i KAPACITET unutar proračunske tablice iz Exel-a, odnosno file-a 'KARAKTERISTIKE.XLS. To se postiže pomoću funkcije @OLE (Object Linking and Embedding).

3.3.2. Integrirani rješavači (solveri)

LINGO je opremljen sa opsežnim kompletom brzih i integriranih solvera za linearno, nelinearno, kvadratno, stohastičko i cjelobrojno optimiranje, (slika 15). Njihova prednost je izostavljanje potrebe za određivanjem vrste solvera jer se oni automatski određuju na temelju formulacije postavljenih matematičkih modela.



Slika 15: Prikaz postojećih solvera unutar LINGO-a

3.3.2.1. Linearni solveri

Unutar LINGO-a postoje tri solvera za linearne modela:

- 1. Primal i dual simplex solver** čije se iteracije vrše na vanjskom rubu dozvoljenog područja, odnosno područja mogućih rješenja.
- 2. Barrier Solver** koji je opsijski solver za koji je potreban dodatna licenca jer omogućuje brže rješavanje modela s više od 5000 ograničenja ili modela s visokom degeneracijom nego Simplex solver. To se postiže jer se njegove iteracije kreću s unutarnje strane područja mogućih rješenja.
- 3. Cjelobrojni solver** (Integer Solver) za cjelobrojna ograničenja koji djeluje zajedno sa linearnim i nelinearnim solverom.

3.3.2.2. *Nelinearni solveri*

Nelinearni solveri omogućuju nalazak lokalnog ili globalnog optimuma na više načina, a njihov popis je u nastavku:

1. Opći nelinearni solver koji koristi GRG algoritam (Generalized Reduced Gradient) za rješavanje nelinearnih modela, a također je prisutan i u Microsoft Excel-u.

2. Globalni solver s pretraživanjem područja mogućih rješenja sve do pronalaska globalnog optimuma. To se postiže pretvaranjem nekonveksnih, nelinearnih problema u nekoliko konveksnih, linearnih podproblema. Zatim se koristi metoda grananja i ograđivanja (Branch and bound) kojom se iscrpno pretražuju podproblemi zbog pronalaženja globalnog rješenja. Njegova prednost pred solverima za lokalno pretraživanje je što se ne zaustavlja nailaskom na prvi lokalni optimum i daje bolja rješenja kod nekonveksnih modela.

3. Multistart Solver služi kod ograničenog vremena zbog kojeg bi traženje globalnog optimuma bilo previše dugotrajno. Zato ovaj solver nalazi „dobra“ rješenja u kratkom vremenu. To se postiže stvaranjem skupa kandidata za startnu točku pretraživanja unutar prostora mogućih rješenja. Zatim, nelinearni solver odabire podskup tih kandidata s kojima se pokreće niz lokalnih optimizacija. Za nekonveksne nelinearne modele je kvaliteta postignutog rezultata veća nego kod Općeg nelinearnog solvera.

4. Kvadratni solver detektira postojanje kvadratne strukture unutar funkcije cilja ili nekih ograničenja što mu omogućuje brže rješavanje modela nego opći nelinearni solver.

5. Solver za stohastičko programiranje uključuje rizik u višedijelne optimizacijske modele, maksimizira očekivani profit i sažima rezultate u histograme pokazujući distribuciju mogućeg profita. Glavna mu je karakteristika modeliranje i optimizacija modela s nepouzdanim elementima kroz višedijelno stohastičko linearno i nelinearno programiranje.

Na primjeru u nastavku biti će primijenjen nelinearni globalni solver.

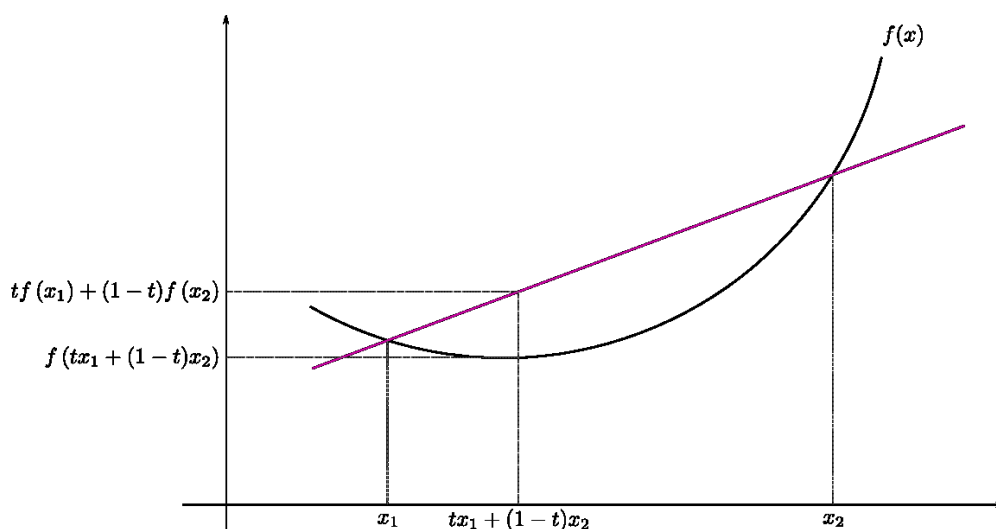
3.3.3. Nelinearni solver za globalni optimum - Global Solver

Kroz povijest su se nelinearni modeli izbjegavali iz dva razloga; prvo što je za njihovo rješavanje potrebno puno vremena, a drugo što je rezultat njihova rješavanja tradicionalnim solverima bio često lokalni optimum iako je postojalo bolje rješenje na nekoj udaljenosti. Nastajanje takovog problema je karakteristično za nekonveksne funkcije jer je kod konveksnih lokalni optimum ujedno i globalni, dok strogo konveksne funkcije imaju najviše jedan globalni optimum. Taj problem je riješio Globalni solver u LINGO-u u verzijama 8.0 i višim, koji jamči pronalazak globalnog optimuma. Potrebno ga je uključiti odabirom u opsijskom izborniku (slika 15).

3.3.3.1. Konveksnost funkcije

Svaka funkcija $f(x)$ definirana na nekom intervalu je konveksna, (slika 16), ako za svake dvije točke x_1 i x_2 , ($x_1 \neq x_2$), iz njezine domene X i svaki $t \in [0,1]$ vrijedi:

$$f(t \cdot x_1 + (1 - t) \cdot x_2) \leq t \cdot f(x_1) + (1 - t) \cdot f(x_2)$$

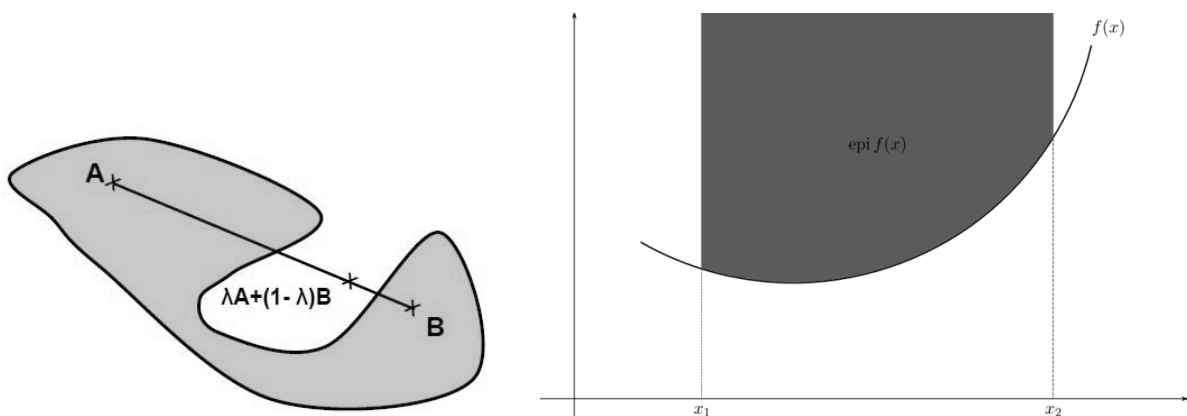


Slika 16: Konveksna funkcija na intervalu

Dok je strogo konveksna funkcija ako za svake dvije točke x_1 i x_2 , ($x_1 \neq x_2$), iz njezine domene X i svaki $t \in [0,1]$ vrijedi:

$$f(t \cdot x_1 + (1 - t) \cdot x_2) < t \cdot f(x_1) + (1 - t) \cdot f(x_2)$$

To pojednostavljeno znači da je funkcija konveksna ako je smještena „ispod“ ili na ravnom segmentu koji povezuje dvije točke na intervalu. Odnosno, ako je njezin epigraf (skup točaka iz domene s istom ili višom vrijednošću funkcije) konveksan skup. Konveksan skup je definiran kada s svaku točku A i B unutar skupa vrijedi da je i $\lambda A + (1 - \lambda)B$ također u skupu, za svaki $0 \leq \lambda \leq 1$, slika (17).



Slika 17: Nekonveksan skup i epigraf konveksne funkcije

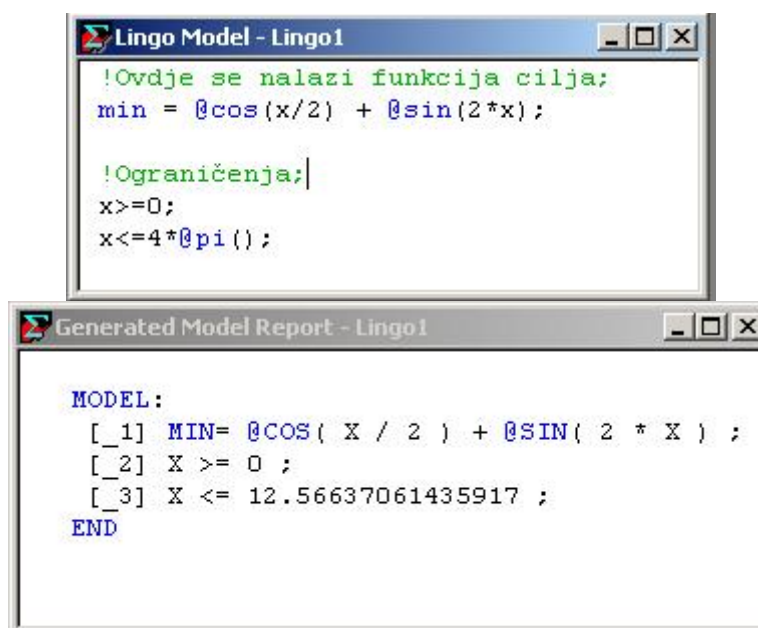
3.3.3.2. Optimiranje nekonveksne funkcije cilja

Korištenjem tradicionalnih nelinearnih softvera rješenje nekonveksne funkcije može biti jedno od lokalnih optimuma što ne treba biti globalno najbolje rješenje. Na sljedećoj nekonveksnoj nelinearnoj funkciji traži se minimum, a korišten je Globalni solver za nalaženje globalnog optimuma, unutar danih ograničenja:


$$f(x) = \cos(2/x) + \sin(2 \cdot x) \rightarrow \min, x \in (0, 4\pi)$$

Pomoću pravila LINGO-ve sintakse unosi se gornja funkcija i glavni prozor s mogućnošću upisivanja komentara koji softver ignorira (otvorenih s simbolom ! , a zatvorenih pomoću ;) , a služe za bolje snalaženje tokom kasnijeg pregledavanja formuliranog modela. Oznaka @ služi za definiranje matematičkih, finansijskih i ostalih LINGO-vih funkcija, [12.].

Nakon formuliranog modela u glavnom prozoru generira se izvješće o modelu odabirom LINGO→Generate→Display model ili naredbom Crtl+G, slika(18).



Slika 18: Prozori postavljenog modela i generiranje izvješća o modelu

Nakon postavljenog modela optimiranje se obavlja odabirom *Solve* ili () na alatnoj traci. Uz prethodno postavljen globalni solver rezultat je globalni minimum, $x = 5.543058$, a $f(x) = -1.928208$, a izvješće (Solution Report) je slijedeće:

Global optimal solution found.

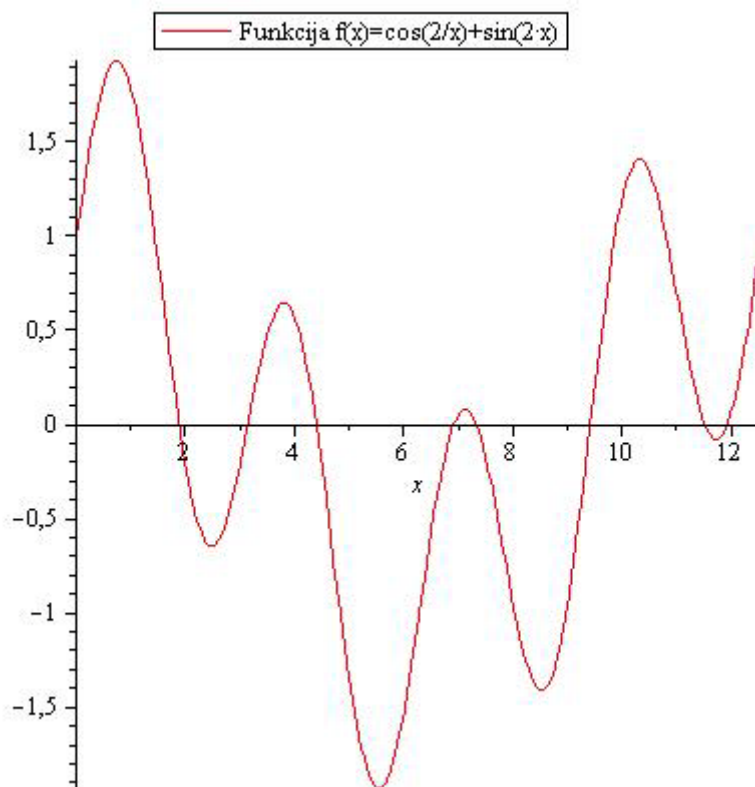
Objective value: -1.928208

Total solver iterations: 101

Model Class: NLP

Variable	Value
X	5.543058

Donji graf prikazuje optimiranu funkciju s optimalnom točkom minimuma u $x = 5.543058$, (slika 19). Iz nje je vidljivo da je optimum ujedno i globalni optimum, a ne neki od lokalnih, zbog korištenja globalnog solvera.

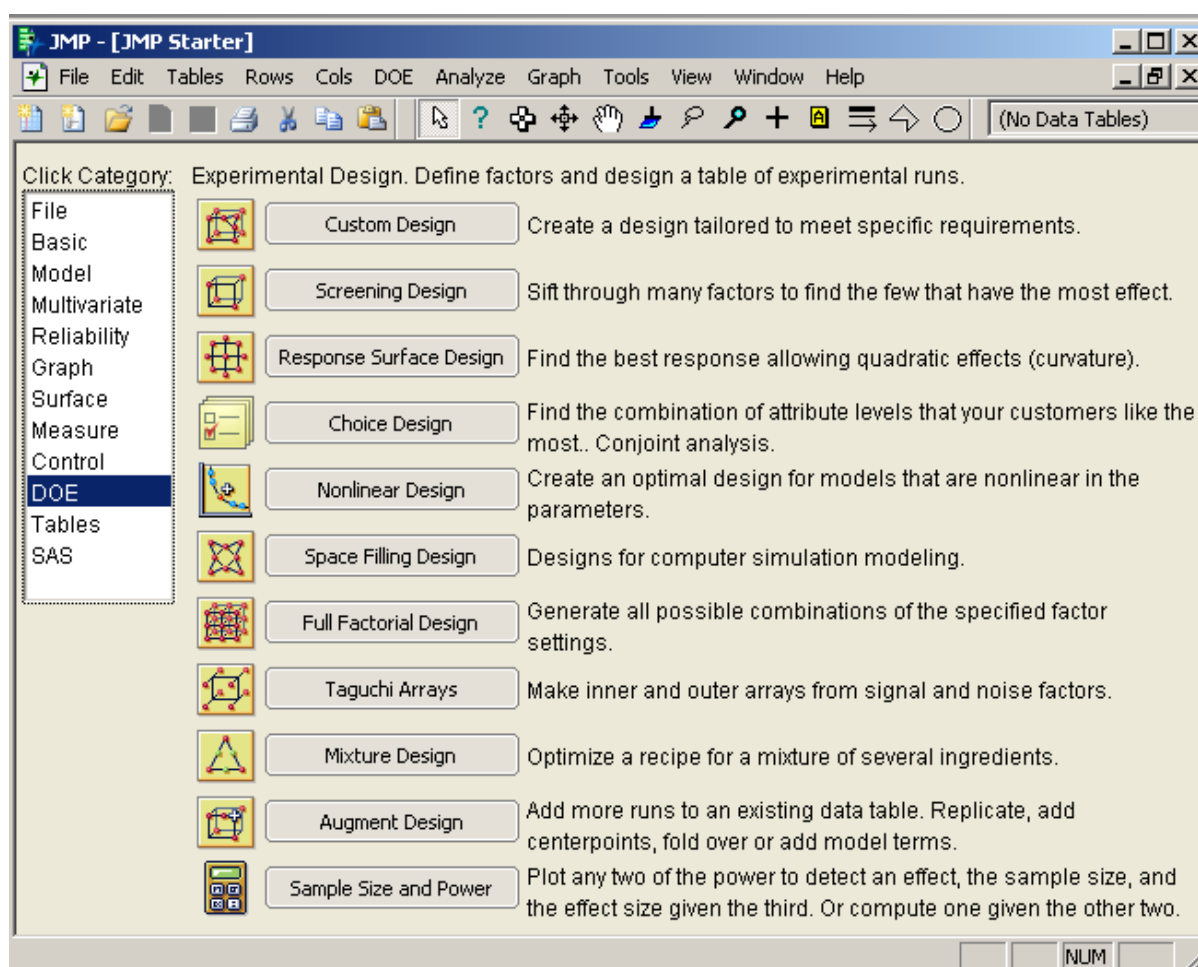


Slika 19: Graf funkcije $f(x) = \cos(2/x) + \sin(2 \cdot x)$

3.3. Softverski paket: JMP®8

JMP, odnosno izgovara se „jump“, je softver za statističku analizu i obradu podataka. Razvijen je od strane SAS-a, a djeluje kao nezavisna aplikacija koja nije uključena u softver SAS (Statistical Analysis System), koja omogućuje dinamičnu grafičku vizualizaciju i obradu podataka. Zbog dozvoljenih 10^9 stupaca i nepostojanja ograničenja za redove JMP omogućuje obradu i interpretaciju ogromnih količina podataka iz realnih znanstvenih i inženjerskih problema širom svijeta, [15].

Uz različite vrste statističkih mogućnosti i alata softvera JMP pokrivena opsežnim priručnikom [13], tema daljnjeg teksta i primjera je modul DOE (Design of Experiments), odnosno dio aplikacije zadužen za planiranje pokusa, (slika 20).



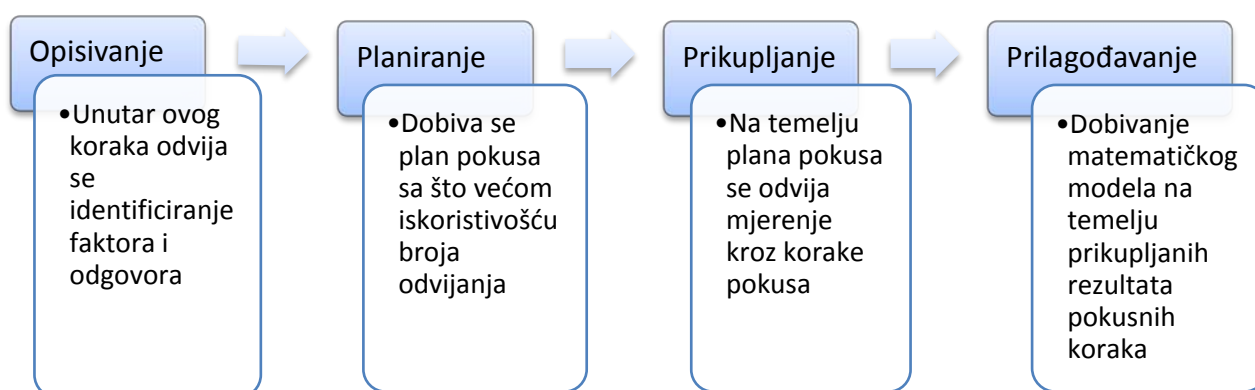
Slika 20: JMP-ov polazni okvir i opcije unutar modula DOE

Glavna svrha uvrštavanja programa JMP u ovo poglavlje je njegov modul za planiranje pokusa zbog dobivanja i optimiranja nelinearne funkcije cilja provođenjem pokusa, a ne optimiranje direktnim unošenjem poznatog nelinearnog optimizacijskog problema za što softver nije predviđen. Odlike koje su doprinijele njegovom korištenju u nadolazećim primjerima su intuitivnost, vizualnost i jednostavnost prilikom korištenja.

3.3.1. Modul za planiranje pokusa

SAS-ov softverski paket JMP je posebno prilagođen za planiranje i analizu planova pokusa. Kroz „point-and-click“ način rada osigurava razne alate za planiranje, obradu i analizu eksperimentalnih podataka. Glavne etape u modulu za planiranje pokusa su izrada plana pokusa, prilagođavanje rezultata matematičkom modelu i interpretacija podataka, (slika 21).

Budući da su produktivnost i sve veći zahtjevi u kvaliteti važne težnje u svakom poslovanju, razvijaju se metode za njihovo poboljšavanje. One su se mijenjale od vremenski dugotrajnih i skupih metoda pokušaja i pogreške, do efikasnih i jeftinih statističkih metoda koje su prisutne u JMP softveru. Planiranje pokusa u JMP-u je fokusirano na faktore, odgovore, modele i korake eksperimentiranja, dok je glavni cilj naći utjecaj faktora na odgovor, odnosno rezultat, [14].



Slika 21: Glave etape provođenja pokusa unutar JMP-a

3.3.2. Pokusi odzivnih površina

Zbog potreba dobivanja nelinearnog modela, odnosno funkcija cilja koje opisuju procese iz narednih primjera korišteni su planovi pokusa odzivnih površina korištenjem JMP softvera. Oni tradicionalno uključuju mali broj kontinuiranih faktora (najčešće 2 do 8). Postulirani, tj. *a priori* model za odzivnu površinu u pokusu je obično kvadratni (odzivna jednakost drugog reda) donjeg općeg oblika, s pripadajućim oznakama varijabli, (tablica 6).

$$y = a_0 + \sum a_i x_i + \sum a_{ij} x_i x_j + \sum a_{ii} x_i^2 + \dots$$

Tablica 6: Elementi općeg postuliranog matematičkog modela

Varijable	Opis varijable
Y	- odziv, odnosno interesna mjera izvršitelja pokusa
x_i	- razina faktora i
x_j	- razina faktora j
a_0, a_i, a_{ij}, a_{ii}	- polinomni koeficijenti

Kvadratni *a priori* matematički model za dva faktora je sljedećeg oblika:

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_{12} x_1 x_2 + a_{11} x_1^2 + a_{22} x_2^2$$

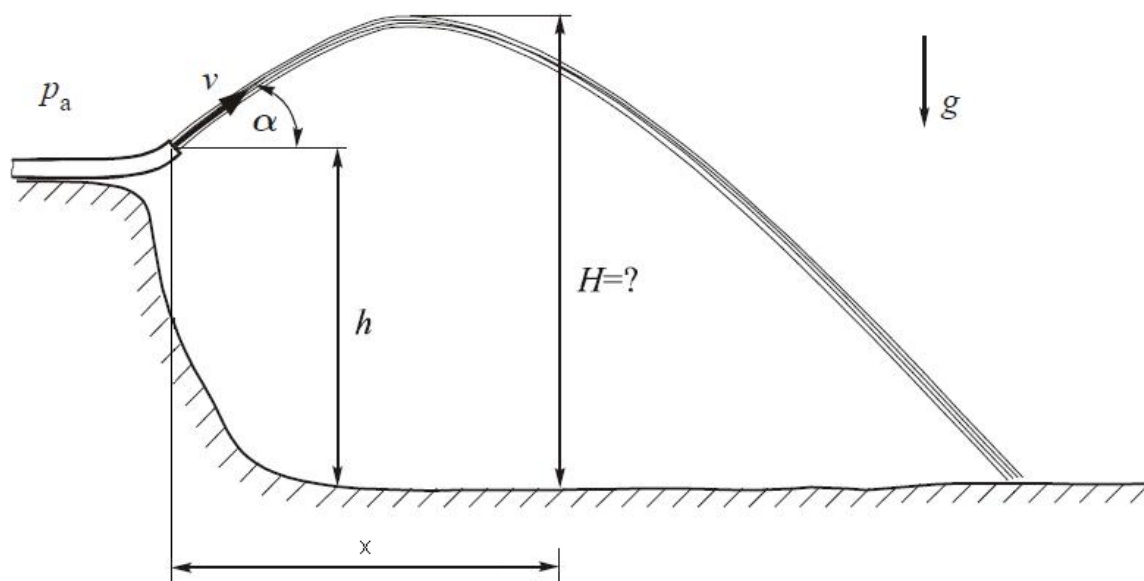
dok je kod tri faktora model oblika:

$$y = a_0 + a_1 x_1 + a_2 x_2 + a_3 x_3 + a_{12} x_1 x_2 + a_{13} x_1 x_3 + a_{23} x_2 x_3 + a_{11} x_1^2 + a_{22} x_2^2 + a_{33} x_3^2$$

Suprotno od planova pokusa odabiranja (eng. Screening Designs) unutar JMP-a, pokusi odzivnih površina (eng. response surface experiments) se provode kada su već poznati utjecajni faktori. Glavni njihov cilj je stvoriti kvalitetan model koji osigurava dobru vezu između faktora koji se promatraju i traženog odgovora. Tako dobiveni model za predviđanje se koristi s ciljem dobivanja boljih operacijskih parametara procesa koji se promatra, odnosno njihovog optimuma. Planovi pokusa odzivnih površina koje JMP podržava su Kompozitni i Box-Behnken planovi.

4. OPTIMIZACIJA U MAPLE-u

U daljnjem tekstu ovog poglavlja biti će prikazan realni primjer i njegovo rješavanje korištenjem Maple-a. Prije same optimizacije izvršeno je postavljanje matematičkog modela kroz izvod iz zadanih podataka. Primjer se temelji na izračunu maksimalne visine i pripadajuće udaljenosti koju dosegne mlaz fluida istjecanjem iz otvora cijevi pod određenim kutom pri stalnoj brzini.



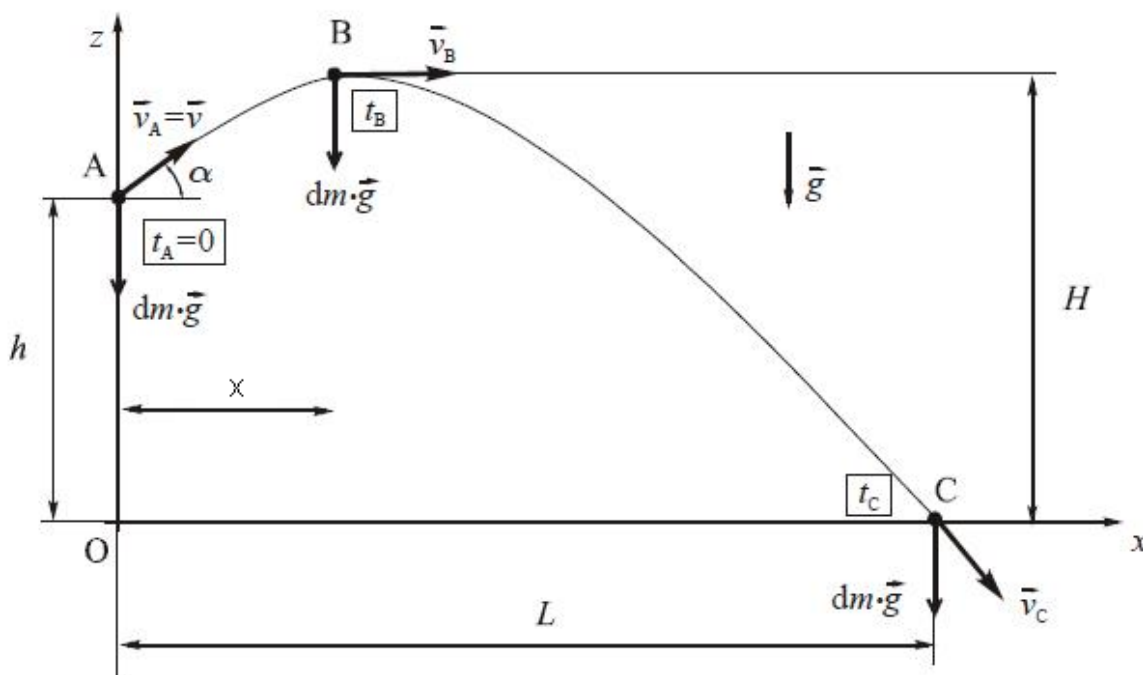
Slika 22: Istjecanje mlaza fluida iz otvora cijevi

4.1. Opis primjera za testiranje softvera Maple

Mlaz fluida prema slici 14., je sa svih strana okružen zrakom pod konstantnim atmosferskim tlakom, tako da je rezultirajuća sila tlaka na svaku česticu fluida u mlazu jednaka nuli, te od vanjskih sila ostaje samo sila težine. Na visini h se nalazi otvor cijevi iz koje izlazi mlaz fluida stalnom brzinom v , pod kutom α . Svakačestica fluida u mlazu se giba poput materijalne točke u polju gravitacije (kosi hitac).

Budući da svaka čestica u izlaznom mlazu ima brzinu v , dovoljno je promatrati

gibanje jedne čestice fluida, a gibanje svih ostalih čestica je potpuno identično. Zato je oblik mlaza jednak obliku putanje koju opisuje jedna materijalna točka izbačena brzinom v pod kutom α s visine h u odnosu na koordinatni sustav prikazan na slici 15.



Slika 23: Putanja čestice fluida

Uz pretpostavku idealnog fluida i uz zanemarivanje trenja između zraka i fluida, potrebno je koristeći Maple odrediti maksimalnu visinu H i duljinu x koja se u tom trenutku postigne, s obzirom na dane podatke u tablici 6.

Tablica 7: Zadani podaci u primjeru istjecanja fluida

Oznaka	Opis	Iznos i dimenzija
V	Brzina izlaznog mlaza	8 m/s
h	Visina otvora cijevi	1 m
α	Kut istjecanja fluida	49°
g	Gravitacijska konstanta	9,81 m/s ²
P_a	Iznos tlaka na česticu	$1 \cdot 10^5 Pa = \text{konst.}$

4.2. Izvod matematičkog modela

Gibanje se rastavlja u smjeru osi x i osi z , a u svakom trenutku na česticu fluida djeluje samo sila težine $d\vec{F} = dm \cdot \vec{g}$, jer je rezultantna sila tlaka na česticu jednaka nuli.

Primjenjujući II. Newtonov zakon $d\vec{F} = dm \cdot \vec{a}$, iz sljedećeg izraza:

$dm \cdot (ax\vec{i} + ay\vec{j} + az\vec{k}) = dm \cdot (-g) \cdot \vec{k}$ proizlazi da je $a_z = -g$, $a_x = 0$ i $a_y = 0$, a negativan smjer $-g$ ovisi o položaju koordinatnog sustava.

Prikazom akceleracije kao druge derivacije položaja dobivaju se sljedeće jednakosti:

$ax = \ddot{x} = \frac{d^2x}{dt^2} = 0$, $az = \ddot{z} = \frac{d^2z}{dt^2} = -g$, pa se integriranjem u vremenu $\int_0^t dt$ ovih jednakosti dobiva:

$$\dot{x} = \frac{dx}{dt} = vx = C_1, \quad \dot{z} = \frac{dz}{dt} = vz = -g \cdot t + C_2.$$

Konstante integracije C_1 i C_2 se određuju iz početnih uvjeta u kojem vremenski trenutak $t=0$ odgovara trenutku nailaska čestice fluida u točku A (slika 15.) u kojoj su komponente brzine:

$vx = v \cdot \cos\alpha$, $vz = v \cdot \sin\alpha$ u trenutku $t=0$, pa se uvrštavanjem ovih izraza u gornje jednakosti dobiva: $C_1 = v \cdot \cos\alpha = vx$ i $C_2 = v \cdot \sin\alpha = vz$.

Iz izraženih konstanti integracija se nakon njihovog uvrštavanja dobivaju izrazi za komponente vektora brzine u x i z osi u svakom trenutku t :

$$\frac{dx}{dt} = vx = \cos\alpha \cdot v, \quad \frac{dz}{dt} = vz = -g \cdot t + \sin\alpha \cdot v$$

Integriranjem gornjih jednakosti u vremenu $\int_0^t dt$, dobije se promjena putanje čestice fluida u vremenu, pa slijede sljedeći izrazi:

$$x = \int_0^t v \cdot \cos\alpha dt = v \cdot \cos\alpha \cdot t + C_3,$$

$$z = \int_0^t (v \cdot \sin\alpha - g \cdot t) dt = \int_0^t v \cdot \sin\alpha \cdot dt - \int_0^t g \cdot t \cdot dt = v \cdot \sin\alpha \cdot t - \frac{1}{2} t^2 \cdot g + C_4$$

Konstante integracije C_3 i C_4 se ponovno dobiju iz početnih uvjeta, tj. u trenutku $t=0$ čestica fluida se nalazi u točki A s koordinatama $x = x_A = 0$ i $z = z_A = h$, što uvrštenjem u gornje jednakosti daje $C_3 = 0$ i $C_4 = h$, odnosno sljedeće izraze:

$$x = v \cdot \cos\alpha \cdot t$$

$$z = v \cdot \sin\alpha \cdot t - \frac{1}{2} t^2 \cdot g + h, \text{ koji definiraju položaj čestice za svaki trenutak } t.$$

Izlučivanjem iz izraza za položaj x čestice t , i uvrštavanjem dobivenog izraza u funkciju položaja z u ovisnosti o vremenu t , dobivaju se sljedeće relacije:

$$t = \frac{x}{v \cdot \cos\alpha}$$

$$z = v \cdot \sin\alpha \cdot \frac{x}{v \cdot \cos\alpha} - \frac{1}{2} \cdot \left(\frac{x}{v \cdot \cos\alpha} \right)^2 \cdot g + h, \text{ ova jednakost je funkcija ovisnosti položaja } z \text{ o}$$

položaju x i koristit će se kao funkcija cilja nelinearnog programiranja u Maple-u.

4.2.1. Definiranje matematičkog modela nelinearnog programiranja

Na temelju prethodnog izvoda relacija koje definiraju položaj z čestice fluida u odnosu na položaj x i u odnosu na proteklo vrijeme t , potrebno je definirati matematički model nelinearnog programiranja. To se postiže pomoću funkcije cilja i ograničenja.

Cilj ovog primjera je odrediti maksimalnu visinu H i duljinu x koja se u tom trenutku postigne, s obzirom na dane podatke u tablici 6.

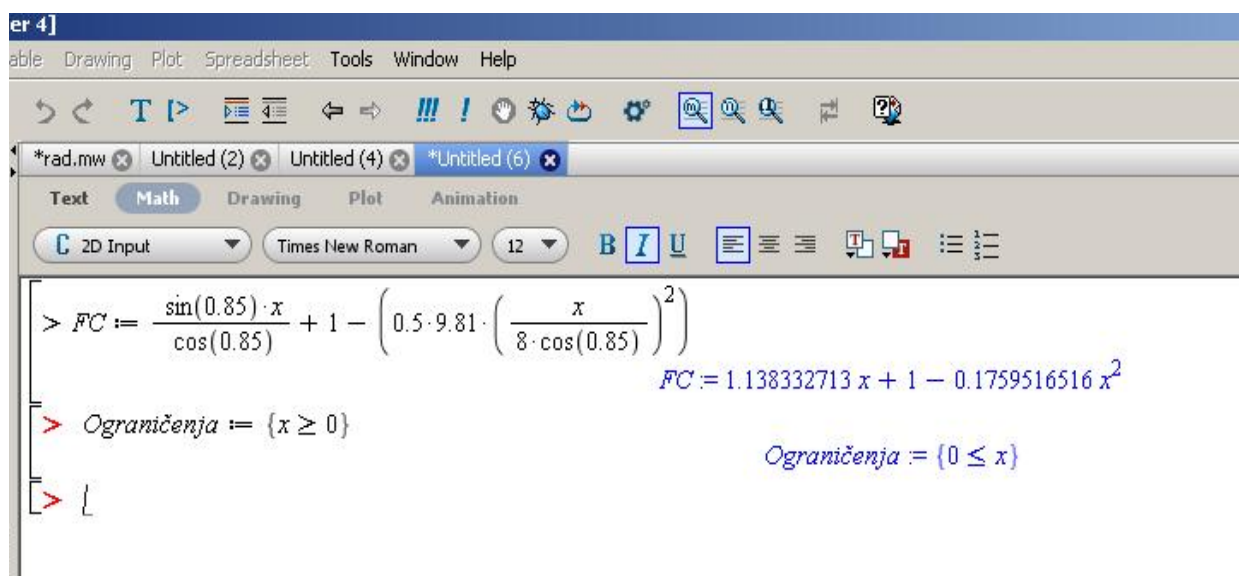
Za funkciju cilja se koristi funkcija jedne zavisne varijable z o položaju x , odnosno $\max(z) = \max f(x)$, gdje x može biti definiran na cijelom intervalu pozitivnih vrijednosti. U tablici 7. se nalazi formulirani matematički model NP-a.

Tablica 8: Formulirani model nelinearnog programiranja

Matematički model	
Funkcija cilja $f(x)$	$v \cdot \sin\alpha \cdot \frac{x}{v \cdot \cos\alpha} - \frac{1}{2} \cdot \left(\frac{x}{v \cdot \cos\alpha}\right)^2 \cdot g + h$
Ograničenja $g(x)$	$x \in [0, +\infty]$

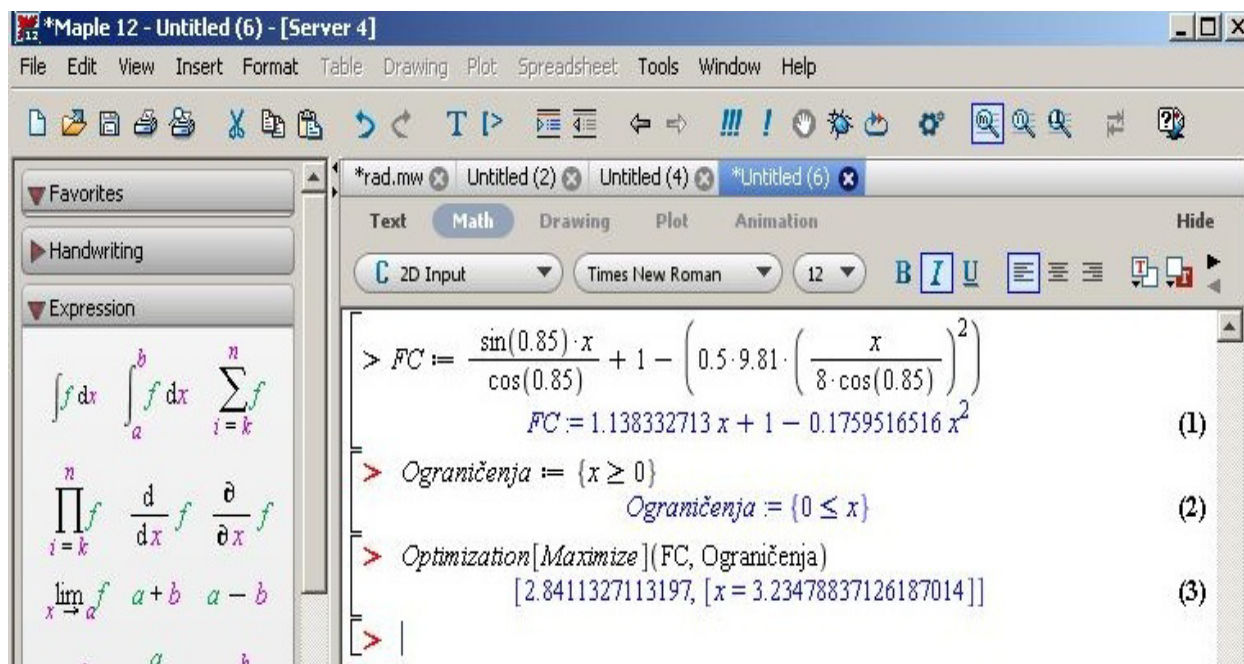
4.3. Postavljanje i rješavanje primjera

Pomoću gornjeg postavljenog matematičkog modela NP-a, potrebno je unijeti pripadajuću funkciju cilja i ograničenja u radni list Maple-a. To se postiže otvaranjem novog prozora u obliku radnog lista (Worksheet Mode) gdje se pojavljuje Maple-ov odzivni znak $>$ i unošenjem $f(x)$ u obliku matematičke 2-D Math (Math mode) notacije, slika 16. Nakon upisanih izraza izvršavanje cijelog radnog lista (svih naredbi) se izvodi se pomoću ikone !!! na alatnoj traci.



Slika 24: Prikaz unesenog matematičkog modela u Maple

Nakon izvršenja upisanog modela upisuje se naredba za optimizaciju *Optimization[Maximize](FC, Ograničenja)*, gdje je korišten algoritam Maximize za nelinearne funkcije, oblika Maximize(**obj**, **constr**). Oznaka *obj* se odnosi na algebarski upisanu funkciju cilja s imenom upisanim prilikom modeliranja. Oznaka *constr* se odnosi na grupu zadanih ograničenja i opcijski se unosi, slika 17.



Slika 25: Unos naredbe za optimizaciju Maximize

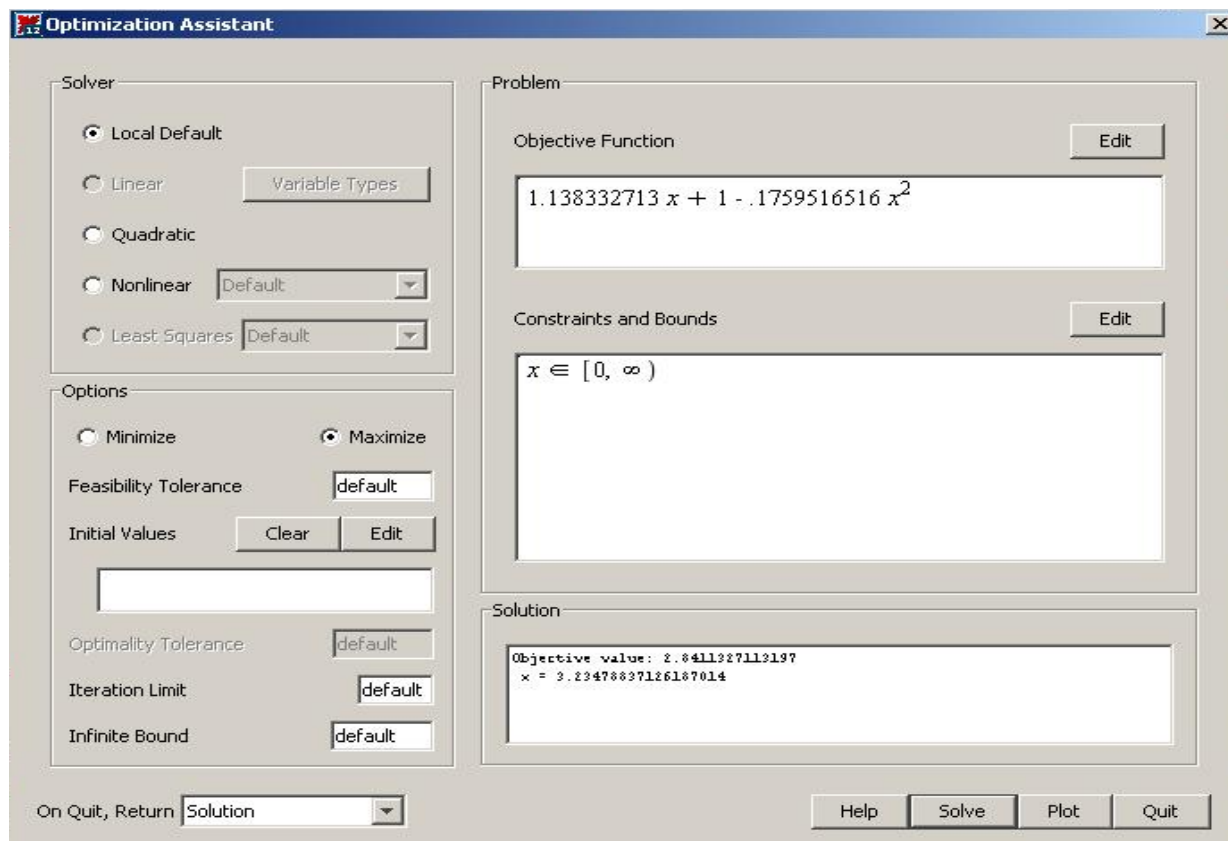
Optimizacija se također u Maple-u može provesti korištenjem modula Optimizacijski asistent (Optimization Assistant) pomoću odabira tj. desnim klikom na upisanu funkciju cilja. Unutar polja za uređivanje ograničenja (Edit Constraints and Bounds) definira se zadano ograničenje $x \in [0, +\infty]$, tako da se odabere opcija zauzimanja pozitivnih vrijednosti za varijable ograničenja.

Također se modulu Optimizacijski asistent može pristupiti pomoću sljedeće naredbe:

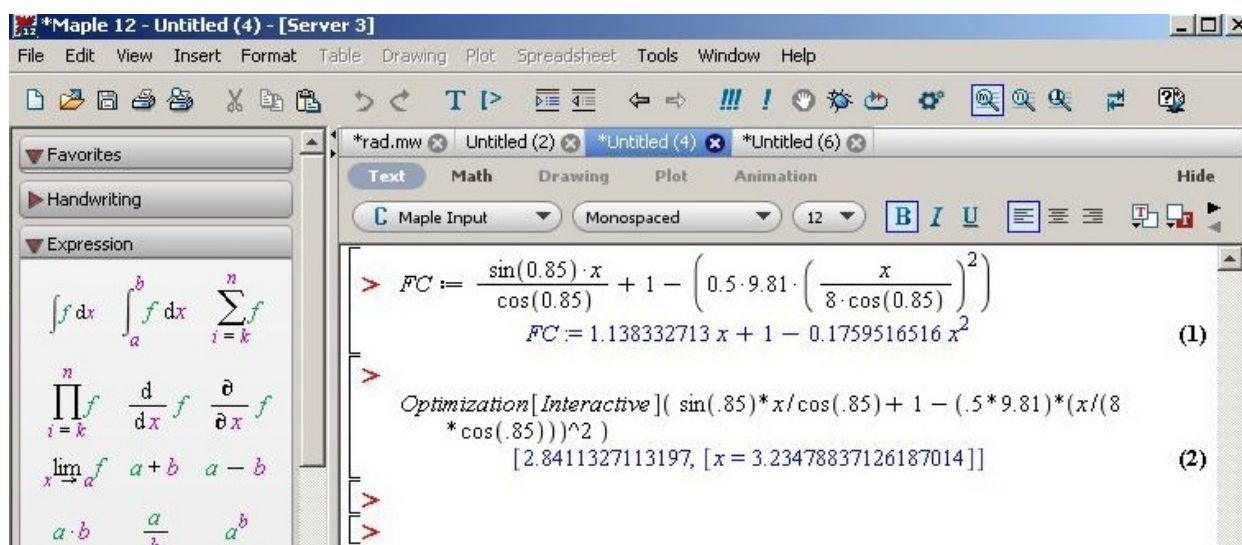
```
Optimization[Interactive]( sin(.85)*x/cos(.85) + 1 - (.5*9.81)*(x/
(8*cos(.85)))^2 )
```

, odnosno upisom Optimization[Interactive] i pripadajuće funkcije cilja u zagradi.

Na sljedećim slikama 18. i 19. je dan prikaz Optimizacijskog asistenta i pripadajućeg rješenja u radnom prozoru Maple-a.



Slika 26: Optimizacijski asistent



Slika 27: Rješenje dobiveno pomoću Optimizacijskog asistenta

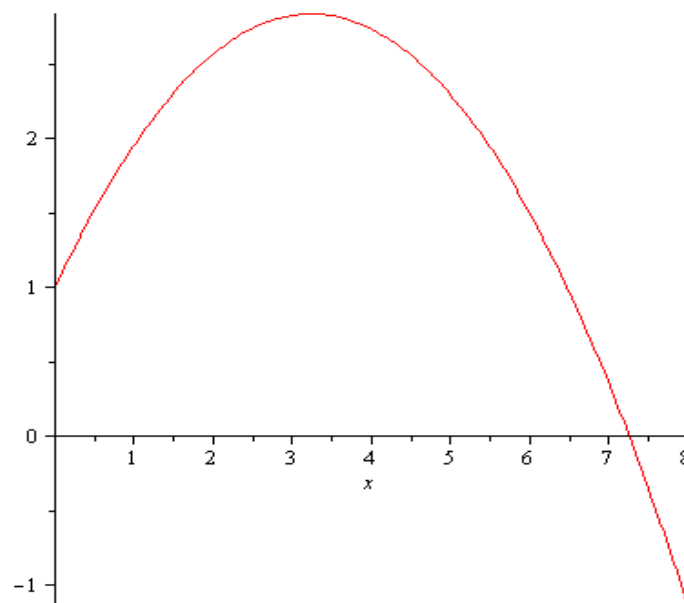
4.3.1. Optimalno rješenje i interpretacija

Nakon obavljenog izvršenja optimizacije pomoću zadanih podataka iz tablice 6. dobiva se maksimum funkcije cilja, a to je maksimalna visina koju postiže mlaz fluida, odnosno $\max FC = H = 2,84$ m što se postiže kod udaljenosti $x = 3,23$ m vidljivo iz slike 17. i 15. u točki B.

Očito je da će čestica fluida u točki B imati komponentu brzine u smjeru osi z jednaku nula i da je odgovarajući interval za ograničenja $x \in [0, +\infty]$.

4.4. Grafički prikaz funkcije cilja

Korištenjem naredbe `plot`, upisivanjem funkcije cilja i zadavanjem intervala za x koordinatu na grafičkom prikazu moguće je vidjeti putanju čestice fluida. Upisana naredba izgleda ovako: $\text{plot}\left(\frac{\sin(0.85) \cdot x}{\cos(0.85)} + 1 - \left(0.5 \cdot 9.81 \cdot \left(\frac{x}{8 \cdot \cos(0.85)}\right)^2\right), x = 0 .. 8\right)$, a raspon je u intervalu od 0 do 8 po x koordinatnoj osi. Izvršenjem naredbe se dobiva sljedeći graf:

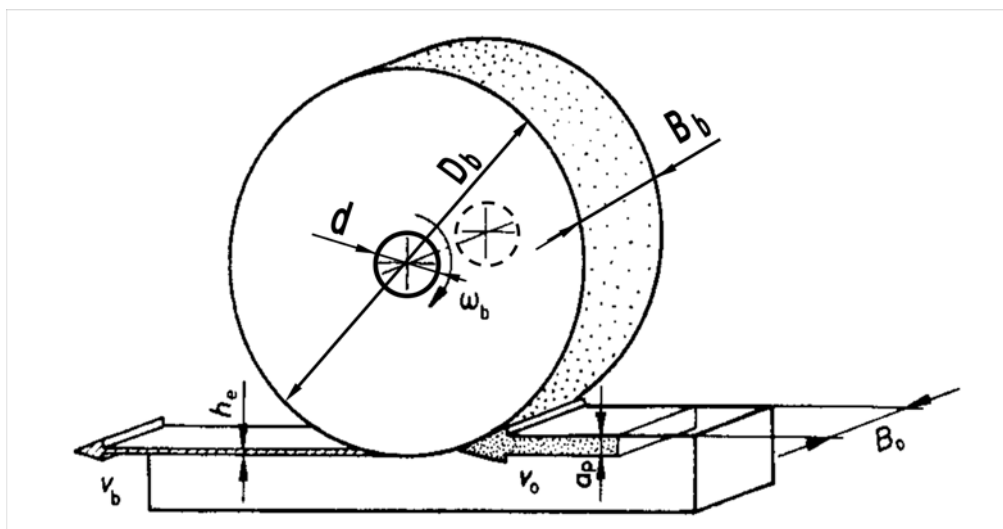


Slika 28: Graf putanje čestice fluida dobiven pomoću funkcije `plot`

5. OPTIMIZACIJA PARAMETARA BRUŠENJA KORIŠTENJEM JMP-a

U ovom poglavlju biti će prikazana optimizacija dvaju parametara procesa obodnog brušenja ravne površine, istosmjernim uzdužnim načinom (slika 29), temeljem podataka ispitivanja iz literature [16]. Faktori koji se promatraju su glavno gibanje uzrokovano rotacijom alata, odnosno tangencijalna brzina v_g (m/s) i pomoćno gibanje uzrokovano pomicanjem alata, odnosno brzina pomicanja v_p (m/min).

Težište nadolazećeg teksta je na pronalaženju optimalnih postavki navedenih parametara u pogledu kvalitete površine završne obrade, koja se očituje kroz parametar hrapavosti profila površine, odnosno amplitudni parametar R_a (μm) koji označava srednje aritmetičko odstupanje. Također, drugi element kvalitete površine koja se optimizira je uzdužni parametar, odnosno broj vrhova elemenata profila (eng. peak count) P_c (broj elemenata vrhova profila) / (dužina ocjenjivanja), tj (vrh/cm).



Slika 29: Orodno brušenje ravne površine, istosmjernim uzdužnim načinom

Na temelju metodologije odzivne površine (RMS – response surface methodology), u nastavku je prikazano pronalaženje utjecaja iznosa faktora na tražene odzive kroz nalaženje nelinearnih modela drugog stupnja $y_1 = f_1(x_1, x_2)$ i $y_2 = f_2(x_1, x_2)$ koji njihov utjecaj objašnjavaju, a time omogućuju i optimizaciju.

Pronalazak adekvatnog matematičkog modela koji postiže zadovoljavajuću aproksimaciju prave funkcionalne veze između traženog odziva i grupe nezavisnih varijabli započinje postuliranjem *a priori* linearnog, odnosno modela prvog reda oblika

$$y = a_0 + a_1 X_1 + a_2 X_2 + a_{12} X_1 X_2$$

Nakon provedenog pokusa zasnovanom na faktorskom planu i analizom aproksimacije stvarnih vrijednosti i onih dobivenih modelom, utvrđena je neadekvatnost modela prvog reda zbog postojanja zakrivljenosti u sustavu. Zato je izvršen prelazak na nelinearni model, odnosno model drugog reda i kompozitno planiranje pokusa.

Polinomni koeficijenti, (tablica 6), se dobivaju iz sustava od n jednadžbi (za n provođenja pokusa) i p nepoznanica (za p koeficijenata u *a priori* modelu). Sustav se može napisati u matričnom obliku kao

$$\mathbf{y} = \mathbf{X} \mathbf{a} + \mathbf{e}$$

gdje je \mathbf{y} je odzivni vektor, \mathbf{X} je matrica modela ili planska matrica koja ovisi o odabranim pokusnim točkama korištenim u planu pokusa i postuliranom modelu. Matrice \mathbf{a} i \mathbf{e} su matrice koeficijenata i odstupanja.

Općenito, sustav jednadžbi nije moguće jednostavno riješiti jer je broj jednadžbi manji od broja nepoznanica, odnosno postoji n jednadžbi i $n + p$ nepoznanica. Zato se za pronalaženje rješenja, odnosno modela, koristi posebna matrična metoda utemeljena na kriteriju najmanjih kvadrata, pa su takvi rezultati procjene koeficijenata prikazani oznakom $\hat{\mathbf{a}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y}$, gdje je \mathbf{X}^T transponat od \mathbf{X} . Ovaj proračun biti će prepušten softveru JMP, koji će poslužiti za direktno dobivanje traženih koeficijenata.

Proces brušenja koji se ispituje i optimizira je završno brušenje fiksne debljine materijala od 0,02 mm, s stalnom i ostalim parametrima kao što su obradak, rezna površina brusa i uvjeti brušenja. Tehnološki i ekonomski opravdani rasponi promatranih faktora tangencijalne brzine v_g (m/s) nalaze se između 15 i 25, a brzine pomicanja v_p (m/min) između 0,9 i 2,4.

5.1. Priprema za faktorsko planiranje pokusa

Na temelju definiranog cilja optimiranja i pokusa iz prethodnog teksta tablice br. 9 i 10. donose pregled faktora i odziva ispitivanja. Početni plan pokusa koji se koristi je potpuni faktorski 2^2 gdje indeks predstavlja broj promatranih faktora, a baza broj razina koje svaki faktor zauzima.

Tablica 9: Domena faktorskog plana pokusa

Faktor	Oznaka	-1 razina	0 razina	+1 razina
tangencijalna brzina v_g (m/s)	x_1	15	20	25
brzina pomicanja v_p (m/min)	x_2	0,9	1,65	2,4

Tablica 10: Odzivi traženih matematičkih modela

Odziv	Oznaka
parametar hrapavosti profila površine, odnosno amplitudni parametar R_a (μm) koji označava srednje aritmetičko odstupanje	Y_{1i}
uzdužni parametar, odnosno broj vrhova elemenata profila (eng. peak count) P_c (broj elemenata vrhova profila) / (dužina ocjenjivanja), tj (vrh/cm)	Y_{2i}

5.1.1. Faktori i provedba centriranja i ujednačavanja mjere

Zbog premještanja krajnjih vrijednosti faktora x_1 i x_2 na razine -1 i +1 zbog ujednačavanja i jednostavnosti prikazivanja plana pokusa neovisno o rasponu i mjernim jedinicama pojedinih faktora, odvija se centriranje i skaliranje faktora. Nove uvedene varijable se nazivaju kodirane varijable, [16].

Kodirane varijable su rezultat omjera dvije vrijednosti fizikalnih veličina pa nemaju mjernu jedinicu i bezdimenzijske su veličine što omogućuje svim faktorima isti interval varijacije i laku međusobnu usporedbu efekata faktora.

Pretvaranje originalnih vrijednosti varijabli V u kodirane varijable x odvija se pomoću sljedeće jednakosti, gdje je V_0 fizikalna centralna vrijednost:

$$x = \frac{V - V_0}{\text{korak}}, \quad \text{korak} = \frac{V_{+1} - V_{-1}}{2}$$

Korak faktora x_1 je $\frac{25-15}{2} = 5$, dok je $\text{korak}_2 = \frac{2,4-0,9}{2} = 0,75$.

Parametar hrapavosti R_a ima standardnu devijaciju mjerenja $0,002 \mu\text{m}$, dok je standardna devijacija broja vrhova elemenata profila P_c 2 vrha/cm, a koristit će se prilikom ispitivanja kvalitete linearnog modela dobivenog faktorskim planom pokusa.



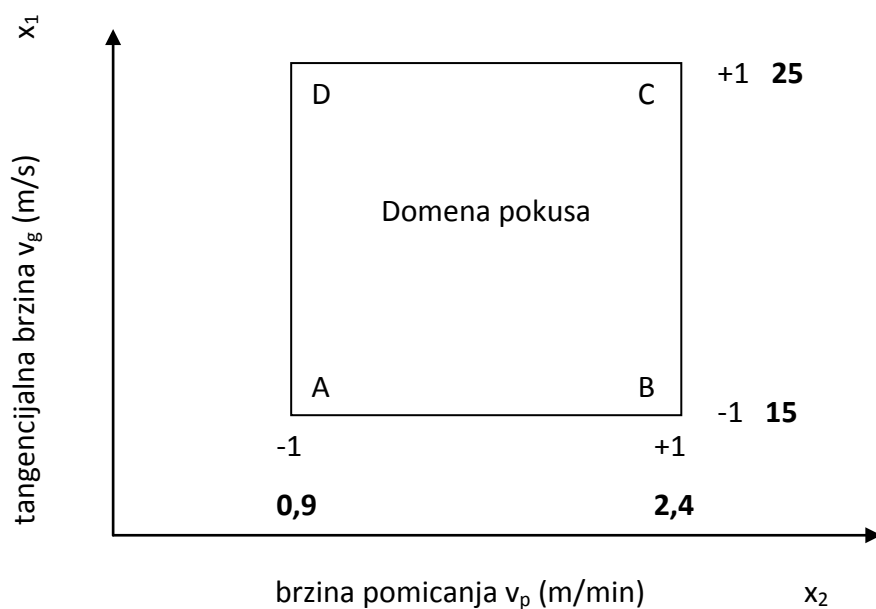
Slika 30: Definiranje parametra P_c , odnosno broja vrhova elementa profila, [18]

5.2. Planiranje i provođenje faktorskog pokusa

Prvi odabrani plan pokusa je potpuni faktorski 2^2 s linearnom modelom oblika $y = a_0 + a_1 X_1 + a_2 X_2 + a_{12} X_1 X_2$.

Zbog utvrđivanja adekvatnosti modela koji će biti dobiven, odabiru se dvije kontrolne točke (replikanta) u centru domene pokusa.

Slika 31. prikazuje potpuni faktorski 2^2 plan pokusa, odnosno domenu ispitivanja s točkama kombinacija faktora za izvršavanje pokusa.



Slika 31: Potpuni faktorski 2^2 plan pokusa

U tablici 11. se nalaze rezultati ispitivanja, odnosno traženi odzivi prema postavljenom planu pokusa, rezultati su uzeti iz [16].

Tablica 11: Rezultati faktorskog pokusa

Red. br. ispitivanja	tangencijalna brzina x_1 (m/s)	brzina pomicanja x_2 (m/min)	R_a (nm)	P_c (vrh/cm)
1. (A)	-1	-1	194	77,8
2. (B)	-1	+1	282	68,4
3. (D)	+1	-1	120	65,3
4. (C)	+1	+1	91	96,1
5.	0	0	233	63,8
6.	0	0	235	61,9

5.2.1. Izračun i značenje koeficijenata modela

Koeficijent a_0 je presjek modela, a za njegovo pronalaženje potrebno je dodijeliti vrijednost 0 (u kodiranim jedinicama) za oba faktora. Takva točka pokusa se nalazi u središtu domene ispitivanja, a odziv ima oznaku y_0 .

$$y_0 = a_0 + a_1 \cdot 0 + a_2 \cdot 0 + a_{12} \cdot 0 \cdot 0 \rightarrow y_0 = a_0$$

Efekt tangencijalne brzine, odnosno faktora x_1 je koeficijent a_1 koji objašnjava kako varira odziv u ovisnosti o prvom faktoru, a utjecaj faktora x_2 je isključen. Za njegov izračun koriste se točke u pokusu kojima je koordinata faktora $x_1 = +1$, a to su točka D(+1,-1) i točka C(+1,+1); kao i točke kojima je koordinata faktora $x_1 = -1$, a to su točke A(-1,-1) i točka B(-1,+1). Dobivaju se sljedeće jednakosti, odnosno 4 jednadžbe s 4 nepoznanice:

$$y_D = a_0 + a_1 \cdot (+1) + a_2 \cdot (-1) + a_{12} \cdot (+1) \cdot (-1) = a_0 + a_1 - a_2 - a_{12}$$

dok se odziv u točki C s dodijeljenom oznakom y_C računa kao:

$$y_C = a_0 + a_1 \cdot (+1) + a_2 \cdot (+1) + a_{12} \cdot (+1) \cdot (+1) = a_0 + a_1 + a_2 + a_{12}$$

pa zbroj gornjih odziva daje sljedeći izraz:

$$y_D + y_C = 2 \cdot (a_0 + a_1)$$

Ponavljajući gornji postupak za točke A i B, donje slojeve faktora x_1 nastaju odzivi y_A i y_B , a njihov zbroj daje jednakost:

$$y_A + y_B = 2 \cdot (a_0 - a_1),$$

a njegovim oduzimanjem od gornjeg izraza nastaje:

$$4 \cdot a_1 = -y_A + y_D - y_B + y_C \rightarrow a_1 = \frac{1}{2} \left[\frac{y_D + y_C}{2} - \frac{y_A + y_B}{2} \right]$$

Budući da su izrazi $\frac{yD+yC}{2}$ i $\frac{yA+yB}{2}$ srednje vrijednosti odziva u gornjem i donjem sloju, odzivi u tim točkama C_+ i C_- imaju oznaku \bar{y}_+ i \bar{y}_- , pa je

$$a_1 = \frac{1}{2}[\bar{y}_+ - \bar{y}_-].$$

Koeficijent a_1 je stoga polovina razlike između srednjih vrijednosti u gornjem sloju faktora x_1 i srednje vrijednosti u donjem sloju.

$$a_1 = 0.25 \cdot (-194 + 120 - 282 + 91) = -66,25$$

Na gornji način se također dobiva i koeficijent a_2 koji je jednak prosječnoj varijaciji odziva kada s faktor x_2 mijenja od 0 do vrijednosti gornjeg sloja +1. Zato on predstavlja utjecaj brzine pomicanja.

$$a_2 = \frac{1}{4} \cdot (-y_A - y_D + y_B + y_C)$$

Također, na analogan način se dobiva i vrijednost koeficijenta a_{12} koji predstavlja iznos interakcije između faktora x_1 i x_2 , tj. varijaciju odziva između nulte i gornje vrijednosti faktora x_1 prilikom te iste promjene faktora x_2 , a vrijedi i obrnuto.

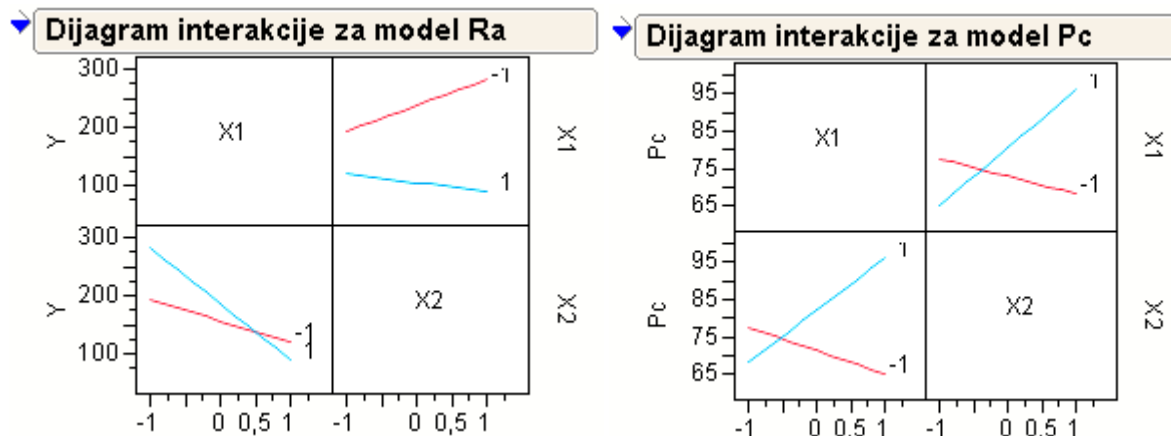
$$a_{12} = a_1 = \frac{1}{2} \left[\frac{yC+yB}{2} - \frac{yD+yA}{2} \right]$$

Nakon upisanih vrijednosti rezultata pokusa u potpuni faktorski plan za 2 faktora u JMP-u, dobivaju se sljedeće vrijednosti koeficijenata za oba tražena modela Ra i Pc.

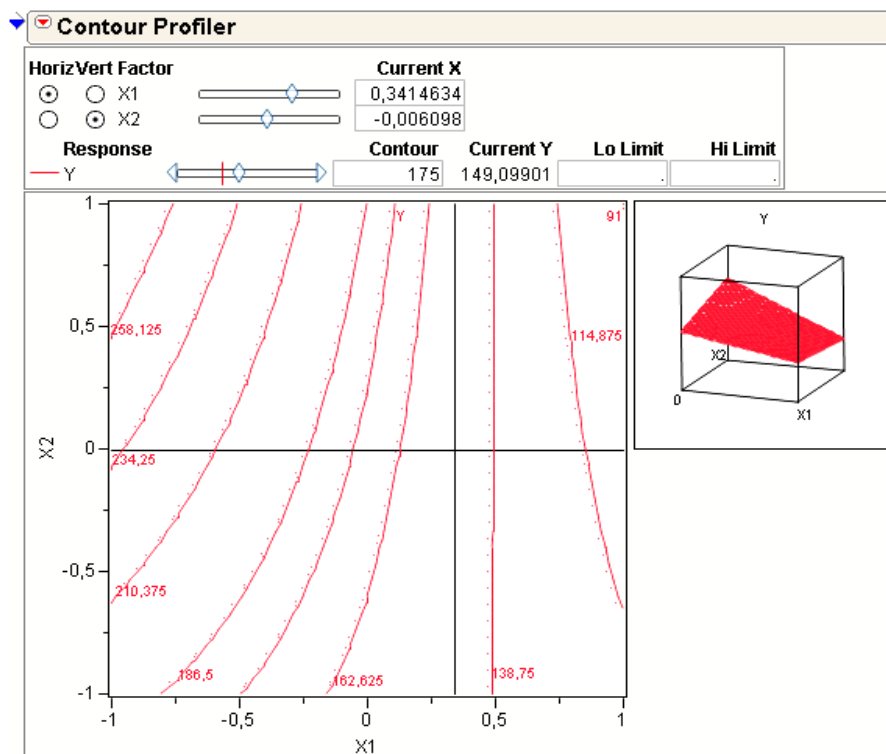
Parameter Estimates	
Term	Estimate
Intercept	171,75
X1	-66,25
X2	14,75
X1*X2	-29,25

Parameter Estimates	
Term	Estimate
Intercept	76,9
X1	3,8
X2	5,35
X1*X2	10,05

Donja slika prikazuje interakcijski dijagram za oba modela, na kojem je vidljiv utjecaj jednog faktora na gornjem i donjem sloju na drugi faktor. Na y os je vrijednost odziva, dok je na x osi vrijednost faktora. Slika br. 33 prikazuje konturni dijagram za model Ra.



Slika 32: Interakcijski dijagrami za modele Ra i Pc



Slika 33: Konturni dijagram linearnog modela Ra

5.2.2. Utvrđivanje adekvatnosti dobivenih linearnih modela

Temeljem prethodno izračunatih koeficijenata dobiveni su sljedeći matematički modeli, koje JMP prikazuje na sljedeći način:

<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">▼ Linearni model Ra</div> $171,75$ $+ -66,25 * X_1$ $+ 14,75 * X_2$ $+ X_1 * [X_2 * -29,25]$	<div style="border: 1px solid black; padding: 2px; display: inline-block; margin-bottom: 5px;">▼ Linearni model Pc</div> $76,9 + 3,8 * X_1 + 5,35 * X_2 + X_1 * [X_2 * 10,05]$
---	--

Na primjeru linearnog modela koji opisuje utjecaj faktora na parametar hrapavosti Ra provodi se provjera adekvatnosti modela, odnosno da li model zadovoljavajući opisuje stvarne vrijednosti. To se postiže uspoređivanjem odziva dobivenih modelom i kontrolnih točaka dobivenih u 5. i 6. pokusu, odnosno u centru domene pokusa.

Presjek modela je koeficijent $a_0 = y_0 = a_0 + a_1 \cdot 0 + a_2 \cdot 0 + a_{12} \cdot 0 \cdot 0$, a dobiva se iz aritmetičke sredine uzorka točaka A, B, C i D, ujedno je i odziv za centar pokusne domene.

$$n = 4 \text{ (A, B, C, D)}$$

$$\bar{x} = a_0 = 1/n \cdot (194 + 282 + 120 + 91) = 171,75 \text{ [nm]}$$

Korištenjem standardne devijacije mjere hrapavosti $\sigma_0 = 2 \text{ nm}$, standardna devijacija razdiobe aritmetičkih sredina uzoraka veličine $n = 4$ iznosi:

$$\sigma_x = \frac{\sigma_0}{\sqrt{n}} = \frac{2}{\sqrt{4}} = 1$$

, pa se uz sigurnost od 3 standardne devijacija dobiva 99,73%-tna vjerojatnost da je aritmetičkih sredina uzoraka veličine $n = 4$ (iz točaka A, B, C, D) unutar intervala od:

$$\hat{a}_0 = 171,75 \pm 3 \text{ [nm]}$$

Na temelju prethodnog izraza je vidljivo da će za različite provedene pokuse i time očitane pripadajuće 4 točke (A, B, C, D) iz faktorskog plana pokusa, odziv sredine domene biti ovisan o nastalim modelima, uz 99,73% odziva smještenih u gornji interval.

Također, uz očitavanje kontrolnih točaka dobivenih u 5. i 6. pokusu, odnosno u centru domene pokusa i standardnu devijaciju mjere hrapavosti $\sigma_0 = 2$ nm, dobiva se interval očitanih kontrolnih točaka.

$$n = 2 (233, 235)$$

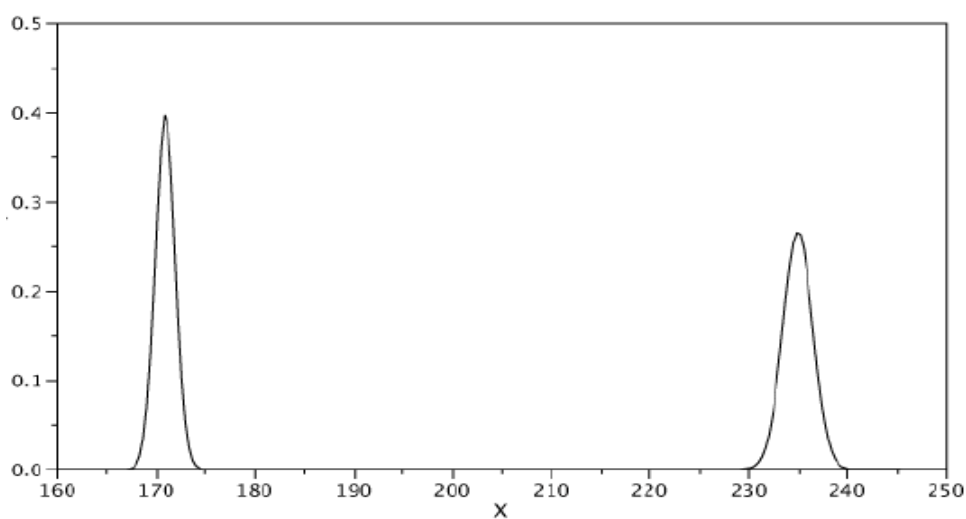
$$\bar{x} = a_0 = 1/n \cdot (233 + 235) = 234 \text{ [nm]}$$

Standardna devijacija razdiobe aritmetičkih sredina uzoraka veličine $n = 2$ iznosi:

$$\sigma_x = \frac{\sigma_0}{\sqrt{n}} = \frac{2}{\sqrt{2}}$$

, pa se uz sigurnost od 3 standardne devijacija dobiva 99,73%-tna vjerojatnost da je aritmetičkih sredina uzoraka veličine $n = 2$ (iz točaka 233, 235) unutar intervala od:

$$\hat{y}_0 = 234 \pm 3 \cdot \frac{2}{\sqrt{2}} = 234 \pm 4,24 \text{ [nm]}$$



Slika 34: Usporedba odziva dobivenih modelom i očitanih iz kontrolne točke

Temeljem izračuna i slike 34. vidljivo je da se intervali ne poklapaju. Stoga odziv modela ne daje traženu vrijednost očitane kontrolne točke, pa je očito da linearni model dobiven faktorskom planom pokusa ne opisuje adekvatno promatrani proces.

U nastavku je proveden kompozitni plan pokusa s postavljenim modelom drugog reda s ciljem zadovoljavajućeg opisivanja traženih procesa nelinearnim modelima.

5.3. Priprema za centralno kompozitno planiranje pokusa

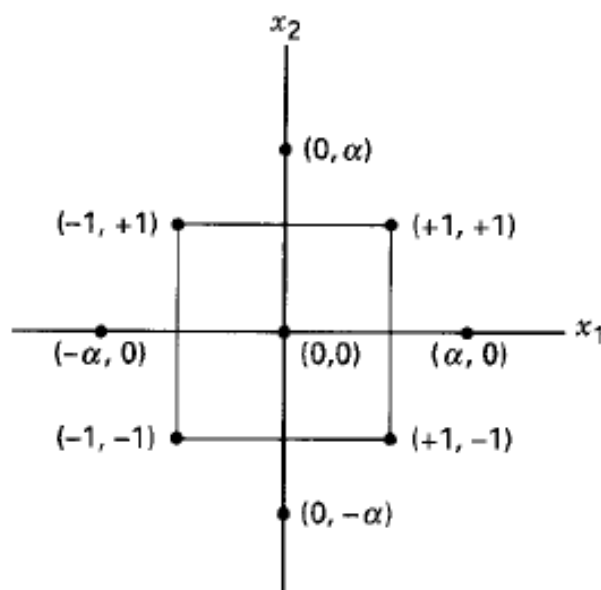
Budući da je metodologija odzivne površine (RSM – response surface methodology) sekvencijalna procedura kojom je nakon provedenog faktorskog plana pokusa i provjere ispravnosti linearnog modela testiranjem dobivenih vrijednosti u centru studijske domene i njihovih izmjerenih vrijednosti utvrđena neispravnost, provedeno je centralno kompozitno planiranje pokusa. Ono se provodi zbog signifikantnih zakrivljenosti u sustavu gdje je potrebno primijeniti model drugog reda, odnosno nelinearni model, s sljedećim izgledom za dva faktora:

$$y = a_0 + a_1x_1 + a_2x_2 + a_{12}x_1x_2 + a_{11}x_1^2 + a_{22}x_2^2$$

Uz četiri točke iz kutova pokusne domene za provedbu pokusa iz faktorskog plana i dvije testne točke u sredini domene za potpuni kompozitni plan pokusa potrebne su još četiri aksijalne (zvjezdaste) točke i dvije dodatne točke u sredini domene. Slika br. 35 prikazuje kompozitni plan pokusa za dva faktora, dok tablica 12 daje popis točaka iz kompozitnog plana pokusa.

Tablica 12: Vrste točaka iz centralnog kompozitnog plana pokusa

Vrste točaka	Opis
Faktorske	Potpuni faktorski plan, dvije razine po faktoru.
Aksijalne (zvjezdaste)	Kod svake točke su svi faktori osim jednog na nultoj razini, a sve se točke nalaze na istoj udaljenosti od središta domene.
Centralne	Sve vrijednosti faktora su na nultoj razini za svaku točku.



Slika 35: Domena centralnog kompozitnog plana pokusa s pripadajućim točkama, [17]

Kod centralnog kompozitnog plana pokusa svaki faktor se sastoji od pet razina $(-\alpha, -1, 0, +1, +\alpha)$, dok se broj izvođenja pokusa sastoji od zbroja pokusa iz faktorskog n_f , aksijalnog (zvjezdastog) n_α i centralnog n_0 dijela. On je najšire korišten plan pokusa za prilagođavanje procesa modelu drugog reda, iako postoje još također kvalitetni planovi, [17].

Broj izvođenja centralnog kompozitnog pokusa: $n = n_f + n_\alpha + n_0 = 4 + 4 + 1 \cdot 4 = 12$ izvođenja

5.3.1. Smještaj aksijalnih (zvjezdastih) točaka

Nakon provedenog $2^{k=2}$ faktora potpunog faktorskog pokusa s $n_f = 4$ izvršenja i utvrđene neadekvatnosti dobivenog modela prvog reda sekvencijalno se prelazi na centralni kompozitni plan (central composite design ili CCD). Time su dodani aksijalni pokusi (zvjezdaste točke na udaljenosti α od centra pokusne domene) i do datne centralne točke čime se omogućava nelinearnost modela kroz drugu potenciju faktora. Određeno je postojanje 4 centralne točke upisom u izbornik JMP-a, a također i udaljenost α . Aksijalna vrijednost α nije ovisna jednoznačno [16], pa je predmet rasprave među raznim autorima što uzrokuje dvojbenost kod smještaja aksijalnih točaka.

Preciznost koeficijenata iz postuliranog modela je ovisna o smještaju pokusnih točaka, odnosno aksijalnih točaka. Kod faktorskih planova postoji optimalan smještaj točaka, što nije slučaj sa kvadratnim modelima drugog reda. Važno je omogućiti „dobar“ odziv modela u području od interesa, što se postiže kada su odstupanja stabilna za željeno područje X.

Kriterij izovarijance

Za postizanje odstupanja odziva koje je jednako u svim točkama isto odaljenima od središta domene ispitivanja primjenjuje se rotacija kada je α jednakom korijenu broja izvođenja pokusa u potpunom faktorskom pokusu.

$$\alpha = n_f^{1/4}$$

Budući da je glavna svrha metodologije odzivne površine (RSM) optimizacija, lokacija optimuma je nepoznata u vremenu prije izvođenja pokusa, pa je primjena rotacije važna zbog žele postizanja iste preciznosti odziva u svim smjerovima.

Kriterij približne ortogonalnosti

Iznos α je funkcija zbroja pokusa iz faktorskog n_f , aksijalnog (zvjezdastog) n_α i centralnog n_0 dijela, tablica 13.

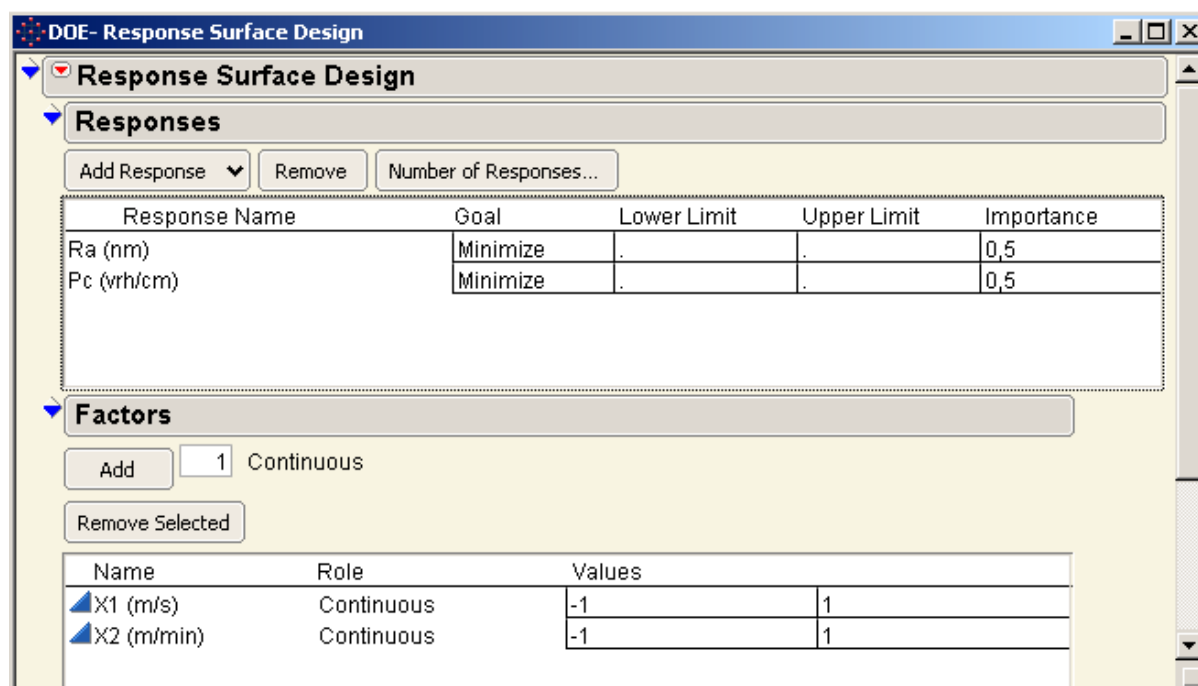
$$\alpha = \left(\frac{n_f \left(\sqrt{n_0 + n_f + n_\alpha} - \sqrt{n_f} \right)^2}{4} \right)^{1/4}$$

Tablica 13: Vrijednost α kao funkcija n_f , n_α i n_0 za kriterij približne ortogonalnosti

Broj faktora:	2	3
Plan, n_f , n_α	2^2 , $n_f = 4$, $n_\alpha = 4$	2^3 , $n_f = 8$, $n_\alpha = 6$
$n_0 = 1$	$\alpha = 1,000$	$\alpha = 1,215$
$n_0 = 2$	$\alpha = 1,078$	$\alpha = 1,287$
$n_0 = 3$	$\alpha = 1,174$	$\alpha = 1,353$
$n_0 = 4$	$\alpha = 1,210$	$\alpha = 1,414$

5.4. Centralno kompozitni pokus

Uz odabrane četiri centralne točke, i četiri točke iz potpunog faktorskog pokusa i poštujući kriterij približne ortogonalnosti odabrana α je 1,210. U nastavku je obavljeno unošenje pokusnih podataka prema [16] u softver JMP, za dva tražena modela Ra i Pc.



Slika 36: Postavljanje odziva i faktora u izbornik JMP-a

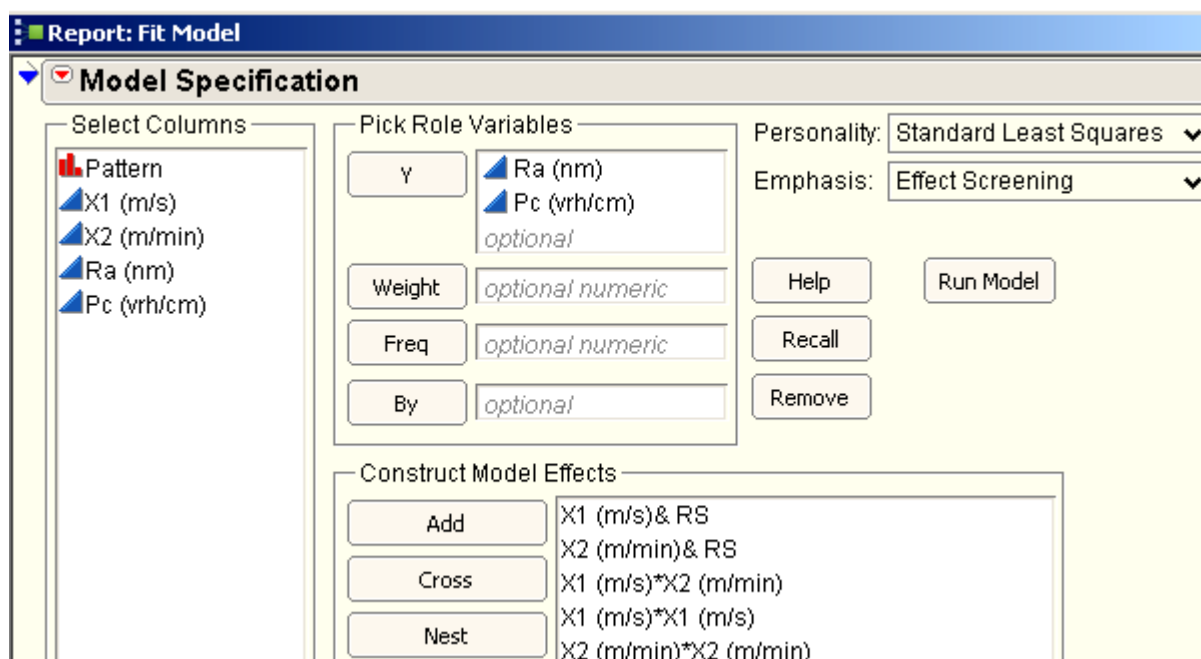
Unutar JMP-ovog okvira za model odzivne površine, potrebno je izvršiti unos traženih odziva s obzirom na promatrani broj faktora, slika 36. Za oba odzivna modela potrebno je postaviti minimum s obzirom na optimizacijski kriterij, a budući da svrha optimizacije izbor faktora s obzirom na dva odziva, u okviru značaj (importance) postavlja se koeficijent važnosti pojedine odzivne površine za ukupnu optimizaciju. Značaj oba odziva je postavljen na 0,5.

U nastavku postavljanja plana pokusa potrebno je unijeti broj centralnih točaka i vrijednost α , kao i rezultate pokusa prema centralnom kompozitnom planu, tablica 14.

Tablica 14: Centralni kompozitni plan pokusa i rezultati za oba odziva uneseni u JMP

Design	Pattern	X1 (m/min)	X2 (m/s)	Ra (nm)	Pc (vrh/cm)
1	A0	1,21	0	195	60,4
2	00	0	0	233	63,8
3	--	-1	-1	194	77,8
4	a0	-1,21	0	154	52,3
5	00	0	0	235	61,9
6	00	0	0	232	61,5
7	--+	-1	1	120	65,3
8	+-	1	-1	282	68,4
9	0a	0	-1,21	278	87
10	++	1	1	91	96,1
11	0A	0	1,21	122	95,7
12	00	0	0	230	60,5

Nakon unesenih podataka u proračunsku tablicu, slika 37., obavlja se popunjavanje JMP-ovog okvira za prilagođavanje modela. U ovom koraku unose se zavisne varijable odziva Ra i Pc, kao faktori i njihove interakcije u okvir za postavljanje efekata modela, slika 38. Formiranje traženih matematičkih modela ostvaruje se odabirom *Run Model*.



Slika 37: JMP-ov izbornik za izradu matematičkog modela željenih odziva

5.4.1. Nelinearni modeli Ra i Pc i njihovi koeficijenti

Na temelju sljedećeg općeg izgleda $y = a_0 + a_1x_1 + a_2x_2 + a_{12}x_1x_2 + a_{11}x_1^2 + a_{22}x_2^2$ potrebni koeficijenti dobiveni su u JMP-u, a iznos im je vidljiv u slici br. 38.

Iznos koeficijenata za model Ra		Iznos koeficijenata za model Pc	
Term	Estimate	Term	Estimate
Intercept	232,37034	Intercept	62,085426
X1 (m/s)	15,67651	X1 (m/s)	4,5034785
X2 (m/min)	-65,49465	X2 (m/min)	3,7133743
X1 (m/s)*X2 (m/min)	-29,25	X1 (m/s)*X2 (m/min)	10,05
X1 (m/s)*X1 (m/s)	-39,19571	X1 (m/s)*X1 (m/s)	-4,326304
X2 (m/min)*X2 (m/min)	-21,77887	X2 (m/min)*X2 (m/min)	19,579167

Slika 38: Iznosi koeficijenata nelinearnih modela Ra i Pc dobivenih u JMP-u

Dobiveni koeficijenti imaju vrijednost p (p -value), odnosno iznos koji označava vjerojatnost da je koeficijent nesignifikantan $<0,0001$ što im omogućuje ulaženje u model sa svojim iznosom. Ukoliko bi vrijednost p bila blizu 1, pripadajući koeficijenti ulaze u model za vrijednošću 0. U nastavku su prikazani dobiveni modeli, odnosno odzivne funkcije Ra i Pc.

▼ **Nelinearni model Ra**

```
232,370339132097
+ 15,676510493346 * X1 (m/s)
+ -65,494645073757 * X2 (m/min)
+ X1 (m/s) * [ X2 (m/min) * -29,25 ]
+ X1 (m/s) * [ X1 (m/s) * -39,195710432176 ]
+ X2 (m/min) * [ X2 (m/min) * -21,778867320367 ]
```

▼ **Nelinearni model Pc**

```
62,0854257666354
+ 4,50347853699374 * X1 (m/s)
+ 3,71337432522156 * X2 (m/min)
+ X1 (m/s) * [ X2 (m/min) * 10,05 ]
+ X1 (m/s) * [ X1 (m/s) * -4,3263035817914 ]
+ X2 (m/min) * [ X2 (m/min) * 19,5791673559861 ]
```

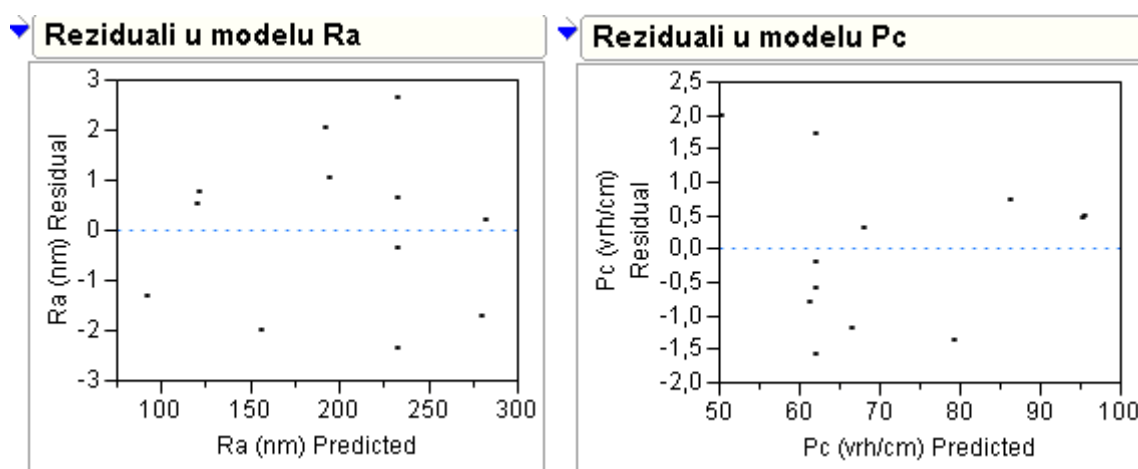
5.4.2. Analiza rezultata

Na temelju dobivenih modela tablica br. 15 pokazuje odzive dobivene za faktore definirane točkama u planu pokusa i njihove rezidualne, odnosno razlike između stvarnih i odzivnih vrijednosti.

Tablica 15: Usporedba odziva modela i stvarnih vrijednosti dobivenih pokusom

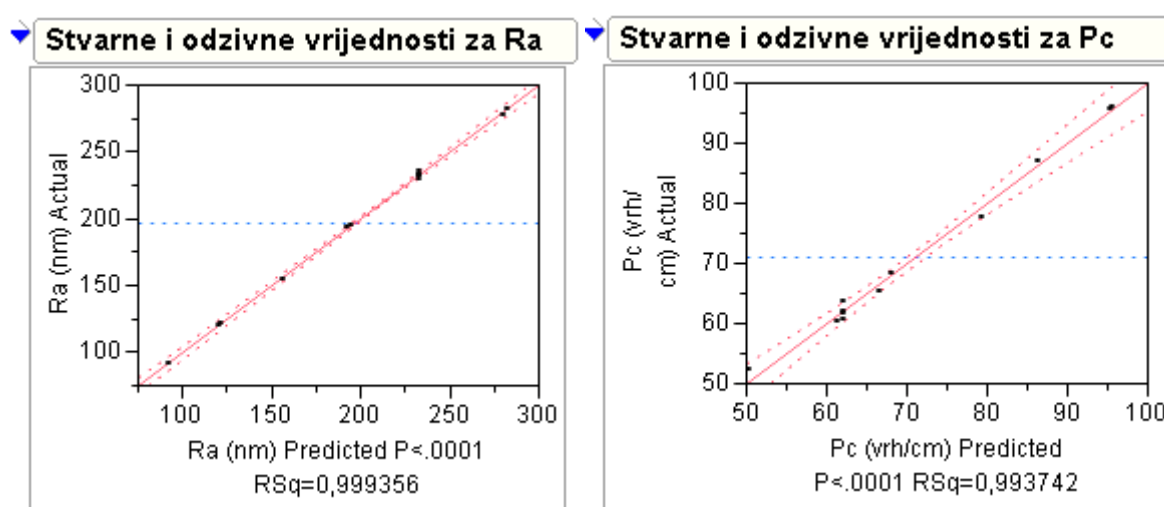
X1(m/min)	X2 (m/s)	Ra (nm)	Pc (vrh/cm)	Predicted Ra (nm)	Residual Ra (nm)	Predicted Pc (vrh/cm)	Residual Pc (vrh/cm)
1,21	0	195	60,4	193,952477	1,04752281	61,2004937	-0,8004937
0	0	233	63,8	232,370339	0,62966087	62,0854258	1,71457423
-1	-1	194	77,8	191,963896	2,03610404	79,1714367	-1,3714367
-1,21	0	154	52,3	156,015322	-2,0153218	50,3020757	1,99792434
0	0	235	61,9	232,370339	2,62966087	62,0854258	-0,1854258
0	0	232	61,5	232,370339	-0,3703391	62,0854258	-0,5854258
-1	1	120	65,3	119,474606	0,52539419	66,4981853	-1,1981853
1	-1	282	68,4	281,816917	0,18308305	68,0783938	0,32160625
0	-1,21	278	87	279,73242	-1,73242	86,2581018	0,74189824
1	1	91	96,1	92,3276268	-1,3276268	95,6051424	0,4948576
0	1,21	122	95,7	121,235379	0,76462105	95,2444676	0,45553237
0	0	230	60,5	232,370339	-2,3703391	62,0854258	-1,5854258

Također, gornja tablica u vizualnoj interpretaciji koristeći funkcijski odnos dobivenih odziva i reziduala ima izgled u slici br. 39., gdje je vidljiva nasumična rasprostranjenost reziduala.



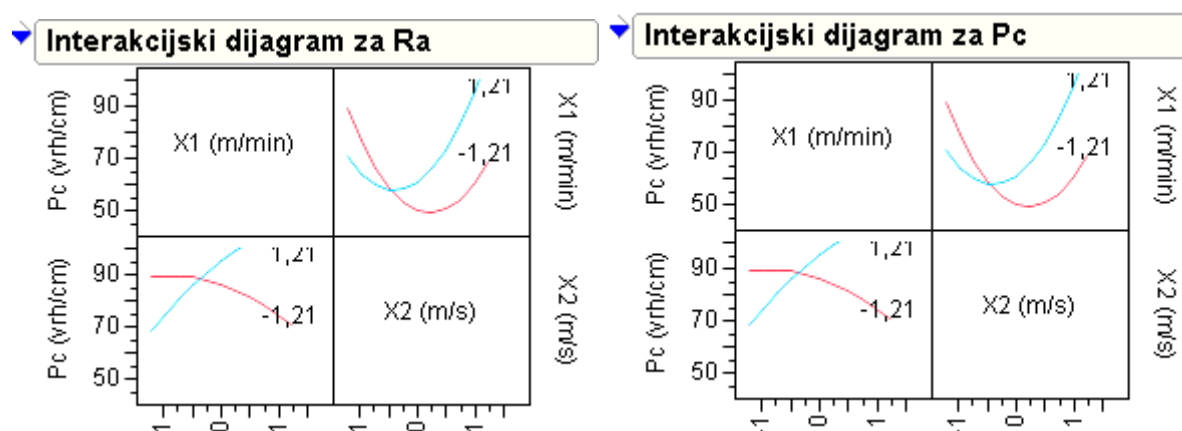
Slika 39: Prikaz razlika stvarnih i odzivnih vrijednosti za model Ra i Pc

Kvadrat koeficijenta korelacije između stvarnih vrijednosti i dobivenih odziva modelom, odnosno koeficijent determinacije R^2 pokazuje kvalitetu modela. Njegov iznos od 0,9993 za model Ra i 0,9937 za model Pc pokazuje reprezentativnost oba modela, pokazujući postotke objašnjenja stvarnih vrijednosti odziva u pokusnim točkama kroz promjenu nezavisnih varijabli, tj. faktora x_1 i x_2 . To je vidljivo u slici br. 40. gdje je prikazana usporedba stvarnih i odzivnih vrijednosti u intervalu pouzdanja.



Slika 40: Prikaz kvalitete modela Ra i Pc kroz koeficijente determinacije

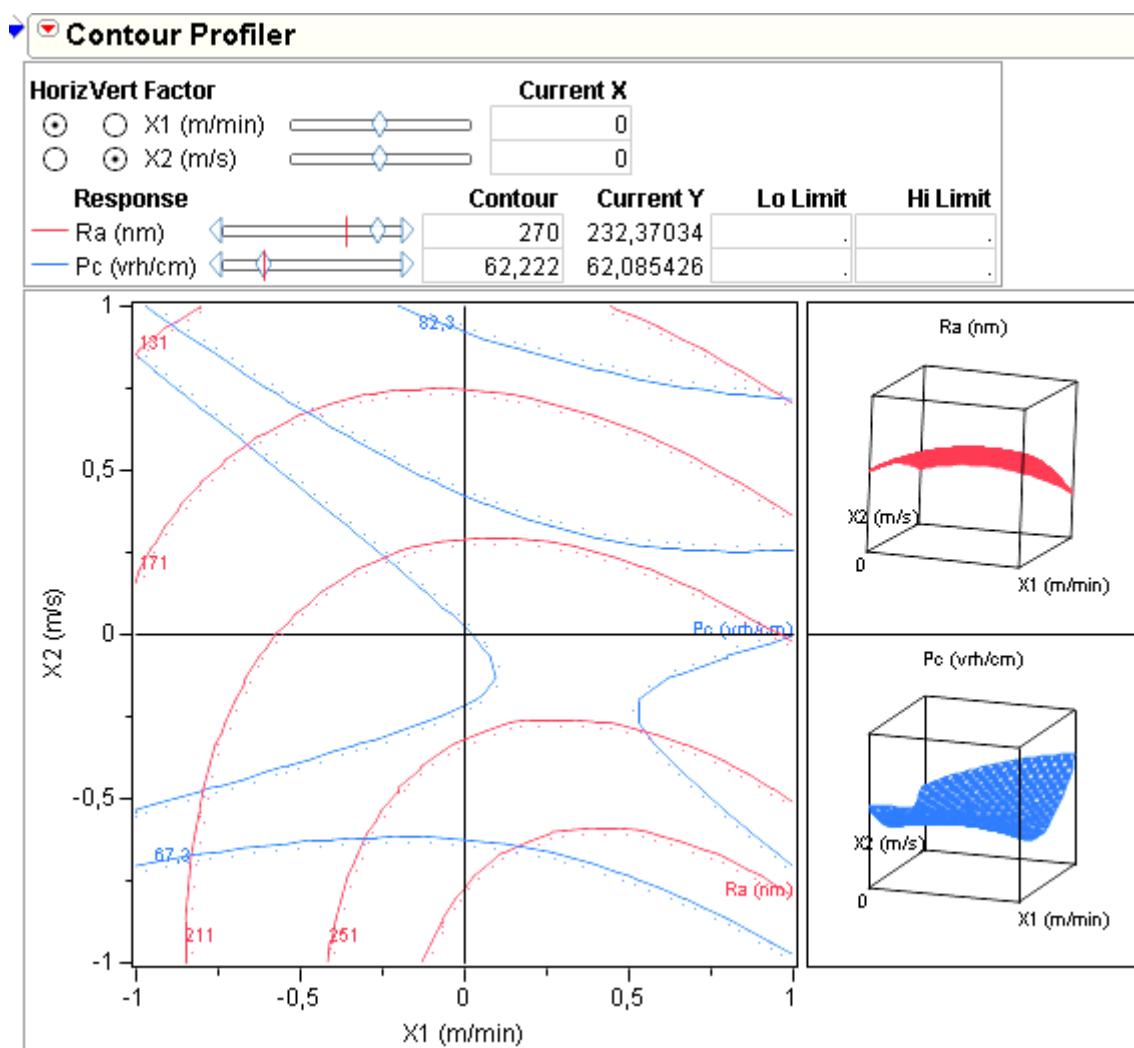
Donja slika br 41. prikazuje interakcijski dijagram za oba modela, na kojem je vidljiv utjecaj jednog faktora na gornjem i donjem sloju na drugi faktor.



Slika 41: Interakcijski dijagrami za nelinearne modele Ra i Pc

5.4.3. Optimizacija traženih faktora x_1 i x_2

Odzivne površine dobivenih nelinearnih modela Ra i Pc imaju izgled vidljiv u slici br. 42., dok se za optimizaciju faktora koriste oba modela sa koeficijentom važnosti svakog modela od 0,5. Donji konturni dijagram pokazuje vrijednost od Ra=232,37 i Pc=62,085 za središte pokusne domene.



Slika 42: Odzivne površine nelinearnih modela i konturni dijagram

Budući da je optimizacija temeljena na dva nezavisna modela gdje optimum jednoga nije ujedno i optimum drugoga, tražene vrijednosti faktora su utemeljene na najbolje mogućem kompromisu odziva Ra i Pc iste važnosti.

Optimizacija je dobivena na jednostavan i intuitivan način u JMP softveru pomoću funkcije poželjnosti (eng. desirability function), kao dio optimizacije metodologijom poželjnosti (desirability optimization methodology – DOM). Ona omogućuje integriranje višestrukih odziva u jedan kojemu se traži optimum faktora s najpoželjnijim iznosom od 1, a najnepoželjnijim od 0.

Ovakav način optimizacije započinje definiranjem funkcije poželjnosti za svaki od modela zasebno gdje im se traži minimum, dok je očito da će vrijednosti 1 i 0 pojedinačnih funkcija poželjnosti zauzeti minimalne i maksimalne vrijednost funkcija cilja unutar dozvoljenog intervala faktora od -1 i +1.

Na slici br. 43. vidljivo je ručno unošenje vrijednosti potrebnih za definiranje pojedinačnih funkcija poželjnosti. Budući da maksimum funkcije Pc na intervalu od -1 i +1 iznosi 95,6 (dobiveno pomoću JMP-ovog alata za predviđanje odziva modela fiksiranjem jednog odziva, a mijenjanjem drugog – slika 44) dodijeljeni mu je najnepoželjniji iznos 0. Minimum od 52,75 Pc ima najpoželjniji faktor 1, a sredina odziva modela od 74,175 ima faktor poželjnosti 0,5. Također, potrebno je definirati kriterij optimuma što je minimum kao i vrijednost važnosti pojedine funkcije poželjnosti u iznosu od 0,5 za svaku.

Za model Ra su vrijednosti postavljene na isti način, korištenjem JMP-ovog alata na slici br. 44, te je vidljivo da je za istu kombinaciju faktora nemoguće postići najpoželjnije vrijednosti u ova odziva Ra i Pc.

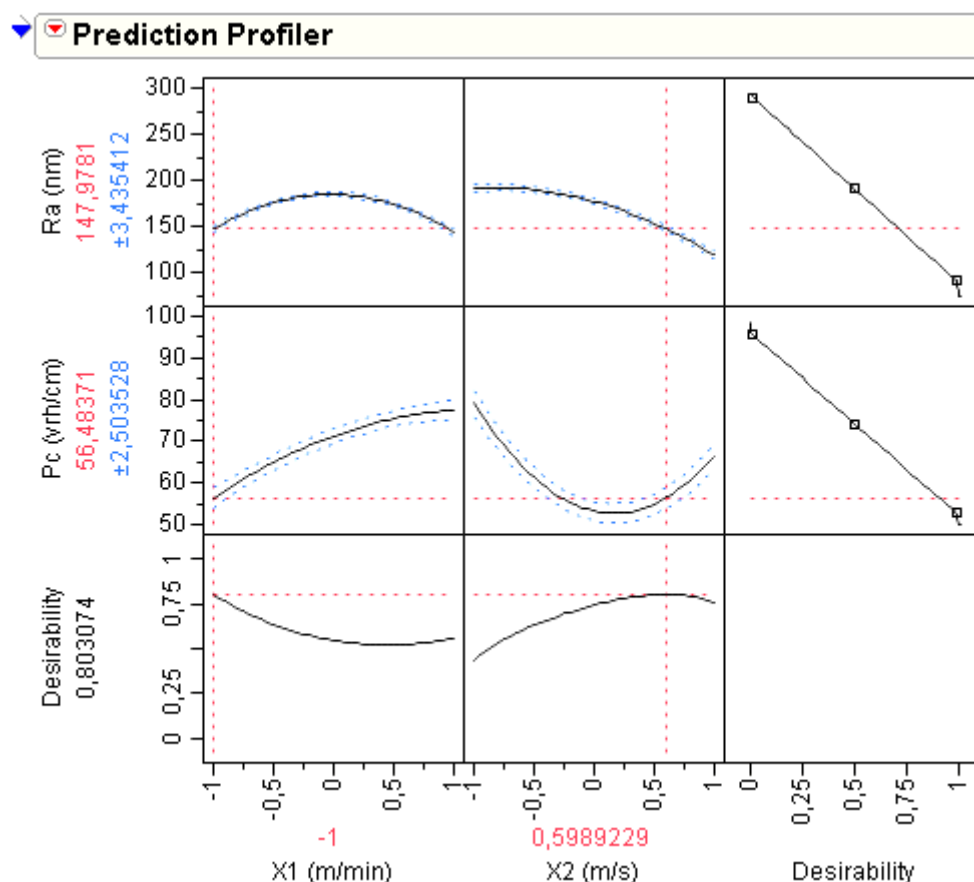
Response	High Value	Middle Value	Low Value	High Desirability	Middle Desirability	Low Desirability
Ra (nm)	290	191	92	0	0,5	1
Pc (vrh/cm)	95,6	74,175	52,75	0	0,5	1

Slika 43: Postavljanje vrijednosti za pojedinačne funkcije poželjnosti modela Ra i Pc

Budući da se pojedinačne vrijednosti funkcije poželjnosti formiraju dodjeljivanjem odzivima y_i vrijednosti poželjnosti u rasponu od $0 \leq d \leq 1$, globalna funkcija poželjnosti koja je i glavni cilj optimizacije za više funkcija cilja dobiva se pomoću geometrijske sredine pojedinačnih vrijednosti funkcija poželjnosti d_i uzimajući u obzir više različitih odziva u isto vrijeme.

$$D = \sqrt[n]{d_1 d_2 \cdots d_n}$$

Vrijednost globalne poželjnosti od 1 označava postizanje optimalnih vrijednosti za svaki od pojedinačnih modela, odnosno sve pojedinačne funkcije imaju vrijednost 1. Promatrani proces brušenja postiže najbolji mogući kompromis globalne funkcije poželjnosti u iznosu od 0,8 s kombinacijom faktora x_1 koji predstavlja kodirani iznos brzine pomicanja i x_2 koji predstavlja kodirani iznos tangencijalne brzine ($x_1 = -1$ i $x_2 = 0,599$), slika 44.



Slika 44: Optimalni faktori dobiveni pomoću globalne funkcije poželjnosti

5.4.3.1. Zaključak

Optimumi, tj. minimalne vrijednosti funkcija cilja Ra i Pc unutar kodiranog intervala +1 i -1, kao i kombinacija kodiranih faktora su vidljivi u tablici br. 16.

Tablica 16: Vrijednosti optimuma funkcija Ra i Pc unutar raspona kodiranih faktora -1 i 1

Funkcija cilja	Faktor x_1	Faktor x_2	Vrijednost $f(x)$
Ra	1	1	92,37 (nm)
Pc	-1	0,15	52,75 (vrh/cm)

Najbolji mogući kompromis, (slika 44), odnosno optimum dobiven pomoću funkcije poželjnosti (desirability function) je nakon izvršenog dekodiranja u fizikalne vrijednosti faktora prikazan u tablici br. 17.

$$\begin{aligned}
 V_p &= \text{korak}_1 \cdot x_1 + V_{0,1} & V_p &= 0,75 \cdot -1 + 1,65 & V_p &= 0,9 \text{ (m/min)} \\
 V_g &= \text{korak}_2 \cdot x_2 + V_{0,2} & V_g &= 5 \cdot 0,599 + 20 & V_g &= 22,995 \text{ (m/s)}
 \end{aligned}$$

Tablica 17: Najbolji mogući kompromis kombinacije traženih faktora

Tangencijalna brzina (kodirana)	Brzina pomicanja (kodirana)	Tangencijalna brzina v_p (m/min)	Brzina pomicanja v_g (m/s)	Ra (nm)	Pc (vrh/cm)
-1	0,599	0,9	22,995	147,98	56,48

Optimalne postavke parametara procesa brušenja su glavno gibanje uzrokovano rotacijom alata, odnosno tangencijalna brzina $v_g = 22,995$ (m/s) i pomoćno gibanje uzrokovano pomicanjem alata, odnosno brzina pomicanja $v_p = 0,9$ (m/min), obzirom na kvalitetu površine završne obrade formiranu parametrima Ra i Pc.

6. OPTIMIZACIJA U MATLAB-U

U ovom poglavlju provedena je optimizacija primjera iz [20], [21] i [22] korištenjem optimizacijske skupine alata iz MATLAB-a, odnosno „Optimization Toolbox“ softver. On uključuje funkcije za rješavanje raznih vrsti optimizacijskih problema kao npr. nelinearnih s i bez ograničenja. Većina tih funkcija su Matlab-ove M-datoteke napisane Matlab-ovom sintaksom što čini specijaliziranje algoritme.

Također biti će korišten optimizacijski alat s grafičkim korisnički sučeljem (GUI - graphical user interface) koji služi za odabir optimizacijskih alata, postavljanje optimizacijskih parametara i izvršavanje optimizacije. On omogućuje odabir adekvatnih algoritama i donosi prikladno sučelje za razne slučajeve. Ostvaruje također i jednostavno postavljanje optimizacijskog problema i njegovu modifikaciju, a i olakšava upis odgovarajuće sintakse za optimizacijske funkcije.

6.1. Nelinearna optimizacija s ograničenjima

U [20] pomoću rotacijskog centralnog kompozitnog plana pokusa dobivena je sljedeća funkcija cilja, koja je polinom drugog reda s tri varijable gdje odziv predstavlja postignutu hrapavost prilikom procesa čeonog glodanja:

$$\begin{aligned} Ra = & 4,9271 - 0,0574 \cdot v_c + 1,1808 \cdot f_z - 0,0273 \cdot a_p \\ & + 1,78035 \cdot 10^{-4} \cdot v_c^2 + 11,6182 \cdot f_z^2 + 0,031243 \cdot a_p^2 - \\ & - 0,025 \cdot v_c \cdot f_z - 4,7619 \cdot 10^{-4} \cdot v_c \cdot a_p + 0,29762 \cdot f_z \cdot a_p \end{aligned}$$

varijabla v_c predstavlja brzinu rezanja (m/min), varijabla f_z je posmak (mm/zubu) dok je a_p dubina rezanja (mm). Čeono glodanje je obavljano na obradnom centru Spinner VC560, s glodalom Helido S845 F45SX D063-05-22-R16, dok se ostali podaci za nalaze u [20].

U daljnjem tekstu je prikazana optimizacija gornje funkcije cilja kroz traženje minimuma uz ograničenja postavljena tehnološkim mogućnostima strojne obrade.

Tablica 18: Dozvoljeni intervali varijabli za pretraživanje optimalnog rješenja, [20]

Varijabla	Minimum	Maksimum
$v_c(x_1)$	150	180
$f_z(x_2)$	0,24	0,36
$a_p(x_3)$	1,30	2,70

Za korištenje optimizacijskih alata potrebno je najprije definirati funkciju cilja kao M-datoteku pomoću Matlab-ove sintakse, a zatim ograničenja u drugu M-datoteku.

6.1.1. Postavljanje problema za provođenje optimizacije

1. korak – M-datoteka za funkciju cilja

M-datoteka sadrži tekst Matlab-ovih naredbi i ima ekstenziju .m, a otvara se iz glavnog izbornika ili naredbom *edit „ime funkcije koju unosimo“* u naredbeni okvir. Npr.:

```
function y= ime_funkcije(x1,x2, ..., xn)
y= ...
```

gdje ime funkcije i datoteka moraju biti identični, odnosno funkcija *OdzivRa* mora biti pohranjena u datoteci *OdzivRa.m*. Nakon unošenja i usnimavanja u odabrani direktorij funkcija cilja u M-datoteci ima sljedeći izgled:

```
function f = OdzivRa( x )
%Ra Funkcija cilja ima odziv hrapavost
% x(1) je brzina rezanja, x(2) je posmak,
% a, x(3) je dubina rezanja

f = 4.9271 - 0.0574 * x(1) + 1.1808 * x(2) - 0.0273 * x(3) ...
+ 1.78035*10^(-4) * x(1)^2 + 11.6182 * x(2)^2 + ...
0.031243*x(3)^2-0.025*x(1)*x(2) - 4.7619*10^(-4) * x(1)*x(3) ...
+ 0.29762*x(2)*x(3);

end
```

2. korak – M-datoteka za ograničenja

Unos funkcija ograničenja se ostvaruje pomoću forme $c(x) \leq 0$ i $ceq(x) = 0$, tako da se unos ograničenja mora preoblikovati zbog poštivanja sintakse. Također, potrebno je zbog funkcija iz optimizacijskih alata unijeti obe funkcije ograničenja; jednakost i nejednakost. Budući da primjer nema ograničenja jednakosti, funkcije $ceq(x)$ se unosi prazna u obliku [].

Unesena ograničenja poštujući sintaksu imaju slijedeći oblik u M-datoteci po nazivom OgranicenjaRa.m :

```
function [c, ceq] = OgranicenjaRa(x)
c (1) = 150 - x(1);
c (2) = x(1) - 180;
c (3) = 0.24 - x(2);
c (4) = x(2) - 0.36;
c (5) = 1.3 - x(3);
c (6) = x(3) - 2.7;
ceq = [ ];
```

6.1.2. Provođenje optimizacije

Optimizacija postavljenog matematičkog modela je provedena na dva načina. Prvi način koristi optimizacijsko korisničko grafičko sučelje GUI (Graphical User Interface), dok drugi način koristi naredbene funkcije unutar glavnog naredbenog okvira.

6.1.2.1. Način 1. – Korištenje grafičkog korisničkog sučelja

Provođenje optimizacije započeto je otvaranjem grafičkog korisničkog sučelja GUI upisom naredbe *optimtool* u naredbeni okvir.

1. korak – Odabir željenog solvera

Početno postavljeni solver je *fmincon* - Constrained nonlinear minimization koji je namijenjen za traženje minimuma nelinearne funkcije cilja s ograničenjima.

2. korak – Odabir algoritma

Odabrani algoritam je *Active set* jer omogućuje nelinearna ograničenja, neovisno o tome što ona ne postoje u korištenom primjeru.

3. korak – Unos funkcije cilja

Funkcija cilja postavljena u M-datoteci se unosi pomoću oznake @ koja označava ne direktno pozivanje funkcije. Upisuje se *@OdzivRa* u okvir „objective function“.

4. korak – Postavljanje startne točke

Postavljena inicijalna točka je $[0 \ 0 \ 0]$ u kojoj solver *fmincon* započinje za traženjem minimuma.

5. korak – Unos ograničenja

U polje *Nonlinear constraint function* unose se ograničenja postavljena M-datotekom, također pomoću oznake @. Upisuje se *@OgranicjenjaRa*.

Solver: fmincon - Constrained nonlinear minimization

Algorithm: Active set

Problem

Objective function: @OdzivRa

Derivatives: Approximated by solver

Start point: [0 0 0]

Nonlinear constraint function: @OgranicjenjaRa

Derivatives: Approximated by solver

Slika 45: Izgled postavljenog korisničkog grafičkog sučelja za optimizaciju u Matlab-u

6. korak – Pokretanje optimizacije

Odabirom opcije *Start* unutar okvira za pokretanje optimizacije dobiva se optimalno rješenje prema slici br. 46., gdje je iznos postignutog odziva $\text{OdzivRa}(x_1=179,793 \ x_2=0,24 \ x_3=1,3)$ jednaka 0.234766.

Tablica 19: Prikaz dobivenog rješenja optimizacijom

Final point:		
1	2	3
179,793	0,24	1,3

6.1.2.2. Način 2. – Korištenje naredbene funkcije

Nakon postavljenog optimizacijskog problema u poglavlju 6.1.1., kroz izradu M-datoteke funkcije cilja i ograničenja, ovdje je prikazana direktna optimizacija u naredbeni okvir.

1. korak – Odabir postavki

Na temelju sljedeće strukture postavki odabran je iterativan prikaz s active-set algoritmom. Ovaj izraz se izvršava u naredbenom prozoru i koristi funkciju `optimset`:

```
options = optimset('Display','iter','Algorithm','active-set')
```

2. korak – Izvršavanje optimizacije

Korištenjem solvera *fmincon* i postavljenih opcija *options* kroz izvršavanje sljedećeg izraza dobiva se minimalna vrijednost funkcije cilja i vrijednosti varijabli x_1 , x_2 i x_3 . Šest praznih uglatih zagrada označavaju vrste ograničenja koje se ne koriste.

```
[x,fval]=fmincon(@OdzivRa,[0 0 0],[[],[],[],[],[],[],[],@OgranicenjaRa,options)
```

Optimalni rezultati dobiveni korištenjem naredbene funkcije dobiveni solverom *fmincon* prikazani su u nastavku, gdje je *fval* vrijednost funkcije cilja, a matrica *x* vrijednosti varijabli x_1 , x_2 i x_3 :

x =

```
179.7934  0.2400  1.3000
```

fval =

```
0.2348
```

6.1.3. Analiza rješenja i grafički prikaz funkcije cilja

Za čeonu glodanje, prema [29], korištenjem Matlab-a dobivene su optimalne vrijednosti faktora:

varijabla v_c predstavlja brzinu rezanja (m/min) = 179,7934, varijabla f_z je posmak (mm/zubu) = 0,24 dok je a_p dubina rezanja (mm) = 1,3

Tako postavljeni faktori uzrokuju kvalitetu obrađene površine s $R_a = 0,2348$ (μm).

Grafički prikaz funkcije je ostvaren za fiksnu optimalnu vrijednost varijable $x_3 = 1,3$, dok su ostale varijabilne u rasponima zadanim ograničenjima, ostvarene pomoću naredbe *meshgrid*. Najprije je unesena nova funkcija *OdRaop(x,y)* gdje je u *OdzivRa.m* umjesto x_3 unesen 1,3 i izgleda je:

```
function f = OdRaop(x,y)
%Ra Funkcija cilja ima odziv hrapavost
% x je brzina rezanja, y je posmak,
% a, z je dubina rezanja

f = 4.9271 - 0.0574 * x + 1.1808 * y - 0.0273 * 1.3 ...
+ 1.78035*10^(-4) * x.^2 + 11.6182 * y.^2 + ...
0.031243*1.3^2-0.025*x.*y - 4.7619*10^(-4) * x*1.3 ...
+ 0.29762*y*1.3;

end
```

Raspon za x i y je definiran pomoću ograničenja i rasponom od 0,1 i 0,01, a zadan funkcijom `meshgrid`:

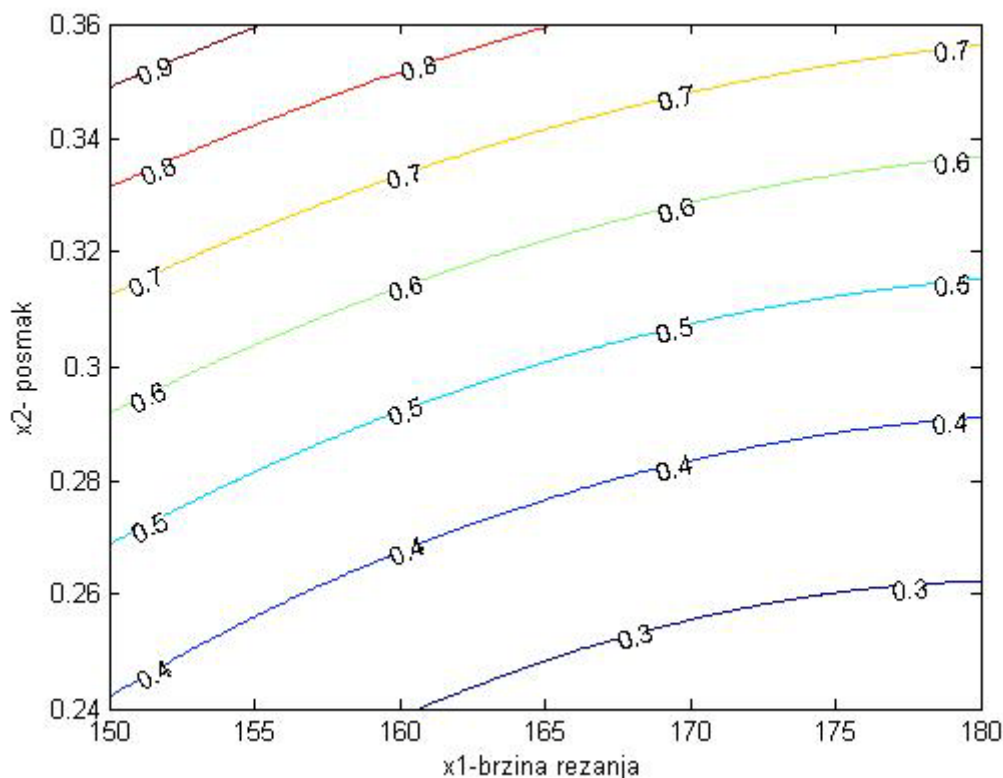
```
[x,y] = meshgrid(150:0.1:180,0.24:0.01:0.36)
```

Odziv funkcije `RaOdap(x,y)` za zadane x i y je definiran pomoću naredbe: `z=OdRaap(x,y)`, dok se konturni dijagram izvršava pomoću sljedećih upisa u naredbeni okvir:

```
[C,h] = contour(x,y,z),xlabel('x1-brzina rezanja'),ylabel('x2-posmak')
```

```
set(h,'ShowText','on','TextStep',get(h,'LevelStep'))
```

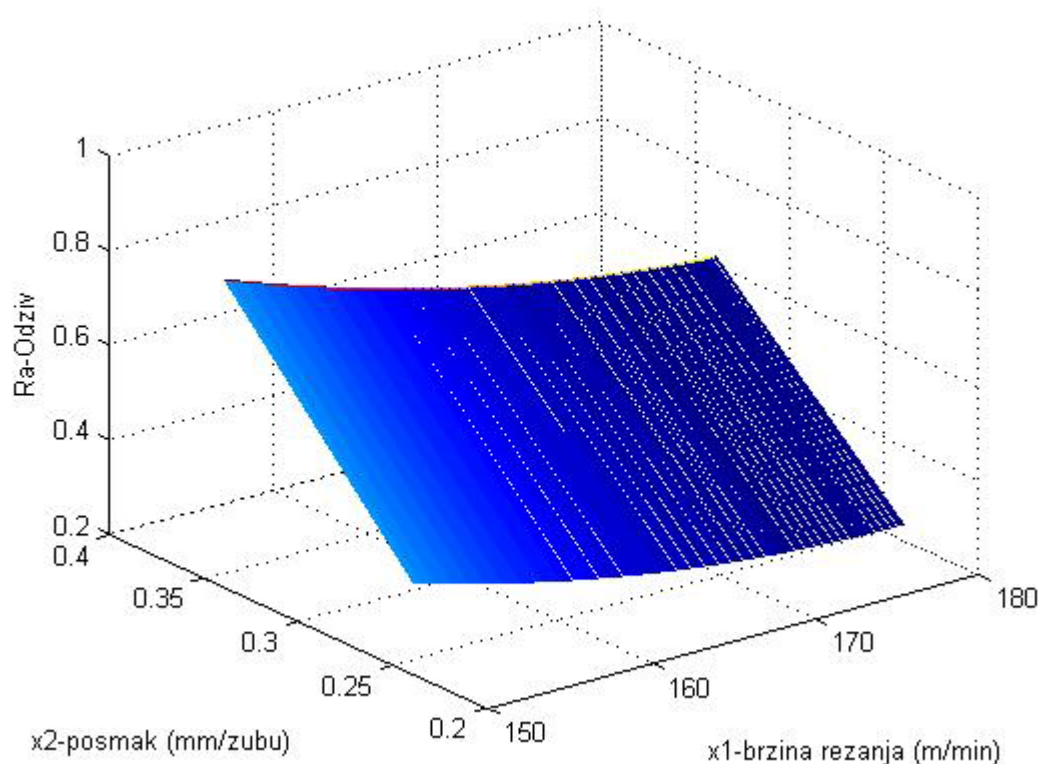
Dobiveni konturni dijagram gdje su faktori brzine rezanja i posmaka prikazani u zadanim rasponima i vrijednosti R_a za svaku njihovu kombinaciju uz konstantnu dubinu rezanja od 1,3 prikazan je slikom 46.



Slika 46: Konturni dijagram uz konstantan iznos dubine rezanja od 1,3 mm

Prostorna krivulja (slika 47.) postavljena zadanim rasponom faktora x i y pomoću naredbe meshgrid uz fiksnu vrijednost dubine rezanja od 1,3 zadana je naredbom:

```
mesh(x,y,z),xlabel('x1-brzina rezanja (m/min)'),ylabel('x2-posmak (mm/zubu)'),zlabel('Ra-Odziv')
```



Slika 47: Prostorna krivulja za fiksnu vrijednost dubine rezanja od 1,3 mm

6.2. Usporedba rezultata prema [21] i [22]

U danjem tekstu provedena je optimizacija primjera iz [21] i [22] s ciljem usporedbi dobivenih rezultata. a) Funkcija cilja prema [21] ima sljedeći izgled:

```
function [ f ] = Usporedba1( x )
```

```
f= 0.237 - 0.00175 * x(1) + 8.693 * x(2) - 0.00159 * x(3);  
end
```

Ograničenja postavljena u Matlab-ovoj M-datoteci su sljedećeg oblika:

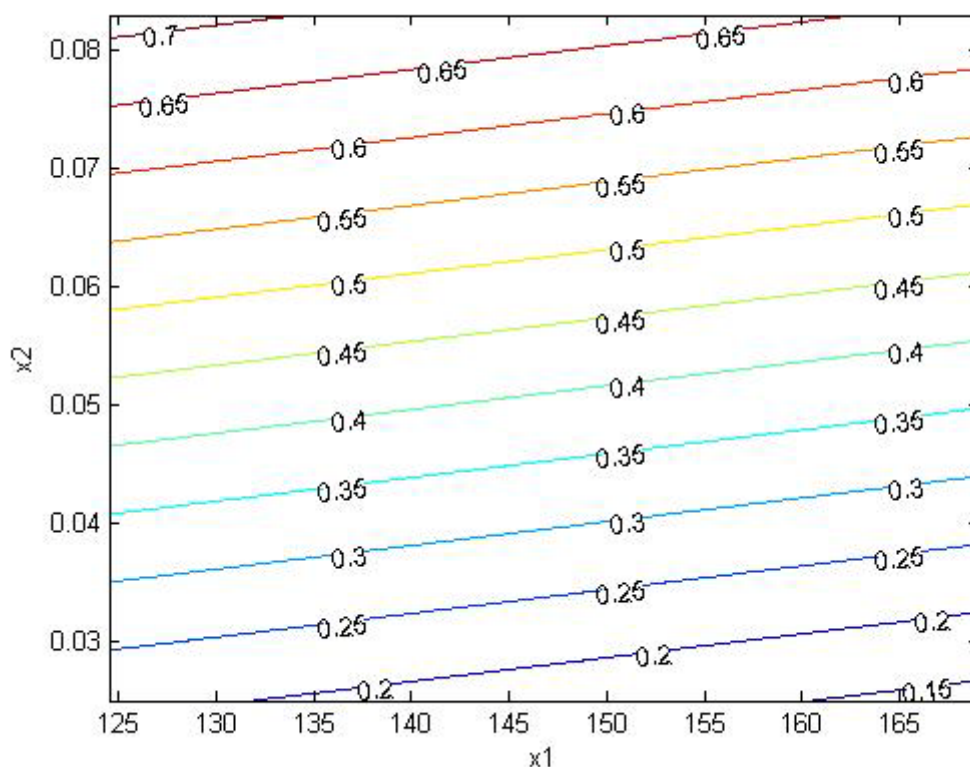
```
function [c, ceq] = OgranicenjaUsporedba1(x)
c (1) = 124.53 - x(1);
c (2) = x(1) - 169.03;
c (3) = 0.025 - x(2);
c (4) = x(2) - 0.083;
c (5) = 6.2 - x(3);
c (6) = x(3) - 14.8;
ceq = [ ];
```

, pa se provedenom optimizacijom tražeći minimum pomoću naredbe *fmincon* dobivaju rezultati:

```
>>options = optimset('Display','iter','Algorithm','active-set')
```

```
>> [x,fval]=fmincon(@Usporedba1,[0 0 0],[[],[],[],[],[],[]],@OgranicenjaUsporedba1,options)
```

```
x =                fval =
169.0300  0.0250  14.8000                0.1350
```



Slika 48: Konturni dijagram funkcije Usporedba1(x_1 , x_2) uz fiksni $x_3=14,8$

b) Funkcija cilja prema [22] ima sljedeći izgled:

```
function [ f ] = Usporedba2( x )

f= 4.829001 - 0.00632 * x(1) - 7.280413 * x(2) + 1.554342 * x(3) ...
   - 3.567335 * 0.8 + 4.314476 * 10^(-6) * x(1)^2 + 15.833074 ...
   * x(2)^2 + 20.124416 * x(3)^2 + 1.776459 * 0.8^2;
end
```

Ograničenja postavljena u Matlab-ovoj M-datoteci su sljedećeg oblika:

```
function [c, ceq] = OgranicenjaUsporedba2(x)
c (1) = 375 - x(1);
c (2) = x(1) - 675;
c (3) = 0.125 - x(2);
c (4) = x(2) - 0.35;
c (5) = 0.05 - x(3);
c (6) = x(3) - 0.2;
ceq = [ ];
```

, pa se provedenom optimizacijom tražeći minimum pomoću naredbe *fmincon* dobivaju rezultati:

```
>>options = optimset('Display','iter','Algorithm','active-set')
```

```
>> [x,fval]=fmincon(@Usporedba2,[0 0 0],[[],[],[],[],[],[],@OgranicenjaUsporedba2,options)
```

```
x =                fval =
    675.0000    0.2299    0.0500                0.1030
```

Za crtanje odzivne površine funkcije za $x_3 = 0,05$ raspon za x_1 i x_2 je definiran pomoću ograničenja i rasponom od 0,1 i 0,01, a zadan funkcijom meshgrid:

```
[x,y] = meshgrid(375:0.1:675,0.125:0.01:0.35)
```

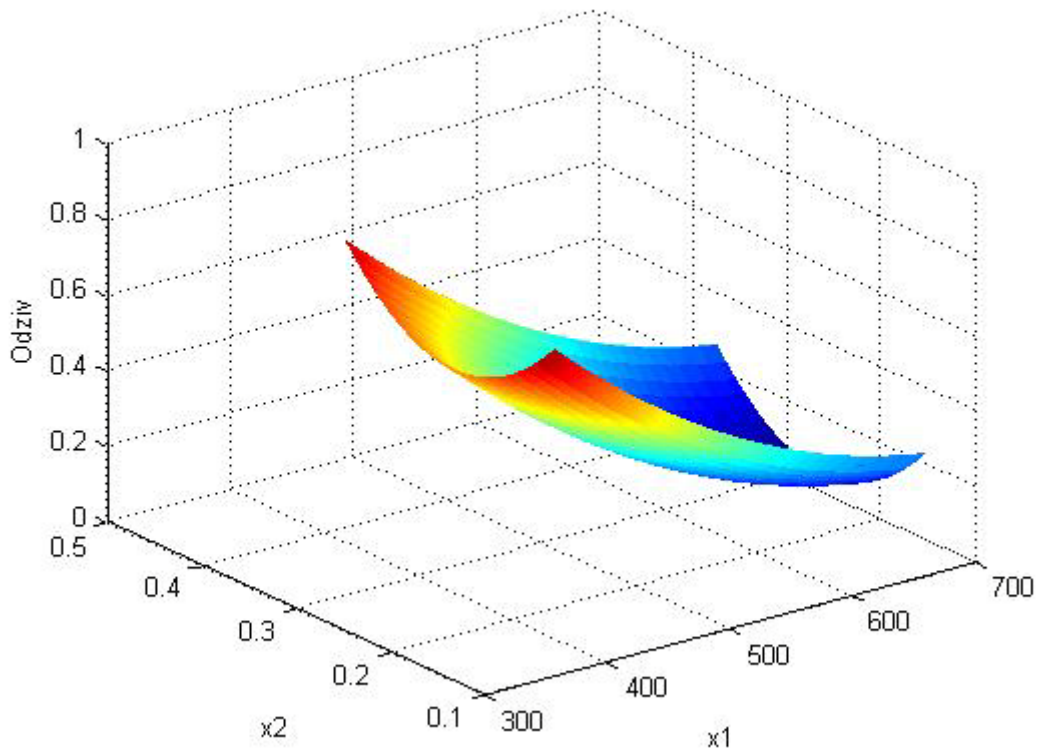
a funkcija *Usporedba2z(x,y)* prilagođena za crtanje ima sljedeći izgled:

```
function [ f ] = Usporedba2z( x,y )

f= 4.829001 - 0.00632 * x - 7.280413 * y + 1.554342 * 0.05 ...
   - 3.567335 * 0.8 + 4.314476 * 10.^(-6) * x.^2 + 15.833074 ...
   * y.^2 + 20.124416 * 0.05^2 + 1.776459 * 0.8^2;
end
```

Donjom naredbom dobije se odzivna površina funkcije Usporedba2z(x,y), slika 49:

```
>> mesh(x,y,z),xlabel('x1'),ylabel('x2'),zlabel('Odziv')
```



Slika 49: Odzivna površina funkcije Usporedba2(x1 , x2) uz fiksni x3=0,05

c) Funkcija cilja prema [21] ima sljedeći izgled:

```
function z = - funkcija(x)
z = -21.5-x(1)*sin(4*pi*x(1))-x(2)*sin(20*pi*x(2));
end
```

Ograničenja su $-3 \leq x(1) \leq 12,1$ i $4,1 \leq x(2) \leq 5,8$,a postavljena u Matlab-ovoj M-datoteci su sljedećeg oblika:

```
function [c, ceq] = Ogranicenja(x)
c (1) = -3 - x(1);
c (2) = x(1) - 12.1;
c (3) = 4.1 - x(2);
c (4) = x(2) - 5.8;
ceq = [ ];
```

, pa se provedenom optimizacijom tražeći minimum pomoću naredbe *fmincon* dobivaju rezultati:

```
>>options = optimset('Display','iter','Algorithm','active-set')
```

```
>> [x,fval]=fmincon(@funkcija,[0 0],[[],[]],[[],[]],[[],[]],@Ogranicenja,options)
```

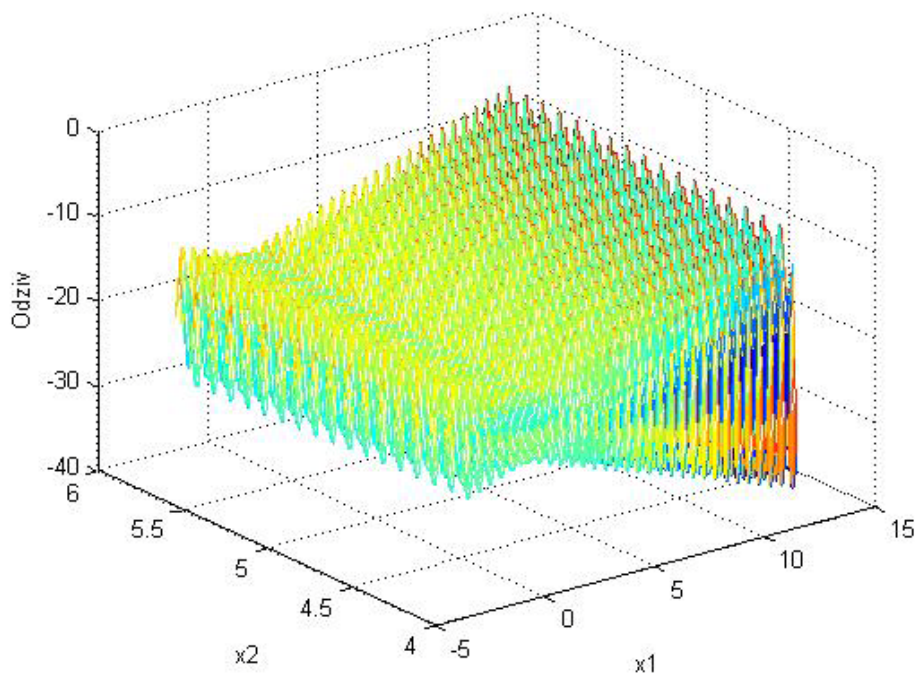
```
x =                                fval =  
12.1000  4.7251                    -37.7328
```

Zbog crtanja raspon za x_1 i x_2 je definiran pomoću ograničenja i rasponom od 0,1 i 0,01, a zadan funkcijom meshgrid:

```
[x,y] = meshgrid(-3:0.1:12.1,4.1:0.01:5.8)
```

a funkcija funkcijaz(x,y) prilagođena za crtanje ima sljedeći izgled:

```
function z=funkcijaz(x,y)  
z = -21.5-x.*sin(4*pi*x)-y.*sin(20*pi*y);  
end
```



Slika 50: Odzivna površina funkcije funkcijaz(x1, x2)

7. ZAKLJUČAK

Raznolikost softverskih paketa osigurava podršku prilikom optimizacije nelinearnih funkcija cilja, uključujući integrirane alate i pripadajuće solvere. Njihovom primjenom omogućeno je jednostavno i intuitivno rješavanje složenih matematičkih optimizacijskih modela, što je prikazano u diplomskom radu kroz različite primjere. Kroz njih je moguće primijeniti postupak i dane podloge, s ciljem optimizacija novih problemskih modela koristeći softvere Maple, Lingo, JMP i Matlab.

Također, prikazano je nastajanje matematičkih modela nelinearnog programiranja koji služe za ostvarivanje kvantitativne analize promatranih procesa, prvenstveno postupcima planiranja pokusa. Kroz praktični primjer obodnog brušenja ravne površine, istosmjernim uzdužnim načinom gdje je potrebno optimizirati faktore: glavno gibanje uzrokovano rotacijom alata, odnosno tangencijalna brzina v_g (m/s) i pomoćno gibanje uzrokovano pomicanjem alata, odnosno brzina pomicanja v_p (m/min); ostvareno je najbolje kompromisno rješenje. Ono ne postiže optimume parametara kvalitete površine završne obrade, ali daje maksimum prikazane funkcije poželjnosti gdje je ona bitan element u nastajanju najboljeg mogućeg kompromisa, poglavlje 5.4.3.

Primjenom centralnog kompozitnog plana pokusa ostvarena je nelinearna funkcija cilja koja dobro opisuje promatrani proces brušenja i slične moguće tehnološke procese, za razliku od prikazanog neučinkovitog faktorskog plana pokusa i proizašle linearne funkcijske veze promatranih faktora i traženih odziva. Ovo iznalaženje matematičkih modela R_a i P_c prilikom provedbe pokusa vizualno je prikazano korištenjem programa JMP u poglavlju 5., a ujedno i donosi primjenjivu podlogu za ostvarivanje sličnih pokusa u tehnološkim procesima.

Veliku obuhvatnost u svojoj primjeni donose softverski paketi Maple i Matlab što je sustavno prikazano kroz poglavlje 4. i 6., gdje se u izrađenim podlogama i kroz analizirane slučajeve obavlja optimizacija nelinearnih funkcija cilja. Također, odabir pravih algoritama i načina optimizacija obzirom na zadane matematičke modele ostvaruju bolja rješenja, posebno kod složenijih problema što je prikazano u usporedbi rezultata. Praktičnosti, vizualnost i intuitivnost prilikom rješavanja optimizacijskih problema je nužno potrebna, a prikazana je kroz informatičku podršku optimizaciji nelinearnih funkcija u primjerima sadržanim u diplomskom radu.

8. LITERATURA

- [1.] Griva, I., Stephen G. Nash, Sofer, A.: Linear and nonlinear optimization, Society for Industrial and Applied Mathematics, Philadelphia, 2009.
- [2.] Barković, D.: Operacijska istraživanja, Ekonomski fakultet, Osijek, 2002.
- [3.] Singiresu, S. Rao.: Engineering Optimization - theory and practice, John Wiley & Sons, Inc., New Jersey, 2009.
- [4.] Kalpić, D., Mornar, V.: Operacijska istraživanja, ZEUS, Zagreb, 1996.
- [5.] Neralić, L.: Uvod u matematičko programiranje, ELEMENT, Zagreb, 2003.
- [6.] Bronson, R., Naadimuthu, G.: Schaum's outline of theory and problems of Operations research, McGraw-Hill, New York, 1997.
- [7.] More, J. J., Wright, S. J.: Optimization Software Guide, Society of Industrial and Applied Mathematics, Philadelphia, PA, 1993.
- [8.] Salvendy, G., Grupa autora.: HANDBOOK OF INDUSTRIAL ENGINEERING: Technology and Operations Management – Third Edition, John Wiley & Sons, Inc., New York, 2001.
- [9.] Abell, L. Martha, Braselton P. James.: Maple by Example, Elsevier Academic Press, London, 2005.
- [10.] Maplesoft, www.maplesoft.com/products/maple/, 2010.
- [11.] LINDO Systems, <http://www.lindo.com/>, 2010.
- [12.] Schrage, L., *Optimization Modeling with LINDO, 6th.*, LINDO Systems Inc, Chicago, 2006.
- [13.] SAS Institute Inc.: JMP® 8 Statistics and Graphics Guide, SAS Institute Inc., Cary, NC, USA, 2009.
- [14.] SAS Institute Inc.: JMP Design of Experiments, SAS Institute Inc., Cary, NC, USA, 2005.
- [15.] SAS Institute Inc., <http://jmp.com/>, 2010.
- [16.] Jacques, G., Creighton, L.: Introduction to Design of Experiments, SAS Institute Inc., Cary, NC, USA, 2007.
- [17.] Montgomery, Douglas C.: Design and Analysis of Experiments, John Wiley & Sons,

- Inc., New York, 2008.
- [18.] Mahović, S.: Nanomjeriteljstvo – predavanja, FSB, Zagreb, 2007.
- [19.] McMahon, D.: MATLAB® Demystified, McGraw-Hill, USA, 2007.
- [20.] Bajić, D., Jozić, S., Podrug, S.: Design of experiment's application in the optimization of milling process, METALURGIJA 49 (2010) 2, 123-126
- [21.] Šprljan, M.: Diplomski rad, FSB, Zagreb, 2010.
- [22.] Šimić, I.: Diplomski rad, FSB, Zagreb, 2010.