Praktične metode optimizacije

Igor Grešovnik
Maj 2012

Verzija 3.4
(Verzija 1: Februar 2005)

1 KLASIČNE METODE ZA REŠEVANJE NELINEARNIH OPTIMIZACIJSKIH PROBLEMOV

1.1 Definicija in reševanje optimizacijskih problemov

Na splošno pomeni *optimirati* narediti neko stvar tako, da je ta najugodnejša oziroma najboljša glede na dane možnosti. Ta definicija zajema zelo različne probleme z najrazličnejših področij, kar vodi v temu ustrezno pestrost pristopov za reševanje problemov optimiranja. Znan primer je problem trgovskega potnika, pri katerem načrtujemo pot po omrežju cestnih povezav med mesti tako, da začenši v nekem mestu obiščemo vsa mesta iz predpisane množice ter se na koncu vrnemo v izhodišče ter pri tem prepotujemo čim manjšo razdaljo.

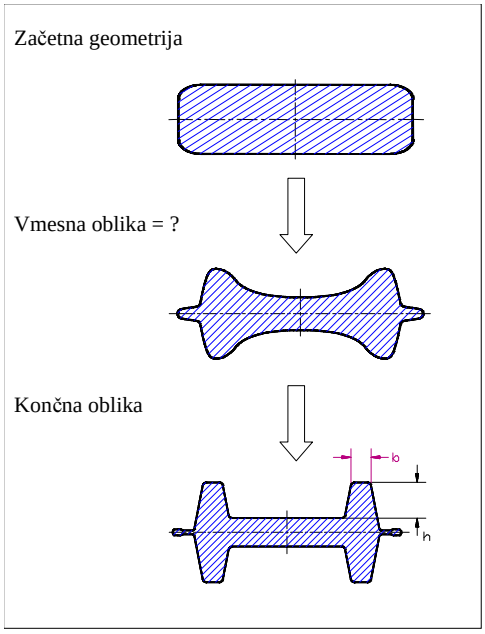
Tu se bomo omejili na situacije, pri katerih lastnosti sistema, ki ga optimiramo, določimo z danim končnim številom spremenljivih parametrov, ki se lahko zvezno spreminjajo. Predpostavili bomo tudi določene lastnosti optimizacijskih kriterijev, na primer zveznost od parametrov. Temu primerno bomo optimiranje izdelkov in procesov na splošno opredelili kot probleme nelinearnega programiranja oblike

$$\begin{array}{ll} \textit{minimiziraj} & f(\mathbf{x}), \qquad \mathbf{x} \in \mathbb{R}^n \\ \textit{ob pogojih} & c_i(\mathbf{x}) = 0, \ i \in E \\ \textit{in} & c_j(\mathbf{x}) \geq 0, \ \ j \in I \ , \end{array} \tag{1.1}$$

V vektorju **x** so zbrani optimizacijski parametri, s katerimi opišemo sistem, ki ga optimiramo. Optimizacijske kriterije zajamemo v *namenski funkciji f*, ki jo definiramo tako, da je njena vrednost čim nižja, tem boljši je sistem. Poleg ekstremnosti namenske funkcije zahtevamo, da sistem zadošča določenim omejitvam, s čimer omejimo množico naborov parametrov, med katerimi iščemo optimalnega. Omejitve delimo na enakostne, pri katerih zahtevamo, da ima določena količina odvisna od sistema točno določeno vrednost, ter neenakostne, pri katerih

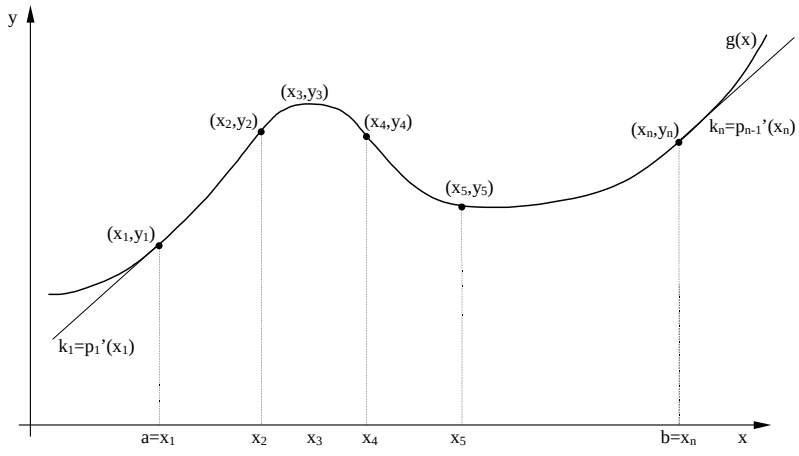
zahtevamo, da je določena količina nad predpisano mejo. Omejitve velikokrat izhajajo iz tehnoloških omejitev. Tako pri preoblikovalnih procesih pogosto upoštevamo geometrijske omejitve, ki izhajajo iz dejstva, da se orodje na začetku procesa ne more zadirati v preoblikovanec. Funkcije $c_i(\mathbf{x})$ imenujemo *omejitvene funkcije*, E in I pa sta množici indeksov enakostnih oziroma neenakostnih omejitev. Namenska in omejitvene funkcije so odvisne od odziva oz. lastnosti sistema, ki ga določajo optimizacijski parametri. Za ovrednotenje teh funkcij pri danih vrednostih parametrov lahko uresničimo sistem, ki ga obravnavamo, ter vrednosti izmerimo na resničnem sistemu. Za namene optimizacije je to navadno predrago, zato za izračun teh funkcij resnični sistem nadomestimo z numerično simulacijo, s katero aproksimiramo lastnosti oziroma odziv sistema ter preko izračunanih vrednosti izračunamo funkcije pri dani konstrukciji, ki jo definirajo optimizacijski parametri **x**.

Najpogosteje optimiramo sisteme, ki smo jih že zasnovali tako, da služijo svojemu namenu, vendar želimo izboljšati njihovo učinkovitost. Pri optimiranju se omejimo na del lastnosti, ki določajo celoten sistem. Tako lahko na primer pri večstopenjskem kovanju izdelka optimiramo obliko vmesnih orodij, ne spreminjamo pa oblike surovca, snovi, iz katere je surovec in njegove predhodne obdelave, stroja, na katerem poteka preoblikovanje in ostalih stvari (Sl. 1).

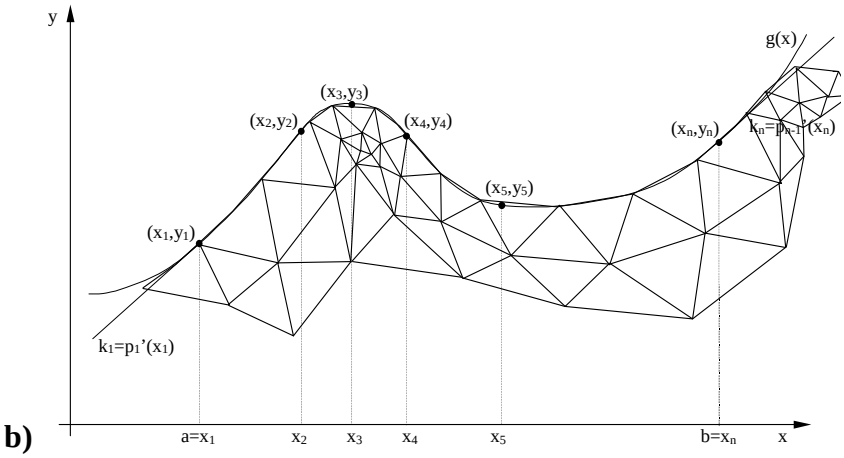


Sl. 1: shema optimiranja preoblikovalnega procesa, pri katerem lahko spreminjamo obliko orodja za vmesno stopnjo, ostali parametri procesa pa so fiksni.

Pri optimiranju oblike kot v zgornjem primeru ne moremo optimirati celotne oblike, tako da bi iskali optimalen položaj vsake točke na površini orodja. Namesto tega izberemo prikladno aproksimacijo površine, ki jo določimo s končnim številom parametrov. Ta pristop je ekvivalenten diskretizaciji kontinuumskih modelov po metodi končnih elementov, kjer je površina teles določena s koordinatami vozliščnih točk, ki pripadajo zunanjim ploskvam končnih elementov. Pri optimizaciji oblike bi zato lahko uporabili diskretizacijo s končnimi elementi in bi bili optimizacijski parametri koordinate vozlišč na površini. Tak pristop je pogosto neuporaben zaradi prevelikega števila optimizacijskih parametrov, posledica šesar je prevelika časovna zahtevnost numeričnega reševanja problema in različne numerične težave, na primer pojav lažnih lokalnih minimumov in s tem povezane neregularne rešitve optimizacijskega problema. Zato pri optimiranju oblike uporabimo metode, s katerimi definiramo površino teles z majhnim številom parametrov, vendar dovolj velikim, da lahko v okviru optimizacijskega modela znatno spremememenimo konstrukcijo sistema. V ta namen lahko uporabimo parametrično družino ploskev, kjer s koordinatami kontrolnih točk definiramo površino dela telesa, diskretizacijo numeričnega modela pa prilagodimo konstrukciji tako, da površinska vozlišča ležijo na tako definirani površini. Sl. 2 shematično prikazuje definicijo oblike v dveh dimenzijah s kubičnim zlepkom, ki ga določajo optimizacijski parametri, v tem primeru koordinate kontrolnih točk zlepka. Mrežo končnih elementov, ki določa numerični model sistema, prilagodimo tako definirani obliki. Na ta način optimizacijski parametri posredno določajo vhodne podatke numeričnega modela, ki so v tem primeru koordinate vizlišč mreže končnih elementov.



a)



b)

Sl. 2: Definicija oblike dela sistema s kubičnim zlepkom, ki deifnira rob dvodimenzionalnega objekta. Optimizacijksi parametri so ordinate kontrolnih točk zlepka $x_1=y_1$, $x_2=y_2$, …, $x_{n<}=y_n$., abscise kontrolnih točk so fiksne. Slika **a)** prikazuje zlepek pri danih parametrih, slika **b)** pa del mreže končnih elementov v bližini parametrizirane površine, ki je prilagojena tako definiranemu robu.

Postopek, pri katerim določimo odvisnost vhodnih podatkov numeričnega modela, ki ga uporabimo za analizo (aproksimacijo) odziva optimiranega sistema od optimizacijskih parametrov, pravimo *parametrizacija*. S parametrizacijo izberemo končno število optimizacijskih parametrov, ki določajo konstrukcijo sistema oziroma numeričnega modela, s katerim nadomestimo dejanski sistem v optimizacijskem postopku. Optimizacijski parametri se ne nanašajo vedno neposredno na lastnosti sistema, ampak so to lahko abstraktne spremenljivke, ki jih uvedemo samo za namene numeričnega opimiranja in od katerih so lastnosti sistema (na primer oblika orodij) posredno odvisne. Pri parametrizaciji se omejimo na tiste lastnosti sistema, ki jih lahko spreminjamo in lahko z njihovo spremembo bistveno izboljšamo sistem. Pri tem lahko kombiniramo zelo različne lastnosti kot na primer obliko orodja ali surovca, začetno temperaturo ali njeno porazdelitev, snovne lastnosti preoblikovanca in podobno.

V praksi je zelo pomembno, kako definiramo optimizacijski problem. Poleg parametrizacije (izbora števila parametrov in odvisnosti konstrukcije sistema od teh) sem spada še definicija namenske in omejitvenih funkcij. Namenska funkcija je določena s cilji, ki jih želimo doseči z optimiranju sistema. Pri optimiranju industrijskih procesov je prvobiten cilj doseči čim večji dobiček pri izdelavi naročene količine izdelkov, vendar z numeričnimi modeli težko obravnavamo celoten proizvodni proces tako celovito, da bi lahko neposredno ocenili pričakovano vrednost dobička pri dani konstrukciji procesa. Zato se pri optimiranju omejimo na kritične dele, za kateri na podlagi izkušenj vemo, kaj bi bilo potrebno izboljšati in

katere probleme odpraviti za boljše doseganje prvobitnega cilja. To lahko dostikrat jasno opredleimo s količinami, ki jih je možno oceniti na podlagi numeričnih modelov.

Primer je obraba orodja pri hladnem kovanju. Ta neposredno vpliva na življenjsko dobo orodja in s tem na število orodij, ki jih bomo rabili za izdelavo določene serije izdelkov in na število prekinitev proizvodnega procesa, ki bodo potrebne za menjavo orodja. Oboje vpliva na povečanje cene izdelave serije (oz. povprečne cene izdelka). Vemo torej, da lahko povečamo učinkovitost sistema, če je možno s spremembo tehnoloških parametrov zmanjšati hitrost obrabe na tistih mestih na orodju, ki se najbolj obrabljajo. Pri večstopenjskim kovanju lahko to poskusimo doseči s spremembo oblike vmesnega orodja (Sl. 1), s čimer spremenimo tok snovi med preoblikovanjem in s tem pogoje na stiku med orodjem in preoblikovalcem, ki najbolj vplivajo na obrabo. Izberemo ustrezno parametrizacijo obleke orodja (Sl. 2), namensko funkcijo definiramo kot hitrost obrabe na najbolj kritičnem mestu končnega orodja, ki jo pri danih vrednostih optimizacijskih parametrov izračunamo z numeričnim modelom, ter rešimo problem oblike (1.0.

Kot rezultat reševanja tako nastavljenega problema lahko pričakujemo proces s spremenjeno obliko vmesnega orodja, pri katerem je obraba v najbolj kritičnem mestu na končnem orodju manjša kot pri prvotnem procesu. Ker pa smo se pri definiciji problema ozko omejili na posamezne lastnosti procesa, se lahko pojavijo novi problemi drugje. Pri spremenjenem procesu se lahko poveča obraba na drugih mestih na orodju, ki so bila prej manj kritična, in je lahko tam večja kot je bila pred optimiranjem na najbolj kritičnem mestu. Ta problem bi načeloma odpravili tako, da bi namensko funkcijo definirali kot maksimalno hitrost obrabe po celotni površini orodja. Z rešitvijo tako definiranega problema bi morali dobiti novo konstrukcijo preoblikovalnega procesa, pri kateri je življenjska doba orodja daljša kot v začetnem procesu, pojavijo pa se lahko numerični problemi. Tako definirana namenska funkcija namreč ni nujno zvezno odvedljiva na celotnem območju v katerem zajemamo poskuse, tudi, če je obraba v dani točki zvezno odvedljiva funkcija optimizacijskih parametrov. Obrabo računamo kot časovni integral količine, v kateri nastopajo deformacije in napetosti, računanja maksimalnih vrednosti takšnih količin pa je numerično nestabilno. Te okoliščine lahko tako drastično vplivajo na učinkovitost reševanja optimizacijskega problema, da ni možno dobiti rešitve v dopustnih časovnih mejah. Za rešitev opisanega problema je potrebno definicijo namenske funkcije preoblikovati tako, da je numerično stabilna, upošteva obrabo po celotnem orodju in veliko bolj zajame vrednosti obrabe na območjih, kjer je obraba bližje maksimalni. Takšna funkcija je primerna za numerične postopke, z njeno minimizacijo pa bomo verjetno zmanjšali maksimalno obrabo in s tem podaljšali življenjsko dobo orodja, čeprav funkcija ne meri neposredno maksimalne obrabe. Navedenim lastnostim ustreza naslednja oblika namenske funkcije:

$$f(\mathbf{x}) = \sqrt[n]{\frac{\int (W(\mathbf{r},\mathbf{x}))^n dS}{S}}, \tag{1.2}$$

kjer je $W(\mathbf{r},\mathbf{x})$ indikator obrabe v točki $\mathbf{r}$ orodja za končno fazo preoblikovanja, ki je odvisen še od optimizacijskih parametrov $\mathbf{x}$ (ti določajo obliko orodja za vmesno fazo preoblikovanja), integral teče po po površini orodja, $n>>1$ pa je cela potenca. Integriranje stabilizira namensko funkcijo, potenciranje pod integralom in korenjenje integrala pa sorazmerno bolj uteži prispevke področij, kjer je obraba večja. Če n postavimo na 1, je namenska funkcija povprečje indikatorja obrabe po celotni površini, višje vrednosti n pa povzročijo, da je ta vedno bolj sorazmerna maksimalni vrednosti indikatorja obrabe.

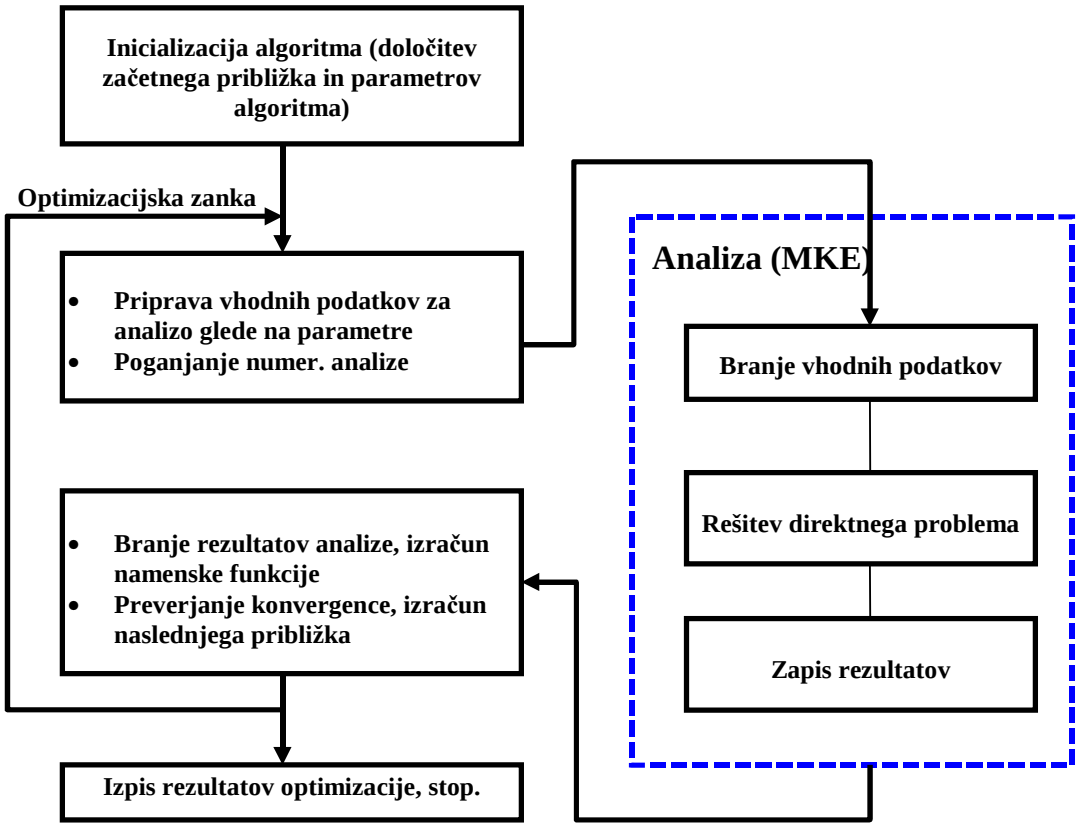
Če nastavimo optimizacijski problem na zgoraj opisani način, lahko ob podaljšanju življenjske dobe orodja, kar je bil zastavljen cilj, lahko še vedno naletimo na težave na drugih področjih. Pri spremenjeni obliki vmesnega orodja se lahko zgodi, da po končani zadnji fazi procesa orodje ni dokončno zapolnjeno in oblika izdelka odstopa od predpisane. Rezultat optimizacijskega postopka je torej proces, pri katerem je življenjska doba orodja podaljšana, izdelki pa so neuporabni. Rešitev je lahko dodatna omejitev pri definiciji optimizacijskega procesa, kjer zahtevamo, da je po končani zadnji fazi procesa orodje popolnoma zapolnjeno, kar lahko matematično izrazimo z zahtevo, da je prostornina reže med orodjem in preoblikovalcem večja ali enaka nič.

Na splošno lahko pri optimiranju kompleksnih sistemov kontroliramo samo omejeno število lastnosti sistema. Zato je pomembno, da pri definiciji optimizacijskega problema upoštevamo čim več znanja o sistemu. Predvsem je potrebno predvideti, katere lastnosti sistema se utegnejo pri optimiranju lastnosti, na katere se osredotočimo, poslabšati, ter z definicijo namenske in omejitvenih funkcij vključiti varovalke, ki to preprečijo. V praksi je optimiranje sistemov velikokrat proces, ki ga izvajamo v iteracijah. Po rešitvi optimizacijskega problema je potrebno z uporabo strokovnega znanja o sistemu preučiti spremenjeno konstrukcijo ter oceniti vplive sprememb na lastnosti sistema, ki jih v numeričnem modelu nismo kontrolirali. Če se izkaže, da smo zaradi neupoštevanja nekaterih možnosti pri optimizacijskem postopku dobili sistem, pri katerem so kakšne kritične lastnosti slabše kot pri prvotnem sistemu, moramo problem na novo formulirati in v fdefinicijo vključiti dotična spoznanja.

Poleg strokovnega znanja o sistemu je za uspešno uporabo optimizacijskih postopkov potrebno tudi dobro poznavanje lastnosti numeričnega modela sistema (natančnost, časovno zahtevnost reševanja in druge omejitve) in optimizacijskih postopkov. Pri optimiranju preoblikovalnih procesov je velikokrat poglavitna ovira za učinkovito optimiranje natančnost, ki jo lahko dosežemo z uporabljenim numeričnim modelom glede na poznavanje snovnih parametrov in modelov ter ostalih podatkov o sistemu. Zelo nelinearen odziv, veliko število optimizacijskih

parametrov in numerični šum, s katerim so obremenjeni rezultati modela so dejavniki, ki lahko vplivajo na to, da je numerično reševanje optimizacijskega problema prezahtevno in ga ne moremo izvesti v realnih časovnih okvirih. V takšnih primerih je pomembno poiskati kompromise med velikim številom parametrom in s tem možnostjo izboljšave, ki jo dopušča model, ter bolj grobim optimizacijskim modelom in zato manjšo časovno zahtevnostjo, med definicijo namenske in omejitvenih funkcij tako, da čimbolj neposredno odražajo cilje optimiranja in med modificiranimi formulacijami, ki dajo podoben rezultat, pa so lažje za optimizacijski algoritem, med doslednostjo numeričnega modela in dodatki, ki zmanjšajo časovno zahtevnost ali zgladijo odziv in podobno.

Sl. 3 prikazuje reševanje optimizacijskih problemov v primeru, ko za numerično analizo sistema uporabimo metodo končnih elementov [1]-[3]. Reševanje je razdeljeno na dva dela: na desni strani je okolje za numerično analizo, s katerim izračunamo odziv sistema pri danih optimizacijskih parametrih, na levi strani pa je optimizacijski algoritem, ki rešuje problem (1.0. Za reševanje kompleksnih problemov je ključna še povezava med obema deloma[4],[5]. Ta poskrbi za oblikovanje vhodnih podatkov za numerično analizo v skladu s trenutnimi vrednostmi parametrov, ki jih določi optimizacijski algoritem, za izvedbo numerične analize sistema v simulacijskem okolju ter za zbiranje bistvenih rezultatov in inzračun namenskih in omejitvenih funkcij in po možnosti njihovih odvodov, ki se posredujejo algoritmu.



Sl. 3: Shema reševanja optimizacijskih problemov.

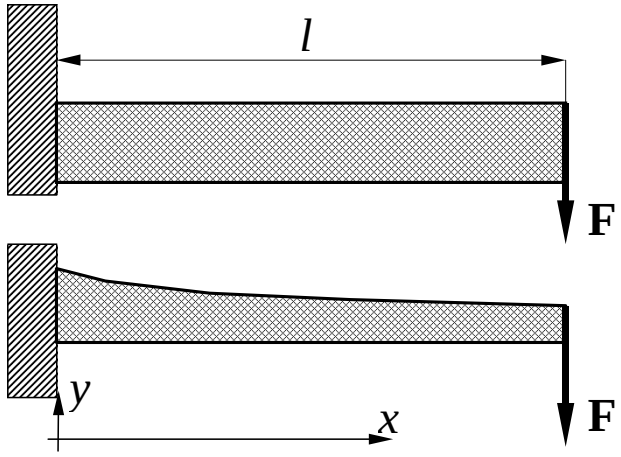
Delovanje zgornje sheme v praksi lahko ponazorimo z enostavnim primerom (Sl. 4). Optimirati želimo obliko konzole predpisane dolžine *l* in debeline *d*, ki je na eni strani vpeta v steno, na drugi strani pa obremenjena z maksimalno silo *F*. Cilj je porabiti čim manj snovi, pri čemer mora konzola ohraniti zahtevano nosilnost, želimo še, da je spodnji del konzole raven. Zahtevo po nosilnosti izrazimo s pogojema, da pri znani obremenitvi povesek konca konzole ne presega u_0 in da efektivna napetost nikjer v konzoli ne preseže meje σ_0, ki jo izberemo malo pod mejo plastičnosti.

Obliko zgornje površine parametriziramo z družino parametrično odvisnih krivulj $h(\mathbf{p}, x)$, kjer so **p** optimizacijski parametri. Problem nastavimo kot

$$\begin{aligned} &\min \int_0^l h(\mathbf{p}, x)\, d\,x \\ &u_{yl} \geq -u_0 \\ &\overline{\sigma}_{\max} \leq \sigma_0 \end{aligned} \tag{1.3}$$

ali prevedeno na obliko (1.0

$$\begin{aligned} &\min\ f(\mathbf{p}) = \int_0^l h(\mathbf{p}, x)\, d\, x \\ &c_1(\mathbf{p}) = u_{yl}(\mathbf{p}) - u_0 \geq 0 \\ &\sigma_0 - \overline{\sigma}_{\max}(\mathbf{p}) \geq 0 \end{aligned} \tag{1.4}$$



Sl. 4: Optimiranje oblike konzole, zgoraj začetna oblika, spodaj spremenjena oblika med optimizacijskim procesom.

Za reševanje zgornjega optimizacijskega problema uporabimo enega od standardnih algoritmov, npr. SQP[6],[13] J. L. Nazareth, *The Newton – Cauchy Framework – A Unified Approach to Unconstrained Nonlinear Minimisation,* Springer – Verlag, Berlin, 1994.

[14] W.H. Press, S.S. Teukolsky, V.T. Vetterling, B.P. Flannery, *Numerical Recipies in C – the Art of Scientific Computing,* Cambridge University Press, Cambridge, 1992.

[15]. Namensko funkcijo *f* pri danih parametrih lahko izračunamo z analitično integracijo, za izračun omejitvenih funkcij c_1 in c_2 pa izvedemo numerično analizo po metodi končnih elementov, s katero izračunamo napetosti in pomike v konzoli. Izračun zajema najprej pripravo vhodnih podatkov za numerično analizo glede na vrednosti parametrov **p**. V tem primeru je prikladen način sorazmeren razteg mreže končnih elementov, ki jo pripravimo vnaprej za ravno obliko, glede na funkcijo *h*(**p**, *x*). Po numerični izračunu (desna stran sheme na Sl. 3), ki je v tem primeru lahko kar linearna elastična analiza, iz rezultatov razberemo pomik na desnem koncu nosilca in največjo efektivno napetost po celotni prostornini, ter izračunamo c_1 in c_2. V programskem okolju je navadno za pripravo vhodnih podatkov numerične analize, izvedbo analize in izračun vrednosti funkcij iz (1.2) narejena posebna funkcija, ki jo kliče optimizacijski algoritem na levi strani sheme.

Nadaljevanje tega poglavja je posvečeno optimizacijskim algoritmom, torej levemu delu sheme na Sl. 3. Ob tem poudarimo, da je optimiranje ena najosnovnejčih aktivnost v inženirskih področjih. Vedno je na nek način prisotno pri konstruiranju novih stvari, vendar je pristop, ki se najpogosteje uporablja, še vedno optimiranje sistemov na roke. Pri takšnem načinu se pri zasnovi sistema na podlagi razpoložljivega znanja sklepa o tem, kakšen vpliv bodo imele določene konstrukcijske odločitve na lastnosti sistema, in se s postopnim popravljanjem, ki lahko vključuje tudi prototipne izvedbe, poskuša doseči čim boljšo konstrukcijo. Tudi pri tem načinu lahko uporabljamo numerične modele, ki omogočajo testiranje sistema brez izdelave prototipa oziroma jih uporabimo za boljši vpogled v proces. Avtomatično optimiranje z uporabo optimizacijskih algoritmov je komplementarno temu pristopu in ga uporabljamo predvsem, kadar želimo ali smo primorani izboljšati lastnosti sistema bolj, kot je to možno s poskušanjem in popravljanjem. Uporabno je predvsem v primerih, ko imamo opravka z večjim številom parametrov in kompleksnim odzivom sistema, kar je pri problemih preoblikovanja pogosta situacija.

1.2 Minimizacija brez omejitev

V tem podpoglavju obravnavamo reševanje minimizacijskega problema brez omejitev, iščemo torej najmanjšo vrednost, ki jo zavzame funkcija *n* spremenljivk na celotnem definicijskem prostoru:

$$\min\ f(\mathbf{x}),\quad \mathbf{x}\in \mathbb{R}^n \tag{1.5}$$

Navadno nas bolj zanimajo vrednosti parametrov $\mathbf{x}^*$, pri katerih funkcija zavzame najmanjšo vrednost, kar zapišemo z

$$\mathbf{x}^* = \arg\min\ f(\mathbf{x}). \tag{1.6}$$

Ločimo med globalnim in lokalnim minimumom funkcije. Funkcija *f* zavzame v $\mathbf{x}^*$ *globalni minimum*, če je

$$f(\mathbf{x}^*) \le f(\mathbf{x})\ \ \forall \mathbf{x}\in \mathbb{R}^n, \tag{1.7}$$

lokalni minimum pa, če je vrednost funkcije najmanjša v neki okolici $\mathbf{x}^*$,

$$\exists \varepsilon > 0,\ \ f\left(\mathbf{x}^*\right) \leq f\left(\mathbf{x}\right)\ \ \forall \mathbf{x}, \left\|\mathbf{x}-\mathbf{x}^*\right\| < \varepsilon\,. \tag{1.8}$$

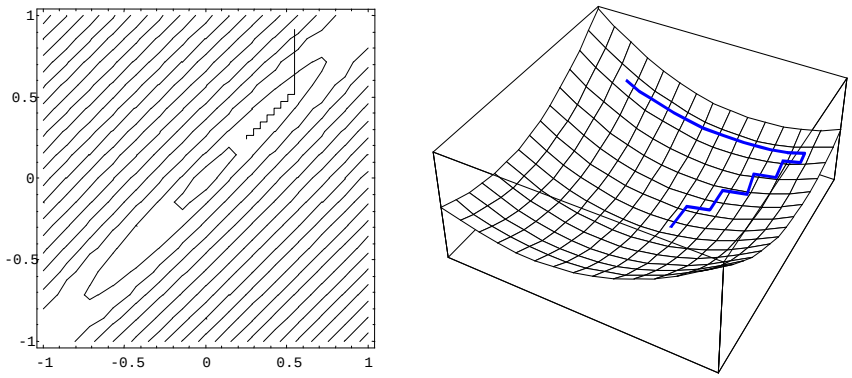
Če je v (1.0 ali (1.0 znak za striktno neenakost, govorimo o striktnem lokalnem oziroma globalnem minimumu.

1.2.1 Hevristične metode

Osnovna zasnova metod za iskanje minimuma temelji na konstrukciji zaporedja točk (približkov), kjer je vrednost funkcije v vsaki točki zaporedja nižja kot v prejšnjem približku. Takšne postopke je enostavneje izvajati v eni dimenziji, zato se zdi uporabna zamisel zaporedoma izvajati ta postopek v različnih smereh. Da bo postopek čimbolj učinkovit, poskusimo izračunavati približke za minimum funkcije v danih smereh. Minimizacijo v dani smeri $\mathbf{s}_1$ z začetnm približkom $\mathbf{x}_1$ definiramo kot reševanje problema

$$\alpha^* = \arg\min\ f\left(\mathbf{x}_1 + \alpha \mathbf{s}_1\right). \tag{1.9}$$

Zelo enostaven algoritem bi lahko zaporedoma izvajal minimizacijo funkcije v *n* neodvisnih predpisanih smereh, na primer vzporedno s koordinatnimi osi (metoda izmeničnih smeri). Čeprav se zdi zamisel dobra na prvi pogled, tak algoritem v splošnem zelo počasi konvergira Sl. 5. To si intuitivno razlagamo s tem, da algoritem ne upošteva možnosti korelacije med spremenljivkami, tako da minimizacija v eni smeri pokvari minimizacijo v ostalih smereh (podpoglavje 1.2.2).

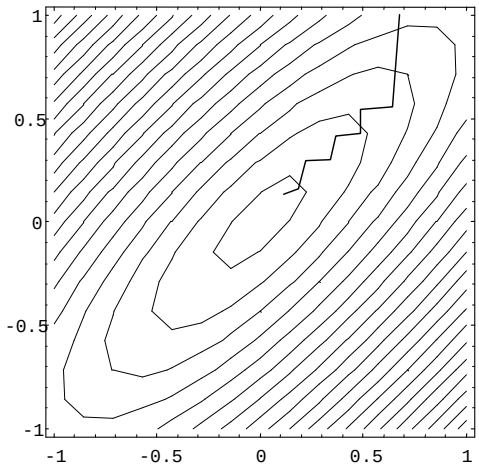


Sl. 5: Počasna konvergenca metode izmeničnih smeri.

Na prvi pogled se zdi, da bi lahko učinkovitost metode izmeničnih snovi popravili z minimizacijami v smeri gradienta funkcije, v kateri vrednost funkcije najhitreje pada (metoda najstrmejšega spusta). V splošnem se tudi ta metoda izkaže za zelo neučinkovitov praksi. Zanjo obstaja teoretični dokaz o konvergenci, vendar

lahko metoda v bližini rešitve konvergira s poljubno nizko hitrostjo linearne konvergence.

Omenjena dejstva kažejo, da je za konstrukcijo učinkovitih minimizacijskih algoritmov potrebna bolj rigulozna matematična obravnava. Moderni algoritmi imajo trdno matematično osnovo, kljub temu pa je ze učinkovitost in robustnost zelo pomembna domiselnost pri obravnavi posameznih detajlov, pri čemer lahko delovanje nekaterih hevrističnih rešitev preverimo šele z numeričnim eksperimentiranjem.



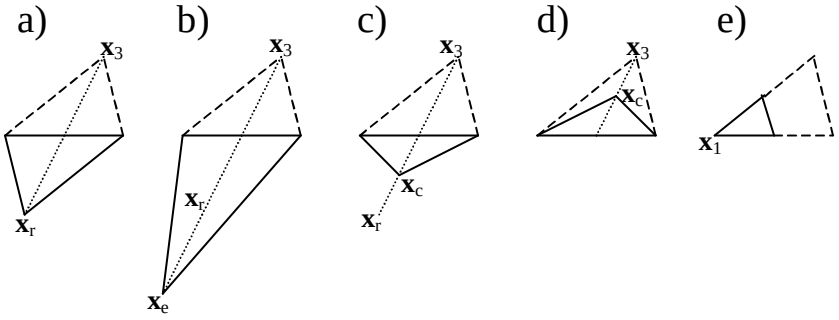
Sl. 6: Oscilacijsko vedenje metode najstrmejšega spusta.

Od hevrističnih metod tu omenimo še Nelder-Meadovo simpleksno metodo. Za to metodo ne potrefujemo odvodov namenske funkcije. Namesto posameznih zaporednih približkov vzdržujemo podatke o vrednostih funkcije v n+1 točkah, ki predstavljajo oglišča regularnega simpleksa v n dimenzionalnem vektorskem prostoru. Oglišča simpleksa sistematično premikamo glede na vrednosti namenske funkcije v the točkah. V posamezni iteraciji izvedemo naslednje poteze (Sl. 7):

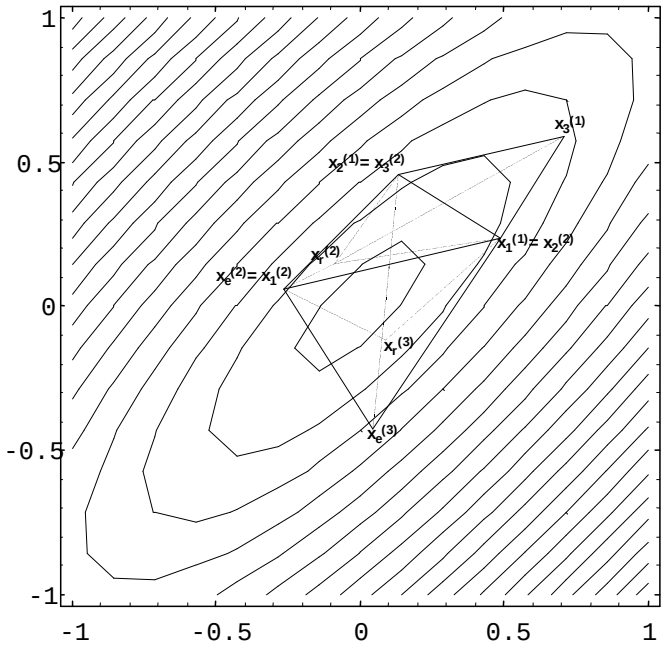
a) Zrcaljenje – oglišče z najvišjo vrednostjo f prezrcalimo čez središče preostalih oglišč, dobljeno oglišče obdržimo, če ima f nižjo vrednost kot prej.
b) Širjenje – če ke korak a) uspešen, poskusimo še s podaljšanjem za določen faktor v smeri zrcaljenja, dobljeno točko obdržimo kot oglišče simpleksa, če ima funkcija f v njem nižjo vrednost kot po koraku a).
c) Zunanje krčenje: če korak a) ni bil uspešen, prezrcaljeno oglišče pomaknemo v smeri zrcaljenja nazaj proti središču preostalih oglišč in sprejmemo dobljeno točko kot novo oglišče, če je vrednost funkcije nižje kot v ustreznem oglišču pred zrcaljenjem.

d) Notranje krčenje: če korak c) ni bil uspešen, pomaknemo oglišče z najvišjo vrednostjo funkcije proti zveznici s središčem preostalih oglišč, dobljeno točko sprejmemo, če je vrednost funkcije v njem nižja kot v začetnem oglišču.
e) Oženje: če z nobenim od korakov a) do d) nismo dobili novega oglišča z nižjo vrednostjo namenske funkcije, kot je v oglišču z najvišjo vrednostjo, potem vsa oglišča razen tistega z najnižjo vrednostjo funkcije pomaknemo proti oglišču z najnižjo vrednostjo.

a)



b)



Sl. 7: a) možne poteze v iteraciji Nelder-Meadovega simpleksnega algoritma in **b)** oris delovanja metode v praksi.

Za Nelder Meadovo metodo ne obstaja dokaz o konvergenci, teoretično je možno konstruirati celo primere, ko konvergira k točki, ki ni lokalni minimum funkcije. V praksi pa se metoda izkaže za dokaj robustno linearno konvergentno

metodo, za katero ne potrebujemo odvodov funkcije in ki se relativno dobro obnese tudi ob prisotnosti znatnega numeričnaga šuma.

1.2.2 Osnove za razvoj učinkovitejših algoritmov

Za konstrukcijo učinkovitejših algoritmov predpostavimo določene lastnosti namenske funkcije kot na primer dvakratno zvezno odvedljivost. V tem primeru lahko funkcijo razvijemo v Taylorjevo vrsto do drugega reda in jo tako lokalno aproksimiramo s kvadratnim polinomom. Iz razvoja okrog lokalnega minimuma sledijo zadostni pogoji za striktni lokalni minimum funkcije:

$$\mathbf{g}^* = 0 \tag{1.10}$$

in

$$\mathbf{s}^T \mathbf{G}^* \mathbf{s} > 0 \ \ \forall \mathbf{s}, \tag{1.11}$$

kjer smo uporabili oznake

$$\mathbf{g}(\mathbf{x}) = \nabla f(\mathbf{x}), \ \mathbf{g}^* = \mathbf{g}(\mathbf{x}^*),$$
$$\mathbf{G}(\mathbf{x}) = \nabla^2 f(\mathbf{x}), \ \mathbf{G}^* = \mathbf{G}(\mathbf{x}^*).$$

Enačba (1.0 torej pove, da je minimum stacionarna točka funkcije *f*, (1.0 pa, da je Hessova matrika drugih odvodov $\mathbf{G}$ v lokalnem minimumu pozitivno definitna.

S Taylorjevim razvojem prvega reda gradienta funkcije okrog dane točke $\mathbf{x}^{(k)}$ dobimo

$$\mathbf{g}(\mathbf{x}^{(k)} + \delta) = \mathbf{g}^{(k)} + \nabla \mathbf{g}^{(k)} \delta^{(k)} + O\left(\left\|\delta^{(k)}\right\|^2\right). \tag{1.12}$$

Če v skladu s (1.0 zgornjo enačbo postavimo na 0 in zanemarimo ostanek, dobimo enačbo točke, v kateri ima Taylorjeva aproksimacija drugega reda namenske funkcije stacionarno točko, ki je hkrati lokalni minimum, če je $\mathbf{G}$ pozitivno definitna. To lahko uporabimo v iteracijski metodi za tvorbo naslednjega iteracijskega postopka:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \delta^{(k)}; \quad \delta^{(k)} = -\left(\mathbf{G}^{(k)}\right)^{-1} \mathbf{g}^{(k)}. \tag{1.13}$$

Takšni metodi pravimo *Newtonova metoda* po analogiji z metodo za reševanje sistemov nelinearnih enačb. Metoda dobro konvergira v bližini minimuma – za splošne dvakrat zvezno odvedljive funkcije je kvadratično konvergentna, pri

kvadratnih funkcijah pa konvergira v enem koraku. Globalna konvergenca (to je konvergenca proti lokalnemu minimumu iz kateregakoli začetnega približka) brez modifikacij ni zagotovljena. To lahko popravimo z uvedbo linijskih minimizacij v smeri $\delta^{(k)}$, s čimer omejimo korak. Problem so tudi območja, kjer Hessova matrika ni pozitivno definitna. Tam lahko smer minimizacije $\delta^{(k)}$ nadomestimo recimo z negativnim gradientom ($G^{(k)}$ nadomestimo z identično matriko) ali s kakšnim drugim popravkom, ki zagotovi pozitivno definitnost. Pri praktični uporabnosti je problem tudi v tem, da je druge odvode navadno težko izračunati. Možno je numerično računanje iz vrednosti ali gradientov funkcije, kar pa je navadno predrag in numerično nestabilen postopek.

Zaradi navedenega so v praksi veliko bolj učinkoviti algoritmi, ki se izognejo uporabi drugih odvodov namenske funkcije. Pri konstrukciji takšnih metod najprej pretehtamo lastnosti pri minimizaciji kvadratnih funkcij, pri katerih Newtonova metoda konvergira v enem koraku. Zaželjeno je najti metodo, ki brez neposredne uporabe drugih odvodov minimizira kvadratno funkcijo v končnem številu korakov.

1.2.2.1 Konjugirane smeri

Za lokalne konvergenčne lastnosti pri metodah, ki temeljijo na zaporednih linijskih minimizacijah, je pomemben pojem *konjugiranih smeri*. Po definiciji so neničelni vektorji $\mathbf{s}^{(1)}, \mathbf{s}^{(2)}, ..., \mathbf{s}^{(n)}$ konjugirani glede na pozitivno definitno matriko **G**, če velja

$$\mathbf{s}^{(i)T}\,\mathbf{G}\,\mathbf{s}^{(j)} = 0 \quad \forall i \neq j\,. \tag{1.14}$$

Enostavno lahko preverimo, da so konjugirani vektorji linearno neodvisni. To pomeni, da z njimi ne moremo tvoriti linearne kombinacije z neničelnimi koeficienti, ki je enaka nič, ne obstajajo torej koeficienti α_i, tako da velja

$$\sum_{i=1}^{n} \alpha_i \mathbf{s}^{(j)} = 0, \exists i\ \alpha_i \neq 0\,.$$

Če predpostavimo $\alpha_j \neq 0$, transponirano zgornjo enačbo pomnožimo z $\mathbf{G}\mathbf{s}^{(j)}$ in ob upoštevanju konjugiranosti (1.0 dobimo

$$\alpha_j\, \mathbf{s}^{(j)\,T}\mathbf{G}\mathbf{s}^{(j)} = 0\,,$$

kar je v nasprotju z zahtevo o pozitivni definitnosti **G**, saj je po definiciji pozitivne definitnosti produkt ob koeficientu α_j večji od nič za vsak od nič različen vektor $\mathbf{s}^{(j)}$.

Pomen konjugiranih smeri za minimizacijo lahko uvidimo pri obravnavi kvadratne funkcije s pozitivno definitno Hessovo matriko. V splošni obliki lahko kvadratno funkcijo *n* spremenljivk zapišemo kot

$$q(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T\mathbf{G}\mathbf{x} + \mathbf{b}^T\mathbf{x} + c\,, \tag{1.15}$$

kjer je **G** simetrična konstantna matrika, $\mathbf{b}^T$ konstantni vektor in *c* konstantni skalar. Z odvajanjem vidimo, da je **G** matrika drugih odvodov *q*. Če je ta pozitivna, ima funkcija eno samo *stacionarno točko* $\mathbf{x}^*$, ki je rešitev enačbe

$$\nabla q(\mathbf{x}) = \mathbf{G}\mathbf{x} + \mathbf{b} = 0\,. \tag{1.16}$$

Enoličnost je posledica pozitivne definitnosti (in zato obrnljivosti) matrike **G**, $\mathbf{x}^*$ pa je tudi *globalni minimum* funkcije, kar je razvidno iz Taylorjevega razvoja drugega reda okrog $\mathbf{x}^*$:

$$q(\mathbf{x}) = q(\mathbf{x}^*) + \frac{1}{2}(\mathbf{x}-\mathbf{x}^*)^T\mathbf{G}(\mathbf{x}-\mathbf{x}^*)\,. \tag{1.17}$$

Po definiciji pozitivne definitnosti je namreč produk na desni strani večji od nič za vsak $\mathbf{x} \neq \mathbf{x}^*$ in zato $q(\mathbf{x}) > q(\mathbf{x}^*)\,\forall\,\mathbf{x} \neq \mathbf{x}^*$. Ker je $q(\mathbf{x})$ polinom drugega reda, je Taylorjev razvoj drugega reda natančno enak funkciji, zato v (1.0 ni ostanka, upoštevali pa smo tudi, da je gradient v stacionarni točki 0, zaradi česar odpade člen prvega reda. Če v (1.0 uvedemo novo neodvisno spremenljivko $\mathbf{y} = \mathbf{x} - \mathbf{x}^*$, dobimo enačbo kvadratne funkcije *q* v premaknjenem koordinatnem sistemu, kjer izhodišče sovpada s stacionarno točko funkcije. Takšna oblika je nekoliko enostavnejša kot (1.0, ker ne vsebuje linearnega člena. Pri študiju večine minimizacijskih algoritmov lahko brez škode za splošnost ugotovitev premaknemo koordinatno izhodišče, ker so algoritmi invariantni na takšno transformacijo, kar z drugimi besedami pomeni, da so vsi koraki algoritma na funkciji izraženi v transformiranih koordinatah identični tistim, ki bi se izvedli na originalni obliki funkcije, če jih izrazimo v netransformiranih koordinatah. Ravno tako ničesar ne spremeni dodajanje konstante funkciji, ki jo minimiziramo, zato lahko na primer pri študiju minimizacijskih algoritmov v formuli kvadratne funkcije (1.0 izpustimo konstantni člen.

V $\mathbb{R}^n$ lahko vedno najdemo *n* vektorjev $\mathbf{s}^{(i)}$, ki so konjugirani glede na **G**. Tvorimo jih lahko iz poljubne *n*-terice linearno neodvisnih vektorjev s postopkom, ki je analogen Gramm-Schmidtovi ortogonalizaciji, le da skalarne produkte s konstruiranimi ortogonalnimi vektorji nadomestimo s produkti **Gs** s konstruiranimi konjugiranimi vektorji.

Ker so konjugirani vektorji neodvisni, lahko *n*-terico konjugirsnih vektorjev uporabimo kot bazne vektorje v $\mathbb{R}^n$. Zato lahko poljuben vektor **x** zapišemo v obliki

$$\mathbf{x} = \mathbf{x}^{(1)} + \sum_{i=1}^{n} \alpha_i \mathbf{s}^{(i)}, \tag{1.18}$$

kjer je $\mathbf{x}^{(1)}$ poljubna izhodiščna točka. To lahko naredimo tudi za stacionarno točko $\mathbf{x}^*$ funkcije $q(\mathbf{x})$:

$$\mathbf{x}^* = \mathbf{x}^{(1)} + \sum_{i=1}^{n} \alpha^*_i \mathbf{s}^{(i)} \ . \tag{1.19}$$

Ko upoštevamo (1.0 in (1.0 ter konjugireanost vektorjev $\mathbf{s}^{(i)}$, lahko funkcijo $q(\mathbf{x})$ brez konstantnega člena zapišemo kot

$$q(\mathbf{x}) = \frac{1}{2}\left(\mathbf{x}-\mathbf{x}^*\right)^T \mathbf{G}\left(\mathbf{x}-\mathbf{x}^*\right) = \frac{1}{2}\left(\alpha-\alpha^*\right)\mathbf{S}^T\mathbf{G}\mathbf{S}\left(\alpha-\alpha^*\right) = \tilde{q}(\alpha). \tag{1.20}$$

V zgornji enačbi smo uporabili notacijo $\alpha = [\alpha_1, \alpha_2, ..., \alpha_n]$ in uvedli matriko **S**, katere stolpci so vektorji $\mathbf{s}^{(i)}$. Ker so ti konjugirani med seboj, je produkt $\mathbf{S}^T\mathbf{G}\mathbf{S}$ diagonalna matrika, katere diagonalne elemente označimo z d_i. Z uvedbo izbiro $\mathbf{s}^{(i)}$ za bazne vektorje in uvedbo novih spremenljivk α_i spremenljivk smo torej dobili nov koordinatni sistem, v katerem ima *q* enostavno obliko:

$$\tilde{q}(\alpha) = \frac{1}{2}\sum_{i=1}^{n}\left(\alpha_i - \alpha_i^*\right)^2 d_i \ . \tag{1.21}$$

V sistemu spremenljivk α je Hessova matrika *q* diagonalna, zato so spremenljivke v tem sistemu neskloplјene v smislu, da lahko $q(\alpha)$ minimiziramo z zaporednimi minimizacijami v koordinatnih smereh. Ta postopek je identičen zaporednim linijskim minimizacijam $q(\mathbf{x})$ v smereh $\mathbf{s}^{(i)}$.

Algoritmu, ki izvaja zaporedne linijske minimizacije funkcije v konjugiranih smereh, pravimo *metoda konjugiranih smeri*. Takšen algoritem minimizira kvadratno funkcijo s pozitivno definitno Hessovo matriko v največ *n* natančnih linijskih minizacijah, začenši v poljubni točki $\mathbf{x}^{(1)}$. Poleg tega vsak zaporedni približek $\mathbf{x}^{(k)}$ minimizira funkcijo na množici

$$\left\{\mathbf{x};\ \mathbf{x} = \mathbf{x}^{(1)} + \sum_{j=1}^{k} \alpha_j \mathbf{s}^{(j)}\ ,\ \alpha_j \in \mathbb{R}\right\}. \tag{1.22}$$

Na podlagi zgornje ugotovitve je bolj jasno, zakaj je lahko metoda najstrmejšega spusta lahko tako neučinkovita. Funkcija lokalno res najhitreje pada v smeri negativnega gradienta, vendar gradienti, ki jih izračunamo po vsakokratni minimizaciji, niso nujno konojugirani glede na **G**, zato minimizacija v smeri naslednjega gradienta tudi v primeru kvadratne funkcije pokvari minimalnost v prejšnjih smereh.

Iz definicije (1.0 vidimo, da so lastni vektorji **G** ortogonalni vektorji konjugirani glede na **G**. Kvadratno funkcijo lahko torej natančno minimiziramo z natančnimi linijskimi minimizacijami v smereh lastnih vektorjev **G**, vendar lahko tudi brez poznavanja lastnih vektorjev **G** tvorimo konjugirane smeri s poljubno začetno smerjo in to uporabimo pri konstrukciji minimizacijskih algoritmov.

1.2.3 Metode konjugiranih gradientov

Pri metodah konjugiranih Gradientov začnemo s smerjo negativnega gradienta

$$\mathbf{s}^{(1)} = -\mathbf{g}^{(1)} = -\nabla f\left(\mathbf{x}^{(1)}\right), \tag{1.23}$$

kjer je $\mathbf{x}^{(1)}$ začetni približek. Vsak naslednje približek $\mathbf{x}^{(k+1)}$ dobimo z natančno minimizacijo funkcije v smeri $\mathbf{s}^{(k)}$ iz $\mathbf{x}^{(k)}$. Smeri minimizacije (angleško »search directions«) $\mathbf{s}^{(k)}, k>1$ generiramo iz $-\mathbf{g}^{(k)}$ in tako, da so konjugirane glede na Hessovo matriko **G**, če metodo uporabimo na kvadratni funkciji.

Za kvadratno funkcijo (1.0 velja

$$\gamma^{(k)} = \mathbf{G}\delta^{(k)}, \tag{1.24}$$

kjer je $\gamma^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$ in $\delta^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$, za poljubna $\mathbf{x}^{(k)}$ in $\mathbf{x}^{(k+1)}$. Ker dobimo $\mathbf{x}^{(k+1)}$ z minimizacijo vzdolž $\mathbf{s}^{(k)}$ z začetno točko $\mathbf{x}^{(k)}$, sta vektorja $\delta^{(k)}$ in $\mathbf{s}^{(k)}$ vzporedna in lahko pišemo

$$\delta^{(k)} = \alpha^{(k)}\,\mathbf{s}^{(k)},$$

kjer je $\alpha^{(k)}$ skalar, ki ga določa minimum funkcije na premici $\mathbf{x}(\alpha) = \mathbf{x}^{(k)} + \alpha\mathbf{s}^{(k)}$. Iz (1.0 zato sledi, da mora veljati

$$\mathbf{s}^{(i)T}\gamma^{(j)} = 0 \quad j \neq i\,, \tag{1.25}$$

če hočemo, da so smeri $\mathbf{s}^{(k)}$ konjugirane. Poleg tega velja

$$\mathbf{s}^{(k)\,T}\mathbf{g}^{(k+1)} = 0\,, \tag{1.26}$$

saj je $\mathbf{x}^{(k+1)}$ minimum funkcije na premici in mora biti v tej točki odvod funkcije v smeri $\mathbf{s}^{(k)}$ enak nič. Z uporabo (1.0 in (1.0 dobimo

$$\begin{aligned}&\mathbf{s}^{(i)T}\mathbf{g}^{(k+1)} = \\ &\mathbf{s}^{(i)(T)}\left(\mathbf{g}^{(k+1)} - \mathbf{g}^{(k)} + \mathbf{g}^{(k)} - \mathbf{g}^{(k-1)} + \ldots - \mathbf{g}^{(i+1)} + \mathbf{g}^{(i+1)}\right) = \\ &\mathbf{s}^{(i)(T)}\left(\gamma^{(k)} + \gamma^{(k-1)} + \ldots\gamma^{(i+1)} + \mathbf{g}^{(i+1)}\right) = 0 \quad \forall i, k > i\end{aligned} \tag{1.27}$$

To pomeni, da je gradient kvadratne funkcije v točki $\mathbf{x}^{(k+1)}$ pravokoten na vse smeri linijskih minimizacij v prejšnjih korakih:

$$\mathbf{s}^{(i)T}\mathbf{g}^{(k+1)} = 0 \quad \forall k, i \leq k\,. \tag{1.28}$$

To v bistvu sledi za vsako metodo konjugiranih smeri tudi iz razprave ob (1.0.

Za to, kako dejansko konstruiramo smeri $\mathbf{s}^{(k)}$ tako, da so v primeru kvadratne funkcije konjugirane, je več možnosti. Pri Fletcher-Reevesovi metodu tvorimo $\mathbf{s}^{(k+1)}$ iz $-\mathbf{g}^{(k+1)}$ s posplošitvijo Gramm-Schmidtove ortogonalizacije z nastavkom

$$\mathbf{s}^{(k+1)} = -\mathbf{g}^{(k+1)} + \sum_{j=1}^{k} \beta^{(j)}\mathbf{s}^{(j)}\,. \tag{1.29}$$

Ko pomnožimo formulo z $\gamma^{(i)}$ in upoštevamo pogoj za konjugiranost smeri minimizacije (1.0, dobimo za koeficiente

$$\beta^{(i)} = \frac{\mathbf{g}^{(k+1)T}\gamma^{(i)}}{\mathbf{s}^{(i)T}\gamma^{(i)}} = \frac{\mathbf{g}^{(k+1)T}\left(\mathbf{g}^{(i+1)} - \mathbf{g}^{(i)}\right)}{\mathbf{s}^{(i)T}\left(\mathbf{g}^{(i+1)} - \mathbf{g}^{(i)}\right)},$$

vendar lahko formulo poenostavimo. Iz konstrukcije (enačbi (1.0 in (1.0) sledi, da $\mathbf{g}^{(k)}$ napenjajo isti posprostor kot $\mathbf{s}^{(k)}$. $\mathbf{g}^{(k+1)}$ je po (1.0 ortogonalen na podprostor, ki ga napenjajo $\mathbf{s}^{(i)}, i \leq k$, zato je ortogonalen tudi na podprostor, ki ga napenjajo $\mathbf{g}^{(i)}$ in zato na vse $\mathbf{g}^{(i)}$. V zgornji formuli je tako samo koeficient $\beta^{(k)}$ različen od nič, ob upoštevanju ortogonalnosti in (1.0 za $\mathbf{s}^{(k)}$ pa dobimo

$$\beta^{(k)} = \frac{\mathbf{g}^{(k+1)^T}\mathbf{g}^{(k+1)}}{\mathbf{g}^{(k)^T}\mathbf{g}^{(k)}}. \tag{1.30}$$

Fletcher-Reevesov algoritem pri kvadratnih funkcijah konvergira k minimumu v n korakih. Pri minimizaciji splošnih funkcij običajno po vsakih n korakih postavimo smer minimizacije na negativni gradient. Takšen algoritem je globalno konvergenten in konvergira superlinearno v bližini minimuma. V praksi se izkaže manj učinkovit kot recimo kvazi-Newtonove metode, predvsem je dokaj občutljiv na natančnost linijskih minimizacij. Prednost algoritma je enostavnost formule za izračun smeri minimizacije, ki zahteva samo $O(n)$ računskih operacij in tudi toliko spomina v vsaki iteraciji. Zaradi tega je algoritem primeren za probleme z velikim številom spremenljivk (tudi nekaj sto tisoč), posebej, če je izračun namenske funkcije in gradienta časovno nezahteven.

Konstruiramo lahko tudi metode konjugiranih smeri, ki ne potrebujejo odvodov funkcije. Po eni strnani lahko odvode vedno računamo numerično, vendar je numerično odvajanje nestabilen postopek zaradi računanja majhnih razlik vrednosti funkcije in so takšni algoritmi zelo občutljivi na šum. Powellova metoda konjugiranih smeri[6] generira konjugirane smeri v primeru kvadratne funkcije brez uporabe odvodov. Za generacijo konjugiranih smeri brez uporabe odvodov mora izvajati dodatne linijske minimizacije in konvergira v primeru kvadratne funkcije po n^2 linijskih minimizavijah. Cena za to je višja, kot bi jo plačali za numerično odvajanje pri Fletcher-Reevesovi metodi (kadar je funkcija dovolj gladka in je to možno), ker so za dobro delovanje metode potrebne relativno natančne linijske minimizacije. Pri veliko primerih iz prakse se izkaže Nelder-Meadova simpleksna metoda za bolj učinkovito, kadar niso na voljo odvodi funkcije.

1.2.4 Kvazi-Newtonove metode

Omenili smo že hitro lokalno konvergenco Newtonove metode (1.0, katere glavna težava je pomanjganje globalnih konvergenčnih lastnosti. V primeru, ko Hessova matrika ni pozitivno definitna, korak metoda sploh ni definiran, pa tudi sicer se lahko zgodi, da metoda konvergira samo, kadar je začetni približek dovolj blizu minimuma funkcije.

S prilagoditvijo Newtonove metode izpeljemo razred kvazi-Newtonovih metod, s katerimi poskušamo obdržati dobro lokalno in hkrati zagotoviti globalno konvergenco. Pri the metodah tvorimo v zaporednih iteracijah aproksimacijo $\mathbf{H}^{(k)}$ inverza Hessove matrike $\mathbf{G}^{(k)^{-1}}$, to pa namesto za neposreden izračun naslednjega približka kot v (1.0 uporabimo podobno kot pri metodah konjugiranih gradientov za izračun smeri linijske minimizacije:

$$\mathbf{s}^{(k)} = -\mathbf{H}^{(k)}\mathbf{g}^{(k)}. \tag{1.31}$$

Naslednji približek $\mathbf{x}^{(k+1)}$ je točka, ki jo dobimo z linijsko minimizacijo funkcije iz $\mathbf{x}^{(k)}$ vzdolž $\mathbf{s}^{(k)}$. V naslednji iteraciji popravimo $\mathbf{H}$ z uporabo vrednosti in gradienta funkcije v $\mathbf{x}^{(k+1)}$, da dobimo $\mathbf{H}^{(k+1)}$, ter izračunamo novo smer minimizacije $\mathbf{s}^{(k+1)}$ v skladu s (1.0. Če na začetku nimamo nobene informacije o drugih odvodih, lahko $\mathbf{H}^{(1)}$ postavimo na katerokoli pozitivno definitno matriko in navadno vzamemo kar $\mathbf{H}^{(1)} = \mathbf{I}$. Če so $\mathbf{H}^{(k)}$ pozitivno definitne, je funkcija padajoča v pripadajočih smereh $\mathbf{s}^{(k)}$.

Konstruirti moramo formulo za popravek $\mathbf{H}^{(k)}$ brez uporabe drugih odvodov funkcije tako, da ta ta ob danem začetnim približkom konvergira k $\mathbf{G}^{(k)^{-1}}$. $\mathbf{H}^{(k)}$ moramo torej dopolniti z informacijo o drugih odvodih funkcije, ki jo pridobimo z izračunom f in ∇f v dveh zaporednih približkih $\mathbf{x}^{(k)}$ in $\mathbf{x}^{(k+1)}$. Izhajamo iz enačbe (1.0, ki velja za kvadratno funkcijo. Za popravek zahtevamo, da ta enačba velja s $\mathbf{H}^{(k+1)^{-1}}$ namesto $\mathbf{G}^{(k)}$, kar da kvazi-Newtonov pogoj za popravek:

$$\mathbf{H}^{(k+1)}\gamma^{(k)} = \delta^{(k)}. \tag{1.32}$$

Zgornja enačba ne določa popravka enolično, ker predstavlja samo *n* enačb za popravek elementov matrike. Ena možnost je, da dodamo simetrično matriko ranga 1 matriki $\mathbf{H}^{(k)}$, torej

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \mathbf{u}\mathbf{u}^{T}. \tag{1.33}$$

Ko ta nastavek vstavimo v (1.0, dobimo enačbo

$$\mathbf{H}^{(k)}\gamma^{(k)} + \mathbf{u}\mathbf{u}^{T}\gamma^{(k)} = \delta^{(k)}. \tag{1.34}$$

Z upoštevanjem, da je $\mathbf{u}^{(T)}\gamma^{(k)}$ skalar, množenje matrik asociativno in množenje s skalarjem komutativno, vidimo, da mora biti $\mathbf{u}$ sorazmeren z $\delta^{(k)} - \mathbf{H}^{(k)}\gamma^{(k)}$. Ko vstavimo nastavek

$$\mathbf{u} = a\left(\delta^{(k)} - \mathbf{H}^{(k)}\gamma^{(k)}\right)$$

v (1.0, dobimo $a = 1\Big/\sqrt{\left(\delta^{(k)} - \mathbf{H}^{(k)}\gamma^{(k)}\right)^{T}\gamma^{(k)}}$ in iz tega naslednjo formulo za popravek ranga 1:

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \frac{\left(\delta^{(k)} - \mathbf{H}^{(k)}\gamma^{(k)}\right)\left(\delta^{(k)} - \mathbf{H}^{(k)}\gamma^{(k)}\right)^{(T)}}{\left(\delta^{(k)} - \mathbf{H}^{(k)}\gamma^{(k)}\right)^{(T)}\gamma^{(k)}}. \tag{1.35}$$

V primeru kvadratne funkcije s pozitivni definitno Hessovo matriko metoda ranga 1 konvergira po največ $n+1$ natančnih linijskih minimizacijah. Pri tem je $\mathbf{H}^{(n+1)} = \mathbf{G}^{-1}$ pod pogojem, da so smeri $\delta^{(1)}, ..., \delta^{(n)}$ linearno neodvisne, za kar niso potrebne natančne linijske minimizacije. Poglavitna slabost je v tem, da metoda ne ohranja pozitivne definitnosti $\mathbf{H}^{(k)}$ in da lahko pri splošnih funkcijah imenovalec v (1.0 postane nič.

Boljše metode dobimo s popravkom ranga 2, torej z nastavkom

$$\mathbf{H}^{(k+1)} = \mathbf{H}^{(k)} + \mathbf{u}\mathbf{u}^T + \mathbf{v}\mathbf{v}^T. \tag{1.36}$$

Kvazi-Newtonov pogoj (1.0 da v tem primeru enačbo

$$\delta^{(k)} = \mathbf{H}^{(k)}\gamma^{(k)} + \mathbf{u}\mathbf{u}^T\gamma^{(k)} + \mathbf{v}\mathbf{v}^T\gamma^{(k)}. \tag{1.37}$$

Tej enačbi lahko zadostimo, če postavimo **u** sorazmeren $\delta^{(k)}$ in **v** sorazmeren $\mathbf{H}^{(k)}\gamma^{(k)}$. Z reševanjem ločenih enačb za obe skupini sorazmernih vektorjev dobimo Davidon – Fletcher – Powellovo formulo oziroma DFP popravek:

$$\mathbf{H}_{DFP}^{(k+1)} = \mathbf{H} + \frac{\delta\delta^T}{\delta^T\gamma} - \frac{\mathbf{H}\gamma\gamma^T\mathbf{H}}{\gamma^T\mathbf{H}\gamma}, \tag{1.38}$$

kjer smo izpustili indekse *k*.

Drugačno formulo za popravek ranga 2 dobimo, če si zamislimo popravljanje aproksimirane **G** namesto njenega inverza. Pišemo $\mathbf{B}^{(k)} = \mathbf{H}^{(k)^{-1}}$. Kvazi-Newtonov pogoj (1.0 je izpolnjen velja za DFP formulo, vendar sedaj primeru aproksimiramo $\mathbf{B} = \mathbf{H}^{-1}$ in moramo zado v formuli zamenjati $\gamma^{(k)}$ in $\delta^{(k)}$. Tako dobimo

$$\mathbf{B}_{BFGS}^{(k+1)} = \mathbf{B} + \frac{\gamma\gamma^T}{\gamma^T\delta} - \frac{\mathbf{B}\delta\delta^T\mathbf{B}}{\delta^T\mathbf{B}\delta}.$$

Vseeno je bolje računati $\mathbf{H}^{(k+1)}$, ker se s tem izognemo reševanju sistema enačb za določitev smeri minimizacije v (1.0. Uporabimo naslednjo formulo, ki zadošča $\mathbf{B}_{BFGS}^{(k+1)}\mathbf{H}_{BFGS}^{(k+1)} = \mathbf{I}$ za zhornjo enačbo:

$$\mathbf{H}_{BFGS}^{(k+1)} = \mathbf{H} + \left(1 + \frac{\gamma^T \mathbf{H} \gamma}{\delta^T \gamma}\right) \frac{\delta \delta^T}{\delta^T \gamma} - \left(\frac{\delta \gamma^T \mathbf{H} + \mathbf{H} \gamma \delta^T}{\delta^T \gamma}\right). \tag{1.39}$$

To se imenuje Broyden – Fletcher – Goldfarb – Shanno-va oziroma BFGS formula za popravek, ustrezno kvazi-Newtonovo metoda pa navadno imenujemo kar *metoda BFGS*.

Pri kvadratnih funkcijah metoda BFGS z natančnimi linijskimi minimizacijami konvergira k minimumu v največ *n* iteracijah z $\mathbf{H}^{(n+1)} = \mathbf{G}^{-1}$, generirane smeri so konjugirane in če izberemo $\mathbf{H}^{(0)} = \mathbf{I}$, so to konjugirani gradienti. Za splošne funkcije metoda konvergira superlinearno in je globalno konvergentna za strogo konveksne funkcije, če izvajamo natančne linijske minimizacije.

Kvazi-Newtonova metoda s popravkom BFGS ali krajše metoda BFGS je danes eden najbolj uporabljanih minimizacijskih algoritmov v praksi. Numerični eksperimenti kažejo na boljše performance v primerjavi z drugimi metodami s podobnimi teoretičnimi lastnostmi, kar med drugim kaže tudi na velik pomen numeričnega eksperimentiranja pri izboru metod za uporabo v praksi. Pri tem se izkaže tudi, da metoda ni zelo občutljiva na natančnost linijskih minimizacij. V praksi je to pomembna lastnost, ker pomeni določen prihranek pri računskem času in manjšo občutljivost na numeričen šum.

1.3 Optimizacija z omejitvami

Kadar so prisotne omejitve, rešujemo splošen optimizacijski problem oblike (1.0. Problem je v splošnem precej težji od problema brez omejitev, rešujemo pa ga navadno tako, da ga prevedemo na ekvivalenten problem brez omejitev ali na serijo takšnih problemov, katerih rešitve konvergirajo k rešitvi originalnega problema.

Za obravnavo uvedemo nekaj osnovnih pojmov. Točka **x** je dovoljena oziroma možna točka (angleško »feasible point«), če so v **x** izpolnjene vse omejitve. Množica vseh dovoljenih točk je *dovoljeno območje* problema (angleško »feasible region«). Za dano omejitev iz (1.0 pravimo, da je *aktivna* v poljubni točki $\mathbf{x}'$, če je v tej tolki ustrezna omejitvena funkcija enaka nič. Množico indeksov aktivnih omejitev v tej točki označimo z

$$✌' = ✌(\mathbf{x}') = \{i;\, c_i(\mathbf{x}') = 0\}. \tag{1.40}$$

Dana omejitev je aktivna v neki točki, če je ta točka na robu dovoljenega območja te omejitve. Predvsem je pomembna množica aktivnih omejitev v rešitvi ✌*. Omejitve, ki v rešitvi niso aktivne, lahko prerturbiramo za majhno konstanto, ne da bi spremenili rešitev.

Gradient omejitvene funkcije imenujemo normalni vektor omejitve in ga označimo z $\mathbf{a}_i = \nabla c_i$. Normalne vektorje omejitev uredimo po stoplpcih v *Jacobijevo matriko* **A**.

Oglejmo si najprej problem z enakostnimi omejitvami. Recimo, da naredimo majhen *dovoljen* korak δ iz lokalnega minimuma, tako da je dobljena točka v dovoljenem območju. Iz Taylorjevega razvoja omejitvene funkcije prvega reda dobimo

$$c_i\left(\mathbf{x}^* + \delta\right) = c_i^* + \delta^T \mathbf{a}_i^* + o\left(\|\delta\|\right).$$

Ker je δ dovoljen korak, je $c_i\left(\mathbf{x}^* + \delta\right) = c_i^* = 0$. V limiti, ko krajšamo dolžino koraka proti nič, odpadejo členi višjega reda in je $\delta^T \mathbf{a}_i = 0$. Ko upoštevamo še ostale omejitve, lahko definiramo *dovoljeno smer*, za katero velja

$$\mathbf{s}^T \mathbf{a}_i^* = 0 \quad \forall i \in E\,. \tag{1.41}$$

Če je **s** dovoljena smer, je takšna tudi –**s**. Ker je $\mathbf{x}^*$ lokalni minimum, v tej točki ne obstaja dovoljena smer, v kateri *f* pada, saj bi lahko v tem primeru *f* zmanjšali s poljubno malim premikom v tej smeri. Iz tega sledi, da je

$$\mathbf{s}^T \mathbf{g}^* = 0$$

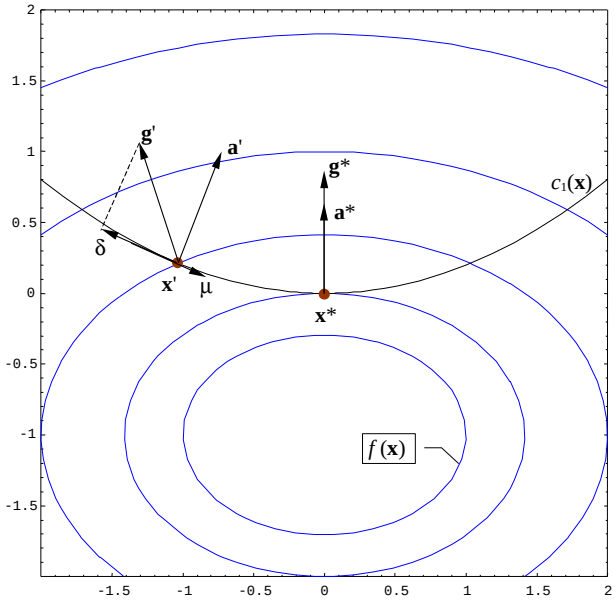
za vsako dovoljeno smer **s**. Ko upoštevamo še (1.0 in ker enačbi veljata za vsako dovoljeno smer **s**, vidimo, da mora biti $\mathbf{g}^*$ linearna kombinacija gradientov omejitvenih funkcij, zapišemo lahko torej

$$\mathbf{g}^* = \sum_{i \in E} \mathbf{a}_i^* \lambda_i^* = \mathbf{A}^* \lambda^*\,. \tag{1.42}$$

Koeficiente linearne kombinacije λ_i^* imenujemo *Lagrange-ovi množitelji* (tudi multiplikatorji, angleško Lagrange multipliers) in jih uredimo v vektor Lagrangeovih množiteljev, ki ga označimo z λ^*. Če $\mathbf{g}^*$ ne bi bil linearna kombinacija vektorjev $\mathbf{a}_i$, bi ga lahko izrazili kot

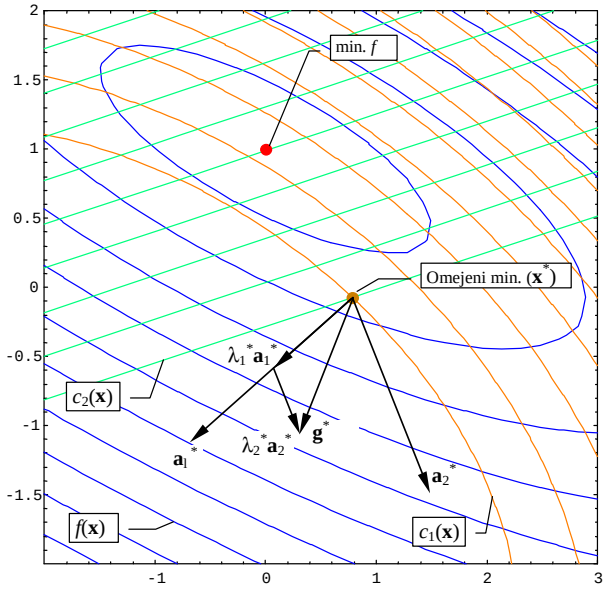
$$\mathbf{g}^* = \mathbf{A}^* \lambda^* + \mathbf{t},$$

kjer je **t** vektor pravokoten na vse $\mathbf{a}_i$. Potem bi bila **-t** dovoljena smer (kez izpolnjuje pogoj (1.0), v kateri bi funkcija *f* padala. *f* bi torej lahko zmanjšali s premikom za poljubno mali korak v tej smeri, zato $\mathbf{x}^*$ ne bi bil lokalni minimum. To je ilustrirano na Sl. 8.



Sl. 8: Ilustracija potrebnih pogojev za lokalni minimum z enakostnimi omejitvami.

Enačba (1. 0 je potreben pogoj, da je $\mathbf{x}^*$ lokalni minimum problema z enakostnimi omejitvami, in predstavlja osnoven koncept minimizacije zvezno odvedljivih funkcij z omejitvami. Podoben pogoj velja za primer, ko nastopajo tudi enakostne omejitve, le da moramo v tem primeru upoštevati samo omejitve, ki so v lokalnem minimumu aktivne. Neenakostne omejitve, ki so aktivne v rešitvi, lahko namreč zamenjamo z enakostnimi, ne da bi se spremenila rešitev problema, kar je ilustrirano na Sl. 9.



Sl. 9: Optimizacijski problem z dvema neenakostnima omejitvama. Konture omejitvenih funkcij so narisane samo v nedovoljenem območju. V rešitvi sta aktivni obe neenakostni omejitvi, ki ju lahko zato nadomestimo z enakostnima, ne da bi to vplivalo na rešitev.

Pogoj (1.0 je osnova *metode Lagrangeovih multiplikatorjev*, pri kateri rešujemo problem (1.0 z iskanjem $\mathbf{x}^*$ in λ^*, ki zadoščata

$$\mathbf{g}(\mathbf{x}) = \sum_{i \in E} \lambda_i \mathbf{a}_i(\mathbf{x})$$

in (1.43)

$$c_i(\mathbf{x}) = 0, \; i \in E \, .$$

Z *Lagrangeovo funkcijo*

$$\text{☹}(\mathbf{x}, \lambda) = f(\mathbf{x}) - \sum_i \lambda_i c_i(\mathbf{x}) \tag{1.44}$$

lahko zgornje enačbe zapišemo v obliki

$$\overline{\nabla} \text{☹}(\mathbf{x}, \lambda) = 0 \, , \tag{1.45}$$

kjer je

$$\overline{\nabla} = \begin{bmatrix} \nabla_x \\ \nabla_\lambda \end{bmatrix}.$$

gradient v $m+n$ dimenzionalnem prostoru spremenljivk $\mathbf{x}$ in λ. Metoda Lagrangeovih množiteljev ima podobne pomanjkljivosti kot Newtonova metoda – ker ne upošteva pogojev drugega reda, lahko konvergira k stacionarni točki, ki ni rešitev problema, potrebni pa so tudi dodatni postopki, ki zagotavljajo globalno konvergenco.

Če je Jacobijeva matrika omejitev ranga m, so Lagrangeovi množitelji enolično definirani in jih izračunamo z

$$\lambda^* = \mathbf{A}^{*+}\mathbf{g} \;; \quad \mathbf{A}^{*+} = \left(\mathbf{A}^{*T}\mathbf{A}^*\right)^{-1}\mathbf{A}^T .$$

$\mathbf{A}^{*+}$ je generaliziran inverz matrike $\mathbf{A}^*$. Lagrangeovi množitelji imajo tudi jasno praktično inerpretacijo. Če dano omejitev perturbiramo z dodatkom majhne konstante na desni strani, tako da se glasi $c_i(\mathbf{x}) = \varepsilon_i,\ i \in E$, ustrezen Lagrangeov množitelj pove, kako občutljiva je vrednost namenske funkcije v rešitvi na to motnjo:

$$\frac{df}{d\varepsilon_i} = \lambda_i .$$

1.3.1 Pogoji za lokalni minimum z omejitvami

V tem podpoglavju še malo natančneje opredelimo pogoje za lokalni minimum pri problemu z omejitvami (1.0. Zaradi enostavnosti privzamemo, da so v rešitvi problema gradienti omejitvenih funkcij aktivnih omejitev linearno neodvisni, da je torej rang matrike **A** enak številu njenih stolpcev.

Če je $\mathbf{x}^*$ lokalni minimum, potem obstajajo Lagrangeovi množitelji λ^*, tako da $\mathbf{x}^*$ in λ^* izpolnjujejo naslednje zahteve:

$$\begin{aligned} &\nabla_x \text{☹}(\mathbf{x}, \lambda) = 0 \\ &c_i(\mathbf{x}) = 0, \quad i \in E \\ &c_i(\mathbf{x}) \geq 0, \ i \in I \\ &\lambda_i \geq 0, \ i \in I \\ &\lambda_i c_i(\mathbf{x}) = 0 \ \ \forall i . \end{aligned} \tag{1.46}$$

S tem so določeni potrebni pogoji prvega reda za omejen lokalni minimum, ki jim pravimo tudi Kuhn-Tuckerjevi pogoji. Pogoj $\lambda_i^* c_i^* = 0$ imenujemo s posebnim imenom komplementadnostni pogoj. Po tem pogoju omejitvena funkcija in ustrezen množitelj ne moreta biti oba različna od nič, enakostnim omejitvam, ki v rešitvi niso aktivne, torej pripadajo ničelni množitelji. Primer $\lambda_i^* = c_i^* = 0$ lahko nastopi, če ima funkcija minimum brez omejitev na robu dovoljenega območja.

Označimo z **W** Hessovo matriko Lagrangeove funkcije glede na spremnljivke **x**. Iz (1.0 sledi

$$\mathbf{W} = \nabla_x^2 \text{☹}(\mathbf{x}, \lambda) = \nabla^2 f(\mathbf{x}) - \sum_i \lambda_i \nabla^2 c_i(\mathbf{x}). \tag{1.47}$$

Za ilustracijo pogojev drugega reda si poglejmo problem z izključno enakostnimi omejitvami. Lagrangeovo funkcijo razvijemo okrog minimuma v Taylorjevo vrsto do drugega reda. Privzamemeo, da so $\mathbf{a}_i^*$ linearno neodvisni in obravnavamo spremembo funkcije pri majhnem premiku δ v dovoljenem območju. Ker je premik v dovoljenem območju, je $f(\mathbf{x}+\delta) = \text{☹}(\mathbf{x}+\delta, \lambda)$. Upoštevamo šestacionarnost ☹ v $\mathbf{x}^*$ in λ^* in dobimo

$$\begin{aligned} & f(\mathbf{x}^* + \delta) = \text{☹}(\mathbf{x}^* + \delta, \lambda^*) \approx \\ & \text{☹}(\mathbf{x}^* + \delta, \lambda^*) + \delta^T \underbrace{\nabla_x \text{☹}(\mathbf{x}^*, \lambda^*)}_{=0} + \tfrac{1}{2}\delta^T \mathbf{W} \delta = . \\ & f^* + \tfrac{1}{2}\delta^T \mathbf{W} \delta \end{aligned}$$

Ker je $\mathbf{x}^*$ lokalni minimum, prirastek funkcije pri malem premiku v dovoljeni smeri ne sme biti manjši od nič, zato mora v lokalnem minimumu veljati

$$\mathbf{s}^T \mathbf{W} \mathbf{s}^* \geq 0 \tag{1.48}$$

za vsako dovoljeno smer, ki izpolnjuje pogoj

$$\mathbf{a}_i^{*T} \mathbf{s} = 0 \quad \forall i \in E. \tag{1.49}$$

To je potreben pogoj za to, da je $\mathbf{x}^*$ lokalni minimum, ki ga z besedami izrazimo kot pogoj, da mora imeti Lagrangeova funkcija nenegativno ukrivljenost vzdolž katerekoli dovoljene smeri.

Za natančnejšo splošno formulacijo zadostnih pogojev za lokalni minimum definiramo še množico striktno aktivnih omejitev

$$\mathcal{I}_+^* = \left\{ i; i \in E \vee \lambda_i^* > 0 \right\} \tag{1.50}$$

in ustreznih dovoljenih smeri[1]

$$G^* = \left\{ \mathbf{s}; \mathbf{s} \neq 0 \wedge \begin{array}{c} \mathbf{a}_i^{*T} \mathbf{s} = 0, i \in \mathcal{I}_+^* \\ \mathbf{a}_i^{*T} \mathbf{s} \geq 0, i \in \mathcal{I}^* \setminus \mathcal{I}_+^* \end{array} \right\}. \tag{1.51}$$

Izrek o zadostnih pogojih za omejen lokalni minimum pravi, da če pri $\mathbf{x}^*$ obstajajo takšni množitelji λ^*, da so izpolnjeni pogoji (1.0, in če je

$$\mathbf{s}^T \mathbf{W}^* \mathbf{s} > 0 \quad \forall \mathbf{s} \in G^*, \tag{1.52}$$

je $\mathbf{x}^*$ *strikten lokalni minimum* problema (1.0.

1.3.2 Kazenske metode

Pogosto uporabljan pristop pri reševanju problemov z omejitvami je dodajanje kazenskih členov namenski funkciji. Kazenske člene konstruiramo za vsako omejitev tako, da so blizu nič, ko je omejitev izpolnjena, in zelo hitro naraščajo, ko absolutna vrednst ustrezne omejitvene funkcije narašča.

Namensko funkcijo z dodanimi kazenskimi členi imenujemo *kazenska funkcija*. V primeru enakostnih omejitev lahko definiramo kazensko funkcijo recimo kot

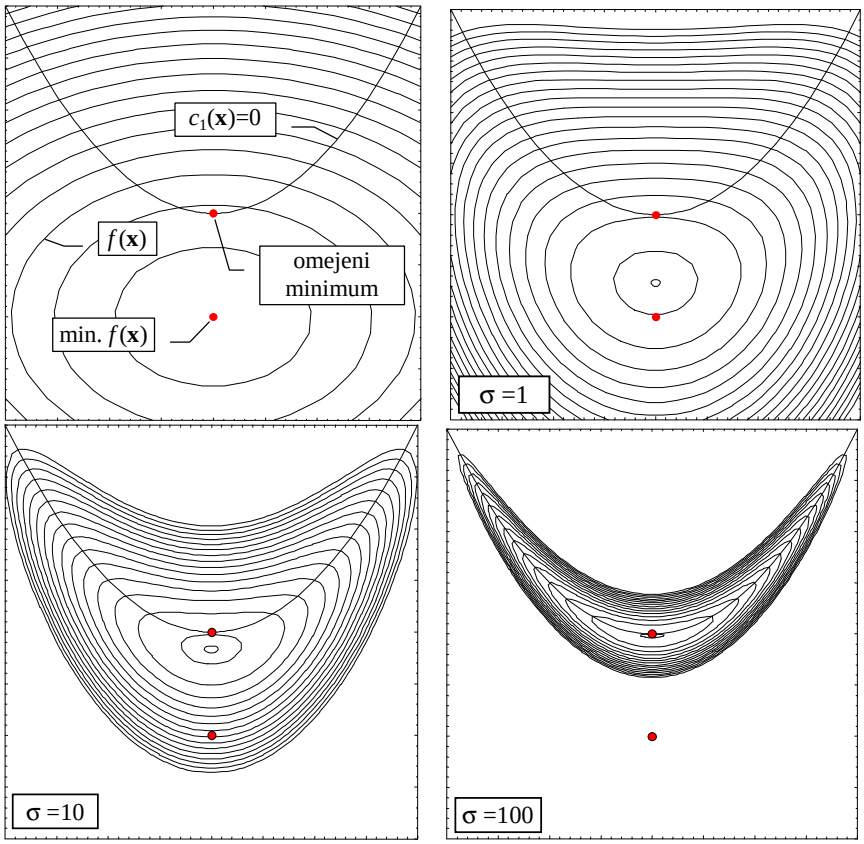
$$\phi_\sigma(\mathbf{x}) = f(\mathbf{x}) + \sigma \sum_{i \in E} (c_i(\mathbf{x}))^2 \tag{1.53}$$

Če je *kazenski koeficient* (angleško »penalty coefficient«) dovolj velik, potem minimum funkcije $\phi_\sigma(\mathbf{x})$ približno sovpada z rešitvijo originalnega problema z omejitvami (Sl. 10).

Problem je v tem, da postane pri velikih kazenskih koeficientih Hessova matrika funkcije ϕ slabo pogojena, kar zmanjša učinkovitost optimizacijskih algoritmov. V praksi zato začnemo z zmerno vrednostjo kazenskega koeficienta σ,

[1] Za to enačbo morajo veljati še določeni regularnostni pogoji, ki morajo biti izpolnjeni, da lahko definiramo dovoljene smeri z linearizacijo omejitev, in so obdelani recimo v [6]. Ti pogoji niso izpolnjeni le v nekaterih eksotičnih primerih.

poiščemo minimum (1.0, potem pa postopoma višamo kazenski koeficient ter minimiziramo kazensko funkcijo. Pri tem za začetni približek vedno vzamemo prejšnjo rešitev in tako z dobrim začetnim približkom zagotovimo boljšo konvergenco minimizacijskega algoritma. Kazenski koeficient navadno vsakič povečamo za določen faktor (recimo 10), postopek pa ustavimo, ko zaporedni približki konvergirajo glede na izbrani kriterij.



Sl. 10: Konvergenca minimumov kazenske funkcije k rešitvi problema z omejitvijo s povečevanjem kazenskega koeficienta.

Zaporedni minimumi funkcije (1.0 konvergirajo k rešitvi originalnega problema, ko σ večamo čez vse meje. V nekaterih primerih je lahko problem izbrati dovolj velik začetni kazenski koeficient, da je ϕ navzdol omejena, zato je potreben še varnostni mehanizem, s katerim to zagotovimo. Ker moramo v vsaki iteraciji rešiti zahteven minimizacijski problem, je metoda računsko zahtevna. Težko je najti dober recept za to, kakšna je dobra strategija za povečevanje kazenskega člena in kako natančno se splača minimizirati kazensko funkcijo znotraj iteracij, oboje pa je odločilno za učinkovitost. Dobra lastnost pristopa je v tem, da omogoča enostavno in neposredno prikrojitev minimizacijskih algoritmov za reševanje problemov z

omejitvami, s čimer dobimo postopek, ki sicer ni najbolj efektiven, je pa ob dovolj pazljivi izvedbi dokaj robusten.

Pri neenakostnih omejitvah se pogosto uporabljajo namesto kazenskih pregradni členi (angleško »barrier terms«), ki naraščajo v neskončnost, ko se vrednost omejitvenih funkcij bliža nič. Tipična barierna funkcija je oblike

$$\phi_\sigma(\mathbf{x}) = f(\mathbf{x}) - \sigma \sum_{i \in I} \ln(c_i(\mathbf{x})) \tag{1.54}$$

Minimumi pregradnih funkcij konvergirajo k rešitvi prvotnega problema, ko manjšamo σ proti 0. Ker pregradni členi niso definirani v območju, kjer je $c_i \leq 0$, moramo vedno zagotoviti začetni približek v dovoljenem območju, zaradi česar se za ustrezne metode v angleščini pogosto uporablja izraz »interior point methods«.

1.3.3 Zaporedno kvadratno programiranje

Za veliko množico minimizacijskih problemov z omejitvami velja za najbolj učinkovito metodo reševanja metoda zaporednega kvadratnega programiranja (angleško »sequential quadratic programming« ali kratko SQP). Pri izpeljavi medode izhajamo iz uporabe Newtonove metode za iskanje stacionarne točke Lagrangeove funkcije (1.0. S tem dobimo za izračun koraka sistem linearnih enačb

$$\left[\overline{\nabla}^2 ☹^{(k)}\right]\begin{bmatrix}\delta\mathbf{x}\\ \delta\lambda\end{bmatrix} = -\overline{\nabla} ☹^{(k)} \tag{1.55}$$

oziroma

$$\begin{bmatrix}\mathbf{W}^{(k)} & -\mathbf{A}^{(k)}\\ -\mathbf{A}^{(k)^T} & \mathbf{0}\end{bmatrix}\begin{bmatrix}\delta\mathbf{x}\\ \delta\lambda\end{bmatrix} = \begin{bmatrix}-\mathbf{g}^{(k)} + \mathbf{A}^{(k)}\lambda^{(k)}\\ \mathbf{c}^{(k)}\end{bmatrix}. \tag{1.56}$$

Zgornja iteracijska formula ni neposredno uporabna za reševanje problema (1.0, saj bi lahko tak algoritem konvergiral k stacionarni točki Lagrangeove funkcije, ki ne izpolnjuje zadostnega pogoja za rešitev. Poleg tega so s takšnim postopkom povezani podobni problemi z globalno konvergenco in rešljivostjo sistema enačb. Metodo zato modificiramo podobno kot Newtonovo metodo za reševanje problemov brez omejitev, vendar običajno uberemo malo drugačen pristop. Namesto sistema enačb, ki ga rešimo, da določimo smer linijske minimizacije, pridemo do problema

$$\begin{aligned} &\min_{\delta} && q^{(k)}(\delta) = \frac{1}{2}\delta^T \mathbf{W}^{(k)}\delta + \mathbf{g}^{(k)^T}\delta + f^{(k)} \\ &\textit{z omejitvami} && \mathbf{l}^{(k)}(\delta) = \mathbf{A}^{(k)^T}\delta + \mathbf{c}^{(k)} = 0\,. \end{aligned} \tag{1.57}$$

Zgornji problem je Taylorjeva aproksimacija originalnega problema, kjer namensko funkcijo aproksimiramo z razvojem drugega reda z upoštevanjem ukrivljenosti omejitvenih funkcij v Hessovi matriki, za omejitve pa uporabimo aproksimacijo prvega reda okrog $\mathbf{x}^{(k)}$.

Hessovo matriko Lagrangeove funkcije v zaporednih iteracijah aproksimiramo podobno kot pri kvazi-Newtonovih metodah, recimo s popravkom BFGS. Lagrangeove multiplikatorje, ki jih rabimo za izračun gradienta Lagrangeove funkcije, aproksimiramo z Lagrangeovimi množitelji linearnih omejitev v rešitvi problema (1.0.

Za rešitev problema (1.0 s kvadratno namensko funkcijo in linearnimi omejitvami obstajajo učinkovite metode, s katerimi dobimo natančno rešitev v končnem številu iteracij. V primeru enakostnih omejitev projiciramo namensko funkcijo na tangentni prostor omejitev in tako dobimo reduciran problem, ki je minimizacija kvadratne funkcije s številom spremenljivk zmanjšanim za število linearno neodvisnih omejitvenih funkcij. Malo težji je problem z neenakostnimi omejitvami, pri katerem moramo izmenično reševati ekvivalentne probleme, kjer aktivne neenakostne omejitve zamenjamo z enakostnimi. V vsaki iteraciji se z iskanjem v smeri rešitve problema z enakostnimi omejitvami premaknemo ali v rešitev ali v prvo točko, kjer kakšna prej neaktivna omejitev postane aktivna. V slednjem primeru zamenjamo nabor aktivnih omejitev in nadaljujemo s postopkom, v primeru, ko ne aktiviramo nove omejitve, pa je rešitev prvotnega problema rešitev zadnjega nadomestnega problema z enakostnimi omejitvami.

2 STOHASTIČNE METODE

In the present chapter some of the basis of nonlinear programming is outlined. This knowledge is important for understanding the practical requirements for implementation of the algorithmic part in the optimisation shell. The literature cited in this chapter is mostly related to the mathematical and algorithmic background of optimisation and less to practical implementation (except references [7], [13] and [14]). Some implementation aspects are stressed in the next chapter within a larger framework of the optimisation shell. The need for hierarchical and modular implementation, which is stated there, is partially based on the heterogeneity of optimisation algorithms evident from the present chapter.

In practice it is not always obvious which algorithm to use in a given situation. This depends first of all on the case being solved. Although the theory can offer substantial support for making the judgment, most of the literature on optimisation methods recognize the significance of numerical experimentation alongside the theoretical development. This implies a significant aspect that was borne in mind during development of the optimisation shell. The shell should not only include a certain number of algorithms, but also provide an open framework for incorporation of new algorithms and testing them on simple model functions as well as on practical problems.

Many issues important for engineering practice were not taken into account. One of them is handling multiple conflicting optimisation criteria, i.e. solving the problem stated as

$$\begin{aligned} &\textit{minimise} && [f_1(\mathbf{x}),\ f_2(\mathbf{x}),\ ...,\ f_m(\mathbf{x})] \\ &\textit{subject to} && \mathbf{x} \in \Omega\ . \end{aligned} \tag{2.58}$$

A common approach is to weight the individual criteria, which leads to the problem

$$\begin{aligned} &\textit{minimise} && f(\mathbf{x}) = w_1 f_1(\mathbf{x}) + w_2 f_2(\mathbf{x}) + ... + w_m f_m(\mathbf{x}) \\ &\textit{subject to} && \mathbf{x} \in \Omega\ , \end{aligned} \tag{2.59}$$

where w_1, …, w_m are positive weighting coefficients. The problem which arises is how to choose these coefficients. The choice is made either on the basis of experience or in an iterative process where optimisation is performed several times and coefficients are varied on the basis of the optimisation results.

Sometimes it is more convenient to designate one criterion as a primary objective and to constrain the magnitude of the others, e.g. in the following way:

$$\begin{aligned} &\textit{minimise} && f_1(\mathbf{x}) \\ &\textit{subject to} && f_2(\mathbf{x}) \le C_2, \\ & && \dots \\ & && f_m(\mathbf{x}) \le C_m, \\ & && \mathbf{x} \in \Omega. \end{aligned} \tag{2.60}$$

This approach suffers for a similar defect as weighting criteria, i.e. the solution depends on the choice of coefficients C_2, …, C_m. Attempts to overcome this problem lead to consideration of Pareto optimality[17][18] and solution of the min-max problem[17], [16].

Another important practical issue is optimisation in the presence of numerical noise. Most of the methods considered in this chapter are designed on the basis of certain continuity assumptions and do not perform well if the objective and constraint functions contain a considerable amount of noise. This can often not be avoided due to complexity of the applied numerical models and their discrete nature (e.g. adaptive mesh refinement in the finite element simulations).

A promising approach to optimisation in the presence of noise incorporates approximation techniques[19], [20]. In this approach successive low order approximations of the objective and constraint functions are made locally on the basis of sampled function values and/or derivatives. This leads to a sequence of approximate optimisation subproblems. They refer to minimisation of the approximate objective functions subject to the approximate constraints and to additional step restriction, which restricts the solution of the subproblem to the region where the approximate functions are adequate. The subproblems are solved by standard nonlinear programming methods. For approximations more data is usually sampled than the minimum amount necessary for determination of the coefficients of the approximate functions, which levels out the effect of noise. A suitable strategy must be defined for choosing the limits of the search region and for the choice of sampling points used for approximations (i.e. the plan of experiments)[19].

A common feature of all methods mentioned in this chapter is that they at best find a local solution of the optimisation problem. There are also methods which can (with a certain probability) find the global solution or more than one local solution at once. The most commonly used are simulated annealing[14],[17],[18] and genetic algorithms[17],[18]. Most of these methods are based on statistical search, which means that they require a large number of function evaluations in order to accurately locate the solution. This makes them less convenient for use in conjunction with expensive numerical simulations, except in cases where global solutions are highly desirable. Use of these techniques can also be suitable for finding global solutions of certain optimisation problems which arise as sub-problems in optimisation algorithms and in which the objective and constraint functions are not defined implicitly through a numerical simulation.

3 ZAPOREDNE APROKSIMACIJE Z OMEJENIM KORAKOM

Še ena možnost pri opzimizaciji z nelinearnimi odzivnimi funkcijami je uporaba zaporednih lokalnih aproksimacij odziva, kjer v vsakem koraku rešimo podproblem, ki ga dobimo z aproksimiranimi odzivnimi funkcijami pravega problema. Vsak tak približek osnovnega problema rešujemo na omejenem območju, kjer so aproksimacije odzivnih funkcij dovolj dober približek originalnih. Lokalne aproksimacije odzivnih funkcij tvorimo na podlagi posameznih izračunov odzivnih funkcij, ki jih sproti izračunavamo v izbranih točkah na območju, kjer rešujemo aproksimiran problem. Takšne metode so navadno zelo zahtevne za implementacijo, saj kolikor toliko dobro delujoča implementacija navadno zahteva implementacijo cele množice postopkov klasične optimizacije za reševanje posameznih podproblemov, ki nastopajo pri takšnih metodah optimizacije, poleg tega pa še vrste postopkov, ki jih pri klasičnih metodah ne srečamo. Njihove prednosti so v tem, da so uporabne tudi v primeru, ko nimamo gradientov odzivnih funkcij, ob pravilni implementaciji dokaj dobro delujejo tudi v primeru, ko so odzivne funkcije obremenjene s šumom (podobno kot stohastične metode).

Pri omenjenih metodah želimo množico točk, ki predstavljajo neko funkcijsko zvezo, zamenjati s približno funkcijo z določenimi zveznostnimi lastnostmi. Ena od možnosti je aproksimacija s polinomi določenega reda ali s trigonometrijsko vrsto. To je učinkovito, kadar potrebujemo približek na omejenem območju. V nasprotnem primeru potrebujemo veliko število členov aproksimacije, posebej, ko imamo več neodvisnih spremenljivk. Aproksimacija s polinomi je slabo pogojena, ko je število členov veliko, pri tem nastopajo tudi neželene oscilacije[21] Igor Grešovnik, Linear Approximation with Regularization and Moving Least Squares, electronic book, 2007.

[22] Igor Grešovnik, The Use of Moving Least Squares for a Smooth Approximation of Sampled Data ("Uporaba metode premičnih najmanjših kvadratov za gladko aproksimacijo vzorčenih podatkov."), Journal of Mechanical Engineering 53(2007)9, 582-598, original dosegljiv na anslovu http://www.sv-

jme.eu/scripts/download.php?file=/data/upload/2007/9/SV-JME_53%282007%2909_582-598_Gresovnik.pdf

[23],[27]. Ta problem lahko rešujemo z odsekovno polinomsko aproksimacijo [28],[29]. Pri tem pristopu se odpovemo zveznosti poljubne stopnje. V primeru več neodvisnih spremenljivk navadno potrebujemo strukturirano razdelitev območja aproksimacije.

Alternativna možnost za aproksimacijo odzivnih funkcij je metoda premičnih najmanjših kvadratov. Z metodo lahko sestavimo gladko aproksimacijo, ki se prilega podatkom na večjem območju, za ceno reševanja sistema enačb pri vsakem izračunu vrednosti aproksimacije. Ni potrebna kakršnakoli particija območja aproksimacije in za gladke funkcije lahko dosežemo poljubno natančnost aproksimacije z uporabo omejenega števila baznih funkcij, če lahko ustrezno povečujemo gostoto vzorčenja. Oris metode je podan v poglavju 3.1. Lastnosti aproksimacije so v poglavju 3.1.2.1 prikazane na primeru z analitično funkcijo ene spremenljivke.

V tem poglavju so opisane grobe osnove te družine metod. Več o metodah aproksimacije, ki so primerne za uporabo s temi metodami, je opisano v [21] Objavljen je bil tudi članek, ki del vsebne črpa iz tem opisanih v tem poglavju [22].

3.1 Opis aproksimacijske metode

3.1.1 Linearna aproksimacija po metodi najmanjših kvadratov z utežmi

Pri linearni metodi najmanjših kvadratov aproksimiramo neznano funkcijo $f(\mathbf{x})$, kjer je $\mathbf{x} \in$ ^{N}, z linearno kombinacijo *n* baznih funkcij $f_1(\mathbf{x})$, …, $f_n(\mathbf{x})$ na podlagi znanih (vzorčenih) vrednosti funkcije v določenem številu točk:

$$y_k = f(\mathbf{x}_k) + r_k\ ,\ k = 1,...,m\,. \tag{61}$$

Člen r_k predstavlja naključno napako (šum) pri merjenju ali računanju vrednosti funkcije. Aproksimacija

$$y(\mathbf{x};\mathbf{a}) = \sum_{j=1}^{n} a_j f_j(\mathbf{x}) \tag{62}$$

se mora čim bolje ujemati z vzorčenimi vrednostmi, torej

$$y(\mathbf{x}_k) \approx y_k \quad \forall k = 1, ..., m \ . \tag{63}$$

V enačbi (6) so a_j koeficienti aproksimacije, ki jih je potrebno določiti. To storimo tako, da poiščemo najboljše ujemanje v smislu najmanjših kvadratov, torej z minimizacijo naslednje funkcije koeficientov:

$$\phi(\mathbf{a}) = \sum_{k=1}^{m} w_k^2 \left(y(\mathbf{x}_k) - y_k\right)^2 = \sum_{k=1}^{m} w_k^2 \left(\left(\sum_{j=1}^{n} a_j f_j(\mathbf{x}_k) \right) - y_k \right)^2 . \tag{64}$$

V zgornji enačbi smo koeficiente a_i zapisali v vektor **a**. Nenegativne uteži w_k merijo sorazmerno pomembnost vzorčnih točk. Večje so vrednosti uteži, bolj je aproksimacija prilagojena pripadajočim vzorčenim vrednostim na račun slabšega ujemanja z vrednostmi z manjšimi utežmi.

Metoda najmanjših kvadratov z utežmi ima statističen pomen [14]. Predpostavimo, da so merske napake r_k iz (5) porazdeljene normalno z znanimi standardnimi deviacijami σ_k in je (6) pravilen model za $f(\mathbf{x})$, ter postavimo v enačbi (8) $w_k = 1/\sigma_k$. Potem dobimo z minimizacijo funkcije $\phi(\mathbf{a})$ iz (8) tiste vrednosti koeficientov **a**, pri katerih je "verjetnost", da pri nekem poskusu izmerimo vrednosti $\{y_k\}$ iz (7), največja. Pri tem se izraz "največja verjetnost" nanaša na maksimum verjetnostne gostote za spremenljivke $\{y_k\}$. Čeprav porazdelitve merskih napak pogosto niso normalne in uporabljeni modeli niso natančni, je uporaba najmanjših kvadratov pri aproksimaciji podatkov običajna in se izkaže za primerno v veliko situacijah, ko ne razpolagamo s fizikalno utemeljenimi modeli.

Minimizacijo $\phi(\mathbf{a})$ lahko izvedemo z iskanjem stacionarne točke, zato zahtevamo

$$\frac{d\,\phi(\mathbf{a})}{d\,a_i} = 2\sum_{k=1}^{m} \left(w_k^2 \left(\sum_{j=1}^{n} a_j f_j(\mathbf{x}_k) - y_k \right) f_i(\mathbf{x}_k) \right) = 0 \quad \forall i = 1, ..., n \, . \tag{65}$$

Dobimo linearen sistem enačb za koeficiente **a**,

$$\mathbf{C}\mathbf{a} = \mathbf{b} \, , \tag{66}$$

kjer so

$$C_{ij} = \sum_{k=1}^{m} {w_k}^2 f_i(\mathbf{x}_k) f_j(\mathbf{x}_k) \tag{67}$$

in

$$b_i = \sum_{k=1}^{m} {w_k}^2 f_i(\mathbf{x}_k) y_k \ . \tag{68}$$

3.1.2 Premični najmanjši kvadrati

Izbira baznih funkcij je odločilna za to, kako natančno lahko aproksimacijo prilagodimo podatkom. Če nimamo primernega fizikalnega modela, pogosto vzamemo množico monomov do določene stopnje, kar je utemeljeno s Taylorjevim izrekom [24]. Za dobro aproksimacijo funkcije na večjem območju je potrebno ustrezno povečati število členov. Posebej pri večjem številu neodvisnih spremenljivk je lahko težko določiti primerno število baznih funkcij. Pri uporabi aproksimacijskih polinomov obstaja nevarnost neželenih oscilacij [28], v splošnem pa se lahko pojavijo težave s slabo pogojenostjo sistema (10) ali celo s singularnostjo matrike, ko so bazne funkcije linearno odvisne na podani množici točk [25].

Zgoraj navedene težave lahko olajšamo s prostorsko omejitvijo vpliva vzorčenih vrednosti, tako da le-te bistveno vplivajo na vrednost approksimacije le v neki okolici ustreznih vzorčnih točk. Da ohranimo zveznost in zagotovimo zmožnost prilagoditve na celotnem območju, ki nas zanima, obdržimo obliko (6), vendar s spremenljivimi koeficienti, ki so zvezno odvisni od neodvisnih spremenljivk. Aproksimacijo zapišemo kot

$$y(\mathbf{x}) = \sum_{j=1}^{n} a_j(\mathbf{x}) \ f_j(\mathbf{x}) \ . \tag{69}$$

Neznane koeficiente **a**(**x**) moramo posebej izračunamo v vsaki točki, kjer izračunamo vrednost aproksimacije. Pri metodi premičnih najmanjših kvadratov [30] v ta namen uvedemo utežne funkcije, ki padajo z razdaljo od pripadajočih vzorčnih točk. Koeficiente **a**(**x**) izračunamo v vsaki točki **x**, kjer računamo vrednost aproksimacije, z reševanjem sistema enačb, ki je ekvivalenten sistemu (10) do (12), pri čemer so uteži w_k(**x**) odvisne od točke izračuna:

$$\begin{aligned}
&\mathbf{C}(\mathbf{x})\,\mathbf{a}(\mathbf{x}) = \mathbf{b}(\mathbf{x}),\\
&C_{ij}(\mathbf{x}) = \sum_{k=1}^{m} w_k(\mathbf{x})^2\, f_i(\mathbf{x}_k)\, f_j(\mathbf{x}_k)\ .\\
&b_i(\mathbf{x}) = \sum_{k=1}^{m} w_k(\mathbf{x})^2\, f_i(\mathbf{x}_k)\, y_k
\end{aligned} \tag{70}$$

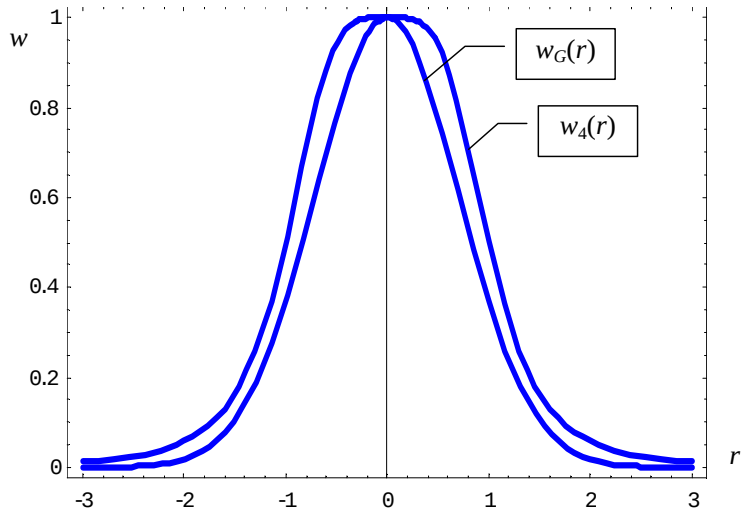
Utežne funkcije $w_k(\mathbf{x})$ morajo v vseh smereh monotono padati z razdaljo od pripadajočih vzorčnih točk $\mathbf{x}_k$. Z gladkimi utežnimi funkcijami zagotovimo gladko aproksimacijo. V tem prispevku uporabimo naslednjo splošno obliko utežnih funkcij:

$$w_k(\mathbf{x},\mathbf{d}) = w\left(\left\| \mathbf{D}^{-1}(\mathbf{x}-\mathbf{x}_k) \right\|_2 \right)\ . \tag{71}$$

V enačbi (15) je **D** matrika, ki določa efektivni doseg vpliva vzorčnih točk v različnih koordinatnih smereh. Konkretno obliko $w(r)$ lahko pripravno prilagodimo namenu. Primera sta Gaussova in racionalna oblika (Sl. 1),

$$\begin{aligned}
&w_G(r) = e^{-r^2},\\
&w_p(r) = \frac{1}{1+|r|^p},\ p = 2, 3, 4, \ldots
\end{aligned}\ . \tag{72}$$

Možna je tudi uporaba funkcij s končnim nosilcem, s čimer popolnoma izločimo vpliv oddaljenih točk na vrednost aproksimacije v dani točki.

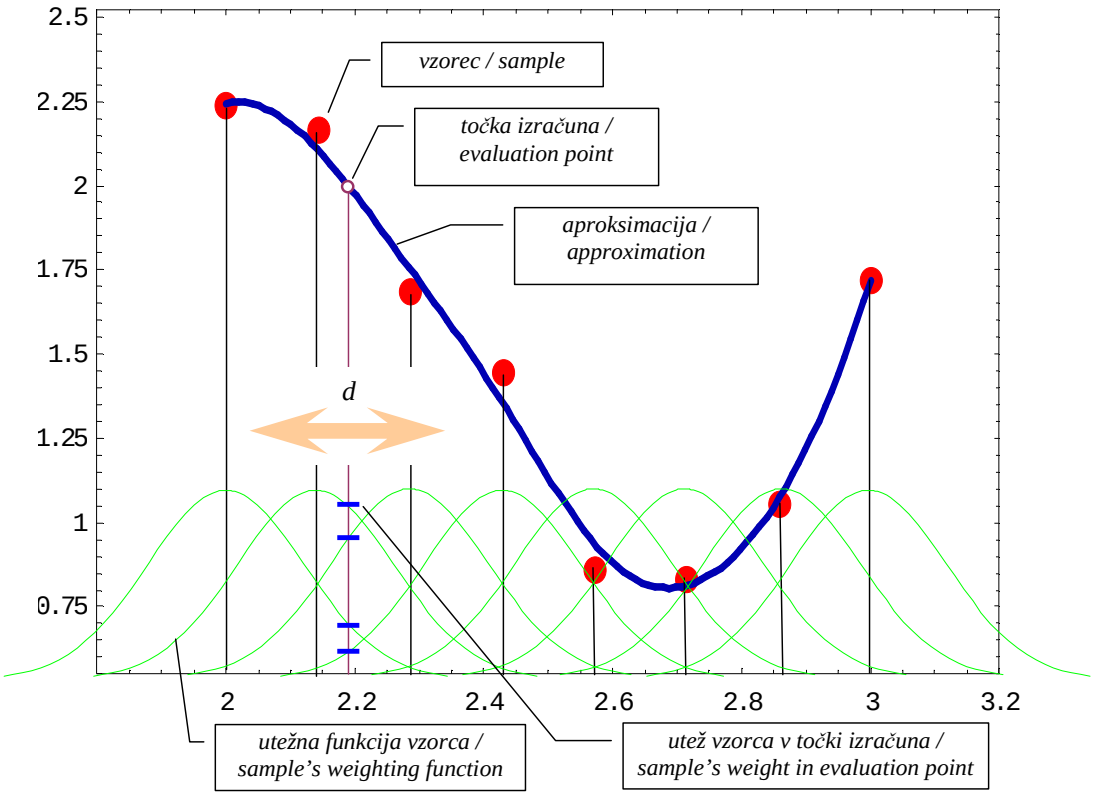


Sl. 1: Obliki utežnih funkcij $w_G(r)$ in $w_4(r)$.

Ker prostorska odvisnost uteži $w_k(\mathbf{x})$ in posledično $a_j(\mathbf{x})$ omogočata lokalno prilagajanje aproksimacije vzorčenim podatkom, ni potrebna velika množica baznih funkcij $f_j(\mathbf{x})$, če želimo funkcijo aproksimirati po večjem območju. Pri veliki gostoti vzorčnih točk in malih napakah lahko zadoščajo že monomi do prvega reda. Uporaba kvadratične polinomske baze je glede na izkušnje primerna za veliko število različnih problemov. Cena za majhno število funkcij $f_j(\mathbf{x})$, ki določa število enačb v sistemu (14), je, da je aproksimacijska funkcija podana implicitno in moramo zato rešiti sistem enačb pri vsakem izračunu vrednosti aproksimacije. Metoda je zato manj primerna, kadar je potrebno izračunati vrednost aproksimacijske funkcije v veliko točkah.

Efektivni dosegi vpliva d_i so odločilni parametri aproksimacije, katerih izbira je tesno povezana z izbiro baze in lastnostmi aproksimirane funkcije. Zelo grobo vodilo pri izbiri d_i je, da naj efektivni doseg vpliva ne bo veliko večji od velikosti območja, na katerem lahko linearna kombinacija baznih funkcij s konstantnimi koeficienti dobro aproksimira funkcijo. Po drugi strani mora biti efektivni doseg v prisotnosti šuma večji, da se pri aproksimaciji izravna vpliv naključnih napak.

Sl. 2 shematično prikazuje metodo premičnih najmanjših kvadratov, kjer aproksimiramo osem vrednosti s polinomskimi baznimi funkcijami drugega reda {1, x, x^2}. Utežne funkcije pripadajoče vzorčnim točkam so narisane v spodnjem delu slike. Za izbrano točko izračuna aproksimacije so označene vrednosti uteži za vplivne vzorčne točke.



Sl. 2: Shematičen prikaz aproksimacije po metodi premičnih najmanjših kvadratov.

3.1.2.1 Študija: aproksimacija analitične funkcije

V tem poglavju je obravnavana uporaba aproksimacije po metodi premičnih najmanjših kvadratov na osnovi podatkov, ki so dobljeni z vzorčenjem analitične funkcije z dodano naključno napako:

$$f(x)=\sin(4x)+0.5e^{0.5x}+2R(rnd()-0.5)\ , \tag{73}$$

kjer je *rnd*() enakomerno porazdeljena naključna spremenljivka na intervalu [0,1] in *R* konstanta, ki določa raven šuma. Funkcijo vzorčimo na intervalu $[x_l, x_r]=[0, 5]$ v različnih številih *m* enakomerno oddaljenih točk. Študiramo lastnosti aproksimacije glede na raven šuma *R*, število vzorčnih točk *m* in efektivni vplivni doseg *d*. Uporabimo kvadratično polinomsko bazo:

$$f_1(x)=1,\ f_2(x)=x,\ f_3(x)=x^2\ . \tag{74}$$

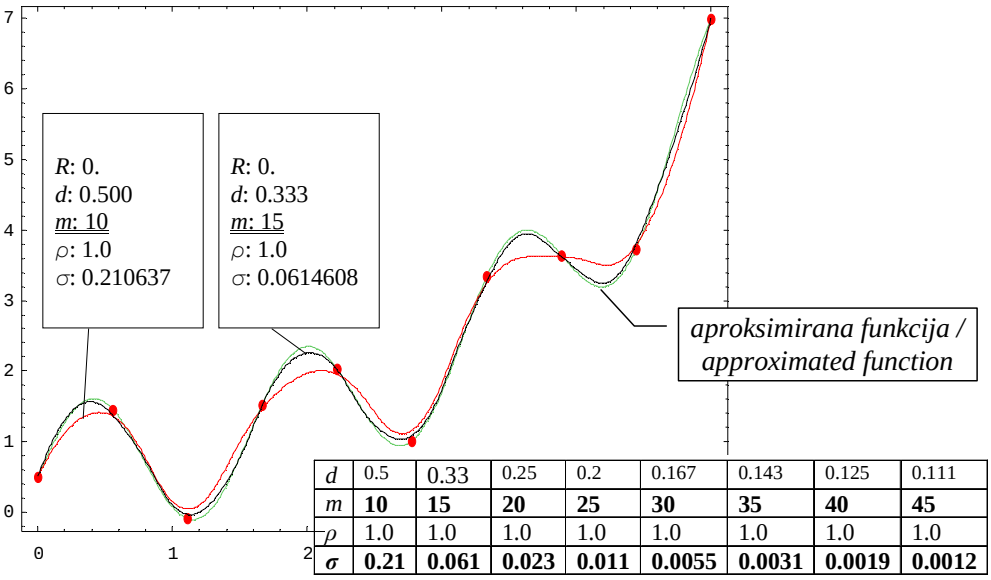
Za primerjavo rezultatov uvedemo mero za normalizirano gostoto vzorčenja, ki določa število vzorčnih točk na efektivni doseg *d*:

$$\rho = m\frac{d}{x_r - x_l} \ . \tag{75}$$

Kot mero za aproksimacijo napake uporabimo koren iz povprečne vrednosti vsote kvadratov odstopanj aproksimacije od *f* v N_e=600 točkah:

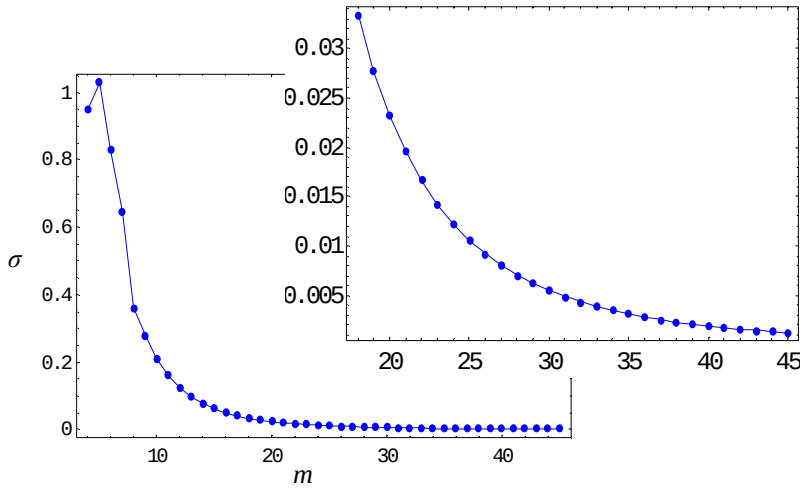
$$\sigma = \sqrt{\frac{1}{N_e}\sum_{i=1}^{N_e}\left(\, y(x_i) - f(x_i)\right)^2} \ ; \ x_i = x_l + (i-1)\frac{x_r - x_l}{N_e - 1} \ . \tag{76}$$

Najprej študiramo aproksimacijo brez prisotnosti šuma. Aproksimacijo izračunamo za različna števila vzorčnih točk, kjer izberemo efektivni doseg tako, da je normirana gostota vzorčenja konstantna in sicer $\rho = 1$. Sl. 3 prikazuje aproksimacijo z 10 in 15 točkami, ki ju primerjamo z aproksimirano funkcijo. Aproksimacijske napake so za različna števila točk prikazana v tabeli, ki je priložena k Sl. 3, ter na Sl. 4. Ko presežemo določeno število točk, napaka monotono pada z *m*. Tako je zaradi tega, ker efektivni doseg pada z naraščajočim *m*, zaradi konstantnega ρ pa se število vzorčnih točk, ki so v povprečju vsebovane znotraj efektivnega dosega, ne spreminja. Iz enačb (14), (15) in (16) je razvidno, da je vrednost aproksimacije v dani točki **x** približno takšna, kot če bi aproksimirali funkcijo na intervalu reda velikosti $2d$ po navadni metodi najmanjših kvadratov z ustrezno razporeditvijo vzorčnih točk, ustreznimi vrednostmi uteži in uporabljenimi baznimi funkcijami (18). Pričakujemo lahko, da se napaka takšne aproksimacije v točki **x** manjša, ko manjšamo dolžino intervala.



d	0.5	0.33	0.25	0.2	0.167	0.143	0.125	0.111
m	**10**	**15**	**20**	**25**	**30**	**35**	**40**	**45**
ρ	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0
σ	**0.21**	**0.061**	**0.023**	**0.011**	**0.0055**	**0.0031**	**0.0019**	**0.0012**

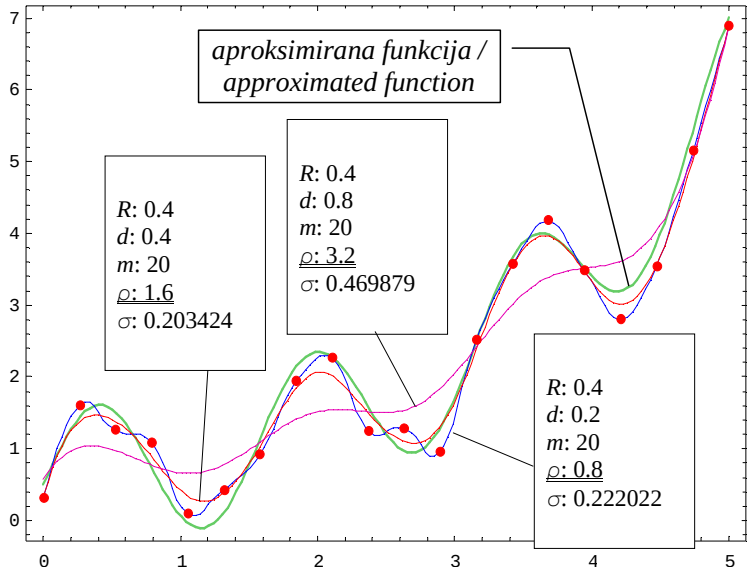
Sl. 3: Aproksimacija na podlagi vzorčenih vrednosti brez šuma. Izmerki so narisani le za primer z 10 točkami, aproksimacija pa za 10 in 15 točk.



Sl. 4: Odvisnost napake aproksimacije σ od števila vzorčnih točk za vzorčenje brez šuma in pri konstantnem ρ=1.0. Dodan je bolj podroben prikaz pri večjih m.

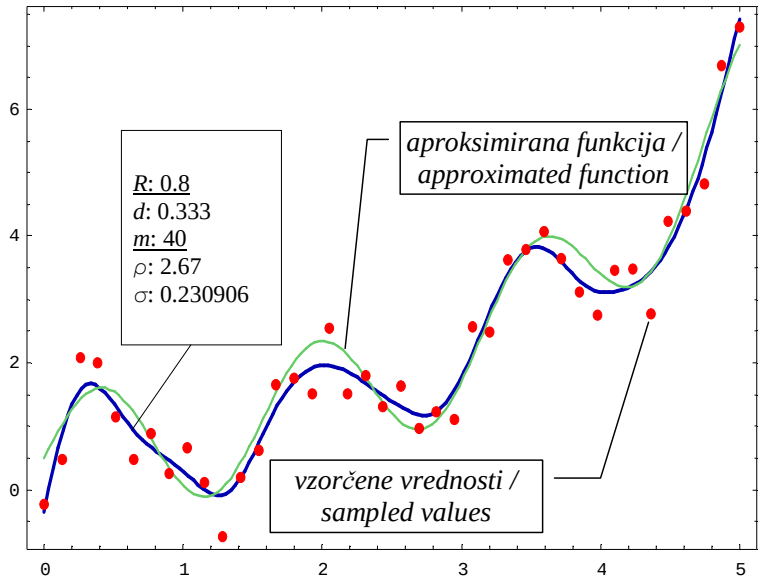
V nadaljevanju obravnavamo aproksimacijo na podlagi podatkov s šumom. Sl. 5 prikazuje aproksimacijo na podlagi 20 vrednosti z ravnjo šuma R=0,4 pri treh različnih vrednostih efektivnega dosega d. Jasno je viden vpliv efektivnega dosega na kakovost aproksimacije. Ko je d prevelik, postanejo izbrane bazne funkcije

nezadostne za aproksimacijo funkcije na intervalih velikostnega reda 2*d* okrog točke izračuna, na katerih vzorčne točke pomembno prispevajo k aproksimaciji (enačbe (14), (15), (16)). Na Sl. 5 se to odrazi v preveč zravnani aproksimcaiji pri največjem *d*=0,4, ki ne zmore slediti nihanju aproksimirane funkcije. Ko je *d* premajhen, aproksimacija teži k interpolaciji vzorčenih vrednosti in zato sledi tudi naključnim fluktuacijam, ki so posledice šuma.

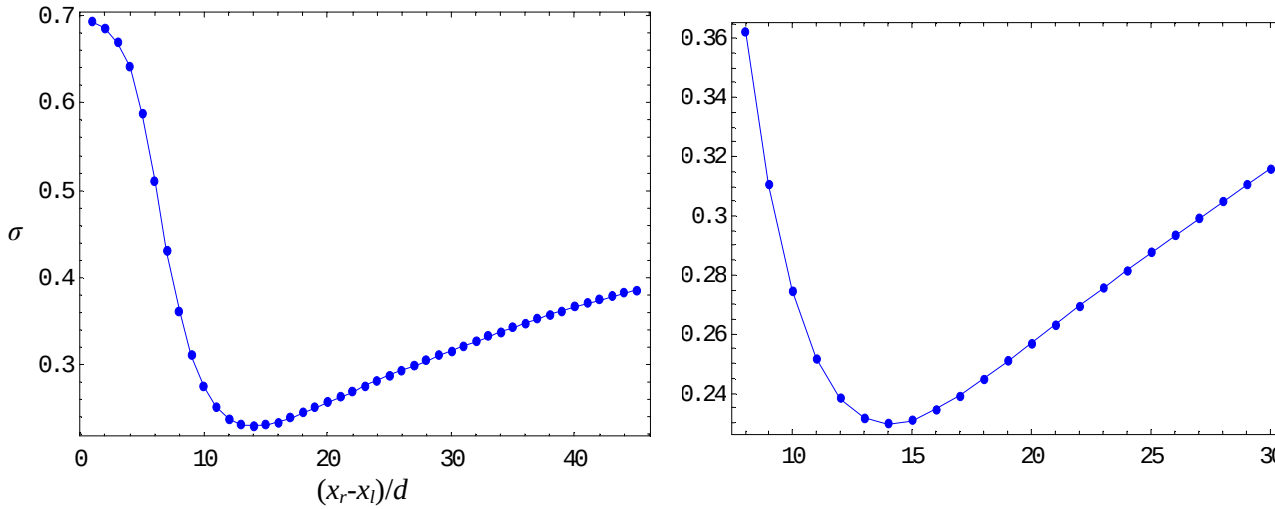


Sl. 5: Učinek efektivnega dosega vpliva vzorčnih točk na aproksimacijo na osnovi 20 vzorčnih točk s šumom.

Videti je, da pri dani funkciji, ki jo aproksimiramo, izbranih vzorčnih točkah in določeni ravni šuma obstaja optimalen efektiven doseg, pri katerem je mera za napako σ minimalna. To je vidno tudi na Sl. 7, kjer je prikazana odvisnost mere za napako aproksimacije σ od količine, ki je obratno sorazmerna z *d*. Graf se nanaša na večjo raven šuma (*R*=0.8) in število vzorčnih točk (*m*=40). Za vsako točko grafa so uporabljene iste izmerjene vrednosti funkcije v vzorčnih točkah. Praviloma bi morali vrednosti na grafu povprečiti po več naključnih izidih meritev. Graf bi v tem primeru vseboval naključne fluktuacije, vendar bi bil trend podoben. Sl. 6 prikazuje vzorčene vrednosti uporabljene pri Sl. 7 in aproksimacijo na podlagi teh podatkov z efektivnim dosegom *d*, ki je blizu optimalnega.



Sl. 6: Aproksimacija na osnovi 40 vzorčnih točk z višjo ravnjo šuma (R=0.8).

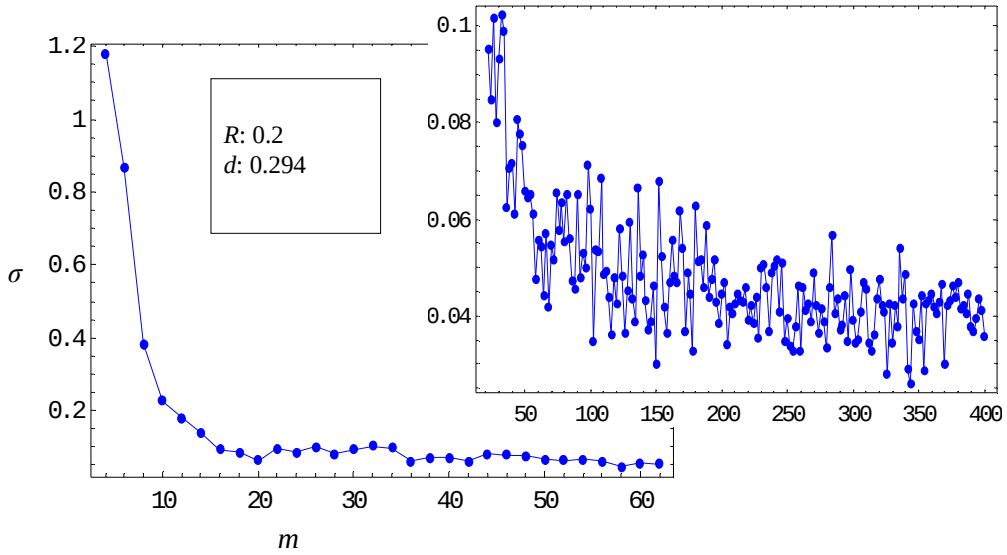


Sl. 7: Odvisnost mere za napako aproksimacije σ od obratnega efektivnega dosega za podatke s Sl. 6 s povečanim detajlom na desni strani slike.

Podobno kot v primeru brez šuma pričakujemo, da lahko kakovost aproksimacije izboljšujemo z večanjem števila vzorčnih točk m. Vendar šum preprečuje neomejeno hitro večanje natančnosti z večanjem m. To se zgodi v območju, kjer amplituda šuma doseže podoben velikostni red kot povprečna aproksimacijska napaka v primeru, ko ni šuma. Od tu naprej lahko aproksimacijsko

napako še vedno manjšamo z večanjem gostote vzorčenja, vendar je to posledica dejstva, da naključne napake zaradi šuma povprečimo po večjem številu vzorčnih točk, ki jih zajamemo znotraj območja vpliva okrog vsake točke, v kateri izračunamo aproksimacijo.

Opisan efekt je jasno viden na Sl. 8 , kjer je narisana napaka aproksimacije pri različnih številih vzorčnih točk. Do približno m=15 napaka hitro pada z rastočim m zaradi zmanjševanja razdalje med točkami. Učinek šuma v tem delu ni izrazit, ker je premajhna količina podatkov poglaviten vir nenatančnosti. Pri večjih m je nadaljnje izboljševanje natančnosti omejeno z naključnimi napakami v podatkih. Aproksimacijske napake se počasi zmanjšujejo s povečevanjem gostote vzorčenja in naključno kolebajo zaradi stohastične narave vzorčenja. Na Sl. 8 vsaka točka na grafu predstavlja en sam izid zajemanja podatkov, ki vsebujejo naključne napake. Trend počasnega izpovprečenja napak pri povečani gostoti vzorčenja je vseeno razviden, ker je prikazano večje število točk.



Sl. 8: Aproksimacijska napaka v odvisnosti od števila vzorčnih točk pri R=0.2 in s konstantnim efektivnim dosegom vpliva d=5/17. Vsaka točka na grafu se nanaša na en sam izid zajemanja vrednosti.

3.1.2.2 Primer: glajenje meritev pri poskusu

Na Sl. 9 je prikazana uporaba aproksimacije po metodi premičnih najmanjših kvadratov pri glajenju signala senzorja za merjenje temperature. Podatki so iz [31] in

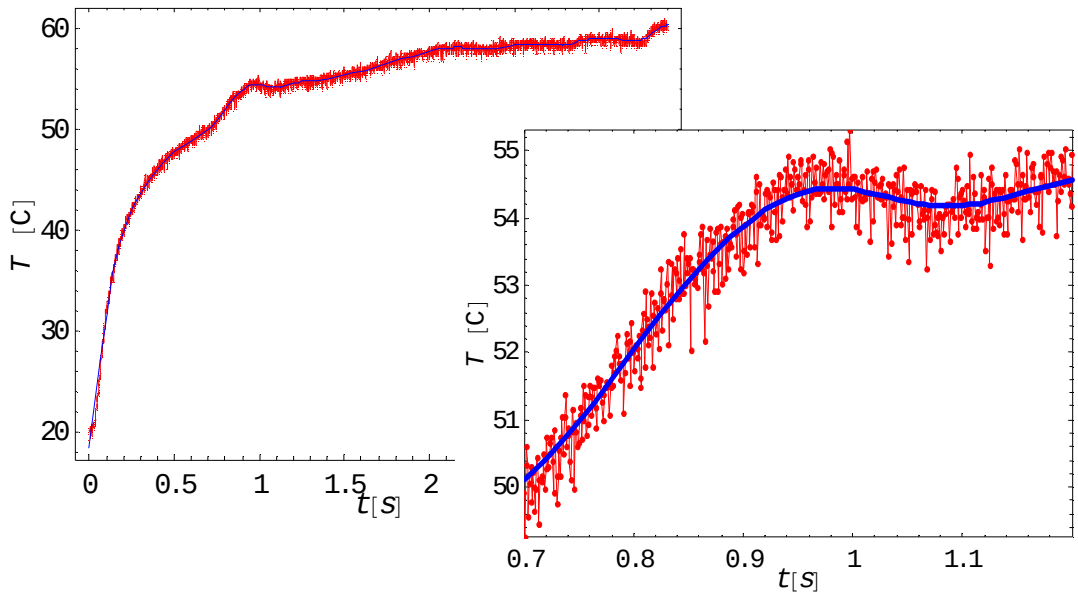
so bili uporabljeni za določitev parametrov prestopa toplote in trenja med kovinami z mazanjem iz meritev pri laboratorijskem testu zoževanja traku.

Test je bil simuliran z metodo končnih elementov, iskani modelski parametri pa so bili določeni z minimizacijo mere za odstopanje meritev od simuliranih podatkov, definirane kot

$$f(\mathbf{p}) = \sum_{i=1}^{n} \int_{t_{\min}}^{t_{\max}} W_i \left(M_i^{(m)}(t) - M_i^{(c)}(\mathbf{p},t) \right)^2 \, dt \quad . \tag{77}$$

V zgornji enačbi i teče po uporabljenih časovno odvisnih merjenih količinah, $M_i^{(m)}(t)$ je ustrezna izmerjena količina (sila na trak ali temperatura v dani točki), $M_i^{(c)}$ ustrezna količina izračunana s simulacijo poskusa pri poskusnih vrednostih iskanih parametrov in W_i uteži izbrane za posamezne vrste meritev.

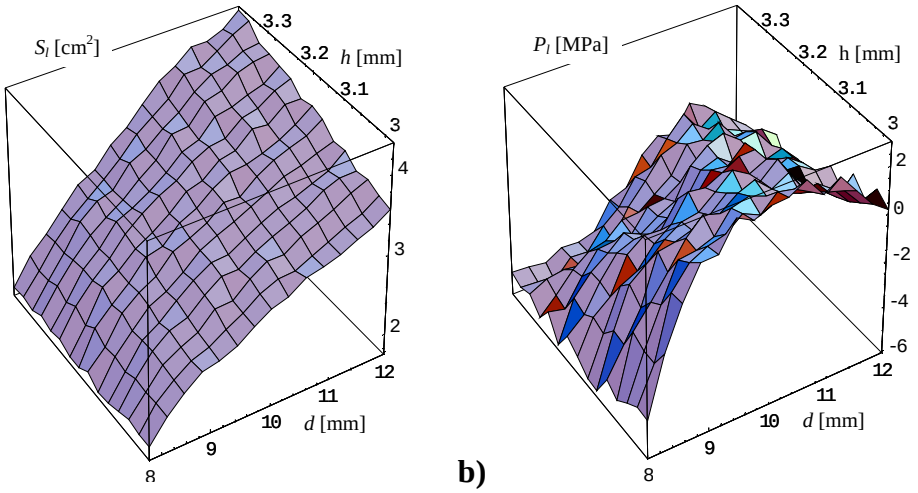
Ker je bila pri numerični simulaciji uporabljena integracija s prilagodljivim korakom in zaradi merskega šuma (Sl. 9) vsebuje funkcija $f(\mathbf{p})$ naključne oscilacije, kar preprečuje učinkovito uporabo numeričnih minimizacijskih metod. Ta problem je bil odpravljen z zamenjavo interpoliranih meritev z gladkimi aproksimacijami po metodi premičnih najmanjših kvadratov (neprekinjena črta na Sl. 9). Pri aproksimaciji je bila uporabljena kvadratična baza (18) z utežnimi funkcijami oblike $w_G(r)$ iz (16). Efektivni doseg vpliva je bil postavljen na $d = 0.1$ s. Na ta način nismo zameglili prehodnih pojavov pri časovnem poteku meritev, hkrati pa je bil vpliv šuma dovolj izravnana (povečan detajl na Sl. 9), da smo dobili gladko odzivno funkcijo f, pri kateri smo lahko učinkovito določili minimum.



Sl. 9: Zglajen signal (neprekinjena črta) temperaturnega senzorja s povečanim detajlom na desni strani.

3.1.2.3 Aproksimacija odzivnih funkcij pri optimizaciji

V naslednjem primeru obravnavamo optimizacijo, kjer imamo opravka z odzivnimi funkcijami s šumom. Podatki (Sl. 10) so iz [32], kjer smo optimirali širino in višino kanala, ki je izdelan z napihovanjem, z namenom zmanjšati tveganje za nastanek razpok, ki nastanejo zaradi lokalizacije deformacije. Naloga je bila postavljena kot maksimizacija površine preseka kanala $S_i(d,h)$ (Sl. 10 a)) pri omejitvi, da je tlak $P_i(d,h)$, pri katerem se začne lokalizacija, pod predpisano mejo. Zaradi tehničnih zahtev sta optimizacijska parametra omejena z $8\,mm \le d \le 12\,mm$ in $3\,mm \le h \le 3.4\,mm$.



a) **b)**

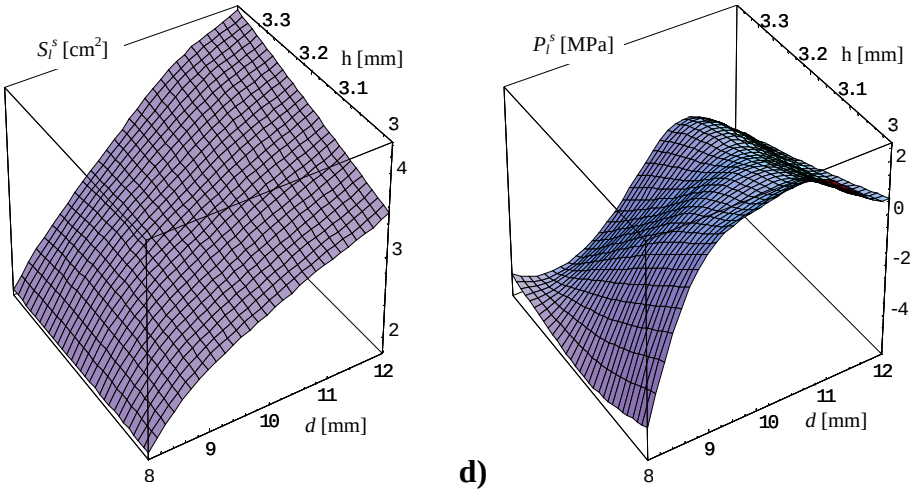
Sl. 10: Namenska (a) in omejitvena funkcija (b) optimizacijskega problema izračunani z metodo končnih elementov na pravilni mreži 20x20 točk.

Šum v odzivnih funkcijah izvira iz numerične simulacije procesa napihovanja, ki je uporabljena za izračun odzivnih funkcij [32]. Uporabiti je bilo potrebno prilagodljivo izboljševanje mreže končnih elementov z visoko gostoto elementov v območju lokalizacije, zaradi česar bi bilo težko zmanjšati raven šuma. Da smo lahko poiskali optimalno rešitev, smo najprej izračunali odzivni funkciji v pravilni mreži točk. Dobljene podatke smo zgladili z aproksimacijo po metodi premičnih najmanjših kvadratov z obliko utežnih funkcij $w_G(r)$, efektivnima

dosegoma $d_1 = 1mm$ in $d_2 = 0.1mm$ v smereh *d* in *h* ter kvadratnimi polinomskimi baznimi funkcijami

$$f_1(\mathbf{x}) = 1;\ \ f_2(\mathbf{x}) = x_1;\ \ f_3(\mathbf{x}) = x_2;\ \ f_4(\mathbf{x}) = x_1^2;\ \ f_5(\mathbf{x}) = x_2^2;\ \ f_6(\mathbf{x}) = x_1 x_2, \quad (78)$$

kjer je x_1=*d* in x_2=*h*. Aproksimiran odziv je prikazan na Sl. 11.



c) **d)**

Sl. 11: Zglajene odzivne funkcije s slike Sl. 10.

V [32] je bil optimizacijski problem rešen z aproksimiranim odzivom z metodo zaporednega kvadratičnega programiranja. Za to je potrebno dovolj gosto vzorčenje odziva po celotnem dovoljenem območju v prostoru optimizacijskih parametrov. Pri večjem številu parametrov to postane zdaleč prezahtevno.

3.1.3 Povzetek uporabe metode premičnih najmanjših kvadratov pri optimizaciji

Metoda premičnih najmanjših kvadratov je vsestranska in prilagodljiva aproksimacijska metoda, ki je zaradi posebnih značilnosti uporabna pri reševanju številnih praktičnih problemov.

Za aproksimacijo ni potrebna kakšna posebna ureditev vzorčnih točk ali particija območja aproksimacije. S primernimi utežnimi funkcijami lahko z uporabo razmeroma majhnega števila baznih funkcij aproksimiramo gladke funkcije na poljubno velikem območju. Velikost sistema enačb za določitev koeficientov aproksimacije se ne poveča, če povečamo gostoto vzorčenja. Po drugi strani je

slabost metode, da moramo sistem enačb za določitev prostorsko odvisnih koeficientov rešiti posebej v vsaki točki, kjer izračunamo aproksimacijo. To v nekaterih primerih predstavlja oviro za uporabnost metode zaradi velike časovne zahtevnosti. Metoda je bolj primerna, kadar aproksimacije ni potrebno izračunati v velikem številu točk.

Pri danem naboru baznih funkcij se z aproksimacijo filtrirajo višjefrekvenčne oscilacije glede na efektivni doseg vpliva vzorčnih točk, kar lahko uporabimo za kompenzacijo vpliva šuma v vzorčenih podatkih. Večji efektivni doseg pomeni boljše glajenje, vendar tudi slabšo zmožnost prilagajanja aproksimirani funkciji. Nasprotno z manjšanjem efektivnega dosega aproksimacija vedno bolj interpolira vzorčene vrednosti. Pri tem pa je potrebna previdnost, saj postane sistem enačb za določitev koeficientov zelo slabo pogojen, ko se efektivni doseg približa redu velikosti razdalje med vzorčnimi točkami ali manj. V praksi moramo doseči primeren kompromis med opisanimi učinki z ustrezno nastavitvijo efektivnega dosega glede na raven šuma, gostoto vzorčenja in lastnosti aproksimirane funkcije. Probleme s slabo pogojenostjo lahko lajšamo z izbiro utežnih funkcij, ki počasneje padajo z razdaljo od vzorčnih točk.

Pri aproksimaciji po metodi premičnih najmanjših kvadratov lahko enostavno širimo območje aproksimacije ali povečamo gostoto podatkov s sprotnim dodajanjem vzorčnih točk, v katerih izračunamo aproksimirano funkcijo. Zaradi tega je metoda primerna za uporabo tudi v optimizacijskih metodah, ki temeljijo na zaporedni aproksimaciji odzivnih funkcij. Obstajajo pa v ta namen tudi enostavnejše metode kot npr. Shepherdova metoda, veliko pa se uporablja tudi aproksimacija z nevronskimi mrežami, ki je zelo prilagodljiva oblika aproksimacije.

Reference:

[1] M. A. Crisfield, *Non-Linear Finite Element Analysis of Solids and Structures*, Vol. 1, John Wiley & Sons, Chichester, 1991.

[2] M. A. Crisfield, *Non-Linear Finite Element Analysis of Solids and Structures*, Vol. 2, John Wiley & Sons, Chichester, 1997.

[3] D. R. J. Owen, E. Hinton, *Finite Elements in Plasticity*, Pineridge Press, Swansea, 1980.

[4] Grešovnik I.: A General Purpose Computational Shell for Solving Inverse and Optimisation Problems - Applications to Metal Forming Processes. *Ph.D. thesis*, University of Wales Swansea, 2000. (www.c3m.si/inverse/doc/phd)

[5] Rodič T. and Gresovnik I.: A computer system for solving inverse and optimization problems. *Eng. Comput.*, vol. 15, no. 7, pp. 893-907, 1998.

[6] R. Fletcher, *Practical Methods of Optimization (second edition)*, John Wiley & Sons, New York, 1996).

[7] J. E. Dennis (Jr.), R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, 1996.

[8] D. P. Bertsekas, *Nonlinear Programming (second edition)*, Athena Scientific, Belmont, 1999.

[9] *Mathematical Optimization*, electronic book at http://csep1.phy.ornl.gov/CSEP/MO/MO.html , Computational Science Education Project, 1996.

[10] A. V. Fiacco, G. P. McCormick, *Nonlinear Programming – Sequential Unconstrained Minimisation Techniques*, Society for Industrial and Applied Mathematics, Philadelphia, 1990.

[11] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Athena Scientific, Belmont, 1996.

[12] S.R. Singiresu, *Engineering Optimization – Theory and Practice (third edition)*, John Wiley & Sons, New York, 1996.

[13] J. L. Nazareth, *The Newton – Cauchy Framework – A Unified Approach to Unconstrained Nonlinear Minimisation*, Springer – Verlag, Berlin, 1994.

[14] W.H. Press, S.S. Teukolsky, V.T. Vetterling, B.P. Flannery, *Numerical Recipies in C – the Art of Scientific Computing*, Cambridge University Press, Cambridge, 1992.

[15] C. T. Lawrence, A. L. Tits, *Nonlinear Equality Constraints in Feasible Sequential Quadratic Programming*, Optimization Methods and Software, Vol. 6, 1996, pp. 265 - 282.

[16] J. L. Zhou, A. L. Tits, *Nonmonotone Line Search for Minimax Problems*, Journal of Optimization Theory and Applications, Vol. 76, No. 3, 1993, pp. 455 - 476.

[17] A. D. Belgundu, T. R. Chandrupatla: *Optimization Concepts and Applications in Engineering*, Prentice Hall, New Jersy, 1999.

[18] S.R. Singiresu, *Engineering Optimization – Theory and Practice (third edition)*, John Wiley & Sons, New York, 1996.

[19] F. van Keulen, V. V. Toropov, *Multipoint Approximations for Structural Optimization Problems with Noisy Response Functions*, electronic document at http://www-tm.wbmt.tudelft.nl/~wbtmavk/issmo/paper/mam_nois2.htm.

[20] J. F. Rodriguez, J. E. Renaud, *Convergence of Trust Region Augmented Lagrangian Methods Using Variable Fidelity Approximation Data*, In: WCSMO-2 : proceedings of the Second World Congress of Structural and Multidisciplinary Optimization, Zakopane, Poland, May 26-30, 1997. Vol. 1, Witold Gutkowski, Zenon Mroz (editors), 1st ed., Lublin, Poland, Wydawnictwo ekoincynieria (WE), 1997, pp. 149-154.

[21] Igor Grešovnik, Linear Approximation with Regularization and Moving Least Squares, electronic book, 2007.

[22] Igor Grešovnik, The Use of Moving Least Squares for a Smooth Approximation of Sampled Data ("Uporaba metode premičnih najmanjših kvadratov za gladko aproksimacijo vzorčenih podatkov."), Journal of Mechanical Engineering 53(2007)9, 582-598, original dosegljiv na anslovu http://www.sv-jme.eu/scripts/download.php?file=/data/upload/2007/9/SV-JME_53%282007%2909_582-598_Gresovnik.pdf

[23] Alfio Quarteroni, Riccardo Sacco, Fausto Saleri, *Numerical Mathematics*. Texts in Applied Mathematics 37, Springer-Verlag, New York, 2000.

[24] I. N. Bronstein, K. A. Semendjajew, G. Musiol and H. Muehlig. Handbook of Mathematics, 5th Edition. Harri Deutsch, 2000.

[25] Z. Bohte, *Numerične metode*, Društvo matematikov, fizikov in astronomov SRS, Ljubljana, 1987.

[26] Zvonimir Bohte, *Numerično reševanje sistemov linearnih enačb*, Društvo fizikov, matematikov in astronomov Slovenije, Ljubljana, 1994.

[27] Jože Petrišič, *Reševanje enačb*, Univerza v Ljubljani – Fakulteta za strojništvo, 2006.

[28] Jože Petrišič, *Interpolacija in osnove računalniške grafike*, Univerza v Ljubljani – Fakulteta za strojništvo, 1999.

[29] I. Emri, R. Cvelbar. Uporaba gladilnih funkcij za glajenje podatkov podanih v diskretni obliki. Strojniški vestnik – Journal of Mechanical Engineering, Vol. 52, p.p. 181-194, 2006.

[30] P. Breitkopf, A. Rassineux, P. Villon, An Introduction to the Moving Least Squares Meshfree Methods, Revue Europenne des lments Finis, Volume 11 - No 7-8, 2002.

[31] I. Grešovnik, S. Stupkiewicz, T. Rodič: Sensitivity analysis and inverse modelling of limits of lubrication. FP6 European project ENLUB (G1RD-CT-2002-00740), Development of New Environmentally Acceptable Lubricants, Tribological Tests and Models for European Sheet Forming Industry. Final technical report, 2006.

[32] I. Grešovnik, S. Hartman, T. Rodič. Reducing localisation induced defects at blow forming in terms of optimal shape design. In E. Onate (ed.), Computational plasticity: fundamentals and applications, Proceedings of the eighth international conference on computational plasticity held in Barcelona, Spain, 5th-7th September, 2005. Part 1, p.p. 388-391.