Univerza
*v Ljubljani*
Fakulteta
*za gradbeništvo*
*in geodezijo*

*Jamova 2*
*1000 Ljubljana, Slovenija*
*telefon (01) 47 68 500*
*faks (01) 42 50 681*
*fgg@fgg.uni-lj.si*

Visokošolski program Gradbeništvo,
Konstrukcijska smer

Kandidat:

**Matej Kocjan**

# Določevanje optimalne zasnove konstrukcijskih elementov

**Diplomska naloga št.: 319**

**Mentor:**
prof. dr. Jože Korelc

Ljubljana, 30. 10. 2008

## IZJAVA O AVTORSTVU

Podpisani **MATEJ KOCJAN** izjavljam, da sem avtor diplomske naloge z naslovom: »**DOLOČEVANJE OPTIMALNE ZASNOVE KONSTRUKCIJSKIH ELEMENTOV**«.

Izjavljam, da se odpovedujem vsem materialnim pravicam za potrebe elektronske separatorke FGG.

Ljubljana, 17.10.2008

## IZJAVE O PREGLEDU NALOGE

Nalogo so si ogledali učitelji konstrukcijske smeri:

## BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

**UDC:** **624.014.2(043.2)**

**Avtor:** **Matej Kocjan**

**Mentor:** **izr. prof. dr. Jože Korelc**

**Naslov:** **Določevanje optimalne zasnove konstrukcijskih elementov**

**Obseg in oprema:** **82 str., 18 pregl., 80 sl.**

**Ključne besede:** **optimizacija, jeklene konstrukcije**

**Izvleček**

Diplomska naloga obravnava določevanje optimalne zasnove konstrukcijskih elementov. Postopek optimalne zasnove, ki je opisan v prvem delu naloge, je sestavljen iz dveh ločenih delov. V začetni fazi optimiziranja izvedemo topološko optimizacijo, s katero se iz grobo opisane konstrukcije približamo optimalni obliki. Pri tem imamo na voljo dve metodi, in sicer optimiziranje na mejno obtežbo ter optimiziranje na maksimalno togost pri danem volumnu. V drugi fazi obliko, dobljeno s topološko optimizacijo, približno opišemo z novimi parametri na način, ki upošteva tudi tehnologijo izdelave končne konstrukcije in račun poženemo ponovno. V tej fazi uporabimo optimizacijo na mejno obtežbo. V drugem delu naloge so opisani postopki, uporabljeni na dejanskih primerih. Vsi matematični postopki se izvajajo v programskem paketu `Mathematica` s pomočjo dodatkov `AceGen` in `AceFEM`, ki sta okolji za analizo po metodi končnih elementov. Topološko optimizacijo smo izvedli za primer konzole, preklade in stebra, med tem ko smo optimizacijo parametrov izvedli samo na primeru konzolo. Na koncu smo še preverili ali dobljeni rezultat zadosti zahtevam standarda EN 1993-1-1.

## BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION

**UDK:** **624.014.2(043.2)**

**Author:** **Matej Kocjan**

**Supervisor:** **assoc. prof. dr. Jože Korelc**

**Title:** **Determination of optimal topology of structural components**

**Notes:** **82 p., 18 tab., 81 fig.**

**Key words:** **optimization, steel construction**

**Abstract**

The thesis discusses determination of structural elements' optimal design. The first part of the thesis describes the optimum design which consists of two separate parts. The topology optimization, which puts roughly outlined construction to an optimal form, is made in the initial phase of optimization. Hereby we have two methods at our disposal; the first being, to optimize on a limit load, and the second, to optimize on maximum rigidity for a given volume. In the second stage, the form obtained by topology optimization is described with new parameters in a way that also takes technology of producing the final construction into account. In this stage we use the optimization on limited load. The second part of the thesis describes procedures that are used in actual examples. All mathematical procedures are implemented in the software `Mathematica` with the help of `AceGen` and `AceFEM` appenix, which are analysis programs using the method of finite elements. We carried out the topology optimization for the cantilever, beam and column examples whereas the optimization of parameters was made only for a cantilever example. Results obtained in the research were checked for the EN 1993-1-1 requirements.

## ZAHVALA

Za pomoč in nasvete pri pisanju diplomske naloge se zahvaljujem mentorju izr. prof. dr. Jožetu Korelcu. Hvala tudi vsem na Katedri za metalne konstrukcije, ki so mi omogočili prijetno delovno vzdušje

Zahvalil bi se tudi svoji družini, ki mi je skozi vsa leta študija stala ob strani in mi pomagala.

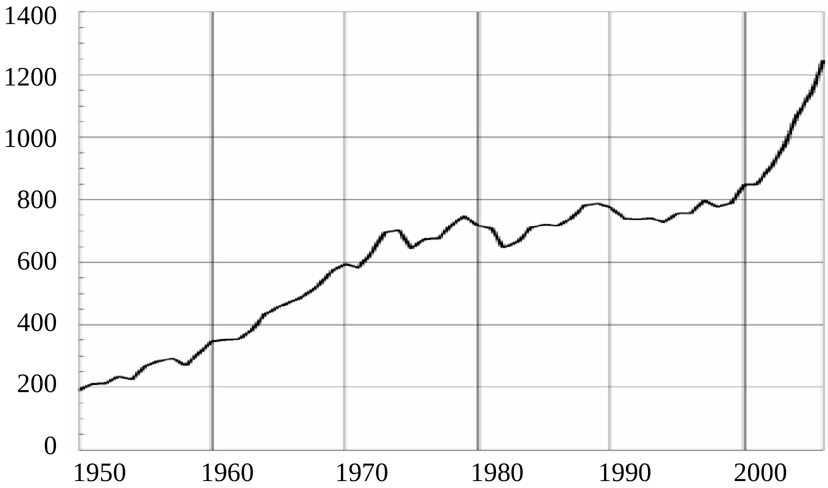
## KAZALO VSEBINE

## KAZALO PREGLEDNIC

# KAZALO SLIK

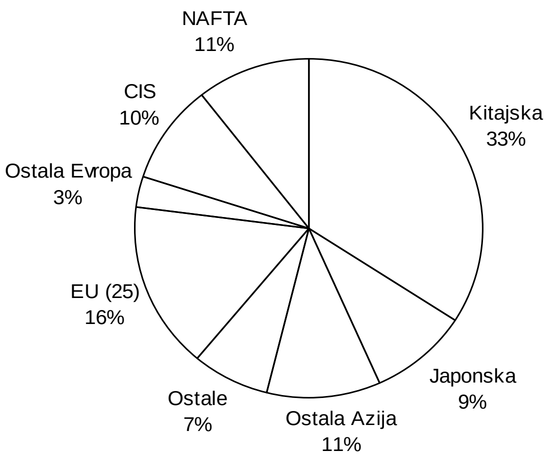
# 1 UVOD

V gradbeništvu se jeklo uporablja predvsem za izdelavo nosilnih konstrukcijskih elementov. Zadnja leta so se največjim porabnikom jekla (le-te so razvite države, predvsem ZDA) pri veliki porabi jekla pridružile tudi države v razvoju – Kitajska, Indija, Rusija in Brazilija. Večja poraba (Slika 1) in povpraševanje po celem svetu posledično pomenita rast cen jekla. Slika 3 nazorno prikazuje izrazito rast cene konstrukcijskega jekla zadnjih nekaj let.



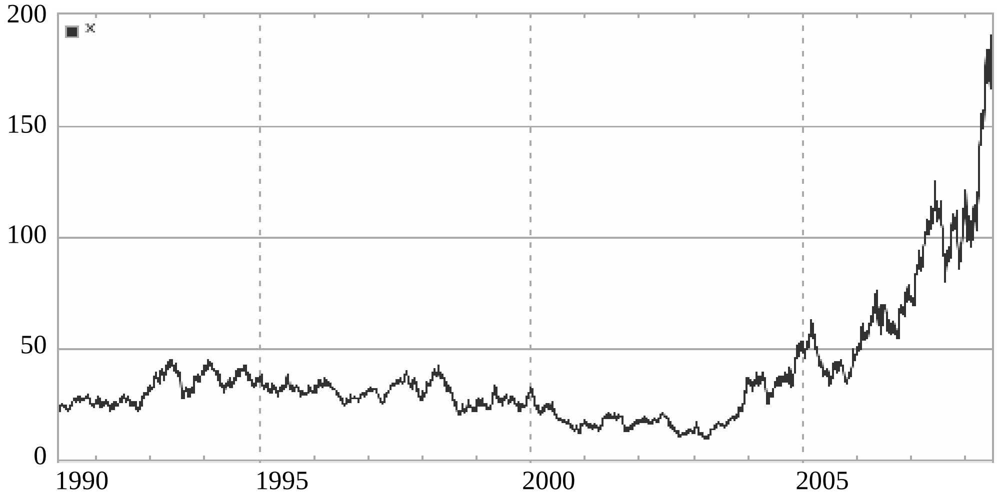

Slika 1: Poraba jekla v letih 1950 – 2006 v milijon tonah
Vir: http://www.worldsteel.org

Analitiki so mnenja, da se bo strma rast cene konstrukcijskega jekla nadaljevala, saj so največje porabnice države, ki so šele na začetku svojega razvoja (Slika 2). Predvsem lahko pričakujemo povečano povpraševanje po jeklu iz strani Indije, Rusije ter Brazilije. Višanje cene jekla (Slika 3) pa v končni fazi pomeni dražjo gradnjo.




Slika 2: Delež proizvodnje jekla za posamezne države
Vir: http://www.worldsteel.org

Višanje cene jekla se lahko začne odražati v upadanju deleža jeklenih objektov. Možna posledica silovite rasti cene je, da betonski objekti postanejo bolj racionalni. Tega si ne želi nihče, ki je kakor koli povezan z jeklarsko industrijo. Na tej stopnji lahko inženirji s svojim bogatim znanjem in eksperimentalnim delom močno vplivajo na nadaljnji razvoj jeklenih konstrukcij. Z optimiziranjem oblik konstrukcijskih delov objektov se lahko privarčuje pri porabi materiala. Na tem mestu je potrebno poudariti, da je tudi sam postopek optimizacije z ekonomskega vidika drag postopek. Optimizacijski algoritmi postajajo vse zahtevnejši ter potrebujejo vse več časa.




Slika 3: Cena konstrukcijskega jekla od leta 1991 v USD/tono
Vir: http://finance.yahoo.com/

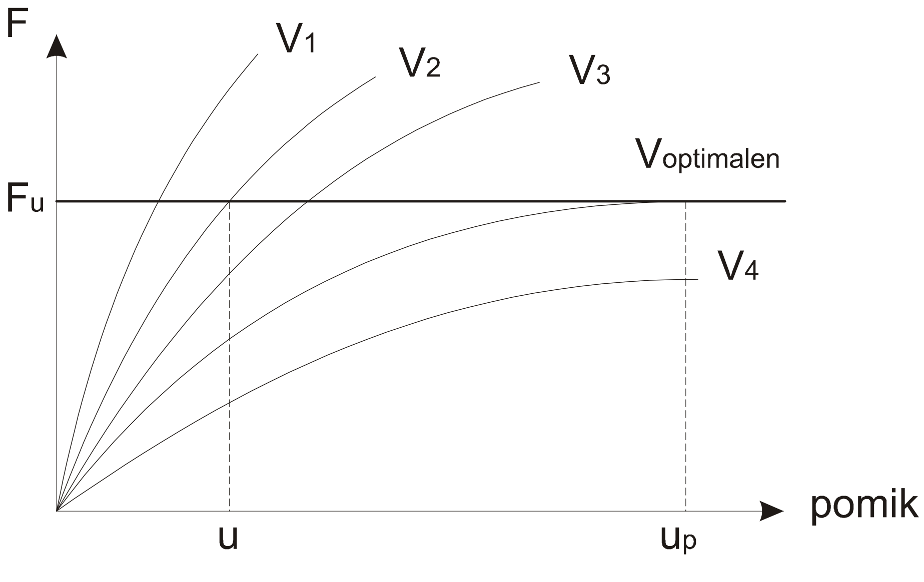
Namen naloge je določitev optimalne zasnove konstrukcije s pomočjo novih matematičnih postopkov in s tem nadgraditi trenutne metode optimizacije, vgrajene v komercialnih programih, za analizo in dimenzioniranje jeklenih konstrukcij.

# 2 OPTIMIZACIJA

## 2.1 Splošno

Zaradi vse večjega povpraševanja in posledično vse večje rasti cen surovin, se pojavi želja po manjši porabi materiala, torej optimizaciji. Optimizacija je prisotna že od nekdaj. Včasih je bila optimizacija pogojena z izkušnjami konstruktorja, danes pa s pridom izkoriščamo vse prednosti računalniške opreme. S hitrim razvojem računalniške opreme postajajo postopki optimiziranja vedno bolj zahtevni in tudi uspešnejši.

Enostavni iteracijski postopki optimiziranja so vgrajeni v večino komercialnih programov za analizo in dimenzioniranje jeklenih konstrukcij. Ti postopki optimizirajo konstrukcijo z izbiro vnaprej znanih parametrov (pogosto se spreminja le en parameter ali pa celoten, vnaprej definiran prerez – IPE, HEA, HEB, …) oblike konstrukcijskega elementa. Program glede na podano obtežbo konstrukcije izbere optimalen prečni prerez konstrukcije, ki še zadosti zahtevam standarda. Tukaj se pojavi vprašanje ali je izbran prerez resnično optimalen in ali za izdelavo takšnega prereza resnično porabimo minimalno količino materiala.




Slika 4: Iskanje optimalnega volumna v odvisnosti od mejne obtežbe

Osnovna ideja diplomske naloge je optimiziranje oblike konstrukcijskega elementa, pri čemer bomo med seboj primerjali dve metodi optimiziranja. V nalogi bosta tako predstavljeni gradientna metoda optimizacije na mejno obtežbo ter gradientna metoda optimizacije na maksimalno togost pri danem volumnu. Vsi matematični postopki se izvajajo v programskem paketu Mathematica[1] s pomočjo dodatkov AceGen[2] in AceFEM[3]. Mathematica je zelo močno, svetovno znano matematično orodje, ki se uporablja za simbolno in numerično računanje. Dodatka AceGen in AceFEM pa sta okolja za analizo po metodi končnih elementov. Bistvena prednost izbrane programske opreme je v možnosti kombinacije simbolnih in numeričnih izračunov, kateri omogočajo rešitev do sedaj težko rešljivih problemov.

Naloga gradientne metode optimizacije je iskanje minimuma namenske funkcije $\Phi$, ki jo sestavlja vsota osnovne funkcije in različne kazenske funkcije. Postopek iskanja minimuma funkcije poteka iterativno; to pomeni, da je v vsakem koraku konstrukcija bližje končni obliki. Pri računanju minimuma namenske funkcije $\Phi$ pa si pomagamo s tako imenovanim gradientom.

Da se oblika konstrukcije lahko spreminja, je potrebno celoten konstrukcijski element opisati s parametri oblike $\phi_i$. Ker je namen naloge optimizirati splošno geometrijo (topologijo) konstrukcijskega elementa, je potrebno vse izvajati s 3D modeli. V nalogi smo tako geometrijo konstrukcijskih elementov opisali s parametri, vse materialne lastnosti pa so ostale konstantne tekom računa.

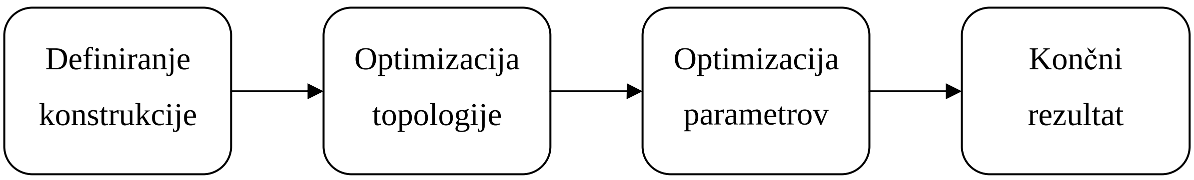
Gradient, ki ga dobimo kot odvod namenske funkcije $\Phi$ po izbranih parametrih $\phi_i$, nam v postopku optimiziranja določa smer spreminjanja velikosti parametrov. Tako smo v vsakem iterativnem koraku bližje iskanemu minimumu. Na tem mestu se je potrebno zavedati, da dobljeni minimum običajno ni globalni minimum funkcije $\Phi$. Zaradi kompleksnosti namenskih funkcij lahko optimizacijski algoritem konvergira k lokalnem minimumu, ki ni nujno tudi globalni minimum.

[1] http://www.wolfram.com/
[2] http://www.fgg.uni-lj.si/Symech/
[3] http://www.fgg.uni-lj.si/Symech/

## 2.2 Potek optimizacije

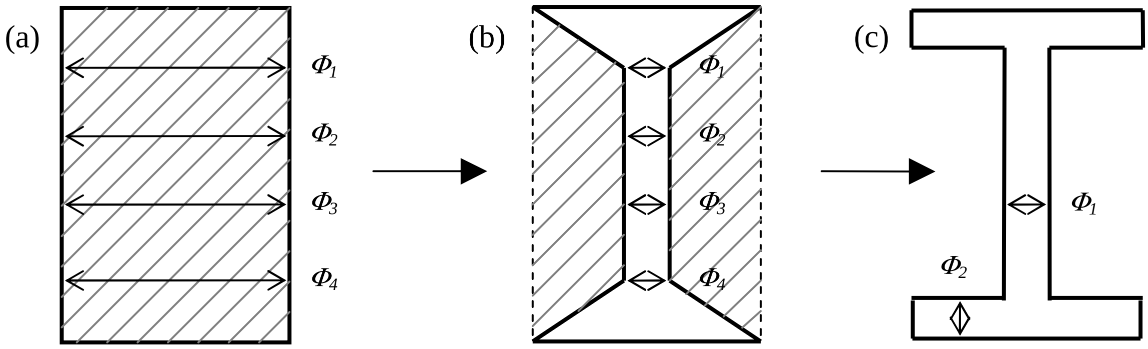
Že na samem začetku se je potrebno vprašati, do katere stopnje bi radi optimizirali naše konstrukcije. V primerjavi z ostalimi panogami, kjer se izdelki izdelujejo množično in v velikih količinah, so končni izdelki v gradbeništvu edinstveni. Vsak objekt je zgodba zase in tako se tudi konstrukcijski elementi med seboj razlikujejo, zato je to smiselno upoštevati pri samem postopku. Matematični postopki so s pomočjo najnovejših računalniških programov sposobni izvesti zelo komplekse primere, vendar se je potrebno vprašati o uporabnosti dobljenih rezultatov.

Optimizacija oblike konstrukcije je možna tako, da se ob tem porabi resnično minimalni volumen materiala, vendar se pri tem pojavljajo poljubne oblike prerezov. Poleg tega, da se v gradbeništvu izdela le nekaj enakih elementov, smo omejeni tudi pri izbiri oblike. Slednje je še posebej prisotno pri jeklenih konstrukcijah, katere so večinoma sestavljene iz ravnih pločevin. Tako lahko že takoj na začetku omejimo končni nabor rezultatov in s tem skrajšamo postopke. Pri tem se, na račun izvedljivosti, nekoliko oddaljimo od povsem optimalne oblike, saj bi bila izdelava popolnoma optimalne oblike predraga in nesmiselna.




Slika 5: Shematski prikaz celotnega poteka optimizacije

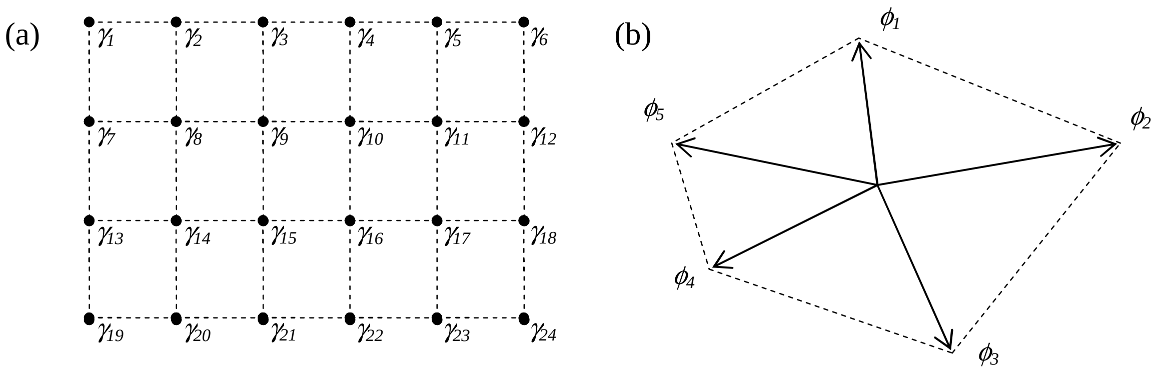
Pri upoštevanju opisane poenostavitve lahko celoten postopek optimiziranja izvedemo z dvema ločenima postopkoma optimiziranja. V začetni fazi optimiziranja izvedemo topološko optimizacijo, s katero se iz grobo opisane konstrukcije približamo optimalni obliki. V drugi fazi pa obliko, dobljeno s topološko optimizacijo, približno opišemo z novimi parametri na način, ki upošteva tudi tehnologijo izdelave končnega elementa in račun poženemo ponovno.




Slika 6: Prikaz postopka optimizacije: začetna geometrija (a), rezultat topološke optimizacije (b) in rezultat parametrične optimizacije (c)

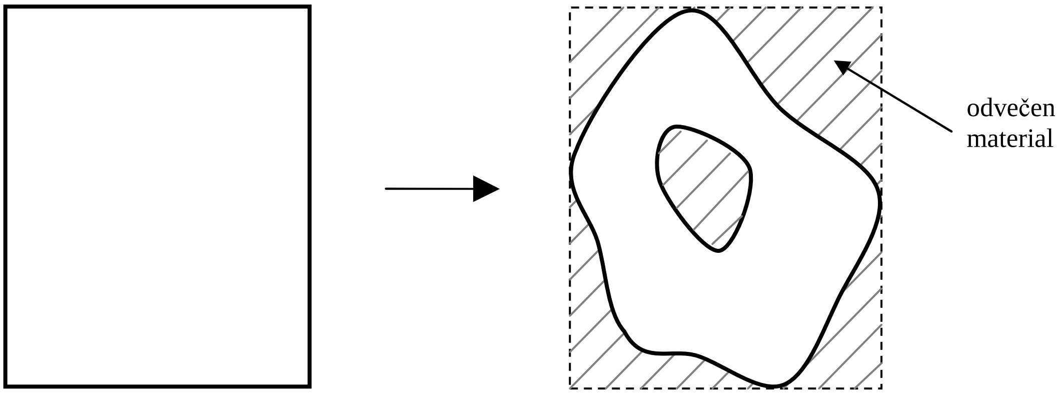
### 2.2.1 Optimizacija topologije

Postopek topološke optimizacije je računsko izredno zahteven. Konstrukcijo lahko opišemo na dva načina, in sicer z mrežo točk, katerim spreminjamo gostoto materiala ali s parametri, s katerimi spreminjamo obliko geometrije (Slika 7). Pri spreminjanju gostote dobimo optimalno obliko konstrukcije tako, da točke, v katerih gostota materiala pade pod minimalno vrednost, obravnavamo kot nepotrebne in jih izločimo iz nadaljnjega računa. Po drugi strani pa se pri spreminjanju oblike geometrije s pomočjo parametrov geometrija spreminja sproti.




Slika 7: Topološka optimizacija: mreža točk z gostoto materiala $\gamma_i$ (a), opis geometrije s parametri $\phi_i$ (b)

V nalogi smo se osredotočili le na metodo, ki za optimiziranje uporablja spreminjanje parametrov. Z večanjem števila parametrov oblike dobimo boljše rezultate (bolj zvezne oblike konstrukcije), vendar porabimo več časa za sam izračun. Zaradi zahtevnosti postopkov in že omenjenih problemov pri dejanski izdelavi elementov, v začetni fazi konstrukcijo grobo opišemo. Za opis konstrukcije uporabimo grobo mrežo parametrov. S tem zmanjšamo natančnost samega izračuna vendar veliko pridobimo na sami dolžini postopka.

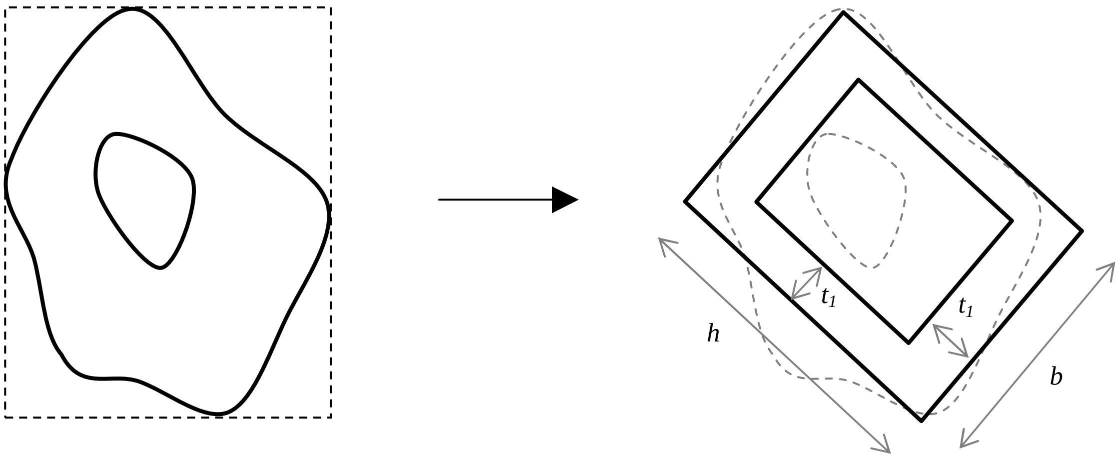



Slika 8: Shematski prikaz topološke optimizacije

Rezultat topološke optimizacije je približek optimalne konstrukcije, vendar natančnost dobljenih rezultatov v tej fazi ni odločilna (Slika 8). Naloga topološke optimizacije je predvsem oblikovno približevanje optimalni obliki, pri tem pa nas dejanske velikosti parametrov ne zanimajo. Ravno tako kot končne dimenzije tudi oblika ni nujno optimalna, saj to ni končni rezultat.

### 2.2.2 Optimizacija parametrov

Obliko, dobljeno s topološko optimizacijo, nato poenostavimo in opišemo le z nekaj parametri. V tem koraku začnemo upoštevati možnost izdelave končne oblike konstrukcije. Tako recimo konstrukcijo opišemo samo z ravnimi pločevinami in spreminjamo dimenzijo le-teh, ne pa tudi obliko.

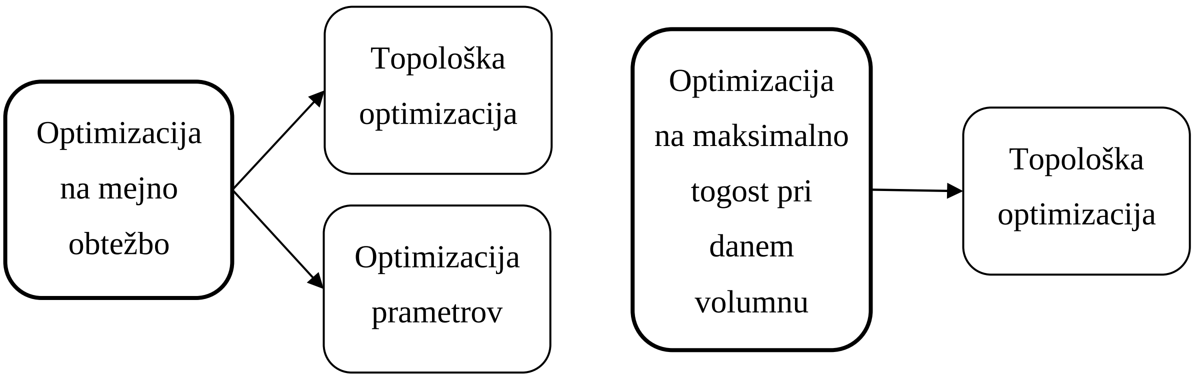



Slika 9: Shematski prikaz parametrične optimizacije

Pri optimizaciji parametrov se sicer s spreminjanjem prej določene oblike nekoliko oddaljimo od minimalne porabe materiala, se pa zato približamo stanju, katerega lahko brez dodatnih tehnologij tudi izvedemo, kar je najbolj pomembno. V tej fazi intuitivno, glede na topološko optimizacijo, določimo obliko prečnega prereza in ga zapišemo z minimalno potrebnimi parametri (Slika 9).

# 3 POSTOPEK OPTIMIZACIJE

Kot je bilo omenjeno, bo v nalogi opisana gradientna metoda optimizacije oblike konstrukcijskega elementa na mejno obtežbo in na maksimalno togost pri danem volumnu. Optimizacijo na mejno obtežbo ločimo na topološko optimizacijo in optimizacijo parametrov, med tem ko optimizacijo na maksimalno togost pri danem volumnu ločimo samo na topološko optimizacijo




Slika 10: Shematski prikaz ločevanja optimizacijskih postopkov

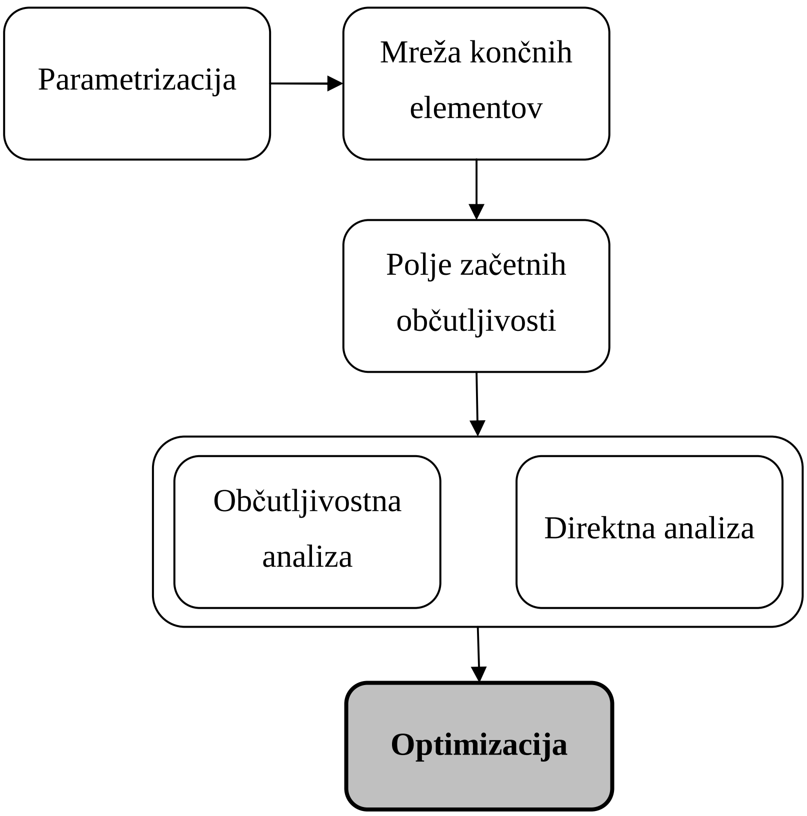
Metoda optimizacije se izvaja iterativno v matematičnem okolju `Mathematica`. Glavni problem gradientne metode optimizacije oblike je natančen izračun začetnih občutljivosti, za kar je potrebno opisati mrežo končnih elementov v odvisnosti od projektnih parametrov $\phi_i$. Namen začetnih občutljivosti je natančen opis sprememb koordinat vozlišč točk končnih elementov glede na poljubno izbran projektni parameter. Polje začetnih občutljivosti nam daje gradient, ki ga v nadaljevanju potrebujemo pri občutljivostni analizi.

Pri optimizaciji oblike je potrebno skonstruirati ustrezno 3D mrežo končnih elementov za natančen izračun direktne analize in izračun občutljivostne analize. Pri računu občutljivostne analize je potrebno upoštevati pravilen pristop za izračun polja začetnih občutljivosti. Neprimerna izbira metode izračuna polja začetnih občutljivosti ima kot posledico neoptimalne rezultate, lahko pa se celo zgodi, da je rezultat napačen. Tako direktna kot občutljivostna analiza se izvajata v vsakem iteracijskem koraku ločeno.

Ko poznamo polje začetnih občutljivosti in imamo izračunan njegov gradient, le-tega uporabimo v občutljivostni analizi. Rezultat občutljivostne analize nam daje nov gradient, katerega potrebujemo v optimizacijskem algoritmu. Gradient občutljivostne analize nam da osnovne podatke o tem, kako naj spreminjamo velikost izbranih parametrov $\phi_i$. Občutljivostna analiza se, kot je bilo že omenjeno, izvaja v vsakem iteracijskem koraku posebej in ločeno od direktne analize.

Obliko konstrukcijskega elementa smo optimizirati na mejno obtežbo in na minimalen pomik. Optimizacija na mejno obtežbo pomeni, da moramo konstrukcijo voditi do same porušitve, kar pa je izredno zahtevno. Da bi bil optimizacijski algoritem sposoben doseči mejno obtežbo (Slika 4), je potrebno dodati dva v naprej pripravljena modela končnih elementov, in sicer model, ki konstrukcijo obremenjuje z linijsko obtežbo ter model, ki v predpisani točki meri pomik. S pomočjo teh dveh modelov končnih elementov smo nato sposobni konstrukcijo obremeniti do mejne obtežbe.

Pri optimizaciji na maksimalno togost pri danem volumnu obliko konstrukcije optimiziramo, da dosežemo minimalen pomik v izbrani točki pri vnaprej izbranem konstantnem volumnu konstrukcije. Iščemo konstrukcijo, ki bo imela pri danem volumnu maksimalno togost.




Slika 11: Shematski prikaz celotnega postopka optimizacije

Med samim optimizacijskim algoritmom izvajamo v vsakem koraku tudi direktno analizo. Direktna analiza nam daje pomike vozlišč mreže končnih elementov, katere lahko kasneje uporabimo za izračun drugih notranjih količin konstrukcije (deformacij, napetosti, …).

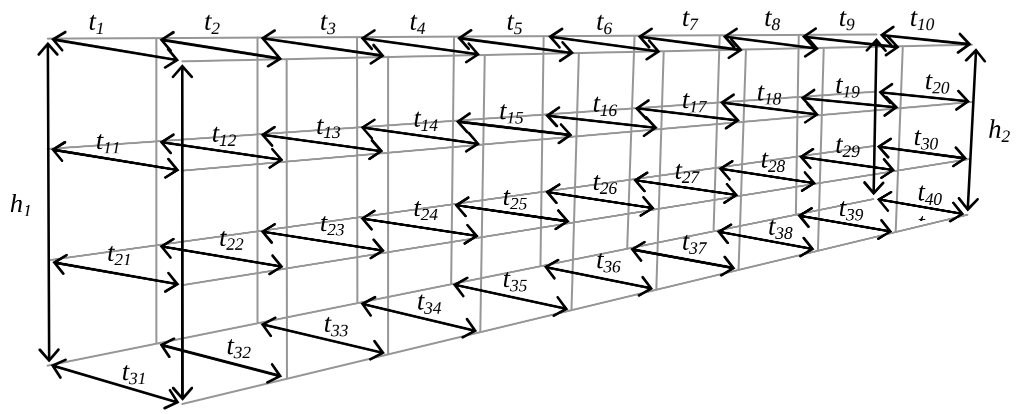
Postopek optimizacije se konča, ko dosežemo predpisano natančnost. Podroben potek posameznih korakov optimizacijskega algoritma (Slika 11), je opisan v nadaljevanju.

## 3.1 Parametrizacija

Bistvo same optimizacije je optimalna izbira vnaprej izbranih parametrov oblike, to pa dosežemo s postopnim spreminjam le-teh. Da pa bi lahko optimizacijski algoritem poganjali, moramo celotno obliko konstrukcije opisati s parametri. Gostota mreže parametrov je odvisna od pričakovanega spreminjanja oblike; več kot imamo parametrov, več možnosti spreminjanja oblike imamo (bolj gladke so končne oblike).

Sam postopek optimizacije omogoča izbiro poljubno mnogih in različnih tipov parametrov. Število parametrov je omejeno z računskim časom postopka. Kot parameter si lahko izberemo parametre oblike (višino, debelino, dolžino), gostoto, volumen, napetost in še kaj. Za sam postopek optimizacije je potrebno pametno razmisliti, katere parametre bi radi optimizirali. V našem primeru smo se odločili za optimizacijo oblikovnih parametrov, in sicer višine in debeline. Izbira teh parametrov nam omogoča, da lahko dobljene postopke uporabimo tudi v praksi pri izdelavi konstrukcij. Obliko elementov konstrukcijskega sklopa lažje in ceneje spreminjamo kakor mehanske lastnosti delov konstrukcij.

Parametrizacijo vpeljemo v sam postopek s konstruiranjem ustrezne mreže končnih elementov. Zaradi uporabe programske opreme, ki omogoča simbolno generacijo končnih elementov, lahko mrežo končnih elementov opišemo v simbolni obliki. Mrežo končnih elementov zapišemo v odvisnosti od parametrov $\phi_i$. Tako so koordinate vozlišč končnih elementov zapisane v obliki izrazov in so eksplicitno izražene s projektnimi parametri $\phi_i$.



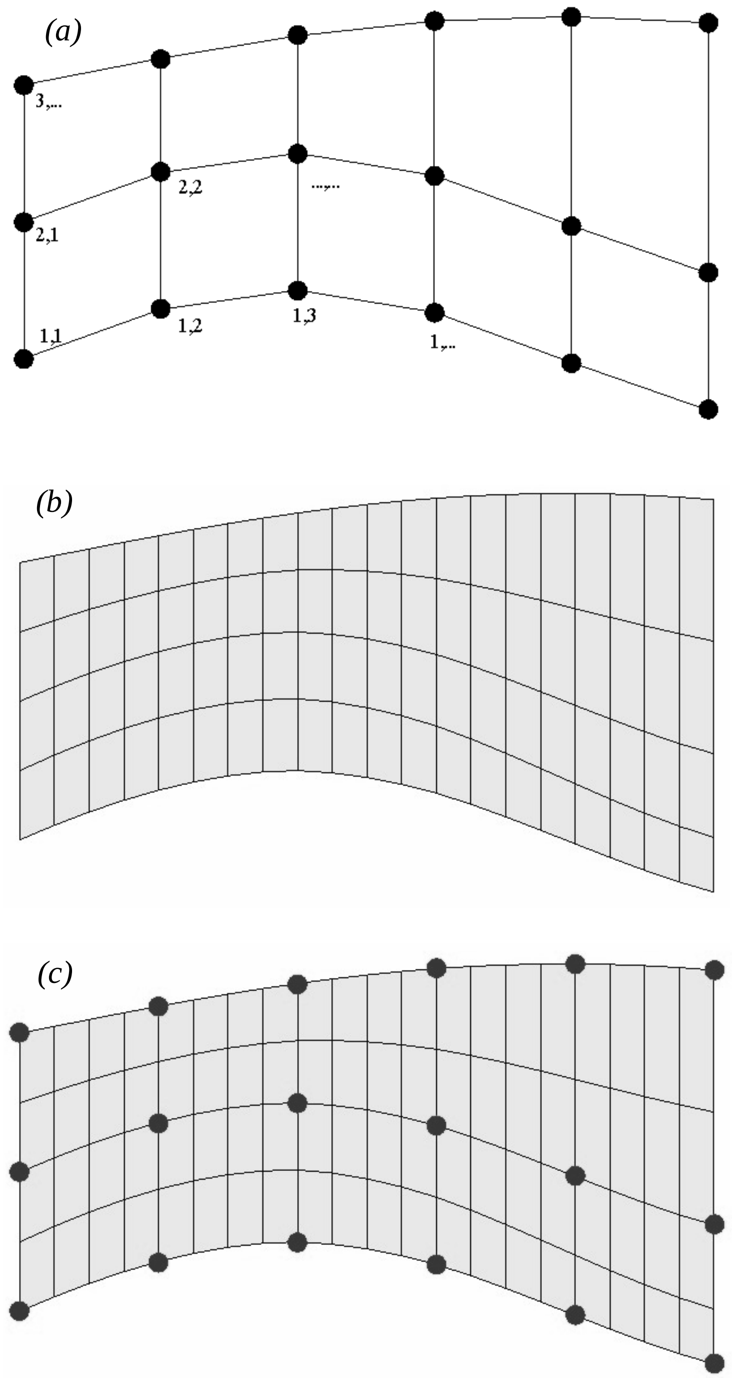

Slika 12: Primer parametrizacije 3D mreže končnih elementov

Ker želimo spreminjati obliko konstrukcije po višini in debelini, se moramo problema lotiti s 3D modeli, kar naredi postopek bistveno zahtevnejši. 3D mreže končnih elementov podaljšajo čas izračuna, saj mora program izvesti veliko več izračunov, kot če bi obravnaval enostavno 2D ali lupinasto konstrukcijo. Slika 12 prikazuje primer parametrizacije 3D mreže končnih elementov. V tem primeru imamo definirane parametre višine na začetku konstrukcije $h_1$ in na koncu konstrukcije $h_2$ ter mrežo 40-ih parametrov debeline $t_i$ (10 parametrov $t_i$ po dolžini konstrukcije in 4 parametri $t_i$ po višini konstrukcije).

## 3.2 Definiranje mreže končnih elementov

Potrebno je poudariti, da je interpolacijska mreža, katero opišemo s parametri $\phi_i$, namenjena določitvi polja začetnih občutljivosti (poglavje 3.3) ter da njena gostota ne vpliva na samo točnost rezultatov direktne analize. Njena gostota je odvisna od gostote parametrov $\phi_i$ in se navezuje na to, kako gosto želimo spreminjati parametre. Definiranje mreže končnih elementov s programom `AceFEM` je opisano kasneje (poglavje 3.2).

Interpolacijska mreža za opis parametrov $\phi_i$, katero uporabimo za določitev polja začetnih občutljivosti, ni nujno enaka mreži končnih elementov za opis oblike konstrukcije, katero uporabimo v direktni analizi. Mreži sta lahko različni, ker se analizi izvajata ločeno. Problem nastane, ko želimo rezultate prve analize upoštevati v drugi analizi. Takrat je potrebno rezultate interpolirati in tako zagotoviti enotno mrežo. Problemu se izognemo z izbiro enake interpolacijske mreže za določitev polja začetnih občutljivosti in analize konstrukcije.




Slika 13: Interpolacijska mreža *(a)*, mreža končnih elementov *(b)* in skupna mreža *(c)*

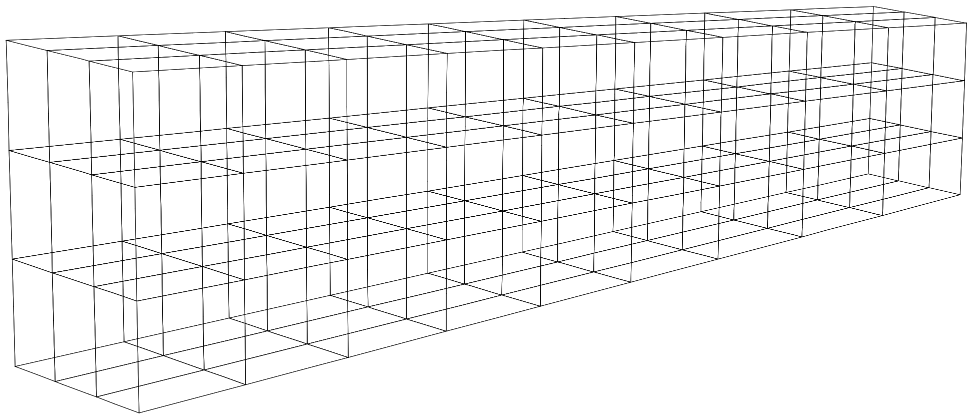
Kot je bilo že omenjeno v enem izmed prejšnjih poglavij, interpolacijsko mrežo zapišemo v odvisnosti od izbranih parametrov oblike $\phi_i$. Ker obravnavamo 3D model, seveda definiramo 3D mrežo. V programu `AceFEM` je za konstruiranje mreže končnih elementov namenjen ukaz

SMTMesh[4]. Za potrebe polja začetnih občutljivosti definiramo interpolacijsko mrežo, vendar se sama tipologija ne spremeni veliko.

```
SMTMesh["konzola","H1",{nx,ny,nz},
Table[{
Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/(nparamX-
1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/(nparamX-
1))/L0+hi[[2]]) (j-1)/(nparamY-1),+ti[[j,i]]/2}, {i,nparamX}],
    Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1)
L0/(nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/(nparamX-
1))/L0+hi[[2]]) (j-1)/(nparamY-1),-ti[[j,i]]/2},
{i,nparamX}]},{j,nparamY}]]
```

Slika 14: Primer zapisa 3D mreže končnih elementov z oblikovnimi parametri $t_i$ in višinama $h_1$ in $h_2$

Algoritem je zapisan popolnoma posplošeno in nam omogoča izbiro poljubno velikega števila parametrov $t_i$. Število parametrov $t_i$ je definirano s spremenljivkama nparamX, ki predstavlja število parametrov v smeri dolžine $L_0$ in nparamY, ki predstavlja število parametrov v smeri višine konstrukcije. Skupno število parametrov $t_i$ je tako nparamX*nparamY. Parametri $t_i$ v tem primeru opisujejo debelino elementa po celotni površini, parametra $h_1$ (hi[[1]]) in $h_2$ (hi[[2]]) pa začetno višino elementa oz. končno višino parametra.



Slika 15: Primer 3D modela mreže končnih elementov

Za dovolj natančen račun direktne analize, ki jo izvaja programa AceFEM, je potrebno izbrati dovolj gosto mrežo končnih elementov. To v algoritmu za opis mreže SMTMesh zapišemo s parametri {nx,ny,nz}, kjer posamezen parameter pomeni število končnih elementov na celotni dolžini (oz. višini ali debelini) v smeri koordinatnih smeri.

[4] Ukaz v programu AceFEM, ki skonstruira mrežo končnih elementov

## 3.3 Polje začetnih občutljivosti

Pri gradientni metodi optimizacije oblike je ključnega pomena natančen izračun polja začetnih občutljivosti. Namen polja začetnih občutljivosti je opisati spremembo koordinat vozliščnih točk končnih elementov, glede na poljubne parametre $\phi_i$. Za izračun polja začetnih občutljivosti je bil uporabljen simbolično-numerični pristop. Prednost te metode je v tem, da lahko upravlja s poljubnimi izrazi. To nam daje možnost, da projektni parametri oblike $\phi_i$ v fazi opisa modela in generacije mreže ostanejo v simbolični obliki. Tako so koordinate vozlišč končnih elementov zapisane v obliki izrazov, ki so eksplicitno izraženi s projektnimi parametri $\phi_i$. Izračun polja začetnih občutljivosti lahko nato enostavno izvedemo z odvajanjem koordinat točk mreže končnih elementov po projektnih parametrih oblike $\phi_i$.

Z analitičnim odvajanjem koordinat po parametrih oblike $\phi_i$ dobimo začetne občutljivosti, katere potrebujemo v analizi občutljivosti. Odvajanje smo v programu `Mathematica` oziroma `AceFEM` izvedli z ukazom `Map[D[XYZ,#]&,ϕ]]`, kjer `XYZ` vsebujejo koordinate vozlišč v simbolni obliki, izražene s parametri $\phi_i$.

## 3.4 Občutljivostna analiza

Občutljivostno analizo konstrukcije lahko definiramo kot metodo, s katero opišemo velikost spremembe odziva konstrukcije ob spremembi poljubnih parametrov konstrukcije (dimenzij, materiala, obtežbe). Zaradi visoke zahtevnosti analitične občutljivostne analize ti postopki še niso standardni del komercialnih programov za analizo konstrukcij po metodi končnih elementov, vendar so bistveni del učinkovitih numeričnih algoritmov za optimizacijo konstrukcij.

Poznamo variacijsko in diskretno metodo občutljivostne analize. Pri variacijski metodi na začetku izračunamo odvode, nato pa jih diskretiziramo. Pri diskretni metodi pa na začetku enačbe diskretiziramo, šele nato izračunamo njihove odvode. Odvode lahko izračunamo na več načinov.

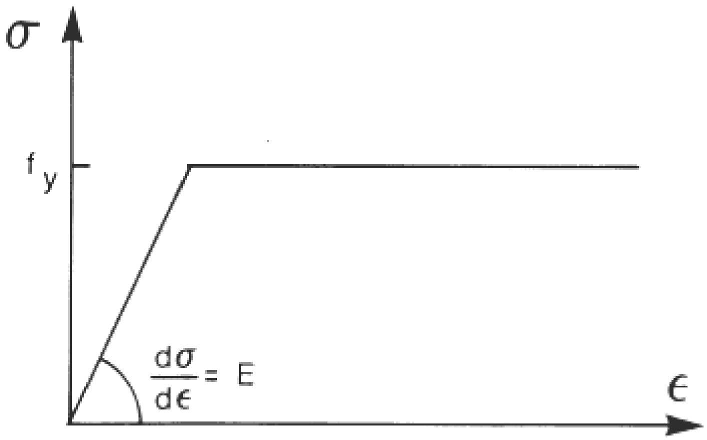
Občutljivostne analize lahko ločimo med seboj po izbiri parametrov. Analizo lahko izvedemo glede na različne parametre. Kadar so odzivne količine pomiki, govorimo o občutljivosti pomikov. Namesto pomikov lahko izberemo kateri koli drug parameter. V vseh analizah, ki so bile narejene za potrebe te naloge, smo za odzivno količino izbrali parametre oblike, katerim smo tekom analize spreminjali vrednosti. V tem primeru govorimo o dimenzijski oziroma oblikovni občutljivosti.

Kot je bilo omenjeno že v prejšnjem poglavju, potrebujemo za izvajanje občutljivostne analize gradient, ki nam ga daje polje začetnih občutljivosti z odvajanjem koordinat mreže končnih elementov po izbranih parametrih $\phi_i$. Analiza je del optimizacijskega postopka, ki ga izvedemo z ukazom `FindMinimum`[5] in se izvede v vsakem iteracijskem koraku. Gradient izračunamo kot odvod namenske funkcije $\Phi$ po projektnih parametrih $\phi_i$. V `Mathematici` to izvedemo z ukazom `Map[D[Φ,#]&,ϕ]`.

Gradient je v tem primeru osnova za spreminjanje parametrov; pove nam namreč, v katero smer naj spreminjamo parameter (naj ga povečujemo ali zmanjšujemo). Posamezna komponenta gradienta ustreza spremembi posameznega parametra.

### 3.5 Materialni model

V direktni analizi smo uporabili elasto-plastičen material. Vhodni podatki so ustrezali lastnostim dejanskih konstrukcijskih jekel ($\sigma_Y = 23{,}5 kN/cm^2$, $E = 21000 kN/cm^2$ in $\nu = 0{,}3$).

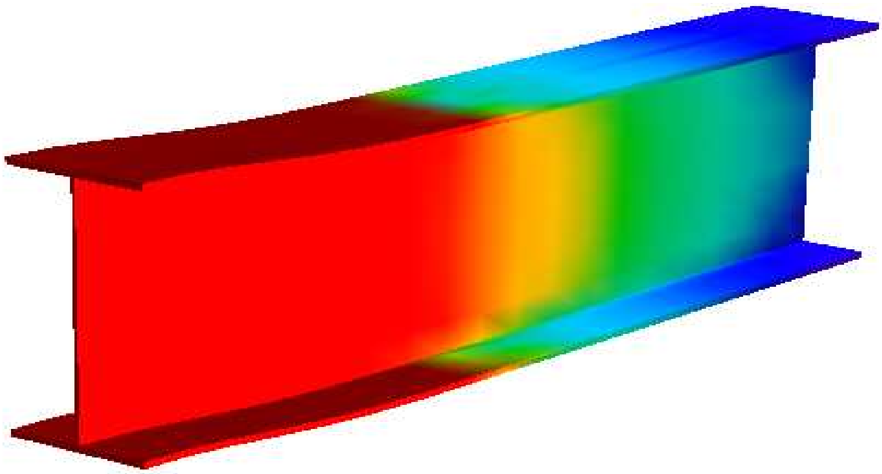



Slika 16: Diagram σ-ε za elasto-plastičen material

[5] Ukaz v programu `Mathematica` s katerim iščemo minimum določene funkcije.

## 3.6 Direktna analiza

Optimizacijski algoritem z ukazom FindMinimum zahteva ločeno izvajanje direktne in občutljivostne analize. Pri obeh analizah uporabljamo enako mrežo končnih elementov.



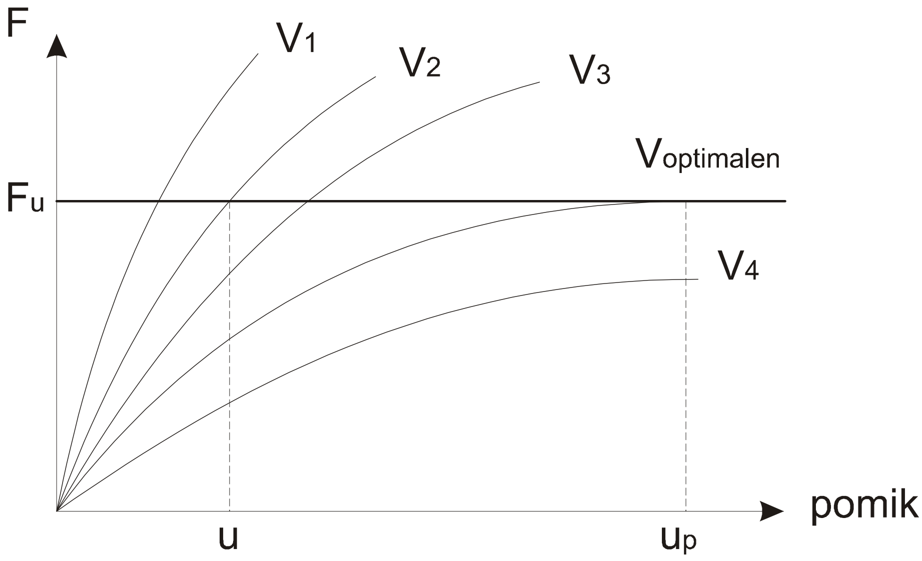
Slika 17: Primer rezultata v programu AceFEM, prikaz Misseovih napetosti

Analizo izvede programsko okolje `AceFEM`. Osnovni rezultat direktne analize so pomiki konstrukcije, ki so izračunani za vsako vozlišče posebej. V nadaljevanju program samodejno izračuna tudi ostale notranje količine konstrukcijskega elementa.

## 3.7 Optimizacija na mejno obtežbo

Kot je bilo omenjeno že v uvodu, želimo obliko konstrukcijskega elementa med drugim optimizirati tudi na mejno obtežbo. Upoštevanje mejne obtežbe pomeni, da pri obremenjevanju konstrukcije pademo v območje plastičnosti in konstrukcijo obremenjujemo vse do porušitve. Pri običajnih postopkih optimizacije upoštevamo mejo elastičnosti kot mejo do katere obremenjujemo konstrukcijo. Prednost takega postopka je linearno obnašanje materiala v tem območju. Obremenjevanje konstrukcije čez mejo elastičnosti, enostavne linearne probleme prevede v nelinearnost. Postopki postanejo kompleksnejši in zahtevnejši, bistveno pa se podaljšajo časi računskih postopkov.

Da bi bil optimizacijski algoritem sposoben doseči mejno obtežbo (Slika 18), je potrebno dodati dva vnaprej pripravljena modela končnih elementov, in sicer model, ki konstrukcijo obremenjuje z linijsko obtežbo ter model, ki v predpisani točki meri pomik. Končni element linijske obtežbe prvotnemu sistemu enačb doda vezno enačbo, s katero je algoritem nato sposoben konstrukcijo voditi do same porušitve.




Slika 18: Iskanje optimalnega volumna v odvisnosti od pomika

### 3.8 Optimizacija na maksimalno togost pri danem volumnu

Druga izmed obravnavanih možnosti optimizacije oblike konstrukcijskega elementa, opisana v tej nalogi, je optimizacija na maksimalno togost pri danem volumnu. Konstrukcijo obremenjujemo v elastičnem območju in merimo pomik v izbrani točki. Optimizacijski algoritem nato spreminja projektne parametre oblike tako, da je pomik v izbrani točki minimalen, volumen konstrukcije pa ostaja konstanten. Tako dobljena konstrukcija predstavlja optimalno izbiro parametrov, ki da pri danem volumnu najbolj togo konstrukcijo.

### 3.9 Namenska funkcija

Optimizacijo konstrukcije prevedemo na problem iskanja minimuma namenske funkcije. Za optimiziranje konstrukcij lahko uporabimo poljubno število funkcij, vendar smotrna izbira le-teh bistveno vpliva na zahtevnost ter predvsem dolžino samega izračuna. V našem primeru smo se odločili, da namensko funkcijo zapišemo kot vsoto treh funkcij $\Phi = \Phi_1 + \Phi_2 + \Phi_3$. Prva zahteva, katero moramo upoštevati pri iskanju minimuma, je minimalna razlika med zahtevano mejno obtežbo in dejansko obtežbo, ki jo prenese trenutna geometrija ($\Phi_1$). Optimiziranje konstrukcij izvajamo zaradi želje po uporabi manjše količine materiala. Zato je na tem mestu smiselno vpeljati drugo funkcijo, ki opisuje volumen konstrukcije ($\Phi_2$). Kot tretjo pa je smiselno vpeljali funkcijo v obliki kazenske funkcije, katera skrbi, da so izbrani parametri vedno nad prej določeno minimalno vrednostjo ($\Phi_3$). V praksi to pomeni, da ne moremo izbirati negativnih debelin.

V samem računu optimizacije minimiziramo namensko funkcijo $\Phi = \Phi_1 + \Phi_2 + \Phi_3$. Oblika posamezne funkcije $\Phi_i$ je opisana pri vsakem primeru posebej.

## 3.10 Račun lokalnega minimuma

Postopek optimizacije prevedemo na račun lokalnega minimuma funkcije $\Phi$. Račun izvedemo v programu `Mathematica` s funkcijo `FindMinimum`[6], pri kateri uporabimo gradientno metodo iskanja minimuma. Kot je bilo že omenjeno v predhodnih poglavjih, v postopku iskanja minimuma uporabimo gradient, ki ga dobimo kot rezultat občutljivostne analize. Ta gradient nam v postopku daje smer spreminjanja velikosti parametrov $\phi_i$ (manjšanje oziroma večanje vrednosti parametrov $\phi_i$).

Algoritem za iskanje minimuma samodejno izvaja občutljivostno analizo (poglavje 3.4) in direktno analizo (poglavje 3.6) po iteracijah. Obe analizi se izvajata v okolju za numerično analizo konstrukcij po metodi končnih elementov (program `AceFEM`).

Ukaz `FindMinimum` išče lokalni minimum funkcije $\Phi = \Phi_1 + \Phi_2 + \Phi_3$. Funkcija $\Phi$ se izvrednoti v vsakem iteracijskem koraku drugače.

```
FindMinimum[analysis["d",φs],{φs,φinit},MaxIterations→600,
Gradient→analysis["s",φs],"Method"→{{{"QuasiNewton",
"StepControl"→"LineSearch"}},{"QuasiNewton"},{"Automatic"}}[[3]]]
```

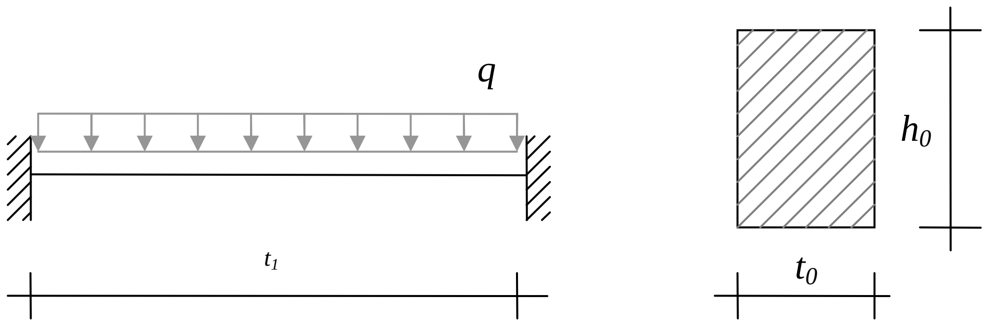
Slika 19: Algoritem za iskanje minimuma, uporabljen v programu Mathematica

[6] `FindMinimum` je ukaz v programu `Mathematica`

# 4 OPTIMIZACIJA TOPOLOGIJE

Pri optimiziranju oblike konstrukcije lahko uporabimo več različnih pristopov optimiziranja. Imamo veliko možnosti izbire namenskih funkcij, s katerimi pridemo do želenega rezultata. V opisanih primerih smo kot namensko funkcijo uporabljali minimalen volumen in minimalni pomik. Vsaka izmed funkcij ima svoje lastnosti, ki jih izkoriščamo, vedno pa namensko funkcijo uporabimo tako, da iščemo njen minimum. Glavni namen naloge je optimizirati obliko konstrukcijskega elementa. Za dosego tega cilja smo si izbrali več različnih statičnih primerov, jih različno obremenili in preučili dobljene rezultate.

Izbrali smo različne statične modele konstrukcije (konzolno vpeta konstrukcija, prekladno vpeta konstrukcija, konstrukcija stebra), katere smo obtežili z različnimi obtežbami. V modelu nismo upoštevali geometrijske nelinearnosti in s tem zanemarili vse fenomene, povezane s stabilnostjo (izbočenje, bočna zvrnitev, uklon, …). Začetni prečni prerez konstrukcije je vedno pravokotnik dimenzij $h_0 \cdot t_0$ .

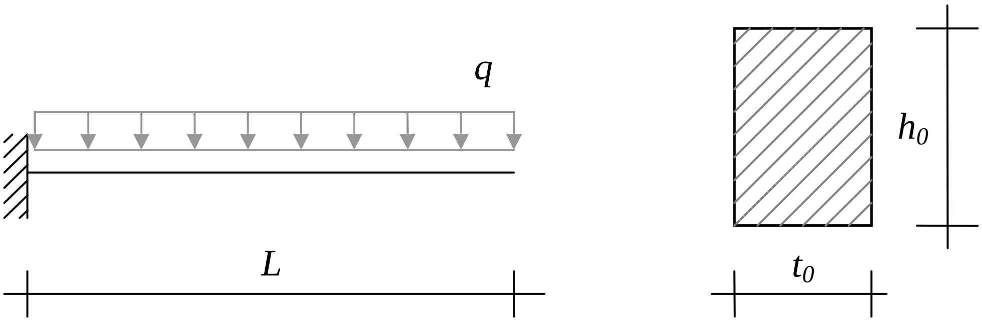



Slika 20: Primer začetne geometrije

Konstrukcijo smo opisali s parametri, in sicer z mrežo debelin $t_i$ ter z višino na začetku $h_1$ in višino na koncu $h_2$. Pri izvajanju samega optimizacijskega algoritma se je pokazala potreba po izbiri različnih projektnih parametrov. Tako smo z različno razporeditvijo in izbiro projektnih parametrov izdelali večje število primerov in jih med seboj primerjali.

Iz rezultatov posameznih analiz je jasno razvidna optimalna oblika konstrukcijskega elementa, ki se približuje obliki I oz T.

## 4.1 Optimizacije konzole na mejno obtežbo




Slika 21: Začetna geometrija konzolno vpetega nosilca

Kot je bilo že omenjeno, smo obravnavali konzolo dolžine $L_0$ in prečnih dimenzij $h_0 \cdot t_0$ (Slika 21). Izbrali smo dve kazenski funkciji, in sicer kazensko funkcijo razlike obtežb $\Phi_1$ in kazensko funkcijo za minimalno dimenzijo parametrov $\Phi_3$ ter funkcijo dejanskega volumna konstrukcije $\Phi_2$. Vse tri funkcije so podrobneje opisane v nadaljevanju.

Optimizacijski algoritem obremenjuje konstrukcijo z linearno obtežbo in računa spremembo med mejno obtežbo trenutno izbrane konstrukcije in predpisano mejno obtežbo, kar predstavlja eno izmed kazenskih funkcij. Med samim postopkom se izračuna tudi dejanski volumen izbrane konstrukcije, ki predstavlja namensko funkcijo.

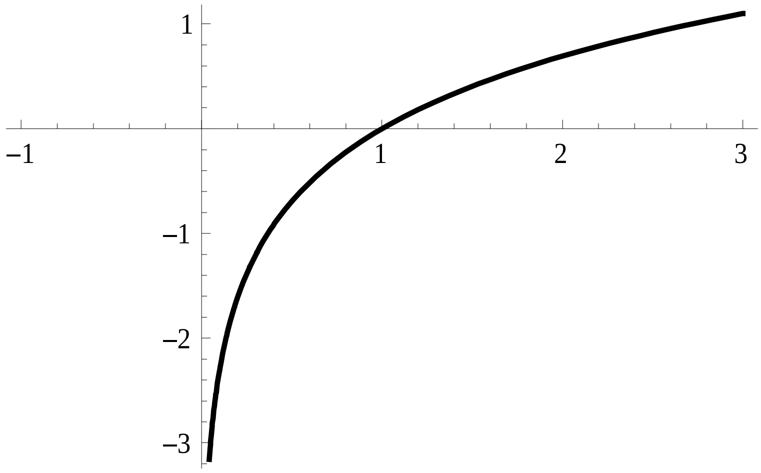
Optimiziranje oblike konstrukcije poteka iterativno, kar pomeni, da se v vsakem koraku projektni parametri spremenijo in račun požene na novo. Tako smo v vsakem izmed iteracijskih korakov bližje rešitvi. Račun se zaključi, ko se doseže zahtevana natančnost.

### 4.1.1 Namenska funkcija

#### 4.1.1.1 Kazenska funkcija obtežbe $\Phi_1$

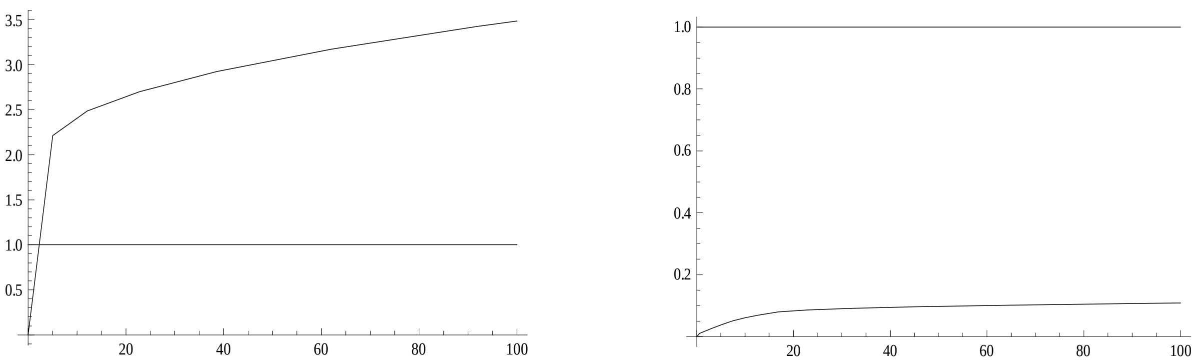
Prva kazenska funkcija kaznuje razliko med predpisano mejno obtežbo in mejno obtežbo, katero je trenutna konstrukcija sposobna prenesti. Pri izbiri kazenske funkcije smo omejeni z zahtevami, katere naj bi ta funkcija imela. Pri kazenski funkciji obtežbe želimo, da so v optimizacijskem procesu vmesni koraki vedno nad predpisano obtežbo. Kot je bilo že omenjeno, pri nadaljnjem izračunu uporabljamo gradiente, ki predstavljajo odvode namenske funkcije, kateri nam povejo ali se bodo parametri manjšali ali večali.

Za naše potrebe smo si izbrali matematično funkcijo logaritem (Slika 22). Sama oblika funkcije $Log(x)$ ne zadošča za potrebe namenske funkcije, zato smo jo malenkostno spremenili.




Slika 22: Funkcija *Log(x)*

S funkcijo obtežbe želimo doseči takšno izbiro parametrov, da bo obtežba, ki jo prenese trenutna izbira geometrije, vedno večja od predpisane vrednosti (Slika 23). Pri odvodu funkcije bi radi dobili negativne vrednosti. Slednje bi pomenilo, da optimizacijski algoritem povečuje parametre in s tem zadržuje dejansko krivuljo obtežbe nad izbrano vrednostjo. To storimo z izbiro $-Log(x)$. Naslednja težava nastopi pri definiranju funkcije pri vrednostih $x<0$, kjer osnovna funkcija $Log(x)$ ni definirana. V tem primeru osnovno funkcijo razvijemo v vrsto druge stopnje okoli vrednosti $\varepsilon load$. S tem algoritem doda parametru zelo veliko utež v primeru, ko krivulja dejanske obtežbe pade pod izbrano vrednost (Slika 23) in jo v naslednjem koraku zopet dvigne nad predpisano vrednost.

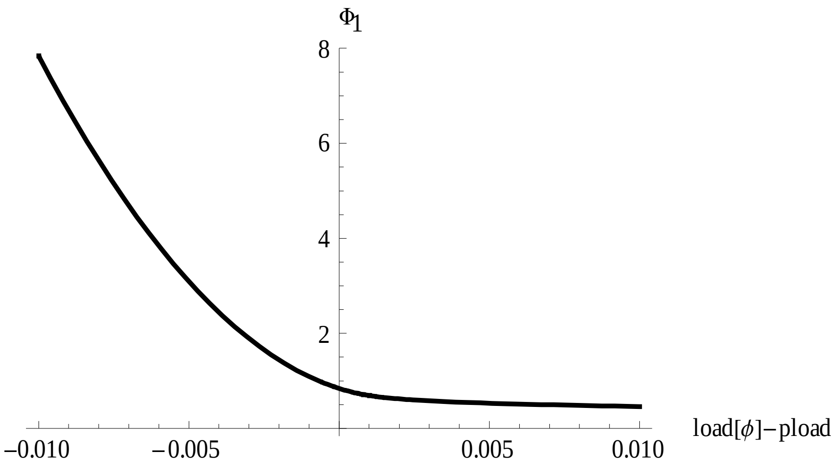



Slika 23: Prikaz dejanske obtežbe nad predpisano vrednostjo (levo) in pod predpisano vrednostjo (desno)

$$\Phi_1 = \begin{cases} -\mu_1 \log(\varepsilon load) - \dfrac{\mu_1 \log(load(\phi_i) - pload - \varepsilon load)}{\varepsilon load} + \dfrac{\mu_1 \log(load(\phi_i) - pload - \varepsilon load)^2}{2\varepsilon load^2}, \\ load(\phi_i) - pload \le \varepsilon load \\ \\ -\mu_1 \log(load(\phi_i) - pload), load(\phi_i) - pload > \varepsilon load \end{cases}$$

| | |
|---|---|
| $\phi_i$ | … parameter oblike |
| $\mu_1$ | … faktor |
| $\varepsilon load$ | … vrednost pri kateri spremenimo funkcijo log v vrsto |
| $load(\phi_i)$ | … obtežba, katero je trenutna konstrukcija sposobna prenesti |
| $pload$ | … predpisana obtežba |

Slika 24: Končna oblika kazenske funkcije $\Phi_1$




Slika 25: Prikaz kazenske funkcije $\Phi_1$

#### 4.1.1.2 Funkcija volumna $\Phi_2$

Druga funkcija je funkcija volumna, s katero želimo minimizirati dejanski volumen konstrukcije. Od samega rezultata optimizacijskega postopka želimo konstrukcijo, za izdelavo katere bo potrebno porabiti minimalen volumen materiala. Za izračun dejanskega volumna konstrukcije uporabimo že vgrajeno funkcijo programa `AceFEM`, in sicer `SMTTask["VolumeSensitivity"]`. Rezultat klicane funkcije je vrednost volumna trenutne oblike konstrukcije glede na izbrane parametre.

Ker pri samem optimizacijskem algoritmu uporabljamo odvode namenske funkcije, tudi v primeru funkcije volumna izračunamo odvode. Slednji skupaj z ostalimi odvodi funkcij predstavljajo gradient, uporabljen v optimizacijskem algoritmu. Odvod funkcije volumna po parametrih oblike $\phi_i$ kot rezultat vrne vektor, katerega uporabimo za določitev smeri spremembe posameznega parametra. Parametri so izbrani tako, da so odvodi funkcije volumna vedno pozitivne vrednosti, zato od samega algoritma zahtevajo stalno zmanjševanje vrednosti parametrov.

#### 4.1.1.3 Kazenska funkcija $\Phi_3$

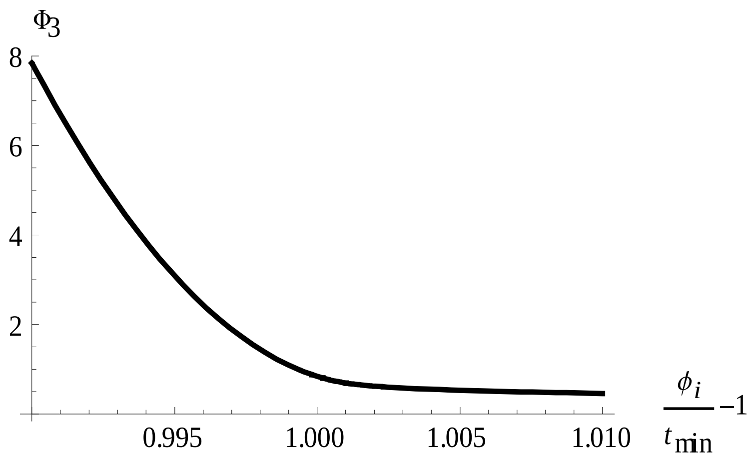
Kazenska funkcija $\Phi_3$ je tretja funkcija, katero smo uporabili v optimizacijskem algoritmu. Kazenska funkcija skrbi, da vrednosti posameznih parametrov oblike $\phi_i$ ne padejo pod predpisano minimalno vrednostjo $t_{\min}$. Podobno kot pri kazenski funkciji razlike obtežbe $\Phi_1$ tudi za kazensko funkcijo izberemo obdelano funkcijo $Log(x)$.

V primeru da vrednost določenega parametra $\phi_i$ pade pod minimalno vrednost $t_{min}$, funkcija pripiše gradientu veliko vrednost in tako se vrednost $\phi_i$ v naslednjem koraku vrne nad vrednost $t_{min}$. Odvodi vrednosti kazenske funkcije $\Phi_3$ so ves čas negativni in nam vrednosti posameznih parametrov povečujejo. Ker so vrednosti zelo majhne, ne vplivajo bistveno na sam rezultat optimizacije; razen v primeru, ko pade vrednost parametra pod minimalno vrednost $t_{min}$.

$$\Phi_3 = \begin{cases} -\mu_2 \log(\varepsilon load) - \dfrac{\mu_2\left(\phi_i - t_{\min}(1+\varepsilon load)\right)}{\varepsilon load \cdot t_{\min}} + \dfrac{\mu_2\left(\phi_i - t_{\min}(1+\varepsilon load)\right)^2}{2\cdot \varepsilon load^2 \cdot t_{\min}^2}, & \dfrac{\phi_i}{t_{\min}} - 1 \le \varepsilon load \\ -\mu_2 \log\left(\dfrac{\phi_i}{t_{\min}} - 1\right), & \dfrac{\phi_i}{t_{\min}} - 1 > \varepsilon load \end{cases}$$

$\phi_i$ … parametri oblike
$\mu_2$ … faktor
$\varepsilon load$ … vrednost pri kateri spremenimo funkcijo log v vrsto
$t_{\min}$ … predpisana minimalna vrednost parametrov oblike

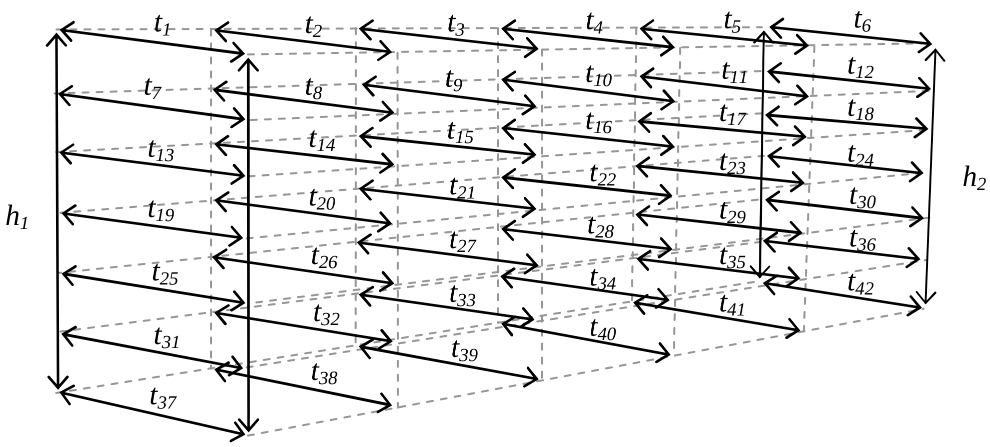
Slika 26: Končna oblika kazenske funkcije $\Phi_3$




Slika 27: Prikaz kazenske funkcije $\Phi 2$

### 4.1.2 Postopek računa

Kot je bilo že omenjeno, smo račun izvajali v okolju `Mathematica`[7], in sicer s podprogramom `AceFEM`[8]. Algoritem celotnega optimizacijskega algoritma je zapisana v poglavju Priloga A: Algoritem optimizacije topologije – mejna obtežba.



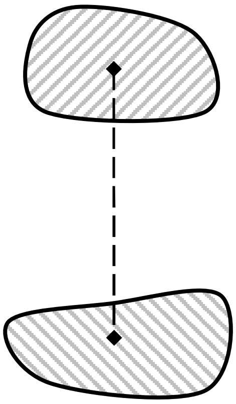

Slika 28: Prikaz izbire parametrov

Konstrukcijo smo opisali s parametri, in sicer z mrežo debelin $t_i$ ter z višino na začetku $h_1$ in višino na koncu $h_2$. Za opis mreže debelin $t_i$ smo uporabili 6 parametrov $\phi_i$ v smeri osi konstrukcije in 7 parametrov $\phi_i$ v smeri višine konstrukcije. Tako smo optimizirali 42 parametrov debeline in 2 parametra višine, torej skupaj 44 parametrov $\phi_i$ (Slika 28).

---

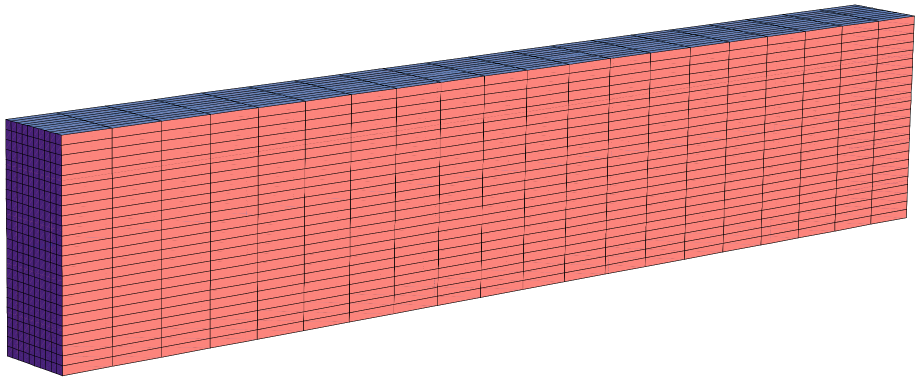
[7] http://www.wolfram.com/
[8] http://www.fgg.uni-lj.si/Symech/

Ker upoštevamo samo materialno nelinearnost (stabilnost zanemarimo), je očiten rezultat tako definiranega problema, da sta dve masi na čim večji medsebojni razdalji (Slika 29). Tak optimizacijski problem je slabo pogojen, zato moramo izvesti dodatne omejitve. Predvsem smo analizirali vpliv odstranitve ali fiksiranje posameznih parametrov in s tem spremenili potek računa.



Slika 29: Optimalna topologija brez upoštevanja stabilnosti

Začetne vrednosti parametrov so bile v vseh primerih enake; $t_i = t_0 = 40cm$ in $h_i = h_0 = 100cm$ (Slika 30). Enake začetne vrednosti nam dajejo možnost primerjave različnih vplivov izbire posameznih parametrov na končno obliko konstrukcijskega elementa.



Slika 30: Začetna geometrija v programu AceFEM, kvader dimenzij $L_0$, $t_0$, $h_0$

### 4.1.3 Postopek optimiziranja

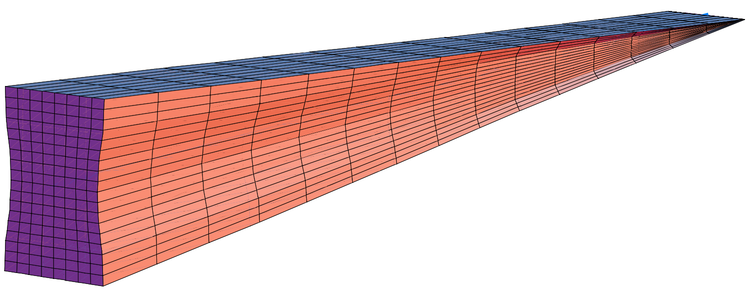
Sledi podroben opis posameznih variant in predstavitev rezultatov. Pri vsaki varianti je tudi podrobno opisan način izbire parametrov. Rezultati so predstavljeni v obliki slik in preglednic izbranih parametrov. Vsi rezultati dobljenih parametrov so opisani v preglednicah v obliki, kot jo kaže spodnja slika (Slika 31).

$$\begin{Bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 \\ \phi_7 & \phi_8 & \phi_9 & \phi_{10} & \phi_{11} & \phi_{12} \\ \phi_{13} & \phi_{14} & \phi_{15} & \phi_{16} & \phi_{17} & \phi_{18} \\ \phi_{19} & \phi_{20} & \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} \\ \phi_{25} & \phi_{26} & \phi_{27} & \phi_{28} & \phi_{29} & \phi_{30} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} & \phi_{36} \\ \phi_{37} & \phi_{38} & \phi_{39} & \phi_{40} & \phi_{41} & \phi_{42} \end{Bmatrix}$$

Slika 31: Zapis parametrov debeline $t_i$

#### 4.1.3.1 Slabo pogojen optimizacijski problem

Analizo smo začeli s konstrukcijo, katero opisuje vseh 44 parametrov $\phi_i$ (42 parametrov debeline $t_i$ ter 2 parametra višine $h_i$) . Algoritem je dobro zmanjšal višino po prerezu, saj je obremenitev na koncu najmanjša in zato potreba po prerezu minimalna. Začetna višina $h_0 = 100cm$ se je pri vpetju spremenila na $h_1 = 66{,}97cm$ ter na koncu konstrukcije $h_2 = 0{,}01cm = h_{\min}$. Sprememba parametrov po širini je minimalna in skoraj neopazna. Končni volumen izbrane geometrije znaša $V = 0.629\text{m}^3$. Rezultat ni zadovoljiv, saj je optimizacija slaba. Definiranje problema je kompleksno in algoritem je konvergiral k napačni točki. Oblika končne geometrije in dobljeni rezultati so predstavljeni v nadaljevanju (Slika 32 ter Preglednica 1).



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 3/8/3
- parametra višine $h_1$ in $h_2$ sta prosta
- vsi parametri debeline $t_i$ so prosti

Slika 32: Končna geometrija izbranega konstrukcijskega elementa – *varianta 1.1*

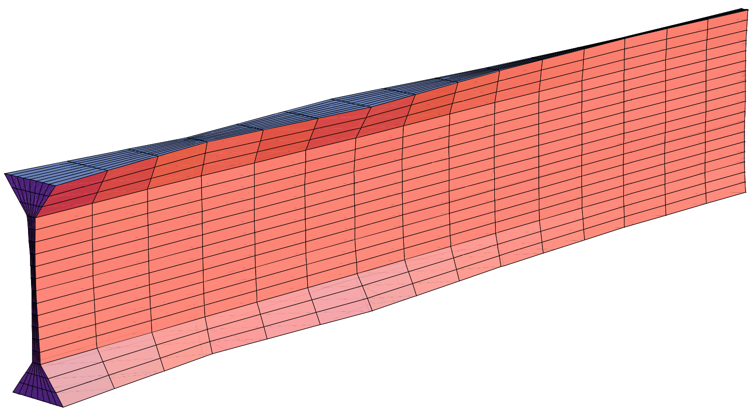
Analiza je potrebovala 69 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 10 ur in 38 minut. Analiza pokaže na kompleksnost optimizacije kot take, saj tudi za slabo definiran optimizacijski problem dobimo rezultate. Tega problema analiza ne detektira.

Preglednica 1: Vrednost posameznih parametrov debeline $t_i$ – *varianta 1.1*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 43,06 | 43,78 | 40,92 | 36,56 | 36,39 | 38,71 |
| $h = 5/6\ h_0$ | 42,20 | 42,59 | 38,50 | 31,96 | 31,77 | 36,42 |
| $h = 4/6\ h_0$ | 38,85 | 37,90 | 35,36 | 31,82 | 31,77 | 36,42 |
| $h = 3/6\ h_0$ | 37,59 | 35,55 | 33,79 | 31,74 | 31,77 | 36,42 |
| $h = 2/6\ h_0$ | 38,85 | 37,90 | 35,36 | 31,82 | 31,77 | 36,42 |
| $h = 1/6\ h_0$ | 42,20 | 42,59 | 38,50 | 31,96 | 31,77 | 36,42 |
| $h = 0\ h_0$ | 43,06 | 43,78 | 40,92 | 36,56 | 36,39 | 38,71 |

#### 4.1.3.2 Dobro pogojen optimizacijski problem

Za dosego boljših rezultatov smo parametra višine $h_1$ in $h_2$ fiksirali in jima določili vrednost $h_0$. To pomeni, da je algoritem spreminjal samo parametre debeline $t_i$, višine pa so ostale konstantne skozi celoten postopek računa. Spreminjanje parametrov po dolžini ni zvezno. V primerjavi s prejšnjo analizo je optimizacijski problem dobro definiran in je že možno zaznati konvergiranje k obliki I. Poleg oblike I pa je prisotno tudi zmanjševanje parametrov debeline proti koncu konstrukcije, kjer so obremenitve najmanjše. Končni volumen izbrane geometrije znaša $V = 0.237\text{m}^3$. Končna geometrija je prikazana v nadaljevanju (Slika 33).



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 3/8/3
- parametra višine $h_1$ in $h_2$ sta konstantna
- vsi parametri debeline $t_i$ so prosti

Slika 33: Končna geometrija izbranega konstrukcijskega elementa – *varianta 1.2*

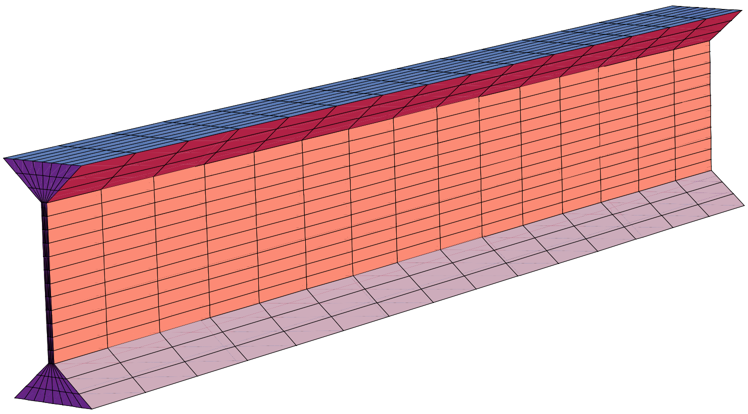
Analiza je potrebovala 191 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 48 ur in 4 minute.

Preglednica 2: Vrednost posameznih parametrov debeline $t_i$ – *varianta 1.2*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 33,58 | 13,07 | 26,46 | 9,41 | 2,20 | 4,75 |
| $h = 5/6\ h_0$ | 5,37 | 3,96 | 5,69 | 2,81 | 1,20 | 2,57 |
| $h = 4/6\ h_0$ | 3,24 | 3,91 | 2,73 | 1,90 | 1,16 | 2,37 |
| $h = 3/6\ h_0$ | 2,74 | 3,32 | 2,81 | 1,80 | 1,53 | 2,69 |
| $h = 2/6\ h_0$ | 3,84 | 3,36 | 2,75 | 2,16 | 0,90 | 2,29 |
| $h = 1/6\ h_0$ | 5,32 | 4,36 | 4,86 | 3,03 | 1,39 | 2,50 |
| $h = 0\ h_0$ | 33,67 | 13,15 | 24,55 | 10,21 | 2,67 | 4,73 |

#### 4.1.3.3 Reduciran set parametrov

Prejšnja analiza je izbirala parametre debeline po dolžini izredno nezvezno, zato smo se odločili, da debelino zgornjega in spodnjega pasu fiksiramo in jima dodelimo konstantno vrednost $t_0$. Prav tako smo pustili konstantni vrednosti $h_1$ in $h_2$. S tem smo prvotnih 44 parametrov zmanjšali na 30 parametrov, kar se močno pozna tudi na samem času izračuna. Končni volumen izbrane geometrije znaša $V = 0.415\text{m}^3$. Dobljeni rezultat je prikazan v nadaljevanju (Slika 34).

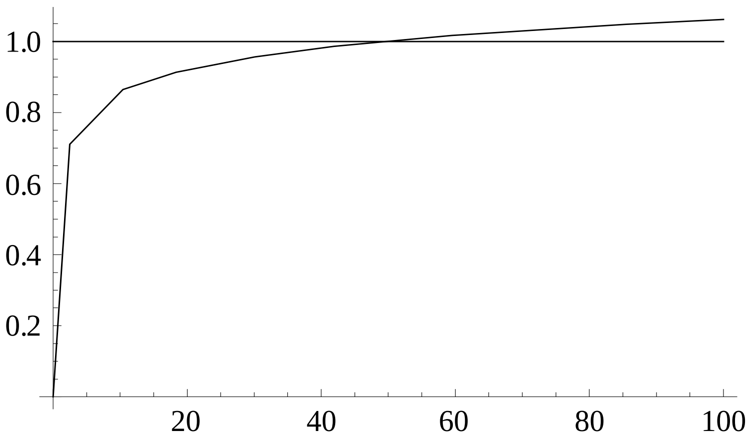


*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 3/8/3
- parametra višine $h_1$ in $h_2$ sta konstantna
- parametri debeline $t_i$ v zgornjem in spodnjem pasu so konstantni

Slika 34: Končna geometrija izbranega konstrukcijskega elementa – *varianta 1.3*

Analiza je potrebovala 186 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 47 ur in 19 minut.




Slika 35: Odvisnost med pomikom in obtežbo za končno geometrijo – *varianta 1.3*

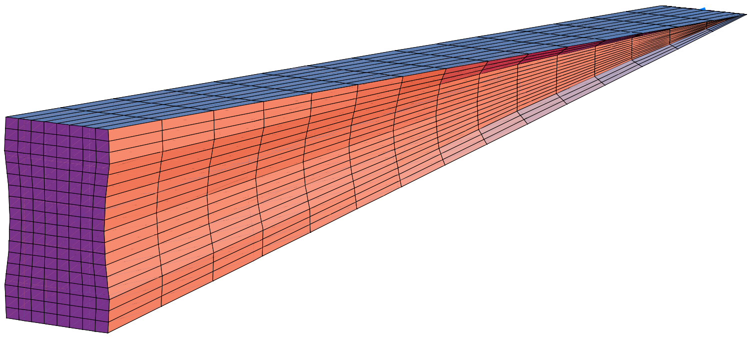
Optimizacijski algoritem med samim izračunom v vsakem iteracijskem koraku izpisuje graf odvisnosti med pomikom in obtežbo. Slika 35 prikazuje odvisnost med pomikom in obtežbo za končno izbiro parametrov $\phi_i$.

Preglednica 3: Vrednost posameznih parametrov debeline $t_i$ – *varianta 1.3*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 |
| $h = 5/6\ h_0$ | 2,82 | 1,96 | 2,25 | 1,57 | 1,30 | 2,53 |
| $h = 4/6\ h_0$ | 2,78 | 2,02 | 2,24 | 1,59 | 1,26 | 2,48 |
| $h = 3/6\ h_0$ | 2,82 | 1,99 | 2,35 | 1,57 | 1,24 | 2,52 |
| $h = 2/6\ h_0$ | 2,88 | 2,03 | 2,31 | 1,60 | 1,27 | 2,48 |
| $h = 1/6\ h_0$ | 2,85 | 2,00 | 2,30 | 1,59 | 1,25 | 2,52 |
| $h = 0\ h_0$ | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 |

#### 4.1.3.4 Slabo pogojen optimizacijski problem

Zanimal nas je še odziv algoritma na prosti višini $h_1$ in $h_2$, vendar s konstantno širino zgornjega in spodnjega pasu. Primer se je izkazal za zelo podobnega varianti 1, z izbiro višin $h_1 = 60{,}20$ in $h_2 = h_{\min} = 0{,}01$. Končni volumen izbrane geometrije znaša $V = 0.617\text{m}^3$. Dobljeni rezultat je prikazan v nadaljevanju (Slika 36).



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 3/8/3
- parametra višine $h_1$ in $h_2$ sta prosta
- parametri debeline $t_i$ v zgornjem in spodnjem pasu so konstantni

Slika 36: Končna geometrija izbranega konstrukcijskega elementa – *varianta 1.4*

Analiza je potrebovala 61 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 8 ur in 55 minut.

Preglednica 4: Vrednost posameznih parametrov debeline $t_i$ – *varianta 1.4*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 |
| $h = 5/6\ h_0$ | 41,00 | 40,37 | 36,75 | 30,81 | 30,62 | 35,79 |
| $h = 4/6\ h_0$ | 38,00 | 36,20 | 33,90 | 30,68 | 30,62 | 35,79 |
| $h = 3/6\ h_0$ | 36,87 | 34,07 | 32,48 | 30,61 | 30,62 | 35,79 |
| $h = 2/6\ h_0$ | 38,00 | 36,20 | 33,90 | 30,68 | 30,62 | 35,79 |
| $h = 1/6\ h_0$ | 41,00 | 40,37 | 36,75 | 30,81 | 30,62 | 35,79 |
| $h = 0\ h_0$ | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 | 40,00 |

### 4.1.4 Povzetek

Metoda se je izkazala za matematično zelo zahtevno. Povprečni čas, potreben za izračun posameznega iteracijskega koraka, traja 10 minut. Pri tem se je potrebno zavedati, da je algoritem za rešitev potreboval tudi do 200 iteracijskih korakov, kar na koncu pomeni nekaj dni računanja.

Najmanjši volumen končne geometrije sicer dobimo v varianti 1.2, vendar je geometrija neuporabna za prakso. Zanimiv rezultat dobimo predvsem v varianti 1.3, kjer se najbolj nazorno pokaže konvergiranje k obliki I profila.

## 4.2 Optimizacije konzole na maksimalno togost pri danem volumnu

V naslednjih primerih je opisan nekoliko drugačen pristop k optimizaciji. Do sedaj smo spoznali gradientno metodo optimizacije na mejno obtežbo. Postopki so se izkazali za računalniško zahtevne in so za svojo izvršitev porabili veliko časa, zato smo se odločili, da metodo naredimo preprostejšo.

Pri obremenjevanju konstrukcije bomo merili pomik na koncu elementa, od katerega zahtevamo, da je kar se da majhen. Pri spreminjanju volumna zahtevamo, da končni volumen konvergira k vnaprej izbrani vrednosti (lahko tudi začetna vrednost $V_0$). S tem ko optimiziramo glede na minimalen pomik, lahko konstrukcijo analiziramo v elastičnem območju in postopek močno pospešimo.

Parametrizacija mreže končnih elementov ostaja nespremenjena, vse lastnosti ostajajo iste. Konstrukcijo smo opisali s projektnimi parametri oblike, in sicer z višino na začetku $h_1$ in višino na koncu $h_2$ ter z mrežo debelin $t_i$. Za opis mreže debelin $t_i$ smo uporabili 6 parametrov $\phi_i$ v smeri osi konstrukcije in 7 parametrov $\phi_i$ v smeri višine konstrukcije. Tako smo optimizirali 42 parametrov debeline in 2 parametra višine, torej skupaj 44 parametrov $\phi_i$.

Za potrebe te metode je bilo potrebno opisati novo namensko funkcijo. Izbrali smo nove tri funkcije, in sicer kazensko funkcijo volumna $\Phi_1$, funkcijo pomika konstrukcije na koncu $\Phi_2$ in kazensko funkcijo za minimalno dimenzijo parametrov $\Phi_3$. Vse tri funkcije so podrobneje opisane v nadaljevanju.

### 4.2.1 Namenska funkcija

Potrebno je bilo opisati tri nove funkcije, ki bodo opisale želen problem optimizacije. Omenjeno je bilo, da potrebujemo funkcijo volumna in pomika. Funkcija $\Phi_3$, ki skrbi za minimalne vrednosti projektnih parametrov $\phi_i$, ostane nespremenjena.

#### 4.2.1.1 Kazenska funkcija volumna $\Phi_1$

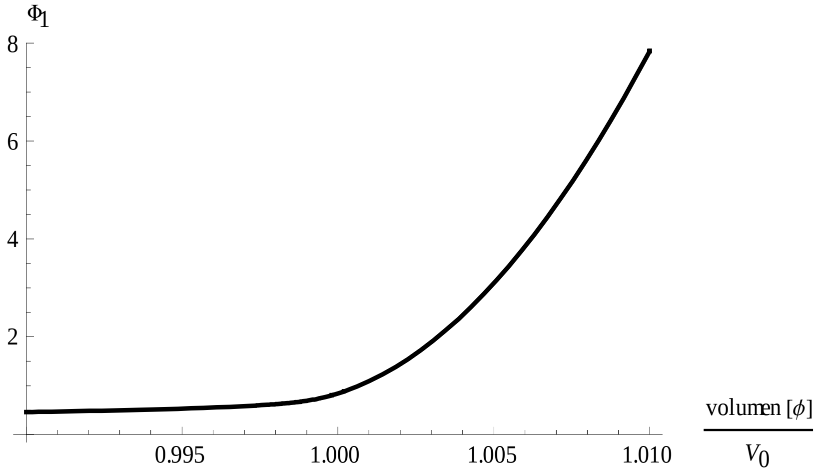
Osnovni namen definiranja kazenske funkcije volumna $\Phi_1$ je vzdrževati vrednost volumna blizu vrednosti $V_0$. Z razliko od prejšnjega primera – ko smo vpeljali kazensko funkcijo volumna, ki skrbi za minimalni volumen – tukaj vpeljemo kazensko funkcijo volumna $\Phi_1$, ki skrbi za optimalno razporeditev materiala po prerezu. To pomeni, da je optimizacijski algoritem prisiljen izbirati takšne projektne parametre $\phi_i$, da je volumen ves čas postopka manjši od $V_0$ in konvergira k izbrani vrednosti $V_0$, saj ga k temu prisili namenska funkcija, ki minimizira pomik.

Tudi tokrat za osnovno funkcijo izberemo $Log(x)$ (Slika 22), kjer spremenljivko predstavlja kvocient dejanskega volumna in izbranega volumna ($volume(\phi_i)/V_0$). Osnovno funkcijo $Log(x)$ smo zapisali v nekoliko spremenjeni obliki in jo v območju nedefiniranosti razvili v vrsto drugega reda. Ko se vrednost koeficienta $volume(\phi_i)/V_0$ dvigne nad vrednost $1-\varepsilon load$ funkcijo razvijemo v vrsto. Pri vrednostih več kot $1-\varepsilon load$ (Slika 38) postane krivulja strmejša, kar pomeni da odvod funkcije $\Phi_1$ hitro narašča. Veliki odvodi v optimizacijskem algoritmu pomenijo hitro zmanjšanje vrednosti projektnih parametrov, oziroma v tem primeru zmanjšanje volumna.

$$\Phi_1 = \begin{cases} -\mu_1 \log(\varepsilon load) - \dfrac{\mu_1 \log\left(\dfrac{V(\phi_i)}{V_0} - 1 + \varepsilon load\right)}{\varepsilon load} + \dfrac{\mu_1 \log\left(\dfrac{V(\phi_i)}{V_0} - 1 + \varepsilon load\right)^2}{2\varepsilon load^2}, & \dfrac{V(\phi_i)}{V_0} - 1 \geq 1 - \varepsilon load \\ -\mu_1 \log\left(-\dfrac{V(\phi_i)}{V_0}\right), & \dfrac{V(\phi_i)}{V_0} - 1 < 1 - \varepsilon load \end{cases}$$

| | |
|---|---|
| $\phi_i$ | … parameter oblike |
| $\mu_1$ | … faktor |
| $\varepsilon load$ | … vrednost pri kateri spremenimo funkcijo log v vrsto |
| $V(\phi_i)$ | … volumen trenutno izbrane oblike |
| $V_0$ | … začetni volumen |

Slika 37: Končna oblika kazenske funkcije $\Phi_1$




Slika 38: Prikaz kazenske funkcije $\Phi_1$

#### 4.2.1.2 Funkcija pomika $\Phi_2$

S funkcijo pomika $\Phi_2$ definiramo velikost pomika v izbrani točki konstrukcije. To storimo v programu `AceFEM` z ukazom `SMTNodeData`[9]. Ukaz v izbrani točki prebere pomik točke v izbrani smeri. Razporeditev materiala po prerezu bo optimalna takrat, ko bo merjeni pomik najmanjši. Ker obravnavamo konzolno konstrukcijo, merimo pomik v končni točki konstrukcije.

$$\boxed{\Phi_2 = Pomik(\phi_i)}$$

Slika 39: Končna oblika funkcije $\Phi_2$

#### 4.2.1.3 Kazenska funkcija $\Phi_3$

Kazenska funkcija $\Phi_3$ je tretja funkcija, katero smo uporabili v optimizacijskem algoritmu. Namen in oblika funkcije je popolnoma identična kazenski funkciji v poglavju 4.1.1.3. Funkcija skrbi, da vrednosti posameznih parametrov oblike $\phi_i$ ne padejo pod predpisano minimalno vrednost $t_{min}$.

[9] `SMTNodeData` je ukaz v programu `AceFEM`, ki prebere podatke shranjene v izbrani točki

$$\Phi_3 = \begin{cases} -\mu_2 \log(\varepsilon load) - \dfrac{\mu_2(\phi_i - t_{\min}(1+\varepsilon load))}{\varepsilon load \cdot t_{\min}} + \dfrac{\mu_2(\phi_i - t_{\min}(1+\varepsilon load))^2}{2 \cdot \varepsilon load^2 \cdot t_{\min}^2}, & \dfrac{\phi_i}{t_{\min}} - 1 \leq \varepsilon load \\ -\mu_2 \log\left(\dfrac{\phi_i}{t_{\min}} - 1\right), & \dfrac{\phi_i}{t_{\min}} - 1 > \varepsilon load \end{cases}$$

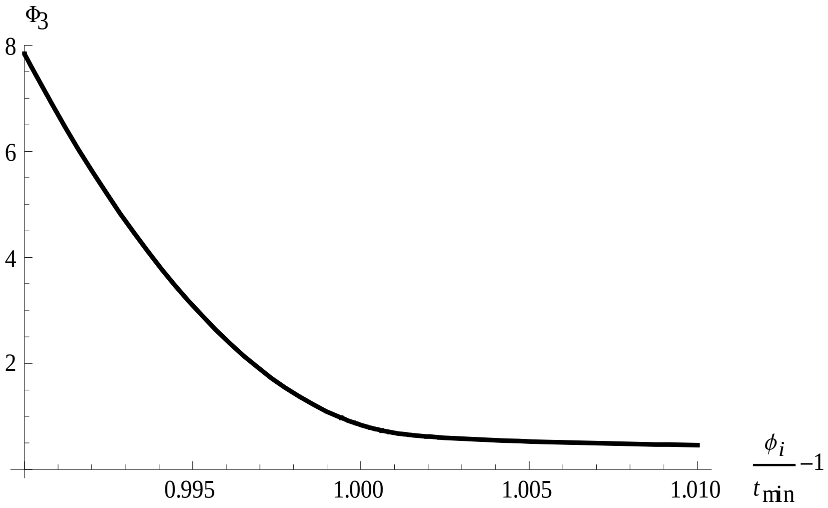
$\phi_i$ … parametri oblike

$\mu_2$ … faktor

$\varepsilon load$ … vrednost pri kateri spremenimo funkcijo log v vrsto

$t_{\min}$ … predpisana minimalna vrednost parametrov oblike

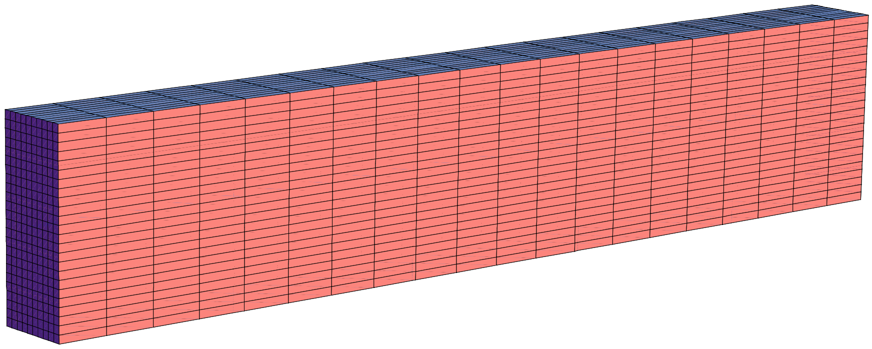
Slika 40: Končna oblika kazenske funkcije $\Phi_3$




Slika 41: Prikaz kazenske funkcije $\Phi_3$

### 4.2.2 Postopek računa

Pri iskanju optimalne oblike konstrukcijskega elementa smo zopet izvedli več različnih analiz in jih med sabo primerjali. Pri različnih analizah smo upoštevali različne parametre in se s tem približevali pravilnosti delovanja optimizacijskega algoritma. S spreminjanjem parametrov smo se omejili na parametre debeline $t_i$ in parametra višine $h_i$. Različni primeri upoštevajo različne parametre.



Slika 42: Začetna geometrija v programu AceFEM

V vseh opisanih primerih gre za konzolno vpetje konstrukcije dolžine $L_0 = 500cm$. Prečni prerez začetne geometrije $t_o \cdot h_0$ je dimenzij $t_0 = 60cm$ in $h_0 = 100cm$. Za opis mreže debelin $t_i$ smo uporabili 6 parametrov $\phi_i$ v smeri osi konstrukcije in 7 parametrov $\phi_i$ v smeri višine konstrukcije (Slika 43). Tako smo optimizirali 42 parametrov debeline in 2 parametra višine, torej skupaj 44 parametrov $\phi_i$.

$$\begin{Bmatrix} \phi_1 & \phi_2 & \phi_3 & \phi_4 & \phi_5 & \phi_6 \\ \phi_7 & \phi_8 & \phi_9 & \phi_{10} & \phi_{11} & \phi_{12} \\ \phi_{13} & \phi_{14} & \phi_{15} & \phi_{16} & \phi_{17} & \phi_{18} \\ \phi_{19} & \phi_{20} & \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} \\ \phi_{25} & \phi_{26} & \phi_{27} & \phi_{28} & \phi_{29} & \phi_{30} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} & \phi_{35} & \phi_{36} \\ \phi_{37} & \phi_{38} & \phi_{39} & \phi_{40} & \phi_{41} & \phi_{42} \end{Bmatrix}$$

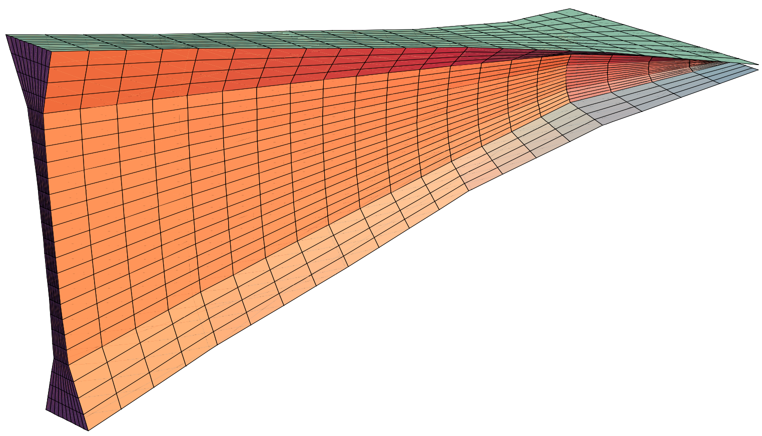
Slika 43: Simbolični prikaz parametrov višine $t_i$

Začetne vrednosti parametrov so bile v vseh primerih enake, $t_i = t_0 = 60cm$ in $h_i = h_0 = 100cm$ (Slika 30). Enake začetne vrednosti nam dajejo možnost primerjave različnih vplivov izbire posameznih parametrov na končno obliko konstrukcijskega elementa. Algoritem za opis problema v `Mathematici` je zapisan v poglavju Priloga B: Algoritem optimizacije topologije – maksimalna togost pri danem volumnu.

### 4.2.3 Postopek optimizacije

#### 4.2.3.1 Slabo pogojen optimizacijski problem

Analizo smo začeli ravno tako kot pri prejšnji metodi s konstrukcijo, katero opišemo z vsemi 44 parametri oblike $\phi_i$ (42 parametrov debeline $t_i$ ter 2 parametra višine $h_i$). Algoritem zopet pričakovano poveča začetno višino pri vpetju in zmanjša višino proti koncu konstrukcije. Tako se je začetna višina $h_0 = 100cm$ spremenila pri vpetju na $h_1 = 313{,}97cm$ in na koncu konstrukcije na $h_2 = 4{,}83cm$. Oblika končne geometrije in dobljeni rezultati so prikazani v nadaljevanju (Slika 44 oziroma Preglednica 5). Problem je slabo pogojen, saj je optimum očiten, ko gre višina nosilca proti neskončnosti.



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta prosta
- vsi parametri debeline $t_i$ so prosti

Slika 44: Končna geometrija izbranega konstrukcijskega elementa – *varianta 2.1*

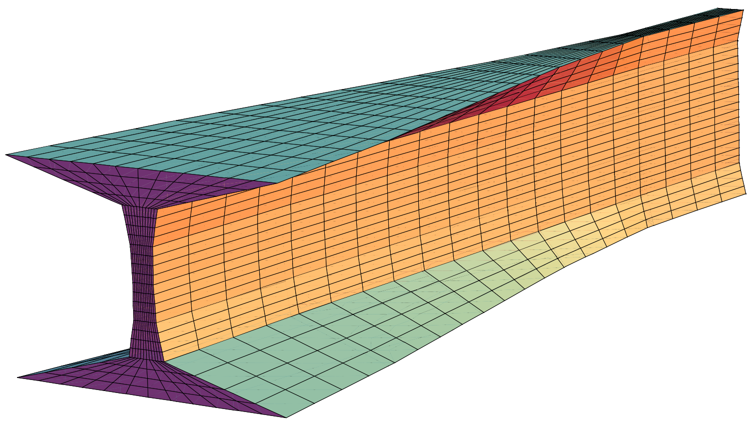
Analiza je potrebovala 592 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 27 ur 54 minut.

Preglednica 5: Vrednost posameznih parametrov debeline $t_i$ – *varianta 2.1*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 39,74 | 34,60 | 44,98 | 48,02 | 84,73 | 165,63 |
| $h = 5/6\ h_0$ | 13,53 | 9,51 | 12,26 | 17,71 | 27,97 | 123,96 |
| $h = 4/6\ h_0$ | 10,22 | 6,35 | 8,28 | 12,36 | 20,87 | 117,94 |
| $h = 3/6\ h_0$ | 9,64 | 5,69 | 7,34 | 10,70 | 19,77 | 117,65 |
| $h = 2/6\ h_0$ | 10,22 | 6,35 | 8,28 | 12,36 | 20,87 | 117,94 |
| $h = 1/6\ h_0$ | 13,53 | 9,51 | 12,26 | 17,71 | 27,97 | 123,96 |
| $h = 0\ h_0$ | 39,74 | 34,60 | 44,98 | 48,02 | 84,73 | 165,63 |

#### 4.2.3.2 Slabo pogojen optimizacijski problem

V naslednji varianti smo parametra višine $h_1$ in $h_2$ fiksirali in jima pripisali konstantno vrednost $h_0$. Slika 45 prikazuje končno obliko konstrukcijskega elementa. Jasno se vidi, da želi optimizacijski algoritem opisati obliko I profila. Širina zgornjega in spodnjega pasu z dolžino pada (Preglednica 6), kar je bilo pričakovano, saj obremenitve konstrukcije z dolžino padajo proti vrednosti nič na koncu.



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- vsi parametri debeline $t_i$ so prosti

Slika 45: Končna geometrija izbranega konstrukcijskega elementa – *varianta 2.2*

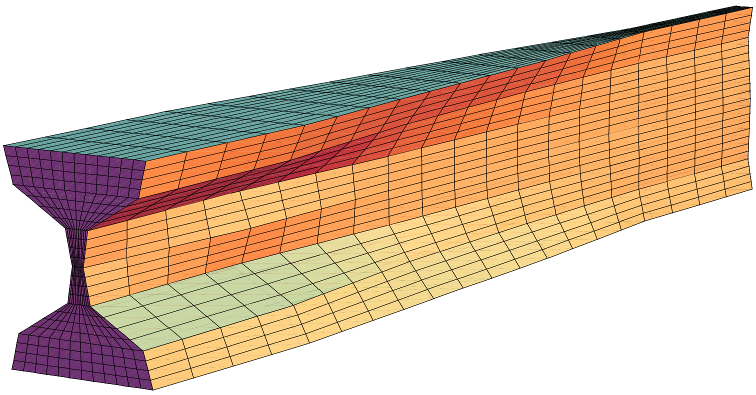
Analiza je potrebovala 97 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 30 minut. Problem je še vedno slabo pogojen, saj teži k neskončno široki pasnici.

Preglednica 6: Vrednost posameznih parametrov debeline $t_i$ – *varianta 2.2*

| | $L = 0\,L_0$ | $L = 1/5\,L_0$ | $L = 2/5\,L_0$ | $L = 3/5\,L_0$ | $L = 4/5\,L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 147,70 | 117,67 | 78,13 | 39,47 | 13,69 | 15,84 |
| $h = 5/6\,h_0$ | 19,23 | 13,55 | 12,06 | 10,12 | 6,30 | 8,19 |
| $h = 4/6\,h_0$ | 12,36 | 8,98 | 8,36 | 7,47 | 5,83 | 8,54 |
| $h = 3/6\,h_0$ | 11,21 | 8,29 | 7,74 | 6,92 | 5,62 | 8,66 |
| $h = 2/6\,h_0$ | 12,36 | 8,98 | 8,36 | 7,47 | 5,83 | 8,54 |
| $h = 1/6\,h_0$ | 19,23 | 13,55 | 12,06 | 10,12 | 6,30 | 8,19 |
| $h = 0\,h_0$ | 147,70 | 117,67 | 78,13 | 39,47 | 13,69 | 15,84 |

#### 4.2.3.3 Dobro pogojen optimizacijski problem

V primerjavi s prejšnjo varianto smo v optimizacijski algoritem vnesli dodatni pogoj, in sicer $t_i \le t_{\max}$. Ta dodatni pogoj poskrbi, da je oblika prečnega prereza konstrukcijskega elementa vedno znotraj predpisanega območja dimenzij $h_i \cdot t_{\max}$. Ravno tako pa imamo konstantna parametra višine $h_1$ in $h_2$ ter vrednost $h_0$. Slika 46 prikazuje končno obliko konstrukcijskega elementa. Algoritem je zopet jasno nakazal obliko I profila. Širina zgornjega in spodnjega pasu je sprva konstantna (zaradi potrebe po materialu izkoriščamo celotno dano širino), nato pa se postopoma zmanjšuje.



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- vsi parametri debeline $t_i$ so prosti
- $t_i \le t_{\max}$

Slika 46: Končna geometrija izbranega konstrukcijskega elementa – *varianta 2.3*

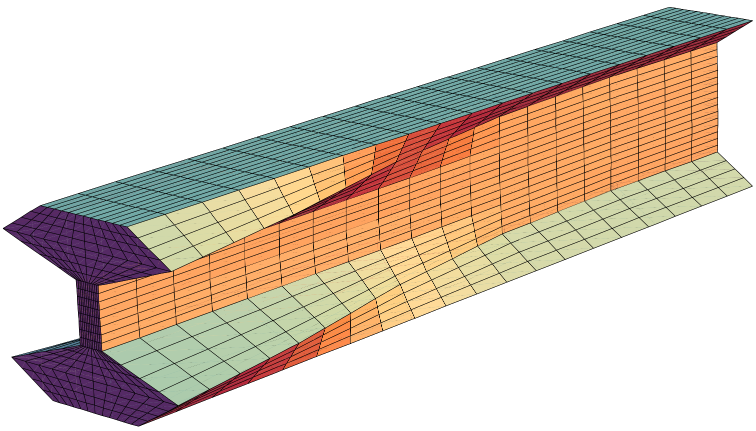
Analiza je potrebovala 87 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 5 ur in 15 minut.

Preglednica 7: Vrednost posameznih parametrov debeline $t_i$ – *varianta 2.3*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 70,00 | 70,00 | 51,97 | 35,32 | 12,03 | 9,77 |
| $h = 5/6\ h_0$ | 61,67 | 54,94 | 27,76 | 14,11 | 5,76 | 4,94 |
| $h = 4/6\ h_0$ | 11,14 | 3,52 | 7,31 | 5,48 | 6,22 | 6,32 |
| $h = 3/6\ h_0$ | 5,43 | 13,90 | 7,20 | 4,52 | 5,23 | 7,46 |
| $h = 2/6\ h_0$ | 11,14 | 3,52 | 7,31 | 5,48 | 6,22 | 6,32 |
| $h = 1/6\ h_0$ | 61,67 | 54,94 | 27,76 | 14,11 | 5,76 | 4,94 |
| $h = 0\ h_0$ | 70,00 | 70,00 | 51,97 | 35,32 | 12,03 | 9,77 |

### 4.2.3.4 Reduciran set parametrov A

Z omejitvijo širine zgornjega in spodnjega pasu smo šli še dlje. Celoten zgornji in spodnji rob smo fiksirali pri vrednosti $t_i = t_0$. V našem primeru to pomeni, da so parametri $t_1$ do $t_6$ ter $t_{37}$ do $t_{42}$ tekom celotnega postopka optimizacije konstantni in zasedajo vrednost $t_0$. Slika 47 prikazuje dobljeni rezultat. Da bi bili rezultati smiselni, smo konstrukciji delno omejili vse dimenzije prostora in zahtevali, da je končni volumen polovico začetnega ($V = 0{,}5 \cdot V_0$). Lepo je vidno, da omejitev debeline zgornjega in spodnjega pasu ne ustreza predpisani obremenitvi, zato algoritem širino pasnice proti središču elementa poveča toliko, da zadovolji zahtevam obremenitve.



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- parametri debeline $t_i$ v zgornjem in spodnjem pasu so konstanti
- $V = 0{,}5 \cdot V_0$

Slika 47: Končna geometrija izbranega konstrukcijskega elementa – *varianta 2.4*

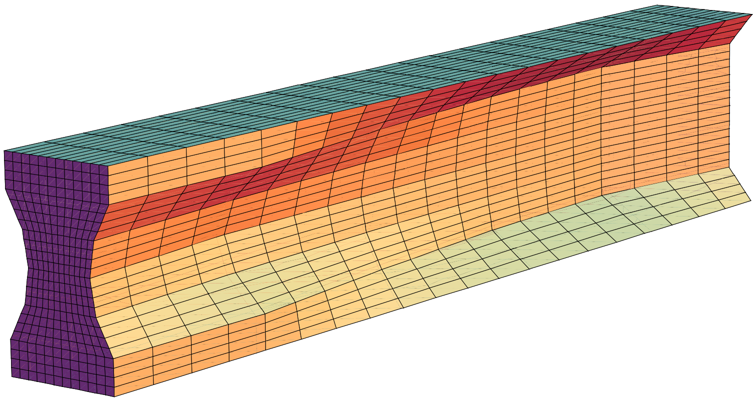
Analiza je potrebovala 287 iteracijskih korakov in je porabila na srednje zmogljivem računalniku 14 ur in 50 minut.

Preglednica 8: Vrednost posameznih parametrov debeline $t_i$ – *varianta 2.4*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |
| $h = 5/6\ h_0$ | 116,40 | 84,12 | 40,84 | 12,81 | 6,36 | 9,20 |
| $h = 4/6\ h_0$ | 15,22 | 15,49 | 15,14 | 11,38 | 7,38 | 9,96 |
| $h = 3/6\ h_0$ | 14,99 | 13,99 | 13,56 | 10,83 | 7,28 | 10,00 |
| $h = 2/6\ h_0$ | 15,22 | 15,49 | 15,14 | 11,38 | 7,38 | 9,96 |
| $h = 1/6\ h_0$ | 116,40 | 84,12 | 40,84 | 12,81 | 6,36 | 9,20 |
| $h = 0\ h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |

#### 4.2.3.5 Reduciran set parametrov B

Iz vseh dosedanjih variant je lepo vidno, da je optimalna oblika konstrukcije oblika I profila. To pomeni veliko širino zgornjega in spodnjega pasu in majhno debelino na sredini. Da bi se izognili povečanju debeline v nižjih plasteh dodamo omejitev $t_i \le t_{\max}$, s čimer dosežemo da postopek optimizacije izbira obliko znotraj začetne konstrukcije. Pri tem zopet upoštevamo omejitev $V = 0{,}5 \cdot V_0$. Slika 48 prikazuje končno obliko geometrije. Prečni prerez blizu vpetja prenaša velike obremenitve, zato je oblika I profila manj očitna; bolj nazorno pa se pokaže proti koncu konzole.

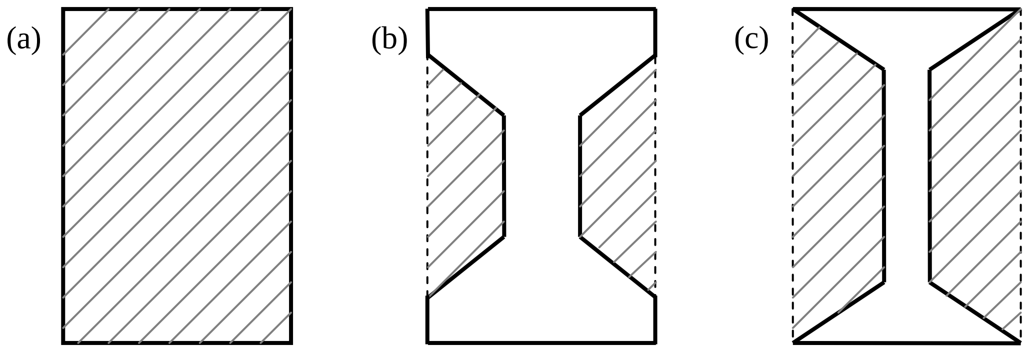


*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- parametri debeline $t_i$ v zgornjem in spodnjem pasu so konstanti
- $V = 0{,}5 \cdot V_0$
- $t_i \leq t_{\max}$

Slika 48: Končna geometrija izbranega konstrukcijskega elementa – *varianta 2.5*

Analiza je potrebovala 69 iteracijskih korakov in je porabila na srednje zmogljivem računalniku 3 ure in 31 minut.




Slika 49: Prikaz začetnega *(a)* in končnega stanja oblike prereza pri vpetju *(b)* ter na koncu elementa *(c)*

Preglednica 9: Vrednost posameznih parametrov debeline $t_i$ – *varianta 2.5*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |
| $h = 5/6\ h_0$ | 60,00 | 60,00 | 34,96 | 15,44 | 7,07 | 32,38 |
| $h = 4/6\ h_0$ | 41,51 | 26,30 | 17,11 | 10,17 | 6,38 | 32,37 |
| $h = 3/6\ h_0$ | 35,69 | 14,35 | 11,05 | 8,37 | 6,12 | 32,46 |
| $h = 2/6\ h_0$ | 41,51 | 26,30 | 17,11 | 10,17 | 6,38 | 32,37 |
| $h = 1/6\ h_0$ | 60,00 | 60,00 | 34,96 | 15,44 | 7,07 | 32,38 |
| $h = 0\ h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |

### 4.2.4 Povzetek

Metoda se je izkazala za matematično manj zahtevno od prejšnje, saj konstrukcijo obremenjujemo le v elastičnem območju. Povprečni čas, potreben za izračun posameznega iteracijskega koraka, je okoli 3 minute, kar je bistveno hitreje.

Primerjava velikosti volumnov med seboj je nesmiselna, saj primeri konvergirajo k različnim končnim volumnom ($V_0$ in $0{,}5 \cdot V_0$). Vsi primeri konvergirajo k obliki I.

## 4.3 Primerjava obeh metod optimizacije topologije

Ob pogledu na obe preglednici (Preglednica 10 in Preglednica 11) je takoj jasno razvidno, da je metoda optimizacije na maksimalno togost pri danem volumnu bistveno hitrejša. Glavni razlog je preračun konstrukcije samo v elastičnem območju, med tem ko metoda optimizacije na mejno obtežbo konstrukcijo preračunava v plastičnem območju, kar nam pa daje bistveno bolj realne rezultate. Če bi želeli konstrukcijo optimizirati za kasnejšo izdelavo, bi se vsekakor odločili za metodo optimizacije na mejno obtežbo, saj se s to metodo približamo dejanski nosilnosti konstrukcije. Ker se ukvarjamo z iskanjem optimalne topologije pa je metoda optimizacije na pomik bolj priročna.

Primerjava dimenzij med seboj bi bila nesmiselna, saj so bili uporabljeni drugačni kriteriji optimizacije ter različne obremenitve konstrukcije. Samo konvergiranje k obliki I profila je močneje pri drugi metodi, saj je večina primerov hitro konvergirala k optimalni obliki.

V želji po optimalni obliki konstrukcije, sestavljene iz pločevin ob upoštevanju optimalne topologije, konstrukcijo parametriziramo na novo in račun poženemo znova. Pri tem upoštevamo, da je konstrukcija oblike I. Uporabimo prvo metodo, saj želimo konstrukcijo obremeniti na mejno obtežbo in dobljene rezultate preveriti po standardu EN 1993-1-1. Podroben opis postopka je opisan v poglavju 5.

Preglednica 10: Primerjava trajanja ene iteracije za metodo mejne nosilnosti

| Varianta | 1.1 | 1.2 | 1.3 | 1.4 |
|---|---|---|---|---|
| Št. iteracij | 69 | 191 | 186 | 61 |
| Trajanje računa | 10 h 38 min | 48 h 4 min | 47 h 19 min | 8 h 55 min |
| Čas ene iteracije | 9 min 15 sec | 15 min 6 sec | 15 min 16sec | 8 min 46sec |

Preglednica 11: Primerjava trajanja ene iteracije za metodo pomikov

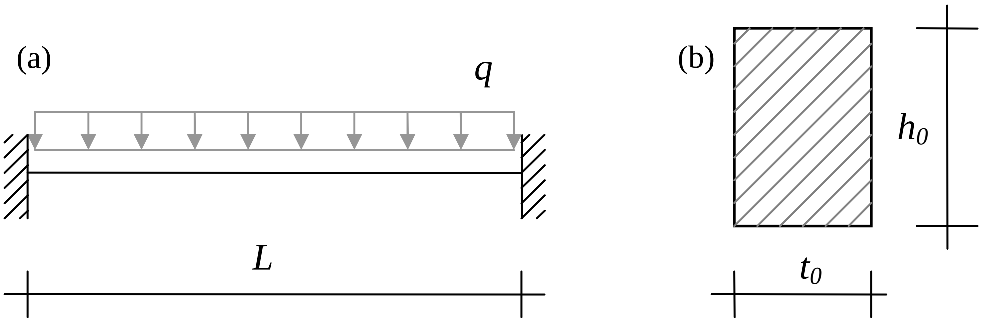
| Varianta | 2.1 | 2.2 | 2.3 | 2.4 | 2.5 |
|---|---|---|---|---|---|
| Št. iteracij | 592 | 97 | 87 | 287 | 69 |
| Trajanje računa | 29 h 54 min | 30 min | 5 h 15 min | 14 h 50 min | 3 h 31 min |
| Trajanje ene iteracije | 3 min 2 min | 20 sec | 3 min 40 sec | 3 min 6 sec | 3 min 4 sec |

V nadaljevanju bomo uporabili metodo optimizacije na pomik še v primeru preklade (obojestransko vpetje konstrukcije) in stebra, katerega bomo tudi tlačno obremenili.

## 4.4 Optimizacije preklade na maksimalno togost pri danem volumnu

Jekleni okvirji so v osnovi sestavljeni iz stebrov in prekladnih nosilcev, zato nas je zanimala tudi optimalna oblika preklade (obojestransko vpetje). V poglavju 4.3 smo ugotovili, da je metoda optimizacije na maksimalno togost pri danem volumnu za določevanje same oblike bolj primerna, saj je hitrejša. Zato smo se odločili, da bomo optimalno obliko iskali s to metodo.

Optimizacijski postopek je podrobneje opisan v poglavju 4.2, tako da so tukaj navedene le osnove metode in podatki, uporabljeni pri izračunu.




Slika 50: Statična zasnova preklanega nosilca (a) in začetna geometrija (b)

Upoštevali smo statično zasnovo obojestransko vpetega nosilca dolžine $L$, obremenjenega z linijsko obtežbo $q$ in začetnega prečnega prereza dimenzij $h_0 \cdot t_0$. Konstrukcijo smo parametrizirali s parametri oblike $\phi_i$, in sicer z višino na začetku $h_1$ in višino na koncu $h_2$ ter z mrežo debelin $t_i$.

### 4.4.1 Namenska funkcija

Do sedaj je bilo vedno potrebno opisati tri funkcije, ki bodo opisale želen problem optimizacije. Ker za optimizacijo uporabljamo metodo optimiziranja na minimalen pomik, so vse funkcije ostale nespremenjene (Poglavje 4.2.1)

$$\Phi_1 = \begin{cases} -\mu_1 \log(\varepsilon load) - \dfrac{\mu_1 \log\left(\dfrac{V(\phi_i)}{V_0} - 1 + \varepsilon load\right)}{\varepsilon load} + \dfrac{\mu_1 \log\left(\dfrac{V(\phi_i)}{V_0} - 1 + \varepsilon load\right)^2}{2\varepsilon load^2}, & \dfrac{V(\phi_i)}{V_0} - 1 \geq 1 - \varepsilon load \\ -\mu_1 \log\left(-\dfrac{V(\phi_i)}{V_0}\right), & \dfrac{V(\phi_i)}{V_0} - 1 < 1 - \varepsilon load \end{cases}$$

$\phi_i$ … parameter oblike

$\mu_1$ … faktor

$\varepsilon load$ … vrednost pri kateri spremenimo funkcijo log v vrsto

$V(\phi_i)$ … volumen trenutno izbrane oblike

$V_0$ … začetni volumen

Slika 51: Končna oblika kazenske funkcije $\Phi_1$

$$\Phi_2 = Pomik(\phi_i)$$

Slika 52: Končna oblika funkcije $\Phi_2$

$$\Phi_3 = \begin{cases} -\mu_2 \log(\varepsilon load) - \dfrac{\mu_2 (\phi_i - t_{\min}(1 + \varepsilon load))}{\varepsilon load \cdot t_{\min}} + \dfrac{\mu_2 (\phi_i - t_{\min}(1 + \varepsilon load))^2}{2 \cdot \varepsilon load^2 \cdot t_{\min}^2}, & \dfrac{\phi_i}{t_{\min}} - 1 \leq \varepsilon load \\ -\mu_2 \log\left(\dfrac{\phi_i}{t_{\min}} - 1\right), & \dfrac{\phi_i}{t_{\min}} - 1 > \varepsilon load \end{cases}$$

$\phi_i$ … parametri oblike

$\mu_2$ … faktor

$\varepsilon load$ … vrednost pri kateri spremenimo funkcijo log v vrsto

$t_{\min}$ … predpisana minimalna vrednost parametrov oblike

Slika 53: Končna oblika kazenske funkcije $\Phi_3$

### 4.4.2 Postopek računa

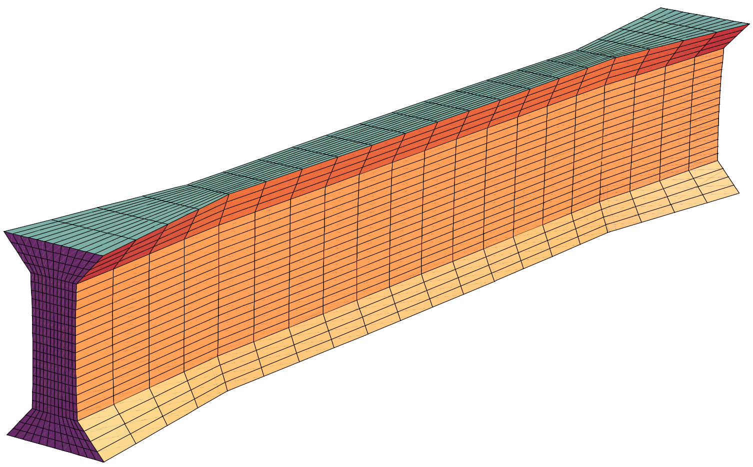
Tako kot do sedaj, smo pri iskanju optimalne oblike konstrukcijskega elementa izvedli več različnih variant in jih med sabo primerjali. Z upoštevanjem različnih projektnih parametrov smo med seboj primerjali vplive parametrov na končni rezultat. Pri spreminjanju parametrov smo se zopet omejili na parametre debeline $t_i$ in parametra višine $h_i$ in tako pri različnih primerih upoštevali različne parametre.

Vsi opisani primeri imajo enako začetno geometrijo. Gre torej za prekladne nosilce konstrukcije dolžine $L_0 = 500cm$, pri čemer je prečni prerez začetne geometrije $t_o \cdot h_0$ konstantnih dimenzij $t_0 = 60cm$ in $h_0 = 100cm$. Za opis mreže debelin $t_i$ smo ponovno uporabili 6 parametrov $\phi_i$ v smeri osi konstrukcije in 7 parametrov $\phi_i$ v smeri višine konstrukcije. Tako smo optimizirali 42 parametrov debeline in 2 parametra višine, torej skupaj 44 parametrov $\phi_i$.

### 4.4.3 Postopek optimizacije

#### 4.4.3.1 Variabilna širina pasnice

Parametra višine $h_1$ in $h_2$ smo fiksirali in jima pripisali konstantno vrednost $h_0$. Slika 45 prikazuje končno obliko konstrukcijskega elementa. Jasno se vidi, da želi optimizacijski algoritem opisati obliko I profila s spremenljivo širino pasnice. Širina zgornjega in spodnjega pasu pada skladno z momentno linijo.



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- vsi parametri debeline $t_i$ so prosti

Slika 54: Končna geometrija izbranega konstrukcijskega elementa – *varianta 3.1*

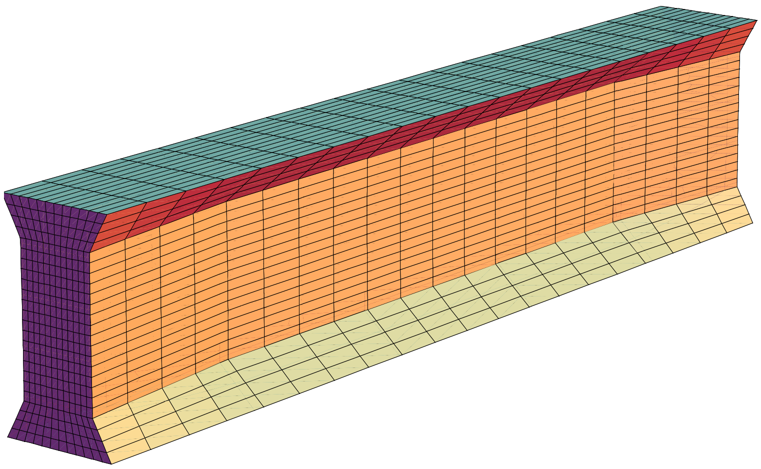
Analiza je potrebovala 134 iteracijskih in je za to porabila na srednje zmogljivem računalniku 7 ur in 57 minut.

Preglednica 12: Vrednost posameznih parametrov debeline $t_i$ – *varianta 3.1*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 52,06 | 22,51 | 25,28 | 25,28 | 22,51 | 52,06 |
| $h = 5/6\ h_0$ | 24,24 | 11,53 | 11,79 | 11,79 | 11,53 | 24,24 |
| $h = 4/6\ h_0$ | 23,04 | 11,92 | 11,31 | 11,31 | 11,92 | 23,04 |
| $h = 3/6\ h_0$ | 22,72 | 12,08 | 11,18 | 11,18 | 12,08 | 22,72 |
| $h = 2/6\ h_0$ | 23,04 | 11,92 | 11,31 | 11,31 | 11,92 | 23,04 |
| $h = 1/6\ h_0$ | 24,24 | 11,53 | 11,79 | 11,79 | 11,53 | 24,24 |
| $h = 0\ h_0$ | 52,06 | 22,51 | 25,28 | 25,28 | 22,51 | 52,06 |

### 4.4.3.2 Konstantne pasnice

Z naslednjo varianto smo šli še dlje. Poleg konstantne višine $h_1$ in $h_2$ smo v algoritem vpeljali tudi konstantne vrednosti parametrov $t_i$ debeline zgornjega in spodnjega pasu. Slika 55 prikazuje končno geometrijo konstrukcije. Zopet je razvidna jasna konvergenca k obliki I, pri tem pa se sama stojina prereza v sredini zmanjša.

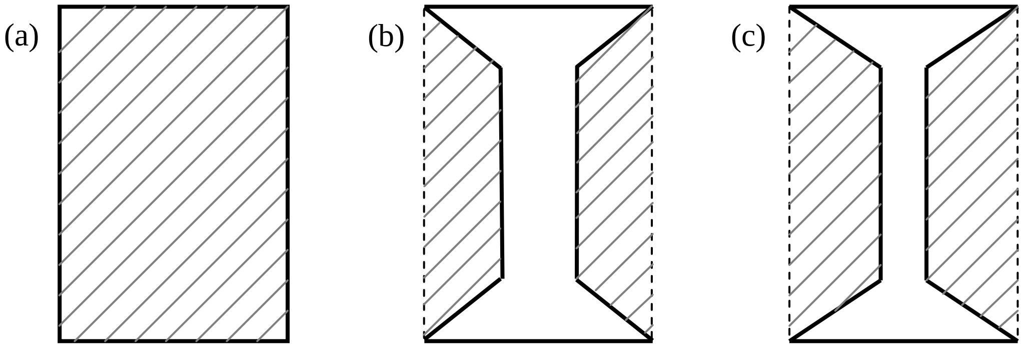


*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- parametri debeline $t_i$ v zgornjem in spodnjem pasu so konstantni
- $V = 0{,}5 \cdot V_0$

Slika 55: Končna geometrija izbranega konstrukcijskega elementa – *varianta 3.2*

Analiza je potrebovala 40 iteracijskih korakov in je na srednje zmogljivem računalniku porabila 1 uro in 35 minut.




Slika 56: Prikaz začetnega *(a)* in končnega stanja oblike prereza pri vpetju *(b)* ter na sredini elementa *(c)*

Preglednica 13: Vrednost posameznih parametrov debeline $t_i$ – *varianta 3.2*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |
| $h = 5/6\ h_0$ | 39,63 | 19,17 | 19,24 | 19,24 | 19,17 | 39,63 |
| $h = 4/6\ h_0$ | 39,33 | 19,68 | 18,85 | 18,85 | 19,68 | 39,33 |
| $h = 3/6\ h_0$ | 39,26 | 19,85 | 18,75 | 18,75 | 19,85 | 39,26 |
| $h = 2/6\ h_0$ | 39,33 | 19,68 | 18,85 | 18,85 | 19,68 | 39,33 |
| $h = 1/6\ h_0$ | 39,63 | 19,17 | 19,24 | 19,24 | 19,17 | 39,63 |
| $h = 0\ h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |

### 4.4.4 Povzetek

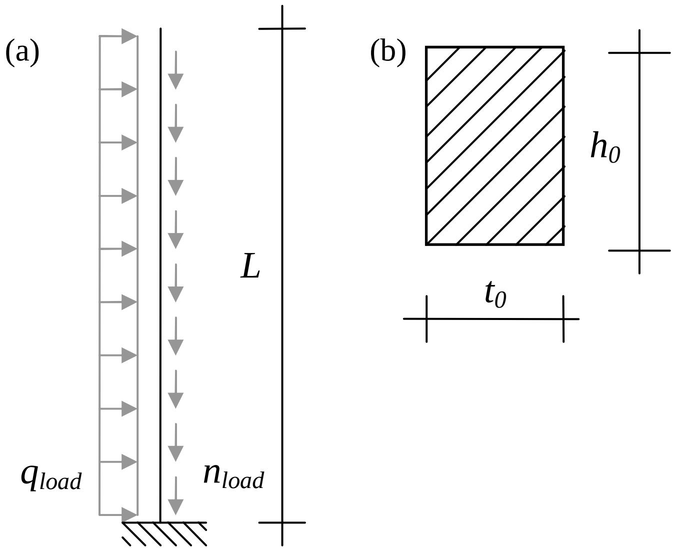
Obe varianti jasno nakazujeta optimalno obliko v obliki prereza I, kar je bilo tudi pričakovano. Manjša potreba po materialu v območju ničelnih momentov je razvidno z zmanjšanjem velikosti parametrov. Povprečna poraba časa za eno iteracijo je okoli 3 minute (Preglednica 14).

Preglednica 14: Primerjava trajanja ene iteracije za metodo pomikov

| | | |
|---|---|---|
| Primer | 3.1 | 3.2 |
| Št. iteracij | 134 | 40 |
| Trajanje računa | 7 h 57 min | 1 h 35 min |
| Trajanje ene iteracije | 3 min 36 sec | 2 min 23 sec |

## 4.5 Optimizacije stebra na maksimalno togost pri danem volumnu

Kot je bilo že omenjeno, sta osnovna elementa jeklenih okvirjev prekladni nosilec in steber. V poglavju 4.4 smo že podrobneje analizirali optimizacijo prekladnih nosilcev, zato je v tem poglavju podrobneje opisana optimizacija oblike prečnega prereza stebra. Osnova za optimizacijo je opisana v poglavju 4.2, in sicer optimizacija na maksimalno togost pri danem volumnu. Razlog za izbiro te metode je zopet v hitrosti samega izračuna. Podroben postopek navedene metode optimizacije je opisan v poglavju 4.2, tu pa so navedene le osnove in podatki, uporabljeni pri izračunu.




Slika 57: Statična zasnova stebra (a) in začetna geometrija (b)

Pri računu optimizacije smo torej upoštevali enostransko vpet steber dolžine $L$ (Slika 57), ki je upogibno obremenjen z linijsko obtežbo $q_{load}$ in tlačno z linijsko obtežbo $n_{load}$. Začetni prečni prerez je dimenzij $h_0 \cdot t_0$. Konstrukcijo smo parametrizirali s parametri oblike $\phi_i$, in sicer z višino na začetku $h_1$ in višino na koncu $h_2$ ter z mrežo debelin $t_i$.

### 4.5.1 Namenska funkcija

Zopet je bilo za opis problema potrebno opisati tri funkcije, ki bodo opisale želen problem optimizacije. Ker za optimizacijo uporabljamo metodo optimiziranja na minimalen pomik, so vse funkcije ostale nespremenjene (Poglavje 4.2.1).

$$\Phi_1 = \begin{cases} -\mu_1 \log(\varepsilon load) - \dfrac{\mu_1 \log\left(\dfrac{V(\phi_i)}{V_0} - 1 + \varepsilon load\right)}{\varepsilon load} + \dfrac{\mu_1 \log\left(\dfrac{V(\phi_i)}{V_0} - 1 + \varepsilon load\right)^2}{2\varepsilon load^2}, x \geq 1 - \varepsilon load \\ -\mu_1 \log\left(-\dfrac{V(\phi_i)}{V_0}\right), x < 1 - \varepsilon load \end{cases}$$

$\phi_i$ … parameter oblike

$\mu_1$ … faktor

$\varepsilon load$ … vrednost pri kateri spremenimo funkcijo log v vrsto

$V(\phi_i)$ … volumen trenutno izbrane oblike

$V_0$ … začetni volumen

Slika 58: Končna oblika kazenske funkcije $\Phi_1$

$$\Phi_2 = Pomik(\phi_i)$$

Slika 59: Končna oblika funkcije $\Phi_2$

$$\Phi_3 = \begin{cases} -\mu_2 \log(\varepsilon load) - \dfrac{\mu_2(\phi_i - t_{\min}(1+\varepsilon load))}{\varepsilon load \cdot t_{\min}} + \dfrac{\mu_2(\phi_i - t_{\min}(1+\varepsilon load))^2}{2 \cdot \varepsilon load^2 \cdot t_{\min}^2}, \dfrac{\phi_i}{t_{\min}} - 1 \leq \varepsilon load \\ -\mu_2 \log\left(\dfrac{\phi_i}{t_{\min}} - 1\right), \dfrac{\phi_i}{t_{\min}} - 1 > \varepsilon load \end{cases}$$

$\phi_i$ … parametri oblike

$\mu_2$ … faktor

$\varepsilon load$ … vrednost pri kateri spremenimo funkcijo log v vrsto

$t_{\min}$ … predpisana minimalna vrednost parametrov oblike

Slika 60: Končna oblika kazenske funkcije $\Phi_3$

### 4.5.2 Postopek računa

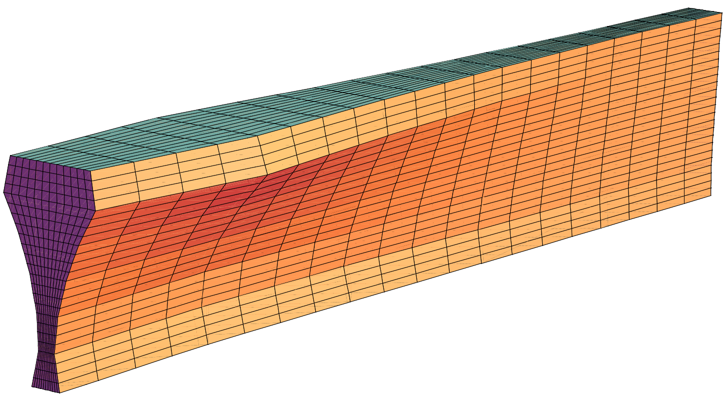
Pri iskanju optimalne oblike prečnega prereza konstrukcije smo zopet upoštevali različno razporeditev parametrov oblike $\phi_i$ ter s tem analizirali vpliv le-teh na končni rezultat. Pri spreminjanju parametrov smo se omejili na parametre debeline $t_i$ in parametra višine $h_i$.

Vsi opisani primeri imajo enako začetno geometrijo, in sicer model stebra, ki je obremenjen tlačno in upogibno. Steber začetne dolžine $L_0 = 500cm$ ima začetni prečni prerez pravokotnik $t_0 \cdot h_0$ z začetnima dimenzijama $t_0 = 60cm$ in $h_0 = 100cm$. Za opis mreže debelin $t_i$ smo uporabili 6 parametrov $\phi_i$ v smeri osi konstrukcije in 7 parametrov $\phi_i$ v smeri višine konstrukcije. Tako smo optimizirali 42 parametrov debeline in 2 parametra višine, torej skupaj 44 parametrov $\phi_i$.

### 4.5.3 Postopek optimizacije

#### 4.5.3.1 Variabilna širina pasnic

Kot v vseh do sedaj izpeljanih analizah smo parametroma višine $h_1$ in $h_2$ pripisali začetno vrednost $h_0$ in ju izločili iz optimizacijskega postopka. Konstantna višina po dolžini konstrukcije pomeni večje spreminjanje parametrov debeline po dolžini (Preglednica 15), saj želi algoritem doseči minimalen volumen. Zopet je jasno videti, da optimizacijski algoritem konvergira k obliki prečnega T prereza. Končna geometrija je prikazana v nadaljevanju.



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- vsi parametri debeline $t_i$ so prosti

Slika 61: Končna geometrija izbranega konstrukcijskega elementa – *varianta 4.1*

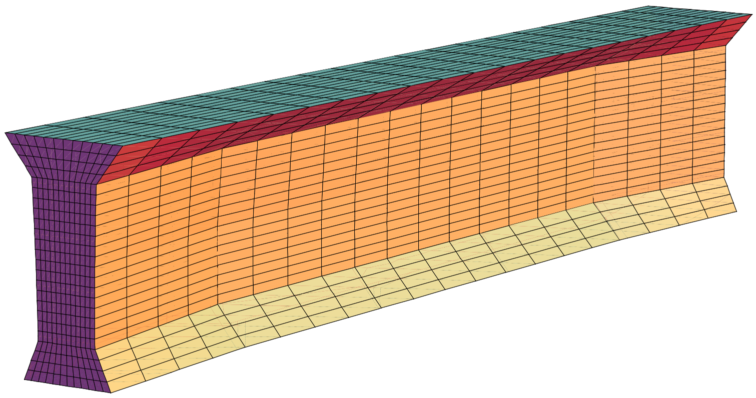
Analiza je potrebovala 42 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 2 uri in 25 minut.

Preglednica 15: Vrednost posameznih parametrov debeline $t_i$ – *varianta 4.1*

| | $L = 0\,L_0$ | $L = 1/5\,L_0$ | $L = 2/5\,L_0$ | $L = 3/5\,L_0$ | $L = 4/5\,L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 41,03 | 52,51 | 37,42 | 26,19 | 20,10 | 19,32 |
| $h = 5/6\,h_0$ | 47,07 | 64,14 | 43,17 | 27,52 | 18,87 | 18,26 |
| $h = 4/6\,h_0$ | 30,66 | 37,96 | 28,72 | 21,72 | 17,69 | 18,42 |
| $h = 3/6\,h_0$ | 18,68 | 16,83 | 16,24 | 15,83 | 15,57 | 17,95 |
| $h = 2/6\,h_0$ | 10,97 | 2,28 | 6,76 | 10,87 | 13,58 | 17,47 |
| $h = 1/6\,h_0$ | 8,17 | 0,01 | 1,35 | 7,87 | 12,80 | 17,62 |
| $h = 0\,h_0$ | 14,51 | 7,90 | 9,83 | 13,37 | 16,39 | 19,08 |

### 4.5.3.2 Konstantna širina zgornje pasnice

Prejšnja varianta jasno nakazuje, da algoritem konvergira k obliki T prereza. Zato smo parametrom debeline $t_i$, ki opisujejo širino zgornjega pasu konstrukcije, predpisali konstantno vrednost $t_0$ in jih tako izločili iz nadaljnjega računa, ostali parametri debeline $t_i$ pa so še vedno del optimizacijskega algoritma. Ravno tako kot zgornji pas smo parametroma višine $h_1$ in $h_2$ predpisali konstantno vrednost $h_0$. Dodatno pa zahtevamo, da je končni volumen $V$ enak polovici začetnega volumna $V_0$ $(V = 0{,}5 \cdot V_0)$. Slika 62 prikazuje končno geometrijo optimalne konstrukcije.



*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- parametri debeline $t_i$ v zgornjem pasu so konstantni
- $V = 0{,}5 \cdot V_0$

Slika 62: Končna geometrija izbranega konstrukcijskega elementa – *varianta 4.2*

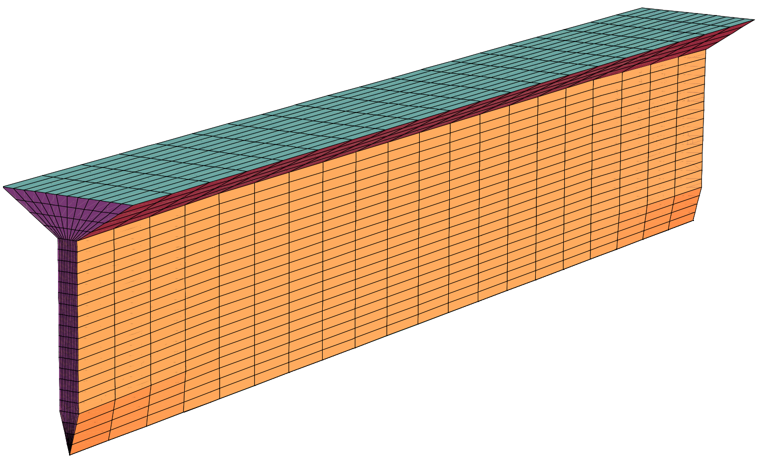
Analiza je potrebovala 67 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 4 ure in 5 minut.

Preglednica 16: Vrednost posameznih parametrov debeline $t_i$ – *varianta 4.2*

| | $L = 0\,L_0$ | $L = 1/5\,L_0$ | $L = 2/5\,L_0$ | $L = 3/5\,L_0$ | $L = 4/5\,L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |
| $h = 5/6\,h_0$ | 32,91 | 6,34 | 4,36 | 2,37 | 1,23 | 30,41 |
| $h = 4/6\,h_0$ | 31,43 | 3,40 | 2,49 | 1,62 | 1,07 | 30,41 |
| $h = 3/6\,h_0$ | 30,35 | 0,60 | 0,58 | 0,66 | 0,70 | 30,36 |
| $h = 2/6\,h_0$ | 29,59 | 0,01 | 0,01 | 0,01 | 0,35 | 30,31 |
| $h = 1/6\,h_0$ | 29,17 | 0,01 | 0,01 | 0,01 | 0,19 | 30,31 |
| $h = 0\,h_0$ | 44,55 | 28,34 | 28,60 | 29,42 | 30,06 | 45,19 |

#### 4.5.3.3 Konstantni širini pasnic različnih dimenzij

Naslednja varianta je podobna prejšnjemu primeru. Parametri debeline $t_i$, ki opisujejo zgornji pas konstrukcije, ostajajo nespremenjeni in imajo predpisano vrednost $t_0$. Dodatno pa smo spremenili spodnji pas konstrukcije. Vsem parametrom debeline, ki opisujejo spodnji pas smo predpisali enako vrednost $t_i$, tako da je širina spodnjega pasu konstantna po dolžini konstrukcije, vendar se njena vrednost spreminja tekom optimizacije. Parametra višine $h_1$ in $h_2$ pa ostajata konstantna in velikosti $h_0$. Rezultat algoritma je zelo presenetljiv. Večina vrednosti je padla na minimalno izbrano vrednost (Preglednica 17).

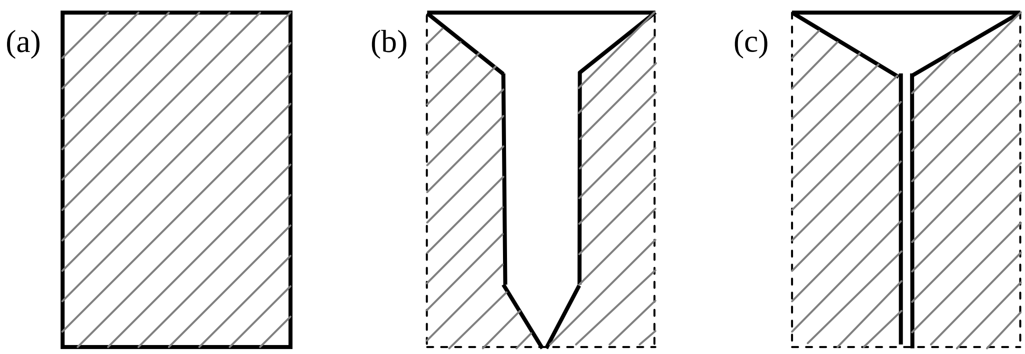


*Osnovni računski parametri:*

- nparamX = 6
- nparamY = 7
- zgostitev 4/12/4
- parametra višine $h_1$ in $h_2$ sta konstantna
- parametri debeline $t_i$ v zgornjem pasu so konstantni
- parametri debeline $t_i$ v spodnjem pasu so vsi enaki
- $V = 0{,}5 \cdot V_0$

Slika 63: Končna geometrija izbranega konstrukcijskega elementa – *varianta 4.3*

Analiza je potrebovala 50 iteracijskih korakov in je za to porabila na srednje zmogljivem računalniku 2 uri in 30 minut.




Slika 64: Prikaz začetnega *(a)* in končnega stanja oblike prereza pri vpetju *(b)* ter na sredini elementa *(c)*

Preglednica 17: Vrednost posameznih parametrov debeline $t_i$ – *varianta 4.3*

| | $L = 0\ L_0$ | $L = 1/5\ L_0$ | $L = 2/5\ L_0$ | $L = 3/5\ L_0$ | $L = 4/5\ L_0$ | $L = L_0$ |
|---|---|---|---|---|---|---|
| $h = h_0$ | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 | 60,00 |
| $h = 5/6\ h_0$ | 9,57 | 0,01 | 0,01 | 0,01 | 0,01 | 8,74 |
| $h = 4/6\ h_0$ | 9,05 | 0,01 | 0,01 | 0,01 | 0,01 | 8,74 |
| $h = 3/6\ h_0$ | 8,66 | 0,01 | 0,01 | 0,01 | 0,01 | 8,72 |
| $h = 2/6\ h_0$ | 8,36 | 0,01 | 0,01 | 0,01 | 0,01 | 8,71 |
| $h = 1/6\ h_0$ | 8,18 | 0,01 | 0,01 | 0,01 | 0,01 | 8,71 |
| $h = 0\ h_0$ | 0,01 | 0,01 | 0,01 | 0,01 | 0,01 | 0,01 |

### 4.5.4 Povzetek

Varianta 4.3 najbolj nazorno prikazuje konvergiranje oblike prečnega prereza konstrukcije k obliki T, kar smo tudi pričakovali. Pri vpetju je poraba materiala večja, kar je v skladu z dano obremenitvijo. Povprečna poraba časa za eno iteracijo je okoli 3 minute (Preglednica 14).

Preglednica 18: Primerjava trajanja ene iteracije za metodo pomikov

| Primer | 4.1 | 4.2 | 4.3 |
|---|---|---|---|
| Št. iteracij | 42 | 67 | 50 |
| Trajanje računa | 2 h 25 min | 4 h 5 min | 2 h 30 min |
| Trajanje ene iteracije | 3 min 27 min | 3 min 40 sec | 3 min |

# 5 OPTIMIZACIJA PARAMETROV NA MEJNO OBTEŽBO

Optimizacija parametrov je nadaljevanje topološke optimizacije. Iz topološke optimizacije lahko razberemo poenostavljeno obliko prečnega prereza h kateri konvergira algoritem (Slika 9). Izbira poenostavljene oblike je ponavadi povezana z možnostjo enostavne izvedbe, ker se izkaže, da izredno kompleksna oblika prečnega prereza v končni fazi ni najbolj optimalna izbira. Ker obliko prečnega prereza poenostavimo, jo lahko opišemo z manjšim številom parametrov $\phi_i$. S tem poskrbimo, da je spreminjanje konstrukcije tekom postopka optimizacije veliko bolj omejeno (oblika se spreminja samo preko parih parametrov). Pri tem se oddaljimo od dejanske optimalne oblike, vendar pridobimo na izvedljivosti same konstrukcije.

## 5.1 EUROCODE EN 1993-1-1

Za potrebe diplomske naloge je bil uporabljen standard EN 1993-1-1. Standard EN 1993-1-1 zajema osnovna pravila za projektiranje jeklenih konstrukcij ter vsa dodatna pravila za projektiranje jeklenih stavb.

### 5.1.1 Kompaktnost prereza

Prečne prereze jeklenih konstrukcij pogosto izdelujemo iz vitkih pločevin, ki so podvržene nevarnosti lokalnega izbočenja. Da bi bila razvrstitev prerezov natančneje opisana, standard EN 1993-1-1 v poglavju 5.5 opisuje njihovo razporeditev po različnih razredih. Standard v ta namen uporablja 4 različne razrede kompaktnosti prereza, kateri nam pri samem dimenzioniranju pomagajo pri izbiri pravilne vrste globalne analize.

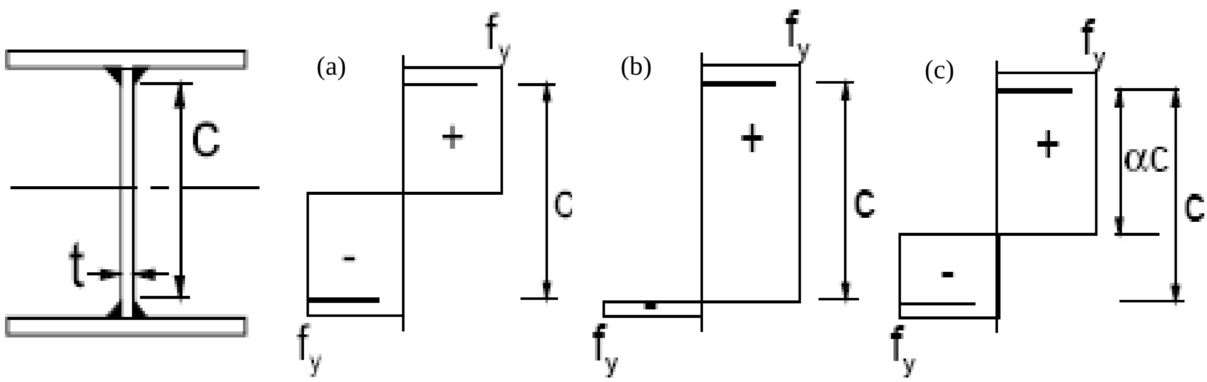
Razvrščanje prečnih prerezov v posamezne razrede kompaktnosti je odvisno od razmerja med širino in debelino tlačno obremenjenih delov prečnega prereza. Tlačno obremenjeni deli prečnega prereza so vsi tisti deli prereza, ki so v obravnavani obtežni kombinaciji v celoti ali delno pod vplivom tlačnih napetosti. Razredi kompaktnosti se med seboj razlikujejo po načinu določanja mejne nosilnosti prereza ter po načinu globalne analize konstrukcije. Pri razvrščanju prerezov v posamezne razrede je potrebno upoštevati, da so različni deli prereza lahko razvrščeni v različne razrede kompaktnosti, prerez kot celoto pa razvrstimo v razred glede na najvitkejši sestavni del prereza.

Pri tej nalogi smo se omejili na plastično analizo, saj je optimizacija, ki upošteva še stabilnost, veliko zahtevnejša in potrebuje tudi analizo občutljivosti na inperfektnost. Da bi zadostili zahtevam standarda 1993-1-1 glede kompaktnosti prečnih prerezov, smo morali zagotoviti 1. razred kompaktnosti. Prečni prerez konstrukcije, ki zadostuje pogoje 1. razreda kompaktnosti, lahko analiziramo po globalni plastični analizi, pri kateri upoštevamo nastanek plastičnih členkov. Prečni prerezi konstrukcije v območju plastičnega členka je podvržen velikim plastičnim deformacijam, zato je možnost nastanka lokalnih izbočitev povečana. Z uvrstitvijo prečnega prereza v 1. razred kompaktnosti zagotovimo, da do lokalnega izbočenja ne pride pred nastankom porušnega mehanizma.

Standard EN 1993-1-1 določa razvrščanje prečnega prereza posebej za stojine ter posebej za pasnice. Vsak del celotnega prečnega prereza je lahko razvrščen v različen razred kompaktnosti, celoten prečni prerez pa uvrstimo v razred glede na najvitkejši del prereza.

#### 5.1.1.1 Razvrstitev stojine

Kot je bilo že omenjeno, se pri razvrščanju prečnega prereza v razrede kompaktnosti določi razred kompaktnosti posebej za stojine.




Slika 65: Prikaz geometrije in poteka napetosti po prerezu stojine za upogib (a), tlak (b) in upogib s tlakom (c)

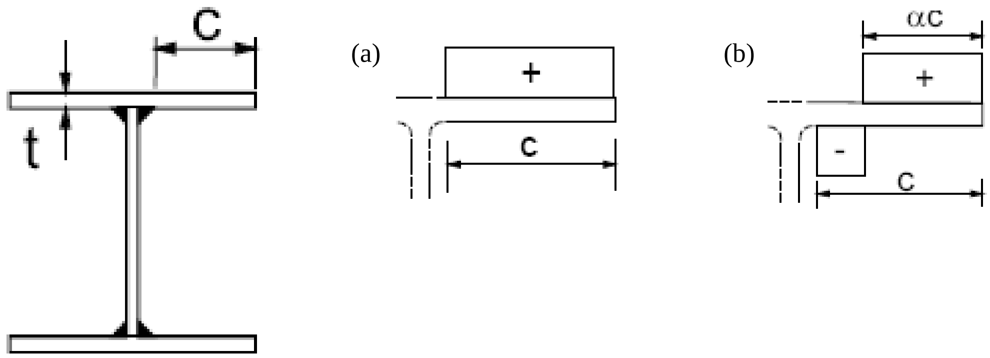
Pri upogibu: $\frac{d}{t_w} \leq 72\varepsilon$

Pri tlaku: $\frac{d}{t_w} \leq 33\varepsilon$

Pri upogibu in tlaku: $\begin{cases} \dfrac{d}{t_w} \leq \dfrac{396\varepsilon}{13\alpha - 1}, \alpha > 0{,}5 \\ \dfrac{d}{t_w} \leq \dfrac{36\varepsilon}{\alpha}, \alpha < 0{,}5 \end{cases}$

### 5.1.1.2 Razvrstitev pasnic

Pri razvrščanju pasnic v posamezne razrede kompaktnosti imamo možnost izbiranja med vroče valjanimi profili in varjenimi profili. Ker smo se v nalogi osredotočili na varjene profile, so vse napisane enačbe samo za varjene profile.




Slika 66: Prikaz geometrije in poteka napetosti po prerezu pasnice za tlak (a) in upogib s tlakom (prosti rob je tlačen) (b)

Tlačno obremenjene pasnice: $\dfrac{c}{t_f} \leq 9\varepsilon$

Tlačno in upogibno obremenjene pasnice (prosti rob je tlačen): $\dfrac{c}{t_f} \leq 9\varepsilon$

### 5.1.1.3 Uporaba pri optimizaciji

Zahteve glede kompaktnosti smo vpeljali že v sam optimizacijski algoritem pri računu I profila. Pri prametrizaciji smo izbrali samo parametra višine stojine in širine pasnice. Debelini stojine in pasnice pa sta odvisni od višine in zahtev po EN 1993-1-1 in se spreminjata s spreminjanjem višine oziroma širine.

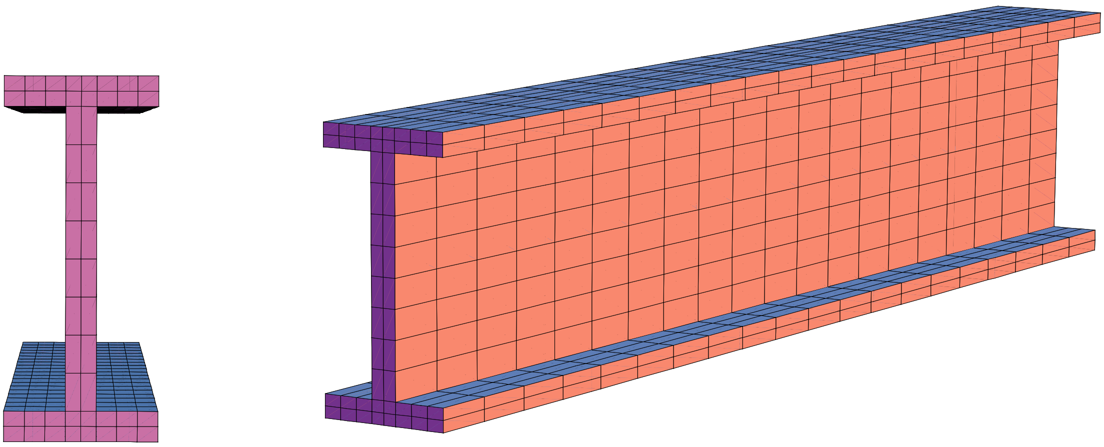
Zahteva za debelino stojine: $t_w = \dfrac{h_w}{72\varepsilon}$

Zahteva za debelino pasnice $t_f = \dfrac{b_f}{18\varepsilon}$

## 5.2 Postopek optimizacije parametrov

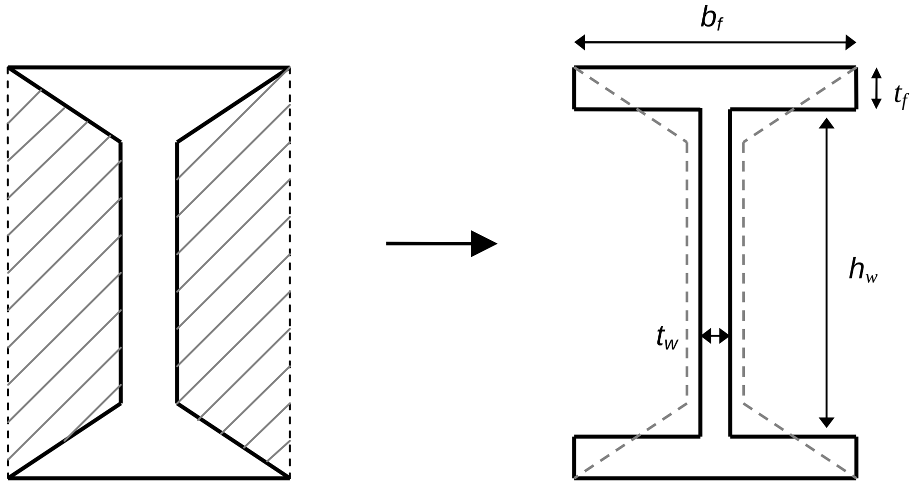
Vsi dosedanji primeri jasno kažejo, da ne glede na izbiro vrste parametrov, optimizacijski algoritem vedno sili konstrukcijo v obliko I profila. Če bi želeli, da bi algoritem izračunal še bolj izrazit I profil, bi bilo potrebno zgostiti mrežo parametrov in seveda mrežo končnih elementov. Velika zgostitev mreže pa pomeni potrebo po močnejši računalniški opremi in kar je najpomembnejše, časi samih izračunov se močno podaljšajo.

Prednost tako opisane mreže je potreba po manjši količini projektnih parametrov $\phi_i$, kar precej skrajša potek analize. Za opis takšne mreže porabimo bistveno manjše število končnih elementov, kar se zopet odraža na času analize. Največja prednost pa se pokaže pri spreminjanju vrednosti parametrov. V vseh dosedanjih primerih je pri spreminjanju parametrov prihajalo do velikih distorzij mreže končnih elementov, tukaj pa se temu izognemo. Pri spremembi velikosti projektnih parametrov pride samo do spremembe velikosti, ne pa tudi do spremembe oblike končnega elementa, kar ima za posledico bolj točne rezultate. Model konstrukcije že v začetni fazi opišemo kot prikazuje Slika 67.



Slika 67: Prikaz 3D modela končnih elementov za obliko I profila

Za opis oblike I profila modela končnih elementov smo uporabili samo dva projektna parametra $\phi_i$, in sicer višino stojine $\phi_1 = h_w$ in širino pasnice $\phi_2 = b_f$ (Slika 68). Debelini stojine in pasnice smo zapisali v odvisnosti od višine oziroma širine. Pri tem smo upoštevali zahteve, podane v poglavju 5.1.1.3 in se s tem izognili nadaljnjim preverjanjem kompaktnosti prereza. Ker upoštevamo samo materialno nelinearnost, je seveda potreben prvi razred kompaktnosti. Tako smo torej debelino stojine zapisali v obliki $t_w = h_w / 72\varepsilon$ in debelino pasnice v obliki $t_f = b_f / 13\varepsilon$. Mrežo končnih elementov smo razdelili na $n_x$ elementov po dolžini, $n_y$ po širini pasnice, $n_z$ po višini stojine in $n_t$ po debelini stojine ter pasnice.




Slika 68: Shematski prikaz optimizacije parametrov

Dimenzije oblike konstrukcije bomo optimizirali na mejno obtežbo z gradientno metodo optimizacije, tako kot smo to počeli v poglavju 4.1.

### 5.2.1 Namenska funkcija

Metoda optimizacije je torej popolnoma enaka metodi, uporabljeni v poglavju 4.1; torej optimiziramo obliko na mejno obtežbo. Izbira funkcij $\Phi_i$ ostaja enaka, torej ostaja tudi oblika posamezne funkcije nespremenjena.

$$\Phi_1 = \begin{cases} -\mu_1 \log(\varepsilon load) - \dfrac{\mu_1 \log(load(\phi) - pload - \varepsilon load)}{\varepsilon load} + \dfrac{\mu_1 \log(load(\phi) - pload - \varepsilon load)^2}{2\varepsilon load^2}, \\ (load(\phi) - pload - \varepsilon load) \le \varepsilon load \\ \\ -\mu_1 \log(load(\phi) - pload), (load(\phi) - pload - \varepsilon load) > \varepsilon load \end{cases}$$

| | |
|---|---|
| $\phi$ | … parameter oblike |
| $\mu_1$ | … faktor |
| $\varepsilon load$ | … vrednost pri kateri spremenimo funkcijo log v vrsto |
| $load(\phi)$ | … obtežba, katero je trenutna konstrukcija sposobna prenesti |
| $pload$ | … predpisana obtežba |

Slika 69: Končna oblika kazenske funkcije $\Phi_1$

$$\Phi_2 = Volumen(\phi_i)$$

Slika 70: Končna oblika funkcije $\Phi_2$

$$\Phi_3 = \begin{cases} -\mu_2 \log(\varepsilon load) - \dfrac{\mu_2(\phi_i - t_{\min}(1+\varepsilon load))}{\varepsilon load \cdot t_{\min}} + \dfrac{\mu_2(\phi_i - t_{\min}(1+\varepsilon load))^2}{2\cdot \varepsilon load^2 \cdot t_{\min}^2}, & \dfrac{\phi_i}{t_{\min}} - 1 \le \varepsilon load \\ -\mu_2 \log\left(\dfrac{\phi_i}{t_{\min}} - 1\right), & \dfrac{\phi_i}{t_{\min}} - 1 > \varepsilon load \end{cases}$$

$\phi_i$ … parametri oblike
$\mu_2$ … faktor
$\varepsilon load$ … vrednost pri kateri spremenimo funkcijo log v vrsto
$t_{\min}$ … predpisana minimalna vrednost parametrov oblike

Slika 71: Končna oblika kazenske funkcije $\Phi_3$

## 5.3 Postopek računa

Računali smo konzolno vpeto konstrukcijo, ki je bila obremenjena z linijsko obtežbo $qload$. Linijsko obtežbo smo na konstrukcijo nanesli s posebnim končnim elementom, kot je bilo opisano v poglavju 3.7. Konstrukcijo smo tako obremenjevali do mejne nosilnosti in iskali optimalne vrednosti parametrov $h_w$ in $b_f$. Izvedli smo dva računa za različno gostoto mreže in medsebojno primerjali rezultate.

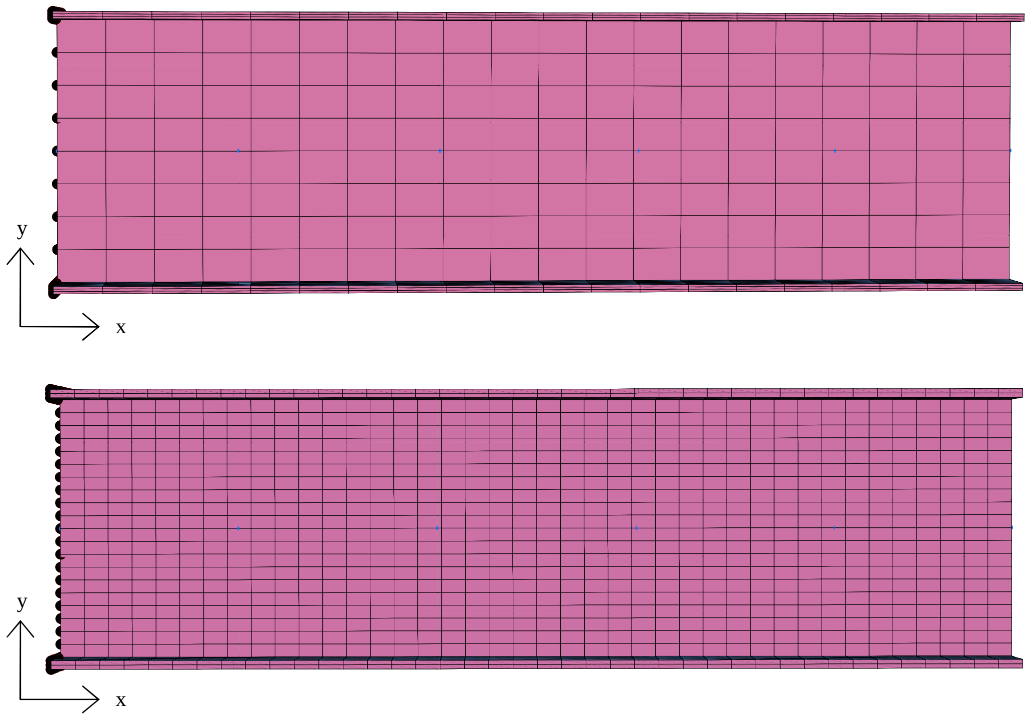
*Vhodni podatki računa:*

- $qload = 10kN/cm = 1000kN/m$
- $L_0 = 500cm$
- $h_{w0} = 100cm$
- $b_{f0} = 50cm$
- $\sigma_y = 23{,}5kN/cm^2$
- $E = 21000kN/cm^2$
- $\nu = 0{,}3$
- *Redka mreža:* 20 elementov po dolžini in 8 po višini stojine
- *Gosta mreža:* 40 elementov po dolžini in 20 po višini stojine

Račun smo izvajali, kot vse do sedaj, v okolju `Mathematica`[10], in sicer s podprogramom `AceFEM`[11]. Algoritem celotnega optimizacijskega algoritma je napisan v poglavju Priloga C: Algoritem optimizacije parametrov.

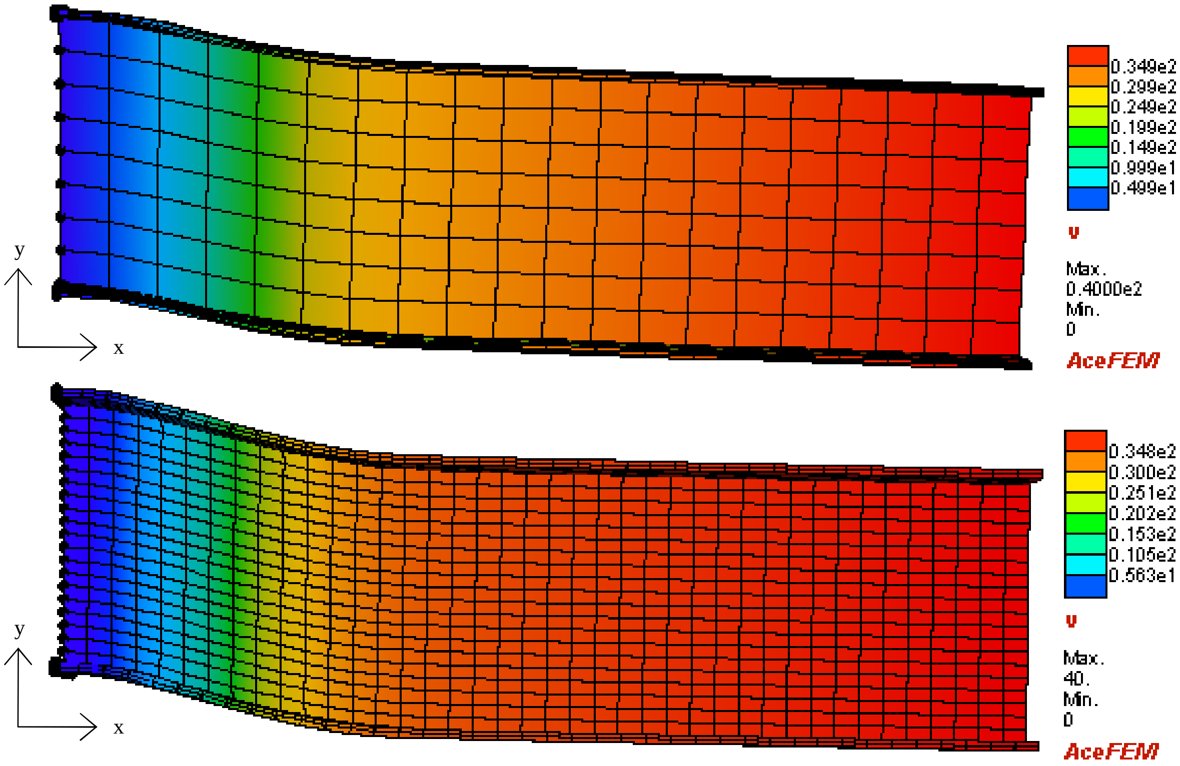
### 5.3.1 Rezultati

Analiza z uporabo redke mreže se je zaključila po 50 iteracijah in je za to porabila na srednje zmogljivem računalniku 10 ur in 7 minut, medtem ko se je analiza z uporabo goste mreže zaključila po 30 iteracijah in je za to porabila na srednje zmogljivem računalniku 9 ur in 39 minut. Vsi dobljeni rezultati so predstavljeni v nadaljevanju (Slika 72 do Slika 76).

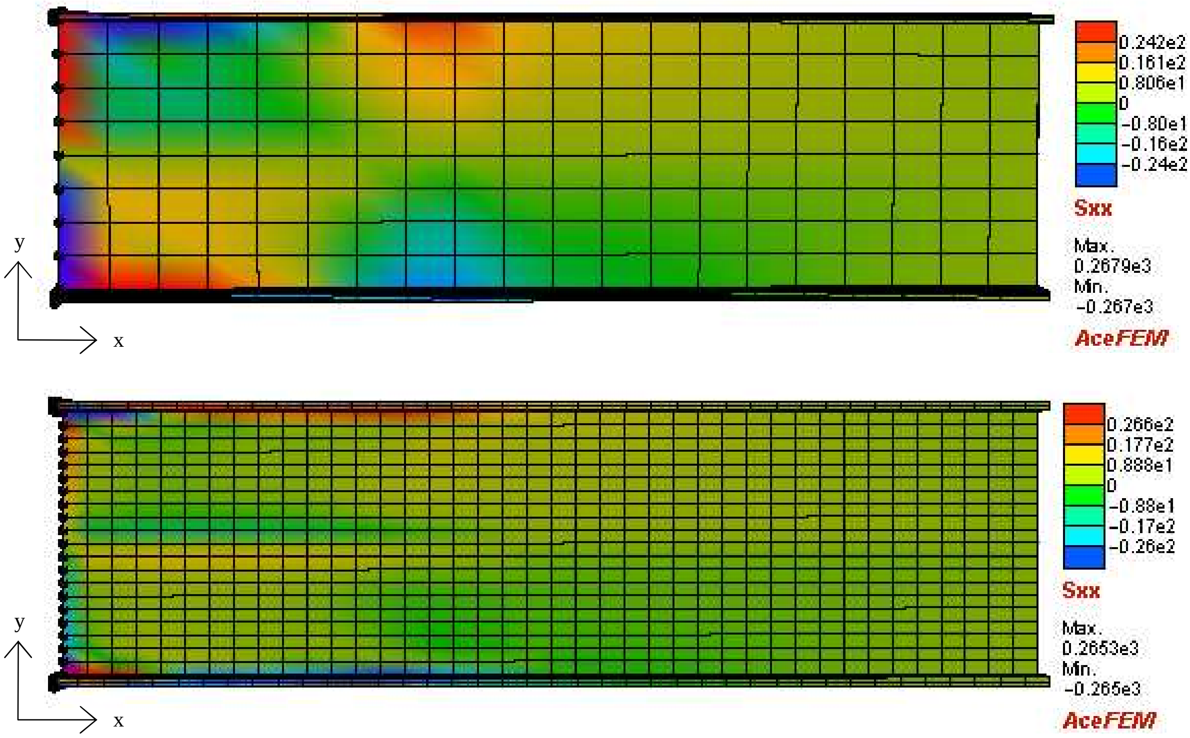



Slika 72: Končna geometrija: redka mreža *(a)* in gosta mreža *(b)*

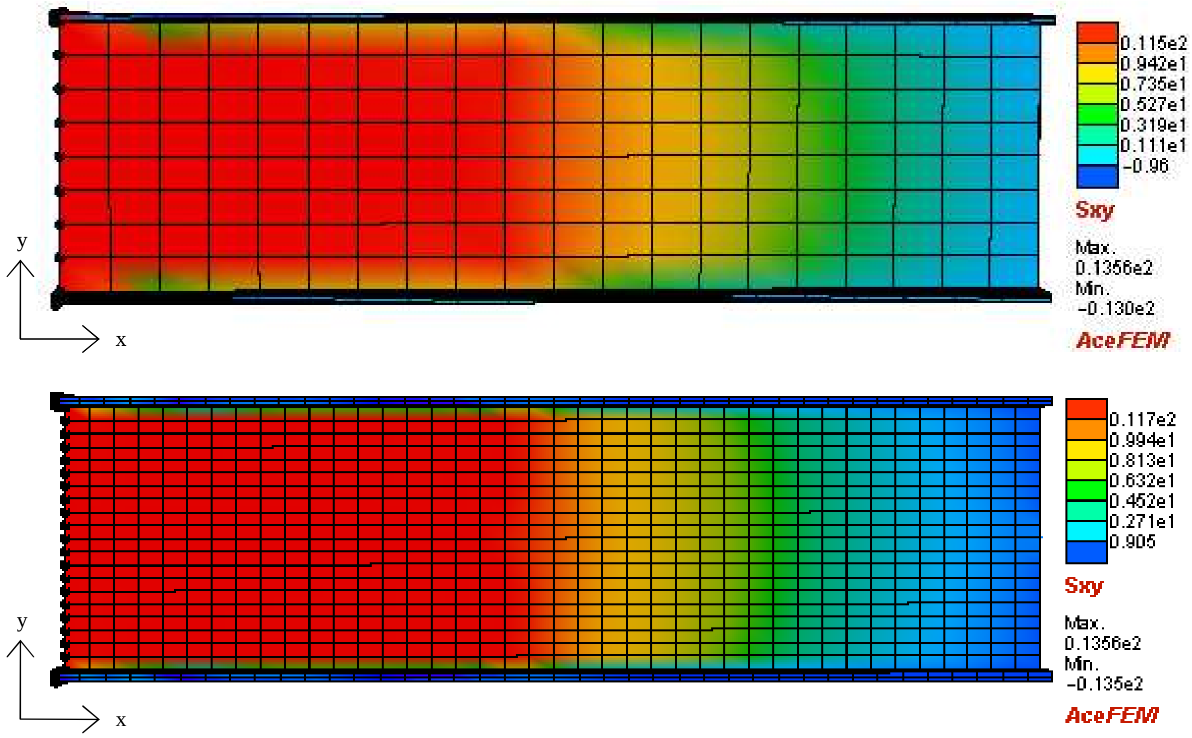
[10] http://www.wolfram.com/
[11] http://www.fgg.uni-lj.si/Symech/



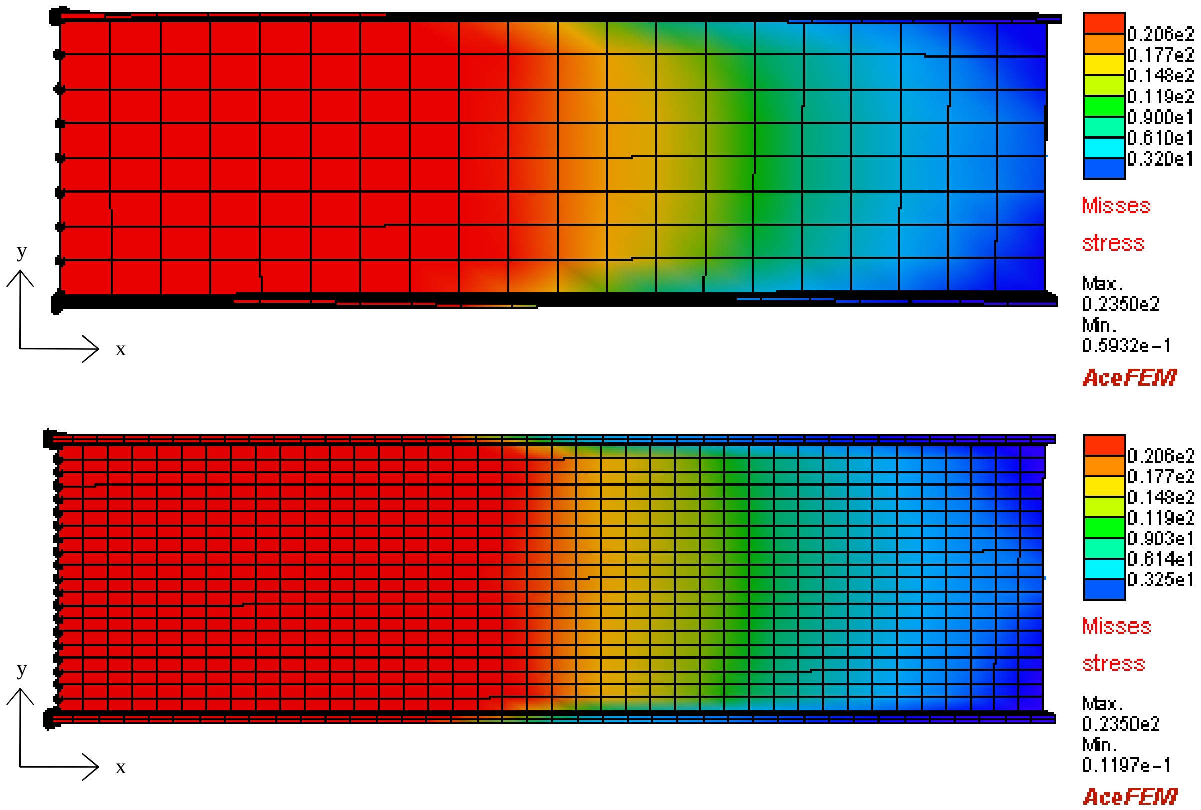

Slika 73: Pomik konstrukcije *v (v smeri y)*: redka mreža *(a)* in gosta mreža *(b)*



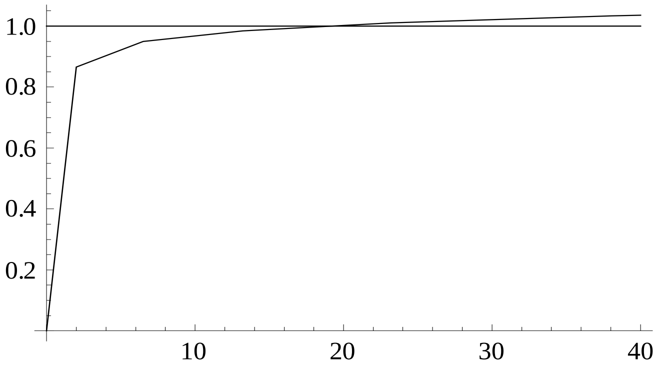

Slika 74: Napetosti $\sigma_{xx}$: redka mreža *(a)* in gosta mreža *(b)*




Slika 75: Napetosti $\sigma_{xy}$: redka mreža *(a)* in gosta mreža *(b)*




Slika 76: Misses-ove napetosti: redka mreža *(a)* in gosta mreža *(b)*




Slika 77: Potek dejanske obtežbe za končno obliko konstrukcije

*Končne dimenzije:*

*Redka mreža:*

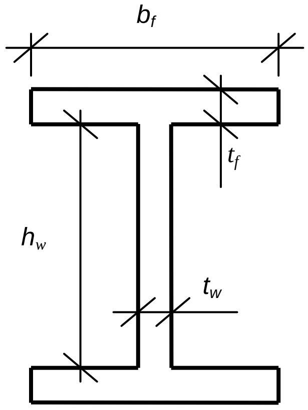
- $h_w = 136{,}54 cm$
- $t_w = 1{,}90 cm$
- $b_f = 67{,}49 cm$
- $t_f = 3{,}75 cm$
- $A = 765 cm^2$

*Gosta Mreža:*

- $h_w = 134{,}42 cm$
- $t_w = 1{,}87 cm$
- $b_f = 78{,}59 cm$
- $t_f = 4{,}37 cm$
- $A = 937 cm^2$

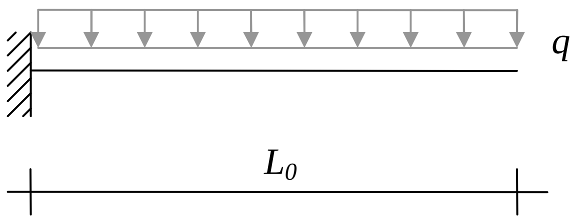
## 5.4 Račun po EC3

Konstrukcije, dobljene z optimizacijskim algoritmom, lahko enostavno preverimo ali zadovoljujejo zahtevam standarda EN 1993-1-1. Pri metodi optimizacije na mejno obtežbo,algoritem konstrukcijo obremenjuje do porušitve. To pomeni, da se konstrukcija obremenjuje v plastičnem območju. Pri računu s polno nosilnostjo prereza standard zahteva 1. razred kompaktnosti (poglavje 5.1.1). Pri računu zahtev standarda EN 1993-1-1 smo obravnavali osnovne zahteve glede nosilnosti prečnega prereza, pri tem pa smo upoštevali plastično analizo.




Slika 78: Geometrija prečnega prereza I profila

### 5.4.1 Obremenitev konstrukcije




Slika 79: Statična zasnova

*Strižna obremenitev konstrukcije (pri vpetju):*

- $V_{Sd} = q \cdot L = 10 \cdot 500$
  $V_{Sd} = 5000 kN$

*Upogibna obremenitev konstrukcije (pri vpetju):*

- $M_{Sd} = \frac{q \cdot L_0^{\,2}}{2} = \frac{10 \cdot 500^2}{2}$
  $M_{Sd} = 1{,}25 \cdot 10^6 \, kNcm$

### 5.4.2 Prečni prerez v strigu

*Površina prečnega prereza:*

*Redka mreža:*

- $A_v = h_w \cdot t_w + 2 \cdot b_f \cdot t_f = 136{,}54 \cdot 1{,}90 + 2 \cdot 67{,}19 \cdot 3{,}75$
  $A_v = 765 cm^2$

*Gosta mreža:*

- $A_v = h_w \cdot t_w + 2 \cdot b_f \cdot t_f = 134{,}42 \cdot 1{,}87 + 2 \cdot 78{,}59 \cdot 4{,}37$
  $A_v = 937 cm^2$

*Nosilnost prečnega prereza na strig:*

*Redka mreža:*

- $V_{pl,Rd} = A_v \dfrac{f_y}{\sqrt{3}} \dfrac{1}{\gamma_{M0}} = 765 \cdot \dfrac{23{,}5}{\sqrt{3}} \cdot \dfrac{1}{1{,}0}$

  $V_{pl,Rd} = 10379 kN$
- $V_{pl,Rd} \geq V_{Sd}$

*Gosta mreža:*

- $V_{pl,Rd} = A_v \dfrac{f_y}{\sqrt{3}} \dfrac{1}{\gamma_{M0}} = 937 \cdot \dfrac{23{,}5}{\sqrt{3}} \cdot \dfrac{1}{1{,}0}$

  $V_{pl,Rd} = 12715 kN$

- $V_{pl,Rd} \geq V_{Sd}$

### 5.4.3 Prečni prerez v enoosnem upogibu

*Plastični odpornostni moment prečnega prereza:*

*Redka mreža:*

- $W_{pl} = b_f \cdot t_f \left(h_w + t_f\right) + \dfrac{t_w \cdot h_w^2}{4} = 67{,}19 \cdot 3{,}75\left(136{,}54 + 3{,}75\right) + \dfrac{1{,}90 \cdot 136{,}54^2}{4}$

  $W_{pl} = 44337 cm^3$

*Gosta mreža:*

- $W_{pl} = b_f \cdot t_f \left(h_w + t_f\right) + \dfrac{t_w \cdot h_w^2}{4} = 78{,}59 \cdot 4{,}37\left(133{,}42 + 4{,}37\right) + \dfrac{1{,}87 \cdot 134{,}42^2}{4}$

  $W_{pl} = 56051 cm^3$

*Nosilnost prečnega prereza na upogib:*

*Redka mreža:*

- $M_{c,Rd} = M_{pl,Rd} = \dfrac{W_{pl} f_y}{\gamma_{M0}} = \dfrac{44337 \cdot 23{,}5}{1{,}0}$

  $M_{pl,Rd} = 1{,}04 \cdot 10^6 kNcm$
- $M_{pl,Rd} < M_{Sd}$

*Gosta mreža:*

- $M_{c,Rd} = M_{pl,Rd} = \dfrac{W_{pl} f_y}{\gamma_{M0}} = \dfrac{56051 \cdot 23{,}5}{1{,}0}$

  $M_{pl,Rd} = 1{,}31 \cdot 10^6 kNcm$

- $M_{Sd} \geq M_{pl,Rd}$

### 5.4.4 Upogib in strig

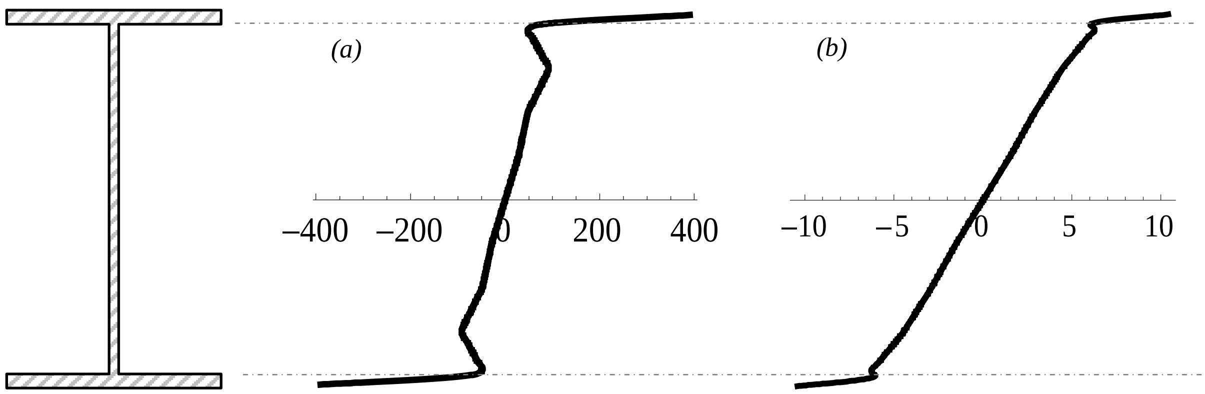
Vpliv prečnih sil na projektno upogibno nosilnost prereza ni potrebno upoštevati, saj je $V_{Sd} \leq 0{,}5 \cdot V_{pl,Rd}$.

### 5.4.5 Ugotovitev

Kontrola prečnega prereza na enoosni upogib ne zadovolji zahtevam standarda EN 1993-1-1 v primeru analize z uporabo redke mreže. Pri analizi z uporabo goste mreže se račun zaključi na varni strani glede na zahteve standarda.

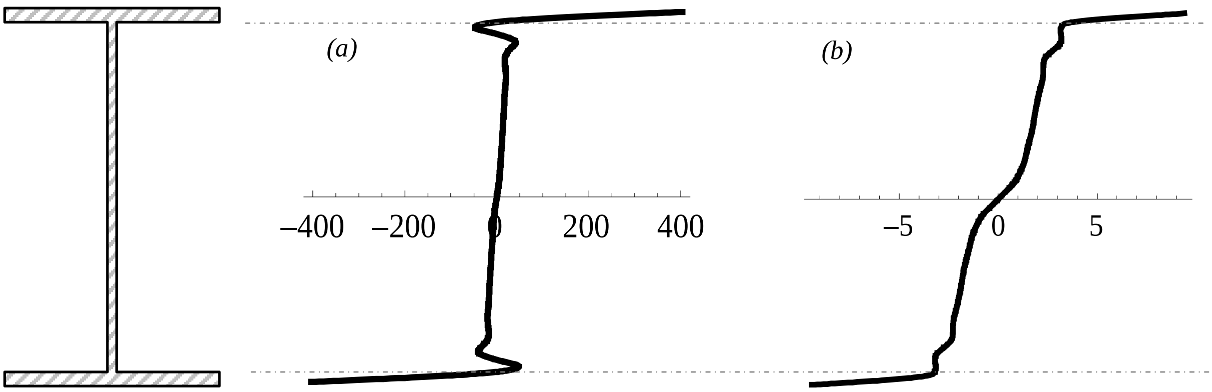
Obremenitev konstrukcije pri vpetju je večja od dejanske nosilnosti prereza, računane po standardu ( $M_{pl,Rd} \leq M_{Sd}$ ). To dokazuje, da so postopki računanja ki jih zahteva standard, na varni strani, saj konstrukcija prenese bistveno več kot to predvidi standard. Razlog za tako obnašanje je v samem računskem postopku (direktna analiza, poglavje 3.6), ki smo ga izvedli tekom optimizacije.

V računu direktne analize smo izbrano konstrukcijo obravnavali kot prostorski problem, zato se pojavi prostorsko napetostno stanje. Potek napetosti pri stičišču pasnice in stojine ter vpetosti na začetku, ob predpostavki idealno elasto-plastičnega materiala, daje možnost povečanje napetosti $\sigma_{xx}$ čez mejo tečenja $\sigma_y$. S tem je konstrukcija sposobna prenesti večje upogibne napetosti kot jih predvidi teorija nosilcev. Slika 80 prikazuje potek napetosti $\sigma_{xx}$ po prerezu konstrukcije pri uporabi redke mreže.




Slika 80: Potek napetosti $\sigma_{xx}$ po višini prereza pri uporabi redke mreže: pri vpetju *(a)* in na sredini nosilca *(b)*

Slika 81 prikazuje potek napetosti $\sigma_{xx}$ po prerezu konstrukcije pri uporabi redke mreže. V primerjavi z uporabo redke mreže, se potek linije bolj približa dejanskemu poteku napetosti (zlasti v prerezu blizu vpetja *(b)*). Razlog za odstopanje dejanske nosilnosti od nosilnosti, računane po postopku, ki ga zahteva standard, pa je tudi v neupoštevanju prečnih sil na upogibno nosilnost. Standard zahteva upoštevanje prečne sile šele kadar je $V_{Sd} \geq 0{,}5 \cdot V_{pl,Rd}$, kar pa v dejanskem stanju ne drži.




Slika 81: Potek napetosti $\sigma_{xx}$ po višini prereza pri uporabi goste mreže: pri vpetju *(a)* in na sredini nosilca *(b)*

# 6 ZAKLJUČEK

Postopki optimiziranja konstrukcij, ki so na voljo kot del komercialnih računalniških programov, ne zadoščajo več zahtevam uporabnikov. Zato smo v nalogi želeli preveriti sodobne pristope k optimiziranju konstrukcij. Z uporabo naprednih računalniških programov lahko optimizacijske postopke zapišemo v bolj kompleksni obliki in s tem pridemo bližje optimalni obliki konstrukcije.

Zaradi zahtevnosti postopkov smo celoten postopek optimizacije razdelili na dva dela. V prvem delu konstrukcijo topološko optimiziramo. Konstrukcijo opišemo s poljubnimi parametri, katere z optimizacijskim algoritmom spreminjamo. Tukaj smo med seboj primerjali dve metodi. Pri metodi optimizacije na mejno obtežbo smo konstrukcijo obremenjevali do porušitve in tako iskali optimalno obliko, pri metodi optimizacije na maksimalno togost pri danem volumnu pa smo konstrukcijo spreminjali tako, da je bil pomik izbrane točke minimalen.

Končna oblika prečnega prereza, ki nam ga daje optimizacija topologije je le grob oris optimalne konstrukcije. Zaradi želje po dejanski izdelavi konstrukcije je končna oblika optimizacije topologije neuporabna, zato konstrukcijo v drugem delu ponovno optimiziramo, in sicer z uporabo optimizacije parametrov. Obliko poenostavimo in prilagodimo dejanskim možnostim izdelave, pri tem pa uporabimo manjše število parametrov. Optimizacijo izvajamo na mejno obtežbo.

Ugotovili smo, da je metoda optimizacije na maksimalno togost pri danem volumnu boljša za uporabo pri optimizaciji topologije. Oblika prečnega prereza tako hitreje konvergira h končni obliki, pri tem pa postopek potrebuje bistveno manj časa. Nasprotno pa moramo pri optimizaciji parametrov nujno uporabiti metodo na mejno obtežbo, saj konstrukcijo optimiziramo za dano obremenitev po Eurocode.

# VIRI

Beg, D. 1999. Projektiranje jeklenih konstrukcij po evropskem standardu ENV 1993-1-1, Ljubljana,Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 219 str.

Choi, K. K., Chang K.-H. 1994. A study of design velocity field computation for shape optimal design, Finite Elements in Analysis and Design 14: 317-341

EN 1993-1-1. 2005. Evrokod 3: Projektiranje jeklenih konstrukcij – 1-1. del: Splošna pravila in pravila za stavbe., Ljubljana, Slovenski inštitut za standardizacijo

Kristanič, N. 2008. Sinteza konstrukcij z uporabo točne občutljivostne analize in optimizacije oblike v nelinearnem področju. Doktorska disertacija. Ljubljana , Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 148 f.

Kristanič, N., Korelc, J. 2005. Občutljivostna analiza in optimizacija na osnovi neposrednega odvajanja parametrično podane mreže končnih elementov v simbolični obliki, Srečanje Kuhljevi dnevi, 2005, Slovensko društvo za mehaniko

Kržič, F. 1994. Jeklene konstrukcije 1, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 208 str.

Povhe, I. 2002. Tridimenzionalna občutljivostna analiza in optimizacija konstrukcij. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo: 94 f.

Tang, P. H., Chang, K.-H. 2001. Integration of toplogy and shape optimization for design of structural components. Struct Multidisc Optim 22: 65-82

# Priloga A: Algoritem optimizacije topologije – mejna obtežba

- ***Konstante***

```
nparamX=6;(* število parametrov v smeri X *)
nparamY=7;(* število parametrov v smeri Y *)

(* Začetne vrednosti geometrije *)
L0=500.;    (* cm *)
H0=100.;    (* cm *)
t0=40.;     (* cm *)
tmin=0.01;  (* cm *)

(* Obtežba in material *)
pload=1;
qload=10;
displacement=100.;(* cm *)
σy=23.5;          (* kN/cm^2*)
Emod=21000.;      (* kN/cm^2*)

(* Faktorji za račun *)
iteracija=0;
μ1=0.1; (* faktor namenske funkije obtežbe *)
μ2=0.1; (* faktor namenske funkije minimalne vrednosti *)
VolNorm=L0 H0 t0/100;
△λmax=0.4;
ord=1;

(* Parametri *)
Clear["ϕ*"];
nparam=nparamX*nparamY+2;
ϕ=Array[ToExpression["ϕ"<>ToString[#]]&,nparam];
ϕinit=Join[Array[t0&,nparamX*nparamY],Array[H0&,2]];
hi=Take[ϕ,-2];
ti=Table[ϕ[[j+nparamX*(i-1)]],{i,nparamY},{j,nparamX}];
srch=MapThread[Rule,{ϕ,ϕinit}];
nx=3(nparamX-1); ny=8; nz=3(nparamY-1);


domain={
  {"konzola","MySED3H1DFLPH1IsoHookeMisses",{"E*"->Emod,"ν*"->0.3,"σy*"->σy}},
  {"cload","MyLineLoadGConst", {"Y *"→qload}},
  {"const","MyPrescribedDispl",{"Prescribed displ"->displacement}}};
```

- ***Polje začetnih občutljivosti***

```
SMTInputData["CDriver"];
SMTAddDomain[domain];

SMTMesh["konzola","H1",{nx,ny,nz},
  Table[{Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1)
    L0/(nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]]) (j-1)/(nparamY-1),+ti[[j,i]]/2},{i,nparamX}],
```

```
Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparam-1))/L0+hi[[2]]) (j-1)/(nparamY-1),-ti[[j,i]]/2},
    {i,nparamX}]},{j,nparamY}],"InterpolationOrder"→ord];
SMTMesh["cload","C1",{nx},{{0,0,0},{L0,0,0}},"InterpolationOrder"→ord];
SMTAddElement["const",{{L0,0,0}}];

SMTAnalysis["SearchFunction"→(#/.srch&)];
XYZ=SMTNodes[[All,2;;-2]];
δXδϕ=Map[Flatten,Transpose[Map[ D[XYZ,#]& ,ϕ]]];
wnode=SMTFindNodes["X"==L0&& "Y"==0&& "Z"==0&][[1]];
λnode=SMTFindNodes["ID"=="load"&][[1]];

grafall={{{0,pload},{displacement,pload}}};
```

- ***Direktna in občutljivostna analiza***

```
analysis[t_,ϕo_?(VectorQ[#,NumberQ]&)]:=(
  iteracija++;
  htoϕ=MapThread[Rule,{ϕ,Map[If[#<tmin,tmin,#]&,ϕo]}];
  htoϕ1=MapThread[Rule,{ϕ,ϕo}];
  Print["ϕ= ",htoϕ];

SMTInputData[];
SMTAddDomain[domain];

SMTMesh["konzola","H1",{nx,ny,nz},
  Table[{Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]]) (j-1)/(nparamY-1),+ti[[j,i]]/2},{i,nparamX}],
  Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]]) (j-1)/(nparamY-1),-ti[[j,i]]/2},
    {i,nparamX}]},{j,nparamY}]/.htoϕ,"InterpolationOrder"→ord];
SMTMesh["cload","C1",{nx},{{0,0,0},{L0,0,0}}/.htoϕ,
    "InterpolationOrder"→ord];

SMTAddElement["const",{{L0,0,0}}/.htoϕ];
SMTAddEssentialBoundary[{"Z"==0&,Null,Null,0},{"X"==0&,0,0,0}];
SMTAddSensitivity[Array[{ToString[ϕ[[#]]],ϕ[[#]]/.htoϕ,_→{2,#}}&,{nparam}]]

SMTAnalysis["Output"→"Polje.out","PostOutput"→"Polje",
    "Solver"→{5,{11,0,0,9,1},{}}];

SMTSave[htoϕ, L0, qload, displacement, σy, Emod, graf, grafall, iteracija,
    nx, ny, nz, nparamX, nparamY, μ1, μ2, hi, t0];

SMTNodeData["Data",N[δXδϕ/.htoϕ]];
SMTNextStep[1,△λmax];
  graf={{0,0}};
  While[
    While[step=SMTConvergence[10^-8,15,{"Adaptive",8,0.001,△λmax,1}],
    SMTNewtonIteration[];];
  If[step[[4]]==="MinBound",
    SMTStatusReport[" Error: △λ < △λmin"];
```

```
    SMTStepBack[];
    ϕdivergence=ϕo;
    ];
  If[Not[step[[1]]],
    SMTPut[SMTIData["Step"]];
  If[t==="s",SMTSensitivity[];];
    AppendTo[graf,{SMTNodeData[wnode,"at"][[2]],
    SMTNodeData[λnode,"at"][[1]]}];    ];
    step[[3]] ,
  If[step[[1]],SMTStepBack[];];
  SMTNextStep[1,step[[2]]]];
  cload=SMTNodeData[λnode,"at"][[1]];
  ∈load=0.00001;

  If[cload-pload<∈load
   ,Φ1=-μ1 Log[∈load]-(μ1 (load[ϕ]-pload-∈load))/
      ∈load+(μ1(load[ϕ]-pload-∈load)^2)/(2 ∈load^2)
   ,Φ1=-μ1 Log[load[ϕ]-pload];];
  Φ2=1/VolNorm volumen[ϕ];
  Φ3=Total[Table[
     If[(ϕ[[i]]/.htoϕ1)/tmin-1<∈load
      ,-μ2 Log[∈load]-(μ2 (ϕ[[i]]-tmin (1+∈load)))/(tmin ∈load)+
         (μ2(ϕ[[i]]-tmin (1+∈load))^2)/(2 tmin^2 ∈load^2)
      ,-μ2 Log[ϕ[[i]]/tmin-1]      ]
     ,{i,Length[ϕ]}]];

  Φ=Φ1+ Φ2+Φ3;
  δΦ1δϕ=Map[D[Φ1,#]&,ϕ];
  δΦ2δϕ=Map[D[Φ2,#]&,ϕ];
  δΦ3δϕ=Map[D[Φ3,#]&,ϕ];
  δΦδϕ=Map[D[Φ,#]&,ϕ];

  vsens=SMTTask["VolumeSensitivity"];
  trans={
      load[ϕ]->cload,
   Derivative[i_][load][_]:→SMTNodeData[λnode,"st"][[Position[i,1][[1,1]]]],
      volumen[ϕ]->vsens[[2,1]],
   Derivative[i_][volumen][_]:→vsens[[2,Position[i,1][[1,1]]+1]],
      htoϕ}//Flatten;

  If[t==="d",
   SMTShowMesh["DeformedMesh"→True, "Field"→"Misses stress",
"Combine"→(Row[{#,ListLinePlot[graf]}]&), "Domains">"konzola",
"Show"→"Window"]
   ];
  AppendTo[grafall,graf];

  If[t==="d"
    ,Φ
    ,Print[Grid[{
       {"i", "Φ1", "Φ2", "Φ3", "Φ", "δΦ1δϕ", "δΦ2δϕ",
       "δΦ3δϕ","δΦδϕ","graf"},
       {iteracija/3, Φ1, Φ2, Φ3, Φ, δΦ1δϕ, δΦ2δϕ, δΦ3δϕ, δΦδϕ,
       ListLinePlot[{graf,{{0,pload},{displacement,pload}}},
       PlotRange→All]}/.trans/.trans},Frame→All]]; δΦδϕ
    ]/.trans )
```

- ***Optimizacija***

```
time1=AbsoluteTime[];
step=0;
steps={};

{Φmin,ϕmin}=FindMinimum[analysis["d",ϕs],{ϕs,ϕinit},
    MaxIterations→600, Gradient→analysis["s",ϕs],
    "Method"→{{{"QuasiNewton","StepControl"→"LineSearch"}},
    {"QuasiNewton"},{"Automatic"}}[[3]]]

time2=AbsoluteTime[];
time=time2-time1;
Print["Za račun potreboval: ",DateList[time][[-4]]-1,"dni
",DateList[time][[-3]],"h ",DateList[time][[-2]],"min ",DateList[time][[-
1]],"sec"]
```

# Priloga B: Algoritem optimizacije topologije – maksimalna togost pri danem volumnu

- ***Konstante***

```
nparamX=6;(* število parametrov v smeri X *)
nparamY=7;(* število parametrov v smeri Y *)

(* Začetne vrednosti geometrije *)
L0=500.;    (* cm *)
H0=100.;    (* cm *)
t0=20.;     (* cm *)
tmin=0.01;  (* cm *)
tmax=30;
V0=L0 H0 t0;

(* Obtežba in material *)
qload=50000/nx;
σy=1000000;  (* kN/cm^2*)
Emod=21000.; (* kN/cm^2*)

(* Faktorji za račun *)
iteracija=0;
μ1=1;
μ2=0.1;
VolNorm=L0 H0 t0/10;
△λmax=0.4;
ord=1;

(* Parametri *)
Clear["ϕ*"];
nparam=nparamX*nparamY;
ϕ=Array[ToExpression["ϕ"<>ToString[#]]&,nparam];
ϕinit=Array[t0&,nparamX*nparamY];
hi=Array[H0&,2];
ti=Table[ϕ[[j+nparamX*(i-1)]],{i,nparamY},{j,nparamX}];
srch=MapThread[Rule,{ϕ,ϕinit}];
nx=4(nparamX-1); ny=12; nz=4(nparamY-1);

domain={"konzola","MySED3H1DFLPH1IsoHookeMisses",{"E *"->Emod,"ν *"-
>0.3,"σy *"->σy}};
```

- ***Polje začetnih občutljivosti***

```
SMTInputData["CDriver"];
SMTAddDomain[domain];

SMTMesh["konzola","H1",{nx,ny,nz},
  Table[{Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]]) (j-1)/(nparamY-1),+ti[[j,i]]/2},{i,nparamX}],
```

```
  Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]]) (j-1)/(nparamY-1),-ti[[j,i]]/2},
{i,nparamX}]},{j,nparamY}],"InterpolationOrder"→ord];
SMTAnalysis["SearchFunction"→(#/.srch&)];
XYZ=SMTNodes[[All,2;;-2]];
δXδϕ=Map[Flatten,Transpose[Map[ D[XYZ,#]& ,ϕ]]];
wnode=SMTFindNodes["X"==L0&& "Y"==0&& "Z"==0&][[1]];
```

- ***Direktna in občutljivostna analiza***

```
analysis[t_,ϕo_?(VectorQ[#,NumberQ]&)]:=(
  iteracija++;
  htoϕ=MapThread[Rule,{ϕ,Map[If[#<tmin,tmin,#]&,ϕo]}];
  htoϕ1=MapThread[Rule,{ϕ,ϕo}];
  Print["ϕ= ",htoϕ];

  SMTInputData[];
  SMTAddDomain[domain];

SMTMesh["konzola","H1",{nx,ny,nz},
  Table[{Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])(j-1)/(nparamY-1),+ti[[j,i]]/2},{i,nparamX}],
  Table[{(i-1) L0/(nparamX-1),-((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])/2+((hi[[1]]-hi[[2]])*(L0-(i-1) L0/
    (nparamX-1))/L0+hi[[2]])(j-1)/(nparamY-1),-ti[[j,i]]/2},
    {i,nparamX}]},{j,nparamY}]/.htoϕ,"InterpolationOrder"→ord];
SMTAddNaturalBoundary["Y"==0&&"Z"==0&,Null,qload,Null];
SMTAddEssentialBoundary[{"Z"==0&,Null,Null,0},{"X"==0&,0,0,0}];
SMTAddSensitivity[Array[{ToString[ϕ[[#]]],ϕ[[#]]/.htoϕ,_→{2,#}}&,{nparam}]]
SMTAnalysis["Output"→"Polje.out","PostOutput"→"Polje",
    "Solver"→{5,{11,0,0,9,1},{}}];
SMTSave[htoϕ, L0, qload, σy, Emod, iteracija, nx, ny, nz, nparamX, nparamY,
    μ1, μ2, hi];

SMTNodeData["Data",N[δXδϕ/.htoϕ]];
SMTNextStep[1,1];
SMTPut[SMTIData["Step"]];
SMTNewtonIteration[];
SMTSensitivity[];

cpomik=SMTNodeData["X"==L0&&"Y"==0&&"Z"==0&,"at"][[1,2]];
vsens=SMTTask["VolumeSensitivity"];
∈load=0.00001;

If[vsens[[2,1]]/V0>(1-∈load)
   ,Φ1=-μ1 Log[∈load]+(μ1 (volumen[ϕ]/V0-1+∈load))/
     ∈load+(μ1 (volumen[ϕ]/V0-1+∈load)^2)/(2 ∈load^2)
   ,Φ1=-μ1 Log[-volumen[ϕ]/V0+1];
   ];
```

```
Φ2=pomik[ϕ];
Φ3=Total[Table[
     If[(ϕ[[i]]/.htoϕ1)/tmin-1<∊load
      ,-μ2 Log[∊load]-(μ2 (ϕ[[i]]-tmin (1+∊load)))/
       (tmin ∊load)+(μ2 (ϕ[[i]]-tmin (1+∊load))^2)/(2 tmin^2 ∊load^2)
      ,-μ2 Log[ϕ[[i]]/tmin-1]]
     ,{i,Length[ϕ]}]];
Φ=Φ1+ Φ2+Φ3;
δΦ1δϕ=Map[D[Φ1,#]&,ϕ];
δΦ2δϕ=Map[D[Φ2,#]&,ϕ];
δΦ3δϕ=Map[D[Φ3,#]&,ϕ];
δΦδϕ=Map[D[Φ,#]&,ϕ];
trans={
  pomik[ϕ]->cpomik,
    Derivative[i_][pomik][_]:→{Take
      [SMTNodeData["X"==L0&&"Y"==0&&"Z"==0&,"st"][[1]],{2,-1,3}]}
      [[1,Position[i,1][[1,1]]]],
  volumen[ϕ]->vsens[[2,1]],
    Derivative[i_][volumen][_]:→vsens[[2,Position[i,1][[1,1]]+1]],htoϕ}
      //Flatten;
    If[t==="d",
   SMTShowMesh["DeformedMesh"→True,
    "Field"→"Misses stress","Domains"->"konzola","Show"→"Window"]
   ];
    If[t==="d"
    ,Φ
    ,Print[Grid[{
      {"i", "Φ1", "Φ2", "Φ3", "Φ", "δΦ1δϕ", "δΦ2δϕ", "δΦ3δϕ" ,"δΦδϕ"},
      {iteracija/3, Φ1, Φ2, Φ3, Φ, δΦ1δϕ, δΦ2δϕ, δΦ3δϕ, δΦδϕ}
      /.trans/.trans},Frame→All]];
    δΦδϕ
    ]/.trans
  )
```

- ***Optimizacija***

```
time1=AbsoluteTime[];
step=0;
steps={};

{Φmin,ϕmin}=FindMinimum[analysis["d",ϕs],{ϕs,ϕinit},
    MaxIterations→600, Gradient→analysis["s",ϕs],
    "Method"→{{{"QuasiNewton","StepControl"→"LineSearch"}},
    {"QuasiNewton"},{"Automatic"}}[[3]]]

time2=AbsoluteTime[];
time=time2-time1;
Print["Za račun potreboval: ",DateList[time][[-4]]-1,"dni
",DateList[time][[-3]],"h ",DateList[time][[-2]],"min ",DateList[time][[-
1]],"sec"]
```

# Priloga C: Algoritem optimizacije parametrov na mejno obtežbo

- ***Konstante***

```
(* Začetne vrednosti geometrije *)
L=500.;   (* dolžina nosilca [cm] *)
hw0=100;  (* začetna višina stojine [cm] *)
tw0=4;    (* začetna debelina stojine [cm] *)
bf0=50;   (* začetna širina pasnice [cm] *)
tf0=4;    (* začetna debelina pasnice [cm] *)
tmin=5;   (* minimalna debelina pločevine [cm]*)

(* Obtežba in material *)
pload=1;
qload=10;
displacement=40.; (* cm *)
σy=23.5; (* kN/cm^2*)
Emod=21000.; (* kN/cm^2*)
∈komp=Sqrt[23.5/σy]

(* Faktorji za račun *)
iteracija=0;
μ1=0.1;   (* faktor namenske funkcije obtežbe *)
μ2=0.1;   (* faktor namenske funkcije minimalne vrednosti *)
VolNorm=L hw0 bf0/10;
△λmax=0.4;
ord=1;

(* Parametri
 ϕ1 = hw - višina stojine,
 ϕ2 = bf - širina pasnice *)

nparam=2;
Clear["ϕ*"];
ϕ=Array[ToExpression["ϕ"<>ToString[#]]&,nparam];
ϕinit=Join[{hw0},{bf0}];
hw=ϕ[[1]];
bf=ϕ[[2]];
tw=hw/(72 ∈komp );
tf=bf/(18 ∈komp);
srch=MapThread[Rule,{ϕ,ϕinit}];

(* Mreža *)
nL=20 ;   (*mreža po dolžini *)
nwh=8 ;   (* mreža v stojini višina, nujno SODO! *)
nfb=8;    (* mreža v pasnici širina *)
nwt=4;    (* mreža v stojini debelina, nujno SODO! *)
nft=3;    (* mreža v pasnici debelina *)

domain={
  {"konzola","MySED3H1DFLPH1IsoHookeMisses",{"E*"->Emod,"ν*"->0.3,"σy*"->σy}},
  {"cload","MyLineLoadGConst", {"Y *"→qload}},
  {"const","MyPrescribedDispl",{"Prescribed displ"->displacement}}};
```

- ***Polje začetnih občutljivosti***

```
SMTInputData["CDriver"];
SMTAddDomain[domain];

 (*  - G E O M E T R I J A -  *)
 (* STOJINA *)
 (* stojina do varov *)
SMTMesh["konzola","H1",{nL,nwt,nwh},
    {{{{0,-hw/2,tw/2},{L,-hw/2,tw/2}},{{0,-hw/2,-tw/2},{L,-hw/2,-tw/2}}},
    {{{0,hw/2,tw/2},{L,hw/2,tw/2}},{{0,hw/2,-tw/2},{L,hw/2,-tw/2}}}}];

 (* ZGORNJA PASNICA *)
 (* sredina pasnice *)
SMTMesh["konzola","H1",{nL,nwt,nft},
    {{{{0,hw/2,tw/2},{L,hw/2,tw/2}},{{0,hw/2,-tw/2},{L,hw/2,-tw/2}}},
    {{{0,tf+hw/2,tw/2},{L,tf+hw/2,tw/2}},{{0,tf+hw/2,-tw/2},
    {L,tf+hw/2,-tw/2}}}}];
 (* pasnica - *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,hw/2,-tw/2},{L,hw/2,-tw/2}},{{0,hw/2,-bf/2},{L,hw/2,-bf/2}}},
    {{{0,tf+hw/2,-tw/2},{L,tf+hw/2,-tw/2}},{{0,tf+hw/2,-bf/2},
    {L,tf+hw/2,-bf/2}}}}];
 (* pasnica + *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,hw/2,+bf/2},{L,hw/2,+bf/2}},{{0,hw/2,tw/2},{L,hw/2,tw/2}}},
    {{{0,tf+hw/2,+bf/2}, {L,tf+hw/2,+bf/2}},{{0,tf+hw/2,tw/2},
    {L,tf+hw/2,tw/2}}}}];

 (* SPODNJA PASNICA *)
 (* sredina pasnice *)
SMTMesh["konzola","H1",{nL,nwt,nft},
    {{{{0,-hw/2,-tw/2},{L,-hw/2,-tw/2}},{{0,-hw/2,tw/2},{L,-hw/2,tw/2}}},
    {{{0,-tf-hw/2,-tw/2},{L,-tf-hw/2,-tw/2}},{{0,-tf-hw/2,tw/2},
    {L,-tf-hw/2,tw/2}}}}];
 (* pasnica - *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,-hw/2,-bf/2},{L,-hw/2,-bf/2}},{{0,-hw/2,-tw/2},
    {L,-hw/2,-tw/2}}},{{{0,-tf-hw/2,-bf/2},{L,-tf-hw/2,-bf/2}},
    {{0,-tf-hw/2,-tw/2},{L,-tf-hw/2,-tw/2}}}}];
 (* pasnica + *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,-hw/2,tw/2},{L,-hw/2,tw/2}},{{0,-hw/2,+bf/2},{L,-hw/2,+bf/2}}},
    {{{0,-tf-hw/2,tw/2},{L,-tf-hw/2,tw/2}},{{0,-tf-hw/2,+bf/2},
    {L,-tf-hw/2,+bf/2}}}}];

SMTMesh["cload","C1",{nL},{{0,0,0},{L,0,0}},"InterpolationOrder"→ord];
SMTAddElement["const",{{L,0,0}}];

SMTAnalysis["SearchFunction"→(#/.srch&)];
XYZ=Transpose[Transpose[SMTNodes][[{2,3,4}]]];
δXδφ=Map[Flatten,Transpose[Map[ D[XYZ,#]& ,φ]]];
wnode=SMTFindNodes["X"==L&& "Y"==0&& "Z"==0&][[1]];
λnode=SMTFindNodes["ID"=="load"&][[1]];

grafall={{{0,pload},{2,pload}}};
```

- ***Direktna in občutljivostna analiza***

```
analysis[t_,ϕo_?(VectorQ[#,NumberQ]&)]:=(
  iteracija++;
  htoϕ=MapThread[Rule,{ϕ,Map[If[#<tmin,tmin,#]&,ϕo]}];
  htoϕ1=MapThread[Rule,{ϕ,ϕo}];
  Print["ϕ= ",htoϕ];

  SMTInputData[];
  SMTAddDomain[domain];

(*  - G E O M E T R I J A -  *)
 (* STOJINA *)
 (* stojina do varov *)
SMTMesh["konzola","H1",{nL,nwt,nwh},
    {{{{0,-hw/2,tw/2},{L,-hw/2,tw/2}},{{0,-hw/2,-tw/2},{L,-hw/2,-tw/2}}},
    {{{0,hw/2,tw/2},{L,hw/2,tw/2}},{{0,hw/2,-tw/2},{L,hw/2,-tw/2}}}}];

 (* ZGORNJA PASNICA *)
 (* sredina pasnice *)
SMTMesh["konzola","H1",{nL,nwt,nft},
    {{{{0,hw/2,tw/2},{L,hw/2,tw/2}},{{0,hw/2,-tw/2},{L,hw/2,-tw/2}}},
    {{{0,tf+hw/2,tw/2},{L,tf+hw/2,tw/2}},{{0,tf+hw/2,-tw/2},
    {L,tf+hw/2,-tw/2}}}}];
 (* pasnica - *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,hw/2,-tw/2},{L,hw/2,-tw/2}},{{0,hw/2,-bf/2},{L,hw/2,-bf/2}}},
    {{{0,tf+hw/2,-tw/2},{L,tf+hw/2,-tw/2}},{{0,tf+hw/2,-bf/2},
    {L,tf+hw/2,-bf/2}}}}];
 (* pasnica + *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,hw/2,+bf/2},{L,hw/2,+bf/2}},{{0,hw/2,tw/2},{L,hw/2,tw/2}}},
    {{{0,tf+hw/2,+bf/2}, {L,tf+hw/2,+bf/2}},{{0,tf+hw/2,tw/2},
    {L,tf+hw/2,tw/2}}}}];

 (* SPODNJA PASNICA *)
 (* sredina pasnice *)
SMTMesh["konzola","H1",{nL,nwt,nft},
    {{{{0,-hw/2,-tw/2},{L,-hw/2,-tw/2}},{{0,-hw/2,tw/2},{L,-hw/2,tw/2}}},
    {{{0,-tf-hw/2,-tw/2},{L,-tf-hw/2,-tw/2}},{{0,-tf-hw/2,tw/2},
    {L,-tf-hw/2,tw/2}}}}];
 (* pasnica - *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,-hw/2,-bf/2},{L,-hw/2,-bf/2}},{{0,-hw/2,-tw/2},
    {L,-hw/2,-tw/2}}},{{{0,-tf-hw/2,-bf/2},{L,-tf-hw/2,-bf/2}},
    {{0,-tf-hw/2,-tw/2},{L,-tf-hw/2,-tw/2}}}}];
 (* pasnica + *)
SMTMesh["konzola","H1",{nL,nfb,nft},
    {{{{0,-hw/2,tw/2},{L,-hw/2,tw/2}},{{0,-hw/2,+bf/2},{L,-hw/2,+bf/2}}},
    {{{0,-tf-hw/2,tw/2},{L,-tf-hw/2,tw/2}},{{0,-tf-hw/2,+bf/2},
    {L,-tf-hw/2,+bf/2}}}}];
SMTMesh["cload","C1",{nL},{{0,0,0},{L,0,0}}/.htoϕ,
    "InterpolationOrder"→ord];
SMTAddElement["const",{{L,0,0}}/.htoϕ];
SMTAddEssentialBoundary[ "X"==0&,0,0,0];
SMTAddSensitivity[Array[{ToString[ϕ[[#]]],ϕ[[#]]/.htoϕ,→{2,#}}&,{nparam}]];
```

```
SMTAnalysis["Output"→"Profil.out", "PostOutput"→"Profil",
    "Solver"→{5,{11,0,0,9,1},{}}];
SMTSave[htoϕ, qload, displacement, σy, Emod, graf, grafall, iteracija,
    komentar, μ1, μ2];
SMTNodeData["Data",N[δXδϕ/.htoϕ]];
SMTNextStep[1,△λmax];
  graf={{0,0}};
  While[
    While[step=SMTConvergence[10^-8,15,{"Adaptive",8,0.001,△λmax,1}],
    SMTNewtonIteration[];];
  If[step[[4]]==="MinBound",
    SMTStatusReport[" Error: △λ < △λmin"];
    SMTStepBack[];
    ϕdivergence=ϕo;
    ];
  If[Not[step[[1]]],
    SMTPut[SMTIData["Step"]];
  If[t==="s",SMTSensitivity[];];
    AppendTo[graf,{SMTNodeData[wnode,"at"][[2]],
    SMTNodeData[λnode,"at"][[1]]}];    ];
    step[[3]] ,
  If[step[[1]],SMTStepBack[];];
  SMTNextStep[1,step[[2]]]];

cload=SMTNodeData[λnode,"at"][[1]];
∈load=0.00001;

If[cload-pload<∈load
   ,Φ1=-μ1 Log[∈load]-(μ1 (load[ϕ]-pload-∈load))/
     load+(μ1(load[ϕ]-pload-∈load)^2)/(2 ∈load^2)
   ,Φ1=-μ1 Log[load[ϕ]-pload];
   ];

Φ2=1/VolNorm volumen[ϕ];
Φ3=Total[Table[
      If[(ϕ[[i]]/.htoϕ1)/tmin-1<∈load
       ,-μ2 Log[∈load]-(μ2 (ϕ[[i]]-tmin (1+∈load)))/
         (tmin ∈load)+(μ2 (ϕ[[i]]-tmin (1+∈load))^2)/(2 tmin^2 ∈load^2)
       ,-μ2 Log[ϕ[[i]]/tmin-1]
       ]
      ,{i,Length[ϕ]}]];
Φ=Φ1+ Φ2+Φ3;

δΦ1δϕ=Map[D[Φ1,#]&,ϕ];
δΦ2δϕ=Map[D[Φ2,#]&,ϕ];
δΦ3δϕ=Map[D[Φ3,#]&,ϕ];
δΦδϕ=Map[D[Φ,#]&,ϕ];

vsens=SMTTask["VolumeSensitivity"];
trans={
  load[ϕ]->cload,
    Derivative[i_][load][_]:→
     SMTNodeData[λnode,"st"][[Position[i,1][[1,1]]]],
       volumen[ϕ]->vsens[[2,1]],
    Derivative[i_][volumen][_]:→vsens[[2,Position[i,1][[1,1]]+1]],
       vsens[[2,Position[i,1][[1,1]]+1]],htoϕ}//Flatten;
```

```
If[t==="d",
   SMTShowMesh["DeformedMesh"→True, "Field"→"Misses stress",
     "Combine"→(Row[{#,ListLinePlot[graf]}]&), "Domains"->"konzola",
     "Show"→"Window"]
   ];

AppendTo[grafall,graf];
  If[t==="d"
    ,Φ
    ,Print[Grid[{
       {"i", "Φ1", "Φ2", "Φ3", "Φ", "δΦ1δϕ", "δΦ2δϕ",
       "δΦ3δϕ","δΦδϕ","graf"},
       {iteracija/3, Φ1, Φ2, Φ3, Φ, δΦ1δϕ, δΦ2δϕ, δΦ3δϕ, δΦδϕ,
       ListLinePlot[{graf,{{0,pload},{displacement,pload}}},
       PlotRange→All]}/.trans/.trans},Frame→All]];
    δΦδϕ
    ]/.trans
  )
```

- ***Optimizacija***

```
time1=AbsoluteTime[];
step=0;
steps={};

{Φmin,ϕmin}=FindMinimum[analysis["d",ϕs],{ϕs,ϕinit},
    MaxIterations→600, Gradient→analysis["s",ϕs],
    "Method"→{{{"QuasiNewton","StepControl"→"LineSearch"}},
    {"QuasiNewton"},{"Automatic"}}[[3]]]

time2=AbsoluteTime[];
time=time2-time1;
Print["Za račun potreboval: ",DateList[time][[-4]]-1,"dni
",DateList[time][[-3]],"h ",DateList[time][[-2]],"min ",DateList[time][[-
1]],"sec"]
```