



Univerza v Mariboru

Fakulteta za strojništvo

Marko MOTALN

**MODELIRANJE IN RAZVOJ KODE
ZA ANALIZO STATIČNEGA ODZIVA
GEOMETRIJSKO NELINEARNIH
RAVNINSKIH KONSTRUKCIJ**

Magistrsko delo

študijskega programa 2. stopnje

Strojništvo

Maribor, julij 2022



Univerza v Mariboru

Fakulteta za strojništvo

MODELIRANJE IN RAZVOJ KODE ZA ANALIZO STATIČNEGA ODZIVA GEOMETRIJSKO NELINEARNIH RAVNINSKIH KONSTRUKCIJ

Magistrsko delo

Študent: Marko MOTALN
Študijski program: študijski program 2. stopnje
Strojništvo
Smer: Konstrukterstvo
Mentor: izr. prof. dr. Marko KEGL
Somentor: doc. dr. Boštjan HARL

Licenca Creative Commons BY-NC-ND

Maribor, julij 2022

ZAHVALA

Zahvaljujem se mentorju izr. prof. dr. Marku Keglu in somentorju doc. dr. Boštjanu Harlu za vse nasvete, pomoč in vodenje pri pripravi magistrskega dela.

Zahvaljujem se tudi svoji družini in vsem ostalim, ki so me na kakršenkoli način podpirali in vzdrževali ugodno študijsko okolje.

Modeliranje in razvoj kode za analizo statičnega odziva geometrijsko nelinearnih ravninskih konstrukcij

Ključne besede: MKE, Abaqus, RNS, statični odziv, geometrijska nelinearnost, materialna nelinearnost, Lagrangeeva formulacija

UDK: 519.876.5(043.2)

Povzetek

V zaključnem delu obravnavamo statični odziv ravninskih konstrukcij s pomočjo metode končnih elementov. Za ravninsko napetostno stanje je izpeljan štirivozliščni izoparametrični končni element, ki ga uporabimo za razvoj lastne programske kode, s pomočjo katere lahko analiziramo linearne in nelinearne odzive konstrukcij. Najprej so teoretično predstavljene vse metode in količine, ki jih kasneje uporabimo pri razvoju algoritmov. Prav tako je predstavljeno teoretično ozadje nastanka nelinearnosti v numeričnih simulacijah. Osnovo za razvoj kode predstavlja formulacija linearnega odziva, ki je osnova za opis iteracijskih in inkrementalnih formulacij, s katerimi lahko opišemo različne nelinearnosti. Naloga torej predstavi celoten razvoj programa za analizo po metodi končnih elementov, s katerim je moč izvesti procesiranje in poprocesiranje linearnih in nelinearnih ravninskih analiz.

Modelling and Code development for Static Response Analysis of Geometrically Nonlinear Planar Structures

Keywords: FEM, Abaqus, Plane stress, static response, geometric nonlinearity, material nonlinearity, Lagrangian formulation

UDC: 519.876.5(043.2)

Abstract

In the final thesis, we discuss the static response of planar structures with the help of the finite element method. A four-node isoparametric finite element is derived for the plane stress state, which we use to develop our own software code, with the help of which we can analyse linear and nonlinear construction responses. First, we theoretically present all methods and quantities, which we use later in the development of algorithms. We also present the theoretical backgrounds of the emergence of nonlinearity in numeric simulations. The basis for the development of the code is the formulation of linear response, which is the basis for the description of iterative and incremental formulations, with which we can describe different nonlinearities. The thesis thus presents the entire development of the software for the analysis according to the finite element method, with which we can perform processing and post-processing of linear and nonlinear planar analyses.



Univerza v Mariboru

Fakulteta za strojništvo

(ime članice UM)

IZJAVA O AVTORSTVU ZAKLJUČNEGA DELA

Ime in priimek študenta/-ke: Marko Motain

Študijski program: 2. STOPNJA STROJNIŠTVO

Naslov zaključnega dela: MODELIRANJE IN RAZVOJ KODE ZA ANALIZO STATIČNEGA ODZIVA
GEOMETRIJSKO NELINEARNIH RAVNINSKIH KONSTRUKCIJ

Mentor/-ica: izr.prof.dr.MARKO KEGL, univ.dipl. inž.str

Somentor/-ica: doc.dr. BOŠTJAN HARL, univ.dipl. inž.str

Podpisani/-a študent/-ka Marko Motain

- izjavljam, da je zaključno delo rezultat mojega samostojnega dela, ki sem ga izdelal/-a ob pomoči mentorja/-ice oz. somentorja/-ice;
- izjavljam, da sem pridobil/-a vsa potrebna soglasja za uporabo podatkov in avtorskih del v zaključnem delu in jih v zaključnem delu jasno in ustrezno označil/-a;
- na Univerzo v Mariboru neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico ponuditi zaključno delo javnosti na svetovnem spletu preko DKUM; sem seznanjen/-a, da bodo dela, deponirana/objavljena v DKUM, dostopna široki javnosti pod pogoji licence Creative Commons BY-NC-ND, kar vključuje tudi avtomatizirano indeksiranje preko spleta in obdelavo besedil za potrebe tekstovnega in podatkovnega rudarjenja in ekstrakcije znanja iz vsebin; uporabnikom se dovoli reproduciranje brez predelave avtorskega dela, distribuiranje, dajanje v najem in priobčitev javnosti samega izvirnega avtorskega dela, in sicer pod pogojem, da navedejo avtorja in da ne gre za komercialno uporabo;
- dovoljujem objavo svojih osebnih podatkov, ki so navedeni v zaključnem delu in tej izjavi, skupaj z objavo zaključnega dela.

Uveljavljam permissivnejšo obliko licence Creative Commons: _____

(navedite obliko)

Kraj in datum:

Maribor, 03.07.2022

Podpis študenta/-ke:

Marko Motain



Univerza v Mariboru

Fakulteta za strojništvo

Številka: 42-84-184/2022-SF

Kraj in datum: MARIBOR, 05.04.2022

Na osnovi 330. člena Statuta Univerze v Mariboru (Uradni list RS, št. 41/2021-UPB13) izdajam

SKLEP O ZAKLJUČNEM DELU

MARKO MOTALN, študent študijskega programa 2. stopnje MAG STROJNIŠTVO, smer KONSTRUKTERSTVO, izpolnjuje predpisane pogoje in se mu dovoljuje izdelati zaključno delo.

Tema zaključnega dela je pretežno s področja katedre/oddelka/inštituta: KATEDRA ZA MEHANIKO.

Mentorji:

izr. prof. dr. MARKO KEGL, univ. dipl. inž. str., mentor

doc. dr. BOŠTJAN HARL, univ. dipl. inž. str., somentor

Naslov zaključnega dela:

Modeliranje in razvoj kode za analizo statičnega odziva geometrijsko nelinearnih ravninskih konstrukcij

Naslov zaključnega dela v angleškem jeziku:

Modelling and Code Development for Static Response Analysis of Geometrically Nonlinear Planar Structures

Rok za izdelavo in oddajo zaključnega dela je 05.04.2023. Zaključno delo je potrebno izdelati skladno z navodili: *Navodila za pripravo diplomskega/magistrskega dela*, in ga oddati v pristojnem referatu članice. Število izvodov: 1. Hkrati se odda tudi izjava mentorja in somentorja o ustreznosti zaključnega dela.

Pravni pouk: Zoper ta sklep je možna pritožba na Senat članice v roku 10 delovnih dni od dneva prejema sklepa.



red. prof. dr. Bojan Dolšak, dekan

Obvestiti:

- kandidata,
- mentorja,
- somentorja,
- arhiv.

KAZALO VSEBINE

1	UVOD	1
1.1	Zapis simbolov in notacija.....	1
1.2	Oprelitev problema	2
1.3	Predpostavke, omejitve in cilji zaključnega dela.....	3
2	TEORETIČNE OSNOVE	5
2.1	Metoda končnih elementov	5
2.2	Nelinearnosti v metodi končnih elementov	8
2.3	Geometrijska nelinearnost	9
2.4	Strukturna nelinearnost.....	11
2.5	Materialna nelinearnost	12
2.6	Lagrangeev pristop	13
2.7	Mere napetosti in deformacij	15
2.7.1	Deformacijski gradient	15
2.7.2	Mere deformacij	17
2.7.3	Mere napetosti	19
2.7.4	Numerični pristop reševanja nelinearnih problemov	20
2.7.5	Variacijski pristop – princip virtualnih pomikov	22
2.8	Numerična integracija.....	23
3	IZPELJAVA KONČNIH ELEMENTOV	25
3.1	Ravninski linearni končni element	25
3.2	Ravninski nelinearni končni element	32
3.2.1	Celotna Lagrangeeva formulacija	35
3.2.1	Posodobljena Lagrangeeva formulacija	46
3.2.1	Povezava med formulacijama	56
3.2.2	Upoštevanje materialne nelinearnosti	57

3.2.3	Konvergenčni kriterij	58
4	IMPLEMENTACIJA KODE.....	59
4.1	Bralnik	62
4.2	Uporabniški vmesnik.....	64
5	REZULTATI SIMULACIJ.....	67
5.1	Linearni odziv	67
5.2	Materialna nelinearnost	72
5.3	Geometrijska nelinearnost	75
5.3.1	Pravokotno obremenjen nosilec	76
5.3.2	Ekscentrično obremenjeni nosilec.....	80
5.3.3	Natezno obremenjena kocka	86
6	ZAKLJUČEK.....	92
7	UPORABLJENI VIRI IN LITERATURA	93

KAZALO SLIK

Slika 2.1: Numerično reševanje primerov iz narave	5
Slika 2.2: Diskretizacija domene na poddomene	6
Slika 2.3: Elementarni delec [3]	7
Slika 2.4: Geometrijska nelinearnost (konzolni nosilec)	9
Slika 2.5: Preskok sistema («Snap Trough»)	10
Slika 2.6: Geometrijska nelinearnost	11
Slika 2.7: Strukturna nelinearnost	11
Slika 2.8: Materialna nelinearnost.....	12
Slika 2.9: Lagrangeev pristop.....	14
Slika 2.10: Shematični prikaz posodobljene in celotne L. formulacije.....	15
Slika 2.11: Polarna dekompozicija deformacijskega gradienta [11].....	18
Slika 2.12: Mere deformacij [12]	19
Slika 2.13: Numerično reševanje nelinearnih enačb	21
Slika 2.14: Modificirana NR-metoda levo, NR-metoda desno	22
Slika 2.15: Gaussova numerična integracija [1].....	24
Slika 3.1: Štirivozliščni ravninski element.....	25
Slika 3.2: Sistem 2 KE	31
Slika 4.1: Razvojno okolje MS Visual studio	59
Slika 4.2: Diagram poteka delovanja programa	60
Slika 4.3: Diagram poteka jedra programa.....	61
Slika 4.4: Oblika zapisa INP	62
Slika 4.5: Uporabniški vmesnik	66
Slika 5.1: Plošča z luknjo	68
Slika 5.2: Konvergenčna analiza plošče z luknjo.....	69
Slika 5.3: Pomiki plošče z luknjo, pridobljeni s programsko opremo Abaqus	69
Slika 5.4: Pomiki lastne programske opreme	70
Slika 5.5: Napetosti v plošči z luknjo.....	71
Slika 5.6: Primerjava pomikov plošče z luknjo.....	72
Slika 5.7: Model materialne nelinearnosti.....	73
Slika 5.8: Vnesene točke materialnega modela.....	74
Slika 5.9: Bilinearni materialni model	74

Slika 5.10: Pomiki za odziv materialne nelinearnosti	75
Slika 5.11: Naraščanje pomika »MNO«	75
Slika 5.12: Prvi primer geometrijske nelinearnosti: pravokotno obremenjen nosilec	76
Slika 5.13: Upogib enostransko vpetega nosilca.....	78
Slika 5.14: Linearna analiza LPO	79
Slika 5.15: Nelinearna analiza LPO	80
Slika 5.16: Ekscentrično obremenjen nosilec	81
Slika 5.17: Pomiki linearne analize ekscentrično obremenjenega nosilca LPO	82
Slika 5.18: Pomiki nelinearne analize ekscentrično obremenjenega nosilca LPO	83
Slika 5.19: Horizontalni pomiki točke prijemališča sile	85
Slika 5.20: Vertikalni pomiki točke prijemališča sile	86
Slika 5.21: Natezno obremenjena kocka	87
Slika 5.22: Pomiki linearne analize kocke za RNS	88
Slika 5.23: Pomiki nelinearne analize kocke za RNS	89
Slika 5.24: Horizontalni pomiki sredine prijemališča sile	91

KAZALO PREGLEDNIC

Tabela 1: Členi sistemske togostne matrike	32
Tabela 2: Primerjava rezultatov za ploščo z luknjo	72
Tabela 3: Pomik pravokotno obremenjenega nosilca v smeri Y za točko prijemališča sile	77
Tabela 4: Horizontalni pomiki točke prijemališča sile.....	84
Tabela 5: Vertikalni pomiki točke prijemališča sile.....	86
Tabela 6: Horizontalni pomiki sredine prijemališča sile.....	90

SIMBOLI

\mathbf{B}_e	Gradientna matrika (matrika odvodov interpolacijskih funkcij)
${}^t\mathbf{B}_L^0, {}^t\mathbf{B}_L$	Linearni del gradientne matrike
${}^t\mathbf{B}_{NL}, {}^t\mathbf{B}_{NL}$	Nelinearni del gradientne matrike
$d\mathbf{X}$	Vektor materialnega vlakna deformiranega stanja
$d\mathbf{x}$	Vektor materialnega vlakna začetnega stanja
d_e	Vektor vozliščnih pomikov elementa
\mathbf{D}	Konstitutivna matrika
E	Modul elastičnosti ali Youngov modul
${}^0E'_{ij}, {}^tE_{ij}, {}^{t+\Delta t}E_{ij}$	Green-Lagrangeev deformacijski tenzor
\mathbf{F}	Deformacijski gradient
${}^0F_{i,m}$	Deformacijski gradient
f	Točkovne obremenitve
\mathbf{F}_e	Vektor obremenitev elementa
\mathbf{F}	Vektor obremenitev sistema
${}^t\mathbf{F}, {}^{t+\Delta t}\mathbf{F}$	Vektor notranjih obremenitev
\mathbf{J}	Jacobijeva matrika
\mathbf{J}	Jacobijan
\mathbf{K}_e	Togostna matrika elementa
\mathbf{K}	Sistemska togostna matrika
${}^t\mathbf{K}_L, {}^t\mathbf{K}_L$	Linearni del togostne matrike
${}^t\mathbf{K}_{NL}, {}^t\mathbf{K}_{NL}$	Nelinearni del togostne matrike
\mathbf{L}	Diferencialni operator
\mathbf{N}_e	Matrika interpolacijskih funkcij elementa

n_i	Smeri raztezkov
n	Število končnih elementov
\mathbf{p}	Vektor volumske sile
\mathbf{R}	Rotacijska matrika
${}^{t+\Delta t}\mathbf{R}$	Zunanje obremenitve
${}^tS, {}^0S$	Površina
${}^t_0S_{ij}, {}^t_0S_{ij}, {}^t_0S_{ij}^{t+\Delta t}$	Drugi Piola-Kirchoffov napetostni tenzor
\mathbf{t}	Vektor površinskih obremenitev
${}^t\mathbf{u}_a^k$	Pomik vozlišča k v smeri a ob času t
$u_i, {}^t u_i, {}^{t+\Delta t} u_i$	Pomiki poljubne točke
\mathbf{U}_e	Vektor vozliščnih pomikov elementa
\mathbf{U}	Vektor vozliščnih pomikov sistema
\mathbf{U}	Desni raztezek
${}^t\mathbf{U}, \Delta\mathbf{U}, {}^{t+\Delta t}\mathbf{U}$	Vektor pomikov
\mathbf{V}	Levi raztezek
${}^tV, {}^0V$	Volumen
W_F	Delo notranjih sil
W_R	Delo zunanjih sil
δW_F	Virtualno delo notranjih sil
δW_R	Virtualno delo zunanjih sil
${}^t\mathbf{x}_a^k$	Koordinata vozlišča k v smeri a ob času t
$x_i, {}^t x_i, {}^{t+\Delta t} x_i$	Koordinate poljubne točke

GRŠKI SIMBOLI

σ	Cauchyjev napetostni tenzor
ε	Tenzor malih deformacij
τ, σ	Členi napetostnega tenzorja
ε^L	Tenzor logaritemskih deformacij
$\delta \mathbf{U}^T$	Transponirana vrednost vektorja virtualnih pomikov
σ_{ij}	Cauchyjev napetostni tenzor
ε_{ij}	Tenzor malih deformacij
${}_0 n_{ij}$	Inkrement nelinearnega dela deformacij
${}_0 \varepsilon_{ij}$	Inkrement linearnega dela deformacij
ξ	Ksi (naravna koordinata)
η	Eta (naravna koordinata)
δ_{ij}	Kroneckerjeva delta
${}_0 u_{i,j}, {}_t u_{i,j}$	Odvodi pomikov glede na koordinate ob času 0 ali t
ε_{ij}	Tenzor malih deformacij
λ_i	Glavni raztezki
${}^t \rho, {}^0 \rho$	Gostota
λ, μ	Lamejevi konstanti

KRATICE

MKE	Metoda končnih elementov
FEM	Finite element method
KE	Končni element
ULF	Posodobljena Lagrangeeva formulacija
TLF	Celotna Lagrangeeva formulacija
MNO	Materialna nelinearna formulacija
LPO	Lastna programska oprema
RNS	Ravninsko napetostno stanje
RDS	Ravninsko deformacijsko stanje
PK	Piola-Kirchoffov napetostni tenzor
2PK	Drugi Piola-Kirchoffov napetostni tenzor
PVD	Princip virtualnega dela
IVW	Notranje virtualno delo
EVW	Zunanje virtualno delo
PDE	Parcialna diferencialna enačba

1 UVOD

Magistrsko delo obravnava modeliranje in razvoj kode za primere ravninskih konstrukcij s pomočjo štirivozliščnih izoparametričnih končnih elementov. Sledita validacija pravilnosti reševanja in primerjava rezultatov s sodobno programsko opremo.

1.1 Zapis simbolov in notacija

Tematika magistrskega dela se precej povezuje z mehaniko kontinuuma, kjer vemo, da imamo največkrat opravka s skalarnimi, vektorskimi ali tenzorskimi polji. Že na začetku tega dela bomo zato pojasnili pravila označevanja, ki jih bomo uporabljali v delu:

- skalarne količine: a, b, c, d, e ,
- vektorji: $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}$,
- matrike: $\mathbf{a}, \mathbf{b}, \mathbf{c}, \mathbf{d}, \mathbf{e}$.

Prav tako bomo v nalogi pogosto uporabili zapise z več notacijami, saj gre za količine, ki so odvisne od nanašanja na domeno ob določenem času ter na čas, v katerem količino razberemo. Notacija bo uporabljena zgolj takrat, ko bo to tudi potrebno (za izpeljavo linearnega KE dodatno notiranje ni potrebno, saj vedno govorimo zgolj o eni časovni domeni).

(ob času)
(nanaša se na čas) **ZNAK** koordinatna os

Omeniti moramo še uporabo Einsteinovega indeksnega zapisa, ki je zelo dobro opisan v številnih knjigah in je izjemno učinkovit za zapisovanje vektorskih in tenzorskih količin:

- primer tenzorske količine 4. reda: C_{ijkl}
- v primeru, da so med indeksi prisotne vejice, te ponazarjajo odvode:

$${}^0u_{1,1} = \frac{\partial u_1}{\partial {}^0x_1}$$

1.2 Opredelitev problema

Numerične metode so v zadnjih desetletjih z izjemno hitro rastjo računalniške moči postale standardno orodje za reševanje najrazličnejših samostojnih ali združenih fizikalnih problemov. V strukturalni mehaniki je zagotovo najbolj poznana metoda končnih elementov (MKE, poznana tudi pod angl. FEM), ki služi kot dobro uveljavljeno orodje za približno reševanje diferencialnih enačb, ki jih ni mogoče rešiti z neposredno integracijo. S tem zagotovimo obravnavo zelo zapletenih poljubnih geometrij s pomočjo diskretizacije na poddomene, ki jim pravimo končni elementi.

Z nastankom klasičnih linearnih simulacij, ki so od začetka veljale zgolj za strukture linearne odziva, so bile sprejete tudi številne predpostavke, ki so za najpogosteje zgolj enofizikalen problem močno olajšale obravnavo. S tem pa je nabor možnosti simuliranja omejen in nam omogoča zgolj izvedbo simuliranja enostavnejših problemov, pri katerih pa so zaradi sprejetih predpostavk pogosto tudi rezultati zgolj želeni približek realni vrednosti. S silovitim napredkom v računalništvu ter zmožnostjo hitrejšega izvajanja računskih operacij so se razvila tudi veliko zahtevnejša in računsko potratnejša simulacijska orodja, s pomočjo katerih lahko izvajamo analize s pomočjo MKE na praktično vsakem želenem področju (analiza trdnin, analiza tekočin, optimizacija topologije, toplotna analiza, lomnomehanske analize, večfizikalne analize itd.).

Na področju numerične mehanike trdnin lahko s pomočjo uporabe statičnih in dinamičnih nelinearnih analiz ter s pravilno fizikalno interpretacijo zagotovimo bolj natančno obravnavo klasičnih problemov, kot tudi obravnavo struktur, narejenih iz najrazličnejših sodobnih kovinskih ali nekovinskih materialov z nelinearnimi odzivi, katerih uporaba se zaradi njihovih zelo dobrih lastnosti stalno povečuje. Prav tako lahko simuliramo proizvodne procese, kjer so prisotne zelo velike spremembe geometrije (pri preoblikovanju imamo opravka z velikimi deformacijami), npr. kovanje.

Obstajajo različni pristopi za reševanje, ki temeljijo na inkrementalnih ali iteracijskih numeričnih pristopih. Pri inkrementalnih metodah se obremenitev razdeli na več delnih inkrementov, kjer za vsak inkrement predvidimo linearen sistem. Natančnost takih sistemov se povečuje s povečevanjem števila inkrementov. Drugi pristop je iteracijski, kjer se med iteracijami togostna matrika ne spreminja. S pomočjo ponavljanja Newton-Rapsonove metode

lahko torej v vsakem koraku izenačimo notranje in zunanje (residualne) sile ter iščemo konvergenco pri določeni rešitvi. Zelo pomembna je pravilna izbira velikosti inkrementov ter števila iteracij, saj je s tem tudi neposredno povezana cena simulacije. V literaturi so se pojavili različni pristopi za formulacijo različnih nelinearnosti, ki pa temeljijo na že zelo dolgo dobro poznanih formulacijah, kot so:

- celotna Lagrangeeva formulacija (angl. *Total Lagrangian formulation*),
- posodobljena Lagrangeeva formulacija (angl. *Updated Lagrangian formulation*),
- materialna nelinearna formulacija (angl. *Material only*).

Skozi nalogo se bomo ukvarjali z reševanjem dvodimenzionalnih problemov, ki bodo vsebovali različne nelinearnosti, ki so med sabo močno povezane in jih je treba dosledno in pravilno fizikalno interpretirati (možnost opisa materialne nelinearnosti glede na opis kinematike). Za popolni opis deformiranega stanja je treba v vsaki točki telesa poznati pomike, deformacije in napetosti, kar storimo s pomočjo:

- ravnovesnih enačb,
- kinematičnih enačb,
- konstitutivnih enačb.

1.3 Predpostavke, omejitve in cilji zaključnega dela

Osnovni cilj tega magistrskega dela je izdelati program, s katerim bo možno simulirati primere RNS za linearne in nelinearne analize. S tem imamo v mislih celotni proces procesiranja in poprocesiranja. Pri tem bomo upoštevali naslednje točke:

- Osnovo zaključnega dela bo predstavljal študij literature na področju nelinearnih numeričnih analiz v strukturalni mehaniki, kjer se bomo v tem delu omejili na študij geometrijske nelinearnosti ter pripadajočih numeričnih metod, ki so potrebne za implementacijo algoritmov.
- Razvili bomo lasten bralnik, ki bo znal prebrati in razčleniti predprocesirane podatke programske opreme Abaqus.

- Razvita bo koda za procesiranje in poprocesiranje linearne analize po metodi končnih elementov za probleme ravninskega napetostnega stanja (RNS) (štirivozliščni izoparametrični končni elementi).
- Razvita bo koda za procesiranje in poprocesiranje nelinearne analize po metodi končnih elementov za probleme ravninskega napetostnega stanja (RNS) (štirivozliščni izoparametrični končni elementi).
- Izvedene bodo numerične simulacije za primere brez in z upoštevanjem geometrijskih nelinearnosti.
- Izvedena bo primerjava med rezultati med različnimi analizami za poljuben ravninski problem, prikazane bodo razlike, prednosti in slabosti med različnimi formulacijami.
- Prikazano bo upoštevanje materialne nelinearnosti na primeru analize MNO.

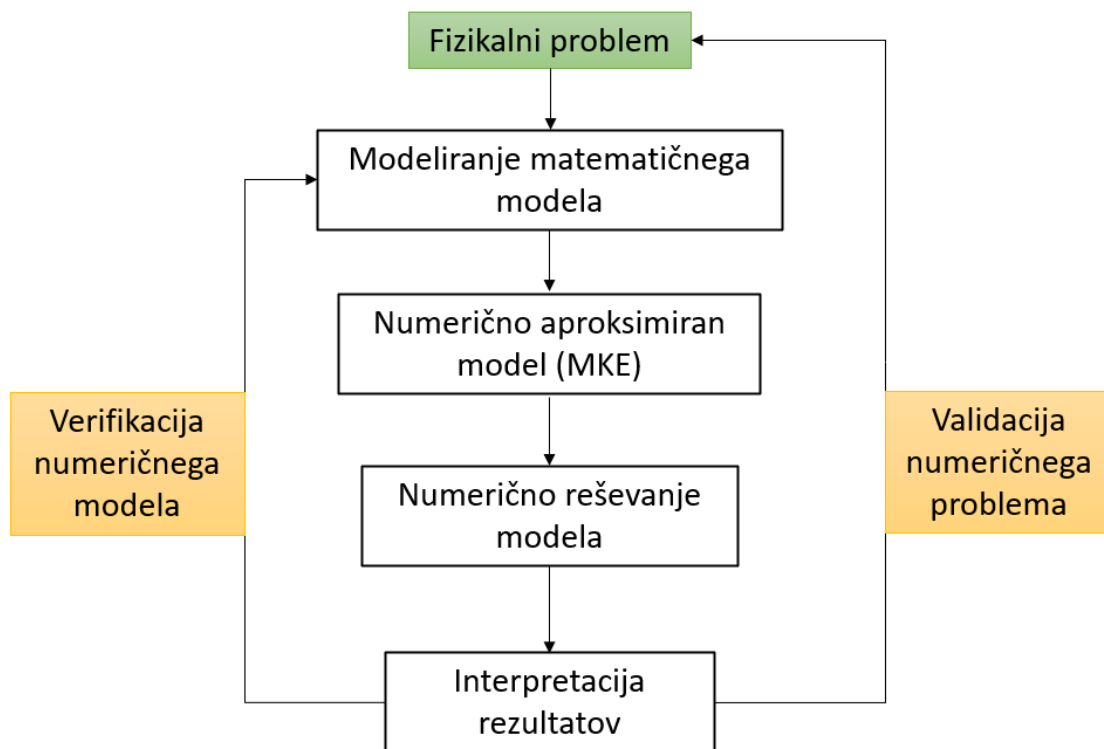
Pri izvajanju zadanih ciljev so bile sprejete tudi nekatere predpostavke:

- Z lastno programsko kodo bo izvedena zgolj analiza procesiranja in postprocesiranja, medtem ko bomo fazo predprocesiranja uvozili iz programskega okolja Abaqus.
- Za izpeljavo končnega elementa in lastne kode bo predpostavljeno ravninsko napetostno stanje (RNS), kjer bo koda izpeljana za štirivozliščne izoparametrične končne elemente.
- Za implementiranje programske kode bosta uporabljena programsko okolje MS Visual Studio in programski jezik C#.

Dokument je sestavljen iz uvoda, kjer predstavimo obravnavano problematiko, uporabljeno označevanje simbolov, zadane cilje, omejitve in predpostavke. Sledi poglavje, v katerem teoretično predstavimo vse osnove, ki jih potrebujemo za razumevanje MKE ter nelinearnosti, ki nastopijo pri izvajanju numeričnih analiz. Teoretično so predstavljene tudi nove količine, ki jih potrebujemo pri izpeljavi in formulaciji nelinearne analize. Z vpeljanimi količinami že delno stopimo v tretje poglavje, kjer izpeljemo formulacije za linearno analizo ter različne vrste nelinearne analize. V četrtem poglavju so predstavljeni implementacija kode bralnika, analize ter diagrami poteka. Prav tako je na kratko predstavljen nastali uporabniški vmesnik. Zadnji dve poglavji zajemata predstavitev rezultatov simulacij in uporabljeno literaturo.

2 TEORETIČNE OSNOVE

Osnovna ideja uporabe numeričnih metod reševanja je, da dejanski problem iz narave najprej ustrezno fizikalno interpretiramo in ga temeljno dobro razumemo. Sledi idealizacija, kjer s pomočjo ustreznih dovoljenih predpostavk sestavimo matematični model, ki ga lahko rešujemo na različne načine. Sledi reševanje matematičnega modela, ki ga v našem primeru diskretiziramo in rešimo s pomočjo numeričnih metod. Na koncu je potrebna ustrezna interpretacija rezultatov. Pravilnost reševanja problema pa mora biti prav tako validirana in verificirana.

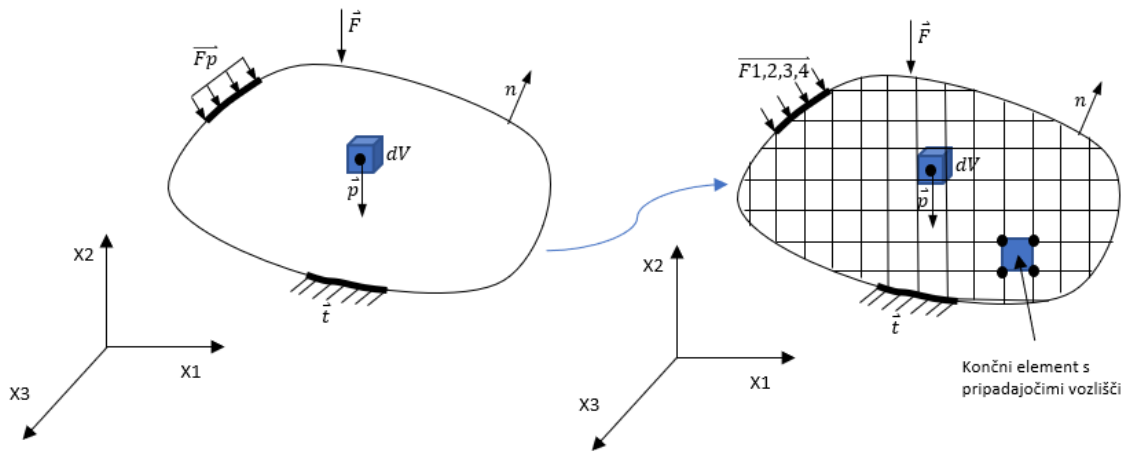


Slika 2.1: Numerično reševanje primerov iz narave

2.1 Metoda končnih elementov

S pomočjo povzetkov iz literature [1][2] vidimo, da lahko k reševanju elastomehanskih problemov pristopimo na različne načine. Neposredni pristop je še vedno zelo omejen, njegove rešitve pa so ob nastopu bolj zahtevnih predpostavk in geometrij izjemno težavne ali

pa še vedno pogosto nerešljive. Na tem področju poznamo Navier-Cauchyjeve, Beltrami-Michelove enačbe ali pa problem rešujemo s pomočjo neposredne integracije. Veliko bolj poznan je variacijski pristop, ki ponuja formulacijo različnih numeričnih pristopov, med katerimi pa je zagotovo najbolj priljubljena metoda končnih elementov (MKE), ki služi kot izjemno močno orodje za približno reševanje diferencialnih enačb. MKE je diskretizacijska metoda, ki sistem z neskončno veliko prostostnih stopenj pretvori (glej sliko 2.2 levo) v diskretni sistem s končnim številom prostostnih stopenj (glej sliko 2.2 desno). Vsak končni element ima znano število vozlišč in poznano polje količin po elementu, ki ga aproksimiramo s pomočjo interpolacijskih funkcij (linearne, kvadratne itd.).



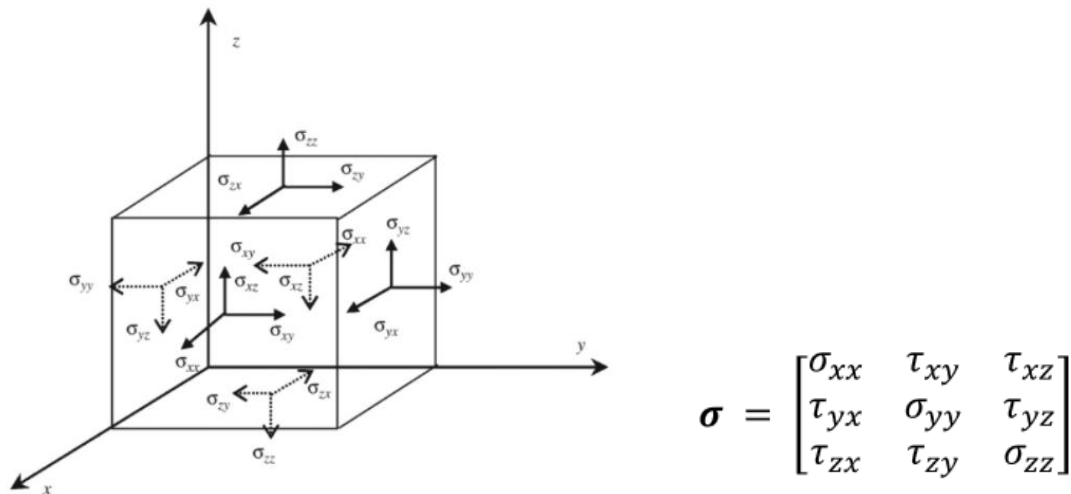
Slika 2.2: Diskretizacija domene na poddomene

Pri osnovni formulaciji imamo na voljo ravnovesne enačbe za obravnavo statičnih problemov:

$$\begin{aligned} \frac{\partial \sigma_{xx}}{\partial x} + \frac{\partial \tau_{yx}}{\partial y} + \frac{\partial \tau_{zx}}{\partial z} + p_x &= 0, \\ \frac{\partial \tau_{xy}}{\partial x} + \frac{\partial \sigma_{yy}}{\partial y} + \frac{\partial \tau_{zy}}{\partial z} + p_y &= 0, \\ \frac{\partial \tau_{xz}}{\partial x} + \frac{\partial \tau_{yz}}{\partial y} + \frac{\partial \sigma_{zz}}{\partial z} + p_z &= 0. \end{aligned}$$

(2.1)

Oznaka p pri tem predstavlja komponente vektorja volumnske sile, τ in σ pa sta člena napetostnega tenzorja z 9 komponentami, ki ga lahko idejno prikažemo s pomočjo elementarnega volumnskega delca na naslednji sliki.



Slika 2.3: Elementarni delec [3]

Sledijo kinematične enačbe ($\varepsilon = \varepsilon(u)$), ki jih lahko prikažemo s pomočjo vektorja specifičnih deformacij.

$$\varepsilon = \left[\frac{\partial u_x}{\partial x} \quad \frac{\partial u_y}{\partial y} \quad \frac{\partial u_z}{\partial z} \quad \frac{\partial u_y}{\partial x} + \frac{\partial u_x}{\partial y} \quad \frac{\partial u_z}{\partial y} + \frac{\partial u_y}{\partial z} \quad \frac{\partial u_x}{\partial z} + \frac{\partial u_z}{\partial x} \right]^T \quad (2.1)$$

Na koncu sledijo še konstitutivne enačbe $\sigma = \sigma(\varepsilon(u))$, ki predstavljajo zvezo med napetostmi in deformacijami. Najbolj enostavno linearno zvezo, ki je najpogosteje uporabljena, si lahko idejno predstavljamo tudi s pomočjo Hookovega zakona [1].

$$\begin{aligned} \sigma_{xx} &= (\lambda + 2\mu)\varepsilon_{xx} + \lambda(\varepsilon_{yy} + \varepsilon_{zz}) \\ \sigma_{yy} &= (\lambda + 2\mu)\varepsilon_{yy} + \lambda(\varepsilon_{xx} + \varepsilon_{zz}) \\ \sigma_{zz} &= (\lambda + 2\mu)\varepsilon_{zz} + \lambda(\varepsilon_{xx} + \varepsilon_{yy}) \\ \tau_{xy} &= 2\mu\varepsilon_{xy} \quad , \quad \tau_{yz} = 2\mu\varepsilon_{yz} \quad , \quad \tau_{zx} = 2\mu\varepsilon_{zx} \end{aligned} \quad (2.2)$$

Z omenjenimi enačbami dobimo sistem 15 PDE, ki predstavlja rešitve za pomike, napetosti in deformacije. Pri tem imamo na voljo še dodatne robne pogoje v obliki predpisanih obremenitev ali pomikov. S pomočjo omenjenih zloženih enačb lahko zapišemo osnovno enačbo za reševanje po MKE, ki pa velja zgolj za linearne probleme in jo bomo kasneje za nelinearno formulacijo nekoliko priredili.

$$\mathbf{F}_e = \mathbf{K}_e \cdot \mathbf{U}_e \quad (2.3)$$

Pri tem je treba upoštevati, da je celotni sistem zgrajen iz končnega števila elementov, katerih matrike s posebnim postopkom, opisanem v poglavju 3, zložimo v končno sistemsko togostno matriko.

$$\mathbf{F}_e = \mathbf{K}_e \cdot \mathbf{U}_e \quad \rightarrow \quad \mathbf{F} = \mathbf{K} \cdot \mathbf{U} \quad (2.4)$$

\mathbf{F} – vektor obremenitev,

\mathbf{K} – togostna matrika,

\mathbf{U} – vektor vozliščnih pomikov,

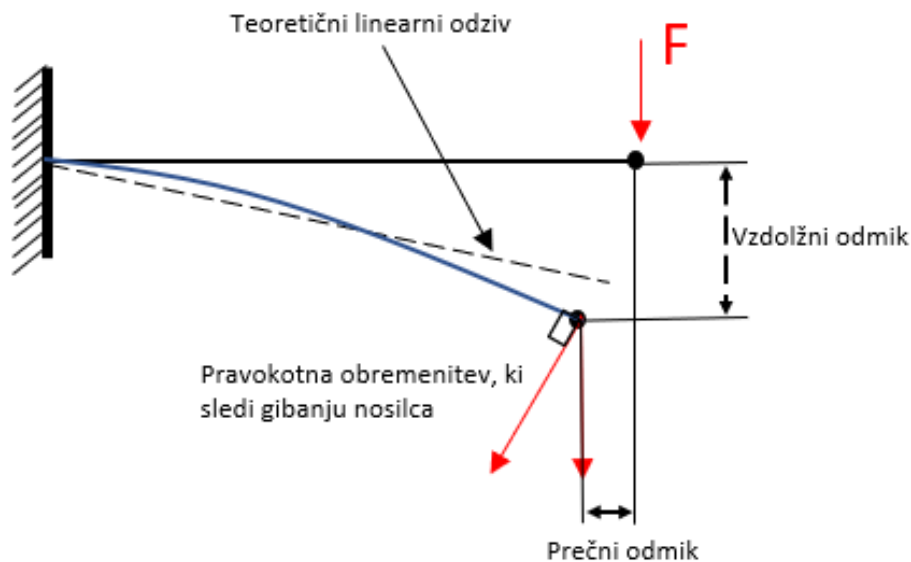
n – število končnih elementov.

2.2 Nelinearnosti v metodi končnih elementov

V klasični strukturni mehaniki so za najpogosteje obravnavane linearne numerične probleme pogosto uporabljene številne predpostavke, ki poenostavijo konstitutivni in kinematični opis problema. Z vidika materialnih lastnosti so torej izbrani linearno elastični modeli, kjer ne presegamo meje tečenja. Kinematično gledano, pa obravnavamo strukture, na katerih rezultirajo mali pomiki, rotacije in deformacije končnega elementa. Posledično lahko togost opišemo kot konstanto, saj se med obremenjevanjem ne spreminja. Ko govorimo o nelinearnih problemih, pa se togost neprestano spreminja, bodisi zaradi materialnega modela ali pa so deformacije, translacije ali rotacije tako velike, da je geometrija sistema spremenjena.

2.3 Geometrijska nelinearnost

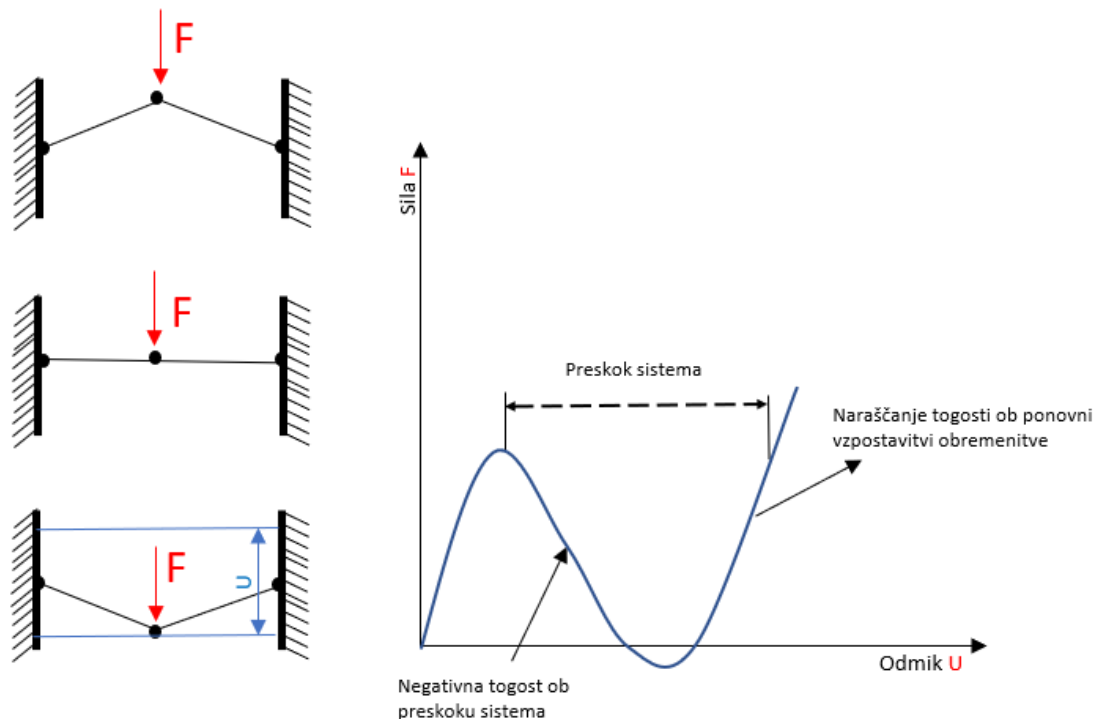
Geometrijska nelinearnost oz. nelinearnost velikih deformacij se pojavi na točki, ko pomiki, rotacije, deformacije povzročijo dovolj veliko spremembo na geometriji sistema in s tem neposredno spremembo togosti, kot lahko vidimo na naslednjem primeru (slika 2.4). Opazimo lahko, da bi pri linearni analizi konzolno vpetega nosilca predpostavljali začetne robne pogoje, ki pa se med razvojem spremembe geometrije nosilca močno spremenijo. Sila, ki ne sledi smeri pravokotno na nosilec, se namreč ob nezanemarljivem odklonu razvije na komponento, ki je pravokotna na nosilec, in na komponento, ki deluje v vzdolžni smeri. S tem se togost sistema bistveno spremeni [4].



Slika 2.4: Geometrijska nelinearnost (konzolni nosilec)

S pomočjo povzetkov iz literature [5], [6] lahko prikažemo tudi drugi primer (slika 2.5), ki ponovno zelo dobro opiše problematiko spremembe geometrije in se navezuje na stabilnost konstrukcij in temelji na primeru dolgega nosilca ali pa npr. velike plošče, ki je členkasto vpeta ob obeh togih straneh. Pri tem pride pri dovolj veliki obremenitvi, ki je prikazana z rdečo barvo, do »preskoka« sistema iz konkavne oblike v konveksno obliko plošče ali nosilca. Med preskokom sistema lahko zabeležimo velike nelinearnosti, ki lahko rezultirajo tudi v obliki negativne togosti sistema. Takoj ko se napetostno stanje v plošči ponovno vzpostavi, pa

lahko vidimo, da se ob nadaljnjem povečevanju sile togost sistema ponovno povečuje. Zavedati se je treba, da bi lahko ob izvajanju linearne analize pridobili popolnoma napačne rezultate.



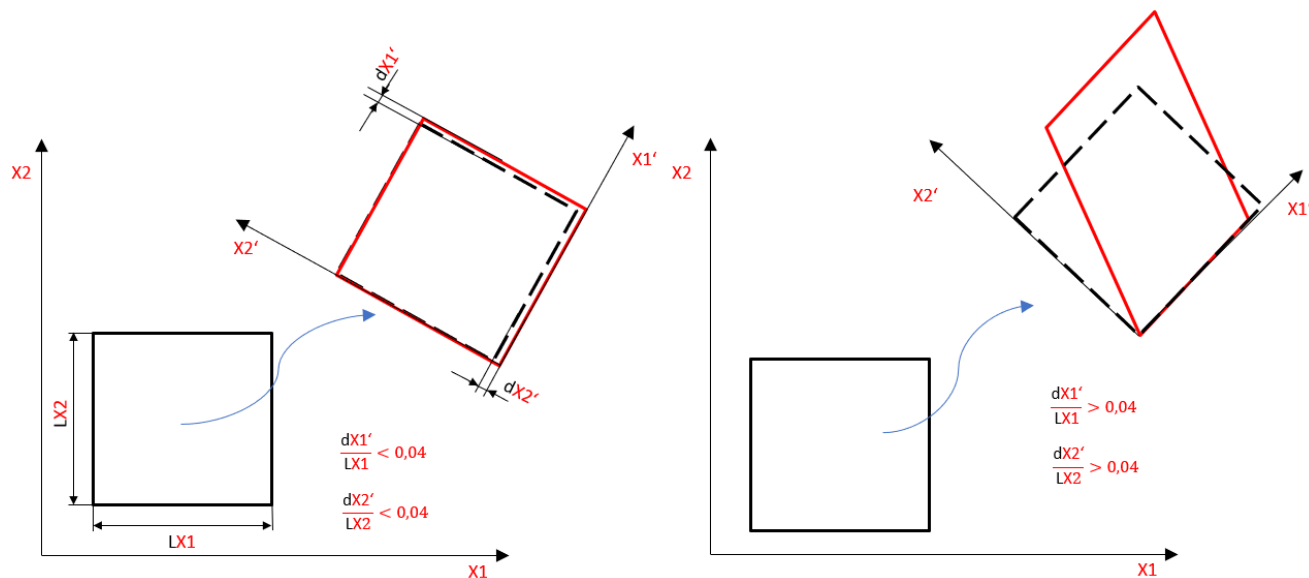
Slika 2.5: Preskok sistema («Snap Trough»)

Avtor [7] deli geometrijsko nelinearnost na dva različna modela, kjer pri prvem (slika 2.6 levo) upošteva velike rotacije, velike translacije, a predvideva, da so deformacije še vedno majhne. V takšnem primeru lahko predvidevamo, da bo razmerje napetosti-deformacije linearno ali pa nelinearno. Opisani primer lahko formuliramo s pomočjo:

- materialne nelinearne formulacije (angl. *Material only*) v primeru, da so rotacije, translacije in deformacije majhne;
- celotne Lagrangeeve formulacije (angl. *Total Lagrangian formulation*) v primeru, ko so rotacije in translacije velike. Kinematično gledano, je formulacija primerna tudi za opis velikih deformacij, vendar pa zaradi nezmožnosti materialnega opisa velikih deformacij za opis desne slike (2.6) v večini ni primerna.

V drugem primeru (slika 2.6 desno) poleg velikih translacij in rotacij nastopijo še velike deformacije, s čimer praviloma nastane tudi potreba po nelinearnem opisu razmerja napetosti-

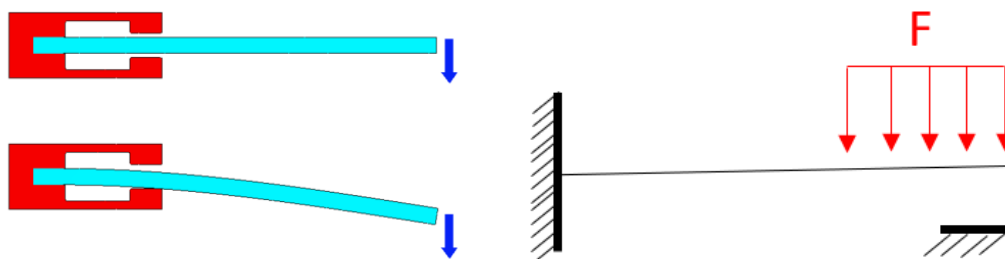
deformacije. Posledično lahko primer formuliramo s posodobljeno Lagrangeovo formulacijo (angl. *Updated Lagrangian formulation*) v kombinacijami z merami, ki popišejo tudi velike deformacije (dejanske logaritmične deformacije).



Slika 2.6: Geometrijska nelinearnost

2.4 Strukturna nelinearnost

Pri strukturni nelinearnosti gre za spremembo robnih pogojev med potekom obremenjevanja komponente. Na sliki desno lahko vidimo, da se konzolno vpet nosilec ob obremenjevanju pomika navzdol, kjer se na točki, ko konica doseže togo vpetje, spremenijo robni pogoji, saj so nadaljnji pomiki v smeri obremenjevanja onemogočeni. Odziv postane nelinearen tudi na sliki 2.7 levo, kjer ob določenem pomiku zabeležimo skrajšanje ročice upogibne obremenitve in posledično spremenjen odziv togosti.



Slika 2.7: Strukturna nelinearnost [8]

2.5 Materialna nelinearnost

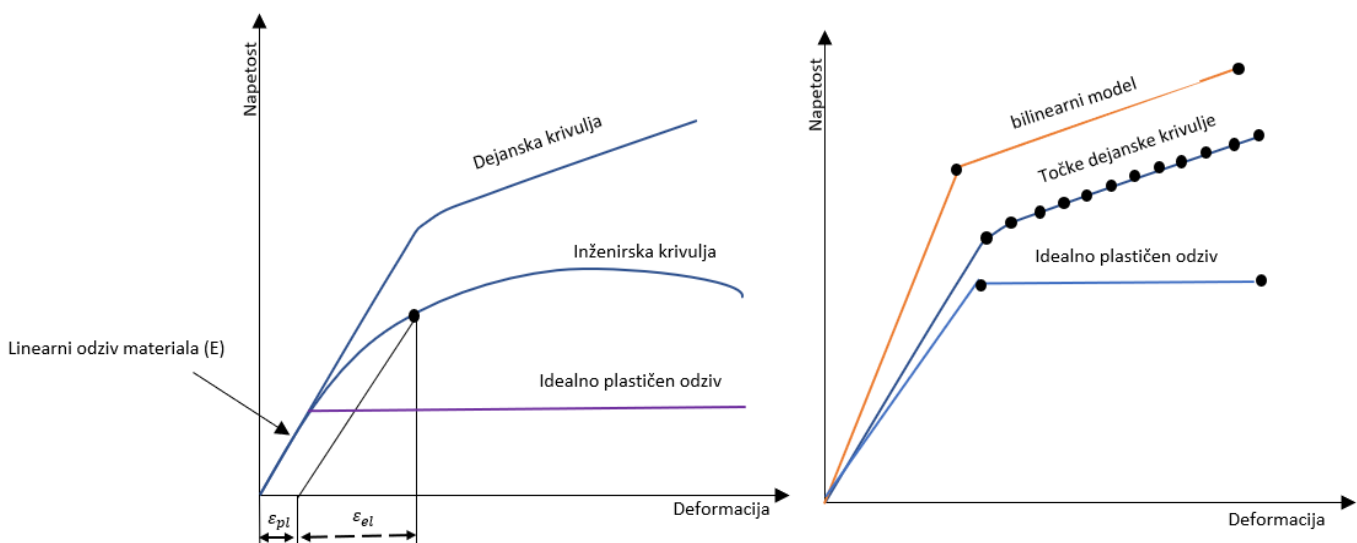
Če povzamemo literaturo [9], ugotovimo, da gre za nelinearnost, ki se v mehaniki trdnin pojavlja najpogosteje, saj lahko odziv elasto-plastičnosti povežemo s konstitutivnim zakonom, ki je neposredno povezan s togostno matriko (glej enačbo 2.5), ki skozi simulacijo ni več konstantna. Gre za izjemno široko področje, kjer so se skozi čas razvili številni modeli, ki opisujejo materialne nelinearnosti. Glavne nelinearnosti pa delimo tudi glede na časovno odvisnost.

Časovno neodvisna materialna nelinearnost:

- nelinearna elastičnost (hiperelastičnost, hipoelastičnost),
- plastičnost (linearna elasto-plastičnost, nelinearna elasto-plastičnost).

Časovno odvisna materialna nelinearnost:

- viskozno obnašanje materiala (viskoelastičnost, viskoplastičnost),
- deformacijsko utrjevanje.



Slika 2.8: Materialna nelinearnost

V numerični mehaniki trdnin lahko materialno nelinearnost opisujemo na različne načine, kot so:

- točke dejanske krivulje,

- odsekovno linearni model,
- potenčni zakon,
- bilinearni model,
- idealno plastični model.

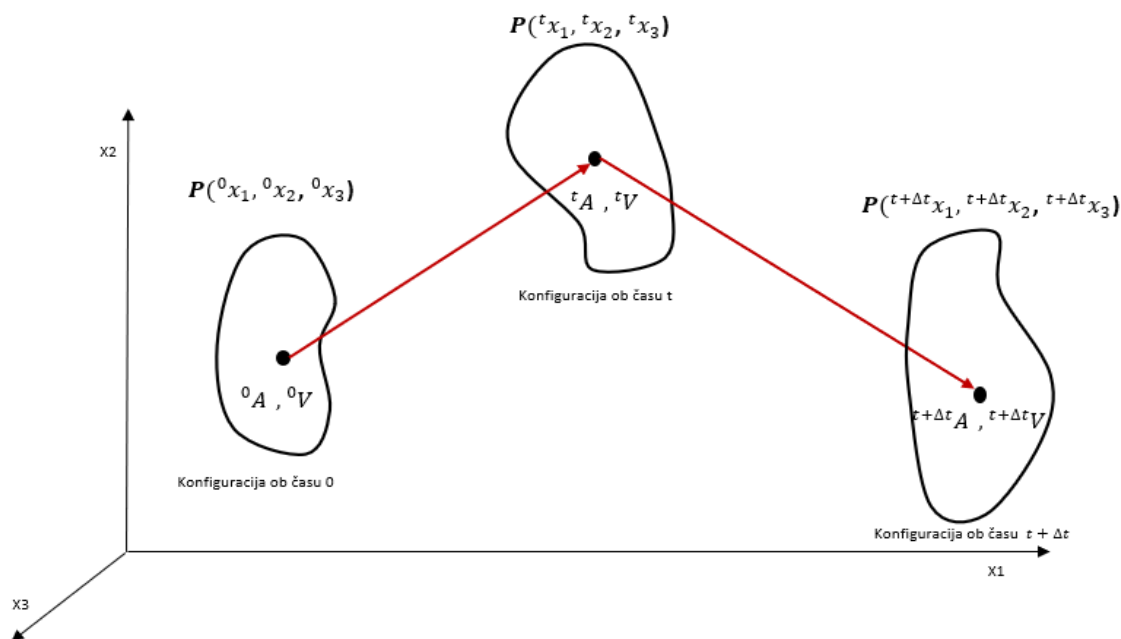
Pri tem storimo določeno napako, za katero se moramo prepričati, da je še sprejemljiva, in ob potrebi povišati število vnesenih točk.

2.6 Lagrangeev pristop

Gibanje materialnega delca je možno opisati na več načinov. V literaturi [2], [10] je najpogosteje uporabljen Eulerjev opis, kjer imenujemo prostorske ali Eulerjeve koordinate, ki predstavljajo nepremično mrežo skozi čas. Spremljamo torej različne količine, ki se gibajo skozi neko točko v odvisnosti od časa (hitrost, tlak, temperatura itd.). Formulacija je zelo dobra za opisovanje količin v mehaniki tekočin, medtem ko se v mehaniki trdnin uporablja bolj redko.

V mehaniki trdnin želimo domeno nekega elementa oz. nosilca spremljati skozi čas od začetne do končne konfiguracije. Vsa polja (pomikov, deformacij, napetosti) so torej materialni objekti. V mehaniki trdnin imamo po navadi opravka z vozlišči, kjer vseskozi spremljamo enak materialni delec (ni pretakanja materiala preko mej KE). V mreži z Lagrangeevim opisom lahko tako relativno na enostaven način priredimo robne pogoje.

Avtor [7] opiše gibanje domene s pomočjo Lagrangeevega pristopa v treh različnih časih, kjer so prisotne vse tri kinematične nelinearnosti. Gre torej za velike pomike točk domene, ki jih spremljamo v začetni konfiguraciji z začetnim volumnom in površino. S premikom domene se prav tako spremenita volumen in površina obravnavane domene, kot prikazuje naslednja slika. Predpostavljamo, da poznamo iskane količine ob času 0 in t , iščemo pa količine v zadnji konfiguraciji $t + \Delta t$.



Slika 2.9: Lagrangeev pristop

Zapišemo lahko zvezo med pomikom materialnega delca iz začetne konfiguracije do poznane konfiguracije ob času t .

$${}^t x_i = {}^0 x_i + {}^t u_i \quad , \quad i = 1,2,3 \quad (2.5)$$

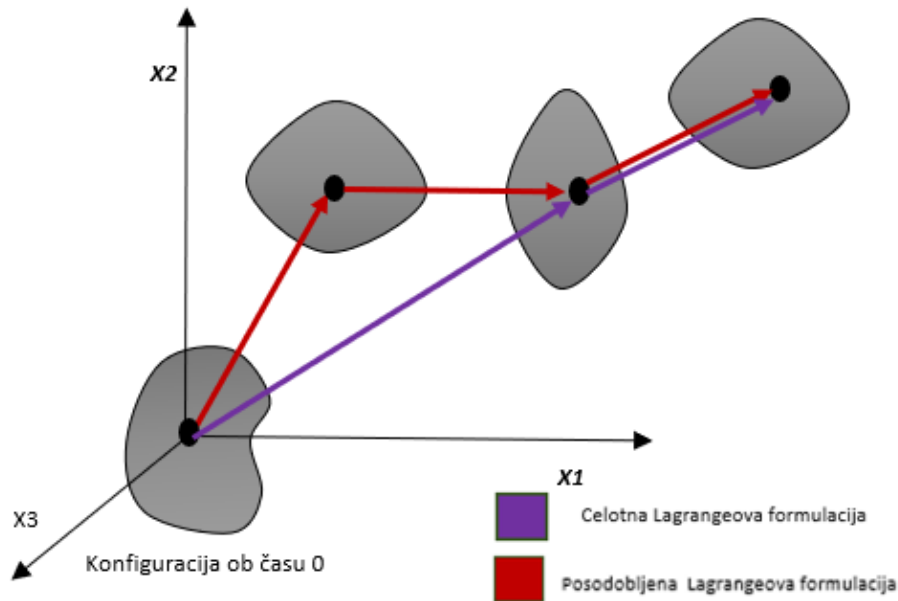
Podobno sledi:

$${}^{t+\Delta t} x_i = {}^0 x_i + {}^{t+\Delta t} u_i \quad (2.6)$$

Na koncu lahko torej zapišemo inkrementalne pomike od časa t do časa $t + \Delta t$.

$$u_i = {}^{t+\Delta t} u_i - {}^t u_i \quad (2.7)$$

Na naslednji sliki sledi shematski prikaz poteka posodobljene in celotne Lagrangeeve formulacije, kjer vidimo, da so pri posodobljeni formulaciji vse količine vezane na zadnjo znano domeno (ob času t), medtem ko pri celotni Lagrangeevi formulaciji vedno znova izhajamo iz začetne domene (ob času $t = 0$). Kot bomo videli v naslednjem poglavju, je to bistvenega pomena v postopku izpeljave KE, saj morajo biti izbrane mere za deformacije in napetosti energetsko kompatibilne in integrirane prek ustrezno izbranega volumna domene, ki se zaradi inkrementacije vseskozi spreminja.



Slika 2.10: Shematični prikaz posodobljene in celotne L. formulacije

2.7 Mere napetosti in deformacij

2.7.1 Deformacijski gradient

Gre za tenzorsko količino, ki je del srca mehanike kontinuuma in je zelo dobro opisana v številnih knjigah ter literaturi. S pomočjo knjige [11] si lahko predstavljamo, da opazujemo opazovano domeno v začetni konfiguraciji (ob času 0) in v končni konfiguraciji (ob času t), kot je prikazano na naslednji sliki. Opazujemo vektorja dveh materialnih delcev, ki sta označena z A in B, ter definiramo pripadajoče krajevne vektorje in pomike iz nedeformirane v deformirano konfiguracijo. Iščemo torej funkcijo, ki slika/povezuje deformirano in nedeformirano stanje.

Iz opisa sledi:

$$d\mathbf{X} = \mathbf{F} \cdot d\mathbf{x} \quad (2.8)$$

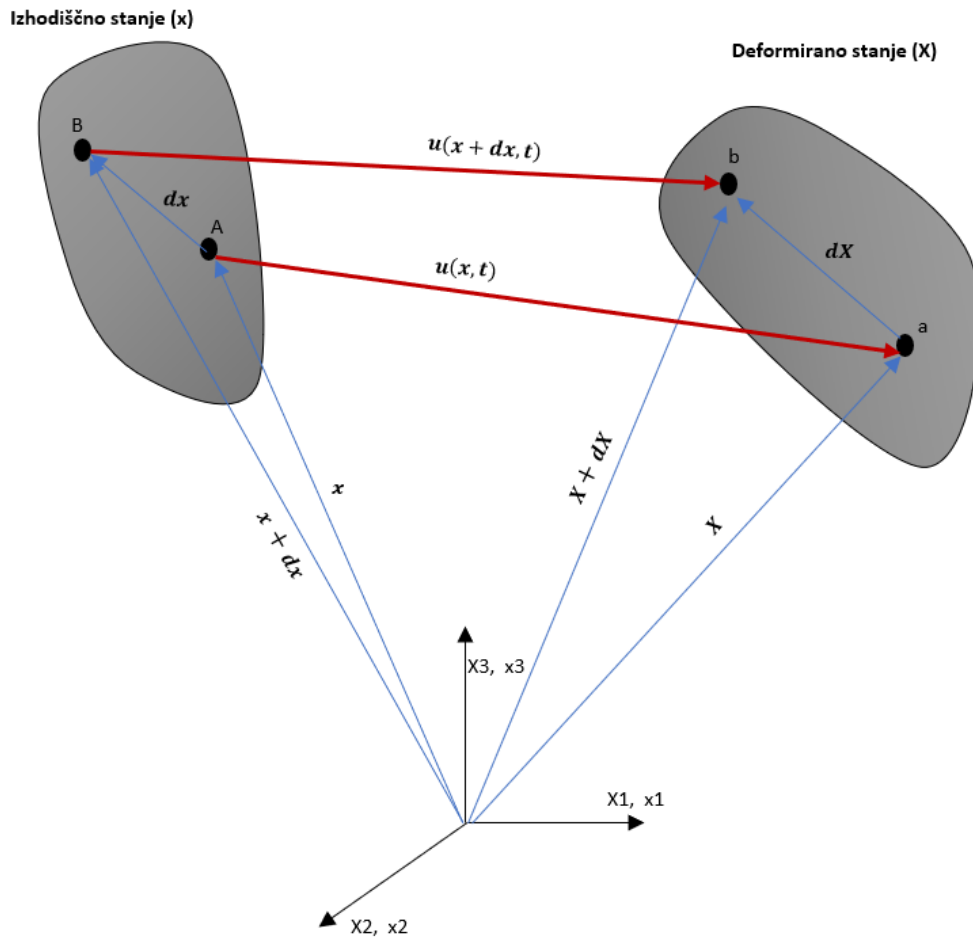
Kjer sta:

$$d\mathbf{X} = [dX_1, dX_2, dX_3]$$

$$d\mathbf{x} = [dx_1, dx_2, dx_3]$$

In deformacijski gradient:

$$\mathbf{F} = F_{ij} = \frac{\partial X_i}{\partial x_j} = \begin{bmatrix} \frac{\partial X_1}{\partial x_1} & \frac{\partial X_1}{\partial x_2} & \frac{\partial X_1}{\partial x_3} \\ \frac{\partial X_2}{\partial x_1} & \frac{\partial X_2}{\partial x_2} & \frac{\partial X_2}{\partial x_3} \\ \frac{\partial X_3}{\partial x_1} & \frac{\partial X_3}{\partial x_2} & \frac{\partial X_3}{\partial x_3} \end{bmatrix} \quad (2.9)$$



Slika 2.11: Prehod iz nedeformiranega v deformirano stanje [13]

Podobno lahko zapišemo tudi za inverz funkcije:

$$dx = \mathbf{F}^{-1} \cdot dX \quad (2.10)$$

Pomik lahko zapišemo tudi z zvezo:

$$\mathbf{u} = \mathbf{X} - \mathbf{x} \quad (2.11)$$

Sledi zveza, kjer deformacijski gradient zapišemo še s pomočjo pomikov:

$$F_{ij} = \frac{\partial X}{\partial X} + \frac{\partial u}{\partial X} = \delta_{ij} + u_{i,j} \quad (2.12)$$

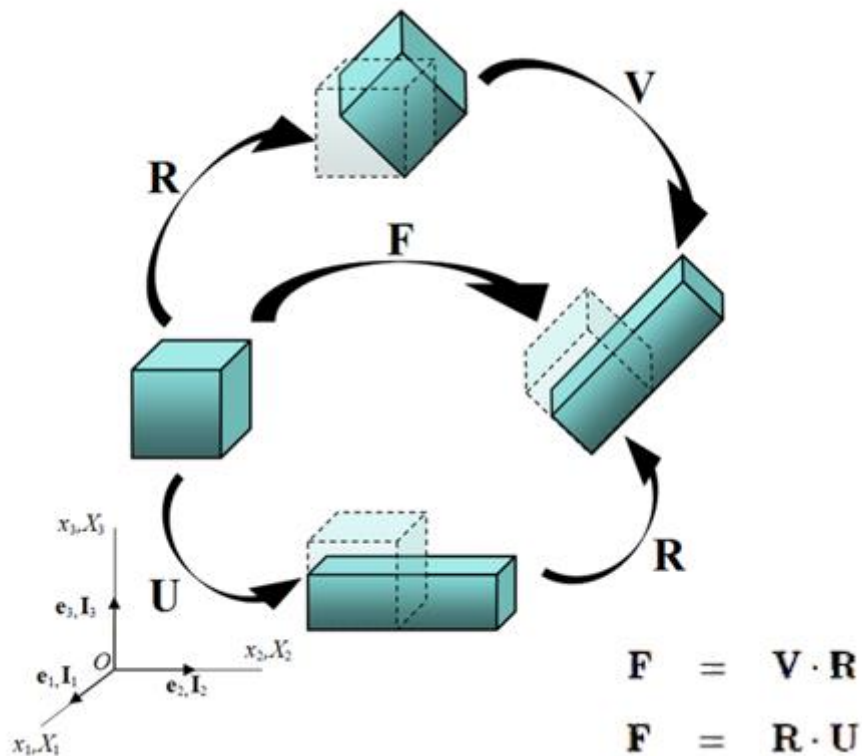
2.7.2 Mere deformacij

Zelo pomembne so tudi mere, s pomočjo katerih zapisujemo deformacije in napetosti, saj je pomembno, da deformacije pravilno opišemo in jih hkrati ločimo od togega gibanja telesa. Želimo si, da deformacije opisujejo majhne ali velike specifične deformacije ter znašajo 0 v primeru, da se je telo zgolj togo premaknilo. Izjemnega pomena je tudi, da sta izbrani meri za napetosti in deformacije med sabo energetsko kompatibilni za računanje. Skozi čas so se razvile številne mere za opis deformacij, kot jih navaja vir [4].

Za izpeljavo nelinearnih končnih elementov je bistvenega pomena Green-Lagrangeev deformacijski tenzor, ki ima sposobnost, da se ob togem pomiku telesa njegova vrednost ne spremeni. Energetsko kompatibilen je z napetostnim tenzorjem 2PK. Kot bo vidno v nadaljevanju te naloge, bomo s pomočjo te dvojice izpeljevali končni element, saj lahko pri izračunu integriramo količine prek znane domene (volumen na začetku inkrementa).

$$E_{ij} = \frac{1}{2} \cdot \left(\frac{\partial u_i}{\partial X_j} + \frac{\partial u_j}{\partial X_i} + \frac{\partial u_k}{\partial X_i} \cdot \frac{\partial u_k}{\partial X_j} \right) = \frac{1}{2} \cdot (\mathbf{F}^T \cdot \mathbf{F} - \mathbf{I}) = \frac{1}{2} \cdot (\mathbf{U}^2 - \mathbf{I}) \quad (2.13)$$

Iz nazadnje zapisane enačbe lahko torej vidimo, da je Green-Lagrangeev tenzor deformacij sestavljen tako, da na njega ne vpliva rotacijska matrika, ki jo izločimo s pomočjo dekompozicije deformacijskega gradienta. Na sliki 2.11 prikazujemo prehod stanja iz nedeformirane lege v deformirano lego na dva različna načina. Prehod je opisan s pomočjo rotacijske matrike in levega (\mathbf{V}) ali desnega raztezka (\mathbf{U}).



Slika 2.12: Polarna dekompozicija deformacijskega gradienta [12]

Mera, ki jo po navadi uporabljamo pri linearnih analizah, v bistvu izhaja iz Green-Lagrangevega opisa z okrajšavo, saj vse člene višjega reda zaradi predpostavke infinitizimalnih deformacij preprosto ne upoštevamo. Gre za tenzor majhnih specifičnih deformacij.

$$\varepsilon_{ij} = \frac{1}{2} \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (2.14)$$

Za primere, ko so zraven velikih pomikov in rotacij prisotne tudi velike deformacije, se za pravilen opis pri posodobljeni Lagrangevi formulaciji za mero deformacij lahko uporabi tudi tenzor dejanskih logaritmičnih deformacij.

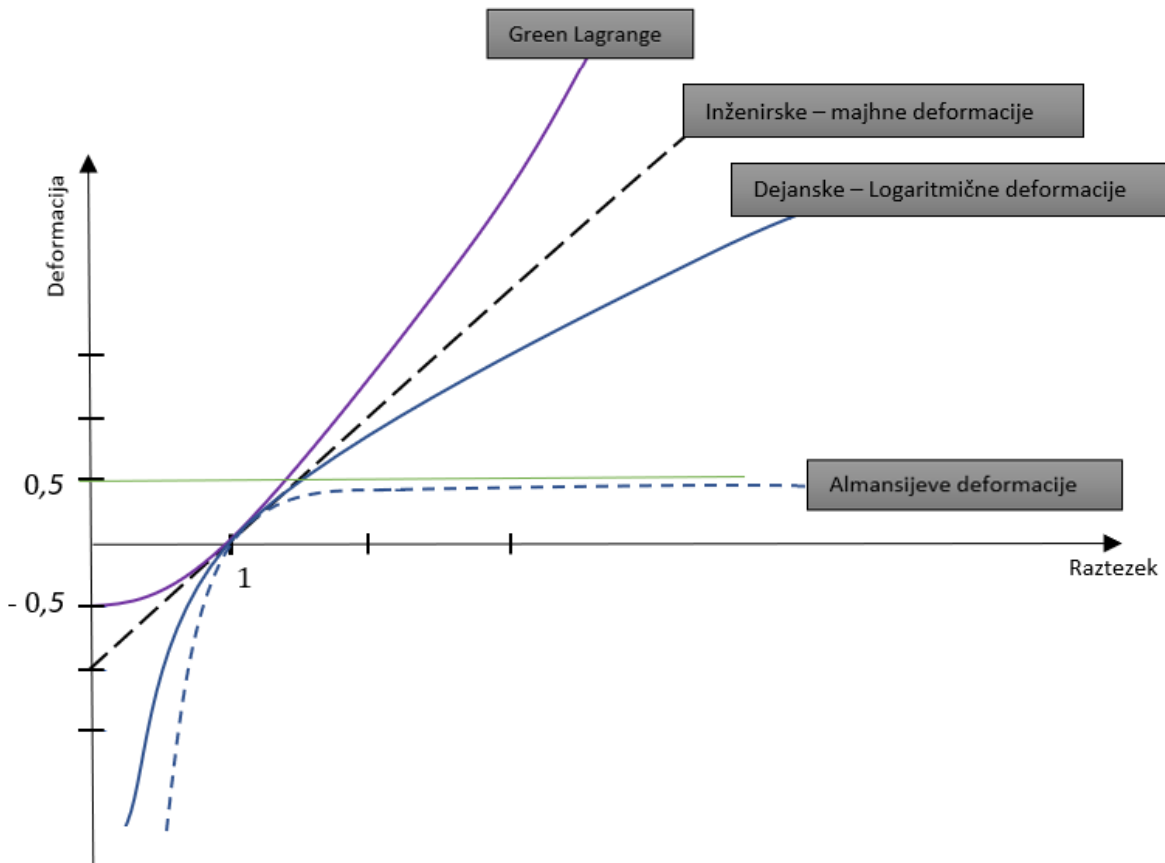
Omenjene deformacije temeljijo na izračunu predhodno predstavljenega levega raztezka, ki ga lahko izračunamo s pomočjo naslednje enačbe:

$$\mathbf{V} = \sqrt{\mathbf{F} \cdot \mathbf{F}^T} \quad (2.15)$$

$$\boldsymbol{\varepsilon}^L = \ln(\mathbf{V}) = \sum_{i=1}^3 \ln \lambda_i n_i n_i^T \quad (2.16)$$

Pri tem so λ_i komponente (glavni raztezki), n_i pa pripadajoče smeri [4].

Obstajajo še številne druge tenzorske količine, ki so se uveljavile predvsem za specifične primere in specifične materialne modele. Naslednja slika prikazuje odstopanja različnih deformacijskih mer in njihove zmogljivosti popisa deformacij v odvisnosti od razteзка.



Slika 2.13: Mere deformacij [13]

2.7.3 Mere napetosti

Podobno kot deformacije lahko tudi napetosti opišemo s pomočjo številnih tenzorskih količin, kjer je bistvenega pomena za izpeljavo nelinearnih KE 2PK (2. Piola-Kirchoffov napetostni tenzor), ki ga lahko zapišemo kot [2]:

$${}^tS_{ij} = \frac{{}^0\rho}{{}^t\rho} \cdot {}^0F_{i,m} \cdot {}^t\sigma_{mn} \cdot {}^0F_{j,n} , \quad (2.17)$$

kjer je ${}^0F_{i,m}$ deformacijski gradient in je enak $\frac{\partial {}^0x_i}{\partial {}^tx_m}$.

V matričnem zapisu velja torej:

$$\mathbf{S} = \mathbf{J} \cdot \mathbf{F}^{-1} \cdot \boldsymbol{\sigma} \cdot \mathbf{F}^{-T} \quad (2.18)$$

Pri tem je treba omeniti, da dejanska fizikalna interpretacija za 2PK ne obstaja, saj gre za količino, ki jo uporabimo kot orodje za izpeljavo KE. V literaturi sicer obstajajo določene izpeljanke in delne interpretacije, zato je treba po koncu iteracij napetosti pretvoriti v Cauchyjev napetostni tenzor, s pomočjo katerega lahko rezultate interpretiramo in jih predstavimo.

$$\boldsymbol{\sigma} = \mathbf{J}^{-1} \cdot \mathbf{F} \cdot \mathbf{S} \cdot \mathbf{F}^T, \quad (2.19)$$

kjer je $J = \mathit{det}(\mathbf{F})$

2.7.4 Numerični pristop reševanja nelinearnih problemov

S pomočjo pregleda literature [13][14] vidimo, da je reševanje nelinearnih problemov možno na več načinov. Najbolj uveljavljene metode so:

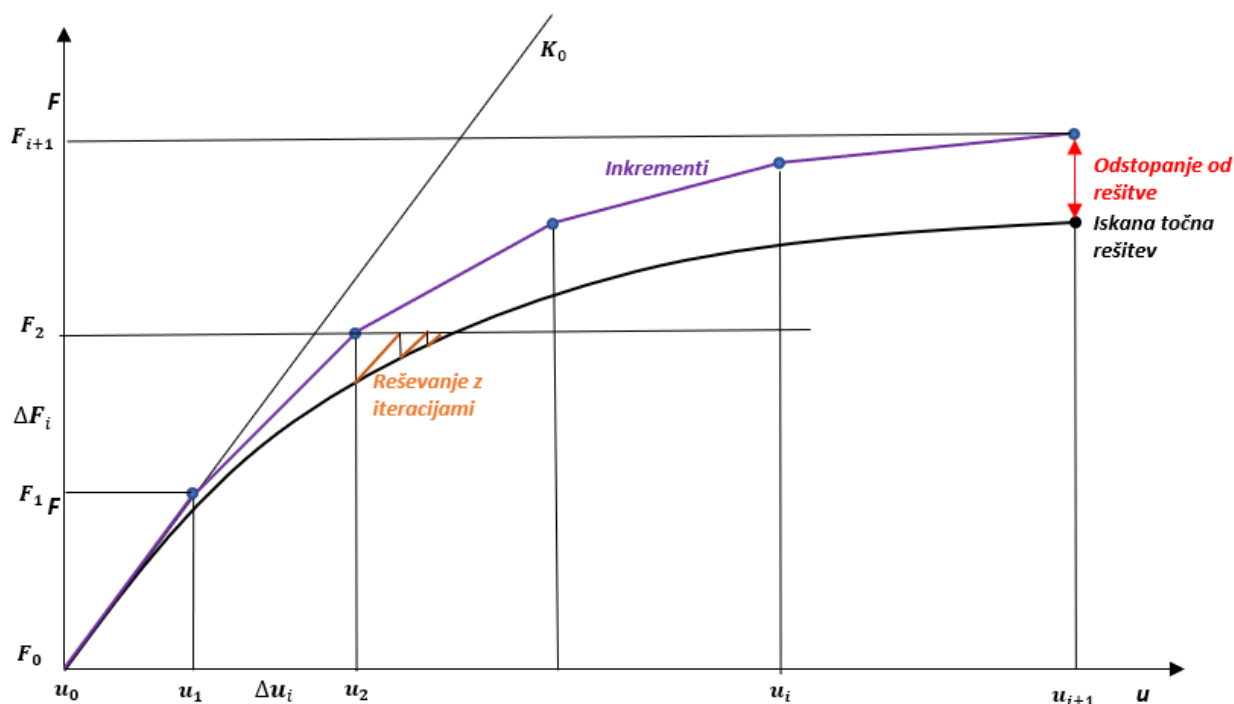
- inkrementne metode,
- iteracijske metode (Newton-Raphsenova ali modificirana Newton-Raphsenova metoda),
- mešane metode.

Pri ponovnem pogledu na enačbo 2.5 lahko vidimo, da gre za togostno matriko, ki je konstantna. Pri nelinearnih analizah pa imamo togostno matriko, odvisno od pomikov KE.

$$\mathbf{F} = \mathbf{K}(\mathbf{U}) \cdot \mathbf{U} \quad (2.20)$$

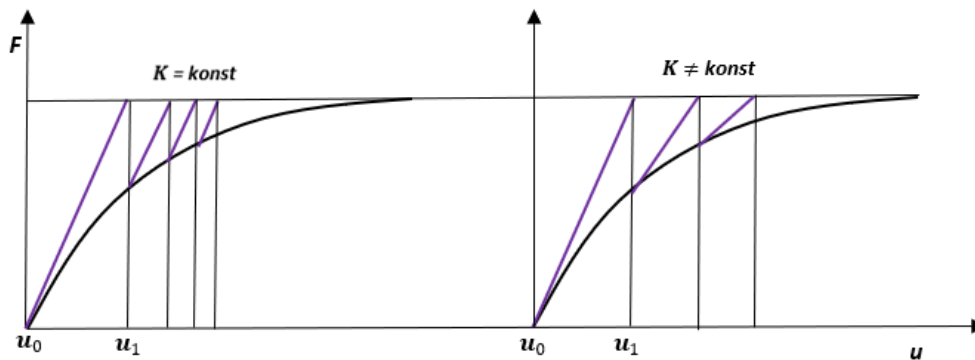
Prvi korak, ki vodi v reševanje s pomočjo inkrementnih metod, je, da silo razdelimo na večje končno število manjših inkrementnih obremenitev. Rezultat takšne enačbe so inkrementni pomiki, ki jih seštevamo. Prednost takšne metode je tudi zmožnost opisa problema, kjer sila ne sledi nosilcu, saj se robni pogoji na koncu vsakega koraka predpišejo v začetni predpisani smeri in s pripadajočo velikostjo inkrementa.

$$\Delta \mathbf{F} = \mathbf{K}(\mathbf{U}) \cdot \Delta \mathbf{U} \quad \rightarrow \quad \Delta \mathbf{U} = \mathbf{K}^{-1}(\mathbf{U}) \cdot \Delta \mathbf{F} \quad (2.21)$$



Slika 2.14: Numerično reševanje nelinearnih enačb

V okviru poteka simulacije, ki je na sliki 2.13 predstavljena z vijolično črto, vidimo, da gre za prikaz inkrementne metode, kjer nastane na koncu pri zadnji točki določen odstop ek od pravilne rešitve. Napako je možno zmanjšati s pomočjo zelo velikega števila inkrementov, kar pa rezultira v zelo dolgotrajni in posledično dragi numerični analizi. Alternativa inkrementom so iteracijske metode, kot je modificirana Newton-Raphsenova metoda (levo na sliki 2.14), kjer vidimo, da skozi vse iteracije uporabljamo enak naklon funkcije ($K=\text{konst}$) tako dolgo, dokler ne dosežemo konvergenčnega kriterija. Ker se naklon funkcije med iteracijami ne spreminja, bomo s to metodo potrebovali večje število iteracij, a je zaradi konstantnega naklona funkcije časovno učinkovitejša. Prednost te metode pa je tudi, da je z vidika doseganja konvergenčnega kriterija bolj robustna in nagnjena h konvergenci. Podobna je tudi metoda, kjer se nakloni funkcije togostne matrike z vsako iteracijo spreminjajo (Newton-Raphsenova metoda), ki pa bo za doseg konvergenčnega kriterija potrebovala manj iteracij, ki bodo zaradi posodabljanja togostne matrike računsko zahtevnejše [14].



Slika 2.15: Modificirana NR-metoda levo, NR-metoda desno

Pri iteracijskih metodah lahko torej kot zunanjo obremenitev uporabimo celotno vrednost zunanje obremenitve, kjer na koncu vsakega koraka zaradi lineariziranega modela nastajajo residualne sile, ki jih je treba izničiti oz. z računskim modelom iterirati, dokler ne izpolnimo konvergenčnega kriterija. V praksi se sicer skoraj obvezno uporabljajo mešane metode, ki zagotovijo optimalni izkoristek računalniške moči in stabilnosti sistema.

$${}^t\mathbf{K} \cdot \Delta \mathbf{U}^i = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{i-1} \quad (2.22)$$

$${}^{t+\Delta t}\mathbf{U}^i = {}^{t+\Delta t}\mathbf{U}^{i-1} + \Delta \mathbf{U}^i \quad (2.23)$$

2.7.5 Variacijski pristop – princip virtualnih pomikov

S pomočjo pregleda literature [1], [16] lahko prikažemo pristop, ki je bistvenega pomena za izpeljavo želene formulacije končnih elementov. Naloga je eksaktno pravilno matematično definirana v »Strong form« izvorni obliki, ki pa je za poljubne robne probleme praktično nerešljiva, saj je sestavljena iz sistema več diferencialnih enačb z dodanimi robnimi pogoji, ki jih razen za npr. enoosno napetostno stanje v praksi ne znamo rešiti.

Alternativo nam torej ponujajo variacijski pristopi s tako imenovano šibko formulacijo »Weak form«, kjer je splošno v literaturi najbolj uveljavljen princip virtualnega dela, ki pravi, da če je telo v statični ravnovesni legi, potem je virtualno delo notranjih sil enako virtualnemu delu zunanjih sil in obratno.

$$\begin{aligned} \text{IVW} &= \text{EVW} \\ \delta W_F &= \delta W_R \end{aligned} \quad (2.24)$$

V enačbo za delo lahko vključimo še virtualne pomike δu (sistem mora veljati za poljubno variacijo), ki so diferencialno majhne spremembe koordinat, pri katerih pa moramo upoštevati predpisane robne pogoje. Pri tem je virtualno delo notranjih sil posledica virtualnih pomikov δu . Izračunamo ga z integracijo zmnožka notranjih napetosti in specifičnih deformacij po celotnem volumnu telesa.

$$W_F = \int_V \delta \boldsymbol{\varepsilon}^T \cdot \boldsymbol{\sigma} \cdot dV \quad (2.25)$$

Nastanek zunanega virtualnega dela lahko teoretično predstavimo z integracijo površinskih sil zgolj po površini, na katero sile delujejo, ter volumskih sil, ki jih integriramo po volumnu telesa. S praktičnega vidika lahko zunanjo obremenitev predstavlja tudi točkovna obremenitev.

$$W_R = \int_{S_t} \delta \mathbf{u}^T \cdot \mathbf{t} \cdot dS_t + \int_V \delta \mathbf{u}^T \cdot \mathbf{p} \cdot dV \quad (2.26)$$

Z enačenjem obeh delov se da pokazati nastanek integralske šibke formulacije ravnotežne enačbe virtualnega dela, ki predstavlja osnovo za vse nadaljnje izpeljave.

$$\int_V \delta \boldsymbol{\varepsilon}^T \cdot \boldsymbol{\sigma} \cdot dV = \delta \mathbf{u}^T \cdot \mathbf{f} + \int_{S_t} \delta \mathbf{u}^T \cdot \mathbf{t} \cdot dS_t + \int_V \delta \mathbf{u}^T \cdot \mathbf{p} \cdot dV \quad (2.27)$$

Pri tem so točkovne obremenitve \mathbf{f} , površinske obremenitve \mathbf{t} ali volumske obremenitve \mathbf{p} .

2.8 Numerična integracija

V algoritmih MKE je najbolj pogosta Gaussova numerična integracija, ki omogoča najbolj natančno numerično integracijo. Postopek temelji na preoblikovanem zapisu integrala (ki ga prevedemo iz osnovnega integrala) s standardnimi integracijskimi mejami $(-1,1)$ [14].

$$\int_{x_0}^{x_n} f(x) dx \quad \rightarrow \quad \int_{-1}^1 f(t) dt \quad (2.28)$$

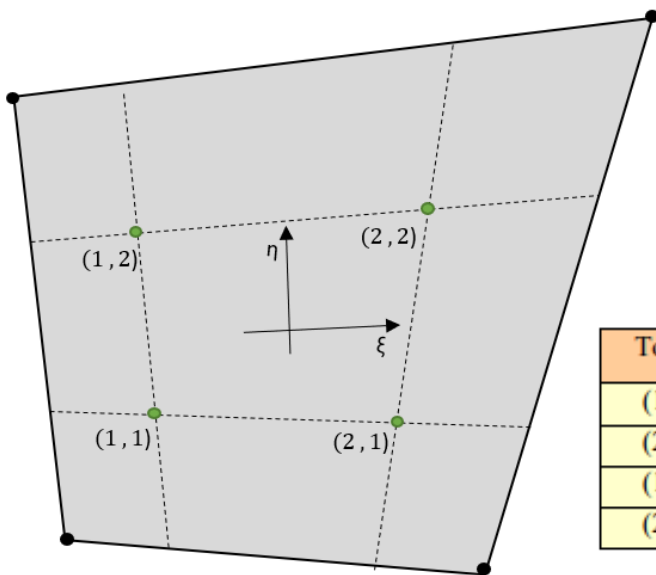
Prevedeni integral je sedaj možno zapisati kot vsoto zmnožkov uteži in funkcij, izračunanih v Gaussovih točkah.

$$\int_{-1}^1 f(\eta) d\eta \approx \sum_{j=1}^N \omega_j \cdot f(\eta_j), \quad (2.29)$$

kjer so vrednosti uteži in Gaussovih točk vnaprej izračunane vrednosti, ki so navedene v številnih učbenikih in literaturi.

V okviru te naloge se bomo ukvarjali s štirivozličnimi KE, kjer bomo dvojni integral za izračun togostne matrike elementa aproksimirali z dvojno vsoto, kjer bodo po elementu skupno 4 integracijske točke, kot je prikazano na naslednji sliki. Integracijo, ki bo uporabljena v nalogi, lahko demonstriramo na primeru, kjer avtorja [1] izpeljeta togostno matriko za ravninski končni element s pomočjo elastičnega potenciala. Izračun togostne matrike elementa pa nato poteka z naslednjo enačbo.

$$\iint_{-1}^1 k_{e1x1x} d\xi d\eta = \sum_{n=1}^N \sum_{m=1}^M w^{(n,m)} \cdot k_{e1x1x}^{(n,m)} \quad (2.30)$$



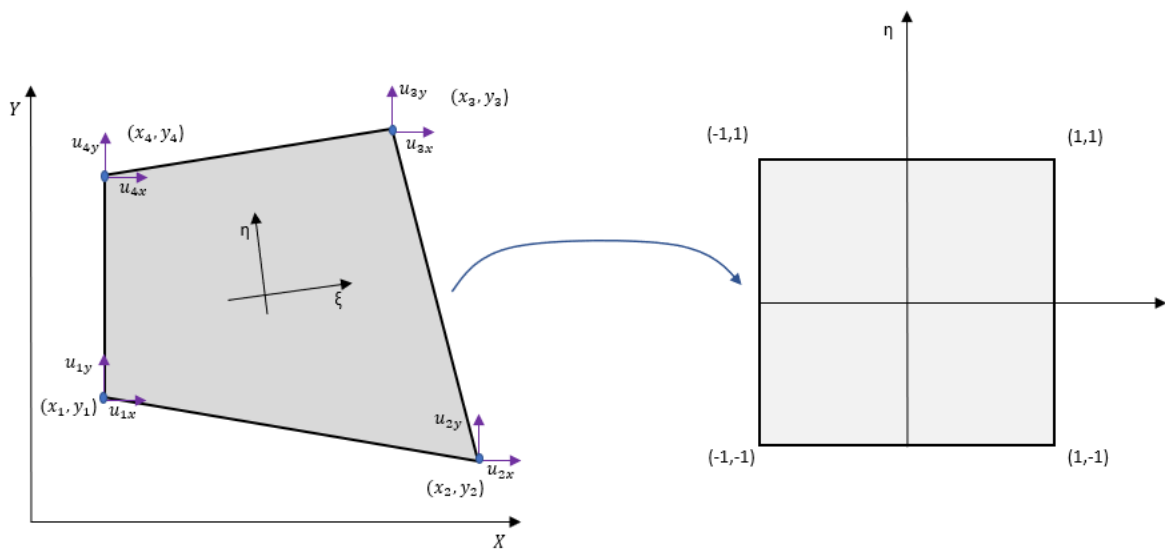
Točka	Utež w	Koordinata ξ	Koordinata η
(1,1)	1.0	-0.57735	-0.57735
(2,1)	1.0	+0.57735	-0.57735
(1,2)	1.0	-0.57735	+0.57735
(2,2)	1.0	+0.57735	+0.57735

Slika 2.16: Gaussova numerična integracija [1]

3 IZPELJAVA KONČNIH ELEMENTOV

3.1 Ravninski linearni končni element

V prvem koraku razvoja kode je bil predmet obravnave linearni ravninski štirivozliščni končni element, ki bo uporabljen za analize brez upoštevanja nelinearnosti. Za obravnavo ravninskih problemov imamo na voljo bogato izbiro končnih elementov, kjer pa se predvsem v zadnjem času vzporedno z napredkom algoritmov za mreženje ostrih robov in zapletenih geometrij zaradi natančnosti izračuna priporoča uporaba štirivozliščnih KE (v primerjavi s trikotnimi elementi, ki imajo sicer sposobnost mreženja zelo zapletenih geometrij). Za izračun vseh potrebnih izrazov za formulacijo MKE veljajo torej predpostavke za linearno analizo. Pri izpeljavi si bomo pomagali z [1], [16].



Slika 3.14: Vozliščni ravninski element

Poznamo formulacije z elementi idealne kvadratne oblike, ki imajo sicer nekoliko poenostavljen zapis izrazov, kar pa rezultira v zelo šibkem opisu poljubne geometrije, zato uporabimo elemente, kjer stranice niso vzporedne, in jih zapišemo s pomočjo preslikave v naravne koordinate, ki imajo pri izračunavanju številne prednosti, saj lahko matematične operacije izvajamo v konstantnem enotskem območju. Tečejo med vrednostjo $(-1,1)$.

Za potrebe zaključne naloge smo izbrali linearne interpolacijske funkcije. Poznamo tudi interpolacijske funkcije višjih redov, ki so natančnejše. Alternativa višanju reda interpolacijskih funkcij pa je lahko tudi preprosto večje število uporabljenih končnih elementov (adaptacija P, H).

$$\begin{aligned}
 N_1 &= \frac{1}{4} \cdot (1 - \xi) \cdot (1 - \eta) \\
 N_2 &= \frac{1}{4} \cdot (1 + \xi) \cdot (1 - \eta) \\
 N_3 &= \frac{1}{4} \cdot (1 + \xi) \cdot (1 + \eta) \\
 N_4 &= \frac{1}{4} \cdot (1 - \xi) \cdot (1 + \eta)
 \end{aligned} \tag{3.1}$$

Z znanimi interpolacijskimi funkcijami pa lahko zapišemo še matriko, ki vsebuje interpolacijske funkcije. Od sedaj naprej bo dodana tudi indikacija s črko **e** (saj gre za enačbe, ki temeljijo zgolj na ravni enega KE).

$$\mathbf{N}_e = \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \tag{3.2}$$

Ker gre za izoparametrični KE (za interpolacijo pomikov in geometrije uporabljamo enake interpolacijske funkcije), lahko v naslednjem koraku zapišemo izraze za izračun poljubne točke in pomika v polju KE.

$$x = \sum_{i=1}^4 N_i \cdot x_i \quad , \quad y = \sum_{i=1}^4 N_i \cdot y_i \tag{3.3}$$

$$u_x = \sum_{i=1}^4 N_i \cdot u_{ix} \quad , \quad u_y = \sum_{i=1}^4 N_i \cdot u_{iy} \tag{3.4}$$

Podobno bi lahko zapisali:

$$\mathbf{u}_e = \mathbf{N}_e \cdot \mathbf{d}_e \tag{3.5}$$

Pri tem lahko definiramo vektor vozliščnih pomikov in diferencialni operator.

$$\mathbf{d}_e = \begin{bmatrix} u_{1x} \\ u_{1y} \\ u_{2x} \\ u_{2y} \\ u_{3x} \\ u_{3y} \\ u_{4x} \\ u_{4y} \end{bmatrix} \quad \mathbf{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \quad (3.6)$$

Pri linearni analizi je kot mera za deformacije izbrana okrajšana oblika Green-Lagrangeevega tenzorja (infinitizimalni deformacijski tenzor), kjer ne upoštevamo členov višjega reda. Deformacije lahko torej zapišemo kot:

$$\varepsilon_{ij} = \frac{1}{2} \cdot \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (3.7)$$

Deformacije zapisujemo v obliki vektorja specifičnih deformacij, ki vsebuje zgolj določene člene tenzorske oblike.

$$\boldsymbol{\varepsilon}_e = \begin{bmatrix} \varepsilon_{xx} \\ \varepsilon_{yy} \\ \varepsilon_{xy} \end{bmatrix}, \quad \boldsymbol{\varepsilon}_e = \begin{bmatrix} \varepsilon_{xx} & \varepsilon_{xy} \\ \varepsilon_{yx} & \varepsilon_{yy} \end{bmatrix} \quad (3.8)$$

Podobno velja tudi za Cauchyjev napetostni tenzor.

$$\boldsymbol{\sigma}_e = \begin{bmatrix} \sigma_{xx} \\ \sigma_{yy} \\ \sigma_{xy} \end{bmatrix}, \quad \boldsymbol{\sigma}_e = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} \\ \sigma_{yx} & \sigma_{yy} \end{bmatrix} \quad (3.9)$$

Deformacije lahko zapišemo še v zvezi, ki jo bomo uporabili kasneje pri izpeljavi.

$$\boldsymbol{\varepsilon}_e = \mathbf{L} \cdot \mathbf{u}_e = \mathbf{L} \cdot \mathbf{N}_e \cdot \mathbf{d}_e \quad (3.10)$$

Vpeljemo novo matriko odvodov interpolacijskih funkcij, ki ji pravimo tudi gradientna matrika elementa.

$$\mathbf{B}_e = \mathbf{L} \cdot \mathbf{N}_e = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \cdot \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \quad (3.11)$$

Iz konstitutivnega zakona poznamo naslednje zveze [17]:

$$\text{Za RNS:} \quad \mathbf{D} = \frac{E}{(1-\nu^2)} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{(1-\nu)}{2} \end{bmatrix} \quad (3.12)$$

$$\text{Za RDS :} \quad \mathbf{D} = \frac{E}{(1+\nu)(1-2\nu)} \begin{bmatrix} (1-\nu) & \nu & 0 \\ \nu & (1-\nu) & 0 \\ 0 & 0 & \frac{(1-2\nu)}{2} \end{bmatrix} \quad (3.13)$$

$$\boldsymbol{\sigma}_e = \mathbf{D} \cdot \boldsymbol{\varepsilon}_e = \mathbf{D} \cdot \mathbf{L} \cdot \mathbf{N}_e \cdot \mathbf{d}_e \quad (3.14)$$

Preden iz virtualnega dela zapišemo enačbo za izračun togostne matrike, je treba še povezati kartezijeve in naravne koordinate, saj je treba zapisane zveze odvajati v pravilni odvisnosti. Vpeljemo torej splošno znano »verižno« pravilo [18].

$$\begin{aligned} \frac{\partial N_i}{\partial \xi} &= \frac{\partial N_i}{\partial x} \cdot \frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y} \cdot \frac{\partial y}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} &= \frac{\partial N_i}{\partial x} \cdot \frac{\partial x}{\partial \eta} + \frac{\partial N_i}{\partial y} \cdot \frac{\partial y}{\partial \eta} \end{aligned} \quad (3.15)$$

Vpeljemo lahko še matrični zapis, kjer ugotovimo, da lahko transformacijo v naravne koordinate in obratno izvajamo s pomočjo Jacobijeve matrike.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \quad (3.16)$$

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \mathbf{J} \cdot \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} \quad (3.17)$$

Podobno sledi za inverz Jacobijeve matrike, kjer izvajamo transformacijo v obratno smer.

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \end{bmatrix} = \mathbf{J}^{-1} \cdot \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} \quad (3.18)$$

Ker smo v elementu že izrazili poljubno točko s pomočjo interpolacijskih funkcij, ki vsebujejo naravne koordinate, lahko v naslednjem koraku zapišemo Jacobijevo matriko s komponentami interpolacijskih funkcij, ki jih znamo brez težav odvajati.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \frac{\partial N_1}{\partial \xi} & \frac{\partial N_2}{\partial \xi} & \frac{\partial N_3}{\partial \xi} & \frac{\partial N_4}{\partial \xi} \\ \frac{\partial N_1}{\partial \eta} & \frac{\partial N_2}{\partial \eta} & \frac{\partial N_3}{\partial \eta} & \frac{\partial N_4}{\partial \eta} \end{bmatrix} \cdot \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} =$$

$$= \begin{bmatrix} \frac{(-1+\eta)}{4} & \frac{(1-\eta)}{4} & \frac{(1+\eta)}{4} & \frac{(-1-\eta)}{4} \\ \frac{(-1+\xi)}{4} & \frac{(-1-\xi)}{4} & \frac{(1+\xi)}{4} & \frac{(1-\xi)}{4} \end{bmatrix} \cdot \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ x_3 & y_3 \\ x_4 & y_4 \end{bmatrix} \quad (3.19)$$

Sledi še zapis gradientne matrike z odvodi po naravnih koordinatah, kjer so v izraz že vključeni členi inverzne Jacobijeve matrike.

$$\mathbf{B}_e = \frac{1}{|\mathbb{J}|} \cdot \begin{bmatrix} J_{22} \cdot \frac{\partial}{\partial \xi} - J_{12} \cdot \frac{\partial}{\partial \eta} & 0 \\ 0 & J_{11} \cdot \frac{\partial}{\partial \eta} - J_{21} \cdot \frac{\partial}{\partial \xi} \\ J_{11} \cdot \frac{\partial}{\partial \eta} - J_{21} \cdot \frac{\partial}{\partial \xi} & J_{22} \cdot \frac{\partial}{\partial \xi} - J_{12} \cdot \frac{\partial}{\partial \eta} \end{bmatrix} \begin{bmatrix} N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 & 0 \\ 0 & N_1 & 0 & N_2 & 0 & N_3 & 0 & N_4 \end{bmatrix} \quad (3.20)$$

Znane količine in zveze lahko vstavimo v osnovno enačbo principa virtualnega dela, ki smo jo izpeljali v prejšnjem poglavju.

$$\int_V \delta \boldsymbol{\varepsilon}^T \cdot \boldsymbol{\sigma} \cdot dV = \delta \mathbf{u}^T \cdot \mathbf{f} + \int_S \delta \mathbf{u}^T \cdot \mathbf{t} \cdot dS + \int_V \delta \mathbf{u}^T \cdot \mathbf{p} \cdot dV \quad (3.21)$$

Vstavimo zvezo deformacij s pomočjo diferencialnega operatorja.

$$\int_V \delta \mathbf{u}^T \cdot \mathbf{L}^T \cdot \boldsymbol{\sigma} \cdot dV = \delta \mathbf{u}^T \cdot \mathbf{f} + \int_S \delta \mathbf{u}^T \cdot \mathbf{t} \cdot dS + \int_V \delta \mathbf{u}^T \cdot \mathbf{p} \cdot dV \quad (3.22)$$

Sledi korak, kjer nadomestimo virtualne pomike s pomočjo matrike interpolacijskih funkcij ter vozliščnih pomikov.

$$\int_V \delta \mathbf{d}_e^T \cdot \mathbf{N}_e^T \cdot \mathbf{L}^T \cdot \boldsymbol{\sigma} \cdot dV = \delta \mathbf{d}_e^T \cdot \mathbf{N}_e^T \cdot \mathbf{f} + \int_S \delta \mathbf{d}_e^T \cdot \mathbf{N}_e^T \cdot \mathbf{t} \cdot dS + \int_V \delta \mathbf{d}_e^T \cdot \mathbf{N}_e^T \cdot \mathbf{p} \cdot dV \quad (3.23)$$

Opazimo lahko, da je virtualni pomik vozliščnih vrednosti spremenljivka, ki ni odvisna od integrala, zato lahko enačbo skrajšamo.

$$\int_V \mathbf{N}_e^T \cdot \mathbf{L}^T \cdot \boldsymbol{\sigma} \cdot dV = \mathbf{N}_e^T \cdot \mathbf{f} + \int_S \mathbf{N}_e^T \cdot \mathbf{t} \cdot dS + \int_V \mathbf{N}_e^T \cdot \mathbf{p} \cdot dV \quad (3.24)$$

Levi del enačbe lahko s prej izpeljano in zapisano konstitutivno zvezo z upoštevanjem naslednjih zvez v nekaj korakih zapišemo kot enačbo za izračun togostne matrike elementa.

$$\mathbf{B}_e^T = \mathbf{N}_e^T \cdot \mathbf{L}^T$$

Desna stran enačaja prikazuje zunanje obremenitve končnega elementa.

$$\mathbf{K}_e = \int_V \mathbf{B}_e^T \cdot \mathbf{D}_e \cdot \mathbf{B}_e \, dV = \int_A \mathbf{B}_e^T \cdot \mathbf{D}_e \cdot \mathbf{B}_e \cdot \mathbf{t} \cdot dA \quad (3.25)$$

$$\mathbf{K}_e = \int_{-1}^1 \int_{-1}^1 \mathbf{B}_e^T \cdot \mathbf{D}_e \cdot \mathbf{B}_e \cdot \mathbf{t} \cdot |J| \cdot d\xi d\eta \quad (3.26)$$

Do sedaj je bilo govora o izpeljavi posameznih KE, ki pa jih je za obravnavo dejanskega problema treba sestaviti v globalne matrike.

$$\mathbf{F}_e = \mathbf{K}_e \cdot \mathbf{d}_e \quad \rightarrow \quad \mathbf{F} = \mathbf{K} \cdot \mathbf{d} \quad (3.27)$$

Togostno matriko sestavljamo po posebnem postopku, kjer najprej določimo število prostostnih stopenj sistema:

$$N_S = N_{PS} \cdot N_V, \quad (3.28)$$

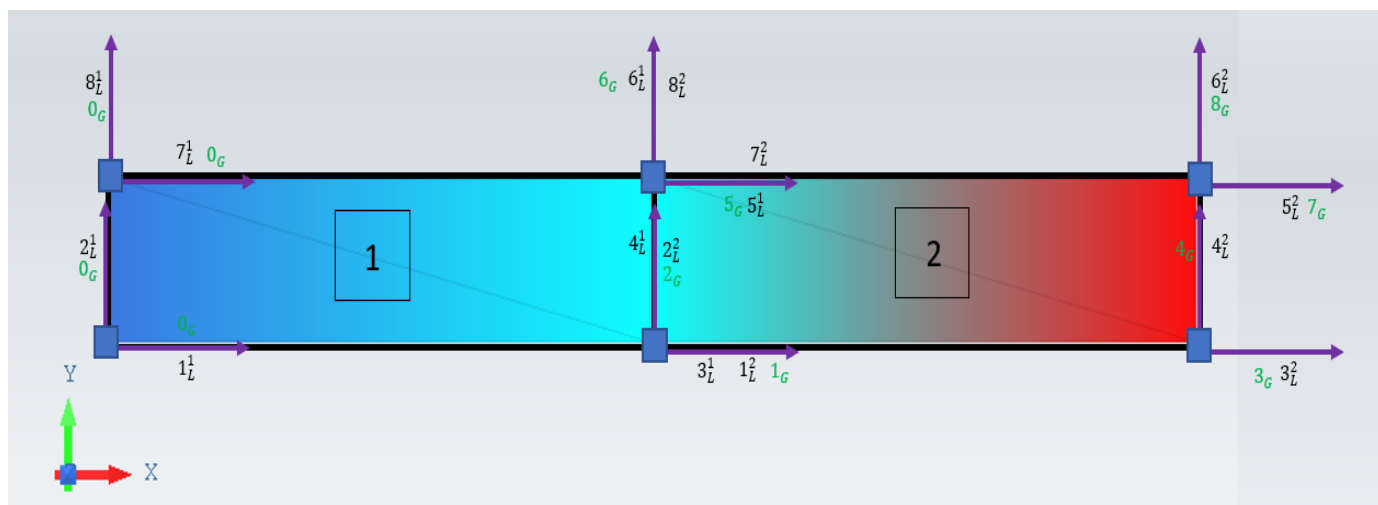
pri tem so:

N_S – celotno število neznank,

N_{PS} – število prostostnih stopenj vozlišča,

N_V – število vozlišč v sistemu.

Pri tem torej nastane togostna matrika velikosti $[N_S \times N_S]$ s pripadajočimi vektorji pomikov in obremenitev velikosti $[N_S]$. Sestava temelji na globalni togostni matriki, ki vsebuje člene globalnih prostostnih stopenj. To pomeni, da se v lokalnih stikajočih se vozliščih, ki imajo skupne lokalne prostostne stopnje, členi togostne matrike seštevajo. Naslednja zelo dobra lastnost je, da lahko v primeru, ko je katera od prostostnih stopenj v vozlišču blokirana, (vpetje) vrstico in stolpec s pripadajočimi členi preprosto zanemarimo. Sestavljanje togostne matrike lahko prikažemo na sistemu, ki je sestavljen iz dveh štirivozličnih KE. V vsakem vozlišču imamo 2 prostostni stopnji, kot prikazuje naslednja slika.



Slika 3.2: Sistem 2 KE

Z zeleno barvo so označene globalne prostostne stopnje, kjer lahko vidimo, da je levi konec nosilca togo vpet, saj so vse 4 prostostne stopnje enake 0. Elementa lahko označimo še z lokalnimi prostostnimi stopnjami, ki so prikazane s črno barvo. Sistemska togostna matrika bo torej velikosti $[8 \times 8]$, njeni členi pa so v naslednji tabeli osenčeni z rumeno barvo. Vidimo lahko, da so členi, označeni s sivo barvo, iz sistemske matrike preprosto izločeni, saj je tam element vpet in bodo pomiki znašali 0 mm. Z vijolično barvo so prikazani členi vozlišč, v katerih k skupni togosti prispevata oba elementa. V programu je za to razvit algoritem, ki avtomatsko zagotavlja ustrezno številčenje prostostnih stopenj in sestavljanje togostne matrike.

Tabela 1: Členi sistemske togostne matrike

	0G	0G	1G	2G	3G	4G	0G	0G	5G	6G	7G	8G
0G	K_{11}^1	K_{12}^1	K_{13}^1	K_{14}^1			K_{17}^1	K_{18}^1	K_{15}^1	K_{16}^1		
0G	K_{21}^1	K_{22}^1	K_{23}^1	K_{24}^1			K_{27}^1	K_{28}^1	K_{25}^1	K_{26}^1		
1G	K_{31}^1	K_{32}^1	$K_{33}^1+K_{11}^2$	$K_{34}^1+K_{12}^2$	K_{13}^2	K_{14}^2	K_{37}^1	K_{38}^1	$K_{35}^1+K_{17}^2$	$K_{36}^1+K_{18}^2$	K_{15}^2	K_{16}^2
2G	K_{41}^1	K_{42}^1	$K_{43}^1+K_{21}^2$	$K_{44}^1+K_{22}^2$	K_{23}^2	K_{24}^2	K_{47}^1	K_{48}^1	$K_{45}^1+K_{27}^2$	$K_{46}^1+K_{28}^2$	K_{25}^2	K_{26}^2
3G			K_{31}^2	K_{32}^2	K_{33}^2	K_{34}^2			K_{37}^2	K_{38}^2	K_{35}^2	K_{36}^2
4G			K_{41}^2	K_{42}^2	K_{43}^2	K_{44}^2			K_{47}^2	K_{48}^2	K_{45}^2	K_{46}^2
0G	K_{71}^1	K_{72}^1	K_{73}^1	K_{74}^1			K_{77}^1	K_{78}^1	K_{75}^1	K_{76}^1		
0G	K_{81}^1	K_{82}^1	K_{83}^1	K_{84}^1			K_{87}^1	K_{88}^1	K_{85}^1	K_{86}^1		
5G	K_{51}^1	K_{52}^1	$K_{53}^1+K_{71}^2$	$K_{54}^1+K_{72}^2$	K_{73}^2	K_{74}^2	K_{57}^1	K_{58}^1	$K_{55}^1+K_{77}^2$	$K_{56}^1+K_{78}^2$	K_{75}^2	K_{76}^2
6G	K_{61}^1	K_{62}^1	$K_{63}^1+K_{81}^2$	$K_{64}^1+K_{82}^2$	K_{83}^2	K_{84}^2	K_{67}^1	K_{68}^1	$K_{65}^1+K_{87}^2$	$K_{66}^1+K_{88}^2$	K_{85}^2	K_{86}^2
7G			K_{51}^2	K_{52}^2	K_{53}^2	K_{54}^2			K_{57}^2	K_{58}^2	K_{55}^2	K_{56}^2
8G			K_{61}^2	K_{62}^2	K_{63}^2	K_{64}^2			K_{67}^2	K_{68}^2	K_{65}^2	K_{66}^2

3.2 Ravninski nelinearni končni element

Sledi izpeljava končnih elementov, ki bodo sposobni popisati tudi predhodno opisane nelinearnosti. Pri tem lahko izhajamo iz principa o minimumu potencialne energije, kot je opisano v knjigi [19]. Druga možnost, ki je v literaturi najbolj razširjena, je izpeljava, ki temelji na principu virtualnih pomikov, ki je bila v prejšnjem poglavju tudi predstavljena. Posebno je treba poudariti, da bo v celotnem podpoglavju izpeljava temeljila na pojasnjenih zvezah knjige avtorja K. J. Bathe, [2], [7], ki je obravnavano tematiko predstavljal tudi v okviru predavanj »MITOPENCOURSEWARE«. Zelo pomembno je tudi, da na tej točki nakažemo uporabo indiciranja in notiranja tenzorskih količin, ki je podrobneje opisana na

začetku tega dokumenta. Izhajajmo torej iz osnovne enačbe PVD, ki je glede na Lagrangeev opis gibanja zapisana z integracijo v znanem časovnem trenutku. Pri uporabi linearnih elementov smo torej uporabljali Cauchyjev napetostni tenzor in infinitizimalni tenzor deformacij, ki pa temeljita na integraciji prek znane domene. Pri tem smo predpostavljali, da se geometrija ne spremeni bistveno.

$$\int_{t_V} {}^t\sigma_{ij} \cdot \delta_t \varepsilon_{ij} \cdot {}^t dV = \int_{t_S} {}^t f_i^S \cdot \delta u_i^S \cdot {}^t dS + \int_{t_V} {}^t f_i^B \cdot \delta u_i \cdot {}^t dV \quad (3.29)$$

PVD v teoriji pravi, da je notranje virtualno delo enako zunanemu virtualnemu delu, kar lahko prevedemo, kot da mora biti leva stran pred enačajem enaka desni strani, ki v prvem integralu predstavlja seštevanje površinskih sil, pomnoženih z virtualnim pomikom, ter podobno v drugem integralu, kjer integriramo prispevke volumskih sil prek znanega volumna ob času t . Leva stran enačaja predstavlja integracijo zmnožka Cauchyjevega napetostnega tenzorja in virtualnih deformacij, ki so posledica virtualnih pomikov. Enačba IVW = EVW mora veljati za vsako variacijo virtualnih pomikov in deformacij, ki zadostijo robnim pogojem.

Bistveno se je zavedati dejstva, da se pri predstavljenih nelinearnostih domena, prek katere integriramo, nezanemarljivo spremeni, zato je naš glavni cilj zapisati enačbo, ki bo zapisano upoštevala.

Če predpostavljamo, da poznamo rešitve ob času t in želimo zapisati rešitev ob času $t + \Delta t$, lahko zapišemo enačbo, ki postavi v ravnovesje notranje in zunanje obremenitve za čas $t + \Delta t$.

$${}^{t+\Delta t} \mathbf{F} = {}^{t+\Delta t} \mathbf{R} \quad (3.30)$$

Ker notranjih sil ob času $t + \Delta t$ ne poznamo, lahko vpeljemo nov inkrement, s pomočjo katerega zapišemo neznane sile s pomočjo sil predhodnega koraka.

$${}^{t+\Delta t} \mathbf{F} = {}^t \mathbf{F} + {}^t \mathbf{K} \cdot \Delta \mathbf{U} \quad (3.31)$$

Cilj tega podpoglavja je torej linearizacija enačbe PVD, ki vsebuje veliko število nelinearnosti, v naslednjo inkrementno obliko:

$${}^t \mathbf{K} \cdot \Delta \mathbf{U} = {}^{t+\Delta t} \mathbf{R} - {}^t \mathbf{F} \quad (3.32)$$

$${}^{t+\Delta t}\mathbf{U} \approx {}^t\mathbf{U} + \Delta\mathbf{U} \quad (3.33)$$

Splošno lahko do rešitve pridemo tudi s pomočjo iterativnega reševanja, kjer v vsakem časovnem koraku izračunavamo razliko med zunanji in notranji silami iz pretekle iteracije in zahtevamo vzpostavljeno ravnovesje. Vse to je možno ob pravilnem izračunavanju notranjih sil po koncu iteracije.

$${}^t\mathbf{K} \cdot \Delta\mathbf{U}^i = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{i-1} \quad (3.34)$$

Enačbo 3.29 lahko sedaj zapišemo še v času $t + \Delta t$. Enačba vsebuje povsem enake člene IVW, kot tudi EVW. Razlika je torej le v tem, da tokrat integriramo po domeni ob času $t + \Delta t$, ki pa je ne poznamo. Prav tako v neznani domeni obravnavamo napetosti, ki so fizično sila na enoto površine v neznani domeni.

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}\sigma_{ij} \cdot \delta {}^t\varepsilon_{ij} \cdot {}^{t+\Delta t}dV = \int_{{}^{t+\Delta t}S} {}^{t+\Delta t}f_i^S \cdot \delta u_i^S \cdot {}^{t+\Delta t}dS + \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}f_i^B \cdot \delta u_i \cdot {}^{t+\Delta t}dV \quad (3.35)$$

Problem lahko rešujemo s pomočjo predhodno predstavljenih količin, 2PK napetostnim tenzorjem ${}^t\mathbf{S}$ in Green-Lagrangeevim deformacijskim tenzorjem ${}^t\mathbf{E}$, ki sta energetske kompatibilna. Velja torej naslednja zveza:

$${}^t\sigma_{ij} \cdot \delta {}^t\varepsilon_{ij} \rightarrow \frac{\text{Virtualno delo ob času } t}{\text{enota trenutnega volumna}}$$

$${}^tS_{ij} \cdot \delta {}^tE_{ij} \rightarrow \frac{\text{Virtualno delo ob času } t}{\text{enota začetnega volumna}}$$

$$\int_{{}^tV} {}^t\sigma_{ij} \cdot \delta {}^t\varepsilon_{ij} \cdot {}^tdV = \int_{{}^tV} {}^tS_{ij} \cdot \delta {}^tE_{ij} \cdot {}^tdV \quad (3.36)$$

Prišli smo do končnega nastavka za razvoj enačb KE, s tem, da predvidimo, da poznamo odziv ob času t , s čimer lahko zapišemo:

$$\int_{{}^{t+\Delta t}V} {}^{t+\Delta t}S_{ij} \cdot \delta {}^{t+\Delta t}E_{ij} \cdot {}^{t+\Delta t}dV = \int_{{}^{t+\Delta t}S} {}^{t+\Delta t}f_i^S \cdot \delta u_i^S \cdot {}^{t+\Delta t}dS + \int_{{}^{t+\Delta t}V} {}^{t+\Delta t}f_i^B \cdot \delta u_i \cdot {}^{t+\Delta t}dV \quad (3.37)$$

Tenzor napetosti in deformacije lahko razstavimo na inkrementa, ki se nanašata na znano začetno domeno ob času 0.

$${}^{t+\Delta t}S_{ij} = {}^tS_{ij} + {}_0S_{ij} \quad (3.38)$$

$${}^{t+\Delta t}{}_0E_{ij} = {}^t{}_0E_{ij} + {}_0E_{ij}$$

V primeru posodobljene Lagrangeeve formulacije bomo vse količine definirali glede na čas t (predvidevamo, da ob tem času poznamo vse količine). Tudi v tem primeru lahko mere napetosti in deformacij razstavimo na inkremente in zapišemo.

$$\begin{aligned} {}^{t+\Delta t}{}_tS_{ij} &= {}^t{}_tS_{ij} + {}_tS_{ij} = {}^t\sigma_{ij} + {}_tS_{ij} \\ {}^{t+\Delta t}{}_tE_{ij} &= {}^t{}_tE_{ij} + {}_tE_{ij} = {}_tE_{ij} \end{aligned} \quad (3.39)$$

Pri tem vemo, da znane napetosti ob času t prestavlja zgolj Cauchyjev tenzor napetosti, medtem ko lahko pri zapisu deformacij znani inkrement do časa t preprosto zavržemo, saj nas zanima le čas od t do $t + \Delta t$.

3.2.1 Celotna Lagrangeeva formulacija

Če ponovno pogledamo sliko 2.10, lahko vidimo, da se napetosti in deformacije ob času $t + \Delta t$ nanašajo na začetno konfiguracijo ob času 0 s pomočjo novo vpeljanih količin 2PK in Green-Lagrangeevega tenzorja deformacij. Kot je bilo omenjeno, bomo torej uporabljali inkrementne metode za iskanje rešitev, zato lahko enačbo 2.8 preoblikujemo in zapišemo naslednjo zvezo, v kateri predstavimo skupni pomik kot seštevek znanega dela in neznanega inkrementa.

$${}^{t+\Delta t}u_i = {}^t u_i + u_i \quad (3.40)$$

Nastavek in osnovna enačba, iz katere izhajamo, sta bila predstavljena v enačbi (3.37). Nadaljujemo lahko z zapisom Green-Lagrangeevih deformacij v elementarni obliki odvodov pomikov ob času t in $t + \Delta t$.

$$\begin{aligned} {}^tE_{ij} &= \frac{1}{2} \cdot ({}^t{}_0u_{i,j} + {}^t{}_0u_{j,i} + {}^t{}_0u_{k,i} \cdot {}^t{}_0u_{k,j}) \\ {}^{t+\Delta t}{}_0E_{ij} &= \frac{1}{2} \cdot ({}^{t+\Delta t}{}_0u_{i,j} + {}^{t+\Delta t}{}_0u_{j,i} + {}^{t+\Delta t}{}_0u_{k,i} \cdot {}^{t+\Delta t}{}_0u_{k,j}) \end{aligned} \quad (3.41)$$

Če uporabimo enačbo (3.38), lahko zapisana člena med sabo odštejemo in zapišemo neznani inkrement:

$${}_0E_{ij} = \frac{1}{2} \cdot ({}_0u_{i,j} + {}_0u_{j,i} + {}_0^t u_{k,i} \cdot {}_0^t u_{k,j} + {}_0u_{k,i} \cdot {}_0^t u_{k,j}) + \frac{1}{2} \cdot ({}_0u_{k,i} \cdot {}_0u_{k,j}) \quad (3.42)$$

Če pobližje pogledamo enačbo, lahko vidimo, da je člen $\frac{1}{2} \cdot ({}_0^t u_{k,i} \cdot {}_0^t u_{k,j} + {}_0u_{k,i} \cdot {}_0^t u_{k,j})$ posledica učinka začetnih pomikov (če gre za prvo iteracijo, je ta člen enak 0), ki pa jih poznamo. Zato gre torej za linearni člen, kljub temu da se zaradi produktov zdi nasprotno. Edini nelinearni del enačbe, ki nastopi, je torej $\frac{1}{2} \cdot ({}_0u_{k,i} \cdot {}_0u_{k,j})$.

Uvedemo lahko linearni in nelinearni člen deformacij, za katera veljajo naslednje zveze:

$${}_0E_{ij} = {}_0\varepsilon_{ij} + {}_0\eta_{ij} \quad (3.43)$$

$${}_0\varepsilon_{ij} = \frac{1}{2} \cdot ({}_0u_{i,j} + {}_0u_{j,i} + {}_0^t u_{k,i} \cdot {}_0^t u_{k,j} + {}_0u_{k,i} \cdot {}_0^t u_{k,j}) \quad (3.44)$$

$${}_0\eta_{ij} = \frac{1}{2} \cdot ({}_0u_{k,i} \cdot {}_0u_{k,j}) \quad (3.45)$$

Knjiga [2] matematično pojasni zvezo $\delta^{t+\Delta t} {}_0E_{ij} = \delta {}_0^t E_{ij}$, ki pravi, da je variacija deformacij ob času $t + \Delta t$ enaka variaciji inkrementa.

Zapišemo lahko še variacijo enačbe deformacij in jo z vsemi pravkar izpeljanimi enačbami vstavimo v osnovni nastavek PVD. Od te točke naprej prav tako vpeljemo člen ${}^{t+\Delta t}R$, ki predstavlja zunanje virtualno EVW. Pri tem smo privzeli, da obremenitve niso odvisne od deformacij.

$$\delta {}_0E_{ij} = \delta {}_0\varepsilon_{ij} + \delta {}_0\eta_{ij} \quad (3.46)$$

$$\int_{0V} {}_0S_{ij} \cdot \delta {}_0E_{ij} \cdot {}_0dV + \int_{0V} {}_0^t S_{ij} \cdot \delta {}_0\eta_{ij} \cdot {}_0dV = {}^{t+\Delta t}R - \int_{0V} {}_0^t S_{ij} \cdot \delta {}_0\varepsilon_{ij} \cdot {}_0dV \quad (3.47)$$

Zapisana enačba vsebuje nekatere nelinearnosti, ki predstavljajo največjo težavo pri reševanju, zato je glavni cilj naslednjih nekaj korakov linearizacija vseh členov višjih redov, za kar pa je potrebna bolj podrobna analiza posameznih segmentov.

- Prvi segment enačbe:

$$\int_{0V} {}_0^t S_{ij} \cdot \delta {}_0\eta_{ij} \cdot {}_0dV$$

${}^t_0S_{ij}$ ob času t poznamo, zato je člen zagotovo linearen glede na u_i , medtem ko lahko za člen $\delta_0\eta_{ij}$ razpišemo variacijo in ugotovimo, da so členi $\delta_0u_{k,j}$ in $\delta_0u_{k,i}$ variacije, ki so konstantne, zato lahko celotni člen pustimo nedotaknjen.

$$\delta_0\eta_{ij} = \frac{1}{2} \cdot ({}_0u_{k,i} \cdot \delta_0u_{k,j}) + \frac{1}{2} \cdot (\delta_0u_{k,i} \cdot {}_0u_{k,j}) \quad (3.48)$$

- Drugi segment enačbe:

$$\int_{0V} {}_0S_{ij} \cdot \delta_0E_{ij} \cdot {}^0dV$$

Člen δ_0E_{ij} je glede na enačbo (3.39) sestavljen iz variacije linearnega in nelinearnega člena. Pri prejšnjem segmentu enačbe smo pokazali, da variacija nelinearnega člena deformacije vsebuje zgolj linearne člene u_i . Preostane nam le člen ${}_0S_{ij}$, ki ga lahko razvijemo v Taylorjevo vrsto ali pa preprosto po vzgledu [10] zapišemo naslednjo zvezo:

$$\begin{aligned} {}^t_0S_{ij} = {}_0C_{ijrs} \cdot {}_0E_{rs} \quad , \quad \delta_0E_{rs} \approx \delta_0\varepsilon_{rs} \\ {}^t_0S_{ij} = {}_0C_{ijrs} \cdot ({}_0\varepsilon_{rs} + {}_0\eta_{rs}) \rightarrow {}^t_0S_{ij} \approx {}_0C_{ijrs} \cdot ({}_0\varepsilon_{rs}) \end{aligned} \quad (3.49)$$

Enačba je namreč glede na u_i nelinearna, ampak lahko glede na to, da bomo reševali metodo s pomočjo inkrementov, razmišljamo v smeri, da bodo inkrementi majhni. Posledično bodo tudi pomiki in deformacije majhni. Z naslednjo enačbo lahko torej zapišemo linearizirani segment, ki vsebuje še zadnji produkt, ki bi povzročil višji red člena u_i .

$${}_0S_{ij} \cdot \delta_0E_{ij} \approx {}_0C_{ijrs} \cdot ({}_0\varepsilon_{rs}) \cdot \delta_0E_{ij} \approx {}_0C_{ijrs} \cdot ({}_0\varepsilon_{rs}) \cdot \delta({}_0\varepsilon_{ij} + {}_0\eta_{ij}) \quad (3.50)$$

Produkt $({}_0\varepsilon_{rs} \cdot \delta_0\eta_{ij})$ torej zanemarimo in zapišemo končno obliko.

$${}_0S_{ij} \cdot \delta_0E_{ij} \approx {}_0C_{ijrs} \cdot ({}_0\varepsilon_{rs}) \cdot \delta({}_0\varepsilon_{ij}) \quad (3.51)$$

- Tretji segment enačbe:

$$\int_{0V} {}^t_0S_{ij} \cdot \delta_0\varepsilon_{ij} \cdot {}^0dV$$

Ča si pobližje pogledamo še zadnji segment, ki nam je ostal, vidimo, da ni nobenih nelinearnosti, saj člen ${}^t_0S_{ij}$ ob času t poznamo, zato je člen zagotovo ponovno linearen glede na u_i . Ugotovimo pa lahko naslednjo zelo pomembno zvezo:

$$\int_{0V} {}^t_0S_{ij} \cdot \delta_0\varepsilon_{ij} \cdot {}^0dV = \int_{0V} {}^t_0S_{ij} \cdot \delta_0E_{ij} \cdot {}^0dV \quad (3.52)$$

Matematično jo lahko pojasnimo s predpostavko, da je vrednost u_i enaka ob času $t + \Delta t$ in času t . S pomočjo enačbe (3.43), kjer smo razstavili celotno deformacijo na linearni in nelinearni del, lahko torej pri $u_i = 0$ ugotovimo, da je nelinearni del deformacij enak 0.

Sledi zapis končne linearizirane enačbe PVD.

$$\int_{0V} {}_0C_{ijrs} \cdot ({}_0\varepsilon_{rs}) \cdot \delta({}_0\varepsilon_{ij}) {}^0dV + \int_{0V} {}^t_0S_{ij} \cdot \delta_0\eta_{ij} \cdot {}^0dV = {}^{t+\Delta t}R - \int_{0V} {}^t_0S_{ij} \cdot \delta_0\varepsilon_{ij} \cdot {}^0dV \quad (3.53)$$

Če enačbo PVD diskretiziramo, dobimo podobno zvezo, kot smo jo zapisali pri zapisu linearnega končnega elementa z nekaterimi razlikami. Leva stran enačaja rezultira v tangentni togostni matriki, ki vsebuje konstitutivno matriko in trenutne napetosti. Obe strani sta prav tako pomnoženi s transponiranim vektorjem virtualnih pomikov, medtem ko desna stran enačaja predstavlja residual med notranjimi in zunanji silami.

$$\begin{aligned} \delta U^T \cdot {}^t_0\mathbf{K} \cdot \Delta U &= \delta U^T \cdot ({}^{t+\Delta t}\mathbf{R} - {}^t_0\mathbf{F}) \\ {}^t_0\mathbf{K} \cdot \Delta U &= {}^{t+\Delta t}\mathbf{R} - {}^t_0\mathbf{F} \end{aligned} \quad (3.54)$$

S tem smo izpeljali enačbo za reševanje po metodi končnih elementov s pomočjo inkrementov pomikov, ki jih na koncu vsakega majhnega koraka seštevamo. Ko govorimo o velikih nelinearnostih (velike rotacije, pomiki, deformacije), nastajajo pri vsakem inkrementu napake, ki se skozi analizo lahko seštevajo in na koncu predstavljajo pomemben vpliv:

$$Napaka = {}^{t+\Delta t}R - \int_{0V} {}^t_0S_{ij} \cdot \delta_0\varepsilon_{ij} \cdot {}^0dV \quad (3.55)$$

Na tej točki lahko vpeljemo še metodo iteriranja v vsakem inkrementu in s tem odstranimo napako ali »out-of-balance virtual work« v obseg absolutnega ali relativnega konvergenčnega kriterija. Pri tem v enačbi predpostavimo, da so notranje obremenitve ${}^t_0\mathbf{F}$ enake notranjim obremenitvam ${}^{t+\Delta t}_0\mathbf{F}^{(0)}$ iz prejšnjega koraka.

S tem lahko zapišemo končno splošno obliko iterativne enačbe, kjer pa se je nujno zavedati, da je treba notranje obremenitve pravilno izračunavati, sicer konvergiramo k napačni vrednosti.

$$\int_{0V} {}_0C_{ijrs} \cdot \Delta({}_0\varepsilon_{rs})^{(k)} \delta({}_0\varepsilon_{ij})^0 dV + \int_{0V} {}^t_0S_{ij} \delta\Delta_0\eta_{ij}^{(k)0} dV = {}^{t+\Delta t}R - \int_{0V} {}^{t+\Delta t}_0S_{ij}^{(k-1)} \delta{}^{t+\Delta t}_0E_{ij}^{(k-1)0} dV \quad (3.56)$$

$${}^t_0\mathbf{K} \cdot \Delta\mathbf{U}^{(k)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}_0\mathbf{F}^{(k-1)} \quad (3.57)$$

Rezultat predstavljajo pomiki, ki jih seštevamo na koncu vsake iteracije.

$${}^{t+\Delta t}\mathbf{U}^{(k)} = {}^t\mathbf{U} + \sum_{i=1}^k \Delta\mathbf{U}^{(i)} \quad (3.58)$$

Celoten postopek iteriranja v inkrementu lahko najlažje predstavimo s pomočjo psevdokode.

// Vhodni podatki

$${}^{t+\Delta t}\mathbf{R}, {}^t\mathbf{U}$$

// Določi relativno ali absolutno tolerančno območje za testiranje konvergence.

// Izračun togostne matrike in vektor notranjih sil iteracije.

$${}^t_0\mathbf{K}, {}^t_0\mathbf{F}$$

// Določitev začetnih vrednosti pomikov in notranjih obremenitev

$${}^{t+\Delta t}_0\mathbf{F}^{(0)} = {}^t_0\mathbf{F}$$

$${}^{t+\Delta t}_0\mathbf{U}^{(0)} = {}^t\mathbf{U}$$

// Zanka: Ponavljaj (1), dokler konvergenčni kriterij ni izpolnjen (k++).

$${}^t_0\mathbf{K} \cdot \Delta\mathbf{U}^{(k)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}_0\mathbf{F}^{(k-1)}$$

$${}^{t+\Delta t}\mathbf{U}^{(k)} = {}^{t+\Delta t}\mathbf{U}^{(k-1)} + \Delta\mathbf{U}^{(k)}$$

// Iz novoizračunanih pomikov določi novo notranjo obremenitev.

Pomiki → deformacije → Napetosti → Obremenitve

$${}^{t+\Delta t}\mathbf{U}^{(k)} \rightarrow {}^{t+\Delta t}_0\mathbf{F}^{(k)}$$

// Določi napako oz. odstopek od ravnovesja.

$$Error_ABS = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}_0\mathbf{F}^{(k)}$$

// Izvedi test konvergence.

//Konec zanke (1)

// Rezultat ${}^{t+\Delta t}\mathbf{U}^{(k)}$

Sledi izpeljava matrik ravninskega izoparametričnega KE na osnovi predstavljenih enačb. Osnovo predstavljajo povsem enake interpolacijske funkcije ter druge osnovne enačbe, skica, kot smo jih uporabili pri izpeljavi linearnega KE (glej prejšnje podpoglavje). Pri izpeljavi smo si pomagali z nekaterimi pojasnjenimi zvezami iz vira [10].

Izhajamo iz enačbe PVD.

$$\int_{0V} {}_0C_{ijrs} \cdot ({}_0\varepsilon_{rs}) \cdot \delta({}_0\varepsilon_{ij}) {}^0dV + \int_{0V} {}^t_0S_{ij} \cdot \delta_0\eta_{ij} \cdot {}^0dV = {}^{t+\Delta t}R - \int_{0V} {}^t_0S_{ij} \cdot \delta_0\varepsilon_{ij} \cdot {}^0dV \quad (3.59)$$

V naslednjem koraku zapišemo izraze za izračun poljubne točke v polju KE ob času 0 in t.

Notacijo ${}^ax_b^c$ si lahko razlagamo kot koordinato z vozliščem c v smeri b ob času a.

$${}^0x_1 = \sum_{i=k}^4 N_k \cdot {}^0x_1^k, \quad {}^0x_2 = \sum_{i=k}^4 N_k \cdot {}^0x_2^k \quad (3.60)$$

$${}^tx_1 = \sum_{i=k}^4 N_k \cdot {}^tx_1^k, \quad {}^tx_2 = \sum_{i=k}^4 N_k \cdot {}^tx_2^k \quad (3.61)$$

Ker gre za izoparametrični element, lahko podobno zapišemo še za pomike.

$$u_i = \sum_{k=1}^4 N_k \cdot u_i^k, \quad {}^tu_i = \sum_{k=1}^4 N_k \cdot {}^tu_i^k \quad (3.62)$$

Sledi zapis deformacij, kjer deformacije razstavimo (glej enačbo 3.43, 3.44, 3.45) in lahko vidimo, da desna matrika predstavlja učinek začetnih pomikov (lastnost celotne Lagrangeeve formulacije).

$$\begin{aligned} {}_0\varepsilon = \begin{bmatrix} {}_0\varepsilon_{11} \\ {}_0\varepsilon_{22} \\ 2 \cdot {}_0\varepsilon_{12} \end{bmatrix} &= \begin{bmatrix} {}^0u_{1,1} \\ {}^0u_{2,2} \\ {}^0u_{1,2} + {}^0u_{2,1} \end{bmatrix} + \begin{bmatrix} {}^t_0u_{1,1} \cdot {}^0u_{1,1} + {}^t_0u_{2,1} \cdot {}^0u_{2,1} \\ {}^t_0u_{1,2} \cdot {}^0u_{1,2} + {}^t_0u_{2,2} \cdot {}^0u_{2,2} \\ {}^t_0u_{1,1} \cdot {}^0u_{1,1} + {}^t_0u_{2,1} \cdot {}^0u_{2,2} + {}^t_0u_{1,2} \cdot {}^0u_{1,1} + {}^t_0u_{2,2} \cdot {}^0u_{2,1} \end{bmatrix} \\ &= \begin{bmatrix} \frac{\partial u_1}{\partial {}^0x_1} \\ \frac{\partial u_2}{\partial {}^0x_2} \\ \frac{\partial u_1}{\partial {}^0x_2} + \frac{\partial u_2}{\partial {}^0x_1} \end{bmatrix} + \begin{bmatrix} \frac{\partial {}^tu_1}{\partial {}^0x_1} \cdot \frac{\partial u_1}{\partial {}^0x_1} + \frac{\partial {}^tu_2}{\partial {}^0x_1} \cdot \frac{\partial u_2}{\partial {}^0x_1} \\ \frac{\partial {}^tu_1}{\partial {}^0x_2} \cdot \frac{\partial u_1}{\partial {}^0x_2} + \frac{\partial {}^tu_2}{\partial {}^0x_2} \cdot \frac{\partial u_2}{\partial {}^0x_2} \\ \frac{\partial {}^tu_1}{\partial {}^0x_1} \cdot \frac{\partial u_1}{\partial {}^0x_1} + \frac{\partial {}^tu_2}{\partial {}^0x_1} \cdot \frac{\partial u_2}{\partial {}^0x_2} + \frac{\partial {}^tu_1}{\partial {}^0x_2} \cdot \frac{\partial u_1}{\partial {}^0x_1} + \frac{\partial {}^tu_2}{\partial {}^0x_2} \cdot \frac{\partial u_2}{\partial {}^0x_1} \end{bmatrix} \quad (3.63) \end{aligned}$$

S pomočjo diferencialnega operatorja lahko zvezo zapišemo kot:

$${}_0\varepsilon = (\mathbf{L} + \mathbf{L}^t\mathbf{u}) \cdot \mathbf{u} =$$

$$\left(\begin{array}{cc} \frac{\partial}{\partial^0 x_1} & 0 \\ 0 & \frac{\partial}{\partial^0 x_2} \\ \frac{\partial}{\partial^0 x_2} & \frac{\partial}{\partial^0 x_1} \end{array} \right) + \left(\begin{array}{cc} \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_1} \\ \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_2} \\ \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_1} + \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_1} + \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_2} \end{array} \right) \begin{array}{c} [u_1] \\ [u_2] \end{array} \quad (3.64)$$

Podobno sledi še za variacijo.

$$\delta_0 \boldsymbol{\varepsilon} = (\mathbf{L} + \mathbf{L}^t \mathbf{u}) \cdot \delta \mathbf{u} = \quad (3.65)$$

S tem lahko zapišemo prvi del člena PVD.

$$\begin{aligned} \int_{0V} {}_0 C_{ijrs} \cdot ({}_0 \varepsilon_{rs}) \cdot \delta ({}_0 \varepsilon_{ij})^0 dV &= \int_{0V} \delta ({}_0 \boldsymbol{\varepsilon})^T \cdot {}_0 \mathbf{D} \cdot ({}_0 \boldsymbol{\varepsilon})^0 dV \\ &= \int_{0V} \delta \mathbf{u}^T \cdot (\mathbf{L} + \mathbf{L}^t \mathbf{u})^T \cdot \mathbf{D} \cdot (\mathbf{L} + \mathbf{L}^t \mathbf{u}) \cdot \mathbf{u}^0 dV \end{aligned} \quad (3.66)$$

$$\mathbf{{}_0 \boldsymbol{\eta}} = \begin{bmatrix} {}_0 \eta_{11} \\ {}_0 \eta_{22} \\ 2 \cdot {}_0 \eta_{12} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} ({}_0 u_{1,1})^2 + ({}_0 u_{2,1})^2 \\ ({}_0 u_{1,2})^2 + ({}_0 u_{2,2})^2 \\ 2 \cdot ({}_0 u_{1,1} \cdot {}_0 u_{1,2} + {}_0 u_{2,1} \cdot {}_0 u_{2,2}) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \left(\frac{\partial u_1}{\partial^0 x_1} \right)^2 + \left(\frac{\partial u_2}{\partial^0 x_1} \right)^2 \\ \left(\frac{\partial u_1}{\partial^0 x_2} \right)^2 + \left(\frac{\partial u_2}{\partial^0 x_2} \right)^2 \\ 2 \cdot \left(\frac{\partial u_1}{\partial^0 x_1} \cdot \frac{\partial u_1}{\partial^0 x_2} + \frac{\partial u_2}{\partial^0 x_1} \cdot \frac{\partial u_2}{\partial^0 x_2} \right) \end{bmatrix} \quad (3.67)$$

Podobno sledi za nelinearni del deformacij.

S pomočjo diferencialnega operatorja lahko zvezo zapišemo kot:

$$\mathbf{{}_0 \boldsymbol{\eta}} = \frac{1}{2} \cdot (\mathbf{L} \mathbf{u}) \cdot \mathbf{u} = \left(\begin{array}{cc} \frac{\partial u_1}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_1} & \frac{\partial u_2}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_1} \\ \frac{\partial u_1}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_2} & \frac{\partial u_2}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_2} \\ \frac{\partial u_1}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_1} + \frac{\partial u_1}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_2} & \frac{\partial u_2}{\partial^0 x_2} \cdot \frac{\partial}{\partial^0 x_1} + \frac{\partial u_2}{\partial^0 x_1} \cdot \frac{\partial}{\partial^0 x_2} \end{array} \right) \begin{array}{c} [u_1] \\ [u_2] \end{array} \quad (3.68)$$

Podobno lahko zapišemo še za variacijo deformacije.

$$\delta_0 \boldsymbol{\eta} = (\mathbf{L}\mathbf{u}) \cdot \delta \mathbf{u} = (\mathbf{L}\delta \mathbf{u}) \cdot \mathbf{u} = \begin{bmatrix} \frac{\partial \delta u_1}{\partial {}^0x_1} \cdot \frac{\partial u_1}{\partial {}^0x_1} + \frac{\partial \delta u_2}{\partial {}^0x_1} \cdot \frac{\partial u_2}{\partial {}^0x_1} \\ \frac{\partial \delta u_1}{\partial {}^0x_2} \cdot \frac{\partial u_1}{\partial {}^0x_2} + \frac{\partial \delta u_2}{\partial {}^0x_2} \cdot \frac{\partial u_2}{\partial {}^0x_2} \\ \frac{\partial \delta u_1}{\partial {}^0x_1} \cdot \frac{\partial u_1}{\partial {}^0x_2} + \frac{\partial u_1}{\partial {}^0x_1} \cdot \frac{\partial \delta u_1}{\partial {}^0x_2} + \frac{\partial \delta u_2}{\partial {}^0x_1} \cdot \frac{\partial u_2}{\partial {}^0x_2} + \frac{\partial u_2}{\partial {}^0x_1} \cdot \frac{\partial \delta u_2}{\partial {}^0x_2} \end{bmatrix} \quad (3.69)$$

Sledi zapis vektorja napetosti (2PK) prek znane konstitutivne zveze.

$${}^t_0 \mathbf{E} = \begin{bmatrix} {}_0E_{11} \\ {}_0E_{22} \\ 2 \cdot {}_0E_{12} \end{bmatrix} = \begin{bmatrix} \frac{\partial {}^t u_1}{\partial {}^0x_1} \\ \frac{\partial {}^t u_2}{\partial {}^0x_2} \\ \frac{\partial {}^t u_1}{\partial {}^0x_2} + \frac{\partial {}^t u_2}{\partial {}^0x_1} \end{bmatrix} + \frac{1}{2} \begin{bmatrix} \left(\frac{\partial {}^t u_1}{\partial {}^0x_1} \right)^2 + \left(\frac{\partial {}^t u_2}{\partial {}^0x_1} \right)^2 \\ \left(\frac{\partial {}^t u_1}{\partial {}^0x_2} \right)^2 + \left(\frac{\partial {}^t u_2}{\partial {}^0x_2} \right)^2 \\ 2 \cdot \left(\frac{\partial {}^t u_1}{\partial {}^0x_1} \cdot \frac{\partial {}^t u_1}{\partial {}^0x_2} + \frac{\partial {}^t u_2}{\partial {}^0x_1} \cdot \frac{\partial {}^t u_2}{\partial {}^0x_2} \right) \end{bmatrix} \quad (3.70)$$

$${}^t_0 \mathbf{S} = \begin{bmatrix} {}_0S_{11} \\ {}_0S_{22} \\ {}_0S_{12} \end{bmatrix} = \mathbf{D} \cdot {}^t_0 \mathbf{E} = \begin{bmatrix} {}_0D_{11} & {}_0D_{12} & 0 \\ {}_0D_{12} & {}_0D_{22} & 0 \\ 0 & 0 & {}_0D_{66} \end{bmatrix} \cdot \begin{bmatrix} {}_0E_{11} \\ {}_0E_{22} \\ 2 \cdot {}_0E_{12} \end{bmatrix} \quad (3.71)$$

Z izpeljanimi zvezami lahko zapišemo drugi segment integralske oblike PVD.

$$\int_{0V} {}^t_0 S_{ij} \cdot \delta_0 \eta_{ij} \cdot {}^0 dV = \int_{0V} \delta_0 \boldsymbol{\eta}^T \cdot {}^t_0 \mathbf{S} \cdot {}^0 dV \quad (3.72)$$

Pri tem avtor [10] pojasni, da bil zapis v takšni obliki neprimeren (matrična oblika 4 členov za formulacijo KE), zato izpelje naslednjo zvezo:

$$\int_{0V} \delta_0 \boldsymbol{\eta}^T \cdot {}^t_0 \mathbf{S} \cdot {}^0 dV = \delta \mathbf{u}^T \cdot \mathbf{L}_n^T \cdot {}^t_0 \mathbf{S} \cdot \mathbf{L}_n \cdot \mathbf{u} \cdot {}^0 dV \quad (3.73)$$

Pri tem vpeljemo še:

$$\mathbf{L}_n = \begin{bmatrix} \frac{\partial}{\partial {}^0x_1} & 0 \\ \frac{\partial}{\partial {}^0x_2} & 0 \\ 0 & \frac{\partial}{\partial {}^0x_1} \\ 0 & \frac{\partial}{\partial {}^0x_2} \end{bmatrix} \quad (3.74)$$

S tem nam za zapis ostane le še zadnji segment, za katerega pa že imamo izpeljane vse zveze, zato lahko zapišemo:

$$\int_{0V} {}^tS_{ij} \cdot \delta_0 \varepsilon_{ij} \cdot {}^0dV = \int_{0V} \delta_0 \boldsymbol{\varepsilon}^T \cdot {}^t\mathbf{S} \cdot {}^0dV = ((\mathbf{L} + \mathbf{L}^t\mathbf{u}) \cdot \boldsymbol{\delta u})^T \cdot {}^t\mathbf{S} \quad (3.75)$$

Podobno kot pri izpeljavi linearnega elementa tudi tokrat v enačbe vstavimo pomike po polju elementa s pomočjo interpolacijskih funkcij (glej enačbo 3.60) ter predstavimo pripadajoča vektorja vozliščnih pomikov.

$${}^t\mathbf{U} = \begin{bmatrix} {}^tu_1^1 \\ {}^tu_2^1 \\ {}^tu_1^2 \\ {}^tu_2^2 \\ {}^tu_1^3 \\ {}^tu_2^3 \\ {}^tu_1^4 \\ {}^tu_2^4 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ u_1^3 \\ u_2^3 \\ u_1^4 \\ u_2^4 \end{bmatrix} \quad (3.76)$$

Zapisano lahko ponovno vstavimo in združimo v enačbi PVD.

$$\int_{0V} \boldsymbol{\delta u}^T \cdot (\mathbf{L} + \mathbf{L}^t\mathbf{u})^T \cdot \mathbf{D} \cdot (\mathbf{L} + \mathbf{L}^t\mathbf{u}) \cdot \mathbf{u} \cdot {}^0dV + \int_{0V} \boldsymbol{\delta u}^T \cdot \mathbf{L}_n^T \cdot {}^t\mathbf{S} \cdot \mathbf{L}_n \cdot \mathbf{u} \cdot {}^0dV = {}^{t+\Delta t}\mathbf{R} - \int_{0V} \delta_0 \boldsymbol{\varepsilon}^T \cdot {}^t\mathbf{S} \cdot {}^0dV \quad (3.77)$$

$$\int_{0V} \boldsymbol{\delta U}^T \cdot {}^t\mathbf{B}_L^T \cdot \mathbf{D} \cdot {}^t\mathbf{B}_L \cdot \mathbf{U} \cdot {}^0dV + \int_{0V} \boldsymbol{\delta U}^T \cdot {}^t\mathbf{B}_{NL}^T \cdot {}^t\mathbf{S} \cdot {}^t\mathbf{B}_{NL} \cdot \mathbf{U} \cdot {}^0dV = {}^{t+\Delta t}\mathbf{R} - \int_{0V} {}^t\mathbf{B}_L^T \cdot {}^t\mathbf{S} \cdot {}^0dV \quad (3.78)$$

Iz že sedaj zelo dobro poznanih členov lahko v naslednjem koraku zapišemo enačbe za izračun togostne matrike.

$${}^t\mathbf{K}_L = \int_{0V} {}^t\mathbf{B}_L^T \cdot \mathbf{D} \cdot {}^t\mathbf{B}_L \cdot {}^0dV \quad (3.79)$$

$${}^t\mathbf{K}_{NL} = \int_{0V} {}^t\mathbf{B}_{NL}^T \cdot {}^t\mathbf{S} \cdot {}^t\mathbf{B}_{NL} \cdot {}^0dV \quad (3.80)$$

$${}^t\mathbf{F} = \int_{0V} {}^t\mathbf{B}_L^T \cdot {}^t\mathbf{S} \cdot {}^0dV \quad (3.81)$$

Z vpeljavo togostnih matrik in vektorja notranjih obremenitev dobi vodilna enačba še končno podobo.

$$({}^t\mathbf{K}_L + {}^t\mathbf{K}_{NL}) \cdot \mathbf{U} = {}^{t+\Delta t}\mathbf{R} - {}^t\mathbf{F} \quad (3.82)$$

Za implementacijo programske kode togostnih matrik je na koncu treba definirati še gradientne matrike.

$${}^t\mathbf{B}_L = {}^t\mathbf{B}_L^0 + {}^t\mathbf{B}_L^1 = (\mathbf{L} + \mathbf{L}^t\mathbf{u}) \cdot \mathbf{N} \quad (3.83)$$

$${}^t\mathbf{B}_{NL} = (\mathbf{L}_n) \cdot \mathbf{N} \quad (3.84)$$

$${}^t\mathbf{B}_L^0 = \begin{bmatrix} \frac{\partial N_1}{\partial^0 x_1} & 0 & \frac{\partial N_2}{\partial^0 x_1} & 0 & \frac{\partial N_3}{\partial^0 x_1} & 0 & \frac{\partial N_4}{\partial^0 x_1} & 0 \\ 0 & \frac{\partial N_1}{\partial^0 x_2} & 0 & \frac{\partial N_2}{\partial^0 x_2} & 0 & \frac{\partial N_3}{\partial^0 x_2} & 0 & \frac{\partial N_4}{\partial^0 x_2} \\ \frac{\partial N_1}{\partial^0 x_2} & \frac{\partial N_1}{\partial^0 x_1} & \frac{\partial N_2}{\partial^0 x_2} & \frac{\partial N_2}{\partial^0 x_1} & \frac{\partial N_3}{\partial^0 x_2} & \frac{\partial N_3}{\partial^0 x_1} & \frac{\partial N_4}{\partial^0 x_2} & \frac{\partial N_4}{\partial^0 x_1} \end{bmatrix} \quad (3.85)$$

$${}^t\mathbf{B}_L^1 =$$

$$\begin{bmatrix} \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_1}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_1}{\partial^0 x_1} & \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_2}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_2}{\partial^0 x_1} & \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_3}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_3}{\partial^0 x_1} & \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_4}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_4}{\partial^0 x_1} \\ \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_1}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_1}{\partial^0 x_2} & \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_2}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_2}{\partial^0 x_2} & \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_3}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_3}{\partial^0 x_2} & \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_4}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_4}{\partial^0 x_2} \\ \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_1}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_1}{\partial^0 x_2} & \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_2}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_2}{\partial^0 x_2} & \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_3}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_3}{\partial^0 x_2} & \frac{\partial^t u_1}{\partial^0 x_1} \cdot \frac{\partial N_4}{\partial^0 x_2} & \frac{\partial^t u_2}{\partial^0 x_1} \cdot \frac{\partial N_4}{\partial^0 x_2} \\ + & + & + & + & + & + & + & + \\ \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_1}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_1}{\partial^0 x_1} & \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_2}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_2}{\partial^0 x_1} & \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_3}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_3}{\partial^0 x_1} & \frac{\partial^t u_1}{\partial^0 x_2} \cdot \frac{\partial N_4}{\partial^0 x_1} & \frac{\partial^t u_2}{\partial^0 x_2} \cdot \frac{\partial N_4}{\partial^0 x_1} \end{bmatrix} \quad (3.86)$$

$${}^t\mathbf{B}_{NL} = \begin{bmatrix} \frac{\partial N_1}{\partial^0 x_1} & 0 & \frac{\partial N_2}{\partial^0 x_1} & 0 & \frac{\partial N_3}{\partial^0 x_1} & 0 & \frac{\partial N_4}{\partial^0 x_1} & 0 \\ \frac{\partial N_1}{\partial^0 x_2} & 0 & \frac{\partial N_2}{\partial^0 x_2} & 0 & \frac{\partial N_3}{\partial^0 x_2} & 0 & \frac{\partial N_4}{\partial^0 x_2} & 0 \\ 0 & \frac{\partial N_1}{\partial^0 x_1} & 0 & \frac{\partial N_2}{\partial^0 x_1} & 0 & \frac{\partial N_3}{\partial^0 x_1} & 0 & \frac{\partial N_4}{\partial^0 x_1} \\ 0 & \frac{\partial N_1}{\partial^0 x_2} & 0 & \frac{\partial N_2}{\partial^0 x_2} & 0 & \frac{\partial N_3}{\partial^0 x_2} & 0 & \frac{\partial N_4}{\partial^0 x_2} \end{bmatrix} \quad (3.87)$$

S tem lahko zaključimo z izpeljavo celotne Lagrangeeve formulacije in imamo vse potrebno za implementacijo programske kode. V matrikah so sicer še vedno odvodi glede na kartezijske koordinate, ki pa jih zlahka pretvorimo v naravne koordinate s pomočjo Jacobijeve matrike (glej izpeljavo linearnega končnega elementa).

$$\begin{bmatrix} \frac{\partial N_i}{\partial^0 x_1} \\ \frac{\partial N_i}{\partial^0 x_2} \end{bmatrix} = {}^0\mathbf{J}^{-1} \cdot \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \end{bmatrix} \quad (3.88)$$

Videli smo, da je treba izračunavati dodatni del linearne togostne matrike, ki je prispevek učinka začetnih pomikov. Prav tako vidimo, da delamo s tenzorskimi količinami, ki jih je treba za fizikalno interpretacijo pretvoriti, kar zahteva nek določen računski napor. Težavo pa nam lahko povzroča tudi dejstvo, da formulacija ne more opisati tudi velikih deformacij v določenih primerih zaradi nezmožnosti opisa konstitutivnih zakonov. Nadaljevali bomo torej s posodobljeno Lagrangeevo formulacijo, ki bo vsebovala nekatere velike podobnosti z že izpeljanim.

3.2.1 Posodobljena Lagrangeeva formulacija

Iz slike 2.10 lahko vidimo, da se napetosti in deformacije ob času $t + \Delta t$ nanašajo na zadnjo izračunano konfiguracijo ob času t . Podobno kot pri izpeljavi po celotni Lagrangeevi formulaciji bomo tudi tukaj uporabili linearizacijo enačb glede na u_i ter reševanje predstavili s pomočjo inkrementne formulacije ter iteracij s pomočjo modificirane Newtonove metode, ki je bila predstavljena. Tudi tokrat izhajamo iz osnovne enačbe PVD (glej enačbo 3.37).

Izhajamo iz enačbe 3.38, kjer smo razstavili napetosti in deformacije na znane in neznanne inkremente.

Nadaljujemo lahko z zapisom Green-Lagrangeevih deformacij v elementarni obliki odvodov pomikov ob času $t + \Delta t$.

$${}^{t+\Delta t}E_{ij} = \frac{1}{2} \cdot ({}^{t+\Delta t}u_{i,j} + {}^{t+\Delta t}u_{j,i} + {}^{t+\Delta t}u_{k,i} \cdot {}^{t+\Delta t}u_{k,j}) \quad (3.89)$$

V enačbi 3.38 je možno videti, da lahko člena odštejemo in zapišemo zgolj inkrement.

$${}^tE_{ij} = \frac{1}{2} \cdot ({}^tu_{i,j} + {}^tu_{j,i} + {}^tu_{k,i} \cdot {}^tu_{k,j}) \quad (3.90)$$

Če pobližje pogledamo enačbo, je zelo zanimiva primerjava s celotno Lagrangeevo formulacijo, kjer vidimo, da člena, ki je bil posledica učinka začetnih pomikov, ni več.

Ponovno lahko uvedemo linearni in nelinearni člen deformacij, za katera veljajo naslednje zveze:

$${}^tE_{ij} = {}_0\varepsilon_{ij} + {}_0\eta_{ij} \quad (3.91)$$

$${}_0\varepsilon_{ij} = \frac{1}{2} \cdot ({}^tu_{i,j} + {}^tu_{j,i}) \quad (3.92)$$

$${}_0\eta_{ij} = \frac{1}{2} \cdot ({}^tu_{k,i} \cdot {}^tu_{k,j}) \quad (3.93)$$

Za Green-Lagrangeev inkrement deformacij lahko zapišemo še variacijo.

$$\delta {}^tE_{ij} = \delta {}_0\varepsilon_{ij} + \delta {}_0\eta_{ij} \quad (3.94)$$

Sledi zapis, kjer variacijo z vsemi pravkar izpeljanimi enačbami vstavimo v osnovni nastavek PVD. Integracija tokrat poteka glede na zadnjo izračunano domeno ob času t .

$$\int_{tV} {}^tS_{ij} \cdot \delta {}^tE_{ij} \cdot {}^tdV + \int_{tV} {}^t\sigma_{ij} \cdot \delta {}_0\eta_{ij} \cdot {}^tdV = {}^{t+\Delta t}R - \int_{tV} {}^t\sigma_{ij} \cdot \delta {}_t\varepsilon_{ij} \cdot {}^tdV \quad (3.95)$$

Zapisana enačba podobno kot pri TLF vsebuje nekatere nelinearnosti, ki predstavljajo največjo težavo pri reševanju, zato je glavni cilj naslednjih nekaj korakov ponovno linearizacija vseh členov višjega reda, za kar pa je potrebna bolj podrobna analiza posameznih segmentov:

- prvi segment:

$$\int_{tV} {}^t\sigma_{ij} \cdot \delta {}_t\eta_{ij} \cdot {}^t dV$$

${}^t\sigma_{ij}$ ob času t poznamo, zato je člen zagotovo linearen glede na u_i , medtem ko lahko za člen $\delta {}_t\eta_{ij}$ razpišemo variacijo in ugotovimo, da sta člena $\delta {}_t u_{k,j}$ in $\delta {}_t u_{k,i}$ variaciji, ki sta konstantni, zato lahko celotni člen pustimo nedotaknjen.

$$\delta {}_0\eta_{ij} = \frac{1}{2} \cdot (\delta {}_t u_{k,i} \cdot {}^t u_{k,j}) + \frac{1}{2} \cdot ({}^t u_{k,i} \cdot \delta {}_t u_{k,j}) \quad (3.96)$$

- Drugi segment:

$$\int_{tV} {}^t S_{ij} \cdot \delta {}_t E_{ij} \cdot {}^t dV$$

Člen $\delta {}_t E_{ij}$ je glede na enačbo (3.91) sestavljen iz variacije linearnega in nelinearnega člena. Pri prejšnjem segmentu enačbe smo pokazali, da variacija nelinearnega člena deformacije vsebuje zgolj linearne člene glede na u_i . Tudi pri tej formulaciji nam preostane le člen ${}_0 S_{ij}$, ki ga lahko razvijemo v Taylorjevo vrsto ali pa preprosto po vzgledu [10] zapišemo naslednjo zvezo.

$$\begin{aligned} {}^t S_{ij} &= {}^t C_{ijrs} \cdot {}^t E_{rs} \quad , \quad \delta {}_t E_{rs} \approx \delta {}_t \varepsilon_{rs} \\ {}^t S_{ij} &= {}^t C_{ijrs} \cdot ({}^t \varepsilon_{rs} + {}^t \eta_{rs}) \quad \rightarrow \quad {}^t S_{ij} \approx {}^t C_{ijrs} \cdot ({}^t \varepsilon_{rs}) \end{aligned} \quad (3.97)$$

Tudi v tem primeru je enačba glede na u_i nelinearna, ampak lahko glede na to, da bomo tudi pri ULF ponovno reševali metodo s pomočjo inkrementov, razmišljamo v smeri, da bodo inkrementi majhni. Posledično bodo tudi pomiki in deformacije majhni. Z naslednjo enačbo lahko torej zapišemo linearizirani segment, ki vsebuje še zadnji produkt, ki bi povzročil višji red člena u_i .

$${}^t S_{ij} \cdot \delta {}_t E_{ij} \approx {}^t C_{ijrs} \cdot ({}^t \varepsilon_{rs}) \cdot \delta {}_t E_{ij} \approx {}^t C_{ijrs} \cdot ({}^t \varepsilon_{rs}) \cdot \delta ({}^t \varepsilon_{ij} + {}^t \eta_{ij}) \quad (3.98)$$

Produkt $({}^t C_{ijrs} \cdot {}^t \varepsilon_{rs} \cdot \delta {}_t \eta_{ij})$ torej zanemarimo in zapišemo končno obliko.

$${}^t S_{ij} \cdot \delta {}_t E_{ij} \approx {}^t C_{ijrs} \cdot ({}^t \varepsilon_{rs}) \cdot \delta ({}^t \varepsilon_{ij}) \quad (3.99)$$

$$\int_{t_V} {}^t \sigma_{ij} \cdot \delta {}_t \varepsilon_{ij} \cdot {}^t dV$$

Ča si pogloblje pogledamo še zadnji segment, ki nam je ostal, vidimo, da podobno kot pri prvem členu tudi tukaj ni nobenih nelinearnosti.

Preostane nam torej zapis končne linearizirane enačbe PVD.

$$\int_{t_V} {}^t C_{ijrs} \cdot ({}_t \varepsilon_{rs}) \cdot \delta ({}_t \varepsilon_{ij}) \cdot {}^t dV + \int_{t_V} {}^t \sigma_{ij} \cdot \delta {}_t \eta_{ij} \cdot {}^t dV = {}^{t+\Delta t} R - \int_{t_V} {}^t \sigma_{ij} \cdot \delta {}_t \varepsilon_{ij} \cdot {}^t dV \quad (3.100)$$

Ponovno smo torej linearizirali zgolj prvi člen, kjer so nastopale nelinearnosti. Na tej točki lahko sistem diskretiziramo v obliko, kjer enačba na levi strani enačaja rezultira v linearni tangentni togostni matriki. Obe strani sta pomnoženi s transponiranim vektorjem virtualnih pomikov, medtem ko desna stran enačaja predstavlja residual med notranjim in zunanjim virtualnim delom.

$$\begin{aligned}\delta \mathbf{U}^T \cdot {}^t \mathbf{K} \cdot \Delta \mathbf{U} &= \delta \mathbf{U}^T \cdot ({}^{t+\Delta t} \mathbf{R} - {}^t \mathbf{F}) \\ {}^t \mathbf{K} \cdot \Delta \mathbf{U} &= {}^{t+\Delta t} \mathbf{R} - {}^t \mathbf{F}\end{aligned}\quad (3.101)$$

Podobno kot pri TLF lahko tudi tukaj predstavimo residual ali »out-of-balance virtual work«.

$$Napaka = {}^{t+\Delta t} R - \int_{t_V} {}^t \sigma_{ij} \cdot \delta {}_t \varepsilon_{ij} {}^t dV \quad (3.102)$$

Na koncu vpeljimo še metodo iteriranja (Newtonova modificirana metoda) v vsakem inkrementu do dosega absolutnega ali relativnega konvergenčnega kriterija po zgledu iz prejšnjega podpoglavja.

$$\begin{aligned}\int_{t_V} {}^t C_{ijrs} \cdot \Delta ({}_t \varepsilon_{rs})^{(k)} \delta ({}_t \varepsilon_{ij}) {}^t dV + \int_{t_V} {}^t \sigma_{ij} \delta \Delta {}_t \eta_{ij}^{(k)} {}^t dV &= {}^{t+\Delta t} R \\ - \int_{t+\Delta t_V^{(k-1)}} {}^{t+\Delta t} \sigma_{ij}^{(k-1)} \delta {}_{t+\Delta t} \varepsilon_{ij}^{(k-1)} {}^{t+\Delta t} dV &\end{aligned}\quad (3.103)$$

$${}^t \mathbf{K} \cdot \Delta \mathbf{U}^{(k)} = {}^{t+\Delta t} \mathbf{R} - \frac{{}^{t+\Delta t} \mathbf{F}^{(k-1)}}{{}^{t+\Delta t}} \quad (3.104)$$

$${}^{t+\Delta t} \mathbf{U}^{(k)} = {}^t \mathbf{U} + \sum_{i=1}^k \Delta \mathbf{U}^{(i)} \quad (3.105)$$

Celoten postopek iteriranja v inkrementu lahko podobno kot za TLF predstavimo s pomočjo psevdokode.

```

// Vhodni podatki
 ${}^{t+\Delta t}\mathbf{R}, {}^t\mathbf{U}$ 
// Določi relativno ali absolutno tolerančno območje za testiranje konvergence.
// Izračun togostne matrike in vektor notranjih sil iteracije.
 ${}^t\mathbf{K}, {}^t\mathbf{F}$ 
// Določitev začetnih vrednosti pomikov in notranjih obremenitev.
 ${}^{t+\Delta t}\mathbf{F}^{(0)} = {}^t\mathbf{F}$ 
 ${}^{t+\Delta t}\mathbf{U}^{(0)} = {}^t\mathbf{U}$ 
// Zanka ponavlja (1), dokler konvergenčni kriterij ni izpolnjen (k++).
 ${}^t\mathbf{K} \cdot \Delta\mathbf{U}^{(k)} = {}^{t+\Delta t}\mathbf{R} - \frac{{}^{t+\Delta t}\mathbf{F}^{(k-1)}}{{}^{t+\Delta t}}$ 
 ${}^{t+\Delta t}\mathbf{U}^{(k)} = {}^{t+\Delta t}\mathbf{U}^{(k-1)} + \Delta\mathbf{U}^{(k)}$ 
// Iz novoizračunanih pomikov določi novo notranjo obremenitev.
Pomiki → deformacije → Napetosti → Obremenitve
 ${}^{t+\Delta t}\mathbf{U}^{(k)} \rightarrow \frac{{}^{t+\Delta t}\mathbf{F}^{(k)}}{{}^{t+\Delta t}}$ 
// Določi napako oz. odstopek od ravnovesja.
 $Error\_ABS = {}^{t+\Delta t}\mathbf{R} - \frac{{}^{t+\Delta t}\mathbf{F}^{(k)}}{{}^{t+\Delta t}}$ 
// Izvedi test konvergence.
//Konec zanke (1)
// Rezultat  ${}^{t+\Delta t}\mathbf{U}^{(k)}$ 

```

S tem, ko poznamo shemo algoritma, lahko nadaljujemo z izpeljavo ravninskega izoparametričnega KE. Osnovo predstavljajo povsem enake interpolacijske funkcije ter druge osnovne enačbe, skica, kot smo jih uporabili pri izpeljavi linearnega KE (glej pripadajoče podpoglavje). Tudi tokrat smo si pri izpeljavi pomagali z nekaterimi pojasnjenimi zvezami iz vira [10].

Izhajamo iz enačbe PVD.

$$\int_{tV} {}_t C_{ijrs} \cdot ({}_t \varepsilon_{rs}) \cdot \delta({}_t \varepsilon_{ij}) {}^t dV + \int_{tV} {}^t \sigma_{ij} \cdot \delta {}_t \eta_{ij} {}^t dV = {}^{t+\Delta t} R - \int_{tV} {}^t \sigma_{ij} \cdot \delta {}_t \varepsilon_{ij} {}^t dV \quad (3.106)$$

V naslednjem koraku zapišemo izraze za izračun poljubne točke v polju KE ob času t .

$${}^t x_1 = \sum_{i=k}^4 N_k \cdot {}^t x_1^k, \quad {}^t x_2 = \sum_{i=k}^4 N_k \cdot {}^t x_2^k \quad (3.107)$$

Podobno sledi še za pomike.

$$u_i = \sum_{k=1}^4 N_k \cdot u_i^k, \quad {}^t u_i = \sum_{k=1}^4 N_k \cdot {}^t u_i^k \quad (3.108)$$

Sledi zapis deformacij, kjer deformacije razstavimo (glej enačbe 3.43, 3.44, 3.45).

$${}^t \boldsymbol{\varepsilon} = \begin{bmatrix} {}^t \varepsilon_{11} \\ {}^t \varepsilon_{22} \\ 2 \cdot {}^t \varepsilon_{12} \end{bmatrix} = \begin{bmatrix} {}^t u_{1,1} \\ {}^t u_{2,2} \\ {}^t u_{1,2} + {}^t u_{2,1} \end{bmatrix} = \begin{bmatrix} \frac{\partial u_1}{\partial {}^t x_1} \\ \frac{\partial u_2}{\partial {}^t x_2} \\ \frac{\partial u_1}{\partial {}^t x_2} + \frac{\partial u_2}{\partial {}^t x_1} \end{bmatrix} \quad (3.109)$$

S pomočjo diferencialnega operatorja lahko zvezo zapišemo kot:

$${}^t \boldsymbol{\varepsilon} = ({}^t \mathbf{L}) \cdot \mathbf{u} = \begin{pmatrix} \frac{\partial}{\partial {}^t x_1} & 0 \\ 0 & \frac{\partial}{\partial {}^t x_2} \\ \frac{\partial}{\partial {}^t x_2} & \frac{\partial}{\partial {}^t x_1} \end{pmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad (3.110)$$

Za zapis PVD bomo potrebovali še variacijo.

$$\delta {}^t \boldsymbol{\varepsilon} = ({}^t \mathbf{L}) \cdot \delta \mathbf{u} = \quad (3.111)$$

$$\int_{tV} {}_t C_{ijrs} \cdot ({}_t \varepsilon_{rs}) \cdot \delta({}_t \varepsilon_{ij}) {}^t dV = \quad (3.112)$$

$$= \int_{tV} \delta({}_t \boldsymbol{\varepsilon})^T \cdot {}_t \mathbf{D} \cdot ({}_t \boldsymbol{\varepsilon}) {}^t dV = \int_{tV} \delta \mathbf{u}^T \cdot ({}_t \mathbf{L})^T {}_t \mathbf{D} ({}_t \mathbf{L}) \cdot \mathbf{u} {}^t dV$$

$${}_t \boldsymbol{\eta} = \begin{bmatrix} {}_t \eta_{11} \\ {}_t \eta_{22} \\ 2 \cdot {}_t \eta_{12} \end{bmatrix} = \frac{1}{2} \begin{bmatrix} ({}_t u_{1,1})^2 + ({}_t u_{2,1})^2 \\ ({}_t u_{1,2})^2 + ({}_t u_{2,2})^2 \\ 2 \cdot ({}_t u_{1,1} \cdot {}_t u_{1,2} + {}_t u_{2,1} \cdot {}_t u_{2,2}) \end{bmatrix} = \frac{1}{2} \begin{bmatrix} \left(\frac{\partial u_1}{\partial {}^t x_1} \right)^2 + \left(\frac{\partial u_2}{\partial {}^t x_1} \right)^2 \\ \left(\frac{\partial u_1}{\partial {}^t x_2} \right)^2 + \left(\frac{\partial u_2}{\partial {}^t x_2} \right)^2 \\ 2 \cdot \left(\frac{\partial u_1}{\partial {}^t x_1} \cdot \frac{\partial u_1}{\partial {}^t x_2} + \frac{\partial u_2}{\partial {}^t x_1} \cdot \frac{\partial u_2}{\partial {}^t x_2} \right) \end{bmatrix} \quad (3.113)$$

Podobno sledi za nelinearni del deformacij.

Analogno prejšnjemu podpoglavju lahko s pomočjo diferencialnega operatorja opišemo še nelinearni del deformacij z razliko v času koordinate, po kateri odvajamo.

$${}_t \mathbf{L}_n = \begin{bmatrix} \frac{\partial}{\partial {}^t x_1} & 0 \\ \frac{\partial}{\partial {}^t x_2} & 0 \\ 0 & \frac{\partial}{\partial {}^t x_1} \\ 0 & \frac{\partial}{\partial {}^t x_2} \end{bmatrix} \quad (3.114)$$

Sledi zapis vektorja napetosti prek znane konstitutivne zveze, za kar pa najprej potrebujemo zapisan vektor deformacij. Avtor [10] pri izračunu deformacij uporabi Almansijeve deformacije, ki pa niso idealne za opis velikih deformacij.

$${}_t \mathbf{E} = \begin{bmatrix} {}_t E_{11} \\ {}_t E_{22} \\ 2 \cdot {}_t E_{12} \end{bmatrix} = \begin{bmatrix} \frac{\partial {}^t u_1}{\partial {}^t x_1} \\ \frac{\partial {}^t u_2}{\partial {}^t x_2} \\ \frac{\partial {}^t u_1}{\partial {}^t x_2} + \frac{\partial {}^t u_2}{\partial {}^t x_1} \end{bmatrix} - \frac{1}{2} \begin{bmatrix} \left(\frac{\partial {}^t u_1}{\partial {}^t x_1} \right)^2 + \left(\frac{\partial {}^t u_2}{\partial {}^t x_1} \right)^2 \\ \left(\frac{\partial {}^t u_1}{\partial {}^t x_2} \right)^2 + \left(\frac{\partial {}^t u_2}{\partial {}^t x_2} \right)^2 \\ 2 \cdot \left(\frac{\partial {}^t u_1}{\partial {}^t x_1} \cdot \frac{\partial {}^t u_1}{\partial {}^t x_2} + \frac{\partial {}^t u_2}{\partial {}^t x_1} \cdot \frac{\partial {}^t u_2}{\partial {}^t x_2} \right) \end{bmatrix} \quad (3.115)$$

$${}^t\boldsymbol{\sigma} = \begin{bmatrix} {}^t\sigma_{11} & {}^t\sigma_{12} & 0 & 0 \\ {}^t\sigma_{12} & {}^t\sigma_{22} & 0 & 0 \\ 0 & 0 & {}^t\sigma_{11} & {}^t\sigma_{12} \\ 0 & 0 & {}^t\sigma_{12} & {}^t\sigma_{22} \end{bmatrix} \quad (3.116)$$

$${}^t\boldsymbol{\sigma} = \begin{bmatrix} {}^t\sigma_{11} \\ {}^t\sigma_{22} \\ {}^t\sigma_{12} \end{bmatrix} = \mathbf{D} \cdot {}^t\mathbf{E} = \begin{bmatrix} {}^tD_{11} & {}^tD_{12} & 0 \\ {}^tD_{12} & {}^tD_{22} & 0 \\ 0 & 0 & {}^tD_{66} \end{bmatrix} \cdot \begin{bmatrix} {}^tE_{11} \\ {}^tE_{22} \\ 2 \cdot {}^tE_{12} \end{bmatrix} \quad (3.117)$$

V primeru, da si želimo opisati tudi velike deformacije, avtorja [18], [19] nakažeta možnost zamenjave Almansijevih deformacij z dejanskimi logaritmskimi deformacijami, iz katerih nadalje izračunavamo Cauchyjeve napetosti. Prav tako predstavita uporabo algoritma za korekcijo debeline elementa v sodobnih opreah (Ansys). Do logaritmskih deformacij pridemo s pomočjo uporabe deformacijskega gradienta (glej enačbe 2.8, 2.9, 2.10).

Podobno kot pri izpeljavi TLF-elementa tudi tokrat v enačbe vstavimo pomike po polju elementa s pomočjo interpolacijskih funkcij (glej enačbo 3.60) ter predstavimo vektor vozliščnih pomikov.

$${}^t\mathbf{U} = \begin{bmatrix} {}^tu_1^1 \\ {}^tu_2^1 \\ {}^tu_1^2 \\ {}^tu_2^2 \\ {}^tu_1^3 \\ {}^tu_2^3 \\ {}^tu_1^4 \\ {}^tu_2^4 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} u_1^1 \\ u_2^1 \\ u_1^2 \\ u_2^2 \\ u_1^3 \\ u_2^3 \\ u_1^4 \\ u_2^4 \end{bmatrix} \quad (3.118)$$

Zapisano lahko ponovno vstavimo in združimo v enačbi PVD.

$$\int_{tV} \delta \mathbf{u}^T \cdot ({}^t\mathbf{L})^T \cdot \mathbf{D} \cdot ({}^t\mathbf{L}) \cdot \mathbf{u} \, {}^t dV + \int_{tV} \delta \mathbf{u}^T \cdot {}^t\mathbf{L}_n \cdot {}^t\boldsymbol{\sigma} \cdot {}^t\mathbf{L}_n \cdot \mathbf{u} \, {}^t dV = {}^{t+\Delta t}R - \int_{tV} \delta {}^t\boldsymbol{\varepsilon}^T \cdot {}^t\boldsymbol{\sigma} \cdot {}^t dV \quad (3.119)$$

$$\int_{t_V} \delta U^T \cdot {}^t\mathbf{B}_L^T \cdot {}^t\mathbf{D} \cdot {}^t\mathbf{B}_L \cdot \mathbf{U} \cdot {}^t dV + \int_{t_V} \delta U^T \cdot {}^t\mathbf{B}_{NL}^T \cdot {}^t\boldsymbol{\sigma} \cdot {}^t\mathbf{B}_{NL} \cdot \mathbf{U} \cdot {}^t dV = {}^{t+\Delta t}\mathbf{R} \quad (3.120)$$

$$- \int_{t_V} {}^t\mathbf{B}_L^T \cdot {}^t\boldsymbol{\sigma} \cdot {}^t dV$$

Člen virtualnih pomikov lahko okrajšamo in iz izpeljanih členov v naslednjem koraku zapišemo enačbe za izračun togostne matrike.

$${}^t\mathbf{K}_L = \int_{t_V} {}^t\mathbf{B}_L^T \cdot {}^t\mathbf{D} \cdot {}^t\mathbf{B}_L \cdot {}^t dV \quad (3.121)$$

$${}^t\mathbf{K}_{NL} = \int_{0_V} {}^t\mathbf{B}_{NL}^T \cdot {}^t\boldsymbol{\sigma} \cdot {}^t\mathbf{B}_{NL} \cdot {}^0 dV \quad (3.122)$$

$${}^t\mathbf{F} = \int_{0_V} {}^t\mathbf{B}_L^T \cdot {}^t\boldsymbol{\sigma} \cdot {}^t dV \quad (3.123)$$

Zapišemo lahko še vodilno enačbo v inkrementu z vstavljenimi togostnimi matrikami.

$$({}^t\mathbf{K}_L + {}^t\mathbf{K}_{NL}) \cdot \mathbf{U} = {}^{t+\Delta t}\mathbf{R} - {}^t\mathbf{F} \quad (3.124)$$

Za implementacijo programske kode togostnih matrik je na koncu treba definirati še gradientne matrike.

$${}^t\mathbf{B}_L = ({}^t\mathbf{L}) \cdot \mathbf{N} \quad (3.125)$$

$${}^t\mathbf{B}_{NL} = ({}^t\mathbf{L}_n) \cdot \mathbf{N} \quad (3.126)$$

$${}^t\mathbf{B}_L = \begin{bmatrix} \frac{\partial N_1}{\partial {}^t x_1} & 0 & \frac{\partial N_2}{\partial {}^t x_1} & 0 & \frac{\partial N_3}{\partial {}^t x_1} & 0 & \frac{\partial N_4}{\partial {}^t x_1} & 0 \\ 0 & \frac{\partial N_1}{\partial {}^t x_2} & 0 & \frac{\partial N_2}{\partial {}^t x_2} & 0 & \frac{\partial N_3}{\partial {}^t x_2} & 0 & \frac{\partial N_4}{\partial {}^t x_2} \\ \frac{\partial N_1}{\partial {}^t x_2} & \frac{\partial N_1}{\partial {}^t x_1} & \frac{\partial N_2}{\partial {}^t x_2} & \frac{\partial N_2}{\partial {}^t x_1} & \frac{\partial N_3}{\partial {}^t x_2} & \frac{\partial N_3}{\partial {}^t x_1} & \frac{\partial N_4}{\partial {}^t x_2} & \frac{\partial N_4}{\partial {}^t x_1} \end{bmatrix} \quad (3.127)$$

$${}^t\mathbf{B}_{NL} = \begin{bmatrix} \frac{\partial N_1}{\partial^t x_1} & 0 & \frac{\partial N_2}{\partial^t x_1} & 0 & \frac{\partial N_3}{\partial^t x_1} & 0 & \frac{\partial N_4}{\partial^t x_1} & 0 \\ \frac{\partial N_1}{\partial^t x_2} & 0 & \frac{\partial N_2}{\partial^t x_2} & 0 & \frac{\partial N_3}{\partial^t x_2} & 0 & \frac{\partial N_4}{\partial^t x_2} & 0 \\ 0 & \frac{\partial N_1}{\partial^t x_1} & 0 & \frac{\partial N_2}{\partial^t x_1} & 0 & \frac{\partial N_3}{\partial^t x_1} & 0 & \frac{\partial N_4}{\partial^t x_1} \\ 0 & \frac{\partial N_1}{\partial^t x_2} & 0 & \frac{\partial N_2}{\partial^t x_2} & 0 & \frac{\partial N_3}{\partial^t x_2} & 0 & \frac{\partial N_4}{\partial^t x_2} \end{bmatrix} \quad (3.128)$$

S tem lahko zaključimo z izpeljavo posodobljene Lagrangeeve formulacije, saj imamo vse potrebno za implementacijo programske kode. V matrikah so sicer še vedno odvodi glede na kartezijske koordinate, ki pa jih zlahka pretvorimo v naravne koordinate s pomočjo Jacobijeve matrike (glej izpeljavo TLF).

Metoda ima nekatere bistvene prednosti pred uporabo TLF. Prva stvar, ki jo opazimo, je, da nimamo dveh členov linearne togostne matrike (ni efekta začetnih pomikov). Poleg tega je polje napetosti izraženo s pomočjo Cauchyjevega tenzorja, ki si ga lahko fizikalno interpretiramo in zato ne potrebujemo nobenih transformacij. Seveda pa ima metoda slabost, saj v vsaki iteraciji odvajamo po nazadnje izračunanih koordinatah, kar predstavlja določen računski napor.

3.2.1 Povezava med formulacijama

Pri programiranju algoritmov se pogosto zgodi, da je do enakih rezultatov možno priti na več različnih načinov. Podobno se zdi pri upoštevanju geometrijske nelinearnosti, kjer lahko vpeljemo povezave med obema formulacijama, ki povezujejo transformacije Cauchyjevih napetosti, ki jih uporabljamo pri ULF, ter napetosti 2PK in obratno.

$${}^tS_{ij} = J \cdot F_{ik}^{-1} {}^t\sigma_{kl} F_{jl}^{-1} \quad (3.129)$$

Z ustrezno transformacijo koordinat lahko prikažemo še transformacijo konstitutivnega dela v enačbi:

$${}^tC_{ijkl} = \frac{1}{J} \cdot F_{ia} F_{jb} {}^0C_{abcd} F_{kc} F_{ld} \quad (3.130)$$

Pri tem je v literaturi pogost zapis izotropnega materialnega zakona s pomočjo Lamejevih konstant, ki jih lahko pogosto razberemo v literaturi.

$${}_0 C_{ijkl} = \lambda \delta_{ij} \delta_{kl} + \mu (\delta_{ik} \delta_{jl} + \delta_{il} \delta_{jk}) \quad (3.131)$$

Pri tem vemo, da je δ znak za Kroneckerjevo delto, ki ima lastnosti klasičnega indeksnega zapisa.

V primeru, da smo upoštevali zgornje transformacije, moramo po obeh formulacijah TLF in ULF priti do enakega rezultata. S kinematičnega vidika lahko tako s TLF kot ULF zajamemo vse učinke nelinearnosti (veliki pomiki, velike rotacije, velike deformacije), a vendar je za odgovor na vprašanje, ali bo formulacija dejansko možna, treba odgovoriti še s povezavo konstitutivnega zakona. To pomeni, da morata biti vektor zunanjih obremenitev in togostna matrika v določenem računskem koraku povsem enaka. Pri tem se prav tako pojavi vprašanje, katera formulacija je torej bolj primerna za razvoj algoritma. Izkaže se, da sta v praksi obe metodi uporabljani, kjer pa se po navadi odločimo za metodo, ki bo za dani konstitutivni zakon računsko najhitrejša. Če imamo torej zvezo, ko bo konstitutivni zakon na voljo v obliki 2PK-napetosti, se odločimo za TLF in obratno. Zaradi zgoraj naštetih prednosti smo se v tem delu osredotočili na razvoj algoritmov po posodobljeni Lagrangeevi formulaciji.

3.2.2 Upoštevanje materialne nelinearnosti

To podpoglavje bo predstavljeno zelo na kratko, saj tudi v implementaciji kode materialni nelinearnosti v tem zaključnem delu nismo posvečali posebne pozornosti. Na kratko bo torej predstavljena še formulacija »Material nonlinear only«, ki sicer v praksi redko pride v poštev, a vendar lahko zelo nazorno predstavimo princip formuliranja materialne nelinearnosti v KE.

Formulacija »Material nonlinear only« ne povzema geometrijskih nelinearnosti, ki smo jih opisovali v preteklih dveh poglavjih. To lahko vidimo tudi v osnovni enačbi, iz katere izhajamo glede na [7], kjer lahko vidimo, da je integracija predvidena prek geometrijske domene, ki se zaradi infinitizimalnih pomikov deformacij ne spremeni bistveno. Prav tako lahko zapišemo napetosti s pomočjo Cauchyjevega tenzorja napetosti zaradi veljave naslednje zveze:

$${}^{t+\Delta t}{}_0 S_{ij} = {}^{t+\Delta t} \sigma_{ij} \quad (3.132)$$

$$\int_V C_{ijrs} \cdot \Delta(\varepsilon_{rs}^{(k)}) \cdot \delta(\varepsilon_{ij}) dV = {}^{t+\Delta t}R - \int_V {}^{t+\Delta t}\sigma_{ij}^{(k-1)} \cdot \delta \varepsilon_{ij} dV \quad (3.133)$$

Ker smo zveze, ki pojasnijo nastanek gradientne matrike, ki vsebuje odvode interpolacijskih funkcij, ter izpeljavo s pomočjo virtualnih pomikov do te točke že večkrat nakazali, lahko v naslednjem koraku zapišemo enačbo še v iteracijski obliki, kjer opazimo, da se bo togostna matrika pri MNO-analizi sicer spreminjala, kar pa bo zgolj posledica prispevka materialne matrike. Gradientna matrika namreč skozi celotno simulacijo predstavlja konstantne vrednosti.

$${}^t\mathbf{K} \cdot \Delta \mathbf{U}^{(k)} = {}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(k-1)}, \quad (3.134)$$

kjer togostno matriko in vektor notranjih obremenitev izračunamo na zelo podoben način kot pri linearnem končnem elementu.

$${}^t\mathbf{K} = \int_V \mathbf{B}_L^T \cdot \mathbf{D} \cdot \mathbf{B}_L \cdot dV \quad (3.135)$$

$${}^t\mathbf{F} = \int_V \mathbf{B}_L^T \cdot \boldsymbol{\sigma} \cdot dV \quad (3.136)$$

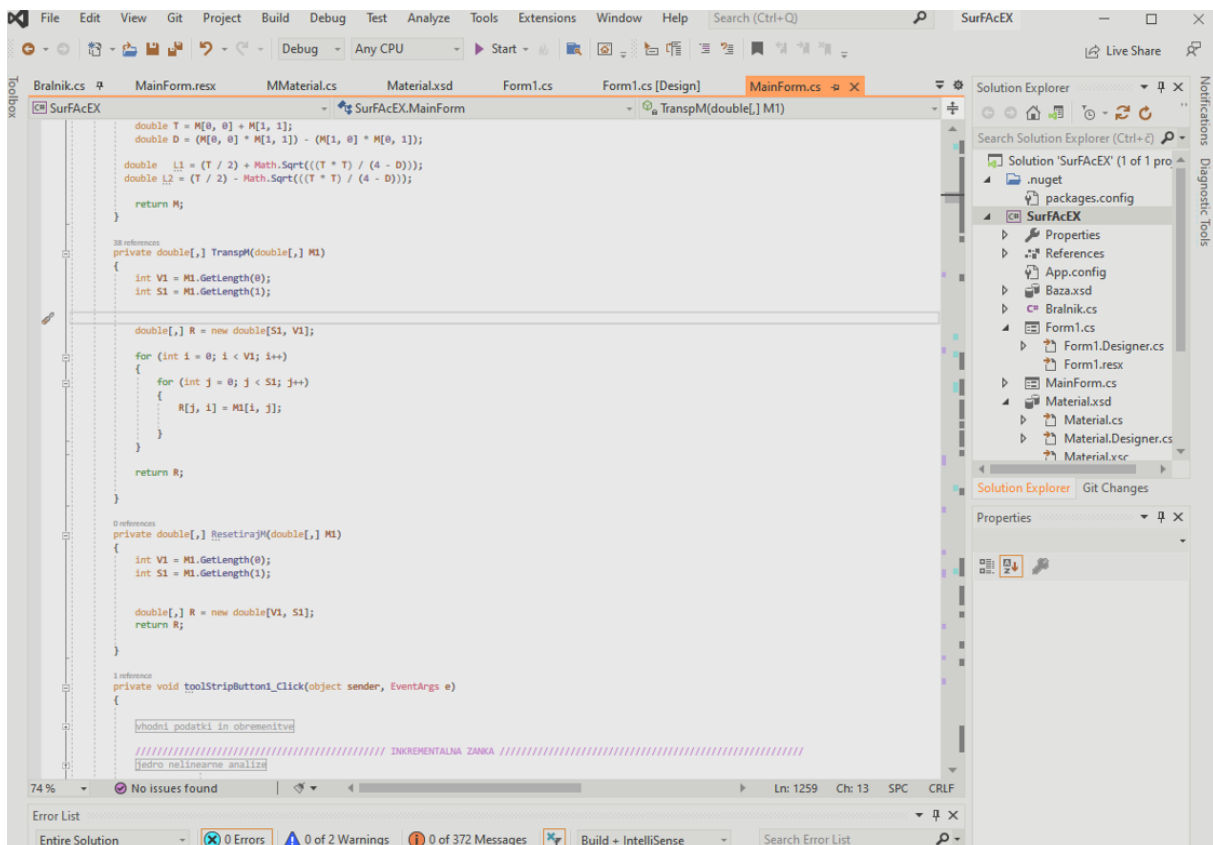
3.2.3 Konvergenčni kriterij

Pri numeričnih metodah se pogosto zavedamo, da je izračunavanje rešitve popolnoma natančno teoretično nemogoče in zaradi inženirske vrednosti tudi nesmiselno. Posledično pri implementaciji reševalnika pogosto posegamo po definiranju zaustavitvenega oz. konvergenčnega kriterija. Avtorja [14] zelo lepo prikažeta zaustavitvene kriterije glede na absolutno napako (razlika med izračunano vrednostjo in pravo vrednostjo) in relativno napako (je razmerje med absolutno napako in pravo vrednostjo). Ker gre za enostavne demonstrativne primere simulacij, smo se za to nalogo odločili za preprost absolutni kriterij, kjer bo torej predpisan maksimalni absolutni odstopok sile, ki bo izračunan kot razlika med notranjimi in zunanji silami.

$$|{}^{t+\Delta t}\mathbf{R} - {}^{t+\Delta t}\mathbf{F}^{(k-1)}| \leq Error = 0,01 \quad (3.137)$$

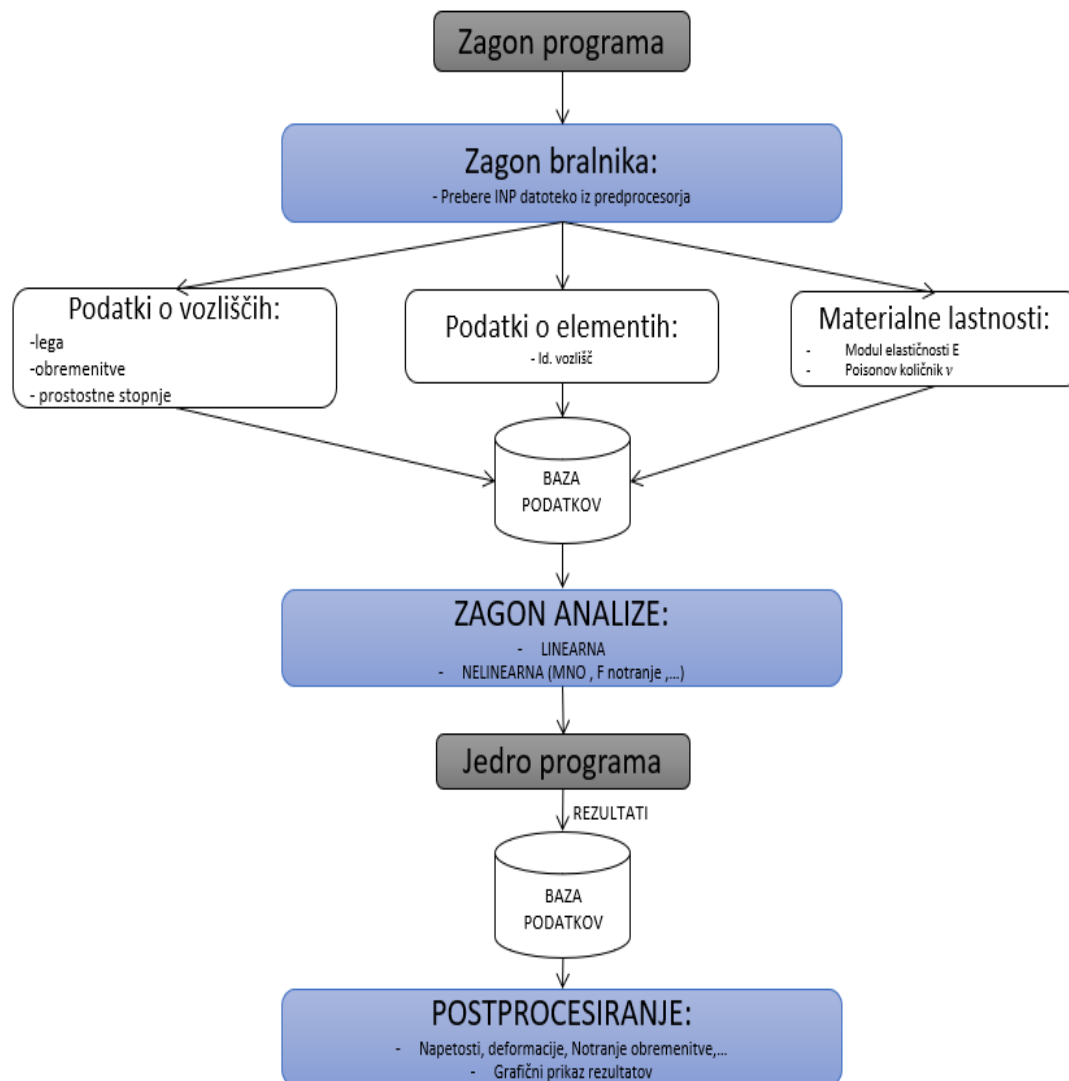
4 IMPLEMENTACIJA KODE

V tem delu sta vse programiranje in razvoj kode potekala v objektno orientiranem programskem jeziku C#, ki ga je razvilo podjetje Microsoft. Gre za jezik, ki je nastal z zgledovanjem po zgradbi jezikov C, C++, Java itd. C# je do uporabnika prijazen in se ga da relativno hitro priučiti. Zaradi orientacije po objektnem programiranju je na aplikacijah zelo enostavno graditi, medtem ko je sintaksa zelo čitljiva in pregledna. Glavno prednost tega jezika zagotovo predstavlja .NET platforma, za katero je jezik sam po sebi oblikovan. S kombinacijo z .NET Framework, ki na nek način predstavlja knjižnico, kjer najdemo veliko rešitev, postanemo pri kodiranju veliko bolj učinkoviti, saj lahko kombiniramo lastno kodo z ogrođjem, ki nam pomaga z razvojem uporabniškega vmesnika, numeričnih algoritmov itd. Ker je C# prav tako zelo razširjen jezik, se je skozi čas razvila velika skupnost uporabnikov s spletnimi portali, kjer je možno najti izjemno veliko rešitev za nastale probleme.

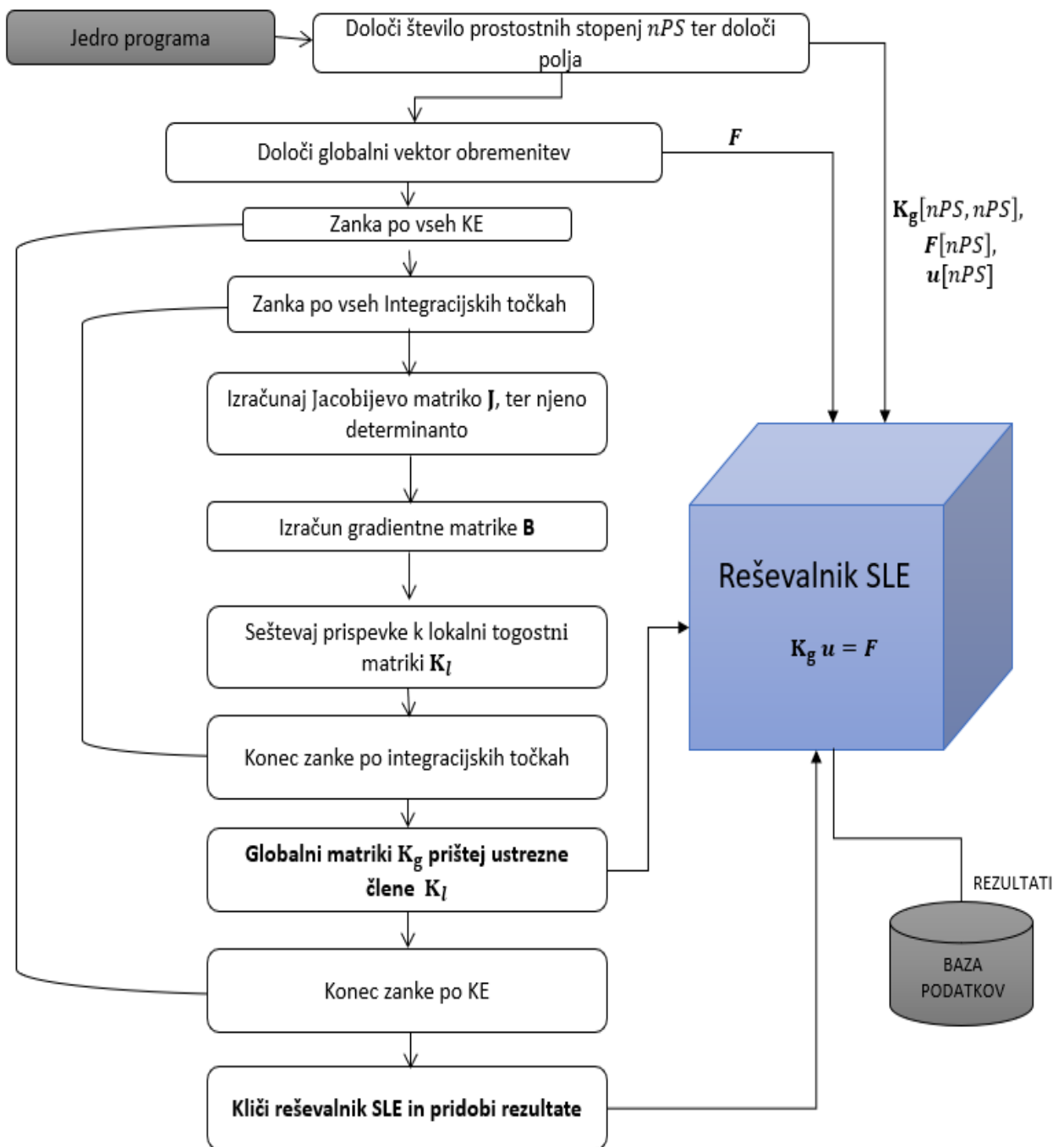


Slika 4.1: Razvojno okolje MS Visual studio

V razvoju programske kode je zagotovo najpomembnejši del numerično reševanje izpeljane diferencialne enačbe po metodi končnih elementov. Gre torej za sestavo algoritma, ki bo reševal do sedaj teoretično predstavljeno problematiko. Jedro programa predstavljajo pravilna obravnava posameznih končnih elementov, numerična integracija ter ustrezna sestava globalne togostne matrike, ki vodi v reševalnik sistema linearnih enačb. Na prvi pogled se zdi, da bo največji računski napor predstavljalo reševanje sistema enačb, kar zagotovo drži, a kljub temu niso zanemarljivi proces grajenja vodilnih globalnih matrik (matrika obremenitev, togostna matrika itd.) ter postprocesiranje in prikaz rezultatov. Delovanje programa in jedro numeričnega reševanja lahko idejno še kar najbolje predstavimo s pomočjo naslednjih diagramov poteka.



Slika 4.2: Diagram poteka delovanja programa



Slika 4.3: Diagram poteka jedra programa

Pri tem diagram poteka jedra programa konceptualno prikazuje obravnavo linearne analize, medtem ko so pri izvedbi nelinearne analize prisotne zgolj določene spremembe, kot sta določevanje obremenitev in določevanje nelinearnih delov matrik. Te minimalne spremembe so bile že pojasnjene s pomočjo psevdokode (glej poglavje 3.2.2).

4.1 Bralnik

V uvodu naloge je bila sklenjena predpostavka, ki predvidi že opravljeno predprocesiranje numerične simulacije v programskem paketu Abaqus. Da bi torej lahko izvajali numerične simulacije, je treba razviti algoritem, ki bo skrbel za prenos predprocesiranih podatkov, ki jih v Abaqusu izvozimo s pomočjo formata INP. Omenjeni format predstavlja vstopne podatke za numerično simulacijo in je v formatu tekstovne datoteke, ki jo je treba prebrati in razčleniti.

```
*Heading
** Job name: Plosca_luknja Model name: Model-1
** Generated by: Abaqus/CAE 2017
*Preprint, echo=NO, model=NO, history=NO, contact=NO
**
** PARTS
**
*Part, name=Part-1
*Node
   1,      0.,      26.25
   2,     27.5,      26.25
   3,     27.5,      35.
   4,      0.,      35.
   5,     55.,      26.25
   6,     27.5,      25.
   7,     35.,      17.5
   8,     55.,      17.5
   9,      0.,      8.75
  10,     27.5,      8.75
  11,     27.5,      10.
  12,     20.,      17.5
  13,      0.,      17.5
  14,     55.,      8.75
  15,     27.5,      0.
  16,     55.,      0.
  17,     55.,      35.
  18,      0.,      0.
  19,  1.01851857,      26.25
  20,  2.03703713,      26.25
  21,  3.05555558,      26.25
  22,  4.07407427,      26.25
  23,  5.09259272,      26.25
  24,  6.11111116,      26.25
  25,  7.12962961,      26.25
  26,  8.14814854,      26.25
  27,  9.16666698,      26.25
```

Slika 4.4: Oblika zapisa INP

Razvit je torej bil algoritem, ki deluje po principu pregleda besedila celotne datoteke in zaznava področja podatkov, ki jih potrebujemo za izvajanje simulacij, ter podatke prenese oz. zapiše v podatkovno bazo. Podatkov o vozliščih in elementih ni težko identificirati, saj so v zelo očitni obliki zapisani v gručah, medtem ko je vredno omeniti, da se za namen definiranja podpor in obremenitev v programski opremi Abaqus definirajo skupine vozlišč (N-set), ki imajo neke posebne lastnosti, in so na njih po navadi predpisane obremenitve (v našem primeru točkovne sile) ter podpore (prostostne stopnje vpetih vozlišč). Algoritem lahko zgolj idejno prikažemo s pomočjo psevdokode v nekaj vrsticah.

```
// Vhodni podatki
```

- Določi ključ za prepoznavo vozlišč, elementov, materialnih podatkov.
- Določi ključ za prepoznavo posameznih skupin vozlišč (N-set) za prepoznavo robnih pogojev ter obremenitev.

```
// Zanka ponavljaljaj (1) za vsako vrstico v tekstovni datoteki
```

- Preveri ključ za prepoznavo vozlišč.

```
// Zanka ponavljaljaj (2), če je ključ pravilen
```

- Prepoznavaj posamezno vozlišče.
- Zapiši vozlišče v podatkovno bazo.

```
//Konec zanke (2)
```

- Preveri ključ za prepoznavo elementov.

```
// Zanka ponavljaljaj (3), če je ključ pravilen.
```

- Prepoznavaj posamezni element.
- Element zapiši v podatkovno bazo.

```
//Konec zanke (3)
```

..... **Nadaljuj za vse ostale želene podatke.....**

```
//Konec zanke (1)
```

4.2 Uporabniški vmesnik

Uporabniški vmesnik je bil izdelan z mislijo o karseda enostavni uporabniški izkušnji ter hitri in enostavni izvedbi simulacij. Dodane ima nekatere elemente, ki služijo testiranju in demonstraciji implementiranih metod. Tabele z vsemi prikazi o podatkih za vsako vozlišče in končni element so namreč odlična rešitev za razvoj metode in hitro dostopanje in razbiranje podatkov. Težava nastane pri obravnavi večjih sistemov z zelo velikim številom končnih elementov, kjer je ves sprotni grafični prikaz zelo zamuden in hkrati nepotreben. Za ta korak smo se odločili zgolj zaradi namena programa, ki je namenjen prikazu in razvoju algoritma. Ob morebitni optimizaciji metode bi podatke zlahka skrili v ozadje, kar pa bi rezultiralo v potrebi po razvoju dodatnega algoritma, ki bi klical in prikazoval točno določene zelene podatke v fazi poprosesanja. Uporabniški vmesnik je tako sestavljen iz funkcij, ki so namenoma obarvane z modro barvo, ki ponazarja interakcijo z uporabnikom, medtem ko praviloma vsi predmeti, obarvani z rumeno barvo, služijo zgolj prikazovanju. Izjemoma lahko v številke 1, 2, 3 za izjemno enostavne primere vnašamo podatke tudi ročno, kar pa je izjemno zamudno in nepraktično opravilo.

1. Preglednica služi prikazu podatkov, ki se tičejo vozlišč. Gre za vhodne podatke (lega, obremenitev, prostostne stopnje vozlišča). Prav tako so dodane koordinate lege vozlišč ob času t , ki se skozi nelinearno simulacijo posodablja.
2. Preglednica služi prikazu elementov in pripadajočih vozlišč. Prav tako so dodani rezultati napetosti in deformacij ob času t , ki se med simulacijo posodablja. Napetosti in deformacije so zapisane za vse integracijske točke v elementu.
3. Preglednica služi prikazu materialnih lastnosti, razbranih iz bralnika. Podatke za materialno nelinearnost je treba vnesti ročno.
4. Številka predstavlja tekstovno datoteko z glavnimi informacijami o analizi (tip analize, število PS, število KE itd.) ter vrstico, v kateri je prikazan napredek simulacije.

5. Preglednica prikazuje vse rezultate, ki se nanašajo na vozlišča KE. Prikazani so torej pomiki, napetosti, deformacije, reakcijske sile, reziduali sil itd.
6. Številko predstavlja risalnik s pripadajočo barvno lestvico, ki grafično prikazuje rezultate iz preglednice 5. Omeniti je treba, da risalnik deluje po principu trikotnih elementov. Dejansko imamo opravka s štirivozliščnimi KE.
7. Polje predstavlja osnovne ukaze, s katerimi lahko zaženemo bralnik in preberemo datoteke INP ali pa ročno spremenjene podatke shranimo. Z ukazom »refresh« se polja izpraznijo in posodobijo.
8. Ukazi predstavljajo zagon želene analize. Na voljo imamo linearno, MNO- in ULF-analizo. Omenimo lahko, da sta na voljo nelinearni analizi le z inkrementi ter analiza, ki deluje s pomočjo inkrementov in iteracij.
9. Ukazi v tem polju predstavljajo nadzor nad natančnostjo in hitrostjo nelinearne analize. Določamo lahko število inkrementov, maksimalno dovoljeno število iteracij, dovoljeno toleranco odstopanja in debelino elementa.
10. V zadnjem polju najdemo ukaze za risalnik. Izbiramo lahko prikaz z različnim skaliranjem pomikov, deformacij ali napetosti.

7 Open ... Save Refresh AnalyzeLinear Analyze_MaterialOnly Material_Data ULF_INCREMENTS ULF_INC_INTF 8 RNS

Nodes

	Vozisce	X	Y	Fx	Fy	AdrX
1	5	0				0
2	0	0				0
3	5	45				1
4	60	45		5000		3
5	60	50		5000		5
6	0	50				7
7	5	40				9
8	5	35				11
9	5	30				13
10	5	25				15
11	5	20				17

Thickness: 1 Num_inc 100 ABS Konv 0,01 Max iter 1000

Elements

	Element	N1	N2	N3
1	2	1	14	41
2	40	10	9	26
3	25	3	24	31
4	30	20	19	25
5	6	44	3	44
6	43	7	3	44
7	42	8	7	42
8	41	9	8	42
9	39	11	10	40

Plot Clear Plot_Deformed Scaling factor 10 Exx S U

Material

	E	Poisson	t
1			

100%

7.011e+02
6.134e+02
5.258e+02
4.382e+02
3.505e+02
2.629e+02
1.753e+02
8.763e+01
0.000e+00

6

Results

	Vozisce	Ux	Uy	U	Defcx	Defcy	Defxy	Sigxx	Sigyy	Sigxy
1	0	0	0	0	0.452133280446...	-0.44464083521...	31301.53480015...	104338.4493338...	-35913...	
2	0	0	0	0	-0.43379347159...	-0.44128591683...	-30031.8557260...	-100106.185753...	-35642.32405...	
3	-179.134066564...	20.30277212418...	180.2809373172...	0.177371831237...	0.277112868482...	0.241280229676...	60116.69810362...	76228.71181233...	19488.01855...	
4	-167.543001507...	661.0595554516...	681.9606977004...	0.019259276914...	-1.03693764685...	0.005502629594...	4444.448446871...	1333.334316304...	444.4431595...	
5	-233.443616751...	661.0595554464...	701.0675131889...	-0.01925927752...	-1.03693764685...	0.005502625492...	-4444.44873089...	-1333.33483702...	444.4428282...	
6	-220.921769004...	-20.3827235270...	221.8600537243...	-0.16183747385...	-0.16375836180...	0.001275288640...	-48684.2267082...	-48994.5239921...	103.0040824...	
7	-141.803036682...	18.10802522156...	142.9545444885...	-0.02190254214...	0.446897785201...	0.026494682256...	25884.64463485...	101613.9282827...	2139.955105...	
8	-108.524873223...	15.83379427217...	109.6738672119...	0.000461560907...	0.453461151162...	-0.00461679571...	31499.97836682...	104676.8352541...	-372.8950388...	
9	-79.7419960823...	13.57341370994...	80.88895783074...	-0.00343535926...	0.452317491915...	0.000804598231...	30521.51268621...	104143.1271081...	64.98678026...	
10	-55.3773140829...	11.31061935301...	56.52058939174...	-0.00275683201...	0.452516624544...	-0.00014082280...	30691.88200238...	104236.0557550...	-11.37414927...	

5

Revilo vozilšč = 44 ::: Element, type=CPS4 :::Število elementov = 21
 ::IZVEDBA LINEARNE ANALIZE
 Revilo PROSTOSTNIH STOPENJ: 84
 :::210000
 Poisson=0,3

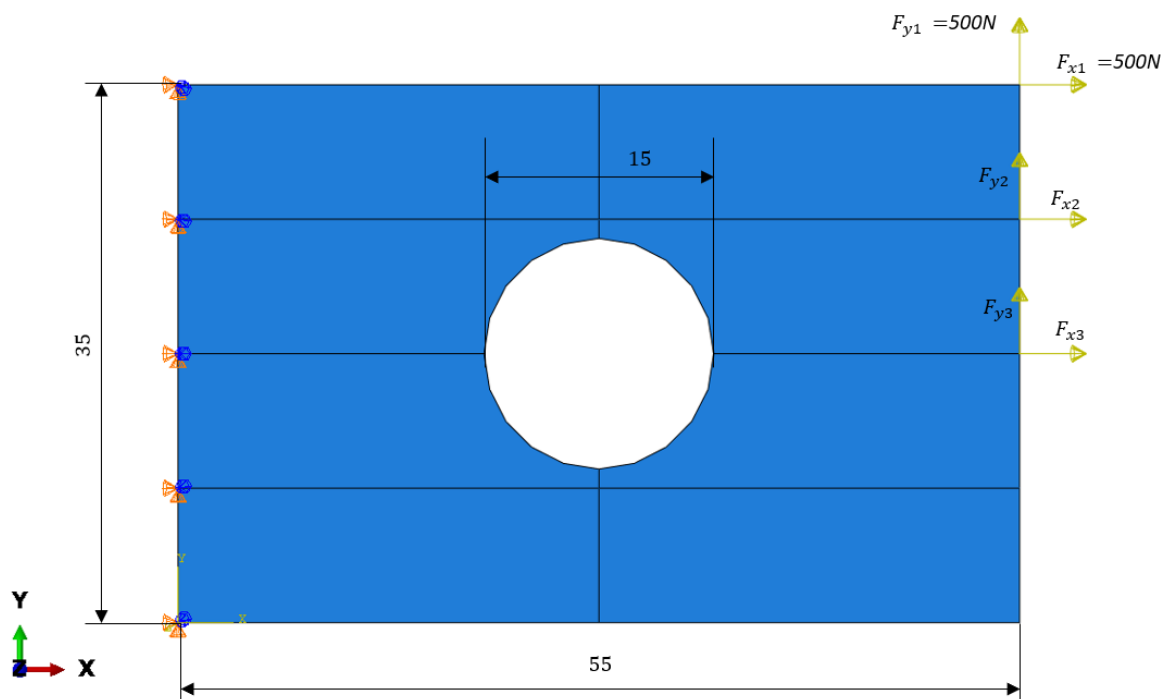
Slika 4.5 Uporabniški vmesnik

5 REZULTATI SIMULACIJ

V tem poglavju bodo predstavljeni rezultati numeričnih simulacij, kjer je namen predstaviti razliko med upoštevanjem različnih nelinearnosti ter preveriti pravilnost reševanja sistemov. Tako bodo najprej predstavljeni rezultati linearne simulacije, ki jim bodo sledili predstavljeni primeri, ki so se v literaturi uveljavili kot merilo »benchmark«, s katerimi lahko demonstriramo različne primere z velikimi deformacijami, pomiki ali pa materialno nelinearnostjo. Prav tako bomo vzporedno primerjali rezultate s programsko opremo Abaqus ter najdeno literaturo avtorjev, ki so se že srečali s temi problemi. Primerjali bomo enake modele in rezultate z našim programom. Sistemi so z velikostjo omejeni, saj je v naši programski rešitvi implementiran lasten enostaven elementarni neposredni reševalnik linearnih enačb, kar pa z večanjem sistema rezultira v zelo dolgem času simulacije.

5.1 Linearni odziv

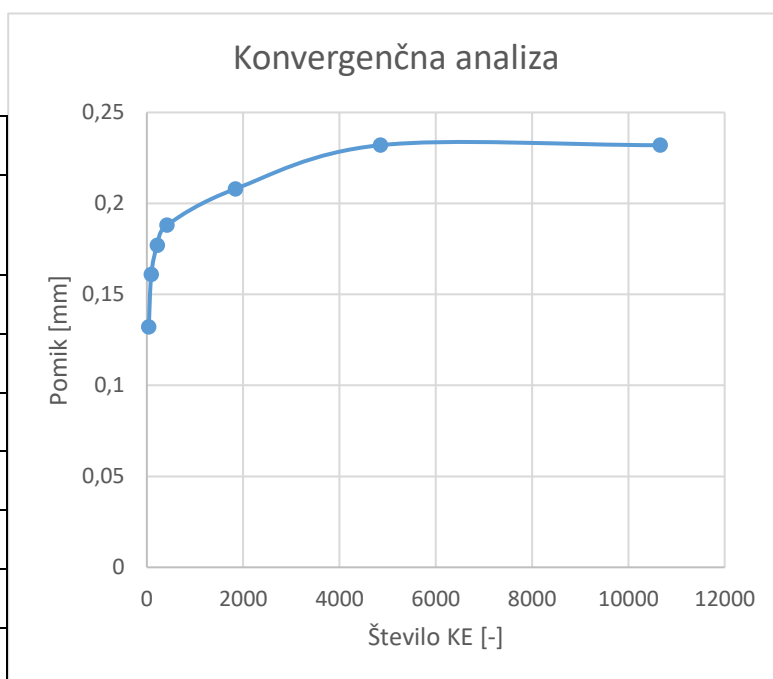
Linearna analiza za RNS je bila izvedena na plošči z luknjo, ki je togo vpeta na eni strani v točkah, ki so na naslednji sliki obarvane temno modro. V teh točkah so vpete vse prostostne stopnje, medtem ko so na drugi strani plošče vidne tri točkovne sile, ki so med seboj povsem enake in znašajo v vsaki posamezni X- in Y-smeri obremenitev velikosti 500 N ($F_1 = F_2 = F_3$). Konstitutivni model je linearno elastičen z modulom elastičnosti $E = 210.000 \text{ Mpa}$ ter Poissonovim količnikom $\nu = 0,3$. Model je bil narejen s pomočjo programske opreme Abaqus, z gaberitnimi merami, ki jih lahko vidimo kotirane na naslednji sliki z dodanimi particijami, s pomočjo katerih lahko kreiramo strukturirano mrežo KE.



Slika 5.1: Plošča z luknjo

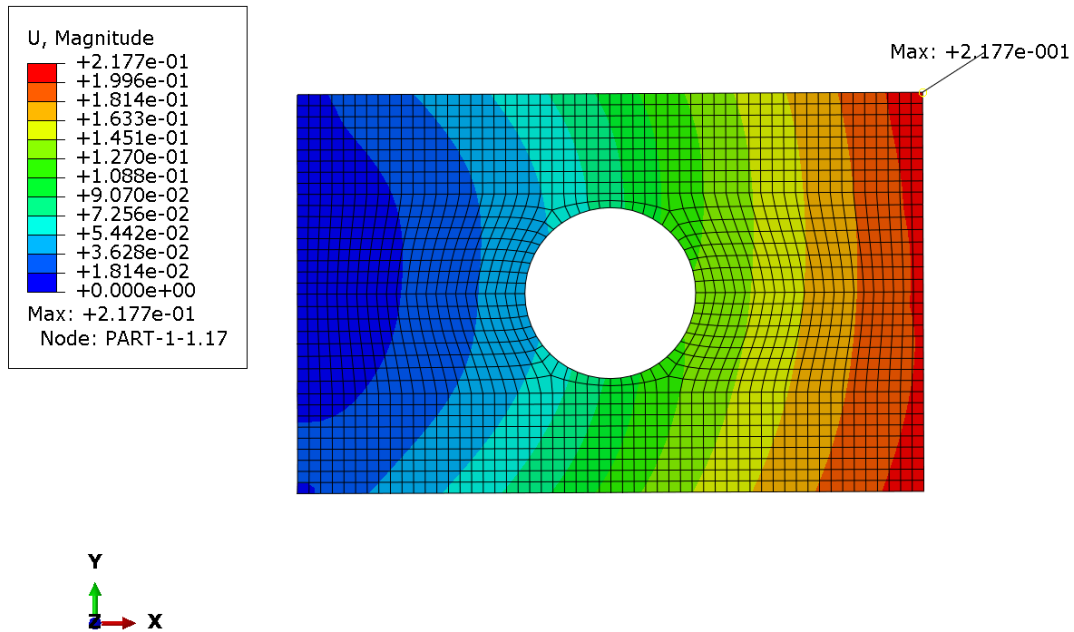
Analiza je bila prav tako izvedena z različno velikostjo mreže KE, kjer pri izvedeni konvergenčni analizi opazimo ustalitev pomikov pri mreži z 4.847 KE.

Konvergenčna analiza		
Velikost KE [mm]	Število KE [-]	Pomik [mm]
10	40	0,132
5	92	0,161
3	216	0,177
2	416	0,188
1	1840	0,208
0,6	4847	0,232
0,4	10660	0,232

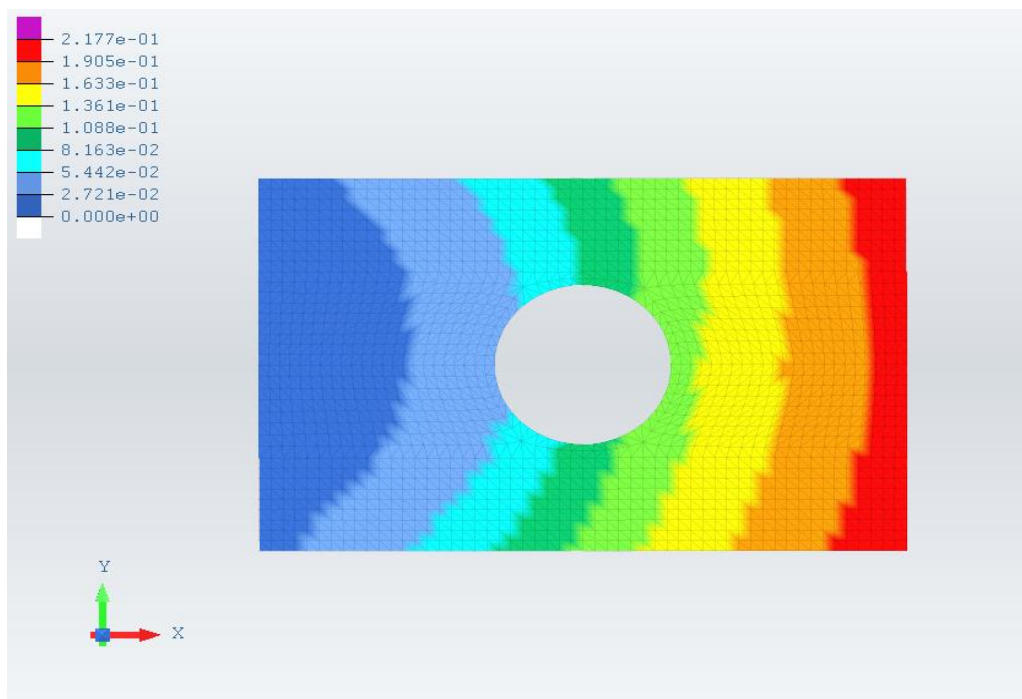


Slika 5.2 Konvergenčna analiza plošče z luknjo

V prvi fazi lahko predstavimo rezultate analize, katerih bistvo je validacija pravilnosti reševanja sistema, kjer opazimo, da v primerjavi s programsko opremo Abaqus dobimo povsem enake rezultate. V skrajni točki, ki je prikazana na sliki, torej opazimo največji absolutni pomik plošče velikosti 0,218 mm.

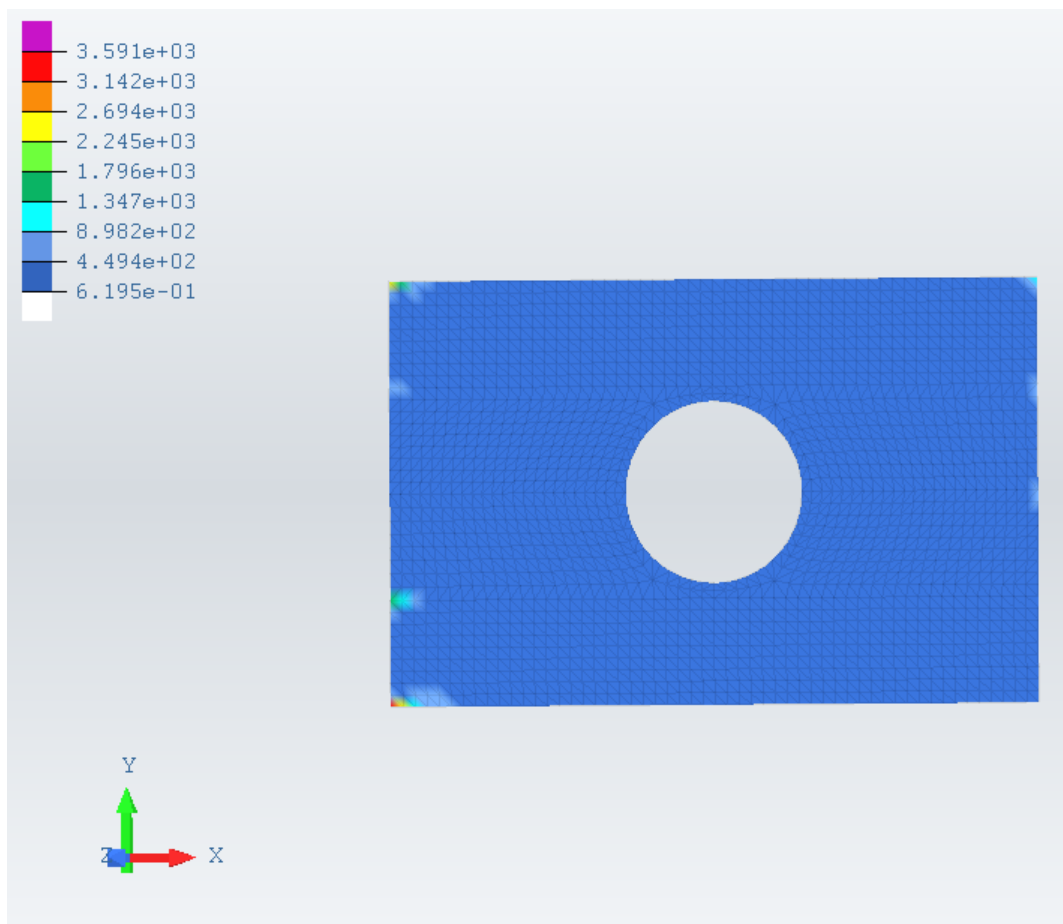


Slika 5.3: Pomiki plošče z luknjo, pridobljeni s programsko opremo Abaqus



Slika 5.4: Pomiki lastne programske opreme

Povsem enake rezultate lahko predstavimo tudi po poprocesiranju, kjer so napetosti v integracijskih točkah povsem enake. Abaqus sicer uporablja povprečenje napetosti v vozliščih, kar je posledica prispevkov sosednjih elementov, kjer se z implementacijo enakega algoritma povsem približamo tudi tem rezultatom. Zaradi enake slike obeh programov bomo tokrat predstavili zgolj napetosti lastne programske opreme, kjer vidimo povišane napetosti v točkah vpetja plošče, ki so sicer povsem nerealne, a so za potrebe validacije pravilnosti izračunavanja povsem uporabne.

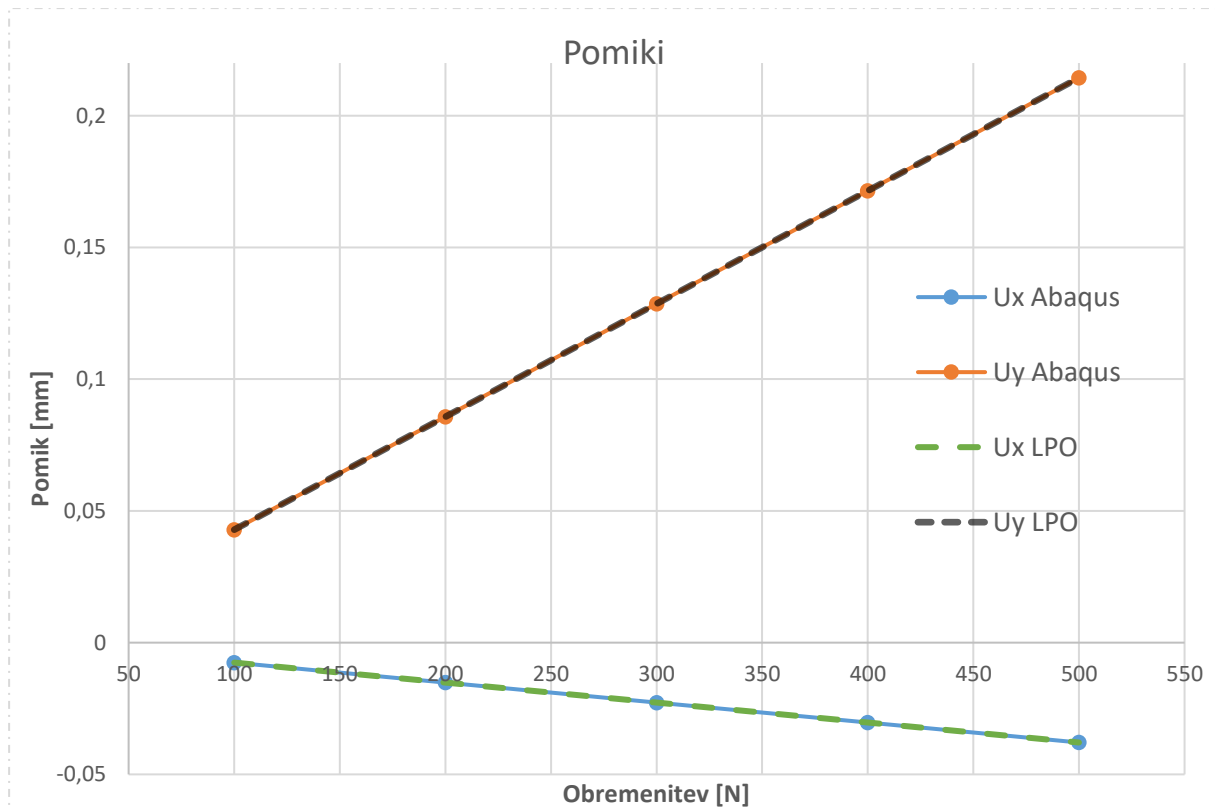


Slika 5.5: Napetosti v plošči z luknjo

Na koncu lahko linearni odziv plošče predstavimo še tabelarično, kjer vpeljimo kratico LPO, ki bo ponazarjala rešitve z lastno razvito programsko opremo. Pomike bomo spremljali v skrajni točki, ki je bila že predhodno predstavljena, saj so se v njej pojavili tudi največji pomiki. Napetosti pa smo za ta primer odčitali v prvi integracijski točki elementa, katerega del je prej opisana točka. Opazimo lahko nično odstopanje v primerjavi s programsko opremo Abaqus ter pričakovani linearni odziv z inkrementacijo obremenitve. Zaradi odsotnosti vseh linearnosti torej posledično rezultati linearno naraščajo in bi jih lahko za nadaljnje naraščanje obremenitve enostavno ekstrapolirali.

Tabela 2: Primerjava rezultatov za ploščo z luknjo

F [N]	Abaqus			LPO		
	Ux [mm]	Uy [mm]	Sigxx [Mpa]	Ux [mm]	Uy [mm]	Sigxx [Mpa]
100	-0,00757	0,0428	23,959	-0,00757	0,0428	23,959
200	-0,0151	0,0857	47,918	-0,0151	0,0857	47,918
300	-0,0227	0,128	71,877	-0,0227	0,128	71,877
400	-0,0303	0,171	95,836	-0,0303	0,171	95,836
500	-0,0378	0,214	119,795	-0,0378	0,214	119,795

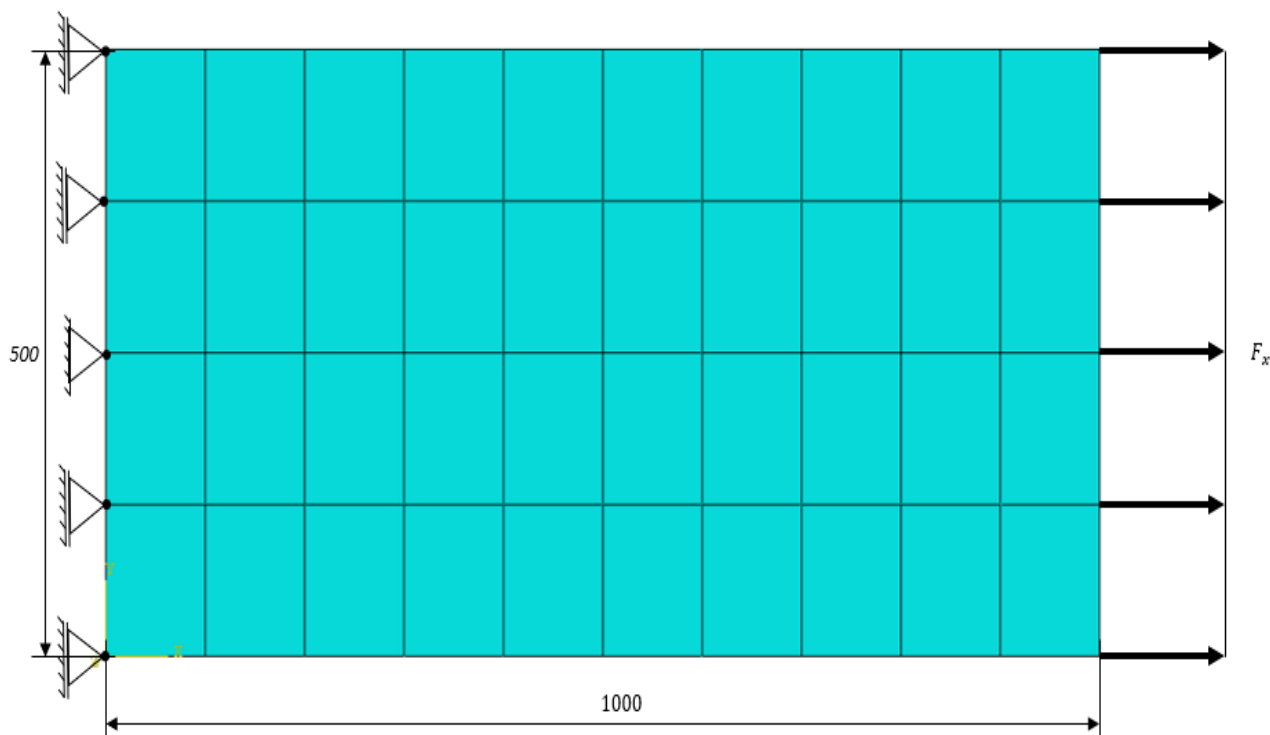


Slika 5.6: Primerjava pomikov plošče z luknjo

5.2 Materialna nelinearnost

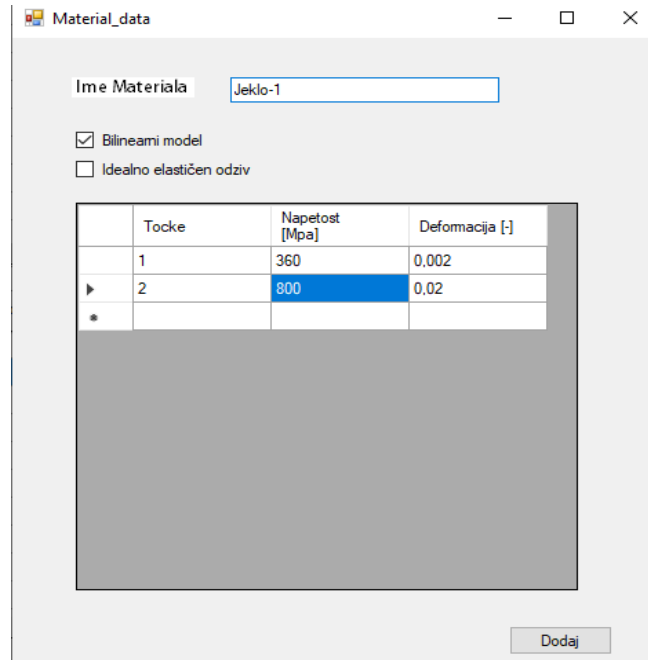
Ker smo se v tem delu osredotočili predvsem na geometrijsko nelinearnost, bo materialna nelinearnost predstavljena le na krajšem elementarnem problemu, s pomočjo katerega lahko razumemo princip upoštevanja materialne nelinearnosti v simulacijah. Za RNS bomo torej

kreirali model natezno obremenjene plošče, po kateri pričakujemo pojav konstantnih napetosti po celotni površini plošče brez upoštevanja geometrijskih nelinearnosti (glej izpeljavo »Material only«). Plošča je na eni strani vpeta, kot je prikazano na naslednji sliki, medtem ko obremenitev predstavlja porazdeljena obremenitev F_x različnih velikosti, ki bodo kasneje prikazane v rezultatih. Model je bil narejen s pomočjo programske opreme Abaqus in je zamrežen s 10 x 4 enako velikih končnih elementov velikosti (100 x 125 mm).

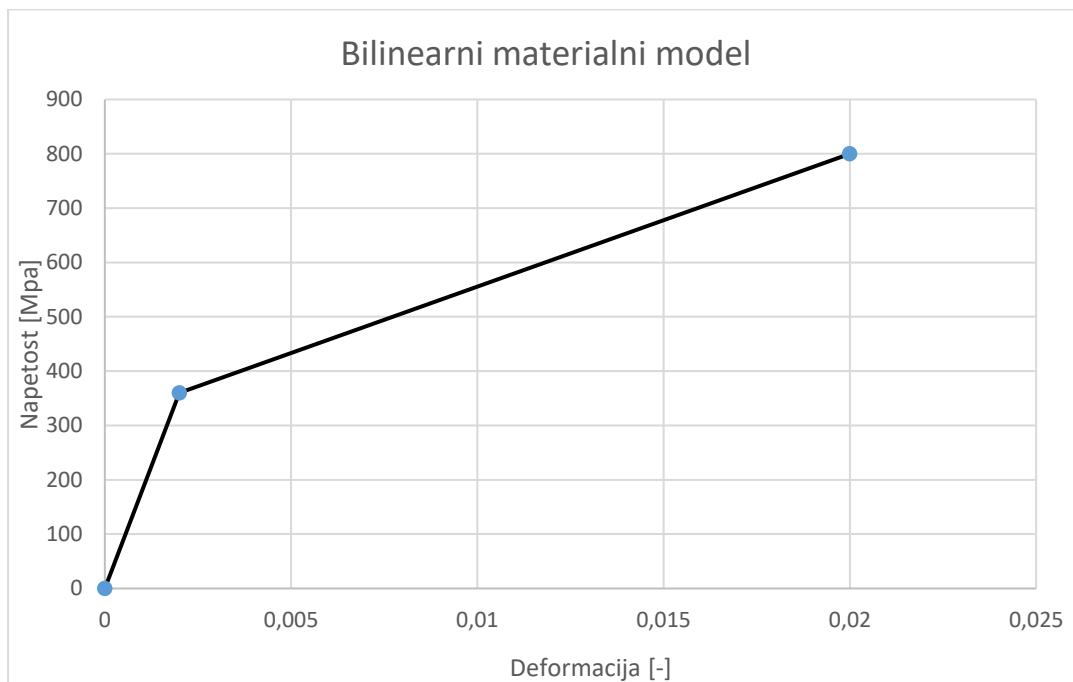


Slika 5.7: Model materialne nelinearnosti

Konstitutivni zakon je predstavljen s pomočjo bilinearnega modela, ki je sestavljen iz dveh delov z različnim modulom elastičnosti. Posledično lahko pričakujemo tudi podoben odziv rezultatov. Poissonov količnik znaša $\nu = 0,3$. Algoritem v programu je poenostavljen na raven enega končnega elementa, kjer se napetost in posledično konstitutivni zakon povprečita glede na celotni KE. V primeru natezno obremenjene plošče je opis povsem zadosten. Kar najbolj natančne rezultate bi dobili s pomočjo zelo natančnega opisa materialnega modela z veliko vnesenimi točkami, podobno kot je to praksa številnih že obstoječih programskih oprem. Prva točka ima privzete vrednosti napetosti in deformacij, ki znašajo 0. Vnesene so bile torej točke, ki so prikazane na naslednji sliki.



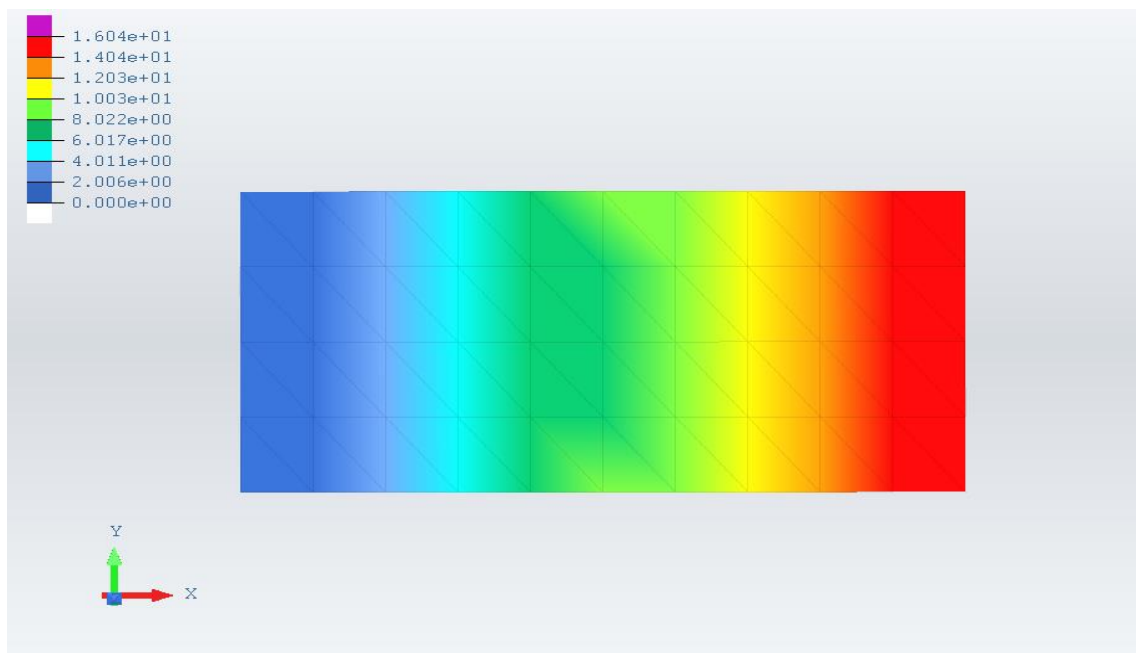
Slika 5.8: Vnesene točke materialnega modela



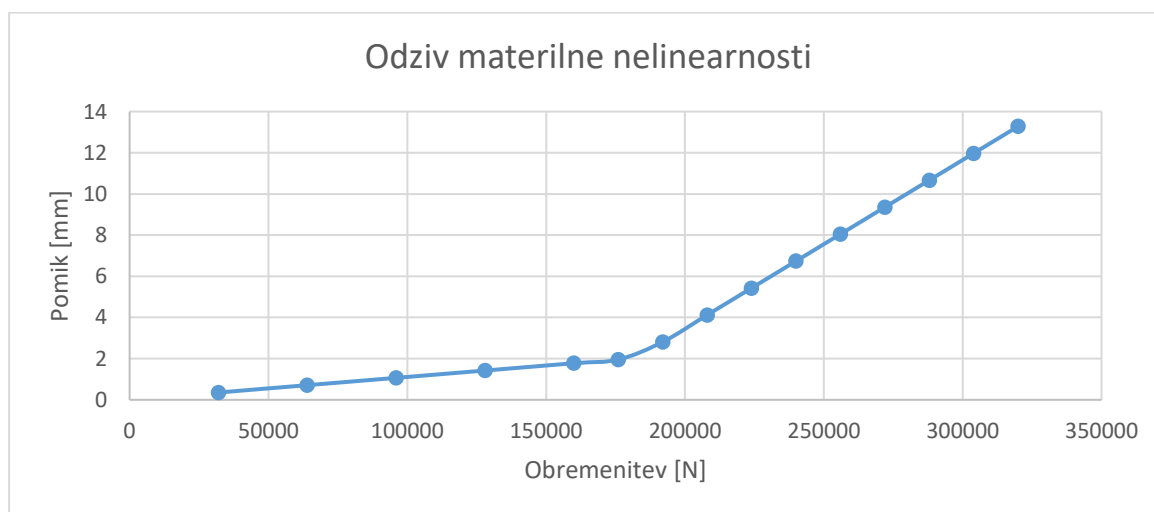
Slika 5.9: Bilinearni materialni model

Sredica plošče se zaradi vpetja pričakovano pomakne za največjo razdaljo. Na površini vpetja sil opazimo, da se pomiki zgladijo in da se pomiki po prerezu z razdaljo od vpetja vedno manj razlikujejo. Če pogledamo napetosti po plošči, opazimo, da se pojavijo največje ravno v smeri X velikosti $\sigma_{xx} = 640 \text{ Mpa}$, medtem ko se plošča v smeri X prav tako povsem enakomerno deformira. Omenjene deformacije dosežejo 1,6 %. Z inkrementi obremenitve lahko opazimo,

da se pričakovano formirata dva različna materialna modula, v odvisnosti katerih pomik linearno narašča. V prvi fazi obremenitev zato opazimo veliko bolj tog odziv.



Slika 5.10: Pomiki za odziv materialne nelinearnosti



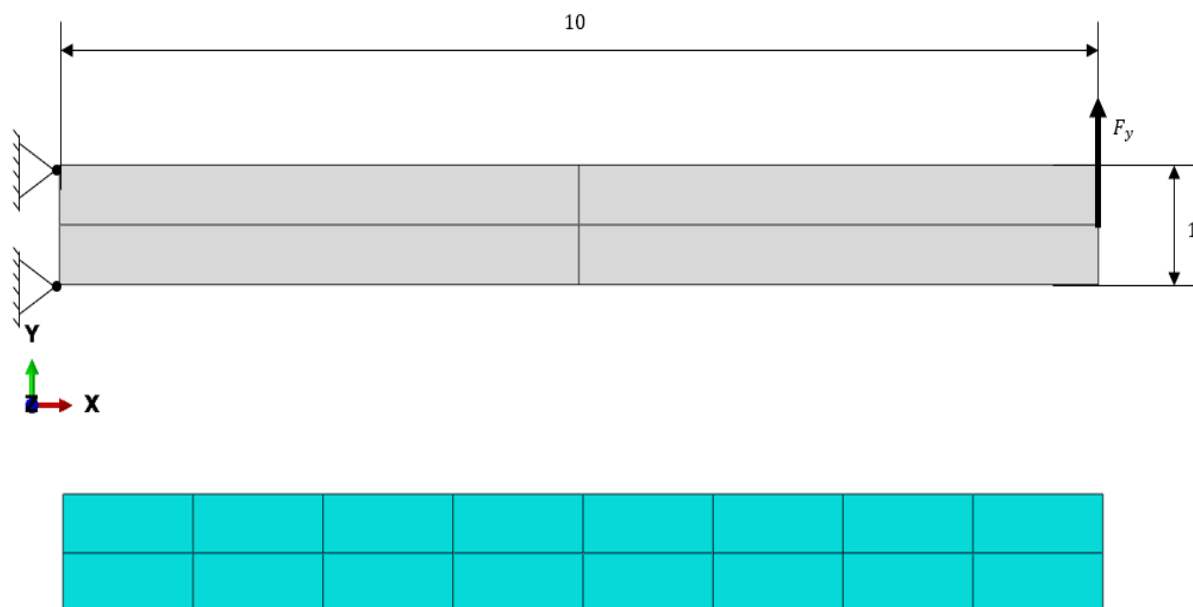
Slika 5.11: Naraščanje pomika »MNO«

5.3 Geometrijska nelinearnost

Geometrijsko nelinearne analize za RNS so bile izvedene na več primerih po vzgledih avtorjev [20], [21], ki sicer razvijata nove končne elemente, ki delujejo na principu brez mrežnih metod. Kljub temu pa v svojih člankih nove končne elemente primerjata z

navadnimi štirivozličnimi izoparametričnimi elementi. Primere, ki so jih torej predstavili avtorji, bomo ponovili ter primerjali dobljene rezultate. Prav tako so bile vzporedno za vsak obremenitveni korak ponovljene tudi numerične simulacije s programsko opremo Abaqus, ki pa uporablja logaritmčne, dejanske deformacije, zato so rezultati pri velikih pomikih, rotacijah, a majhnih deformacijah skoraj enaki, medtem ko pri velikih deformacijah začnejo nastajati odstopanja. Zelo je torej pomembno, kakšne mere uporabljamo za merjenje deformacij in napetosti, saj se izkaže, da so Almansijsve deformacije ob primeru, ko v elementu nastanejo tudi zares velike deformacije, neuporabne.

5.3.1 Pravokotno obremenjen nosilec



Slika 5.12: Prvi primer geometrijske nelinearnosti: pravokotno obremenjen nosilec

Prvi primer geometrijske nelinearnosti za RNS lahko predstavimo na nosilcu, ki je na eni strani togo vpet (blokirane so vse prostostne stopnje), medtem ko imamo na drugi strani nosilca v sredini prijemališče pravokotno delujoče točkovne sile, kot je vidno na sliki 5.12. Silo F_y na model apliciramo z variacijo obremenitev vse od 0 pa do 45 enot. V vsakem koraku silo povečujemo za 5 enot. Pri tem je model dolg 10 enot, višina ter širina pa znašata eno

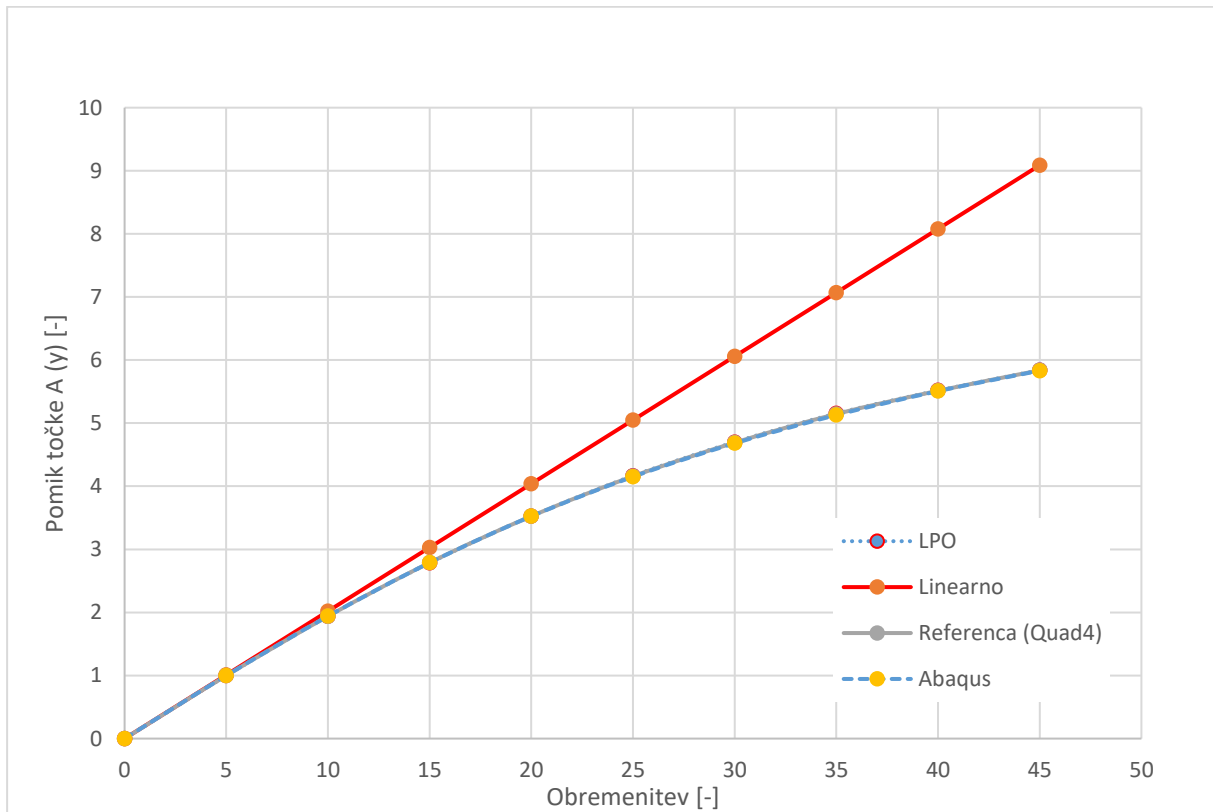
enoto. Konstitutivni model je linearno elastičen z modulom elastičnosti $E = 12.000$ ter Poissonovim količnikom $\nu = 0,2$. Model je bil narejen s pomočjo programske opreme Abaqus, kjer je potekalo tudi mreženje. Mrežo nosilca predstavlja 16 KE, kjer so vsi KE enake velikosti.

Tabela 3 Pomik pravokotno obremenjenega nosilca v smeri Y za točko prijema sile

Obremenitev [-]	Linear- no LPO	Nelinearno LPO	Sprememba [%] linearno- nelinearno	Abaqus	Referenca (Quad 4)
0	0	0	0,00	0	0
5	1,01	0,998	1,17	0,999	0,997
10	2,02	1,939	3,99	1,94	1,938
15	3,03	2,788	7,97	2,787	2,785
20	4,04	3,525	12,73	3,523	3,526
25	5,05	4,158	17,65	4,15	4,165
30	6,06	4,691	22,57	4,68	4,699
35	7,07	5,145	27,21	5,128	5,152
40	8,08	5,513	31,76	5,507	5,517
45	9,09	5,837	35,77	5,831	5,840

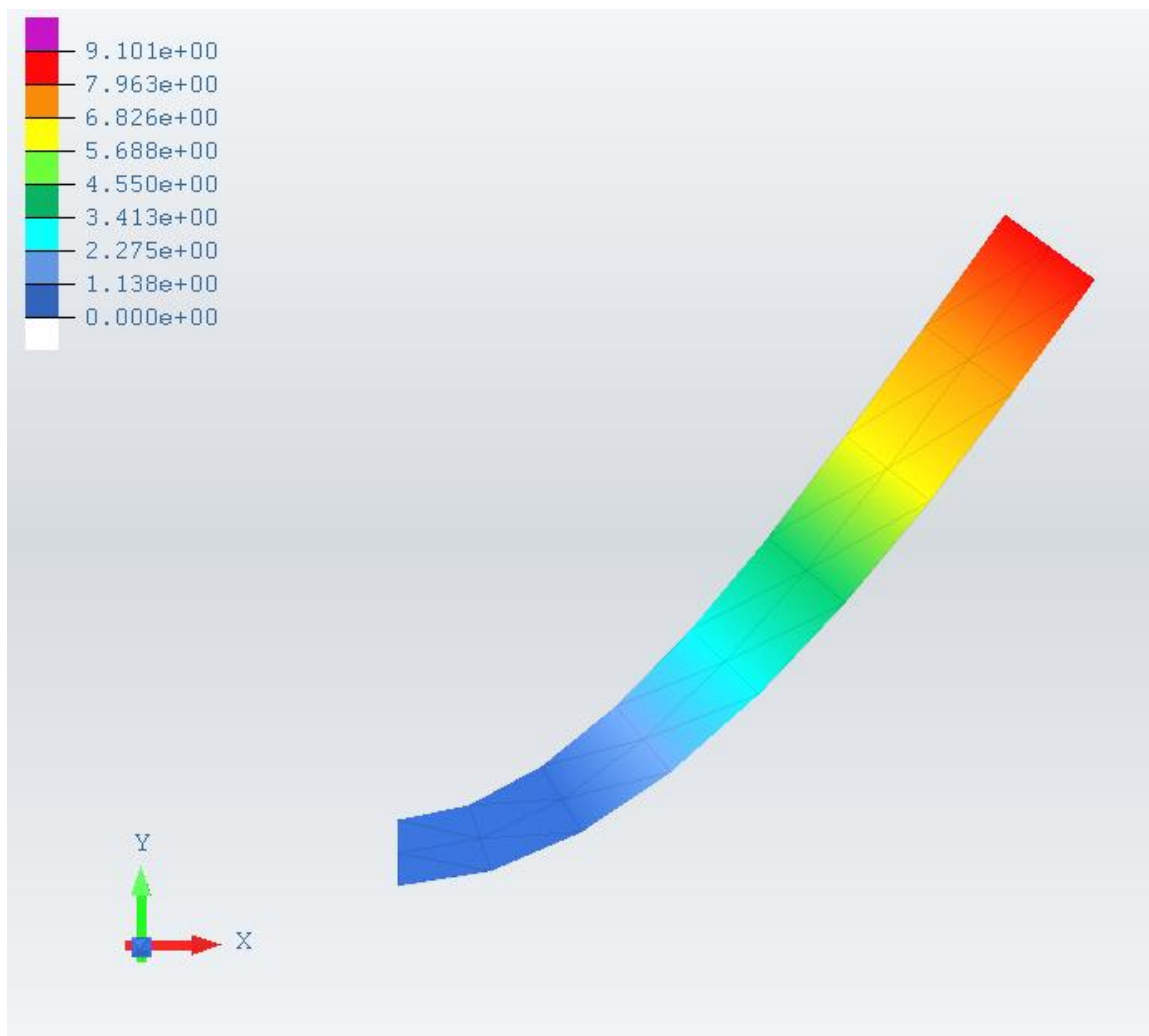
Opazimo lahko, da dobimo v primerjavi z referenco popolnoma enake rezultate. Razliko lahko pripišemo zgolj tolerančnemu kriteriju, ki je v naših izvedbah simulacij znašal 0,01. Podobno kot je bilo predstavljeno, je bil implementiran algoritem za absolutno preverjanje odstopanja sile. Kot vidimo, nastanejo v nosilcu za velike pomike in rotacije, medtem ko so končni elementi bolj ali manj ostali enakih velikosti. V tem primeru torej govorimo o majhnih deformacijah. V tem primeru so se torej Almansijsve deformacije izkazale za zelo uporabne, še posebej, če rezultate primerjamo z Abaqusovimi rezultati, kjer vemo, da so deformacije izražane s pomočjo logaritmskih oz. dejanskih deformacij. Do manjših odstopkov sicer pride, kar pa je bolj ali manj variacija različnih nastavitev velikosti inkrementov ter iteracij.

Zelo podobno sliko lahko vidimo na naslednjem grafu, kjer vidimo, da se vse tri simulacije pokrijejo. V oči pa takoj pade napaka, ki bi jo storili, če bi nosilec simulirali s pomočjo linearnih simulacij. V tabeli lahko vidimo, da bi zgrešili rezultate pomikov za kar 38 %. Z naraščanjem obremenitev bi se trend večanja razlike med obema simulacijama še povečeval.

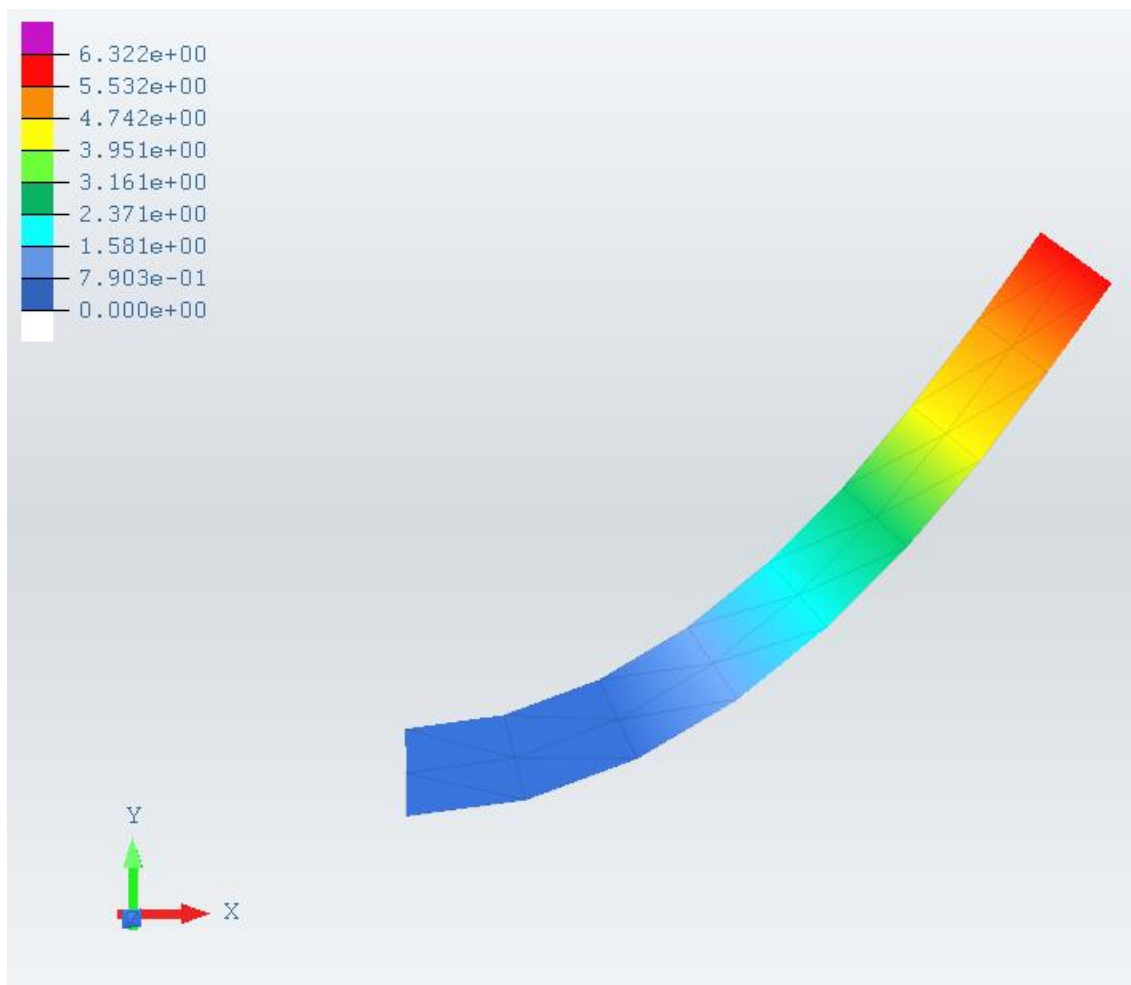


Slika 5.13: Upogib enostransko vpetega nosilca

Sledi slika, na kateri lahko vidimo pomike linearne simulacije, kjer je na risalniku nastavljen skalirni faktor 1. Prikazana so torej dejanska razmerja pomikov, kjer lahko vidimo, da pride do največjih pomikov na skrajnem robu vpetja nosilca. Govorimo o pomikih velikosti 9 enot. Zelo zanimiva je tudi razširitev nosilca, ki se zgodi povečano proti prijemašiču sile. Nekoliko drugačna slika se zdi pri obravnavi z nelinearno analizo, kjer razširitev nosilca ni več opažena. V primerjavi z linearno analizo pa v tem primeru govorimo o veliko manjših pomikih, saj se nosilec z večanjem obremenitve obnaša vedno bolj natezno obremenjeno, kar rezultira v povečani togosti.



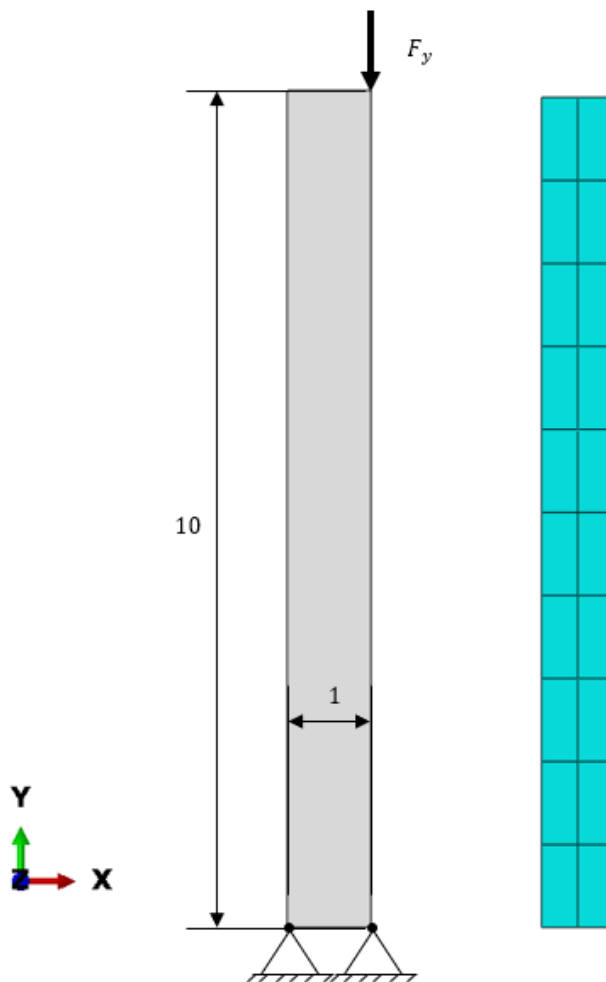
Slika 5.14: Linearna analiza LPO



Slika 5.15: Nelinearna analiza LPO

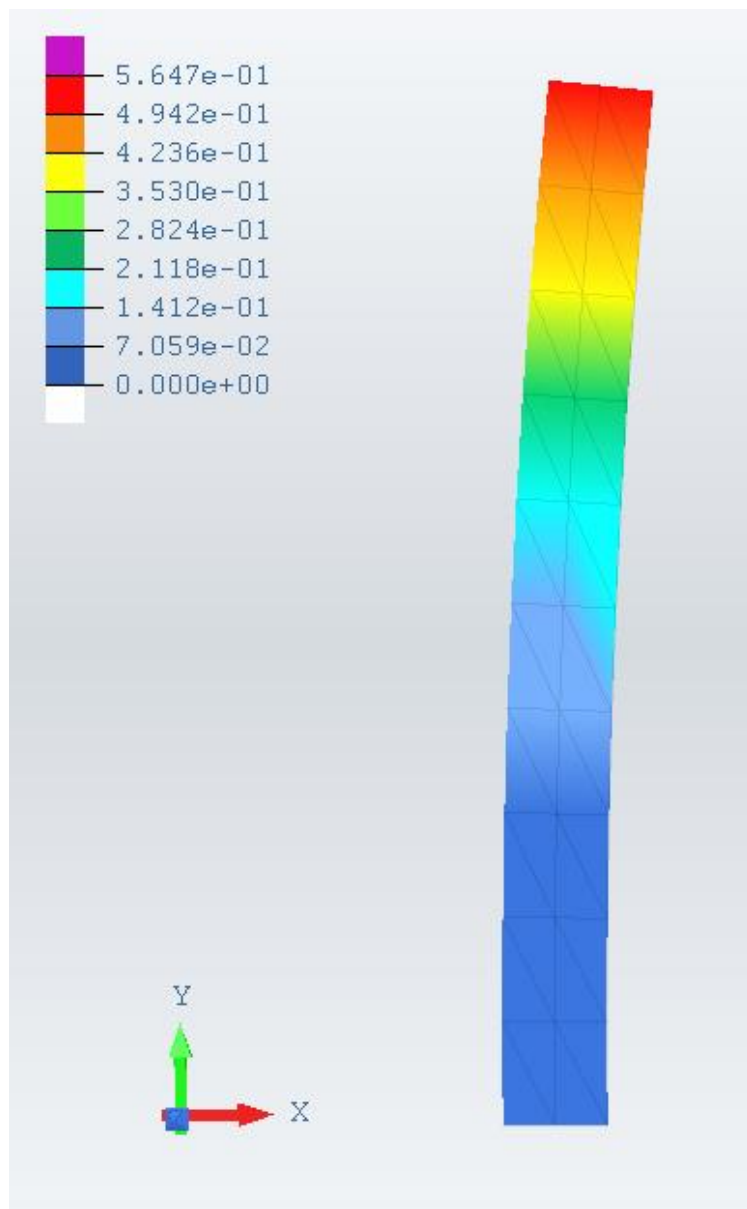
5.3.2 Ekscentrično obremenjeni nosilec

Naslednji primer nelinearne analize v RNS predstavlja ekscentrično obremenjen nosilec, ki je tokrat obrnjen pokončno v smeri Y. Na naslednji sliki lahko vidimo togo vpetje na dnu, kjer so vpete vse prostostne stopnje, medtem ko imamo na vrhu nosilca prijemališče sile, ki nosilec obremeni ekscentrično. Silo F_y tokrat na model apliciramo v nasprotni smeri osi z variacijo obremenitev vse od 0 pa do 2.800 enot. V vsakem koraku silo povečujemo za 200 enot. Pri tem je model visok 10 enot, višina ter širina pa podobno kot v prejšnjem primeru znašata eno enoto. Konstitutivni model je linearno elastičen z modulom elastičnosti $E = 1.000.000$ ter Poissonovim količnikom $\nu = 0$. Model je bil narejen s pomočjo programske opreme Abaqus, kjer je ponovno potekalo tudi mreženje. Mrežo nosilca tokrat predstavlja 20 enakih KE.



Slika 5.16: Ekscentrično obremenjen nosilec

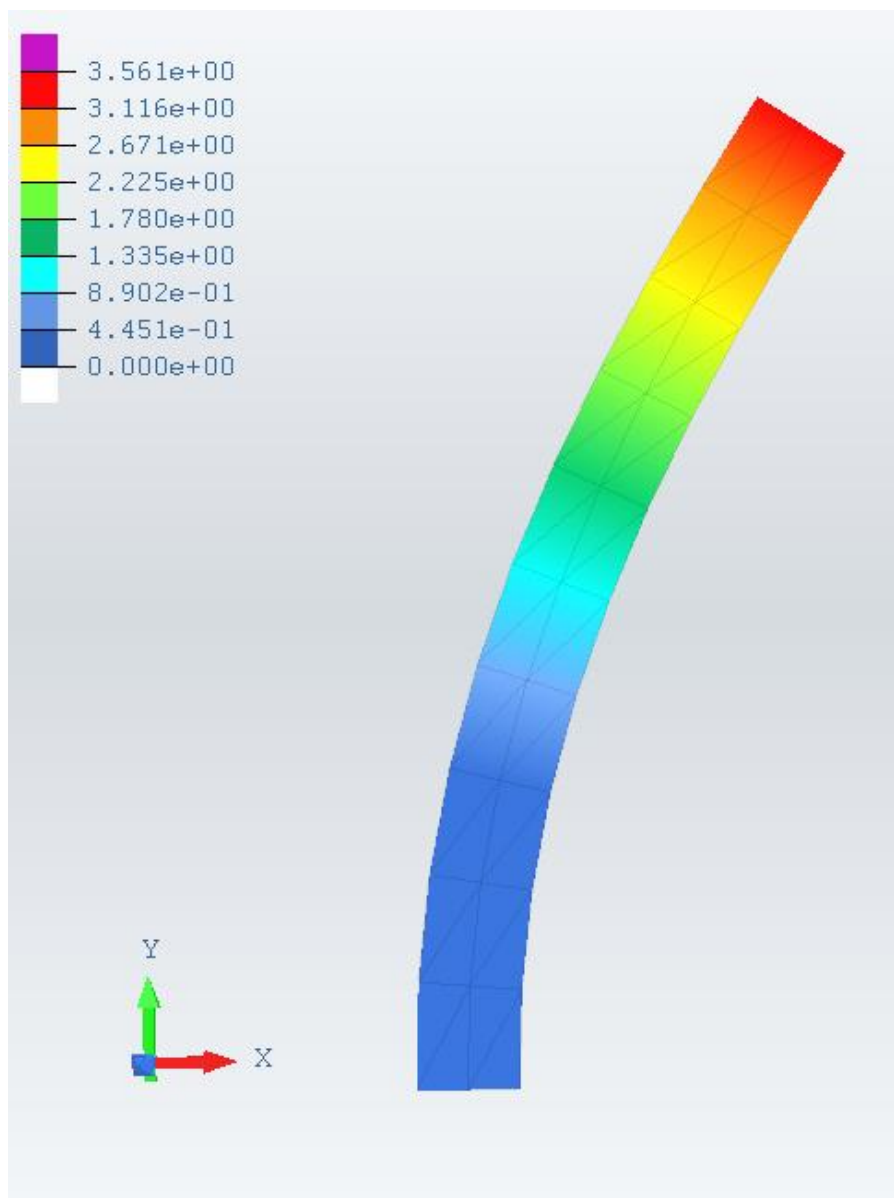
Že takoj po prvem zagonu linearne analize lahko opazimo, da bo tokrat z neupoštevanjem geometrijskih nelinearnosti storjena zelo velika napaka. Slika prikazuje nosilec, obremenjen z največjo obremenitvijo in skalirnim faktorjem 1. Z opaženimi pomiki, ki znašajo v točki prijemališča sile okrog 0,56 enote, lahko opazimo zelo tog odziv, kjer se zaradi vpliva navidezne tlačne obremenitve nosilec zelo malo odkloni.



Slika 5.17: Pomiki linearne analize ekscentrično obremenjenega nosilca LPO

Popolnoma drugačna slika se zdi pri analizi z upoštevanjem geometrijske nelinearnosti. Pomiki skrajne točke znašajo okrog 3,56 mm. Razlika je torej več kot očitna, kar lahko pripišemo odklonu prijemališča sile v horizontalni smeri. Z večanjem odklona se namreč povečuje tudi ročica, ki je neposredna posledica nastanka upogiba. Sledi primerjava rezultatov v tabelarični obliki, kjer lahko ponovno zabeležimo zelo dobro prileganje rezultatov tako z referenčnimi vrednostmi kot tudi s programsko opremo Abaqus. Razlike lahko v največji meri ponovno predpišemo izbiri števila inkrementov ter izbiri dovoljenega

tolerančnega območja. Za ta primer je bila dovoljena absolutna napaka 0,01 enote. Obremenitvam bi lahko pripisali velike pomike, medtem ko se elementi pretirano ne deformirajo. Ponovno je torej opis z izbiro Almansijskih deformacij ustrezen, kar potrjujejo tudi rezultati.



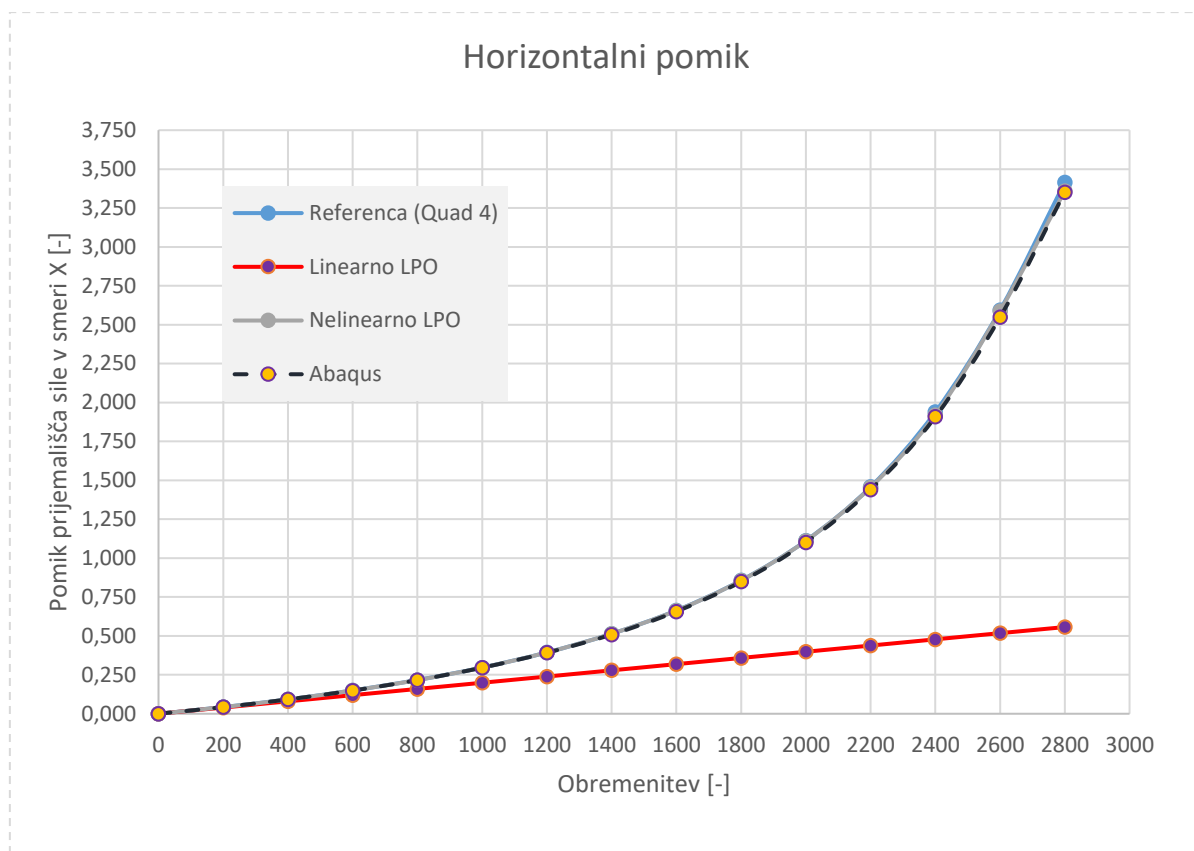
Slika 5.18: Pomiki nelinearne analize ekscentrično obremenjenega nosilca LPO

V primeru, da bi nosilec obravnavali samo z linearno analizo, bi lahko v horizontalni smeri pomika točke zabeležili 500 % spremembo rezultatov v primerjavi z nelinearno simulacijo. V

vertikalni smeri pomikov točke odstotek spremembe naraste na 3.800 %. Opis bi bil ponovno popolnoma neustrezen.

Tabela 4: Horizontalni pomiki točke prijemališča sile

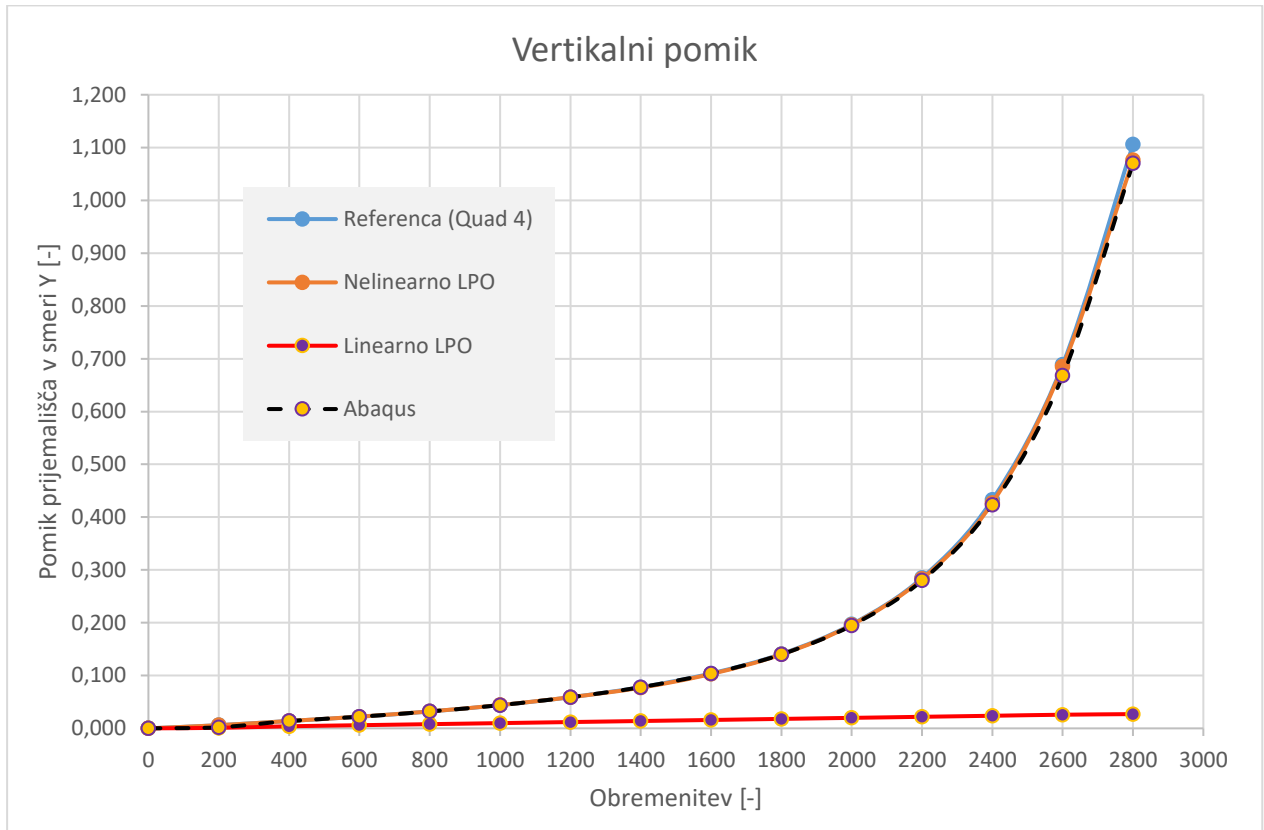
Obremenitev [-]	Linearno LPO	Nelinearno LPO	Sprememba [%] linearno-nelinearno	Abaqus	Referenca (Quad 4)
0	0,000	0,000	0	0,000	0,000
200	0,040	0,043	7	0,043	0,043
400	0,080	0,092	15	0,092	0,092
600	0,120	0,150	25	0,149	0,150
800	0,159	0,217	36	0,216	0,217
1000	0,199	0,297	49	0,296	0,297
1200	0,239	0,394	65	0,392	0,394
1400	0,279	0,514	84	0,509	0,514
1600	0,319	0,663	108	0,657	0,664
1800	0,359	0,856	139	0,849	0,857
2000	0,398	1,111	179	1,101	1,112
2200	0,438	1,455	232	1,440	1,458
2400	0,478	1,920	302	1,908	1,939
2600	0,518	2,588	399	2,548	2,594
2800	0,557	3,360	503	3,352	3,415



Slika 5.19: Horizontalni pomiki točke prijemališča sile

Obremenitev [-]	Linearno LPO	Nelinearno LPO	Sprememba [%] linearno-nelinearno	Abaqus	Referenca (Quad 4)
0	0,000	0,000	0	0,000	0,000
200	0,001	0,006	500	0,002	0,006
400	0,004	0,014	252	0,014	0,014
600	0,006	0,022	272	0,022	0,022
800	0,008	0,032	304	0,032	0,032
1000	0,010	0,044	344	0,044	0,044
1200	0,012	0,059	395	0,059	0,059
1400	0,014	0,077	453	0,078	0,078
1600	0,016	0,103	547	0,104	0,104
1800	0,018	0,140	682	0,140	0,141
2000	0,020	0,196	886	0,195	0,197
2200	0,022	0,283	1194	0,280	0,285
2400	0,024	0,427	1689	0,423	0,433
2600	0,026	0,685	2550	0,668	0,689
2800	0,027	1,076	3885	1,060	1,106

Tabela 5: Vertikalni pomiki točke prijemašča sile

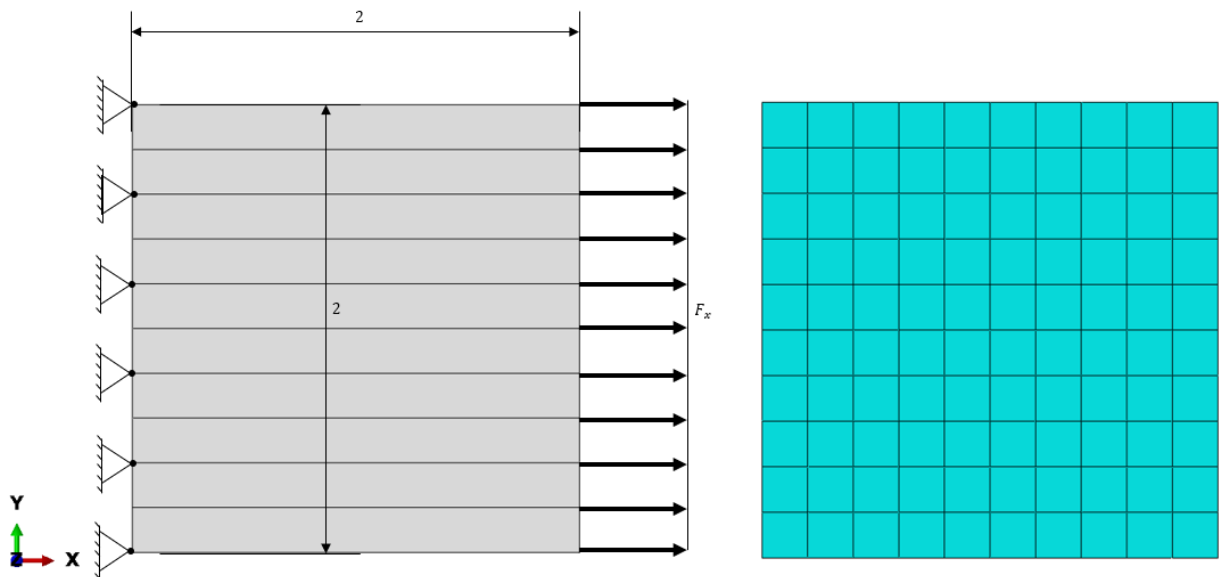


Slika 5.20: Vertikalni pomiki točke prijemašča sile

5.3.3 Natezno obremenjena kocka

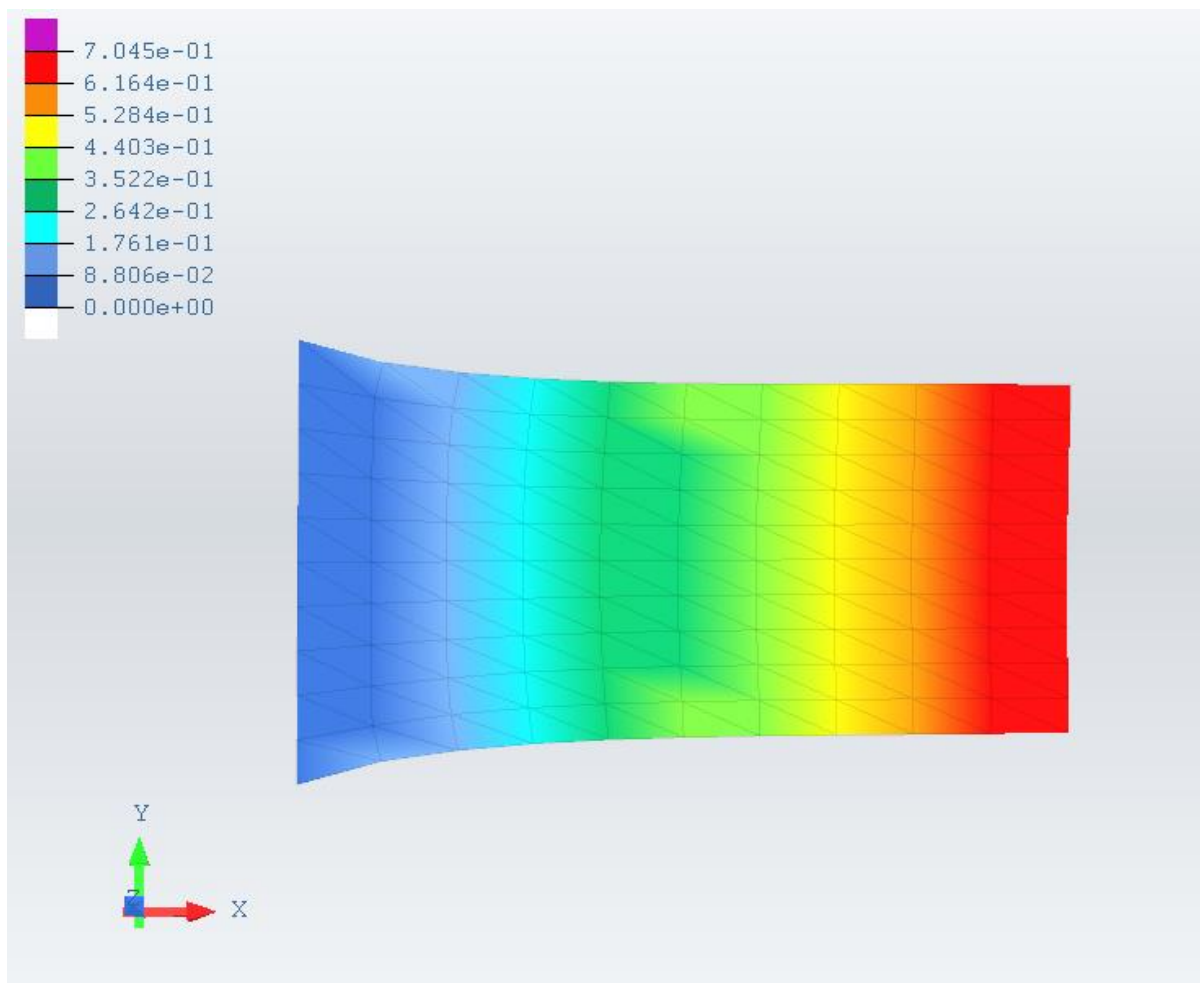
Do sedaj smo se ukvarjali zgolj s primeri, kjer so nastali veliki pomiki, rotacije, a nikoli tudi velike deformacije. Sledi torej obravnava kocke v RNS, ki je natezno obremenjena s porazdeljeno silo F_x . Ker je naš program zmožen le obravnave točkovnih sil, smo silo pretvorili in porazdelili na posamezna vozlišča. Gre za primer, ki je obravnavan v predhodno navedeni literaturi (glej začetek tega poglavja), kot tudi v številnih drugih člankih, kjer obravnavajo problematiko RNS. Prvi je problem predstavil K. J. Bathe za 3D-primer. Kocka je na nasprotni strani obremenitvam togo vpeta po celotni višini (vpete so vse prostostne stopnje). Na naslednji sliki so kotirane dimenzije višine in širine, ki znašajo 2 enoti. Prav tako je treba v programu definirati debelino, ki prav tako znaša 2 enoti. Konstitutivni model je linearno elastičen z modulom elastičnosti $E = 10.000.000$ ter Poissonovim količnikom $\nu =$

0,3. Model je bil narejen s pomočjo programske opreme Abaqus, kjer je potekalo tudi mreženje. Mreža kocke je sestavljena iz 100 enakih štirivozliščnih elementov.



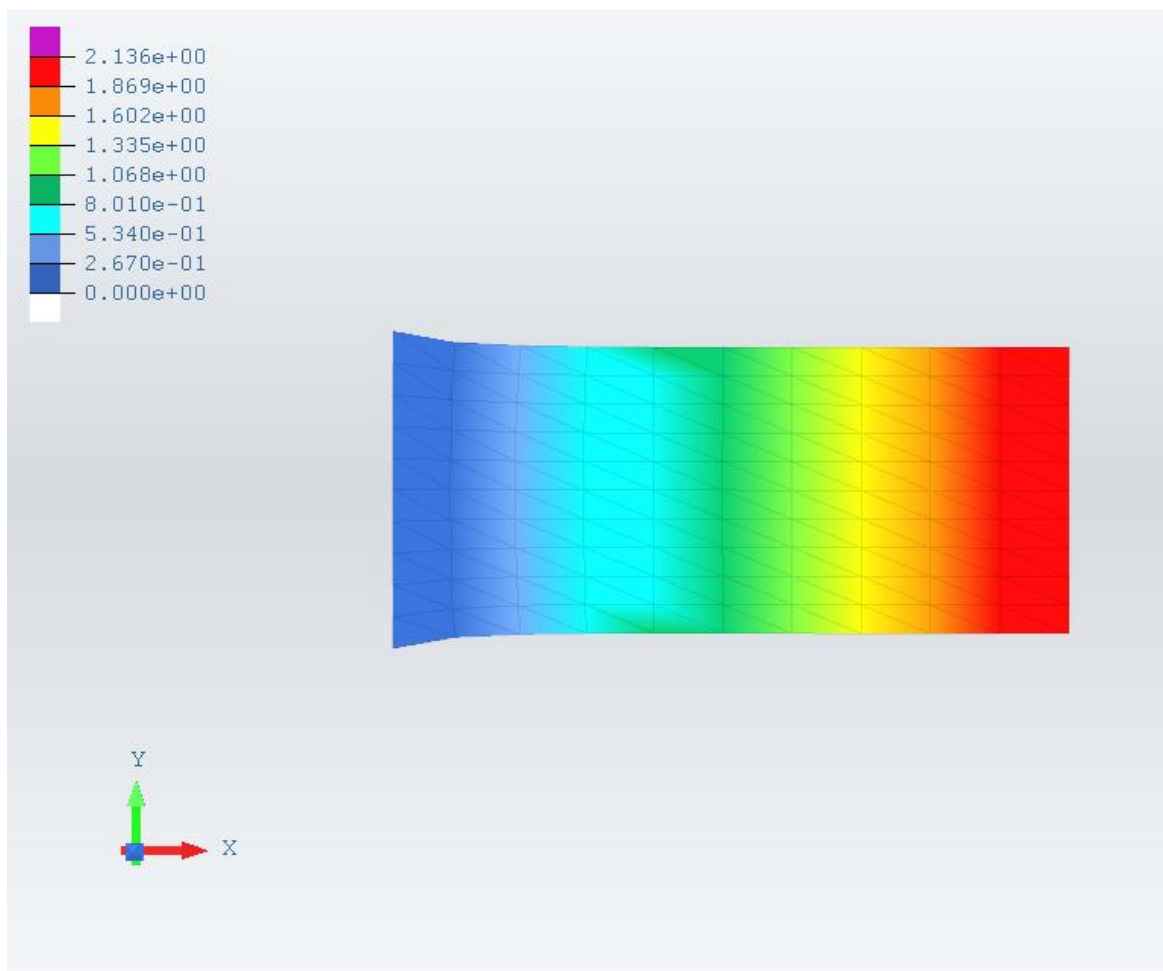
Slika 5.21: Natezno obremenjena kocka

Na prvi sliki lahko predstavimo rezultate linearne analize, kjer opazimo pomik v horizontalni smeri z vrednostjo 0,690 enote. Zdi se, da se nosilec po višini v bližini vpetja ne deformira enakomerno, kar pa se z oddaljevanjem od vpetja popolnoma izravna. Z neupoštevanjem geometrijskih nelinearnosti lahko v primerjavi z nelinearno analizo opazimo 208 % povečanje pomika. Prerez je namreč skozi celotno simulacijo konstanten, kar rezultira v povečani togosti.



Slika 5.22: Pomiki linearne analize kocke za RNS

Pričakovano se z upoštevanjem geometrijskih nelinearnosti pojavi večji pomik, ki pri izračunu z LPO znaša 2,125 enote. Razlaga temelji na posodabljanju geometrije, kjer se z večanjem sile spreminja tudi prerez kocke.



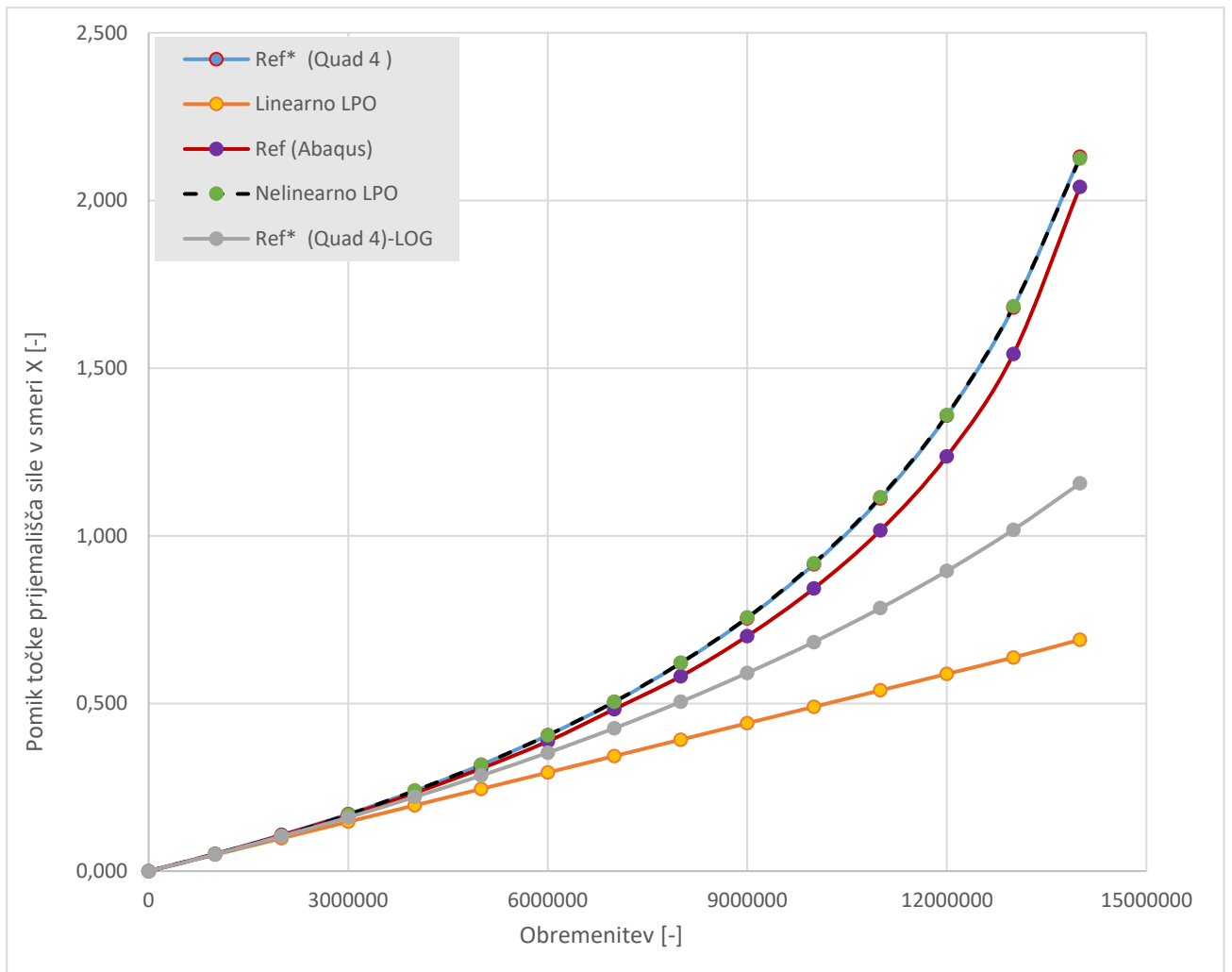
Slika 5.23: Pomiki nelinearne analize kocke za RNS

Kljub temu da smo analizo izvedli in da smo prišli do rezultatov, pa se je treba zavedati, da je za obravnavo velikih deformacij potrebno poznavanje več kriterijev. V tej nalogi smo se ukvarjali s pravilnim kinematičnim opisom, ki ga z izpeljano formulacijo tudi zagotavljamo (zmožni sta ga zagotavljati obe formulaciji TL in UL). Razlike nastanejo med povezavo s konstitutivnim zakonom, ki je prav tako bistvenega pomena za pravilni opis in opis velikih deformacij. Ko imamo opravka z velikimi deformacijami, gre po navadi za nestisljive materiale ali pa za vključeno materialno nelinearnost. Linearno elastičen opis problema predstavlja matrika ${}^t C_{ijkl}$, ki je konstantna ter predstavlja zgolj opis do meje tečenja materialov, kjer pa po navadi ne prihaja do velikih deformacij. Prav tako ne smemo pozabiti, da deformacije merimo s pomočjo Almansijskih deformacij, ki se izkažejo za uporabne in učinkovite zgolj za primer izotropnega materiala, ki ga izpostavimo velikim rotacijam, pomikom, a ne tudi velikim deformacijam. Storjena je bila torej napaka, ki jo lahko opazimo tudi v preglednici z

rezultati ter na naslednjem grafu. Rezultati z Almansijevimi deformacijami se sicer zelo dobro ujemajo z izračunanimi referenčnimi rezultati iz literature. Razlika pa nastane pri primerjavi z Abaqusovimi pomiki, ki se sicer pri maksimalni obremenitvi slučajno precej dobro ujamejo. Če pogledamo trend krivulje, pa vidimo, da bo pri nadaljnjem obremenjevanju prišlo do večjih razlik. Tako Ansysova kot tudi Abaqusova programska oprema namreč uporabljata za izmero deformacij logaritmične deformacije. Prav tako je zanimivo primerjati rezultate z izračunanimi rezultati iz člankov avtorjev [20], [21], kjer so za njihov element uporabljene dejanske logaritmične deformacije pri izračunavanju Cauchyjevih napetosti. Almansijeve deformacije nadomestijo z logaritemskimi, ki temeljijo na levem raztežku, ki smo ga predhodno že predstavili. Prav tako je v tej programski kodi dodana korekcija debeline elementov, kar je v naši simulaciji zanemarjeno. Za popolni opis velikih deformacij se torej zdi, da bi bilo treba dodati še kar nekaj sprememb.

Tabela 6: Horizontalni pomiki sredine prijemališča sile

Obremenitev:	Linearno LPO	Nelinearno LPO	Sprememba [%] LIN-NELIN	Ref* (Abaqus)	Ref* (Quad 4) ALMANSI	Ref* (Quad 4) LOG
0	0,000	0,000	0,00	0,000	0,000	0,000
1000000	0,049	0,051	4,08	0,051	0,051	0,050
2000000	0,098	0,105	7,14	0,107	0,108	0,104
3000000	0,147	0,168	14,29	0,167	0,170	0,160
4000000	0,196	0,240	22,45	0,233	0,239	0,221
5000000	0,245	0,317	29,39	0,306	0,317	0,285
6000000	0,294	0,405	37,76	0,387	0,405	0,353
7000000	0,343	0,505	47,23	0,483	0,504	0,426
8000000	0,392	0,621	58,42	0,581	0,620	0,505
9000000	0,441	0,756	71,43	0,701	0,754	0,591
10000000	0,490	0,918	87,35	0,843	0,915	0,683
11000000	0,539	1,115	106,86	1,016	1,111	0,784
12000000	0,588	1,360	131,29	1,237	1,358	0,895
13000000	0,637	1,684	164,36	1,542	1,681	1,018
14000000	0,690	2,125	207,97	2,040	2,130	1,156



Slika 5.24: Horizontalni pomiki sredine prijemašča sile

6 ZAKLJUČEK

Nalogo lahko zaključimo z mislijo na vse stvari, ki bi jih lahko še dodali. Zdi se namreč, da idej in možnosti nikakor ne zmanjka. To je pokazatelj, kako močan pomen imajo trenutno numerične metode, v kako zelo širokem spektru so uporabne in kako silovito se razvijajo in nadgrajujejo. Omeniti je treba tudi, da je bil končni element generiran povsem ročno, kar je sicer dandanes v primerjavi z avtomatskimi generatorji kode zelo zamudno. Za začetek bi lahko izvedli še analize z različnimi merami deformacij in napetosti. Z njimi bi lahko sistem nadgradili za učinkovito obravnavo velikih deformacij. Izpeljani elementi, s katerimi smo se ukvarjali, so prav tako primerni za analizo RDS-stanja. V program bi torej lahko dodali izbiro RDS. Višali bi lahko interpolacijski red elementa. Nadaljevali bi lahko s 3D-analizo itd. Kljub temu pa lahko povzamemo zastavljene cilje, za katere lahko rečemo, da smo jih v celoti tudi realizirali:

- V prvi fazi je bil uspešno izveden študij literature na področju nelinearnih numeričnih analiz v strukturni mehaniki, ki smo jih predstavili na primerih in teoretično. Predstavili smo tudi metode in formulacije, s katerimi jih lahko obvladujemo.
- Razvit je bil lasten bralnik, s pomočjo katerega smo za vse analize uspešno uvažali podatke iz predprocesorja.
- Uspešno je bila razvita koda za procesiranje in poprocesiranje linearne analize po metodi končnih elementov za probleme ravninskega napetostnega stanja (RNS).
- Uspešno je bila razvita koda za procesiranje in poprocesiranje nelinearne analize po metodi končnih elementov za probleme ravninskega napetostnega stanja (RNS).
- Izvedene so bile številne simulacije za vrsto obremenitvenih primerov, s katerimi smo najprej validirali pravilnost reševanja in primerjali rezultate različnih avtorjev in programskih oprem za primere brez in z upoštevanjem geometrijskih nelinearnosti.
- Na kratko je bilo nakazano in predstavljeno upoštevanje materialne nelinearnosti za primere MNO.

Zaradi lastne programske opreme med izvajanjem analize nekoliko bode v oči počasnost programa ob obravnavi velikih sistemov. Algoritmi reševalnika namreč temeljijo na nekaterih zelo elementarnih rešitvah. Kot v razvoju vsakega programa pa predstavljajo odličen temelj, na katerem je moč graditi naprej.

7 UPORABLJENI VIRI IN LITERATURA

- [1] M. Kegl in B. Harl, *Mehanika III: univerzitetni učbenik*. Maribor: Fakulteta za strojništvo, 2012.
- [2] K.-J. Bathe, *Finite element procedures*. Upper Saddle River [etc.]: Prentice Hall, 1996.
- [3] *The Finite Element Method by G.R. Liu, S. S. Quek - Ebook / Scribd*. Pridobljeno: 13. april 2022. [Na spletu]. Dostopno na: <https://www.scribd.com/book/282573676/The-Finite-Element-Method-A-Practical-Course>
- [4] „Abaqus 6.14 Documentation“. <http://wufengyun.com:888/> (pridobljeno 27. marca 2022).
- [5] American Bureau of Shipping „Guidance Notes on Nonlinear Finite Element Analysis of Marine and Offshore Structures“, ABS Group, 2021.
- [6] Y. Tadano in H. Noguchi, „Geometrically Nonlinear Analyses Using 2Dimensional Generalized Finite Element Enriched by the Quadratic Deformation Mode“, *Journal of Computational Science and Technology*, str. 11–24, jan. 2010.
- [7] B. Klaus-Jürgen, „Nonlinear Analysis Finite Element Procedures for Solids and Structures MIT OpenCourseWare“. <https://ocw.mit.edu/resources/res-2-002-finite-element-procedures-for-solids-and-structures-spring-2010/nonlinear/> (pridobljeno 27. marca 2022).
- [8] „Understanding Nonlinear Analysis“. Pridobljeno: 5. julij 2022. [Na spletu]. Dostopno na: https://www.solidworks.com/sw/docs/NonlinearAnalysis_WP_ENG.pdf
- [9] N. Madutujuh, „Introduction to Material Nonlinearity“, let. 2021, avg. 2021
- [10] J. N. Reddy, *An introduction to the finite element method*, 3rd ed. New York [etc.]: McGraw-Hill, 2006.
- [11] A. Hojjat, *Solution Techniques for Linear and Nonlinear Dynamics of Structures Modeled by Finite Elements*, let. 1976. Stanford University, 1976.
- [12] „Polar Decomposition“. <https://www.continuummechanics.org/polardecomposition.html> (pridobljeno 10. julija 2022).
- [13] J.-Y. Kim, D.-B. Kim, H.-J. Cho, S.-B. Kwon, in Y.-D. Kwon, „A Critical Heat Generation for Safe Nuclear Fuels after a LOCA“, *Science and Technology of Nuclear Installations*, let. 2014, str. 1–8, maj 2014.
- [14] M. Hriberšek in J. Ravnik, *Numerično modeliranje in računalniške simulacije. Del 1, Gradniki numeričnega računanja*. Maribor: Fakulteta za strojništvo, 2012.

- [15] N. Gubelj, *Mehanika loma: univerzitetni učbenik*. Fakulteta za strojništvo, 2009.
- [16] Z. Ren, M. Ulbin, in M. Vesenjak, „Inženirske računalniške simulacije v konstrukterstvu : univerzitetni učbenik“, Univerza v Mariboru, Fakulteta za strojništvo, book, 2018. Pridobljeno: 19. aprila 2022. [Na spletu]. Dostopno na: <https://dk.um.si/IzpisGradiva.php?id=71069>
- [17] N.-H. Kim, *Introduction to Nonlinear Finite Element Analysis*. New York, NY: Springer US, 2015.
- [18] „FEM for Two-Dimensional Solids (Finite Element Method) Part 1“. <http://what-when-how.com/the-finite-element-method/fem-for-two-dimensional-solids-finite-element-method-part-1/> (pridobljeno 27. marca 2022).
- [19] N. Kim, *Introduction to Nonlinear Finite Element Analysis*. New York [etc.]: Springer, 2015.
- [20] Y. Yang, X. Tang, H. Zheng, in Q. Liu, „Four-Node Quadrilateral Element with Continuous Nodal Stress for Geometrical Nonlinear Analysis“, *International Journal of Computational Methods*, let. 15, str. 1850005, apr. 2017, doi: 10.1142/S0219876218500056.
- [21] S. Rajendran, „A partition of unity-based 'FE-meshfree' QUAD4 element for geometric non-linear analysis“, *International Journal for Numerical Methods in Engineering*, jan. 2009, Pridobljeno: 29. aprila 2022. [Na spletu]. Dostopno na: https://www.academia.edu/4005196/A_partition_of_unity_based_%C3%A2_FE%C3%A2_Meshfree%C3%A2_QUAD4_element_for_geometric_non_linear_analysis