RJEŠAVANJE PROBLEMA POPREČNO OPTEREĆENE GREDE PRIMJENOM METODE KONAČNIH ELEMENATA

Atelj, Jurica

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split / Sveučilište u Splitu**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:228:526377

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-25**

Repository / Repozitorij:

Repository of University Department of Professional Studies

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar
DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU

SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Konstrukcijsko strojarstvo

Jurica Atelj

Z A V R Š N I R A D

RJEŠAVANJE PROBLEMA POPREČNO OPTEREĆENE GREDE PRIMJENOM METODE KONAČNIH ELEMENATA

Split, listopad 2020.

SVEUČILIŠTE U SPLITU
SVEUČILIŠNI ODJEL ZA STRUČNE STUDIJE

Preddiplomski stručni studij Konstrukcijsko strojarstvo

Predmet: Izbor materijala

Z A V R Š N I R A D

Kandidat: Jurica Atelj

Naslov rada: Rješavanje problema poprečno opterećene grede primjenom metode konačnih elemenata

Mentor: Igor Gabrić

Split, listopad 2020.

REPUBLIKA HRVATSKA
SVEUČILIŠTE U SPLITU
Sveučilišni odjel za stručne studije

Studij: Konstrukcijsko strojarstvo
Predmet: Izbor materijala
Nastavnik: Igor Gabrić, v. pred.

ZADATAK

Kandidat: Jurica Atelj

Zadatak: Rješavanje problema poprečno opterećene grede primjenom metode konačnih elemenata

U okviru završnog rada potrebno je:

- Uvodno se osvrnuti na metodu konačnih elemenata
- Prikazati teorijski opis rješavanja problema ravninskih greda primjenom metode konačnih elemenata
- Uvodno se osvrnuti o programiranju i programskom jeziku Python
- Izraditi dijagram toka, rastavljen na logičke cjeline, za rješenje općeg slučaja poprečno opterećene grede kao podloge za programiranje
- Izraditi program primjenom programskog jezika Python za rješavanje općeg problema poprečno opterećene grede
- Analitički riješiti dva primjera poprečno opterećene grede primjenom metode konačnih elemenata
- Primjenom izrađenog programa, a uz pomoć računala, riješiti dva primjera, prethodno riješena analitički
- Usporediti i tablično prikazati rezultate za oba primjera dobivene analitičkom metodom i primjenom izrađenog programa
- Iznijeti zaključke
- Navesti literaturu i izvore podataka.

Sažetak

Ovaj rad obuhvaća rješavanje i razumijevanje ravninskog problema poprečno opterećene grede „Metodom konačnih elemenata“. Srž rada je izrada programskog koda u programskom jeziku „Python“ koji omogućava određivanje pomaka čvorova i reakcija u osloncima poprečno opterećene grede. Prije izrade programa izrađeni su dijagrami toka koji su omogućili lakše i jednostavnije snalaženje pri izradi programa. U svrhu testiranja programskog koda riješena su dva primjera problema grede na analitički način, te primjenom programskog koda. Konačno je izvršena usporedba rezultata dobivenih analitički i primjenom računala.

Ključne riječi: Metoda konačnih elemenata, Python, dijagrami toka

Summary (Solving the problem of transversely loaded beams using finite element method)

This paper includes solving and understanding the plane problem of transversely loaded beams "Finite element method". The core of the work is the program code in the programming language "Python" which will enable the determination of node displacements and reactions in the supports of the transversely loaded beam. Before creating the program, diagrams were made that make it easier and simpler to navigate when running the program. To test the program code and compare the results, two examples of beam problems were solved in an analytical way, and by applying the program code.

Key words: Finite element method, Python, flow chart

Sadržaj

Sažetak iii

Summary (Solving the problem of transversely loaded beams using finite element methods) iii

1. Uvod 1

1.1. Metoda konačnih elementa 1

1.2. Metoda konačnih elementa s primjenom u nauci o čvrstoći 1

2. Teorijski opis rješavanja problema grede primjenom MKE 3

2.1. Prosta greda 4

2.2. Kontinuirano opterećena greda 8

3. Dijagrami toka 10

3.1. Dijagrami ulaznih podataka 11

3.2. Dijagrami obrade podataka 14

3.3. Dijagrami izlaznih podataka 22

4. Izrada programskog koda 24

5. Testiranje programa i rješavanje analitičkog načina rješavanja grede MKE 37

5.1. PRIMJER 1 37

5.2. PRIMJER 2 46

5.3. Rješenje primjera 1 primjenom Python programa 57

5.4 Rješenje primjera 2 primjenom Python programa 60

6. Usporedba analitičkog računa i programskog računa 65

6.1. Usporedba prvog primjera 65

6.2. Usporedba drugog primjera 67

7. Zaključak 69

Literatura 71

Popis slika

Slika 1.1 – Primjer MKE u analizi mehaničkog naprezanja programom „AutoDesk inventor“ [1] 2

Slika 2.1 - Element grede opterećen na savijanje [2] 3

Slika 2.2 – Primjer jednostavno opterećene grede [3] 4

Slika 2.3 – Način slaganja globalnih matrica [4] 7

Slika 2.4 – Kontinuirano opterećena greda [3] 8

Slika 2.5 – Primjeri smičnih sila i momenata savijanja za određeni tip grede [5] 9

Slika 3.1 – Primarni blokovi dijagrama toka 10

Slika 3.2 – Primarni dijagram unosa podataka 11

Slika 3.3 – Dijagram toka unosa podataka o čvorovima 12

Slika 3.4 – Dijagram toka unosa podataka o elementu 12

Slika 3.5 – Dijagram obrade podataka 14

Slika 3.6 – Dijagram toka „Matrice elementa“

Slika 3.7 – Dijagram toka „Matrica veza“ 15

Slika 3.8 – Definiranje nul matrica 16

Slika 3.9 – Stvaranje lokalnih matrica krutosti 17

Slika 3.10 – Stvaranje globalnih matrica krutosti 18

Slika 3.11 – Prikaz pomaka na elementu 18

Slika 3.12 – Dijagram generiranje reducirane matrice krutosti 19

Slika 3.13 – Dijagram generiranja reducirane matrice sila 19

Slika 3.14 – Dijagram izračunavanja pomaka 20

Slika 3.15 – Dijagram izračunavanja reakcija u osloncima 21

Slika 3.16 – Dijagram izlaznih podataka 22

Slika 3.17 – Dijagram toka izlaznih podataka pomaka 23

Slika 3.17 – Dijagram toka izlaznih podataka reakcija u osloncima 23

Slika 4.1 – Početak programa i odabir o unosu novih ili korištenju unaprijed određenih ulaznih podataka 24

Slika 4.2 – Unos ulaznih podataka 25

Slika 4.3 – Unos kontinuiranih opterećenja i određivanje ekvivalentnog opterećenja po elementima 26

Slika 4.4 – Unos ulaznih podataka o čvorovima 27

Slika 4.5 – Unos rubnih uvjeta 27

Slika 4.6 – Stvaranje matrice sila 28

Slika 4.7 - Veza između elemenata 29

Slika 4.8 – Generiranje matrice veza 29

Slika 4.9 – Formiranje nul matrica 30

Slika 4.10 – Generiranje lokalne matrice krutosti 31

Slika 4.11 – Zapis lokalnih matrica krutosti u formatu globalne matrice krutosti 32

Slika 4.12 – Formiranje globalne matrice krutosti 32

Slika 4.13 – Reducirana matrica krutosti 33

Slika 4.14 – Formiranje reducirane matrice sila 33

Slika 4.15 – Izračunavanje rubnih pomaka 34

Slika 4.16 – Izračunavanje reakcija u osloncima 34

Slika 4.16 – Formiranje ispisa izračunatih pomaka 34

Slika 4.17 – Ispis izračunatih pomaka 35

Slika 4.18 – Kod za dobivanje liste „pomaci_svi“ 35

Slika 4.19 – Kod za ispisivanje liste „pomaci_svi“ 35

Slika 4.20 – Konačni rezultati svih pomaka 36

Slika 4.21 – Kod za ispis rezultata reakcija u osloncima Slika 4.22 – Ispis rezultata reakcija u osloncima 36

Slika 5.1 – Primjer br. 1 [6] 37

Slika 5.2 – Greda s označenim elementima, čvorovima i pomacima 38

Slika 5.3 – Primjer 1, element 1 40

Slika 5.4 – Primjer 1, element 2 41

Slika 5.5 – Primjer 1, element 3 41

Slika 5.6 – Primjer br. 2 [6] 46

Slika 5.7 – Greda s označenim elementima, čvorovima i pomacima 47

Slika 5.8 – Primjer 2, element 1 50

Slika 5.9 – Primjer 2, element 2 50

Slika 5.10 – Primjer 2, element 3 51

Slika 5.11 – Primjer 2, element 4 52

Slika 5.12 – Ulazni podatci za element (primjer 1) 57

Slika 5.13 – Unos kontinuiranog opterećenja (primjer 1) 57

Slika 5.14 – Unos vrijednosti sila i momenata u čvorovima (primjer 1) 58

Slika 5.15 – Unos stanja pomaka u čvorovima (primjer 1) 58

Slika 5.16 – Međurezultati obrade podataka (primjer 1) 59

Slika 5.17 – Rezultati dobiveni uporabom programa (primjer 1) 59

Slika 5.18 – Ulazni podatci o elementu (primjer 2) 60

Slika 5.19 – Unos kontinuiranog opterećenja (primjer 2) 61

Slika 5.20 – Unos vrijednosti sila i momenata u čvorovima (primjer 2) 61

Slika 5.21 – Unos stanja pomaka u čvorovima (primjer 2) 62

Slika 5.22 – Međurezultati obrade podataka – globalna matrica krutosti (primjer 2) 62

Slika 5.23 – Međurezultati obrade podataka – reducirane matrice (primjer 2) 63

Slika 5.24 – Rezultati programa (primjer 2) 63

Popis tablica

Tablica 6.1 – Usporedba rezultata pomaka primjera 1……...…65

Tablica 6.2 – Usporedba reakcija u osloncima primjera 1……66

Tablica 6.3 – Usporedba rezultata pomaka primjera 2……..67

Tablica 6.4 – Usporedba rezultata reakcija u osloncima primjera 2……..67

1. Uvod

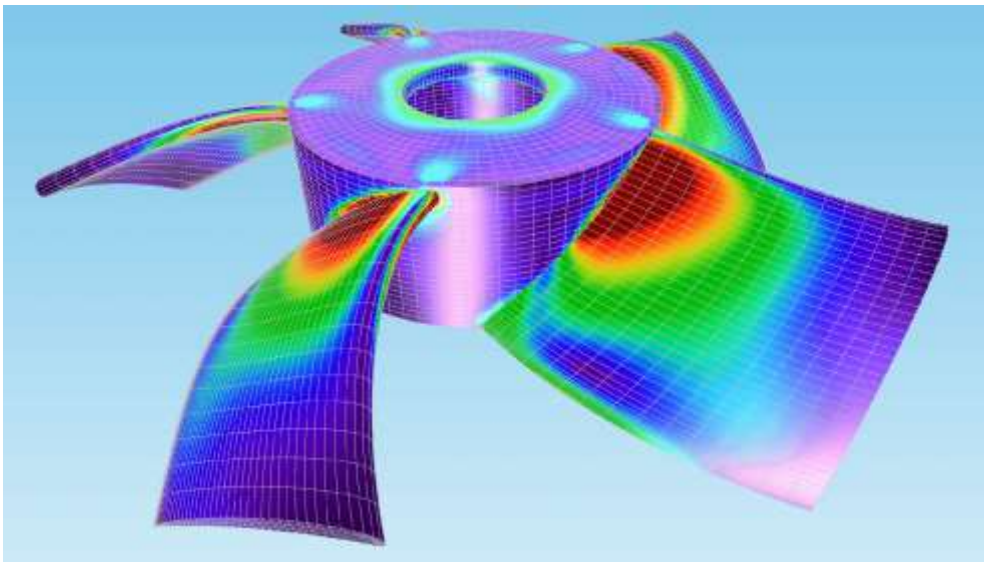
1.1. Metoda konačnih elementa

Metoda konačnih elemenata je jedna od najrasprostranjeniji metoda koja se primjenjuje u strojarskoj praksi. Postupak metode je zamišljen tako da jednu cjelinu dijeli na manji broj dijelova. Ti dijelovi se zovu elementi. Elementi su međusobno povezani u točkama na početku i kraju elementa koje se nazivaju čvorovi. Pronalaženje rješenja se svodi na određivanje integracijskih konstanti s obzirom na osnovne relacije opisane rubnim uvjetima (pomacima u čvorovima). Metodom konačnih elemenata radi se prijelaz iz parcijalne diferencijalne jednadžbe u sustav linearnih običnih jednadžbi, koji se može zapisati u matričnoj formulaciji. Nepoznanice problema su pomaci čvorova, a pomake unutar elementa se određuje interpolacijom. Iz određenih pomaka određuju se ostale veličine (reakcije u osloncima, deformacije, naprezanja..). Kako bi se proračun omogućio, predmet računanja (elementi) se stavljaju u koordinatni sustav. Ovisno o načinu gledanja problema elementi mogu biti jednodimenzionalni (1D), dvodimenzionalni (2D) i trodimenzionalni (3D). Za razliku od prve dvije dimenzije 3D koristi posebni oblik takozvanu mrežu za elemente. Mreža predstavlja 3 ili više rubnih točka međusobno povezane kako bi se opisao konačni element.

U ovom radu koristi se jednodimenzionalni gredni konačni element za rješavanje ravninskih problema poprečno opterećene grede.

1.2. Metoda konačnih elementa s primjenom u nauci o čvrstoći

MKE ima jaku vezu s naukom o čvrstoći. Pošto se nauka o čvrstoći bavi čvrstoćom, krutosti i stabilnosti materijala (strojeva, konstrukcija) primjenom MKE omogućuje se rješavanje kompleksnih problema koje nije moguće riješiti bez primjene računala. Ovisno o gledanom predmetu MKE će omogućiti rastavljanje nekog predmeta na manje dijelove (elemente), te prikazati kakav utjecaj imaju aktivne sile na gledani objekt. Primjenom metode u praksi omogućuje se rješavanje i problema koji uključuju plastične deformacije, vibracije, stabilnost, toplinska naprezanja, te kombinaciju navedenih problema. Na taj način je moguće predvidjeti problem u budućim konstrukcijama prije same izrade i poduzeti mjere da se ti problemi riješe. Danas je metoda toliko rasprostranjena da se koristi u izradi konstrukcija, zrakoplovstvu, testiranju vozila na sudar...



Slika 1.1 – Primjer MKE u analizi mehaničkog naprezanja programom „AutoDesk inventor“ [1]

Na slici 1.1 se vidi nacrtani propeler u AutoDesk Inventor-u s prikazanim mjestima visokog naprezanja. Za izradu analize mehaničkog naprezanja korištena je MKE te se jasno vidi mreža konačnih elemenata. Elementi koji su crveni prikazuju kritična područja pod opterećenjem. Ovakav proračun osjetno olakšava posao u sprječavanju nastajanja mjesta koncentracija naprezanja, te upućuje na lokacije koje je potrebno ojačati.

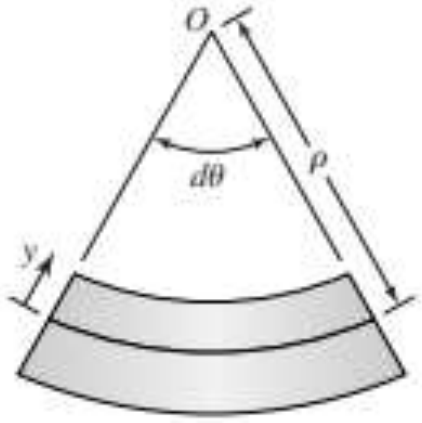
2. Teorijski opis rješavanja problema grede primjenom MKE

U ovom poglavlju je prikazan opći postupak rješavanja ravninskog problema poprečno opterećene grede metodom konačnih elemenata. U konkretnom slučaju riješen je problem jednodimenzionalne grede s neograničenim brojem elemenata. Gledajući globalni koordinatni sustav greda se proteže duž x-osi bez dodatnih elemenata u smjeru y-osi. Krajnji cilj cijelog proračuna je određivanje ukupnih pomaka u čvorovima grede, te reakcije u osloncima.

Bitne pretpostavke:

- Greda je opterećena samo po y-osi (aksijalna sila isključena)
- Materijal grede se promatra kao linearno elastičan, homogen i izotropan (slika 2.1)

Matrica krutosti je preuzeta iz literature [2] već riješena te je uzeta gotova u proračun



Slika 2.1 - Element grede opterećen na savijanje [2]

Svaki konačni element ima dvije rubne točke gdje u svakoj rubnoj točki imaju dva stupnja slobode gibanja tj. dva rubna uvjeta. U slučaju grede to su progib (v) i zakret (θ).

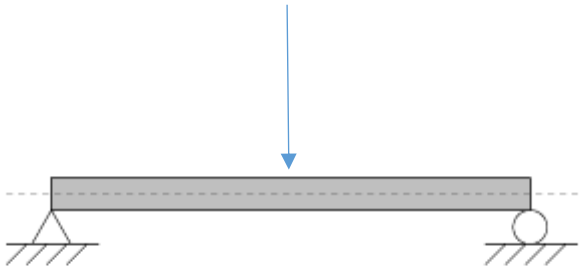
Proračun se izvodi prema slijedećem redoslijedu:

- Definiranje broja elementa i rubnih točki (ulazni podatci)
- Definiranje dužina elemenata Young-ovog modula elastičnosti i momenta inercije (ulazni podatci)
- Definiranje sila i momenata u rubnim točkama (ulazni podatci)
- Definiranje kontinuiranog opterećenja na elementu (ulazni podatci)
- Definiranja rubnih uvjeta u čvorovima (oslonci, uklještenja, zglobovi)
- Određivanje lokalnih i globalne matrice krutosti
- Određivanje matrice pomaka i matrice sila
- Određivanje traženih progiba i reakcija u osloncima

Rad obuhvaća rješavanje problema grede na tri načina. Prvi je problem grede s koncentriranim opterećenjem i momentom, drugi je problem grede s kontinuiranim opterećenjem, te na koncu problem s kombiniranim opterećenjem. Postupak se razlikuje prilikom kreiranja matrice sila gdje se, u problemu s kontinuiranim opterećenjem, pojavljuju dodatni članovi potrebni za rješavanje problema.

Problem grede opterećene i aksijalnom silom nije predmet ovog završnog rada!

2.1. Prosta greda



Slika 2.2 – Primjer jednostavno opterećene grede [3]

Unosom ulaznih vrijednosti za svaki element (duljina elementa, Young-ov modul elastičnosti, aksijalni moment inercije) i računanjem vrijednosti $\frac{E \cdot I}{L^3}$ za svaki element dolazi se do određivanja lokalnih matrica krutosti. Lokalna matrica krutosti za element je preuzeta iz literature te se određuje za svaki element zasebno. Globalna matrica krutosti predstavlja kombinaciju riješenih parcijalnih jednadžbi za svaki element zasebno.

Postupak formiranje lokalnih matrica krutosti elemenata je uvijek isti. Razlika jedino može biti u dužini elementa, aksijalnom momentu inercije i Young - ovom modulu elastičnosti elementa.

Izgled lokalne matrice krutosti za element [2] je

$$[k_e] = \frac{E \cdot I}{L^3} \begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} \tag{2.1}$$

$$\begin{bmatrix} k_{11} & k_{12} & k_{13} & k_{14} \\ k_{21} & k_{22} & k_{23} & k_{24} \\ k_{31} & k_{32} & k_{33} & k_{34} \\ k_{41} & k_{42} & k_{43} & k_{44} \end{bmatrix} = \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{2.2}$$

Prikaz aktivnih sila koje djeluje na gredu su zapisane u obliku matrice sile. Bitno kod kreiranja matrice sila je da se na mjestu djelovanja sile postavlja čvor. Sile djeluje samo u krajnjim točkama elementa! O broju čvorova ovisi i konačni oblik matrice sila. Svaki čvor može imati jednu vertikalno orijentiranu silu i jedan zakretni moment.

Izgled matrice (4x1) sila za svaki konačni element

$$\{F\} = \begin{Bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \end{Bmatrix} \tag{2.3}$$

Kako bi se dobili traženi rezultati treba definirati matricu pomaka elemenata. Pomaci u problemu grede predstavljaju progib (v) i rotaciju (θ) u rubnim točkama elemenata. . Ako u čvoru nama pomaka, pomaku se dodjeljuje vrijednost „0“, a taj redak matrice se zanemaruje kod određivanja reduciranih matrica s ciljem određivanja pomaka (progiba). Pomak će biti „0“ ukoliko se u konkretnom čvoru nalazi uklještenje ili oslonac.

- Konzolno uklještenje – onemogućeni progib i zakretanje
- Oslonac na rubnoj točki – ne postoji progib, a zakretanje je slobodno
- Slobodna rubna točka – slobodni i progib i zakretanje

Izgled matrice pomaka za jedan element gdje „u“ predstavlja progib u rubnoj točki elementa, a „ϑ" kut zakreta u rubnoj točki. Obzirom da element ima dvije krajnje (rubne) točke matrica pomaka elementa je:

$$\{u\} = \begin{Bmatrix} u_1 \\ \vartheta_1 \\ u_2 \\ \vartheta_2 \end{Bmatrix} \tag{2.4}$$

Osnovna matrična jednadžba sustava je:

$$\{F\} = [K] \cdot \{u\} \tag{2.5}$$

Na formuli 2.5 se zasniva rješavanje problema grede primjenom MKE gdje se određivanjem inverzne matrice krutosti matričnom algebrom dobiva

$$[K] \cdot \{u\} = \{F\} / \cdot [K]^{-1} \tag{2.6}$$

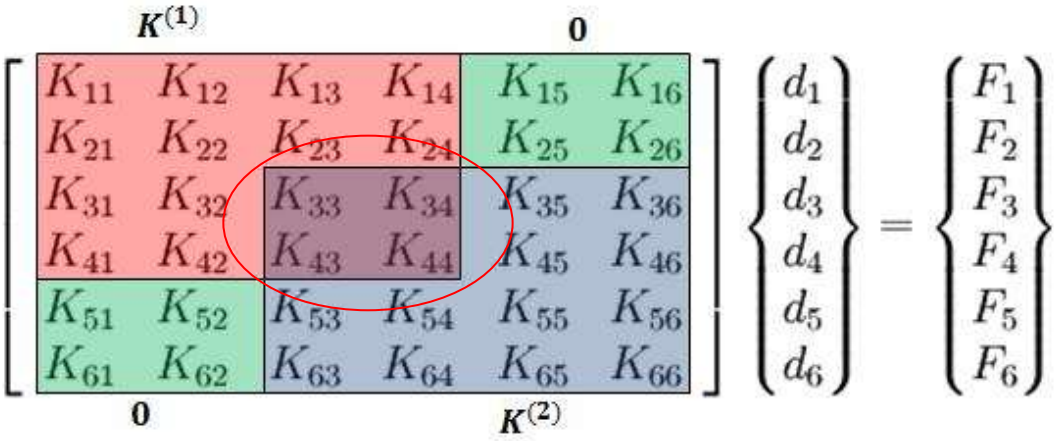
$$[K]^{-1} \cdot [K] \cdot \{u\} = [K]^{-1} \cdot \{F\} \tag{2.7}$$

$$[1] \cdot \{u\} = [K]^{-1} \cdot \{F\} \tag{2.8}$$

$$\{u\} = [K]^{-1} \cdot \{F\} \tag{2.9}$$

$$\frac{E \cdot I}{L^3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \begin{Bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \end{Bmatrix} = \begin{Bmatrix} u_1 \\ \vartheta_1 \\ u_2 \\ \vartheta_2 \end{Bmatrix} \tag{2.10}$$

U izrazu 2.6 prikazane su matrice krutosti, sila i pomaka za jedan element. U praktičnoj ili znanstvenoj primjeni uvijek je zastupljen veći broj elemenata. U slučajnu većeg broja elemenata lokalna matrica krutosti neće biti dovoljna već se kreira globalna matrica krutosti. Ona opisuje međusobnu vezu između svih elemenata grede.



Slika 2.3 – Način slaganja globalnih matrica [4]

Na slici 2.3 crvenom elipsom je istaknuto mjesto međusobne povezanosti dva elementa. Na tom mjestu dolazi do zbrajanja vrijednosti dvije lokalne matrice krutosti. Neovisno o broju elemenata u problemu grede uvijek će postojati zajednički čvorovi susjednih elemenata. Dakle, pomaci susjednih elemenata u zajedničkom čvoru su jednaki. Treba istaknuti da zglobne veze (Gerberovi nosači) nisu predmet ovog završnog rada.

Rješavanjem jednadžbe 2.10 prelazi se u linearni sustav jednadžbe s više nepoznanica iz kojih se računaju pomaci i nagibi u čvorovima.

$$\begin{aligned} k_{11} \cdot F_1 + k_{12} \cdot M_1 + k_{13} \cdot F_2 + k_{14} \cdot M_2 &= u_1 \\ k_{21} \cdot F_1 + k_{22} \cdot M_1 + k_{23} \cdot F_2 + k_{24} \cdot M_2 &= \vartheta_1 \\ k_{31} \cdot F_1 + k_{32} \cdot M_1 + k_{33} \cdot F_2 + k_{34} \cdot M_2 &= u_2 \\ k_{41} \cdot F_1 + k_{42} \cdot M_1 + k_{43} \cdot F_2 + k_{44} \cdot M_2 &= \vartheta_2 \end{aligned} \tag{2.11}$$

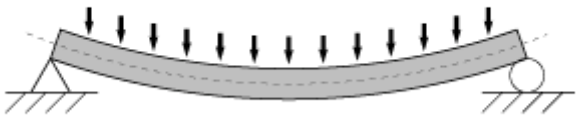
Koristeći Gauss-ovu metodu eliminacije ili sličan način rješavanja problema dobivaju se završni rezultati pomaka čvorova.

Zadnji korak je određivanje reakcija u osloncima. Formula na temelju koje se dobivaju reakcije u osloncima je:

$$\{R\} = [K]\, x\, \{u\} - \{F\} \tag{2.12}$$

Dobivena matrica reakcija $\{R\}$ prikazuje reakcije momenata i sila za sve čvorove. Ukoliko je u nekom čvoru omogućen pomak ili zakret, onda odgovarajuća vrijednost reakcije sile ili momenta u matrici $\{R\}$ iznosi „0“.

2.2. Kontinuirano opterećena greda



Slika 2.4 – Kontinuirano opterećena greda [3]

Metodologija rješavanja kontinuirano opterećene grede se bitno ne razlikuje od problema grede s jednostavnim opterećenjem. Razlika je jer kontinuirano opterećenje ne djeluje u čvorovima nego duž cijelog elementa. U tom slučaju dolazi se na drugačiji način određuju matrice sila. Princip se svodi na postupak zamjene kontinuiranog opterećenja, duž elementa, odgovarajućim ekvivalentnim silama i momentima u rubnim točkama istog elementa.

Za svaku rubnu točku računa se ekvivalentna sila smicanja, te ekvivalentni moment savijanja.

$$F = \frac{q \cdot L}{2} \tag{2.13}$$

$$M = \frac{q \cdot L^2}{12} \tag{2.14}$$

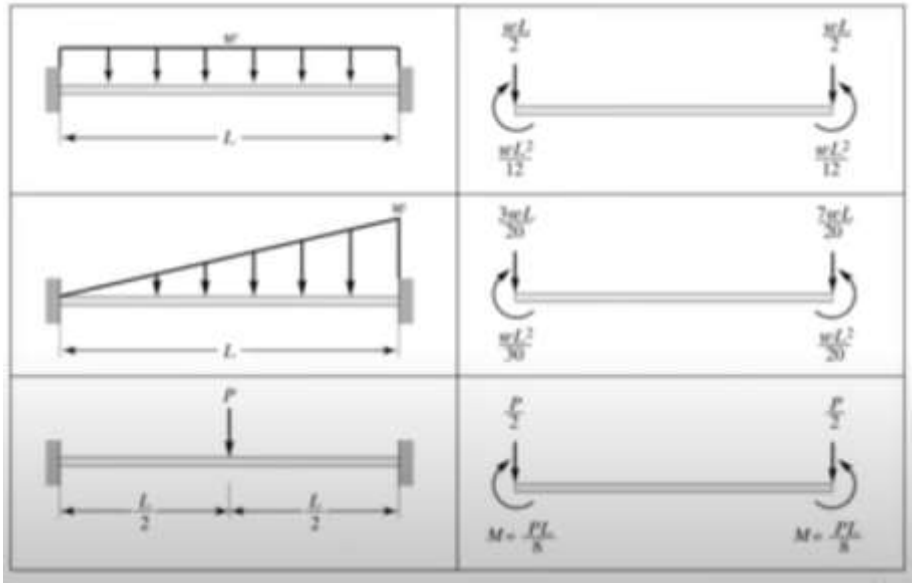
Matrica sila elementa uslijed kontinuiranog opterećenja je:

$$\begin{Bmatrix} F_{1Q} \\ M_{1Q} \\ F_{2Q} \\ M_{2Q} \end{Bmatrix} = \begin{Bmatrix} \frac{q \cdot L}{2} \\ \frac{q \cdot L^2}{12} \\ \frac{q \cdot L}{2} \\ \frac{-q \cdot L^2}{12} \end{Bmatrix} \tag{2.15}$$

Dok je ukupna matrica sila elementa:

$$\{F\} = \begin{Bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \end{Bmatrix} + \begin{Bmatrix} F_{1Q} \\ M_{1Q} \\ F_{2Q} \\ M_{2Q} \end{Bmatrix} \tag{2.16}$$

Konačna matrica sila se dobije zbrajanjem ekvivalentnih sila i momenata uslijed kontinuiranih opterećenja s aktivnim silama i momentima.



Slika 2.5 – Primjeri smičnih sila i momenata savijanja za određeni tip grede [5]

Ovisno o načinu kontinuiranog opterećenja na gredi ekvivalentna sila smicanja, te ekvivalentni moment savijanja će biti različiti.

3. Dijagrami toka

Dijagrami toka daju pojednostavljeni prikaz rješavanja problema koji olakšava izradu programa. Prije pisanja programskog koda izrađuju se dijagrami toka u obliku skupine blokova međusobno povezanih strelicama radi lakšeg praćenja toka od početka problema prema kraju. Dijagrami toka moraju pokazivati jasan prikaz stvaranja programskoga koda od početka do kraja kako bi se analitički postupak implementirao u programski kod.

Radi lakšeg snalaženja za ovaj zadatak dijagrami toka su rastavljeni na logične cjeline jer bi sveukupan dijagram bio nepregledan za praćenje.

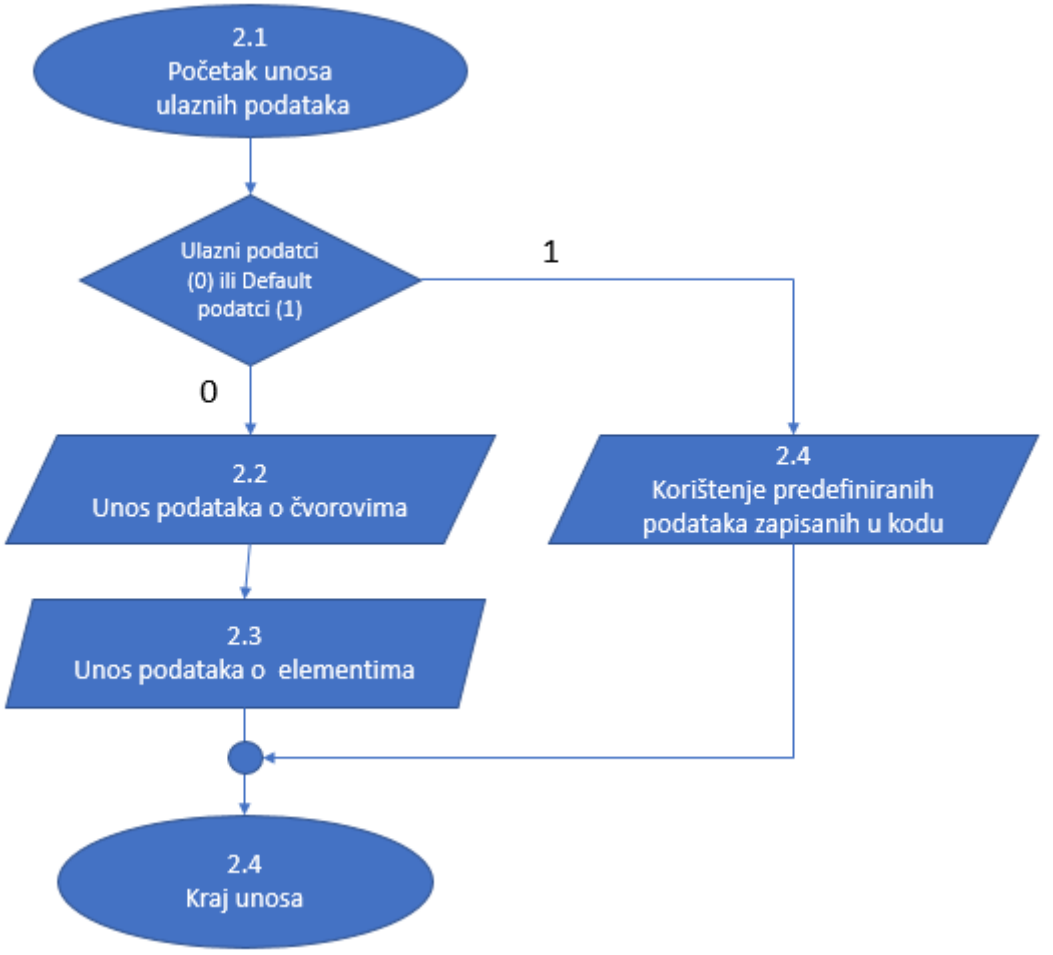
Generalni dijagram toka na kojem se zasnivaju dijagrami u nastavku prikazan je na slici 3.1

Slika 3.1 – Primarni blokovi dijagrama toka

U bloku 2 (Ulazni podaci) biti će opisani svi podatci koji su potrebni za rješavanje konkretnog problema grede.

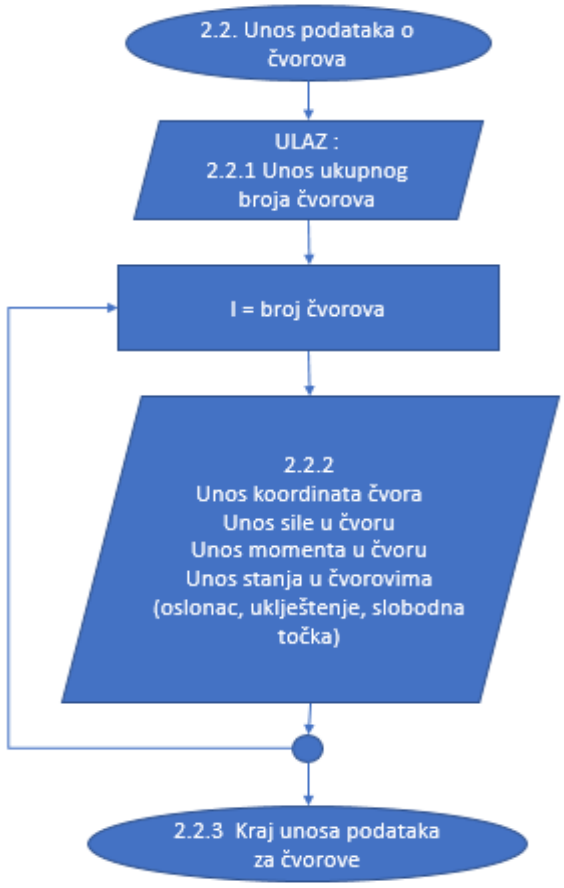
U bloku 3 biti će opisan proces rješavanja problema primjenom MKE, dok će u bloku 4 biti pokazani podaci za ispis na zaslonu računala.

3.1. Dijagrami ulaznih podataka

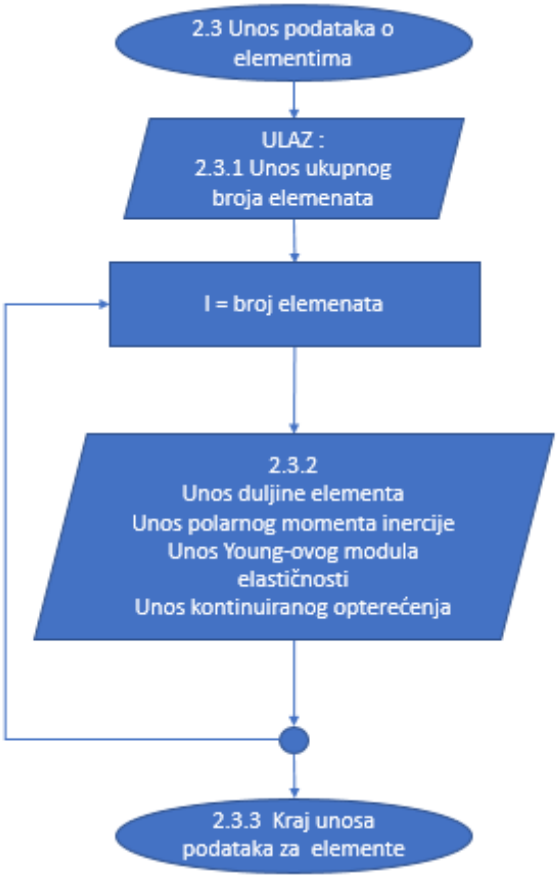


Slika 3.2 – Primarni dijagram unosa podataka

Slika 3.2 prikazuje primarni dijagram unosa podataka. Od njega kreće grananje na manje logičke cjeline. Slika 3.2 započinje odlukom dali se želi koristiti unos podataka ili *„default“* podatci. Unosom broja „0“ otvorit će se mogućnost unosa podataka od strane korisnika, a unosom broja „1“ program koristi podatke definirane u kodu i vrši proračun do kraja i ispis rezultata na zaslon. Nadalje ako se odluči na unos novih podataka unose se podaci o čvorovima i elementima „Unos podataka o elementima“ i „Unos podataka o čvorovima“.



Slika 3.3 – Dijagram toka unosa podataka o čvorovima



Slika 3.4 – Dijagram toka unosa o elementu

Na slikama 3.3 i 3.4 su prikazani dijagrami za unos podataka o čvorovima i elementima. Svaki element i čvor imaju svoju određenu funkciju u MKE te su ukratko opisane u nastavku. Na osnovu dijagrama sa slike 3.2, te blokova 2.2 i 2.3 izraditi će se dio programa za unos ulaznih podataka. Ovisno o broju čvorova i elementa proces će se ponavljati dok se ne izvrše svi potrebni unosi.

U okviru unosa podataka čvorovima (dijagram toka na slici 3.3) , ovisno o broju čvorova, kreira se petlja za unos slijedećih podataka:

- Unos koordinate čvora
- Unos sila i momenta koja djeluje u čvoru
- Definiranje mogućnosti pomaka za svaki čvor (mogućnost pomak u smjeru y osi i zakreta u čvoru)

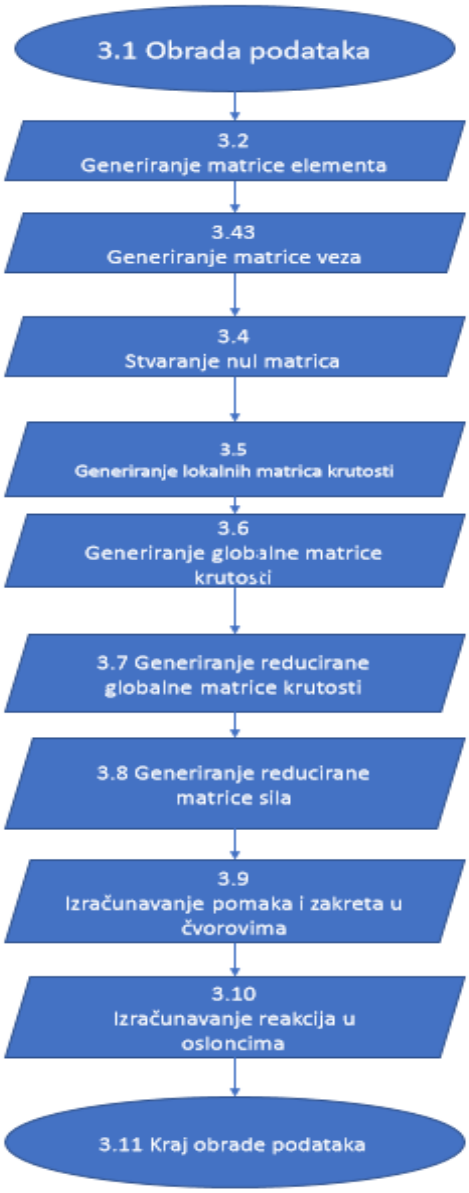
Broj elemenata kod rješavanje problema grede je uvijek za jedan manji od broja čvorova (4 elementa – 5 čvorova.

Na slici 3.4 je prikazan dijagram toka za unos podataka o elementima. Za svaki element unose se slijedeći podaci:

- Unos duljine elementa
- Unos aksijalnog momenta inercije
- Unos Youngovog modula elastičnosti
- Unos kontinuiranih opterećenja za svaki element

Izradom dijagrama toka za ulazne podatke završen je dio unosa informacija o čvorovima i elementima, te se prelazi na srž rješavanja problema - „obradu podataka“.

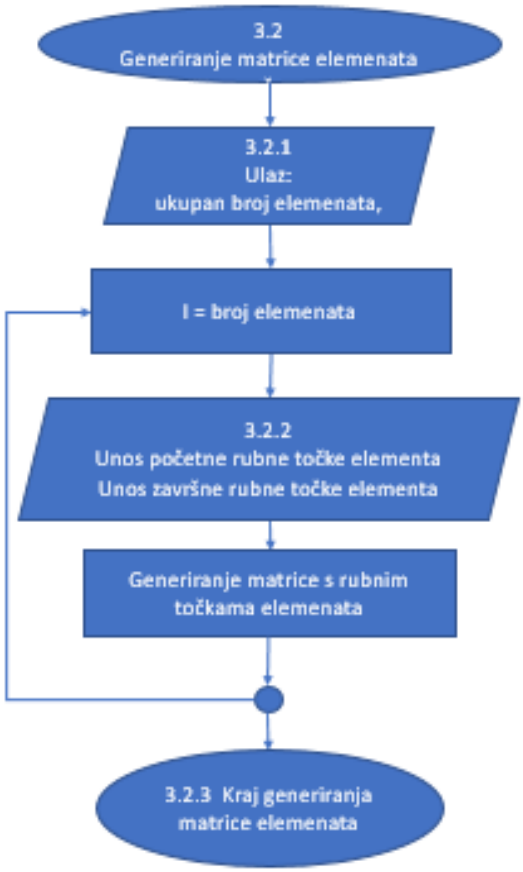
3.2. Dijagrami obrade podataka

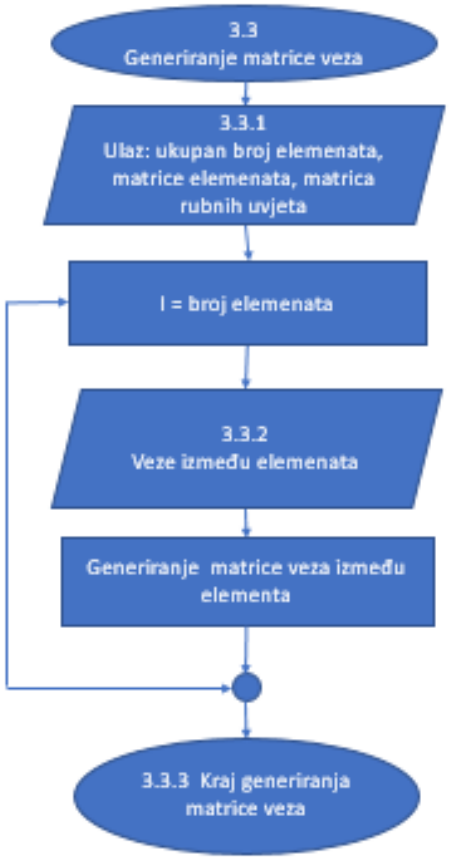


Slika 3.5 – Dijagram obrade podataka

Dijagram toka obrade podataka prikazan na slici 3.5 prikazuje sistematski postupak rješavanje općeg problema savijanja grede kojeg će programski kod koristiti. Kako bi se prikazani postupak mogao prenijeti u program potrebno je definirati određene naredbe za implementaciju metode u računalni jezik.

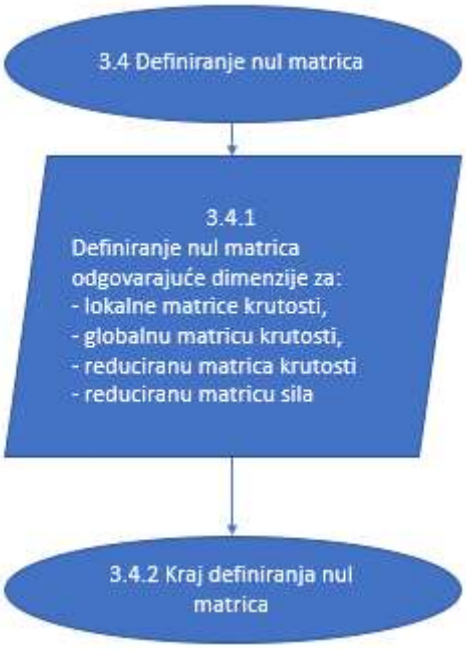
.





Slika 3.6 – Dijagram toka „Matrice elementa“ *Slika 3.7 – Dijagram toka „Matrica veza“*

Na slikama 3.6 i 3.7 prikazani su dijagrami toka za generiranje matrice elementa i matrice veza. Tokom analitičkog rješavanja problema matrica elemenata i matrica veza su praktički nepotrebne, dok za programski način rješavanja neophodne. Korištenjem ranije definirane podatke (početne i završne čvorove elementa i broj elemenata) generira se matrica elementa to jest matrični prikaz broja čvorova između kojih se nalazi konačni element. Na temelju podataka iz matrice elemenata generira se matrica veza. Za svaki element se upisuju redni brojevi veze (svaki čvor ima dvije veze – vertikalni pomak i zakret), tako da svaki element ima u biti dva čvora i četiri veze čiji su pomaci/zakreti omogućeni ili onemogućeni.



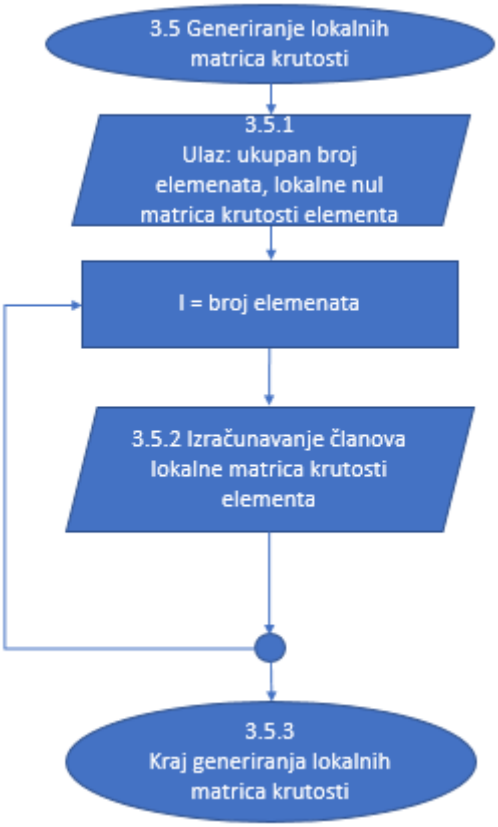
Slika 3.8 – Definiranje nul matrica

Definiranje nul matrica je neophodan dio tokom obrade podataka. Tokom rješavanja zadataka postupak će koristiti veći broj matrica, te je iste potrebno inicijalizirati. Nul matrice su zapravo matrice odgovarajuće dimenzije ispunjene nulama, a koje će se postupno zamijeniti odgovarajućim podatcima. Nul matrice koje se inicijaliziraju su:

- lokalne matrice krutosti (4 x 4)
- globalna matrica krutosti i lokalna matrice krutosti formata globalne matrice krutosti za svaki element zasebno (n x n)
- reducirana matrica krutosti (m x m)
- reducirana matricu sila (n x 1)

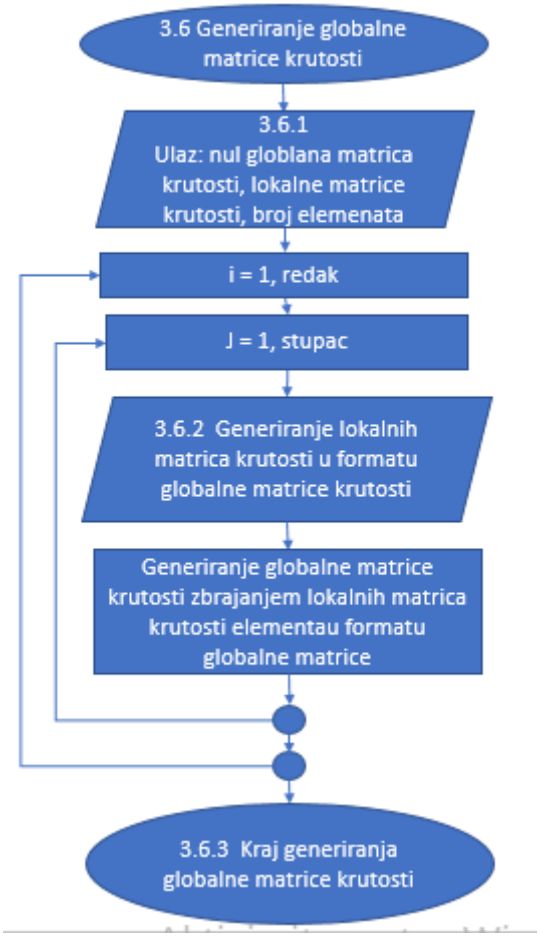
gdje „n“ predstavlja ukupan dvostruku vrijednost ukupnog broja čvorova

Veličina reduciranih matrica ovisi će o rubnim uvjetima . Dimenzija matrice je jednaka razlici ukupnog broja mogućih pomaka/zakreta i broja onemogućenih pomaka/zakreta.



Slika 3.9 – Stvaranje lokalnih matrica krutosti

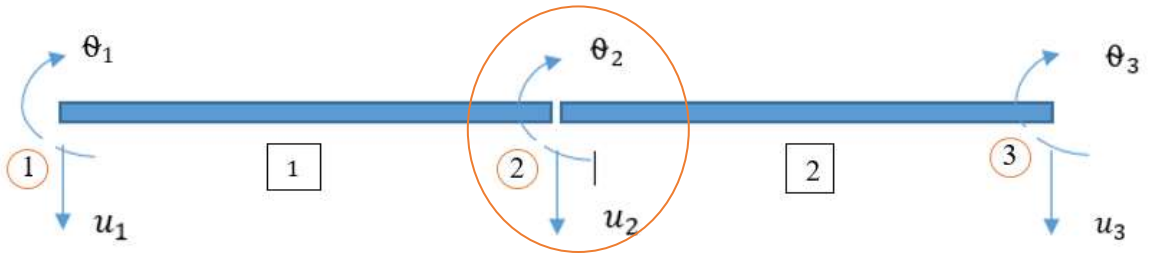
Znajući oblik lokalne matrice krutosti elementa za gredu, te znajući da je određena matrica 4x4 u prije nastaloj nul matrici popunjava se podatcima primjenom formule 1.1. Ovisno o broju elemenata primjenom petlje se generira jednaki broj lokalnih matrica krutosti.



Slika 3.10 – Stvaranje globalnih matrica krutosti

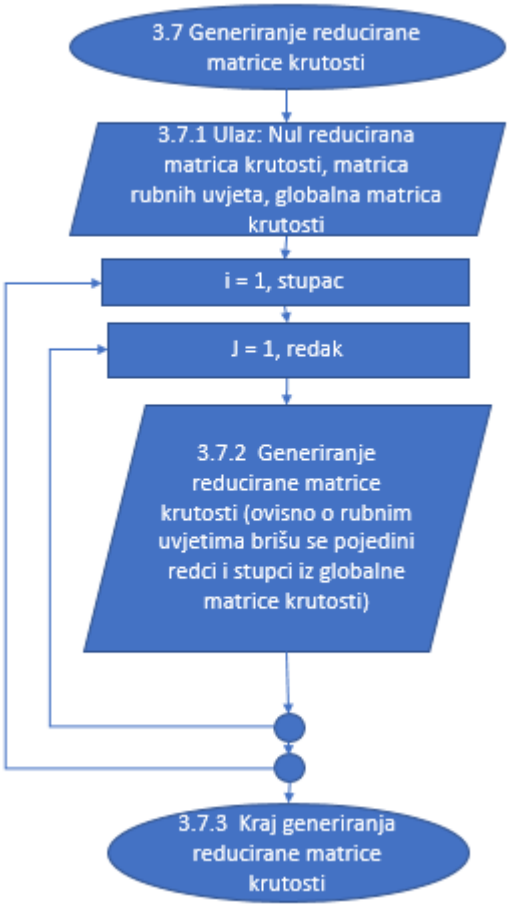
Rješavanjem lokalnih matrica elementa za svaki element posebno dolazi se do dijela gdje sve lokalne matrice treba spojiti u jednu globalnu matricu krutosti što je prikazano na slici 2.3.

Matrica krutosti cijelog sustav sastavlja se od lokalnih matrica krutosti svih elementa. Na mjestu povezivanja dva elementa, to jest u čvorovima, pomaci i zakreti su jednaki za oba elementa.

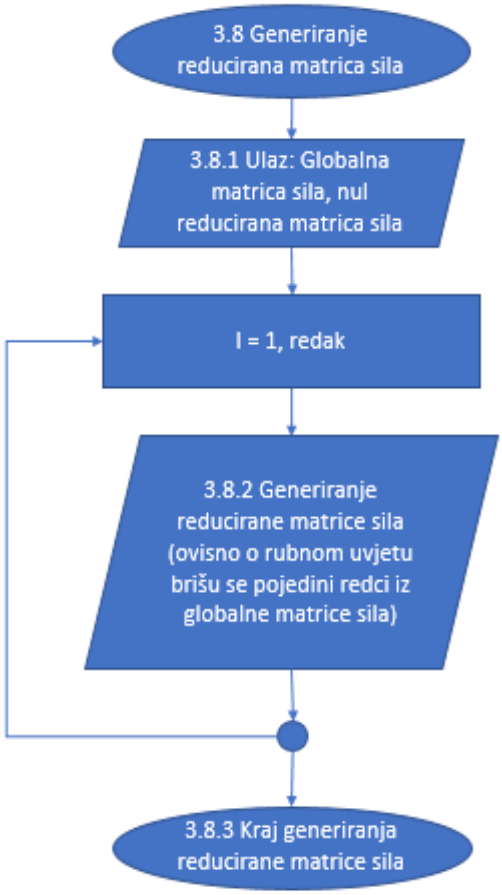


Slika 3.11 – Prikaz pomaka na elementu

Na spoju dva elementa, u slučaju na slici 3.11 u čvoru 2, pomaci i zakreti čvora 2 su zajednički za oba elementa. U matrici krutosti to znači da treba zbrojiti elemente lokalne matrice krutosti koji se odnose isti čvor.

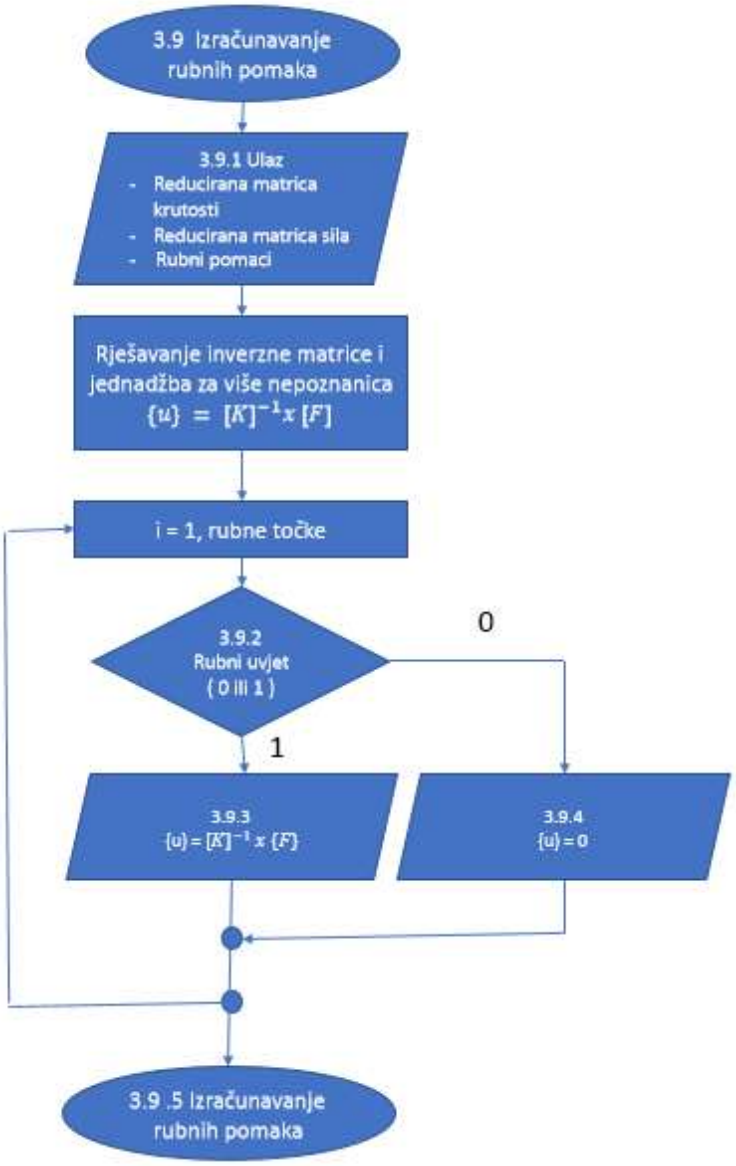


Slika 3.12 – Dijagram generiranje reducirane matrice krutosti



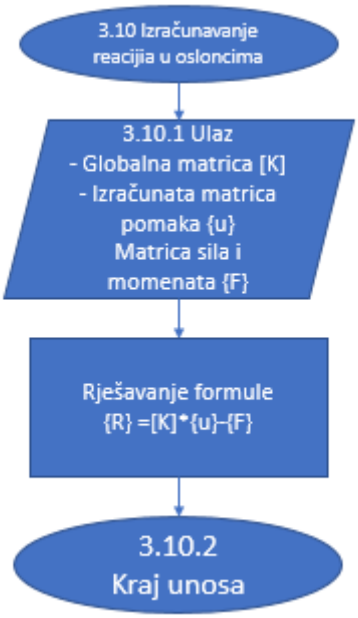
Slika 3.13 – Dijagram generiranja reducirane matrice sila

Generiranje navedena dva dijagrama na Slici 3.12 i Slici 3.13 prikazuju završni korak prije računanja rubnih pomaka u rubnim točkama. Kako bi se mogla izvršiti jednadžba 2.9 potrebno je formirati reducirane matrice. Reducirane matrice se stvaranju ovisno o pomaku u rubnoj točki. U slučaju da nema pomaka u rubnoj točki na mjestu u matrici gdje je određeni pomak (u, θ) ugniježđen, taj red odnosno stupac se brišu. Brisanjem određenih redova i stupca omogućuje se proračun pomaka preko jednadžbe s više nepoznanica.



Slika 3.14 – Dijagram izračunavanja pomaka

Na slici 3.14 je prikazan dijagram toka za računanje pomaka. Da bi se odredili pomaci i zakreti u čvorovima potrebno je riješiti jednadžbu $\{u\} = [K]^{-1} x \{F\}$. Na mjestima gdje su pomaci/zakreti onemogućeni unosi se „0“. Dijagram toka je definiran tako da se prvo izračunaju samo pomaci koji je postoje, a zatim se stvara lista svih pomaka sustava bio on jednak nuli ili proračunat.

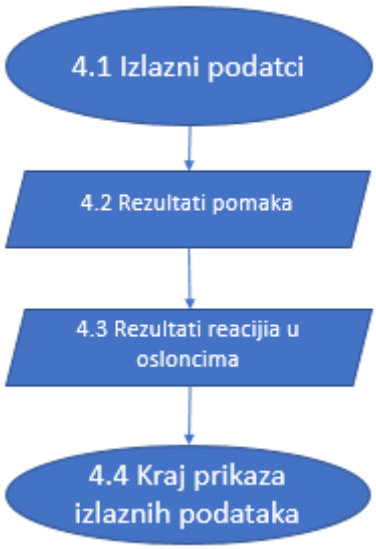


Slika 3.15 – Dijagram izračunavanja reakcija u osloncima

Na slici 3.15 je prikazan dijagram toka određivanja reakcija u osloncima. Izračunavaju se reakcije za svaki čvor iako u svakom čvoru nema oslonca ili uklještenja. Na mjestima gdje nema oslonca ili uklještenja reakcija će biti jednaka 0.

3.3. Dijagrami izlaznih podataka

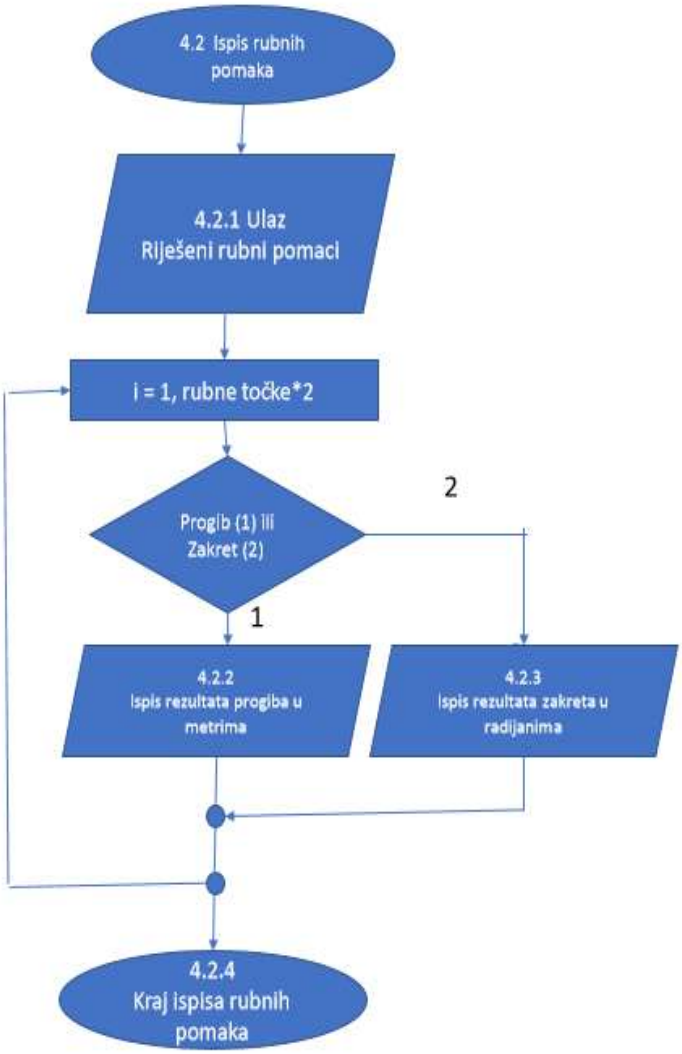
Dijagrami toka izlaznih podataka ukazuju na način prikaza ulaznih podataka, međurezultata i rezultata na zaslonu računala. Sve izračunate vrijednosti se mogu ispisati na zaslon računala ili pohraniti u datoteku. U sljedećem poglavlju prikazani su dijagrami toka za ispisa rezultata tj. rješenja problema grede primjenom MKE.



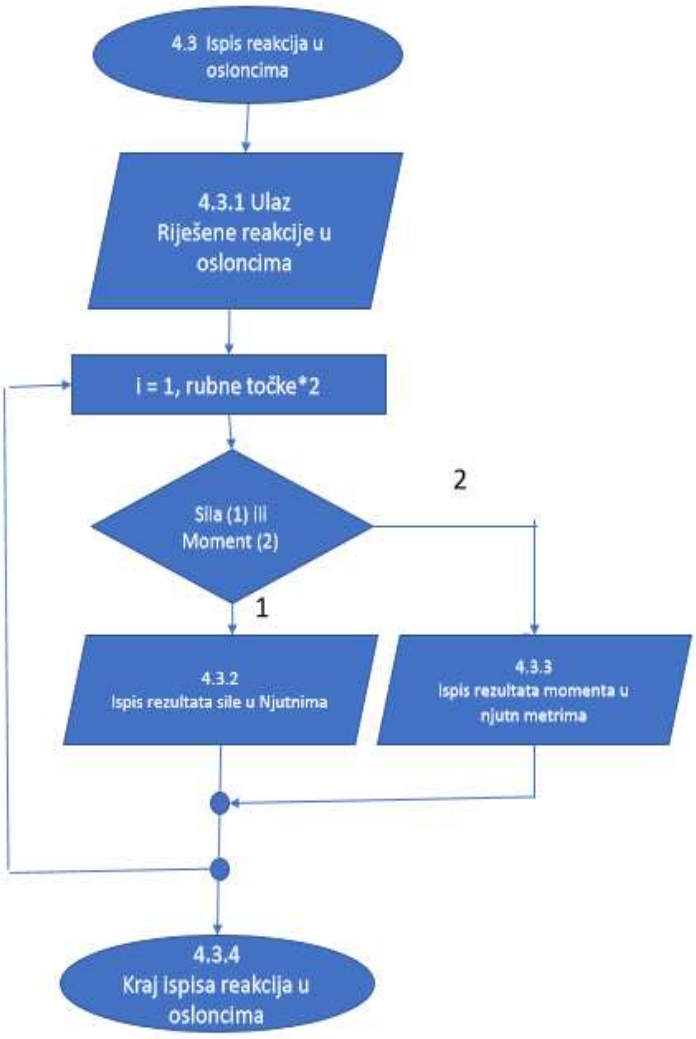
Slika 3.16 – Dijagram izlaznih podataka

Primarni izlazni podatci koji se traže su „Rezultati pomaka“ i „ Rezultati reakcija u osloncima“. U programskom kodu će biti puno više izlaznih podataka koji će biti prikazani, dok će ovdje biti prikazani samo navedeni. Razlog prikaza međurezultata jest provjera ispravnosti rada programa i traženja eventualnih pogrešaka („debugging“).

.



Slika 3.17 – Dijagram toka izlaznih podataka pomaka



Slika 3.18 – Dijagram toka izlaznih podataka reakcija u osloncima

Nakon izrade dijagrama toka prelazi se na izradu programskoga koda. Gotovi dijagrami olakšavaju programiranje.

4. Izrada programskog koda

Na temelju izrađenih dijagrama toka izvršeno je programiranje u programskom jeziku Python. U nastavku su date slike s dijelovima programskog koda i opisom istog uz pozivanje na odgovarajući dijagram toka.

```
1. #=====================================================================
2. # UNOS MODULA ZA RAD PROGRAMA
3. #=====================================================================
4.
5. import numpy
6.
7. #=====================================================================
8. # ODLUKA O UNOŠENJU NOVIH PODATAKA ILI KORIŠTENJU „DEFAULT" PODATAKA
9. #=====================================================================
10.
11.  pocetak = input("Unos podataka(1) ili default podatci(2): ")
12.
13.  if pocetak == "2":
14.      uk_br_elemenata=4
15.      uk_br_cvorova=uk_br_elemenata+1
16.      pocetna_tocka    = [1,2,3,4]
17.      zavrsna_tocka    = [2,3,4,5]
18.      duljina_elementa = [2,1, 0.5, 2]
19.      inercija=[0.000006,0.000006, 0.000006,0.000006]
20.      young=[210000000000,210000000000,210000000000,210000000000]
21.      konstanta=[157500.0,1260000,10080000,157500.0]
22.
23.      sila_kont = [0,-10000,0,0]
24.      info = {'el1':['ne',0], 'el2': ['da',-10000],'el3': ['ne', 0],\
25.              'el4': ['ne', 0]}
26.      Fd={'el0':0,'el5':0,'el1':0, 'el2': -5000.0, 'el3': 0, 'el4':0}
27.      Md={'el0': 0, 'el5': 0, 'el1': 0, 'el2': -833.34,  'el3': 0,\
28.          'el4':0}
29.      sile = [0,-5000,0,0,0] #N
30.      moment = [0,0,0,-5000,0] #Nm
31.      Rubni_uvjeti_progibi_i_rotacije=[0,1,1,1,1,1,0,1,0,0]
32.      F_uk={'sila1': 0,'sila2':-10000.0,'sila3':-5000.0, 'sila4': 0,\
33.            'sila5': 0}
34.      M_uk={'moment1': 0, 'moment2': -833.333334,\
35.            'moment3': 833.333334, 'moment4': -5000, 'moment5': 0}
36.  else:
37.      uk_br_elemenata = int(input('Ukupan broj broj elemenata je: '))
38.      uk_br_cvorova = uk_br_elemenata+1
39.
```

Slika 4.1 – Početak programa i odabir o unosu novih ili korištenju unaprijed određenih ulaznih podataka

Na početku programskog koda vrši se uvoz modula koji će olakšati izradu programa (slika 4.1). Umetnuti modul je „*numpy*“ koji omogućava brzo izvršavanje kompliciranih matematičkih funkcija. Modul se poziva korištenjem naredbe „import“ (redak 5).

Izrada koda na slici 4.1 se vrši na temelju dijagrama toka prikazanog na slici 3.2 (Primarni dijagram unosa podataka). Redak 11 omogućava korisniku unos podataka putem tipkovnice, tj. odluku dali želi unijeti nove ili koristiti unaprijed definirane („*default*“) ulazne podatke. Unaprijed definirani podatci su dio samog programskog koda i pokreću se unosom broj 2. Unosom broja 1 započinje unos ulaznih podataka. Unos podataka započinje unosom ukupnog broja konačnih elemenata (redak 36). Potom program računa ukupan broj čvorova (koji je u slučaju grede za 1 veći od broja elemenata).

```
40. ======================================================================
41. # UNOS/UČITAVANJE ULAZNIH PODATAKA O ELEMENTIMA (E, I, L, KONT.SILA)
42. ======================================================================
43.
44.     pocetna_tocka    = []
45.     zavrsna_tocka    = []
46.     duljina_elementa = []
47.     inercija         = []
48.     young            = []
49.     konstanta        = []
50.
51.     for i in range(uk_br_elemenata):
52.         a = int(input("Element " +str(i+1)+ "pocinje na čvoru : "))
53.         b = int(input("Element " +str(i+1)+ "završava na čvoru: "))
54.         L = float(input("Duzina elemeta "+str(i+1)+ " u [m] je: "))
55.         I = float(input("Aksijalni moment inecije [m4] je: "))
56.         E = float(input('Youngov modul [N/m2] je: '))
57.         con = float((E*I/(L**3)))
58.
59.         pocetna_tocka.append(a)
60.         zavrsna_tocka .append(b)
61.         duljina_elementa.append(L)
62.         inercija.append(I)
63.         young.append(E)
64.         konstanta.append(con)
```

Slika 4.2 – Unos ulaznih podataka

Priloženi kod na slici 4.2 je izrađen na temelju dijagrama toka prikazanog slici 3.4 (dijagram toka za unos podataka o elementima). U linijama programa 44-49 se inicijaliziraju liste . Inicijalizirane liste su jednodimenzionalne matrice koje se u petlji popunjavaju podacima. Unos podataka se vrši u „*for*“ petlji (počev od retka 52). U svakom prolazu kroz petlju upisuje se po jedan podatak u ranije inicijalizirane liste. Ukupan broj prolaza kroz petlju jednak je ukupnom broju konačnih elemenata. Za svaki element korisnik upisuje zadatkom određene podatke. Za unos decimalnih brojeva koristi se „*float* (*input*)“. Početne liste se popunjavaju s novo unesenim podatcima koji se koriste dalje u programu.

```
65.     kont      = []
66.     sila_kont = []
67.     info      = {}
68.
69.     kon_po_elementu=(input("je li na elementu "+str(i+1)+ \
70.                            " kontinuirano opterecenje (da/ne): "))
71.     kont.append(kon_po_elementu)
72.     if kont[i] == "da":
73.        sila_kont.append(float(input("koliko je kont. \
74.                                     opterecenje na elementu "\
75.                                     +str(i+1)+": ")))
76.     else:
77.         sila_kont.append(0)
78.     info["el"+str(i+1)] = [kont[i],sila_kont[i]]
79.
80.     Fd = {"el0": 0, "el"+str(uk_br_cvorova): 0}
81.     Md = {"el0": 0, "el"+str(uk_br_cvorova): 0}
82.
83.     if info["el"+str(i+1)][0] == "da":
84.         Fd["el"+str(i+1)] = duljina_elementa[i]*info["el"+ \
85.            str(i+1)][1]/2
86.     else:
87.         Fd["el"+str(i+1)] = 0
88.
89.     if info["el"+str(i+1)][0] == "da":
90.         Md["el"+str(i+1)] = ((duljina_elementa[i]**2)*\
91.           info["el"+str(i+1)][1])/12
92.     else:
93.         Md["el"+str(i+1)] = 0
```

Slika 4.3 – Unos kontinuiranih opterećenja i određivanje ekvivalentnog opterećenja po elementima

Ulaz podataka o elementima završava unosom podataka o opterećenjima u „*for*“ petlji u retku 73 (slika 4.3). Ovaj dio programa je izrađen na temelju dijagrama toka na slici 3.4.. Provjera dali na nekom elementu postoji kontinuirano opterećenje vrši se preko „*if*“ petlje (redak 74). Ukoliko na promatrani element djeluje kontinuirano opterećenje korisnik upisuje brojčanu vrijednost, a u slučaju ako nema kontinuiranog opterećenja program za promatrani element dodjeljuje vrijednost 0.

Za razliku od prije inicijalizirane liste u dijelu koda pojavljuje se rječnik unutar vitičastih zagrada („*python dictionary*“ - redak 67. Za razliku od obične liste rječnik („*dictionary*“) omogućava zapis podataka s ključem (podaci mogu biti brojčani, tekstualni, logički i sl.). U rječniku je prikazano stanje za svaki postojeći element, to jest postoji li kontinuirano opterećenje i iznos sile.

Primjenom podataka iz rječnika računa se ekvivalentna poprečna sila i ekvivalentni moment savijanja.

Ulaz podataka završava dijagramom toka na slici 3.3 (Dijagram toka za unos podataka o čvorovima). Ovaj dio je realiziran dijelom programa koji je prikazan na slici 4.4 . Na čvorovima se postavljaju sile i momenti te se definiraju pomaci čvorova.

```
94. #=================================================================
95. # UNOS PODATAKA O ČVOROVIMA - sile, momenti i rubni pomaci
96. #=================================================================
97.
98.     Sile   =[]
99.     Moment =[]
100.
101.    for i in range(uk_br_cvorova):
102.            sile1=(float(input("kolika je sila F" + str(i+1) + \
103.                               "na čvoru "+str(i+1)+": ")))
104.
105.            moment1=(float(input("koliki je moment M" + str(i+1)+\
106.                                 "na čvoru "+str(i+1)+": ")))
107.            sile.append(sile1)
108.            moment.append(moment1)
```

Slika 4.4 – Unos ulaznih podataka o čvorovima

Unos podataka se vrši u „*for*“ petlji (počev od retka 101, slika 4.4). U svakom prolazu kroz petlju upisuje se po jedan podatak u inicijalizirane liste „*Sile*“ (redak 98) i „*Moment*“ (redak 99). Ukupan broj prolaza kroz petlju jednak je ukupnom broju čvorova. Ovisno o ukupnom broju čvorova korisnik upisuje iznos sile i momenta za svaki čvor naredbom „*float(input)*“. Naredba „*float*“ omogućuje programu rad s decimalnim brojevima.

```
109. #=================================================================
110. # MJESTO DJELOVANJA  RUBNIH UVJETA
111. # u1,3,5,7... OZNAKA ZA PROGIBE
112. # u2,4,6,8... OZNAKA ZA ROTACIJE
113. #=================================================================
114.     print("================================================")
115.     print(" 0 predstavlja onemogućen zakret/pomak")
116.     print(" 1 predstavlja slobodan zakret/pomak")
117.     print(("================================================")
118.     for i in range(uk_br_cvorova):
119.         Rubni_uvjeti_progibi_i_rotacije=[]
120.
121.         progib=int(input("Progib u"+str(i+1)+"u čvoru "\
122.                                 +str(i+1)+" je {0 ili 1}: "))
123.         rotacija = int(input("Zakret v"+str(i+1)+" u čvoru "\
124.                                     +str(i+1)+" je {0 ili 1}: "))
125.
126.         Rubni_uvjeti_progibi_i_rotacije.append (progib)
127.         Rubni_uvjeti_progibi_i_rotacije.append (rotacija)
128.
129.         print(Rubni_uvjeti_progibi_i_rotacije)
```

Slika 4.5 – Unos rubnih uvjeta

Rubni uvjeti (slika 4.5) unose se preko „*for*“ petlje (slika 4.5, redak 118) ovisnoj o ukupnom broju čvorova. Za svaki čvor na elementu korisnik unosi stanje o pomaku i zakretu 0 ili 1 (redak 121 - 124). Nula definira onemogućen pomak/zakret dok jedinica definira slobodan pomak/zakret. Kao i do sada korisnik upisuje stanje čvora preko naredbe „*input*“. Slika 4.5 prikazuje tvorbu matrice sila.

```
130. #=================================================================
131. # ZBRAJANJE SILA I MOMENATA S EKVIVALENTNIM VRIJEDNOSTIMA
132. #=================================================================
133.
134.       F_uk = {}
135.       M_uk = {}
136.
137.       F_uk["sila"+str(i+1)] = sile[i] + Fd["el"+str(i)]\
138.           + Fd["el"+str(i+1)]
139.
140.       M_uk["moment"+str(i+1)] = moment[i] - Md["el"+str(i)]\
141.           + Md["el"+str(i+1)]
142.
143.       print(F_uk)
144.       print(M_uk)
145.
146. #=================================================================
147. #UVEZIVANJE SILA I MOMENATA U MATRICU SILA
148. #=================================================================
149.
150. sile_i_momenti = []
151. for i in range(uk_br_cvorova):
152.     sile_i_momenti.append(F_uk["sila"+str(i+1)])
153.     sile_i_momenti.append(M_uk["moment"+str(i+1)])
154.
155. print("Matrica sila i momenata= ", sile_i_momenti)
```

Slika 4.6 – Stvaranje matrice sila

Radi kreiranja matrice sila potrebno je sile i momente (slika 4.4, redak 98 i 99) u čvorovima zbrojiti s ekvivalentnim vrijednostima proizišlim iz kontinuiranog opterećenja (slika 4.3). Ekvivalentne vrijednosti se mijenjaju ovisno o fizikalnim svojstvima elementa koja se računaju. Svakom čvoru između dva elementa se pridodaju ekvivalentne vrijednosti izračunate za oba elementa što znači da se na spoju elemenata, u slučaju kontinuiranog opterećenja, uvijek događa jedan proračun više (redak 137 i 140).

U novo nastalim inicijalnim listama *F_uk* (red 134) i *M_uk* (redak 135) uporabom „*for*“ petlje vrši se popunjavanje navedenih listi. Na spoju elementa (čvorovima) ukupna sila/moment je prikazana kao suma aktivnih i ekvivalentnih sila/momenata (redak 137-141). Uz zadnji brojač se nalazi *„(i+1)“* koji omogućava uzimanje svakog drugog člana iz liste. U slučaju kontinuiranog opterećenja koje se prenosi na više elemenata u sumi će uvijek postojati jedan

član više, te će proračunom biti pridodan. Sve ukupne sile i momenti su objedinjeni u zasebnu listu (redak 150 – 153).

Popunjenim vektorom sila završena je priprema ulaznih podataka potrebnih za rad programom. U sljedećem dijelu prikazan je tok obrade podataka putem koda, te potrebnih funkcija za dolazak do rješenja.

```
156. ======================================================================
157. #PRIKAZ MEĐUSOBNIH VEZA IZMEĐU ELEMENATA
158. ======================================================================
159. 
160. Elementi=[]
161. for i in range(int(uk_br_elemenata)):
162.     Priv=[]
163.     Priv.append(pocetna_tocka[i])
164.     Priv.append(zavrsna_tocka[i])
165.     Elementi.append(Priv)
166. #print(Elementi)
```

Slika 4.7 - Veza između elemenata

Obrada podataka započinje dijelom programa na slici 4.7, a temeljenom dijagrama toka na slici 3.6 (*Dijagram toka „Matrice elementa“)*. Matrica elementa povezuje svaki konačni element s pripadajućim čvorovima. Za razliku od prijašnjih lista, navedena lista „*Priv*“ redak 162 se nalazi unutar „*for*“ petlje. Time je omogućena reinicijalizacija liste unutar petlje.

```
167. ======================================================================
168. # GENERIRANJE MATRICE VEZA
169. ======================================================================
170. 
171. Matrica_veza=[]
172. for i in range(uk_br_elemenata):
173.     Matrica_veza_privremena=[]
174. 
175.     Matrica_veza_privremena.append(Elementi[i][0]*2-1)
176.     Matrica_veza_privremena.append(Elementi[i][0]*2)
177. 
178.     Matrica_veza_privremena.append(Elementi[i][1]*2-1)
179.     Matrica_veza_privremena.append(Elementi[i][1]*2)
180.     Matrica_veza.append(Matrica_veza_privremena)
```

Slika 4.8 – Generiranje matrice veza

Na osnovu slike 3.7 (dijagram toka „Matrica veza“) izrađuje se dio koda prikazan na slici 4.8. U „*for*“ petlji vrši se popunjavanje matrice veza (redak 171). U svrhu popunjavanja navedene matrice kreira se nova lista unutar „*for*“ petlje „Matrica_veza_privremena“ (redak 173) u koju se pohranjuju privremeni podatci potrebni za kreiranje matrice veza. Generiranjem matrice veza

dobiva se matrica s brojčanom oznakom pomaka čvorova za svaki element (svaki element ima dva čvora, a svaki čvor jedan vertikalni pomak i jedan zakret – dakle pomak svakog elementa je definiran s četiri „pomaka“) .

```
181. #=====================================================================
182. #FORMIRANJE NUL MATRICA
183. #=====================================================================
184. n=uk_br_cvorova
185. m=Rubni_uvjeti_progibi_i_rotacije
186.
187. lok_mat_el=numpy.zeros((uk_br_elemenata,4,4),dtype='int64')
188.
189. glob_mat_el=numpy.zeros((uk_br_elemenata,n*2,n*2),dtype='int64')
190.
191. Globalna_matrica=numpy.zeros((n*2,n*2),dtype='int64')
192.
193. red_mat_krutost=numpy.zeros((n*2-m.count(0),n*2-
194.                              m.count(0)),dtype='int64')
195.
196. F_reducirana=numpy.zeros((n*2-m.count(0)),dtype='int64')
```

Slika 4.9 – Formiranje nul matrica

Dijagram toka na slici 3.8 (Dijagram definiranje nul matrica) prikazuje inicijalizaciju nul matrica koje će biti korištene u proračunu. Na slici 4.9 prikazan je kod za kreiranje matrica koje je potrebno inicijalizirati:

- Lokalna matrica krutosti za svaki element (redak 187)
- Globalna matrica krutosti za svaki element (nije potrebna u analitičkom proračunu, redak 189)
- Globalna matrica krutosti (redak 191)
- Reducirana matrica krutosti (redak 193)
- Reducirana matrica sila (redak 196)

Nul matrice se definiraju pomoću primjenom funkcije „*zeros*“ koja je dio modula „*numpy*“. Sve matrice definirane su dimenzijom (brojem redaka i stupaca, te tipom podataka s kojima će biti popunjene).

Inicijalizacijom svih nul matrica omogućen je daljnji proračun. Na slici 3.9 prikazan je dijagram toka za generiranje lokalnih matrica krutosti elementa.

```
197. #=================================================================
198. # GENERIRANJE LOKALNIH MATRICA KRUTOSTI ZA SVAKI POJEDINI ELEMENAT
199. #=================================================================
200. 
201. for i in range(uk_br_elemenata):
202.     L = float(duljina_elementa[i])
203.     con = float(konstanta[i])
204.     lok_mat_el[i][0][0]=12.e+00*con
205.     lok_mat_el[i][0][1]=6.e+00*L*con
206.     lok_mat_el[i][0][2]=-12.e+00*con
207.     lok_mat_el[i][0][3]=6.e+00*L*con
208. 
209. 
210.     lok_mat_el[i][1][0]=6.e+00*L*con
211.     lok_mat_el[i][1][1]=4.e+00*L**2*con
212.     lok_mat_el[i][1][2]=-6.e+00*L*con
213.     lok_mat_el[i][1][3]=2.e+00*L**2*con
214. 
215.     lok_mat_el[i][2][0]=-12.e+00*con
216.     lok_mat_el[i][2][1]=-6.e+00*L*con
217.     lok_mat_el[i][2][2]=12.e+00*con
218.     lok_mat_el[i][2][3]=-6.e+00*L*con
219. 
220.     lok_mat_el[i][3][0]=6.e+00*L*con
221.     lok_mat_el[i][3][1]=2.e+00*L**2*con
222.     lok_mat_el[i][3][2]=-6.e+00*L*con
223.     lok_mat_el[i][3][3]=4.e+00*L**2*con
```

Slika 4.10 – Generiranje lokalne matrice krutosti

Na slici 4.10 prikazano je popunjavanje lokalne matrice krutosti matrice „*lok_mat_el*“ (slika 4.9, redak 187). Unutar „*for*“ petlje vrši unos 16 članova lokalne matrice krutosti. U svakom prolazu kroz petlju generira se lokalna matrica krutosti za jedan element. Svaki član lokalne matrice krutosti pomnožen je s konstantom $\frac{E \cdot I}{L^3}$. Prolaskom petlje od redka 204 - 223. generiraju se svi članovi lokalne matrice krutosti jednog elementa. Ukupan broj prolaza kroz petlju jednak je ukupnom broju konačnih elemenata.

Na temelju dijagrama toka sa slike 3.10 (Generiranje globalne matrice krutosti) izvodi se povezivanje lokalnih matrica krutosti u jednu globalnu matricu krutosti primjenom višestrukih petlji. Višestruke petlje omogućavaju prolazak po cijelom globalnom sustavu matrice.

```
224. #=================================================================
225. # GENERIRANJE LOKALNIH MATRICA KRUTOSTI U FORMATU GLOBALNE MATRICE
226. # KRUTOSTI
227. #=================================================================
228. for el_br in range(uk_br_elemenata):
229.     i1=-1
230.     for i in Matrica_veza[el_br]:
231.         i1+=1
232.         j1=-1
233.         for j in Matrica_veza[el_br]:
234.             j1+=1
235.             globalna_mat_el[el_br][i-1][j-
236.                 1]+=lok_mat_el[el_br][i1][j1]
237.
```

Slika 4.11 – Zapis lokalnih matrica krutosti u formatu globalne matrice krutosti

Slika 4.11 prikazuje korištenje trostruke „*for*“ petlje koja programu omogućava kreiranje lokalnih matrica krutosti u formatu globalne matrice krutosti. U ranije inicijaliziranoj nul matrici „globalna „*mat_el*“ (slika 4.9, redak 218) unutar „*for*“ petlji se popunjavaju lokalne matrice krutosti (formata globalne) s podatcima iz lokalnih matrica krutosti elemenata „*lok_mat_elementa*“ (slika 4.9). Ukupan broj novo nastalih matrica jednak je broju konačnih elemenata. Svaka „*globalna_mat_el*“ (redak 235) će biti različita po lokaciji članova spremljenih u matrici.

```
238.    #=================================================================
239.    # GENERIRANJE GLOBALNE MATRICE KRUTOSTI
240.    #=================================================================
241.
242.    for el_br in range(uk_br_elemenata):
243.
244.        for i in range(uk_br_cvorova*2):
245.
246.            for j in range(uk_br_cvorova*2):
247.
248.                Globalna_matrica[i][j]+=globalna_mat_el[el_br][i][j]
```

Slika 4.12 – Formiranje globalne matrice krutosti

Globalna matrica krutosti dobiva se zbrajanjem svih lokalnih matrica krutosti (u formatu globalne matrice). Za generiranje globalne matrice krutosti korištena je trostruka „*for*“ petlja (slika 4.12).

```
249. #=================================================================
250. #REDUCIRANA MATRICA KRUTOSTI
251. #=================================================================
252. 
253. i1=-1
254. j1=-1
255. pomaci_nazivi=[]
256. for i in range(uk_br_cvorova*2):
257.     if (Rubni_uvjeti_progibi_i_rotacije[i]==1):
258.         i1+=1
259.         pomaci_nazivi.append("u"+str(i+1))
260.         j1=-1
261.         for j in range(uk_br_cvorova*2):
262.             if (Rubni_uvjeti_progibi_i_rotacije[j]==1):
263.                 j1+=1
264.                 red_mat_krutosti[i1][j1]=Globalna_matrica[i][j]
```

Slika 4.13 – Reducirana matrica krutosti

Izraz koda slici 4.13 dobiven je iz dijagram toka na slici 3.12 (Dijagram generiranja reducirane matrice krutosti), te prikazuje način implementacije globalne matrice krutosti u formatu reducirane matrice. Korištenjem dvostruke „*for*“ petlje na osnovu stanja pomaka čvorova kreira se reducirana matrica krutosti. U slučaju omogućenog pomaka „1“ globalna matrica ostaje ne promijenjena, a u slučaju onemogućenog pomaka „0“ globalna matrica se reducira za odgovarajući redak i stupac.

```
265. #=================================================================
266. #REDUCIRANA MATRICA SILA I MOMENATA
267. #=================================================================
268. 
269. i1=-1
270. for i in range(uk_br_cvorova*2):
271.     if (Rubni_uvjeti_progibi_i_rotacije[i]==1):
272.         i1+=1
273.         F_reducirana[i1]=sile_i_momenti[i]
```

Slika 4.14 – Formiranje reducirane matrice sila

Dio programa slici 4.14 se temelji na dijagramu toka sa slike 3.13 (Dijagram generiranja reducirane matrice sila i momenata). Reducirana matrica sila kreira se prolaskom „*for*“ petlje preko ukupnog broja čvorova. Ovisno dali je pomak onemogućen „0“ ili omogućen „1“ (redak 271) u čvoru reducira se navedena matrica. Reducirani vektor sila popunjava se podatcima iz liste „*sile_i_momenti*“ (redak 273). Rješavanjem reduciranih matrica programski kod je pripremljen za računanje svih pomaka i zakreta. Slika 3.15 (Dijagram izračunavanja pomaka*)* i slika 3.16 (Dijagram izračunavanja reakcija u osloncima) prikazuju dijagrame toka tj. zadnje faze obrade podataka prije ispisa rezultata.

Dijagram toka na slici 4.15 prikazuje rješavanje matrične jednadžbe 2.9. Python funkcijom „*numpy.linalg.solve*“ jednostavno rješava matrične jednadžbe s većim brojem nepoznanica, te višestruko olakšava programiranje.

```
274. #=================================================================
275. # ODREĐIIVANJE POMAKA (PROGIBA I ROTACIJA)
276. #=================================================================
277. pomaci = numpy.linalg.solve(red_mat_krutosti, F_reducirana)
278.
```

Slika 4.15 – Izračunavanje rubnih pomaka

```
279. #=================================================================
280. # ODREĐIVANJE REAKCIJA U OSLONCIMA
281. #=================================================================
282.
283. reakcije = numpy.dot(Globalna_matrica, pomaci_svi)
284. reakcije_ukupno = reakcije - sile_i_momenti
285. print(reakcije_ukupno)
```

Slika 4.16 – Izračunavanje reakcija u osloncima

Slika 4.16 prikazuje određivanje reakcija u osloncima koristeći Python-ovu naredbu „*numpy.dot*“. „*numpy.dot*“ je dio modula „*numpy*“ kojoj je funkcija množenje matrica. Množenjem matrica kreira se matrica n x 1 od koje se oduzima vektor sila (ukupne sile i momenti, redak 150).

Obradom posljednje dvije funkcije završava se s obradom podataka. Sljedeći korak je ispisivanje rezultata na zaslonu računala. Tijekom izrade programskog koda više puta se pojavljuje naredba „*print*“. Naredba „*print*“ služi za ispisivanje podataka na zaslonu, te za ispisivanje međurezultata. Međurezultati osiguravaju korisniku da provjeri ispravnosti rada programa i traženje eventualnih pogrešaka („debugging“) u fazi testiranja programa. Na slici 3.17 (*Dijagram toka izlaznih podataka pomaka)* i slici 3.18 (*Dijagram toka izlaznih podataka reakcija u osloncima)* prikazani su samo primarni rezultat koji se traže u fazi korištenja programa. To su rezultati pomaka i rezultati reakcija u osloncima.

```
286. print(„Oznake pomaka“)
287. print(pomaci_nazivi)
288. print(„Pomaci čvorova=“)
289. for member in pomaci:
290.     print („%9.6f“ % member)
```

Slika 4.16 – Formiranje ispisa izračunatih pomaka

Slika 4.16 od reda 286 do reda 290 prikazuje kod za ispis izračunatih pomaka. To su svi pomaci koji nisu 0, to jest računati su pomoću reduciranih matrica.

```
Oznake pomaka
['u2', 'u3', 'u4', 'u5', 'u6', 'u8']
Pomaci čvorova=
-0.003305
-0.003726
 0.001020
-0.001175
 0.003119
 0.001176
```

Slika 4.17 – Ispis izračunatih pomaka

Slika 4.17 prikazuje ispis traženih pomaka. Radi urednijeg pregleda napravljen je dodatni kod prikazan na slici 4.18. Od reda 294 do reda 300 prikazan je dio koda koji posprema sve podatke o pomacima u ranije inicijaliziranu listu „*pomaci_svi*“ (redak 293). Razlog nastalog koda je kreiranje liste kompletiranjem pomaka svih čvorova radi olakšanog i urednijeg snalaženja u programu.

```
291. j1=-1
292. sile_nazivi=[]
293. pomaci_svi=[]
294. for i in range(uk_br_cvorova*2):
295.     if (Rubni_uvjeti_progibi_i_rotacije[i]==0):
296.         sile_nazivi.append("F"+str(i+1))
297.         pomaci_svi.append(0)
298.     else:
299.         j1+=1
300. pomaci_svi.append(pomaci[j1])
```

Slika 4.18 – Kod za dobivanje liste „pomaci_svi“

Kreirana lista „*pomaci_svi*“ implementira se u daljnji kod na slici 4.19

```
301. print("Pomaci na čvorovima=")
302. for i in range(len(pomaci_svi)):
303.     if i%2 == 0: print("u"+str(i+1)+"= ""%9.6f"%pomaci_svi[i], "m")
304.     else: print("u"+str(i+1)+"= ""%9.6f" % pomaci_svi[i], "rad")
```

Slika 4.19 – Kod za ispisivanje liste „pomaci_svi“

Priloženi kod na slici 4.19 omogućuje raspisivanje svih pomaka u čvorovima, te ispis mjerne jedinice za pomake. Prilikom kreiranja „*for*“ petlji javlja se funkcija „*len*“. Funkcija „*len*“ predstavlja dužinu liste u zadanoj petlji. Bez navedene funkcije petlja ne bi mogla uzimati

podatke preko petlje. Redovi 303 i 304 omogućuju ispisivanje rezultata pomaka u različitim mjernim jedinicama.

```
Pomaci na čvorovima=
u1=  0.000000 m
u2= -0.003305 rad
u3= -0.003726 m
u4=  0.001020 rad
u5= -0.001175 m
u6=  0.003119 rad
u7=  0.000000 m
u8=  0.001176 rad
u9=  0.000000 m
u10= 0.000000 rad
```

Slika 4.20 – Konačni rezultati svih pomaka

Posljednji dio koda je izrađen prema dijagramu toka na slici 3.18 (Dijagram ispisa reakcija u osloncu).

Rezultati reakcija u osloncima su definirani na isti način kao i rezultati pomaka.

```
305. print("============================================================")
306. print(" F1,3,5,7... predstavlja reakcijske Sile")
307. print(" F2,4,6,8... predstavlja reakcijske Momente")
308. print("============================================================")
309.
310. #print(reakcije_u_osloncima)
311. print("Oznake sila")
312. print(sile_nazivi)
313.
314. print("Reakcije u osloncima=")
315. for i in range(len(reakcije_ukupno)):
316.     if i%2 == 0:
317.         print("F"+str(i+1)+"= ""%9.6f" % reakcije_ukupno[i], "N")
318.     else: print("F"+str(i+1)+"= ""%9.6f" % reakcije_ukupno[i],"Nm")
```

Slika 4.21 – Kod za ispis rezultata reakcija u osloncima

```
==========================================
F1,3,5,7... predstavlja reakcijske Sile
F2,4,6,8... predstavlja reakcijske Momente
==========================================
Oznake sila
['F1', 'F7', 'F9', 'F10']
Reakcije u osloncima=
F1= 2724.530612 N
F2=  0.000000 Nm
F3=  0.000000 N
F4=  0.333333 Nm
F5= -0.000000 N
F6= -0.333333 Nm
F7= 14498.576531 N
F8= -0.000000 Nm
F9= -2223.107143 N
F10= 1482.071429 Nm
```

Slika 4.22 – Ispis rezultata reakcija u osloncima

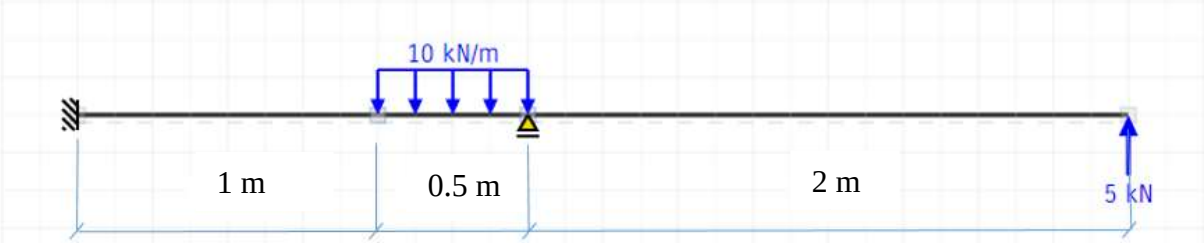
5. Testiranje programa i rješavanje analitičkog načina rješavanja grede MKE

Postupak pri analitičkom rješavanju problema poprečno opterećene grede metodom konačnih elemenata:

- Greda se oslobađa od veza, te rastavlja na konačne elemente i čvorove
- Na temelju rastavljenih elemenata uočava se ukupan broj pomaka koji djeluju na čvorovima
- Na temelju fizikalnih podataka dobiti lokalne i globalne matrice krutosti
- Izračun ukupnog djelovanja sila i zapisa u matričnom obliku
- Na temelju matrične jednadžbe izračunati pomake u čvorovima i reakcije u osloncima

5.1. PRIMJER 1

Na slici 5.1 prikazana je jednostavna poprečno opterećena greda opterećena koncentriranom silom i kontinuiranim opterećenjem. Na temelju zadanih podataka metodom konačnih elemenata odrediti sve pomake u čvorovima i reakcije u osloncima (u = ?, θ = ?, $[R]$ =?). ($R_{1,3,5..}$-predstavlja reakcijske sile, $R_{2,4,6..}$- predstavlja reakcijske momente)!



Slika 5.1 – Primjer br. 1 [6]

F = 5000 N

Q = 10000 N/m

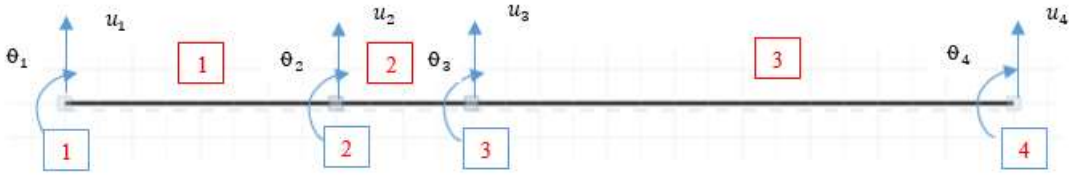
E = 210 GPa = 210000000000 N/m^2

I = 4 * 10^{-6} m^4

$F_1 = \emptyset$ $F_2 = \emptyset$ $F_3 = \emptyset$ $F_4 = 5000$ N

$M_1 = \emptyset$ $M_2 = \emptyset$ $M_3 = \emptyset$ $M_4 = \emptyset$

Na slici 5.2 prikazana je raspodjela grede na elemente, čvorove i pomake koji djeluju na čvor. Greda se dijeli na ukupno 3 elementa i 4 čvora.



Slika 5.2 – Greda s označenim elementima, čvorovima i pomacima

$u_1 = \emptyset$ $u_2 = ?$ $u_3 = \emptyset$ $u_4 = ?$

$\theta_1 = \emptyset$ $\theta_2 = ?$ $\theta_3 = ?$ $\theta_4 = ?$

KONSTANTE KRUTOSTI ZA SVE ELEMENTE

Element 1

$$\frac{E_2 \cdot I_2}{L_2^3} = \frac{210000000000 \cdot 4 \cdot 10^{-6}}{1^3} = 84000 \tag{5.1}$$

Element 2

$$\frac{E_2 \cdot I_2}{L_2^3} = \frac{210000000000 \cdot 4 \cdot 10^{-6}}{0.5^3} = 6720000 \tag{5.2}$$

Element 3

$$\frac{E_3 \cdot I_3}{L_3^3} = \frac{210000000000 \cdot 4 \cdot 10^{-6}}{2^3} = 105000 \tag{5.3}$$

LOKALNE MATRICE KRUTOSTI ELEMENTA

Na osnovu riješenih konstanti krutosti računaju se lokalne matrice krutosti. Definirana lokalna matrica krutosti preuzeta je iz skripte [2]. Množenjem konstantne krutosti s lokanom matricom krutosti dobiva se potpuni izgled matrice krutosti za jedan element.

Element 1

$$[k_1] = \frac{E_1 \cdot I_1}{L_1^3} \cdot \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{5.4}$$

$$[k_1] = 840000 \begin{bmatrix} 12 & 6 \cdot 1 & -12 & 6 \cdot 1 \\ 6 \cdot 1 & 4 \cdot 1^2 & -6 \cdot 1 & 2 \cdot 1^2 \\ -12 & -6 \cdot 1 & 12 & -6 \cdot 1 \\ 6 \cdot 1 & 2 \cdot 1^2 & -6 \cdot 1 & 4 \cdot 1^2 \end{bmatrix} \tag{5.5}$$

$$[k_1] = \begin{bmatrix} 10080000 & 5040000 & -10080000 & 5040000 \\ 5040000 & 3360000 & -5040000 & 1680000 \\ -1008000 & -5040000 & 10080000 & -5040000 \\ 5040000 & 1680000 & -5040000 & 3360000 \end{bmatrix} \tag{5.6}$$

Element 2

$$[k_2] = \frac{E_2 \cdot I_2}{L_2^3} \cdot \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{5.7}$$

$$6720000 \cdot \begin{bmatrix} 12 & 6 \cdot 0{,}5 & -12 & 6 \cdot 0{,}5 \\ 6 \cdot 0{,}5 & 4 \cdot 0.5^2 & -6 \cdot 0.5 & 2 \cdot 0.5^2 \\ -12 & -6 \cdot 0.5 & 12 & -6 \cdot 0.5 \\ 6 \cdot 0.5 & 2 \cdot 0.5^2 & -6 \cdot 0.5 & 4 \cdot 0.5^2 \end{bmatrix} \tag{5.8}$$

$$[k_2] = \begin{bmatrix} 80640000 & 20160000 & -80640000 & 20160000 \\ 20160000 & 6720000 & -20160000 & 3360000 \\ -80640000 & -20160000 & 80640000 & -20160000 \\ 20160000 & 3360000 & -20160000 & 6720000 \end{bmatrix} \tag{5.9}$$

Element 3

$$[k_3] = \frac{E_3 \cdot I_3}{L_3^3} \cdot \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{5.10}$$

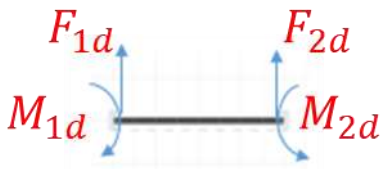
$$105000 \cdot \begin{bmatrix} 12 & 6\cdot 2 & -12 & 6\cdot 2 \\ 6\cdot 2 & 4\cdot 2^2 & -6\cdot 2 & 2\cdot 2^2 \\ -12 & -6\cdot 2 & 12 & -6\cdot 2 \\ 6\cdot 2 & 2\cdot 2^2 & -6\cdot 2 & 4\cdot 2^2 \end{bmatrix} \tag{5.11}$$

$$[k_3] = \begin{bmatrix} 1260000 & 1260000 & -1260000 & 1260000 \\ 1260000 & 168000 & -1260000 & 84000 \\ -1260000 & -126000 & 1260000 & -1260000 \\ 1260000 & 840000 & -1260000 & 1680000 \end{bmatrix} \tag{5.12}$$

KONTINUIRANO OPTEREĆENJE NA ELEMENTIMA

Element 1 – Kontinuirano opterećenje $q = 0$

Na slici 5.3 prikazan je konačni element broj 1 te izračun ekvivalentnih opterećenja u čvoru 1 i 2.



Slika 5.3 – Primjer 1, element 1

$$F_{1d} = \emptyset \tag{5.13}$$

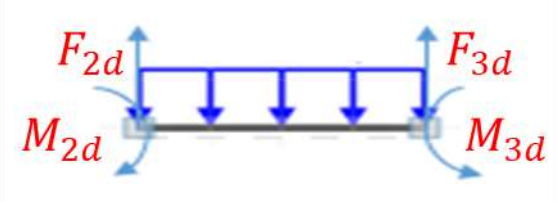
$$M_{1d} = \emptyset \tag{5.14}$$

$$F_{2d} = \emptyset \tag{5.15}$$

$$M_{2d} = \emptyset \tag{5.16}$$

Element 2 – Kontinuirano opterećenje $q = 10\,{}^{\mathrm{kN}}/_{\mathrm{m}}$

Na slici 5.4 prikazan je konačni element broj 2, te izračun ekvivalentnih opterećenja u čvoru 2 i 3.



Slika 5.4 – Primjer 1, element 2

$$F_{2d_{(2)}} = \frac{F * L}{2} = \frac{10000 * 0.5}{2} = \mathbf{2500\ N} \tag{5.17}$$

$$F_{2d_{(2)}} = \frac{-F * L^2}{12} = \frac{10000 * 0.5^2}{2} = \mathbf{-208.33\ Nm} \tag{5.18}$$

$$F_{3d_{(1)}} = \frac{F * L}{2} = \frac{10000 * 0.5}{2} = \mathbf{2500\ N} \tag{5.19}$$

$$M_{3d_{(1)}} = \frac{F * L^2}{12} = \frac{10000 * 0.5^2}{2} = \mathbf{208.33\ Nm} \tag{5.20}$$

Element 3 – Kontinuirano opterećenje $q = 0$

Na slici 5.5 prikazan je konačni element broj 3, te izračun ekvivalentnih opterećenja u čvoru 3 i 4.

Slika 5.5 – Primjer 1, element 3

$$F_{3d} = \emptyset \tag{5.21}$$

$$M_{3d} = \emptyset \tag{5.22}$$

$$F_{4d} = \emptyset \tag{5.23}$$

$$M_{4d} = \emptyset \tag{5.24}$$

GLOBALNA MATRICA KRUTOSTI

Globalna matrica dobiva se formatiranjem (spajanjem) lokalnih matrica krutosti u formatu globalne matrice.

$$[K_G] = [k_1] + [k_2] + [k_3] \tag{5.25}$$

$$[K_G] = 10^7 \begin{bmatrix} 1.008 & 0.504 & -1.008 & 0.504 & 0 & 0 & 0 & 0 \\ -0.504 & 0.504 & -0.504 & 0.168 & 0 & 0 & 0 & 0 \\ -1.008 & -0.504 & 1.008 + 8.064 & -0.504 + 2.016 & -8.064 & 2.016 & 0 & 0 \\ 0.504 & 0.168 & -0.504 + 2.016 & 0.336 + 0.672 & -2.016 & 0.336 & 0 & 0 \\ 0 & 0 & -8.06 & -0.2016 & 8.064 + 0.126 & -2.016 + 0.126 & -0.126 & 0.126 \\ 0 & 0 & 2.016 & 0.336 & -2.016 + 0.126 & 0.168 + 0.672 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & -0.126 & -0.126 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & 0.126 & 0.084 & -0.126 & 0.168 \end{bmatrix} \tag{5.26}$$

$$[K_G] = 10^7 \begin{bmatrix} 1.008 & 0.504 & -1.008 & 0.504 & 0 & 0 & 0 & 0 \\ -0.504 & 0.504 & -0.504 & 0.168 & 0 & 0 & 0 & 0 \\ -1.008 & -0.504 & 9.072 & 1.512 & -8.064 & 2.016 & 0 & 0 \\ 0.504 & 0.168 & 0.1512 & 0.1008 & -2.016 & 0.336 & 0 & 0 \\ 0 & 0 & -8.06 & -0.2016 & 0.336 & -0.126 & -0.126 & 0.126 \\ 0 & 0 & 2.016 & 0.336 & -1.890 & -0.126 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & -0.126 & -0.126 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & 0.126 & 0.084 & -0.126 & 0.168 \end{bmatrix} \tag{5.27}$$

MATRICA SILA

Matrica sila kreira se zbrajanjem vektora sila s ekvivalentnim opterećenjima.

$$\{F\} = \{f\} \mp \{F_d\} = \begin{Bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \\ F_3 \\ M_3 \\ F_4 \\ M_4 \end{Bmatrix} + \begin{Bmatrix} F_{1d} \\ M_{1d} \\ F_{2d_1} + F_{2d_2} \\ M_{2d_1} + M_{2d_2} \\ F_{3d_1} + F_{3d_2} \\ M_{3d_1} + M_{3d_2} \\ F_{4d} \\ F_{4d} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 5000 \\ 0 \end{Bmatrix} + \begin{Bmatrix} 0 \\ 0 \\ 0 - 2500 \\ 0 - 208.33 \\ -2500 + 0 \\ 208.33 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ -2500 \\ -208.33 \\ -2500 \\ 208.33 \\ 5000 \\ 0 \end{Bmatrix} \tag{5.28}$$

OSNOVNA MATRIČNA JEDNADŽBA

$$\{u\} = [K]^{-1} \cdot \{F\} \tag{5.29}$$

Obzirom da je globalna matrica krutosti simetrična matrica vrijedi da je $[K]^{-1} = [K]$.

$$\begin{Bmatrix} u_1 = 0 \\ \vartheta_1 = 0 \\ u_2 \\ \vartheta_2 \\ u_3 = 0 \\ \vartheta_3 \\ u_4 \\ \vartheta_4 \end{Bmatrix} = \left(10^7 \cdot \begin{bmatrix} 1.008 & 0.504 & -1.008 & 0.504 & 0 & 0 & 0 & 0 \\ -0.504 & 0.504 & -0.504 & 0.168 & 0 & 0 & 0 & 0 \\ -1.008 & -0.504 & 9.072 & 1.512 & -8.064 & 2.016 & 0 & 0 \\ 0.504 & 0.168 & 0.1512 & 0.1008 & -2.016 & 0.336 & 0 & 0 \\ 0 & 0 & -8.06 & -0.2016 & 0.819 & -0.189 & -0.126 & 0.126 \\ 0 & 0 & 2.016 & 0.336 & -0.189 & -0.189 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & -0.126 & -0.126 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & 0.126 & 0.084 & -0.126 & 0.168 \end{bmatrix} \right)^{-1} \cdot \begin{Bmatrix} 0 \\ 0 \\ -2500 \\ -208.33 \\ -2500 \\ 208.33 \\ 5000 \\ 0 \end{Bmatrix} \quad (5.30)$$

Eliminacijom redaka i stupaca za koje je pomak jednak nuli dobije se reducirana jednadžba sustava.

REDUCIRANA JEDNADŽBA SUSTAVA

$$\begin{Bmatrix} u_2 \\ \vartheta_2 \\ \vartheta_3 \\ u_4 \\ \vartheta_4 \end{Bmatrix} = \begin{bmatrix} 90720000 & 15120000 & 20160000 & 0 & 0 \\ 15120000 & 10080000 & 3360000 & 0 & 0 \\ 20160000 & 3360000 & 840000 & -1260000 & 84000 \\ 0 & 0 & -1260000 & 1260000 & -1260000 \\ 0 & 0 & 840000 & -1260000 & 1680000 \end{bmatrix} \begin{Bmatrix} -2500 \\ -208.33 \\ 208.33 \\ 5000 \\ 0 \end{Bmatrix} \quad (5.31)$$

Raspisivanjem jednadžbe sustava dobiva se:

$$\begin{gathered} 90720000 \cdot u_2 + 15120000 \cdot \vartheta_2 + 20160000 \cdot \vartheta_3 + 0 \cdot u_4 + 0 \cdot \vartheta_4 = -2500 \\ 15120000 \cdot u_2 + 10080000 \cdot \vartheta_2 + 3360000 \cdot \vartheta_3 + 0 \cdot u_4 + 0 \cdot \vartheta_4 = -208.33 \\ 20160000 \cdot u_2 + 3360000 \cdot \vartheta_2 + 840000 \cdot \vartheta_3 + -1260000 \cdot u_4 + 84000 \cdot \vartheta_4 = -208.33 \\ 0 \cdot u_2 + 0 \cdot \vartheta_2 \pm 1260000 \cdot \vartheta_3 + 1260000 \cdot u_4 - 1260000 \cdot \vartheta_4 = 208.33 \\ 0 \cdot u_2 + 0 \cdot \vartheta_2 + 840000 \cdot \vartheta_3 + -1260000 \cdot u_4 + 1680000 \cdot \vartheta_4 = -2500 \end{gathered} \quad (5.32)$$

Za rješavanje jednadžbi s više nepoznanica korišten je „Online Solver [7]“

$$u_2 = -0.00110\ m$$

$$\vartheta_2 = 0.00003\ rad$$

$$\vartheta_3 = 0.00481\ rad$$

$$u_4 = 0.02548 \text{ m}$$

$$\vartheta_4 = 0.01671 \text{ rad}$$

REAKCIJE U OSLONCIMA

Reakcije u osloncima (sile i momenti) određuju se na temelju jednadžbe:

$$\{R\} = [K]\,X\,\{u\} - \{f\} \tag{5.33}$$

$$\begin{Bmatrix} F_{R1} \\ M_{R1} \\ F_{R2} \\ M_{R2} \\ F_{R3} \\ M_{R3} \\ F_{R4} \\ M_{R4} \end{Bmatrix} = 10^7 \begin{bmatrix} 1.008 & 0.504 & -1.008 & 0.504 & 0 & 0 & 0 & 0 \\ -0.504 & 0.504 & -0.504 & 0.168 & 0 & 0 & 0 & 0 \\ -1.008 & -0.504 & 9.072 & 1.512 & -8.064 & 2.016 & 0 & 0 \\ 0.504 & 0.168 & 0.1512 & 0.1008 & -2.016 & 0.336 & 0 & 0 \\ 0 & 0 & -8.06 & -0.2016 & 0.819 & -0.189 & -0.126 & 0.126 \\ 0 & 0 & 2.016 & 0.336 & -0.189 & -0.189 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & -0.126 & -0.126 & -0.126 & 0.126 \\ 0 & 0 & 0 & 0 & 0.126 & 0.084 & -0.126 & 0.168 \end{bmatrix} \begin{Bmatrix} u_1 = 0 \\ \vartheta_1 = 0 \\ -0.00110 \\ 0.00003 \\ u_3 = 0 \\ 0.00481 \\ 0.02548 \\ 0.01671 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ -2500 \\ -208.33 \\ -2500 \\ 208.33 \\ 5000 \\ 0 \end{Bmatrix} \tag{5.34}$$

$$\begin{aligned} F_{R1} = {} & (-10080000 \cdot u_1 - F_1) + (5040000 \cdot \vartheta_1 - M_1) + (-10080000 \cdot u_2 - F_2) \\ & + (5040000 \cdot \vartheta_2 - M_2) + (0 \cdot u_3 - F_3) + (0 \cdot \vartheta_3 - M_3) \\ & + (0 \cdot u_4 - F_4) + (0 \cdot \vartheta_4 - M_4) \end{aligned} \tag{5.35}$$

$$\begin{aligned} F_{R1} = {} & (-10080000 \cdot 0 - 0) + (5040000 \cdot 0 - 0) + (-10080000 \cdot \\ & -0100110 - -2500) + (5040000 \cdot 0.00003 + 208.33) + (0 \cdot 0 - 2500_3) + \\ & (0 \cdot 0.00481 - 208.33) + (0 \cdot 0.02548 - 5000) + (0 \cdot 0.01671 - 0) \\ & \boldsymbol{F_{R1} = 11229.12\ N} \end{aligned} \tag{5.36}$$

$$\begin{aligned} M_{R1} = {} & (-5040000 \cdot u_1 - F_1) + (5040000 \cdot \vartheta_1 - M_1) + (-5040000 \cdot u_2 - F_2) + \\ & + (1680000 \cdot \vartheta_2 - M_2) + (0 \cdot u_3 - F_3) + (0 \cdot \vartheta_3 - M_3) + (0 \cdot u_4 - F_4) \\ & + (0 \cdot \vartheta_4 - M_4) \end{aligned} \tag{5.37}$$

$$\begin{aligned} M_{R1} = {} & (-5040000 \cdot 0 - 0) + (5040000 \cdot 0 - 0) \\ & + (-5040000 \cdot (-0100110) + 2500) \\ & + (1680000 \cdot 0.00003 + 208.33)(0 \cdot 0 + 2500) \\ & + (0 \cdot 0.00481 - 208.33) + (0 \cdot 0.02548 - 5000) \\ & + (0 \cdot 0.01671 - 0) \\ & \boldsymbol{M_{R1} = 5591.04\ Nm} \end{aligned} \tag{5.38}$$

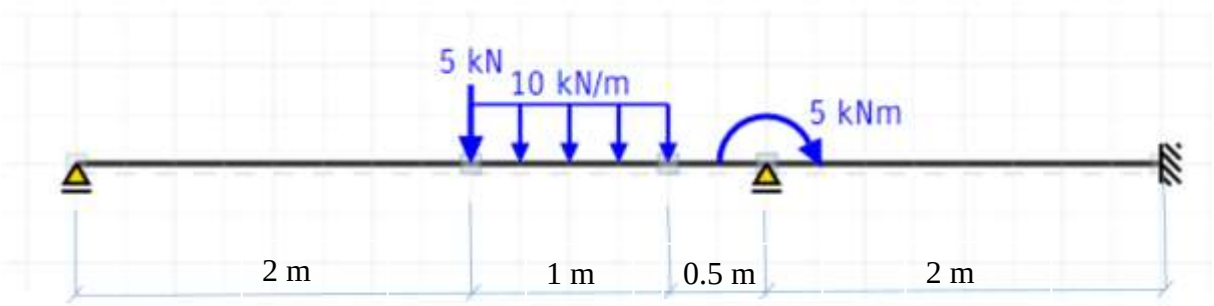
$$F_{R3} = (-0 \cdot u_1 - F_1) + (0 \cdot \vartheta_1 - M_1) + (-80600000 \cdot u_2 - F_2) + (2016000 \cdot \vartheta_2 - M_2) + (8190000 \cdot u_3 - F_3) + (1890000 \cdot \vartheta_3 - M_3) + (1260000 \cdot u_4 - F_4) + (1260000 \cdot \vartheta_4 - M_4) \quad (5.39)$$

$$F_{R3} = (-0 \cdot 0 - 0) + (0 \cdot 0 - 0) + (-80600000 \cdot (-0100110) + 2500) + (2016000 \cdot 0.00003 + 208.33) + (8190000 \cdot 0 - 2500) + (1890000 \cdot 0.00481 - 208.33) + (1260000 \cdot 0.02548 - 5000) + (1260000 \cdot 0.01671 - 00) \quad (5.40)$$

$$\boldsymbol{F_{R3} = -11229.12\ \mathrm{N}}$$

5.2. PRIMJER 2

Na slici 5.6 prikazana je jednostavno poprečno opterećena greda s jednom koncentriranom silom, zakretnim momentom, te kontinuiranim opterećenjem. Na temelju zadanih podataka metodom konačnih elemenata treba odrediti sve pomake u čvorovima i reakcije u osloncima (u=?, $\theta = ?, [R] = ?$). ($R_{1,3,5..}$-predstavlja reakcijske sile, $R_{2,4,6..}$- predstavlja reakcijske momente)!



Slika 5.6 – Primjer br. 2 [6]

Zadano je:

F = 5000 N

M = 5000 Nm

Q = 10000 N/m

E = 210 GPa

$I = 4 * 10^{-6}$ m^4

$F_1 = \emptyset$ $M_1 = \emptyset$

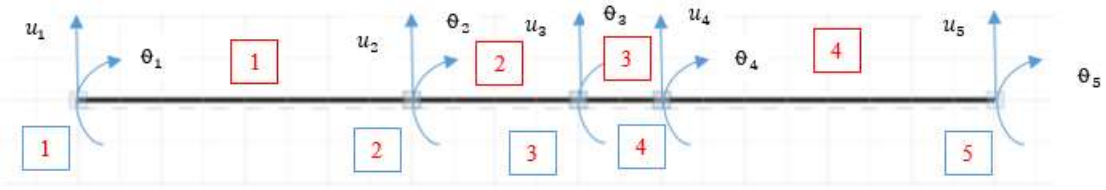
$F_2 = -5000$ N $M_2 = \emptyset$

$F_3 = \emptyset$ $M_3 = \emptyset$

$F_4 = \emptyset$ $M_4 = -5000$ Nm

$F_5 = \emptyset$ $M_5 = \emptyset$

Na slici 5.7 prikazana je raspodjela grede na elemente, čvorove i pomake koji djeluju na čvor. Greda se dijeli na ukupno 4 elementa i 5 čvorova.



Slika 5.7 – Greda s označenim elementima, čvorovima i pomacima

$u_1 = \emptyset$ $u_2 = ?$ $u_3 = ?$ $u_4 = \emptyset$ $u_5 = \emptyset$

$\Theta_1 = ?$ $\Theta_2 = ?$ $\Theta_3 = ?$ $\Theta_4 = ?$ $\Theta_5 = \emptyset$

KONSTANTE KRUTOSTI ZA SVE ELEMENTE

Element 1

$$\frac{E_1 \cdot I_1}{L_1^3} = \frac{210000000000 \cdot 6 \cdot 10^{-6}}{2^3} = 157500 \tag{5.41}$$

Element 2

$$\frac{E_2 \cdot I_2}{L_2^3} = \frac{210000000000 \cdot 6 \cdot 10^{-6}}{1^3} = 1260000 \tag{5.42}$$

Element 3

$$\frac{E_3 \cdot I_3}{L_3^3} = \frac{210000000000 \cdot 6 \cdot 10^{-6}}{0.5^3} = 10080000 \tag{5.43}$$

Element 4

$$\frac{E_4 \cdot I_4}{L_4^3} = \frac{210000000000 \cdot 6 \cdot 10^{-6}}{2^3} = 157500 \tag{5.44}$$

Na osnovu riješenih konstanti krutosti računaju se lokalne matrice krutosti. Definirana lokalna matrica krutosti preuzeta je iz skripte [2]. Množenjem konstantne krutosti s lokanom matricom krutosti dobiva se potpuni izgled matrice krutosti za jedan element.

LOKALNE MATRICE KRUTOSTI ELEMENTA

Element 1

$$[k_1] = \frac{E_1 \cdot I_1}{L_1^3} \cdot \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{5.45}$$

$$[k_1] = 157500 \cdot \begin{bmatrix} 12 & 6 \cdot 2 & -12 & 6 \cdot 2 \\ 6 \cdot 2 & 4 \cdot 2^2 & -6 \cdot 2 & 2 \cdot 2^2 \\ -12 & -6 \cdot 2 & 12 & -6 \cdot 2 \\ 6 \cdot 2 & 2 \cdot 2^2 & -6 \cdot 2 & 4 \cdot 2^2 \end{bmatrix} \tag{5.46}$$

$$[k_1] = \begin{bmatrix} 1890000 & 1890000 & -1890000 & 1890000 \\ 1890000 & 2520000 & -1890000 & 1260000 \\ -1890000 & -1890000 & 1890000 & -1890000 \\ 1890000 & 1260000 & -1890000 & 2520000 \end{bmatrix} \tag{5.47}$$

Element 2

$$[k_2] = \frac{E_2 \cdot I_2}{L_2^3} \cdot \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{5.48}$$

$$[k_2] = 1260000 \cdot \begin{bmatrix} 12 & 6 \cdot 1 & -12 & 6 \cdot 1 \\ 6 \cdot 1 & 4 \cdot 1^2 & -6 \cdot 1 & 2 \cdot 1^2 \\ -12 & -6 \cdot 1 & 12 & -6 \cdot 1 \\ 6 \cdot 1 & 2 \cdot 1^2 & -6 \cdot 1 & 4 \cdot 1^2 \end{bmatrix} \tag{5.49}$$

$$[k_2] = \begin{bmatrix} 15120000 & 7560000 & -15120000 & 7560000 \\ 7560000 & 5040000 & -7560000 & 2520000 \\ -15120000 & -7560000 & 15120000 & -7560000 \\ 7560000 & 2520000 & -7560000 & 5040000 \end{bmatrix} \tag{5.50}$$

Element 3

$$[k_3] = \frac{E_3 \cdot I_3}{L_3^3} \cdot \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{5.51}$$

$$[k_3] = 10080000 \cdot \begin{bmatrix} 12 & 6 \cdot 0{,}5 & -12 & 6 \cdot 0{,}5 \\ 6 \cdot 0{,}5 & 4 \cdot 0.5^2 & -6 \cdot 0.5 & 2 \cdot 0.5^2 \\ -12 & -6 \cdot 0.5 & 12 & -6 \cdot 0.5 \\ 6 \cdot 0.5 & 2 \cdot 0.5^2 & -6 \cdot 0.5 & 4 \cdot 0.5^2 \end{bmatrix} \tag{5.52}$$

$$[k_3] = \begin{bmatrix} 120960000 & 30240000 & -120960000 & 30240000 \\ 30240000 & 10080000 & -30240000 & 5040000 \\ -120960000 & -30240000 & 120960000 & -30240000 \\ 30240000 & 5040000 & -30240000 & 10080000 \end{bmatrix} \tag{5.53}$$

Element 4

$$[k_4] = \frac{E_4 \cdot I_4}{L_4^3} \cdot \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L^2 & -6L & 4L^2 \end{bmatrix} \tag{5.54}$$

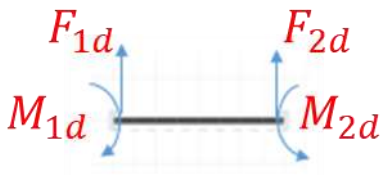
$$[k_4] = 157500 \begin{bmatrix} 12 & 6 \cdot 2 & -12 & 6 \cdot 2 \\ 6 \cdot 2 & 4 \cdot 2^2 & -6 \cdot 2 & 2 \cdot 2^2 \\ -12 & -6 \cdot 2 & 12 & -6 \cdot 2 \\ 6 \cdot 2 & 2 \cdot 2^2 & -6 \cdot 2 & 4 \cdot 2^2 \end{bmatrix} \tag{5.55}$$

$$[k_4] = \begin{bmatrix} 1890000 & 1890000 & -1890000 & 1890000 \\ 1890000 & 2520000 & -1890000 & 1260000 \\ -1890000 & -1890000 & 1890000 & -1890000 \\ 1890000 & 1260000 & -1890000 & 2520000 \end{bmatrix} \tag{5.56}$$

KONTINUIRANO OPTEREĆENJE NA ELEMENTIMA

Element 1 – Kontinuirano opterećenje $q = 0$

Na slici 5.8 prikazan je konačni element broj 1, te izračun ekvivalentnih opterećenja u čvoru 1 i 2.



Slika 5.8 – Primjer 2, element 1

$$F_{1d} = \emptyset \tag{5.57}$$

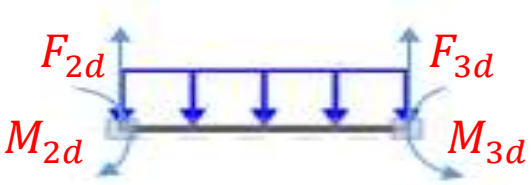
$$M_{1d} = \emptyset \tag{5.58}$$

$$F_{2d} = \emptyset \tag{5.59}$$

$$M_{2d} = \emptyset \tag{5.60}$$

Element 2 – Kontinuirano opterećenje $q = 10\ {}^{\mathrm{kN}}/_{\mathrm{m}}$

Na slici 5.9 prikazan je konačni element broj 2 te izračun ekvivalentnih opterećenja u čvoru 2 i 3.



Slika 5.9 – Primjer 2, element 2

$$F_{2d(2)} = \frac{F * L}{2} = \frac{10000 * 1}{2} = \mathbf{5000\ N} \tag{5.61}$$

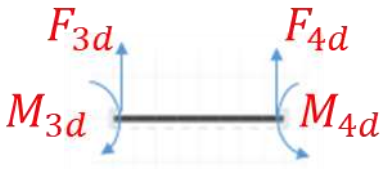
$$M_{2d_{(2)}} = \frac{F*L^2}{12} = \frac{-10000*1^2}{12} = \mathbf{-833.33\ Nm} \tag{5.62}$$

$$F_{3d_{(1)}} = \frac{F*L}{2} = \frac{10000*1}{2} = \mathbf{5000\ N} \tag{5.63}$$

$$M_{3d_{(1)}} = \frac{F*L^2}{12} = \frac{10000*1^2}{12} = \mathbf{833.33\ Nm} \tag{5.64}$$

Element 3 – Kontinuirano opterećenje $q = 0$

Na slici 5.10 prikazan je konačni element broj 3, te izračun ekvivalentnih opterećenja u čvoru 3 i 4.



Slika 5.10 – Primjer 2, element 3

$$F_{3d} = \emptyset \tag{5.65}$$

$$M_{3d} = \emptyset \tag{5.66}$$

$$F_{4d} = \emptyset \tag{5.67}$$

$$M_{4d} = \emptyset \tag{5.68}$$

Element 4 – Kontinuirano opterećenje $q = 0$

Na slici 5.11 prikazan je konačni element broj 4, te izračun ekvivalentnih opterećenja u čvoru 4 i 5.

Slika 5.11 – Primjer 2, element 4

$$F_{4d} = \emptyset \tag{5.69}$$

$$M_{4d} = \emptyset \tag{5.70}$$

$$F_{5d} = \emptyset \tag{5.71}$$

$$M_{5d} = \emptyset \tag{5.72}$$

GLOBALNA MATRICA KRUTOSTI

Globalna matrica dobiva se formatiranjem (spajanjem) lokalnih matrica krutosti u formatu globalne matrice (jednadžba 5.68).

$$[K] = [k_1] + [k_2] + [k_3] + [k_4] \tag{5.73}$$

$$[K] = 10^7 \begin{bmatrix} 0.189 & 0.189 & -0.189 & 0.189 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.189 & 0.252 & -0.189 & 0.126 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.189 & -0.189 & 0.189 + 1.512 & -0.189 + 0.756 & -1.512 & 0.756 & 0 & 0 & 0 & 0 \\ 0.189 & 0.126 & -0.189 + 0.756 & 0.252 + 0.504 & -0.756 & 0.225 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.512 & -0.756 & 1.15122 + 12.09 & -0.756 + 3.024 & -12.096 & 3.024 & 0 & 0 \\ 0 & 0 & 0.756 & 0.225 & -0.756 + 3.024 & 0.504 + 1.008 & -3.024 & 0.504 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.2096 & -3.024 & 12.09 + 0.189 & -3.024 + 0.189 & -0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 3.024 & 0.504 & -3.024 + 0.189 & 1.008 + 0.252 & -0.189 & 0.126 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.189 & -0.189 & 0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.189 & 0.126 & -0.189 & 0.252 \end{bmatrix} \tag{5.74}$$

$$[K] = 10^7 \begin{bmatrix} 0.189 & 0.189 & -0.189 & 0.189 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.189 & 0.252 & -0.189 & 0.126 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.189 & -0.189 & 1.701 & 0.567 & -1.512 & 0.756 & 0 & 0 & 0 & 0 \\ 0.189 & 0.126 & 0.567 & 0.756 & -0.756 & 0.225 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.512 & -0.756 & 13.608 & -2.227 & -12.096 & 3.024 & 0 & 0 \\ 0 & 0 & 0.756 & 0.225 & 2.268 & 1.512 & -3.024 & 0.504 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.2096 & -3.024 & 12.285 & -2.835 & -0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 3.024 & 0.504 & -2.835 & 1.26 & -0.189 & 0.126 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.189 & -0.189 & 0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.189 & 0.126 & -0.189 & 0.252 \end{bmatrix} \quad (5.75)$$

MATRICA SILA

Matrica sila kreira se zbrajanjem vektora sila s ekvivalentnim opterećenjima.

$$\{F\} = \{f\} \mp \{F_d\} = \begin{Bmatrix} F_1 \\ M_1 \\ F_2 \\ M_2 \\ F_3 \\ M_3 \\ F_4 \\ M_4 \\ F_5 \\ M_5 \end{Bmatrix} + \begin{Bmatrix} F_{1d} \\ M_{1d} \\ F_{2d_1} + F_{2d_2} \\ M_{2d_1} + M_{2d_2} \\ F_{3d_1} + F_{3d_2} \\ M_{3d_1} + M_{3d_2} \\ F_{4d} \\ M_{4d} \\ F_{5d} \\ M_{5d} \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ -5000 \\ 0 \\ 0 \\ 0 \\ 0 \\ -5000 \\ 0 \\ 0 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ -5000 - 5000 \\ 0 - 833.33 \\ -5000 + 0 \\ 0 + 833.33 \\ 0 \\ 0 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0 \\ 0 \\ -10000 \\ -833.33 \\ -5000 \\ 833.33 \\ 0 \\ -5000 \\ 0 \\ 0 \end{Bmatrix} \quad (5.76)$$

OSNOVNA MATRIČNA JEDNADŽBA

$$\{u\} = [K]^{-1} \cdot \{F\} \quad (5.77)$$

Obzirom da je globalna matrica krutosti simetrična matrica vrijedi da je $[K]^{-1} = [K]$.

$$\begin{Bmatrix} u_1 = 0 \\ \theta_1 \\ u_2 \\ \theta_2 \\ u_3 \\ \theta_3 \\ u_4 = 0 \\ \theta_4 \\ u_5 = 0 \\ \theta_5 = 0 \end{Bmatrix} = \left(10^7 \begin{bmatrix} 0.189 & 0.189 & -0.189 & 0.189 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.189 & 0.252 & -0.189 & 0.126 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.189 & -0.189 & 1.701 & 0.567 & -1.512 & 0.756 & 0 & 0 & 0 & 0 \\ 0.189 & 0.126 & 0.567 & 0.756 & -0.756 & 0.225 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.512 & -0.756 & 13.608 & -2.227 & -12.096 & 3.024 & 0 & 0 \\ 0 & 0 & 0.756 & 0.225 & 2.268 & 1.512 & -3.024 & 0.504 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.2096 & -3.024 & 12.285 & -2.835 & -0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 3.024 & 0.504 & -2.835 & 1.26 & -0.189 & 0.126 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.189 & -0.189 & 0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.189 & 0.126 & -0.189 & 0.252 \end{bmatrix} \right)^{-1} \cdot \begin{Bmatrix} 0 \\ 0 \\ -10000 \\ -833.33 \\ -5000 \\ 833.33 \\ 0 \\ -5000 \\ 0 \\ 0 \end{Bmatrix} \quad (5.78)$$

Eliminacijom redaka i stupaca za koje je pomak jednak nuli dobije se reducirana jednadžba sustava.

REDUCIRANA JEDNADŽBA SUSTAVA

$$\begin{Bmatrix} \vartheta_1 \\ u_2 \\ \vartheta_2 \\ u_3 \\ \vartheta_3 \\ \vartheta_5 \end{Bmatrix} = \begin{bmatrix} 2520000 & -1890000 & 1260000 & 0 & 0 & 0 \\ -1890000 & 17010000 & 5670000 & -15120000 & 7560000 & 0 \\ 1260000 & 5670000 & 7560000 & -7560000 & 2520000 & 0 \\ 0 & -15120000 & -7560000 & 136080000 & 22680000 & 30240000 \\ 0 & 7560000 & 2520000 & 22680000 & 15120000 & 5040000 \\ 0 & 0 & 0 & 30240000 & 5040000 & 1260000 \end{bmatrix} \cdot \begin{Bmatrix} 0 \\ -10000 \\ -833.33 \\ -5000 \\ 833.33 \\ -5000 \end{Bmatrix} \quad (5.79)$$

$$\begin{gathered} 2520000 \cdot \vartheta_1 - 1890000 \cdot u_2 + 1260000 \cdot \vartheta_2 + 0 \cdot u_3 + 0 \cdot \vartheta_3 + 0 \cdot \vartheta_3 = 0 \\ -1890000 \cdot \vartheta_1 + 17010000 \cdot u_2 + 5670000 \cdot \vartheta_2 - 15120000 \cdot u_3 + 7560000 \cdot \vartheta_3 + 0 \cdot \vartheta_3 = -10000 \\ 1260000 \cdot \vartheta_1 + 5670000 \cdot u_2 + 7560000 \cdot \vartheta_2 - 7560000 \cdot u_3 + 2520000 \cdot \vartheta_3 + 0 \cdot \vartheta_3 = -833.33 \\ 0 \cdot \vartheta_1 - 15120000 \cdot u_2 - 7560000 \cdot \vartheta_2 + 136080000 \cdot u_3 + 22680000 \cdot \vartheta_3 + 30240000 \cdot \vartheta_3 = -5000 \\ 0 \cdot \vartheta_1 + 7560000 \cdot u_2 + 2520000 \cdot \vartheta_2 - 22680000 \cdot u_3 + 15120000 \cdot \vartheta_3 + 5040000 \cdot \vartheta_3 = 833.33 \\ 0 \cdot \vartheta_1 + 0 \cdot u_2 + 0 \cdot \vartheta_2 - 30240000 \cdot u_3 + 5040000 \cdot \vartheta_3 + 1260000 \cdot \vartheta_3 = -5000 \end{gathered} \quad (5.80)$$

Za rješavanje jednadžbi s više nepoznanica korišten je „Online Solver [7]“

$$\vartheta_1 = -0.00330\ rad$$

$$u_2 = 0.00373\ m$$

$$\vartheta_2 = 0.00102\ rad$$

$$u_3 = 0.00118 \text{ m}$$

$$\vartheta_3 = 0.00312 \text{ rad}$$

$$\vartheta_4 = 0.00118 \text{ rad}$$

REAKCIJE U OSLONCIMA

Reakcije u osloncima (sile i momenti) određuju se na temelju jednadžbe:

$$\{R\} = [K] \cdot \{u\} - \{f\} \quad (5.81)$$

$$\begin{Bmatrix} F_{R1} \\ M_{R1} \\ F_{R2} \\ M_{R2} \\ F_{R3} \\ M_{R3} \\ F_{R4} \\ M_{R4} \\ F_{R5} \\ M_{R5} \end{Bmatrix} = 10^7 \begin{bmatrix} 0.189 & 0.189 & -0.189 & 0.189 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.189 & 0.252 & -0.189 & 0.126 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.189 & -0.189 & 1.701 & 0.567 & -1.512 & 0.756 & 0 & 0 & 0 & 0 \\ 0.189 & 0.126 & 0.567 & 0.756 & -0.756 & 0.225 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1.512 & -0.756 & 13.608 & -2.227 & -12.096 & 3.024 & 0 & 0 \\ 0 & 0 & 0.756 & 0.225 & 2.268 & 1.512 & -3.024 & 0.504 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1.2096 & -3.024 & 12.285 & -2.835 & -0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 3.024 & 0.504 & -2.835 & 1.26 & -0.189 & 0.126 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.189 & -0.189 & 0.189 & -0.189 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.189 & 0.126 & -0.189 & 0.252 \end{bmatrix} \cdot \begin{Bmatrix} 0 \\ -0.00330 \\ 0.00373 \\ 0.00102 \\ 0.00118 \\ 0.00312 \\ 0 \\ 0.00118 \\ 0 \\ 0 \end{Bmatrix} - \begin{Bmatrix} 0 \\ 0 \\ -10000 \\ -833.33 \\ -5000 \\ 833.33 \\ 0 \\ -5000 \\ 0 \\ 0 \end{Bmatrix} \tag{5.82}$$

$$\begin{aligned} F_{R1} = {} & (1890000 \cdot u_1 - F_1) + (-180000 \cdot \vartheta_1 - M_1) + (-1890000 \cdot u_2 - F_2) \\ & + (1890000 \cdot \vartheta_2 - M_2) + (0 \cdot u_3 - F_3) + (0 \cdot \vartheta_3 - M_3) \\ & + (0 \cdot u_4 - F_4) + (0 \cdot \vartheta_4 - M_4) + (0 \cdot u_5 - F_5) + (0 \cdot \vartheta_5 - M_5) \end{aligned} \tag{5.83}$$

$$\begin{aligned} F_{R1} = {} & (1890000 \cdot 0 - 0) + \big(-180000 \cdot 0 - (-0.0033)\big) \\ & + \big(-1890000 \cdot 0.00373 - (-10000)\big) \\ & + \big(1890000 \cdot 0.00102 - (-833.33)\big) + \big(0 \cdot 0.00118 - (-5000)\big) \\ & + (0 \cdot 0.00312 + 833.33) + (0 \cdot 0 + 0) + (0 \cdot 0.00118 - (-5000) \\ & + (0 \cdot 0 - 0) + (0 \cdot 0 - 0) \end{aligned} \tag{5.84}$$

$$\boldsymbol{F_{R1} = 2740\ \mathrm{N}}$$

$$\begin{aligned} F_{R4} = {} & (0 \cdot u_1 - F_1) + (0 \cdot \vartheta_1 - M_1) + (0 \cdot u_2 - F_2) + (0 \cdot \vartheta_2 - M_2) \\ & + (-12096000 \cdot u_3 - F_3) + (-30240000 \cdot \vartheta_3 - M_3) \\ & + (1222850000 \cdot u_4 - F_4) + (-28350000 \cdot \vartheta_4 - M_4) \\ & + (-1890000 \cdot \vartheta_4 - F_5) + (-1890000 \cdot \vartheta_5 - M_5) \end{aligned} \tag{5.85}$$

$$\begin{aligned} F_{R4} = {} & (0 \cdot 0 - 0) + (0 \cdot 0 - (-0.0033)) + (0 \cdot 0.00373 - (-10000)) \\ & + (0 \cdot 0.00102 - (-833.33)) + (-12096000 \cdot 0.00118 - (-5000)) \\ & + (-30240000 \cdot 0.00312 + 833.33) + (1222850000 \cdot 0 - 0) \\ & + (-28350000 \cdot 0.00118 - (-5000)) + (-1890000 \cdot 0 - 0) \\ & + (-1890000 \cdot 0 - 0) \end{aligned} \tag{5.86}$$

$$\boldsymbol{F_{R4} = 14931\ \mathrm{N}}$$

$$\begin{aligned} F_{R5} = {} & (0 \cdot u_1 - F_1) + (0 \cdot \vartheta_1 - M_1) + (0 \cdot u_2 - F_2) + (0 \cdot \vartheta_2 - M_2) + (0 \cdot u_3 - F_3) \\ & + (0 \cdot \vartheta_3 - M_3) + (-1890000 \cdot u_4 - F_4) + (-1890000 \cdot \vartheta_4 - M_4) \\ & + (1890000 \cdot \vartheta_4 - F_5) + (-1890000 \cdot \vartheta_5 - M_5) \end{aligned} \tag{5.87}$$

$$\begin{aligned} F_{R5} = {} & (0 \cdot 0 - 0) + (0 \cdot 0 - (-0.0033)) + (0 \cdot 0.00373 - (-10000)) \\ & + (0 \cdot 0.00102 - (-833.33)) + (0 \cdot\cdot\, 0.00118 - (-5000)) \\ & + \ (0 \cdot 0.00312 + 833.33) + (-1890000 \cdot 0 - 0) \\ & + \ (-1890000 \cdot 0.00118 - (-5000)) + \ (1890000 \cdot 0 - 0) \\ & + (-1890000 \cdot 0 - 0) \end{aligned} \tag{5.88}$$

$$\boldsymbol{F_{R5} = \ -2230.2\ \mathrm{N}}$$

$$\begin{aligned} M_{R5} = {} & (0 \cdot u_1 - F_1) + (0 \cdot \vartheta_1 - M_1) + (0 \cdot u_2 - F_2) + (0 \cdot \vartheta_2 - M_2) + (0 \cdot u_3 - F_3) \\ & + \ (0 \cdot \vartheta_3 - M_3) + (1890000 \cdot u_4 - F_4) + \ (1260000 \cdot \vartheta_4 - M_4) \\ & + \ (-1890000 \cdot \vartheta_4 - F_5) + (2520000 \cdot \vartheta_5 - M_5) \end{aligned} \tag{5.89}$$

$$\begin{aligned} M_{R5} = {} & (0 \cdot u_1 - F_1) + (0 \cdot 0 - (-0.0033)) + (0 \cdot 0.00373 - (-10000)) \\ & + (0 \cdot 0.00102 - (-833.33)) + (0 \cdot 0.00118 - (-5000)) \\ & + \ (0 \cdot 0.00312 + 833.33) + (1890000 \cdot 0 - 0) \\ & + \ (1260000 \cdot 0.00118 - (-5000)) + \ (-1890000 \cdot 0 - 0) \\ & + (2520000 \cdot 0 - 0) \end{aligned} \tag{5.90}$$

$$\boldsymbol{M_{R5} = 1486.8\ \mathrm{Nm}}$$

5.3. Rješenje primjera 1 primjenom Python programa

U poglavlju 5.3 i 5.4 prikazana je opcija unosa podataka u programski kod. Unos podataka se vrši pritiskom broja 1. Nakon unosa program započinje s radom, te se otvara mogućnost unosa podataka. Na slici 5.12 prikazan je unos podataka za element (početni i krajnji čvor elementa, dužina elementa, aksijalni moment inercije i Young-ov modul elastičnosti).

```
Unos podataka(1) ili default podatci(2): 1
Ukupan broj elemenata je: 3
Element 1 počinje na čvoru : 1
Element 1 završava na čvoru : 2
Duzina elementa 1 u [m] je: 1
Aksijalni moment inercije [m4] je: 0.000004
Young-ov modul [N/m2] je: 210000000000
Element 2 počinje na čvoru : 2
Element 2 završava na čvoru : 3
Duzina elementa 2 u [m] je: 0.5
Aksijalni moment inercije [m4] je: 0.000004
Young-ov modul [N/m2] je: 210000000000
Element 3 počinje na čvoru : 3
Element 3 završava na čvoru : 4
Duzina elementa 3 u [m] je: 2
Aksijalni moment inercije [m4] je: 0.000004
```

Slika 5.12 – Ulazni podatci za element (primjer 1)

Na slici 5.13 prikazan je unos kontinuiranog opterećenja. U slučaju da postoji kontinuirano opterećenje moguć je unos vrijednosti sile.

```
Da li je na elementu 1 kontinuirano opterećenje (da/ne): ne
Da li je na elementu 2 kontinuirano opterećenje (da/ne): da
Kolika je sila opterećenja [N] na elementu 2: -10000
Da li je na elementu 3 kontinuirano opterećenje (da/ne): ne
```

Slika 5.13 – Unos kontinuiranog opterećenja (primjer 1)

Na slici 5.14 prikazan je unos vrijednosti sila i momenta koji su aktivni na čvorovima.

```
Kolika je sila F1 [N] na čvoru 1: 0

Koliki je moment M1 [Nm] na čvoru 1: 0

Kolika je sila F2 [N] na čvoru 2: 0

Koliki je moment M2 [Nm] na čvoru 2: 0

Kolika je sila F3 [N] na čvoru 3: 0

Koliki je moment M3 [Nm] na čvoru 3: 0

Kolika je sila F4 [Nm] na čvoru 4: 5000

Koliki je moment M4 [Nm] na čvoru 3: 0đ
```

Slika 5.14 – Unos vrijednosti sila i momenata u čvorovima (primjer 1)

Na slici 5.15 prikazan je unos stanja pomaka u čvorovima. U slučaju unosa 1 zakret/pomak je slobodan, a u slučaju unosa 0 zakret/pomak je onemogućen.

```
----------------------------------------------------------------------
 0 predstavlja onemogućen zakret/pomak
 1 predstavlja slobodan zakret/pomak
----------------------------------------------------------------------

Progib  u1 u čvoru 1 je {0 ili 1}: 0

Zakret v1 u čvoru 1 je {0 ili 1}: 0

Progib  u2 u čvoru 2 je {0 ili 1}: 1

Zakret v2 u čvoru 2 je {0 ili 1}: 1

Progib  u3 u čvoru 3 je {0 ili 1}: 0

Zakret v3 u čvoru 3 je {0 ili 1}: 1

Progib  u4 u čvoru 4 je {0 ili 1}: 1

Zakret v4 u čvoru 4 je {0 ili 1}: 1
```

Slika 5.15 – Unos stanja pomaka u čvorovima (primjer 1)

Na slici 5.16 prikazan je ispis globalne matrice sila, te reducirane matrice krutosti i reducirane matrice sila koji su potrebni za izračun traženih vrijednosti.

```
Globalna matrica krutosti=
[[ 10080000,   5040000, -10080000,5040000,     0,  0,   0,      0  ]
 [  5040000   3360000  -5040000   1680000  0,       0,        0,  0]
 [-10080000  -5040000  90720000  15120000 -80640000,20160000, 0,0]
 [5040000, 1680000, 15120000, 10080000, -20160000, 3360000,  0, 0]
 [0, 0, -80640000, -20160000, 81900000,-18900000,-1260000,1260000]
 [0,0,20160000,3360000, -18900000, 8400000,-1260000,       840000]
 [0,    0,    0,   0,   -1260000,   -1260000,   1260000, -1260000]
 [0,    0,   0,   0,   1260000,  840000,   -1260000,     1680000]]
==========================================================================
==========================================================================
Reducirana matrica krutosti=
[[90720000 15120000 20160000        0        0]
 [15120000 10080000  3360000        0        0]
 [20160000  3360000  8400000 -1260000   840000]
 [       0        0 -1260000  1260000 -1260000]
 [       0        0   840000 -1260000  1680000]]
==========================================================================
==========================================================================
 Reducirana matrica sila=
[-2500  -208   208  5000     0]
```

Slika 5.16 – Međurezultati obrade podataka (primjer 1)

Na slici 5.17 prikazan je kraj programa. Na zaslonu se ispisuju ukupni rezultati pomaka, te rezultati reakcija u osloncima.

```
==========================================================================
==========================================================================
Pomaci na čvorovima=
u1=  0.000000 m
u2=  0.000000 rad
u3= -0.001100 m
u4=  0.000028 rad
u5=  0.000000 m
u6=  0.004805 rad
u7=  0.025483 m
u8=  0.016710 rad
==========================================================================
==========================================================================
F1,3,5,7... predstavlja reakcijske Sile
F2,4,6,8... predstavlja reakcijske Momente
==========================================================================
Oznake sila
['F1', 'F2', 'F5']
Reakcije u osloncima=
F1= 11226.814815 N
F2= 5590.222222 Nm
F3=  0.000000 N
F4=  0.333333 Nm
F5= -11226.814815 N
F6= -0.333333 Nm
F7=  0.000000 N
F8=  0.000000 Nm
```

Slika 5.17 – Rezultati dobiveni uporabom programa (primjer 1)

5.4 Rješenje primjera 2 primjenom Python programa

Na slici 5.18 prikazan je unos podataka za element (početni i krajnji čvor elementa, dužina elementa, aksijalni moment inercije i Young-ov modul elastičnosti).

```
Unos podataka(1) ili default podatci(2): 1
Ukupan broj elemenata je: 4
Element 1 počinje na čvoru : 1
Element 1 završava na čvoru : 2
Duzina elementa 1 u [m] je: 2
Aksijalni moment inercije [m4] je: 0.000004
Young-ov modul [N/m2] je: 210000000000
Element 2 počinje na čvoru : 2
Element 2 završava na čvoru : 3
Duzina elementa 2 u [m] je: 1
Aksijalni moment inercije [m4] je: 0.000004
Young-ov modul [N/m2] je: 210000000000
Element 3 počinje na čvoru : 3
Element 3 završava na čvoru : 4
Duzina elementa 3 u [m] je: 0.5
Aksijalni moment inercije [m4] je: 0.000004
Young-ov modul [N/m2] je: 210000000000
Element 4 počinje na rubnoj točki : 4
Element 4 završava na rubnoj točki : 5
Duzina elementa 4 u [m] je: 2
Aksijalni moment inercije [m4] je: 0.000004
Young-ov modul [N/m2] je: 210000000000
```

Slika 5.18 – Ulazni podatci o elementu (primjer 2)

Na slici 5.19 prikazan je unos kontinuiranog opterećenja. U slučaju da postoji kontinuirano opterećenje moguć je unos vrijednosti sile.

```
Da li je na elementu 1 kontinuirano opterećenje (da/ne): ne
Da li je na elementu 2 kontinuirano opterećenje (da/ne): da
kolika je sila opterećenja [N] na elementu 2: -10000
Da li je na elementu 3 kontinuirano opterećenje (da/ne): ne
Da li je na elementu 4 kontinuirano opterećenje (da/ne): ne
```

Slika 5.19 – Unos kontinuiranog opterećenja (primjer 2)

Na slici 5.20 prikazan je unos vrijednosti sila i momenta koji su aktivni na čvorovima.

```
Kolika je sila F1 [N] čvoru 1: 0
Koliki je moment M1 [Nm] čvoru 1: 0
Kolika je sila F2 [N]  na čvoru  2: -5000
Koliki je moment M2 [Nm]  na čvoru  2: 0
Kolika je sila F3 [N]  na čvoru 3: 0
Koliki je moment M3 [Nm]  na čvoru  3: 0
Kolika je sila F4 [N]  na čvoru  4: 0
Koliki je moment M4 [Nm]  na čvoru  4: -5000
Kolika je sila F5 [N]  na čvoru  4: 0
Koliki je moment M5 [Nm]  na čvoru 4: -5000
```

Slika 5.20 – Unos vrijednosti sila i momenata u čvorovima (primjer 2)

Na slici 5.21 prikazan je unos stanja pomaka u čvorovima. U slučaju unosa 1 zakret/pomak je slobodan, a u slučaju unosa 0 zakret/pomak je onemogućen.

```
----------------------------------------------------------------------
 0 predstavlja onemogućen zakret/pomak
 1 predstavlja slobodan zakret/pomak
----------------------------------------------------------------------

Progib  u1 u čvoru 1 je {0 ili 1}: 0

Zakret v1 u čvoru 1 je {0 ili 1}: 1

Progib  u2 u čvoru 2 je {0 ili 1}: 1

Zakret v2 u čvoru 2 je {0 ili 1}: 1

Progib  u3 u čvoru 3 je {0 ili 1}: 1

Zakret v3 u čvoru 3 je {0 ili 1}: 1

Progib  u4 u čvoru 4 je {0 ili 1}: 0

Zakret v4 u čvoru 4 je {0 ili 1}: 1

Progib  u5 u čvoru 5 je {0 ili 1}: 0

Zakret v5 u čvoru 5 je {0 ili 1}: 0
```

Slika 5.21 – Unos stanja pomaka u čvorovima (primjer 2)

Na slici 5.22 prikazan je ispis globalne matrice krutosti.

```
Globalna matrica krutosti=
[[  1260000   1260000  -1260000   1260000         0         0         0
          0         0         0]
 [  1260000   1680000  -1260000    840000         0         0         0
          0         0         0]
 [ -1260000  -1260000  11340000   3780000 -10080000   5040000         0
          0         0         0]
 [  1260000    840000   3780000   5040000  -5040000   1680000         0
          0         0         0]
 [        0         0 -10080000  -5040000  90720000  15120000 -80640000
   20160000         0         0]
 [        0         0   5040000   1680000  15120000  10080000 -20160000
    3360000         0         0]
 [        0         0         0         0 -80640000 -20160000  81900000
  -18900000  -1260000   1260000]
 [        0         0         0         0  20160000   3360000 -18900000
    8400000  -1260000    840000]
 [        0         0         0         0         0         0  -1260000
   -1260000   1260000  -1260000]
 [        0         0         0         0         0         0   1260000
     840000  -1260000   1680000]]
======================================================================
======================================================================
```

Slika 5.22 – Međurezultati obrade podataka – globalna matrica krutosti (primjer 2)

Na slici 5.23 prikazan je ispis reduciranih matrica krutosti.

```
Reducirana matrica krutosti =
[[  1680000  -1260000    840000         0         0         0]
 [ -1260000  11340000   3780000 -10080000   5040000         0]
 [   840000   3780000   5040000  -5040000   1680000         0]
 [        0 -10080000  -5040000  90720000  15120000  20160000]
 [        0   5040000   1680000  15120000  10080000   3360000]
 [        0         0         0  20160000   3360000   8400000]]
======================================================================
======================================================================
 Reducirana matrica sila = [ 0,  -10000, -833, -5000, -5000 ]
```

Slika 5.23 – Međurezultati obrade podataka – reducirane matrice (primjer 2)

Na slici 5.24 prikazan je kraj programa. Na zaslonu se ispisuju ukupni rezultati pomaka, te rezultati reakcija u osloncima

```
Pomaci na čvorovima=
u1=  0.000000 m
u2= -0.003305 rad
u3= -0.003726 m
u4=  0.001020 rad
u5= -0.001175 m
u6=  0.003119 rad
u7=  0.000000 m
u8=  0.001176 rad
u9=  0.000000 m
u10=  0.000000 rad
======================================================================
 F1,3,5,7... predstavlja reakcijske Sile
 F2,4,6,8... predstavlja reakcijske Momente
======================================================================
Oznake sila
['F1', 'F7', 'F9', 'F10']
Reakcije u osloncima=
F1= 2724.530612 N
F2=  0.000000 Nm
F3=  0.000000 N
F4=  0.333333 Nm
F5= -0.000000 N
F6= -0.333333 Nm
F7= 14498.576531 N
F8= -0.000000 Nm
F9= -2223.107143 N
F10 = 1482.071429 Nm
```

Slika 5.24 – Rezultati programa (primjer 2)

Uporabom računala, te implementacijom MKE u programskom jeziku „Python“ znatno je olakšano rješavanje problema grede. Jednostavnost, urednost, te višestruka ušteda na vremenu su neki od prednosti primjene MKE u implementaciji računala. U oko 320 redova programskog

koda omogućeno je rješavanje jednostavnijih problema u djeliću sekunde za koji bi analitičkim postupkom bilo potrebno više sati.

6. Usporedba analitičkog računa i programskog računa

U ovom poglavlju je prikazana usporedba rezultata rješavanja problema grede na analitički način i uz primjenu programskog koda. Rješavanjem zadanih primjera međurezultati su ispali jednaki sve do dijela u kojem se izračunavaju pomaci. Razlog tome je jer su svi međurezultati zaokruženi na cjelobrojnu vrijednost. Do manjeg odstupanja dolazi kod određivanja reakcija i pomaka u čvorovima.

6.1. Usporedba prvog primjera

Tablica 6.1 – Usporedba rezultata pomaka primjera 1

	Analitički proračun	Programski proračun	Razlika u postotcima [%]
u_1 (m)	0	0	0
ϑ_1 (rad)	0	0	0
u_2 (m)	-0.0011	-0.0110	0
ϑ_2 (rad)	0.00003	0.000028	6.7
u_3 (m)	0	0	0
ϑ_3 (rad)	0.00481	0.004805	0.1
u_4 (m)	0.0255	0.0255	0.1
ϑ_4 (rad)	0.01671	0.016710	0

Tablica 6.2 – Usporedba reakcija u osloncima primjera 1

	Analitički proračun	Programski proračun	Razlika u postotcima [%]
F_1 (N)	11229.1	11226.8	0.2
M_1 (Nm)	5591.0	5590.2	0.14
F_3 (N)	-11229.1	-11226.8	0.2

Usporedbom rezultata za primjer 1 u tablici 6.1 vidljivo je da su dobiveni rezultati pomaka i nagiba približno isti. Odstupanja se javljaju nakon treće decimale, dok su neki rezultati potpuno jednaki. U tablici 6.2 prikazani su rezultati reakcija u osloncima. Rezultati reakcija u osloncima se također previše ne razlikuju. Oba primjera su relativno jednostavna što je omogućilo i analitički proračun metodom konačnih elemenata. Prava vrijednost proračuna pomoću računala se vidi kada su u pitanju problemi s velikim brojem elemenata. Točnost rezultata je veća kod računalnog proračuna radi korištenja većeg broja decimalnih mjesta pri proračunu.

6.2. Usporedba drugog primjera

Tablica 6.3 – Usporedba rezultata pomaka primjera 2

	Analitički proračun	Programski proračun	Razlika u postotcima [%]
u_1 (m)	0	0	0
ϑ_1 (rad)	-0.00330	-0.0033	0.3
u_2 (m)	-0.0037	-0.0037	0.1
ϑ_2 (rad)	0.0010	0.0010	0
u_3 (m)	0.0012	-0.0012	0.42
ϑ_3 (rad)	0.0031	0.0031	0.03
u_4 (m)	0	0	0
ϑ_4 (rad)	0.0012	0.0012	0.33
u_5 (m)	0	0	0
ϑ_5 (rad)	0	0	0

Tablica 6.4 – Usporedba rezultata reakcija u osloncima primjera 2

	Analitički proračun	Programski proračun	Razlika u postotcima [%]
F_1 (N)	2740	2724.5	0.56
F_4 (N)	14931	14498.5	3
F_5 (N)	-2250.2	-2223.1	1
M_5 (Nm)	1486.8	1482.1	0.3

Usporedbom rezultata za primjer 2 vidljivo je da se rezultati ne podudaraju dobro kao i u primjeru 1. Razlog ovoga je nešto složenija greda nego u primjeru 1. Analitički proračun postaje

kompliciraniji, greške se umnažaju, pa je i razlika u rezultatima veća. Iako su rezultati pomaka (tablica 6.3) i dalje dosta slični, osjetnije odstupanje se pojavljuje kod rezultata reakcija u osloncima (tablica 6.4).

7. Zaključak

U ovom radu je prikazan postupak rješavanja problema poprečno opterećene grede metodom konačnih elemenata, te izrada programskog koda (programski jezik „Python“) koristeći ranije izrađene dijagrame toka. U uvodnom dijelu prikazan je teorijski opis rješavanja problema grede MKE na temelju kojeg su izrađeni dijagrami toka rastavljeni na logičke cjeline. Dijagrami toka prikazuju algoritam načina rješavanja problema poprečno opterećene grede primjenom MKE. Korištenjem dijagrama toka izrađen je program. Rješavanje problema primjenom MKE analitički ograničeni smo na jednostavne probleme. Rješavanje analitičkog problema grede za zadana dva primjera iz završnog rada utrošeno je otprilike oko 5 sati računanja na papiru. Uzevši u obzir da su oba primjera jednostavnog tipa dolazi se do zaključka da nije najidealnije rješavati kompliciranе primjere problema grede metodom konačnih elemenata. Implementacijom MKE u programski jezik omogućuje se brže i jednostavnije rješavanje problema. Programski kod je napisan u svega 320 redova određenih naredbi i funkcija. Izradom programskog koda višestruko je olakšana obrada podataka, te gledajući dugoročno nemjerljivo uštеđeno na vremenu. Implementiranjem modula „time“ u programskom kodu, te korištenjem njegovih funkcija moguće je provjeriti potrebno vrijeme za izvršavanja proračuna metodom konačnih elemenata. Vrijeme potrebno za proračun pomoću računala je iznosilo svega 0.010991 sekundi. Razlika između proračuna na računalu i analitičkog načina je ogromna. Za rješavanje jednostavnog analitičkog primjera potrebno preko 2 sata rada, dok uporabom računala to vrijeme spada na nekoliko milisekundi. U poglavlju 6 prikazana je usporedba analitičkog i programskog rješavanja zadataka. Rezultati pomaka se u primjeru 1 iznose od 0.1 % do čak 6.5 %, dok u rezultati pomaka u primjeru 2 iznose oko 0.003 do 0.4 %. Rezultati reakcija u osloncima se razlikuju u razlikama od 0,3 – 3% što su zadovoljavajući rezultati. Međutim, u slučaju kompleksnijih problema ove razlike se povećavaju i mogu biti značajne. Razlog odstupanja jest računanje pomoću kalkulatora kod analitičkog postupka gdje se koristi znatno manji broj decimalnih mjesta nego kod proračuna računalom. Primjena računala omogućila je brzu i preciznu obradu velike količine podataka i dobivanje znatno preciznijih rezultata proračuna. Korišteni programski kod je izrađen korištenjem naredbi programskog jezika „Python“. Dodavanjem novih funkcionalnosti korišteni program se može znatno nadograditi. Nadogradnje koje bi znatno olakšale korištenje programom su: unos ulaznih podataka preko tablica (npr. Excel tablice), grafički prikaz grede i pomaka u čvorovima.

Program bi se mogao nadograditi mogućnošću rješavanja problema grede koje uključuju aksijalne sile, Gerbere, grede s 3D probleme grede i slično.

Literatura

1. MISHUMI MESH LAB, https://blog.misumiusa.com/computer-aided-design-pt-4-the-world-of-fea/ (10.09.2020)
2. David Hutton's Fundamental of Finite Element Analysis (01.09..2020)
3. Wikipedia, https://hr.wikipedia.org/wiki/Savijanje (10.09.2020)
4. Wikiversity,https://en.wikiversity.org/wiki/File:Global_Stiffness_Matrix_Assembly.jpg (10.09.2020)
5. Beams - FE Formulation (+ Mathcad), https://www.youtube.com/watch?v=J55VeMfLeWc (10.09.2020)
6. Beam calculator, https://structural-analyser.com/#tab_load_uniform (18.09.2020)
7. Handy math solver, https://handymath.com/cgi-bin/matrix6c.cgi (18.09.2020)