

**SVEUČILIŠTE U ZAGREBU
GRAĐEVINSKI FAKULTET**

**PROGRAMSKA REALIZACIJA
CROSSEVE METODE**

**ZAVRŠNI RAD IZ PREDMETA
GRAĐEVNA STATIKA 2**

Tea Dučić, 0082059238

Mentor: doc. dr. sc. Petra Gidak

Ak. God. 2020./2021.

Zagreb, 23. veljače 2021.

2 SADRŽAJ

1. Uvod.....	1
2. Opis Crossovog postupka.....	4
3. Primjer Crossovog postupka	6
3.1. Postupak rješavanja zadatka iz primjera.....	7
3.2. Rješenje primjera postojećim programom GS_program_Relax_(v0.4).....	12
3.3. Rješenje primjera prilagođenim program GS_program_Relax_(v0.4)	18
3.3.1. Opis prilagodbe programskog koda.....	18
3.3.2. Rješenje primjera prilagođenim programom	20
3.4. Analiza problematike odabira čvorova.....	25
3.5. Zaključak.....	27

1. UVOD

U završnom radu opisuje se Crossova iterativna metoda koja se najčešće koristi za proračun pomičnih i nepomičnih okvirnih statičkih sistema.

Crossova relaksacijska metoda je realizirana unutar programskog koda profesora Fresla. U programu se određuju vrijednosti momenata savijanja na krajevima štapnih elemenata statičkog sistema opterećenog proizvoljnim vanjskim opterećenjem.

Crossova iterativna metoda je relaksacijski postupak kojim se u koracima dolazi do sve točnijeg rješenja pa „ručno“ rješavanje ovog postupka zahtjeva određeno vrijeme, pogotovo ako se radi o kompleksnijim sustavima. Stoga programska realizacija provođenja tog postupka značajno ubrzava postupak izračuna.

Zadatak ovog završnog rada je automatizirati odabir čvorova u postojećem programskom kodu profesora Fresla GS_program_relax_(v0.4), a u kojem trenutno korisnik mora odabrati indeks čvora koji mora uravnotežiti.

2. OPIS CROSSOVOG POSTUPKA

Crossov postupak je iteracijski postupak određivanja momenata savijanja na krajevima elemenata nepomičnog štapnog sistema, odnosno to je varijanta metode pomaka kod koje se sistem linearnih jednažbi rješava iterativno.

Prvo spriječimo sve moguće zaokrete čvorova sistema te sve moguće pomake, odnosno translacije za koje se smatra da su po katovima sistema iste. Zatim se „otpušta“ čvor po čvor.

Prvo se oslobodi čvor s najvećim intenzitetom neuravnoteženog momenta. U tom čvoru nastaje mogućnost zakretanja.

Taj čvor se uravnotežuje na način da se u čvor dodaje moment jednakog intenziteta ali suprotnog smjera od rezidualnog momenta. Na taj način se čvor fizikalno zaokreće, no te zaokrete u postupku nije potrebno računati.

Nakon toga se taj moment razdijeli na priključne štapove. Razdioba se radi pomoću razdjelnog koeficijenta, koji predstavlja omjer rotacijske krutosti pojedinog štapa i zbroja rotacijske krutosti svih štapova spojenih u čvoru. Zbog toga se ovaj postupak zove postupak razdiobe momenata.

Dio od dobivenih momenata se pomoću prijenosnog koeficijenta prebacuje na suprotni kraj štapa, jer se prilikom dodavanja momenata u čvor taj čvor zakreće te se prema tome javljaju reaktivni momenti na suprotnim stranama priključnog štapa.

Nakon što se završi postupak uravnoteženja u jednom čvoru, ponovno se upne, a oslobodi čvor koji sada ima moment najvećeg intenziteta.

Postupak se provodi do željene točnosti.

Konačne vrijednosti momenata se dobiju tako da se na svakom kraju zbroje momenti upetosti, raspodijeljeni momenti i preneseni momenti.

Opisani postupak slijedi iz daljnjeg teksta.

Izraz za momente koji djeluju na krajevima štapa je:

$$M_{i,j} = 4k_{(i,j)}\rho_i + 2k_{(i,j)}\rho_j + \bar{M}_{i,j}$$

Za svaki prirast kuta zaokreta $\Delta\rho_i^{(n_i+1)}$ čvora i , mijenja se i vrijednosti momenata na krajevima štapa koji pripadaju tom čvoru. S obzirom na to da su prilikom zaokreta čvora i , svi zaokreti ostalih čvorova spriječeni pa je $\Delta\rho_{ji} = 0$, ta promjena momenata iznosi:

$$\Delta M_{i,k}^{(n_i+1)} = 4k_{(i,j_i)} \cdot \Delta\varphi_i^{(n_i+1)}.$$

Ako u ovaj izraz uvrstimo izraz za $\Delta\varphi_i^{(n_i+1)} = \frac{m_i^{(n_i)}}{4\sum_{j_i} k_{(i,j)}}$ dobijemo izraz:

$$\Delta M_{i,k}^{(n_i+1)} = 4k_{(i,j)} \frac{m_i^{(n_i)}}{4\sum_{j_i} k_{(i,j)}} = \frac{k_{(i,j)}}{\sum_{j_i} k_{(i,j)}} \cdot m_i^{(n_i)}$$

Zbrojimo rotacijske krutosti svih elemenata priključenih u čvor:

$$k_i = \sum_{j_i} k_{(i,j_i)},$$

Omjer pojedine rotacijske krutosti i zbroja rotacijskih krutosti naziva se razdjelni koeficijent:

$$\mu_{i,j_i} = \frac{4k_{(i,j_i)}}{k_i}, \quad \sum_{j_i} \mu_{(i,j_i)} = 1$$

Uvrštavanjem razdjelnog koeficijenta u jednadžbu za prirast vrijednosti momenta na kraju elementa dobivamo

$$\Delta M_{i,k}^{(n_i+1)} = \frac{k_{(i,j)}}{\sum_{j_i} k_{(i,j)}} \cdot m_i^{(n_i)} = \mu_{(i,j_i)} \cdot m_i^{(n_i)}.$$

Zbroj momenata na krajevima štapa, a s obzirom na to da je $\sum_{j_i} \mu_{(i,j_i)} = 1$ iznosi:

$$\sum_{j_i} \Delta M_{(i,j_i)}^{(n_i+1)} = \sum_{j_i} \mu_{(i,j_i)} \cdot m_i^{(n_i)} = m_i^{(n_i)}$$

Prebacivanjem svih članova na lijevu dobivamo:

$$-\sum_{j_i} \Delta M_{(i,j_i)}^{(n_i+1)} + m_i^{(n_i)} = 0$$

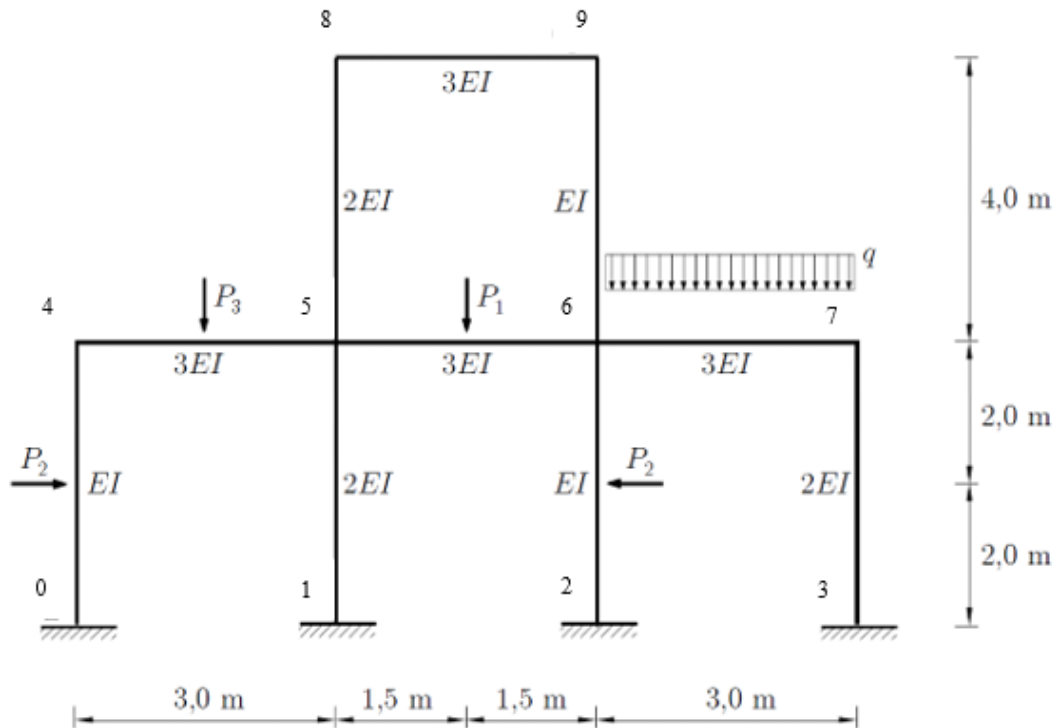
Ovaj izraz nam ukazuje da čvor i možemo uravnotežiti tako da na njega dodamo moment istog intenziteta jednakog rezidualnom momentu, ali suprotnog smisla vrtnje, te ga razdijelimo u omjeru krutosti priključnih štapova.

Nakon uravnoteženja čvora i treba na drugi kraj svakog priključnog štapa dodati moment čija je vrijednost jednaka polovini vrijednosti momenta s kraja i . Dakle „prenesemo pola momenta“, tako da je prijenosni koeficijent $\frac{1}{2}$.

$$\Delta M_{j,i} = 2 \cdot k_{(i,j)} \cdot \Delta \rho_i^{(n_i+1)} = \frac{1}{2} \cdot \Delta M_{(i,j)}^{(n_i+1)}.$$

3. PRIMJER CROSSOVOG POSTUPKA

Crossov iterativni postupak pokazati ću na primjeru:



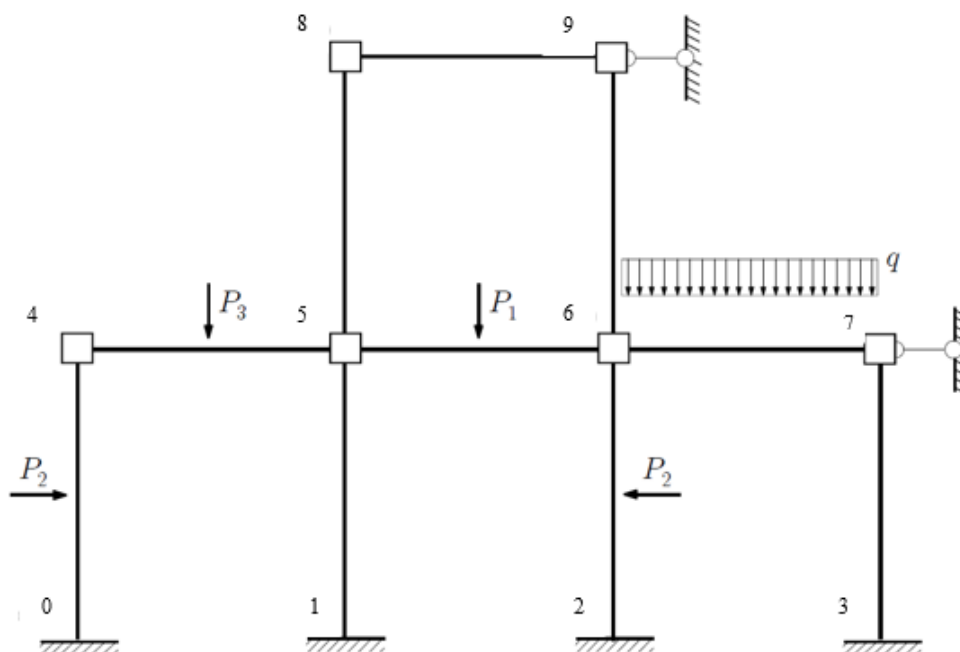
Ulazni podaci:

$$\begin{aligned} P_1 &= 60 \text{ [kN]} & l_{0,4} &= l_{1,5} = l_{2,6} = l_{3,7} = l_{5,8} = l_{6,9} = 4.0 \text{ [m]} \\ P_2 &= 80 \text{ [kN]} & l_{4,5} &= l_{5,6} = l_{6,7} = l_{8,9} = 3.0 \text{ [m]} \\ P_3 &= 90 \text{ [kN]} \\ q &= 18 \text{ [kN/m]} \\ EI &= 150000 \text{ [kNm}^2\text{]} \end{aligned}$$

3.1. Postupak rješavanja zadatka iz primjera

Crossov postupak primjenjiv je samo za nepomične sisteme. S obzirom na to da u primjeru postoji mogućnost translacijskih pomaka, Crossov postupak u ovom slučaju potrebno je provesti tri puta.

U prvom dijelu Crossovim postupkom rješava se nepomični sistem koji se dobije tako da se dodavanjem veza spriječe svi pomaci. Nakon toga se odrede sile u pridržanim vezama koje sprječavaju neovisne translacijske pomake.



U drugom dijelu konstrukciji se daju pomaci na pravcima veza koje sprječavaju neovisne pomake. Nakon pomaka, konstrukcija se pridržava u novom položaju i izračunavaju momenti na krajevima štapova Crossovim postupkom te se odrede sile u pridržanim vezama. Postupak ponavljamo dodavanjem pomaka na pravcu svakog neovisnog pomaka.

Za potrebe objašnjenja programske realizacije Crossovog postupka, gledala sam samo prvo provođenje Crossovog postupka.

Proračun provodimo na shemi sistema. Čvorovi su označeni kvadratićima, u koje se unose razdjelni koeficijenti prema priključnim štapovima. Na krajeve štapa upisuju se vrijednosti pripadnih momenata upetosti.

Zato prvo računamo koeficijent krutosti štapova, a zatim sumu rotacijskih krutosti svakog čvora i razdjelne koeficijente:

KOEFICIJENTI KRUTOSTI ŠTAPOVA:

$$k_{0,4} = \frac{EI}{l_{0,4}} = \frac{150000 [kNm^2]}{4 [m]} = 37500 [kNm] \quad k_{1,5} = \frac{EI}{l_{1,5}} = \frac{300000 [kNm^2]}{4 [m]} = 75000 [kNm]$$

$$k_{2,6} = \frac{EI}{l_{2,6}} = \frac{150000 [kNm^2]}{4 [m]} = 37500 [kNm] \quad k_{3,7} = \frac{2EI}{l_{3,7}} = \frac{300000 [kNm^2]}{4 [m]} = 75000 [kNm]$$

$$k_{4,5} = \frac{3EI}{l_{4,5}} = \frac{450000 [kNm^2]}{3 [m]} = 150000 [kNm] \quad k_{5,6} = \frac{3EI}{l_{5,6}} = \frac{450000 [kNm^2]}{3 [m]} = 150000 [kNm]$$

$$k_{5,8} = \frac{2EI}{l_{5,8}} = \frac{300000 [kNm^2]}{4 [m]} = 75000 [kNm] \quad k_{6,7} = \frac{3EI}{l_{6,7}} = \frac{450000 [kNm^2]}{3 [m]} = 150000 [kNm]$$

$$k_{6,9} = \frac{EI}{l_{6,9}} = \frac{150000 [kNm^2]}{3 [m]} = 37500 [kNm] \quad k_{8,9} = \frac{3EI}{l_{8,9}} = \frac{450000 [kNm^2]}{3 [m]} = 150000 [kNm]$$

SUMA ROTACIJSKIH KRUTOSTI I RAZDJELNI KOEFICIJENTI:

Razdjelnih koeficijenta koji se dobivaju iz omjera rotacijske krutosti pojedinog štapu i zbroja krutosti svih štapova spojenih u čvoru.

Čvor 4:

$$k_4 = 4k_{4,0} + 4k_{4,5} = 750000$$

$$\mu_{4,0} = \frac{k_{4,0}}{k_4} = 0,20$$

$$\mu_{4,5} = \frac{k_{4,5}}{k_4} = 0,80$$

Čvor 5:

$$k_5 = 4k_{5,4} + 4k_{5,1} + 4k_{5,6} + 4k_{5,8} = 180000$$

$$\mu_{5,4} = \frac{k_{5,4}}{k_5} = 0,33 \quad \mu_{5,1} = \frac{k_{5,1}}{k_5} = 0,17$$

$$\mu_{5,6} = \frac{k_{5,6}}{k_5} = 0,33 \quad \mu_{5,8} = \frac{k_{5,8}}{k_5} = 0,17$$

Čvor 7:

$$k_7 = 4k_{7,6} + 4k_{7,3} = 750000$$

$$\mu_{7,6} = \frac{k_{7,6}}{k_7} = 0,67$$

$$\mu_{7,3} = \frac{k_{7,3}}{k_7} = 0,33$$

Čvor 6:

$$k_6 = 4k_{6,5} + 4k_{6,2} + 4k_{6,7} + 4k_{6,9} = 1500000$$

$$\mu_{6,5} = \frac{k_{6,5}}{k_6} = 0,40 \quad \mu_{6,2} = \frac{k_{6,2}}{k_6} = 0,10$$

$$\mu_{6,7} = \frac{k_{6,7}}{k_6} = 0,40 \quad \mu_{6,9} = \frac{k_{6,9}}{k_6} = 0,10$$

Čvor 8:

$$k_8 = 4k_{8,5} + 4k_{8,9} = 900000$$

$$\mu_{8,5} = \frac{k_{8,5}}{k_8} = 0,33$$

$$\mu_{8,9} = \frac{k_{8,9}}{k_8} = 0,67$$

Čvor 9:

$$k_9 = 4k_{9,8} + 4k_{9,6} = 750000$$

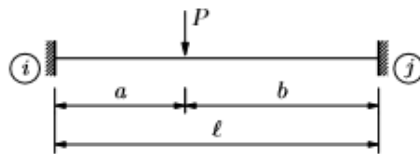
$$\mu_{9,8} = \frac{k_{9,8}}{k_9} = 0,80$$

$$\mu_{9,6} = \frac{k_{9,6}}{k_9} = 0,20$$

MOMENTI UPETOSTI:

S obzirom na to da su u prvom koraku sprječavani svi zaokreti i pomaci, na krajevima elemenata se javljaju momenti upetosti.

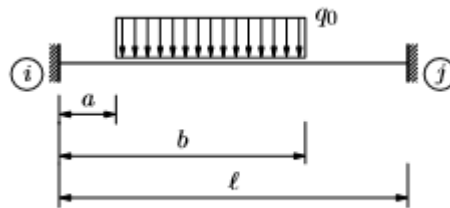
Na krajevima elementa (0,4) i (2,6) javljaju se momenti upetosti od vanjske sile P_2 , na krajevima elementa (5,6) od vanjske sile P_3 , a na krajevima elementa (6,7) od P_1 .



$$\bar{M}_{i,j} = \frac{P \cdot ab^2}{l^2} \quad \bar{M}_{j,i} = -\frac{P \cdot a^2 b}{l^2}$$

$$\text{a s obzirom na to da je } a = b = \frac{l}{2} \quad \bar{M}_{i,j} = -\bar{M}_{j,i} = \frac{P \cdot l_{i,j}}{8}$$

Na krajevima elementa (7,8) javljaju se momenti od opterećenja q :



$$\bar{M}_{i,j} = q_0 \left[\frac{b^2 - a^2}{2} - 2 \frac{b^3 - a^3}{3l} + \frac{b^4 - a^4}{4l^2} \right] \quad \bar{M}_{j,i} = q_0 \left[-\frac{b^3 - a^3}{3l} + \frac{b^4 - a^4}{4l^2} \right]$$

S obzirom na to da opterećenje djeluje po cijeloj dužini elementa imamo:

$$a = 0, \quad b = l \quad \bar{M}_{i,j} = -\bar{M}_{j,i} = \frac{q_0 l^2}{12}$$

Dakle, u primjeru su momenti upetosti:

$$\bar{M}_{0,4} = -\bar{M}_{4,0} = \frac{P_2 \cdot l_{0,4}}{8} = 40.00 \text{ kNm} \quad \bar{M}_{5,6} = -\bar{M}_{6,5} = \frac{P_1 \cdot l_{1,5}}{8} = 22.50 \text{ kNm}$$

$$\bar{M}_{2,6} = -\bar{M}_{6,2} = \frac{P_2 \cdot l_{2,6}}{8} = -40.00 \text{ kNm} \quad \bar{M}_{7,8} = -\bar{M}_{8,7} = \frac{q \cdot l_{7,8}^2}{12} = 13.50 \text{ kNm}$$

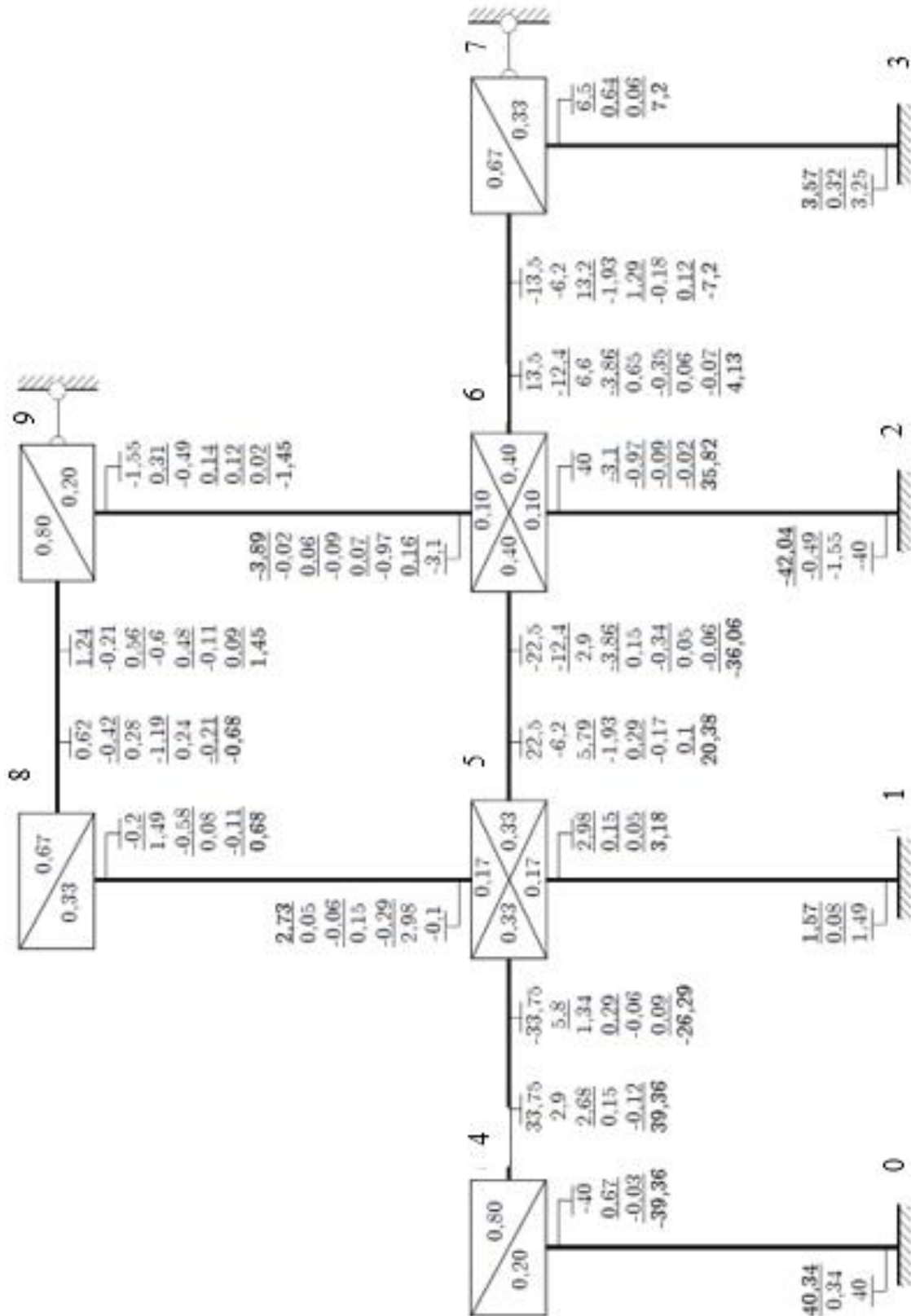
$$\bar{M}_{4,5} = -\bar{M}_{5,4} = \frac{P_3 \cdot l_{1,5}}{8} = 33.75 \text{ kNm}$$

Dobivene rezultate upisujemo u shemi kako je opisano na početku poglavlja.

Odabire se moment najvećeg intenziteta. U ovom primjeru to je neuravnoteženi moment u čvoru 6 pa taj čvor uravnotežujemo. Uravnotežujemo ga na shemi pomoću razdjelnih koeficijenta, na način da neuravnoteženi moment suprotnog smjera množimo razdjelnim koeficijentom. Dobivene rezultate upisujemo u shemu.

Dio momenta prenosimo na druge krajeve priključenih štapova. To radimo na shemi s prijenosnim koeficijentima, koji je u primjeru $1/2$, odnosno $1/2$ iznosa prebacujemo na suprotni kraj štapa.

Nakon što smo završili postupak uravnoteženja u jednom čvoru, ponovno ga upnemo, a postupak ponavljamo na sljedećem čvoru. Taj postupak ponavljamo sve dok u svim čvorovima neuravnoteženi moment bude približno 0.



Konačni momenti na krajevima štapa se dobivaju tako da se zbroje momenti upetosti, raspodijeljeni momenti i preneseni momenti tijekom iteracije.

3.2. Rješenje primjera postojećim programom `GS_program_Relax_(v0.4)`

U programu profesora Fresla izvršava se izračun koji se ručno provodi na shemi sistema, na osnovu prethodno izračunanih razdjelnih koeficijenata, momenata upetosti te određenih prijenosnih koeficijenata.

U kodu se dobiveni razdjelni koeficijenti unesu kao varijabla `distr_f` strukture rječnik, u kojoj ključ predstavlja oznaku štapa (i,j), a vrijednost predstavlja iznos razdjelnog koeficijenta.

Moment upetosti se unese kao varijabla `mom_f` strukture rječnik, u kojoj ključ predstavlja oznaku štapa (i,j), a vrijednost predstavlja iznos momenta upetosti u čvoru i .

Prijenosni koeficijent je $\frac{1}{2}$ pa se unese kao varijabla `co_f`.

```
# razdjelni koeficijenti:
distr_f = { (4, 0): 0.20, (4, 5): 0.80, (5, 4): 0.33, (5, 1): 0.17, (5, 6): 0.33, (5, 8): 0.17,
(6, 5): 0.40, (6, 2): 0.10, (6, 7): 0.40, (6, 9): 0.10, (7, 6): 0.67, (7, 3): 0.33, (8, 5): 0.33,
(8, 9): 0.67, (9, 8): 0.80, (9, 6): 0.20 }

# prijenosni koeficijent:
co_f = 1/2

# momenti upetosti:
mom_fe = { (0, 4): 40.0, (4, 0): -40.0, (2, 6): -40.0, (6, 2): 40.0, (4, 5): 33.75, (5, 4): -33.75,
(5, 6): 22.50, (6, 5): -22.50, (6, 7): 13.5, (7, 6): -13.5 }
```

Pozivom `init_all` funkcije prikazuju se razdjelni koeficijenti, prijenosni koeficijenti, momenti upetosti te se izračunaju rezidualni momenti:

```
init_all (distr_f, co_f, mom_fe)
```

```
distribution factors:
```

```
(4, 0): 0.20
(4, 5): 0.80
(5, 1): 0.17
(5, 4): 0.33
(5, 6): 0.33
(5, 8): 0.17
(6, 2): 0.10
(6, 5): 0.40
(6, 7): 0.40
(6, 9): 0.10
(7, 3): 0.33
(7, 6): 0.67
(8, 5): 0.33
(8, 9): 0.67
(9, 6): 0.20
(9, 8): 0.80
```

```
carry-over factors:
```

```
(4, 0): 1/2
(4, 5): 1/2
(5, 1): 1/2
(5, 4): 1/2
```

```
(5, 6): 1/2
(5, 8): 1/2
(6, 2): 1/2
(6, 5): 1/2
(6, 7): 1/2
(6, 9): 1/2
(7, 3): 1/2
(7, 6): 1/2
(8, 5): 1/2
(8, 9): 1/2
(9, 6): 1/2
(9, 8): 1/2
```

initial moment lists:

```
(0, 4): [40.0]
(1, 5): []
(2, 6): [-40.0]
(3, 7): []
(4, 0): [-40.0]
(4, 5): [33.7]
(5, 1): []
(5, 4): [-33.8]
(5, 6): [22.5]
(5, 8): []
(6, 2): [40.0]
(6, 5): [-22.5]
(6, 7): [13.5]
(6, 9): []
(7, 3): []
(7, 6): [-13.5]
(8, 5): []
(8, 9): []
(9, 6): []
(9, 8): []
```

initial residual moments:

```
4: -6.3
5: -11.3
6: 31.0
7: -13.5
8: 0.00
9: 0.00
```

Definirana je funkcija `_relax_step`, u kojoj je ulazni parametar broj čvora kojeg želimo uravnotežiti. Pri tom se ulazni parametar `nd` kontrolira kroz korisničko sučelje.

```
@interact
def _relax_step ( nd = ('node: ', None) ) :
    if nd != None :
        distribute_and_carry_over (nd)
        calc_residuals()
```

Izvršavanjem ovog dijela koda otvara se prozor za unos broja čvora:



The screenshot shows a window with a light gray background. On the left side, there is a label 'node:' followed by a rectangular text input field. The input field contains the text 'None'. Below the input field, there is a horizontal line of text that is mostly obscured but appears to be a continuation of the code or a description.

Upišemo broj čvora s najvećim neuravnoteženim momentom, a u ovom primjeru to je čvor s indeksom 6. Kao rezultat dobivamo izračun uravnoteženja tog čvora:

```

distributed moments:
(6, 2): 0.10 * -31.0 = -3.1
(6, 5): 0.40 * -31.0 = -12.4
(6, 7): 0.40 * -31.0 = -12.4
(6, 9): 0.10 * -31.0 = -3.1

carry-over moments:
(6, 2) -> (2, 6): 1/2 * -3.1 = -1.5
(6, 5) -> (5, 6): 1/2 * -12.4 = -6.2
(6, 7) -> (7, 6): 1/2 * -12.4 = -6.2
(6, 9) -> (9, 6): 1/2 * -3.1 = -1.6

residual moments:
4: -6.3
5: -17.5
6: 0.00
7: -19.7
8: 0.00
9: -1.6

```

Iz dobivenog rezultata ponovo očitamo čvor čiji je iznos neuravnoteženog momenta najveći što je u ovom slučaju čvor s indeksom 7 te ga upišemo u korisničko sučelje.

Postupak ponavljamo sve dok svi članovi u *residual moments* nisu 0.0.

Nakon ponavljanja određenog broja puta dobiveni su sljedeći rezultati:

```

Čvor 6:
distributed moments:
(6, 2): 0.10 * -31.0 = -3.1
(6, 5): 0.40 * -31.0 = -12.4
(6, 7): 0.40 * -31.0 = -12.4
(6, 9): 0.10 * -31.0 = -3.1

carry-over moments:
(6, 2) -> (2, 6): 1/2 * -3.1 = -1.5
(6, 5) -> (5, 6): 1/2 * -12.4 = -6.2
(6, 7) -> (7, 6): 1/2 * -12.4 = -6.2
(6, 9) -> (9, 6): 1/2 * -3.1 = -1.6

residual moments:
4: -6.3
5: -17.5
6: 0.00
7: -19.7
8: 0.00
9: -1.6

```

```

Čvor 7:
distributed moments:
(7, 3): 0.33 * 19.7 = 6.5
(7, 6): 0.67 * 19.7 = 13.2

carry-over moments:
(7, 3) -> (3, 7): 1/2 * 6.5 = 3.2
(7, 6) -> (6, 7): 1/2 * 13.2 = 6.6

residual moments:
4: -6.3
5: -17.5
6: 6.6
7: 0.00
8: 0.00
9: -1.6

```

```

Čvor 5:
distributed moments:
(5, 1): 0.17 * 17.5 = 3.0
(5, 4): 0.33 * 17.5 = 5.8
(5, 6): 0.33 * 17.5 = 5.8
(5, 8): 0.17 * 17.5 = 3.0

carry-over moments:
(5, 1) -> (1, 5): 1/2 * 3.0 = 1.5
(5, 4) -> (4, 5): 1/2 * 5.8 = 2.9
(5, 6) -> (6, 5): 1/2 * 5.8 = 2.9
(5, 8) -> (8, 5): 1/2 * 3.0 = 1.5

residual moments:
4: -3.4
5: 0.00
6: 9.5
7: 0.00
8: 1.5
9: -1.6

```

```

Čvor 6:
distributed moments:
(6, 2): 0.10 * -9.5 = -1.0
(6, 5): 0.40 * -9.5 = -3.8
(6, 7): 0.40 * -9.5 = -3.8
(6, 9): 0.10 * -9.5 = -0.90

carry-over moments:
(6, 2) -> (2, 6): 1/2 * -1.0 = -0.50
(6, 5) -> (5, 6): 1/2 * -3.8 = -1.9
(6, 7) -> (7, 6): 1/2 * -3.8 = -1.9
(6, 9) -> (9, 6): 1/2 * -0.90 = -0.50

residual moments:
4: -3.4
5: -1.9
6: 0.00
7: -1.9
8: 1.5
9: -2.1

```

Čvor 4:

distributed moments:

$$(4, 0): 0.20 * 3.4 = 0.70$$

$$(4, 5): 0.80 * 3.4 = 2.7$$

carry-over moments:

$$(4, 0) \rightarrow (0, 4): 1/2 * 0.70 = 0.30$$

$$(4, 5) \rightarrow (5, 4): 1/2 * 2.7 = 1.4$$

residual moments:

$$4: 0.00$$

$$5: -0.50$$

$$6: 0.00$$

$$7: -1.9$$

$$8: 1.5$$

$$9: -2.1$$

Čvor 8:

distributed moments:

$$(8, 5): 0.33 * -2.3 = -0.80$$

$$(8, 9): 0.67 * -2.3 = -1.5$$

carry-over moments:

$$(8, 5) \rightarrow (5, 8): 1/2 * -0.80 = -0.40$$

$$(8, 9) \rightarrow (9, 8): 1/2 * -1.5 = -0.80$$

residual moments:

$$4: 0.00$$

$$5: -0.90$$

$$6: 0.20$$

$$7: -1.9$$

$$8: 0.00$$

$$9: -0.80$$

Čvor 5:

distributed moments:

$$(5, 1): 0.17 * 0.90 = 0.20$$

$$(5, 4): 0.33 * 0.90 = 0.30$$

$$(5, 6): 0.33 * 0.90 = 0.30$$

$$(5, 8): 0.17 * 0.90 = 0.10$$

carry-over moments:

$$(5, 1) \rightarrow (1, 5): 1/2 * 0.20 = 0.10$$

$$(5, 4) \rightarrow (4, 5): 1/2 * 0.30 = 0.20$$

$$(5, 6) \rightarrow (6, 5): 1/2 * 0.30 = 0.10$$

$$(5, 8) \rightarrow (8, 5): 1/2 * 0.10 = 0.00$$

residual moments:

$$4: 0.20$$

$$5: 0.00$$

$$6: 0.90$$

$$7: 0.00$$

$$8: 0.00$$

$$9: -0.80$$

Čvor 9:

distributed moments:

$$(9, 6): 0.20 * 0.80 = 0.20$$

$$(9, 8): 0.80 * 0.80 = 0.60$$

carry-over moments:

$$(9, 6) \rightarrow (6, 9): 1/2 * 0.20 = 0.10$$

$$(9, 8) \rightarrow (8, 9): 1/2 * 0.60 = 0.30$$

residual moments:

$$4: 0.20$$

$$5: -0.20$$

$$6: 0.10$$

$$7: -0.20$$

$$8: 0.30$$

$$9: 0.00$$

Čvor 9:

distributed moments:

$$(9, 6): 0.20 * 2.1 = 0.40$$

$$(9, 8): 0.80 * 2.1 = 1.7$$

carry-over moments:

$$(9, 6) \rightarrow (6, 9): 1/2 * 0.40 = 0.20$$

$$(9, 8) \rightarrow (8, 9): 1/2 * 1.7 = 0.80$$

residual moments:

$$4: 0.00$$

$$5: -0.50$$

$$6: 0.20$$

$$7: -1.9$$

$$8: 2.3$$

$$9: 0.00$$

Čvor 7:

distributed moments:

$$(7, 3): 0.33 * 1.9 = 0.60$$

$$(7, 6): 0.67 * 1.9 = 1.3$$

carry-over moments:

$$(7, 3) \rightarrow (3, 7): 1/2 * 0.60 = 0.30$$

$$(7, 6) \rightarrow (6, 7): 1/2 * 1.3 = 0.60$$

residual moments:

$$4: 0.00$$

$$5: -0.90$$

$$6: 0.80$$

$$7: 0.00$$

$$8: 0.00$$

$$9: -0.80$$

Čvor 6:

distributed moments:

$$(6, 2): 0.10 * -0.90 = -0.10$$

$$(6, 5): 0.40 * -0.90 = -0.40$$

$$(6, 7): 0.40 * -0.90 = -0.40$$

$$(6, 9): 0.10 * -0.90 = -0.10$$

carry-over moments:

$$(6, 2) \rightarrow (2, 6): 1/2 * -0.10 = -0.10$$

$$(6, 5) \rightarrow (5, 6): 1/2 * -0.40 = -0.20$$

$$(6, 7) \rightarrow (7, 6): 1/2 * -0.40 = -0.20$$

$$(6, 9) \rightarrow (9, 6): 1/2 * -0.10 = 0.00$$

residual moments:

$$4: 0.20$$

$$5: -0.20$$

$$6: 0.00$$

$$7: -0.20$$

$$8: 0.00$$

$$9: -0.80$$

Čvor 8:

distributed moments:

$$(8, 5): 0.33 * -0.30 = -0.10$$

$$(8, 9): 0.67 * -0.30 = -0.20$$

carry-over moments:

$$(8, 5) \rightarrow (5, 8): 1/2 * -0.10 = -0.10$$

$$(8, 9) \rightarrow (9, 8): 1/2 * -0.20 = -0.10$$

residual moments:

$$4: 0.20$$

$$5: -0.30$$

$$6: 0.10$$

$$7: -0.20$$

$$8: 0.00$$

$$9: -0.10$$

Čvor 5:

distributed moments:

(5, 1): $0.17 * 0.30 = 0.00$
 (5, 4): $0.33 * 0.30 = 0.10$
 (5, 6): $0.33 * 0.30 = 0.10$
 (5, 8): $0.17 * 0.30 = 0.00$

carry-over moments:

(5, 1) -> (1, 5): $1/2 * 0.00 = 0.00$
 (5, 4) -> (4, 5): $1/2 * 0.10 = 0.00$
 (5, 6) -> (6, 5): $1/2 * 0.10 = 0.00$
 (5, 8) -> (8, 5): $1/2 * 0.00 = 0.00$

residual moments:

4: 0.20
 5: 0.00
 6: 0.10
 7: -0.20
 8: 0.00
 9: -0.10

Čvor 7:

distributed moments:

(7, 3): $0.33 * 0.20 = 0.10$
 (7, 6): $0.67 * 0.20 = 0.10$

carry-over moments:

(7, 3) -> (3, 7): $1/2 * 0.10 = 0.00$
 (7, 6) -> (6, 7): $1/2 * 0.10 = 0.00$

residual moments:

4: 0.00
 5: -0.10
 6: 0.10
 7: 0.00
 8: 0.00
 9: -0.10

Čvor 5:

distributed moments:

(5, 1): $0.17 * 0.10 = 0.00$
 (5, 4): $0.33 * 0.10 = 0.00$
 (5, 6): $0.33 * 0.10 = 0.00$
 (5, 8): $0.17 * 0.10 = 0.00$

carry-over moments:

(5, 1) -> (1, 5): $1/2 * 0.00 = 0.00$
 (5, 4) -> (4, 5): $1/2 * 0.00 = 0.00$
 (5, 6) -> (6, 5): $1/2 * 0.00 = 0.00$
 (5, 8) -> (8, 5): $1/2 * 0.00 = 0.00$

residual moments:

4: 0.00
 5: 0.00
 6: 0.00
 7: 0.00
 8: 0.00
 9: -0.10

Čvor 4:

distributed moments:

(4, 0): $0.20 * -0.20 = 0.00$
 (4, 5): $0.80 * -0.20 = -0.20$

carry-over moments:

(4, 0) -> (0, 4): $1/2 * 0.00 = 0.00$
 (4, 5) -> (5, 4): $1/2 * -0.20 = -0.10$

residual moments:

4: 0.00
 5: -0.10
 6: 0.10
 7: -0.20
 8: 0.00
 9: -0.10

Čvor 6:

distributed moments:

(6, 2): $0.10 * -0.10 = 0.00$
 (6, 5): $0.40 * -0.10 = 0.00$
 (6, 7): $0.40 * -0.10 = 0.00$
 (6, 9): $0.10 * -0.10 = 0.00$

carry-over moments:

(6, 2) -> (2, 6): $1/2 * 0.00 = 0.00$
 (6, 5) -> (5, 6): $1/2 * 0.00 = 0.00$
 (6, 7) -> (7, 6): $1/2 * 0.00 = 0.00$
 (6, 9) -> (9, 6): $1/2 * 0.00 = 0.00$

residual moments:

4: 0.00
 5: -0.10
 6: 0.00
 7: 0.00
 8: 0.00
 9: -0.10

Čvor 9:

distributed moments:

(9, 6): $0.20 * 0.10 = 0.00$
 (9, 8): $0.80 * 0.10 = 0.10$

carry-over moments:

(9, 6) -> (6, 9): $1/2 * 0.00 = 0.00$
 (9, 8) -> (8, 9): $1/2 * 0.10 = 0.00$

residual moments:

4: 0.00
 5: 0.00
 6: 0.00
 7: 0.00
 8: 0.00
 9: 0.00

Naredba nsteps vraća broj iteracija:

nsteps

Pozove se funkcija koja vraća redoslijed uzimanja čvorova:

```
relax_order|
[6, 7, 5, 6, 4, 9, 8, 7, 5, 6, 9, 8, 5, 4, 7, 6, 5, 9]
```

Ispisuje se lista svih momenata koji su se javili pri izračunu na svakom od čvorova:

```
print_dict (moment_lists)

(0, 4): [40.0, 0.30, 0.00]
(1, 5): [1.5, 0.10, 0.00, 0.00]
(2, 6): [-40.0, -1.5, -0.50, -0.10, 0.00]
(3, 7): [3.2, 0.30, 0.00]
(4, 0): [-40.0, 0.70, '|', 0.00, '|']
(4, 5): [33.7, 2.9, 2.7, '|', 0.20, 0.00, -0.20, '|', 0.00]
(5, 1): [3.0, '|', 0.20, '|', 0.00, '|', 0.00, '|']
(5, 4): [-33.8, 5.8, '|', 1.4, 0.30, '|', 0.10, '|', -0.10, 0.00, '|']
(5, 6): [22.5, -6.2, 5.8, '|', -1.9, 0.30, '|', -0.20, 0.10, '|', 0.00, 0.00,
'|']
(5, 8): [3.0, '|', -0.40, 0.10, '|', -0.10, 0.00, '|', 0.00, '|']
(6, 2): [40.0, -3.1, '|', -1.0, '|', -0.10, '|', 0.00, '|']
(6, 5): [-22.5, -12.4, '|', 2.9, -3.8, '|', 0.10, -0.40, '|', 0.00, 0.00, '|',
0.00]
(6, 7): [13.5, -12.4, '|', 6.6, -3.8, '|', 0.60, -0.40, '|', 0.00, 0.00, '|']
(6, 9): [-3.1, '|', -0.90, '|', 0.20, -0.10, '|', 0.10, 0.00, '|', 0.00]
(7, 3): [6.5, '|', 0.60, '|', 0.10, '|']
(7, 6): [-13.5, -6.2, 13.2, '|', -1.9, 1.3, '|', -0.20, 0.10, '|', 0.00]
(8, 5): [1.5, -0.80, '|', 0.00, -0.10, '|', 0.00, 0.00]
(8, 9): [0.80, -1.5, '|', 0.30, -0.20, '|', 0.00]
(9, 6): [-1.6, -0.50, 0.40, '|', 0.00, 0.20, '|', 0.00, 0.00, '|']
(9, 8): [1.7, '|', -0.80, 0.60, '|', -0.10, 0.10, '|']
```

Konačni momenti na krajevima štapa se dobivaju zbrajanjem momenata upetosti, raspodijeljenih momenata i prenesenih momenata koji su se javili tijekom iteracije.

```
sum_end_moments ()
```

```
final moments:
(0, 4): 40.3
(1, 5): 1.6
(2, 6): -42.1
(3, 7): 3.5
(4, 0): -39.3
(4, 5): 39.3
(5, 1): 3.2
(5, 4): -26.3
(5, 6): 20.4
(5, 8): 2.6
(6, 2): 35.8
(6, 5): -36.1
(6, 7): 4.1
(6, 9): -3.8
(7, 3): 7.2
(7, 6): -7.2
(8, 5): 0.60
(8, 9): -0.60
(9, 6): -1.5
(9, 8): 1.5

final residual moments:
4: 0.00
5: -0.10
6: 0.00
7: 0.00
8: 0.00
9: 0.00
```

Konačni rezultat je:

```
patch_moments()
```

```
final final moments:
(0, 4): 40.3
(1, 5): 1.6
(2, 6): -42.1
(3, 7): 3.5
(4, 0): -39.3
(4, 5): 39.3
(5, 1): 3.2
(5, 4): -26.2
(5, 6): 20.4
(5, 8): 2.6
(6, 2): 35.8
(6, 5): -36.1
(6, 7): 4.1
(6, 9): -3.8
(7, 3): 7.2
(7, 6): -7.2
(8, 5): 0.60
(8, 9): -0.60
(9, 6): -1.5
(9, 8): 1.5
```

3.3. Rješenje primjera prilagođenim program GS_program_Relax_(v0.4)

3.3.1. Opis prilagodbe programskog koda

Funkcija `_relax_step` prilagođena je na način da se ulazni parametar `nd` više ne kontrolira kroz korisničko sučelje.

Kod funkcije profesora Fresla:

```
@interact
def _relax_step ( nd = ('node: ', None) ) :
    if nd != None :
        distribute_and_carry_over (nd)
        calc_residuals()
```

Kod prilagođene funkcije:

```
def _relax_step (nd):
    distribute_and_carry_over (nd)
    calc_residuals()
```

Pozivom navedene funkcije izvornog programa korisnik treba sam upisati indeks neuravnoteženog čvor te procijeniti je li dostignuta potrebna točnost. Cilj ovog završnog rada je da se određivanje neuravnoteženog čvora i dostizanje predviđene točnosti izvrši programski. Stoga sam uvela dvije nove funkcije radi programskog određivanja opisanog.

Za određivanja čvora s najvećim iznosom neuravnoteženog momenta definirala sam funkciju *odredi_cvor(resid_mom)*.

Funkciji se kao ulazni parametar predaju neuravnoteženi momenti na čvorovima strukture rječnik. Ključ je broj čvora, a vrijednost je iznos neuravnoteženih momenata.

Funkcija uspoređuje sve vrijednosti, te vraća ključ čvora čiji je iznos apsolutne vrijednost momenta najveći. Pri tom, ako imamo dva momenta s istim apsolutnim vrijednostima, uzima se čvor s pozitivnim iznosom momentom. Ako imamo dva ili više momenta istog iznosa funkcija vraća prvi čvor s tim iznosom, odnosno vraća čvor s najmanjim indeksom, s obzirom na to da funkcije koje vraćaju rezidualne elemente (*init_all* i *calc_residual* vraćaju momente s uzlazno poredanim ključevima).

```
def odredi_cvor(resid_mom):
    list_kljuceva=resid_mom.keys()
    max_mom=0
    for i in range(0,len(resid_mom)):
        novi_cvor=resid_mom[list_kljuceva[i]]
        if max_mom.abs()<novi_cvor.abs() or max_mom.abs()==novi_cvor.abs() and max_mom<novi_cvor:
            kluc=list_kljuceva[i]
            max_mom=novi_cvor
    return kluc
```

Kako bi se odredilo kada je postupak iteracije završen, definirala sam novu funkciju *nastavi_dalje(resid_mom, tocnost)*, koja kao ulazne parametre ima neuravnotežene momente strukture rječnik *resid_mom* i točnost izračuna.

Funkcija provjerava je li zadovoljen uvjet da je u bar jednom čvoru neuravnoteženi moment veći od neke unaprijed definirane točnosti te ukoliko postoji takav čvor *nastavi_dalje* vraća *true*, inače vraća *false*

```
def nastavi_dalje(resid_mom, tocnost):
    dalje=false
    for i in range(0,len(resid_mom)):
        if resid_mom[resid_mom.keys()[i]].abs()>tocnost:
            dalje=true

    return dalje
```

Postupak iteracije se izvršava u *while* petlji. Pozivaju se redom funkcije *odredi_cvor* koja određuje čvor čiji je iznos momenta maksimalan, zatim *_relax_steps* koji uravnotežuje dobiveni čvor i na kraju funkcija *nastavi_dalje* koji provjerava postoji li u novim dobivenim rezidualnim momentima neki koji je veći od unaprijed zadane točnosti. Ukoliko postoji vraćena vrijednost *true* te se *while* petlja nastavi izvoditi, u suprotnom varijabla *nastavi* je *false* i izračun je gotov.

```

nastavi=nastavi_dalje(residual_moments, tocnost)
while nastavi:
    cvor=odredi_cvor(residual_moments)
    _relax_step(cvor)
    nastavi=nastavi_dalje(residual_moments, tocnost)

```

3.3.2. Rješenje primjera prilagođenim programom

Izračunom dobivene razdjelne koeficijente upisala sam u varijablu *distr_f* strukture rječnik, i kojoj ključ označava element, a vrijednost je iznos razdjelnog koeficijenta, momente upetosti u *mom_fe*, a prijenosni koeficijent kao varijablu *co_f* te točnost do koje se radi izračun kao varijablu *točnost*.

```

# razdjelni koeficijenti:
distr_f = { (4, 0): 0.20, (4, 5): 0.80, (5, 4): 0.33, (5, 1): 0.17, (5, 6): 0.33, (5, 8): 0.17,
(6, 5): 0.40, (6, 2): 0.10, (6, 7): 0.40, (6, 9): 0.10, (7, 6): 0.67, (7, 3): 0.33, (8, 5): 0.33,
(8, 9): 0.67, (9, 8): 0.80, (9, 6): 0.20 }

# prijenosni koeficijent:
co_f = 1/2

# momenti upetosti:
mom_fe = { (0, 4): 40.0, (4, 0): -40.0, (2, 6): -40.0, (6, 2): 40.0, (4, 5): 33.75, (5, 4): -33.75,
(5, 6): 22.50, (6, 5): -22.50, (6, 7): 13.5, (7, 6): -13.5 }

#točnost
točnost=0

```

Pozivom *init_all* funkcije prikazuju se razdjelni koeficijenti, prijenosni koeficijenti, momenti upetosti te se izračunaju rezidualni momenti:

```
init_all (distr_f, co_f, mom_fe)
```

Crossov iterativni postupak izvršava se unutar while petlje, samo jednim pozivom izvršenja koda:

```

#while petlja računa residual sve dok nije zadovoljen uvjet da su rez. mom. u svim čvorovima < od zadane točnosti
nastavi=nastavi_dalje(residual_moments, tocnost)
while nastavi:
    cvor=odredi_cvor(residual_moments)
    _relax_step(cvor)
    nastavi=nastavi_dalje(residual_moments, tocnost)

```

Rezultat izvršavanja identičan je za svaku iteraciju kao i kod pojedinačnih poziva u izvornom programu:

Čvor 6:

distributed moments:

(6, 2): $0.10 * -31.0 = -3.1$
 (6, 5): $0.40 * -31.0 = -12.4$
 (6, 7): $0.40 * -31.0 = -12.4$
 (6, 9): $0.10 * -31.0 = -3.1$

carry-over moments:

(6, 2) -> (2, 6): $1/2 * -3.1 = -1.5$
 (6, 5) -> (5, 6): $1/2 * -12.4 = -6.2$
 (6, 7) -> (7, 6): $1/2 * -12.4 = -6.2$
 (6, 9) -> (9, 6): $1/2 * -3.1 = -1.6$

residual moments:

4: -6.3
 5: -17.5
 6: 0.00
 7: -19.7
 8: 0.00
 9: -1.6

Čvor 5:

distributed moments:

(5, 1): $0.17 * 17.5 = 3.0$
 (5, 4): $0.33 * 17.5 = 5.8$
 (5, 6): $0.33 * 17.5 = 5.8$
 (5, 8): $0.17 * 17.5 = 3.0$

carry-over moments:

(5, 1) -> (1, 5): $1/2 * 3.0 = 1.5$
 (5, 4) -> (4, 5): $1/2 * 5.8 = 2.9$
 (5, 6) -> (6, 5): $1/2 * 5.8 = 2.9$
 (5, 8) -> (8, 5): $1/2 * 3.0 = 1.5$

residual moments:

4: -3.4
 5: 0.00
 6: 9.5
 7: 0.00
 8: 1.5
 9: -1.6

Čvor 4:

distributed moments:

(4, 0): $0.20 * 3.4 = 0.70$
 (4, 5): $0.80 * 3.4 = 2.7$

carry-over moments:

(4, 0) -> (0, 4): $1/2 * 0.70 = 0.30$
 (4, 5) -> (5, 4): $1/2 * 2.7 = 1.4$

residual moments:

4: 0.00
 5: -0.50
 6: 0.00
 7: -1.9
 8: 1.5
 9: -2.1

Čvor 8:

distributed moments:

(8, 5): $0.33 * -2.3 = -0.80$
 (8, 9): $0.67 * -2.3 = -1.5$

carry-over moments:

(8, 5) -> (5, 8): $1/2 * -0.80 = -0.40$
 (8, 9) -> (9, 8): $1/2 * -1.5 = -0.80$

residual moments:

4: 0.00
 5: -0.90
 6: 0.20
 7: -1.9
 8: 0.00
 9: -0.80

Čvor 7:

distributed moments:

(7, 3): $0.33 * 19.7 = 6.5$
 (7, 6): $0.67 * 19.7 = 13.2$

carry-over moments:

(7, 3) -> (3, 7): $1/2 * 6.5 = 3.2$
 (7, 6) -> (6, 7): $1/2 * 13.2 = 6.6$

residual moments:

4: -6.3
 5: -17.5
 6: 6.6
 7: 0.00
 8: 0.00
 9: -1.6

Čvor 6:

distributed moments:

(6, 2): $0.10 * -9.5 = -1.0$
 (6, 5): $0.40 * -9.5 = -3.8$
 (6, 7): $0.40 * -9.5 = -3.8$
 (6, 9): $0.10 * -9.5 = -0.90$

carry-over moments:

(6, 2) -> (2, 6): $1/2 * -1.0 = -0.50$
 (6, 5) -> (5, 6): $1/2 * -3.8 = -1.9$
 (6, 7) -> (7, 6): $1/2 * -3.8 = -1.9$
 (6, 9) -> (9, 6): $1/2 * -0.90 = -0.50$

residual moments:

4: -3.4
 5: -1.9
 6: 0.00
 7: -1.9
 8: 1.5
 9: -2.1

Čvor 9:

distributed moments:

(9, 6): $0.20 * 2.1 = 0.40$
 (9, 8): $0.80 * 2.1 = 1.7$

carry-over moments:

(9, 6) -> (6, 9): $1/2 * 0.40 = 0.20$
 (9, 8) -> (8, 9): $1/2 * 1.7 = 0.80$

residual moments:

4: 0.00
 5: -0.50
 6: 0.20
 7: -1.9
 8: 2.3
 9: 0.00

Čvor 7:

distributed moments:

(7, 3): $0.33 * 1.9 = 0.60$
 (7, 6): $0.67 * 1.9 = 1.3$

carry-over moments:

(7, 3) -> (3, 7): $1/2 * 0.60 = 0.30$
 (7, 6) -> (6, 7): $1/2 * 1.3 = 0.60$

residual moments:

4: 0.00
 5: -0.90
 6: 0.80
 7: 0.00
 8: 0.00
 9: -0.80

Čvor 5:

distributed moments:

(5, 1): $0.17 * 0.90 = 0.20$
 (5, 4): $0.33 * 0.90 = 0.30$
 (5, 6): $0.33 * 0.90 = 0.30$
 (5, 8): $0.17 * 0.90 = 0.10$

carry-over moments:

(5, 1) -> (1, 5): $1/2 * 0.20 = 0.10$
 (5, 4) -> (4, 5): $1/2 * 0.30 = 0.20$
 (5, 6) -> (6, 5): $1/2 * 0.30 = 0.10$
 (5, 8) -> (8, 5): $1/2 * 0.10 = 0.00$

residual moments:

4: 0.20
 5: 0.00
 6: 0.90
 7: 0.00
 8: 0.00
 9: -0.80

Čvor 9:

distributed moments:

(9, 6): $0.20 * 0.80 = 0.20$
 (9, 8): $0.80 * 0.80 = 0.60$

carry-over moments:

(9, 6) -> (6, 9): $1/2 * 0.20 = 0.10$
 (9, 8) -> (8, 9): $1/2 * 0.60 = 0.30$

residual moments:

4: 0.20
 5: -0.20
 6: 0.10
 7: -0.20
 8: 0.30
 9: 0.00

Čvor 5:

distributed moments:

(5, 1): $0.17 * 0.30 = 0.00$
 (5, 4): $0.33 * 0.30 = 0.10$
 (5, 6): $0.33 * 0.30 = 0.10$
 (5, 8): $0.17 * 0.30 = 0.00$

carry-over moments:

(5, 1) -> (1, 5): $1/2 * 0.00 = 0.00$
 (5, 4) -> (4, 5): $1/2 * 0.10 = 0.00$
 (5, 6) -> (6, 5): $1/2 * 0.10 = 0.00$
 (5, 8) -> (8, 5): $1/2 * 0.00 = 0.00$

residual moments:

4: 0.20
 5: 0.00
 6: 0.10
 7: -0.20
 8: 0.00
 9: -0.10

Čvor 4:

distributed moments:

(4, 0): $0.20 * -0.20 = 0.00$
 (4, 5): $0.80 * -0.20 = -0.20$

carry-over moments:

(4, 0) -> (0, 4): $1/2 * 0.00 = 0.00$
 (4, 5) -> (5, 4): $1/2 * -0.20 = -0.10$

residual moments:

4: 0.00
 5: -0.10
 6: 0.10
 7: -0.20
 8: 0.00
 9: -0.10

Čvor 6:

distributed moments:

(6, 2): $0.10 * -0.90 = -0.10$
 (6, 5): $0.40 * -0.90 = -0.40$
 (6, 7): $0.40 * -0.90 = -0.40$
 (6, 9): $0.10 * -0.90 = -0.10$

carry-over moments:

(6, 2) -> (2, 6): $1/2 * -0.10 = -0.10$
 (6, 5) -> (5, 6): $1/2 * -0.40 = -0.20$
 (6, 7) -> (7, 6): $1/2 * -0.40 = -0.20$
 (6, 9) -> (9, 6): $1/2 * -0.10 = 0.00$

residual moments:

4: 0.20
 5: -0.20
 6: 0.00
 7: -0.20
 8: 0.00
 9: -0.80

Čvore 8:

distributed moments:

(8, 5): $0.33 * -0.30 = -0.10$
 (8, 9): $0.67 * -0.30 = -0.20$

carry-over moments:

(8, 5) -> (5, 8): $1/2 * -0.10 = -0.10$
 (8, 9) -> (9, 8): $1/2 * -0.20 = -0.10$

residual moments:

4: 0.20
 5: -0.30
 6: 0.10
 7: -0.20
 8: 0.00
 9: -0.10

Čvor 7:

distributed moments:

(7, 3): $0.33 * 0.20 = 0.10$
 (7, 6): $0.67 * 0.20 = 0.10$

carry-over moments:

(7, 3) -> (3, 7): $1/2 * 0.10 = 0.00$
 (7, 6) -> (6, 7): $1/2 * 0.10 = 0.00$

residual moments:

4: 0.00
 5: -0.10
 6: 0.10
 7: 0.00
 8: 0.00
 9: -0.10

Čvor 6:

distributed moments:

(6, 2): $0.10 * -0.10 = 0.00$
 (6, 5): $0.40 * -0.10 = 0.00$
 (6, 7): $0.40 * -0.10 = 0.00$
 (6, 9): $0.10 * -0.10 = 0.00$

carry-over moments:

(6, 2) -> (2, 6): $1/2 * 0.00 = 0.00$
 (6, 5) -> (5, 6): $1/2 * 0.00 = 0.00$
 (6, 7) -> (7, 6): $1/2 * 0.00 = 0.00$
 (6, 9) -> (9, 6): $1/2 * 0.00 = 0.00$

residual moments:

4: 0.00
 5: -0.10
 6: 0.00
 7: 0.00
 8: 0.00
 9: -0.10

Čvor 5:

distributed moments:

(5, 1): $0.17 * 0.10 = 0.00$
(5, 4): $0.33 * 0.10 = 0.00$
(5, 6): $0.33 * 0.10 = 0.00$
(5, 8): $0.17 * 0.10 = 0.00$

carry-over moments:

(5, 1) -> (1, 5): $1/2 * 0.00 = 0.00$
(5, 4) -> (4, 5): $1/2 * 0.00 = 0.00$
(5, 6) -> (6, 5): $1/2 * 0.00 = 0.00$
(5, 8) -> (8, 5): $1/2 * 0.00 = 0.00$

residual moments:

4: 0.00
5: 0.00
6: 0.00
7: 0.00
8: 0.00
9: -0.10

Čvor 9:

distributed moments:

(9, 6): $0.20 * 0.10 = 0.00$
(9, 8): $0.80 * 0.10 = 0.10$

carry-over moments:

(9, 6) -> (6, 9): $1/2 * 0.00 = 0.00$
(9, 8) -> (8, 9): $1/2 * 0.10 = 0.00$

residual moments:

4: 0.00
5: 0.00
6: 0.00
7: 0.00
8: 0.00
9: 0.00

Broj koraka je i ovim rješavanjem bio 18:

```
nsteps
    18
relax_order|
[6, 7, 5, 6, 4, 9, 8, 7, 5, 6, 9, 8, 5, 4, 7, 6, 5, 9]
print_dict (moment_lists)

(0, 4): [40.0, 0.30, 0.00]
(1, 5): [1.5, 0.10, 0.00, 0.00]
(2, 6): [-40.0, -1.5, -0.50, -0.10, 0.00]
(3, 7): [3.2, 0.30, 0.00]
(4, 0): [-40.0, 0.70, '|', 0.00, '|']
(4, 5): [33.7, 2.9, 2.7, '|', 0.20, 0.00, -0.20, '|', 0.00]
(5, 1): [3.0, '|', 0.20, '|', 0.00, '|', 0.00, '|']
(5, 4): [-33.8, 5.8, '|', 1.4, 0.30, '|', 0.10, '|', -0.10, 0.00, '|']
(5, 6): [22.5, -6.2, 5.8, '|', -1.9, 0.30, '|', -0.20, 0.10, '|', 0.00, 0.00, '|']
(5, 8): [3.0, '|', -0.40, 0.10, '|', -0.10, 0.00, '|', 0.00, '|']
(6, 2): [40.0, -3.1, '|', -1.0, '|', -0.10, '|', 0.00, '|']
(6, 5): [-22.5, -12.4, '|', 2.9, -3.8, '|', 0.10, -0.40, '|', 0.00, 0.00, '|', 0.00]
(6, 7): [13.5, -12.4, '|', 6.6, -3.8, '|', 0.60, -0.40, '|', 0.00, 0.00, '|']
(6, 9): [-3.1, '|', -0.90, '|', 0.20, -0.10, '|', 0.10, 0.00, '|', 0.00]
(7, 3): [6.5, '|', 0.60, '|', 0.10, '|']
(7, 6): [-13.5, -6.2, 13.2, '|', -1.9, 1.3, '|', -0.20, 0.10, '|', 0.00]
(8, 5): [1.5, -0.80, '|', 0.00, -0.10, '|', 0.00, 0.00]
(8, 9): [0.80, -1.5, '|', 0.30, -0.20, '|', 0.00]
(9, 6): [-1.6, -0.50, 0.40, '|', 0.00, 0.20, '|', 0.00, 0.00, '|']
(9, 8): [1.7, '|', -0.80, 0.60, '|', -0.10, 0.10, '|']

sum_end_moments()                                patch_moments()
final moments:                                    final residual moments:    final final moments:
(0, 4): 40.3                                     4: 0.00                    (0, 4): 40.3
(1, 5): 1.6                                       5: -0.10                   (1, 5): 1.6
(2, 6): -42.1                                    6: 0.00                    (2, 6): -42.1
(3, 7): 3.5                                       7: 0.00                    (3, 7): 3.5
(4, 0): -39.3                                    8: 0.00                    (4, 0): -39.3
(4, 5): 39.3                                     9: 0.00                    (4, 5): 39.3
(5, 1): 3.2                                       (5, 1): 3.2
(5, 4): -26.3                                    (5, 4): -26.2
(5, 6): 20.4                                    (5, 6): 20.4
(5, 8): 2.6                                       (5, 8): 2.6
(6, 2): 35.8                                    (6, 2): 35.8
(6, 5): -36.1                                    (6, 5): -36.1
(6, 7): 4.1                                       (6, 7): 4.1
(6, 9): -3.8                                    (6, 9): -3.8
(7, 3): 7.2                                       (7, 3): 7.2
(7, 6): -7.2                                    (7, 6): -7.2
(8, 5): 0.60                                    (8, 5): 0.60
(8, 9): -0.60                                    (8, 9): -0.60
(9, 6): -1.5                                    (9, 6): -1.5
(9, 8): 1.5                                       (9, 8): 1.5
```


3.4. Analiza problematike odabira čvorova

Uvijek se bira čvor u kojem je intenzitet neuravnoteženog momenta najveći jer u tom slučaju izračun najbrže konvergira.

Navedeno sam pokazala usporedbom potrebnih broja koraka u slučaju da se u svakom koraku bira čvor s maksimalnim rezidualnim momentom, zatim da se u svakom koraku bira čvor s minimalnim rezidualnom momentom te u slučaju slučajnog odabira čvorova.

Za potrebe testiranja dodala sam dvije nove funkcije, funkciju koja bira čvor s najmanjim rezidualnim momentom *odredi_min_cvor(resid_mom)* i funkciju koja bira čvor sa slučajnim odabirom *odredi_random_cvor(resid_mom)*.

```
def odredi_min_cvor(resid_mom):
    list_kljuceva=resid_mom.keys()
    min_mom=100000
    for i in range(0,len(resid_mom)):
        novi_cvor=resid_mom[list_kljuceva[i]].abs()
        if min_mom>novi_cvor and novi_cvor>0:
            kluc=list_kljuceva[i]
            min_mom=novi_cvor
    return kluc

#funkcija određuje random čvor
import random
def odredi_random_cvor(resid_mom):
    list_kljuceva=resid_mom.keys()
    kluc=random.randint(4,9)
    return kluc
```

U *while* petlji za uravnoteže sam pozvala prvo funkciju za odabir čvora s maksimalnim intenzitetom momenta, zatim funkciju s minimalnim intenzitetom momenta, te funkciju koja slučajno bira korake, koju sam pozivala 20 puta kako bi se vidio prosječan broj koraka u slučaju slučajnog odabira.

U opisanim slučajevima dobiven je sljedeći broj koraka:

Način odabira čvorova	Potreban broj koraka
Odabir čvora s maksimalnim rezidualnim momentom	18
Odabir čvora s minimalnim rezidualnim momentom	36
Slučajan odabir čvorova (rezultati dvadeset puta izvršavanja programa)	28, 28, 35, 30, 37, 35, 39, 34, 36, 21, 25, 46, 47, 29, 32, 28, 31, 35, 25, 27

Iz rezultata primjera je vidljivo da izračun Crossovim iterativnim postupkom uvijek relativno brzo konvergira, no najbrža konvergencija je u slučaju odabira čvorova momenta najvećeg intenziteta. U ovom primjeru prosječan broj potrebnih iteracija bio je 32. Izračun je bio duplo brži odabirom maksimalnih rezidualnih momenata u odnosu na odabir minimalnog rezidualnog momenta.

Za usporedbu sam isprobala ponašanje izračuna na primjeru sa 16 čvorova:

```
# razdjelni koeficijenti:
distr_f = { (7, 0): 0.33, (7, 8): 0.34, (7, 14): 0.33, (8, 7): 0.40, (8, 1): 0.10, (8, 9): 0.40, (8, 15): 0.10,
(9, 2): 0.20, (9, 8): 0.30, (9, 10): 0.20, (9, 16): 0.30, (10, 3): 0.33, (10, 9): 0.17, (10, 11): 0.33, (10, 17): 0.17,
(11, 4): 0.20, (11, 10): 0.30, (11, 12): 0.20, (11, 18): 0.30, (12, 5): 0.33, (12, 11): 0.17, (12, 13): 0.33, (12, 19): 0.17,
(13, 6): 0.20, (13, 12): 0.30, (13, 20): 0.50, (14, 7): 0.70, (14, 15): 0.30, (15, 8): 0.30, (15, 14): 0.30, (15, 16): 0.40,
(16, 9): 0.20, (16, 15): 0.30, (16, 17): 0.20, (16, 21): 0.30, (17, 10): 0.33, (17, 16): 0.17, (17, 18): 0.33, (17, 22): 0.17,
(18, 11): 0.40, (18, 17): 0.40, (18, 19): 0.20, (19, 12): 0.40, (19, 18): 0.30, (19, 20): 0.30, (20, 13): 0.30, (20, 19): 0.70,
(21, 16): 0.20, (21, 22): 0.80, (22, 17): 0.80, (22, 21): 0.20}

# prijenosni koeficijent:
co_f = 1/2

# momenti upetosti:
mom_fe = { (0, 7): 20.0, (7, 0): -20.0, (11,10): 13.75, (10, 11): -13.75, (7,14): 27.75, (14, 7): -27.75,
(14, 15): 50.5, (15, 14): -50.5, (22, 17): 10.5, (17, 22): -10.5, (11,18): 55.05, (18, 11): -55.05,
(16, 9): 32.17, (9, 16): -32.17, (22, 17): 10.5, (17, 22): -10.5, (13,20): 11.75, (20, 13): -11.75, (8,1): 33.73, (1, 8): -33.73}

#tocnost
tocnost=0
```

Način odabira čvorova	Potreban broj koraka
Odabir čvora s maksimalnim rezidualnim momentom	56
Odabir čvora s minimalnim rezidualnim momentom	198
Slučajan odabir čvorova (rezultati dvadeset puta izvršavanja programa)	177, 174, 175, 164, 190, 192, 189, 198, 175, 216, 173, 134, 170, 171, 180, 168, 173, 180, 196, 125

Kod većeg broja čvorova bilo je potrebno tri i pol puta manje koraka odabirom čvora s maksimalnim rezidualnim elementom u odnosu na slučajni odabir.

3.5. Zaključak

Crossov postupak je iterativni postupak kojim se na jednostavniji način određuju momenti savijanja na krajevima elemenata statičkog sistema. Zbog svoje jednostavnosti i iterativne prirode pogodan je za programsku realizaciju.

S obzirom na to da ručno određivanje momenata savijanja zahtjeva određeno vrijeme, naročito kod većih sistema, programsko rješavanje zadatka je smanjilo vrijeme rješavanja od nekoliko desetaka minuta na vrijeme do nekoliko milisekundi.

Popis literature

Fresl K.: Građevna statika 2., bilješke s predavanja dostupne na
<http://master.grad.hr/nastava/gS/gS2/gS2.pdf>

Program profesora Fresl K. GS_program_Relax (v0.4) dostupan na
<http://sage.grad.hr:1234/home/pub/62/>

Programski paket Sage <http://sage.grad.hr:1234/>