

SVEUČILIŠTE U ZAGREBU  
GRAĐEVINSKI FAKULTET

DOMAGOJ RUŽAK

**PROGRAMSKA REALIZACIJA  
STATIČKE KONDENZACIJE**  
(ZAVRŠNI RAD)

Zagreb, rujan 2008.

Ovaj rad izrađen je na Katedri za statiku, dinamiku i stabilnost konstrukcija Zavoda za tehničku mehaniku Građevinskog fakulteta Sveučilišta u Zagrebu, kao završni rad na preddiplomskom studiju, pod vodstvom mentora prof. dr. sc. Krešimira Fresla, dipl. ing. građ. Zahvaljujem se mentoru na velikoj pomoći koju mi je pružio, u svakom aspektu i tokom cijele izrade ovog rada. Dodatno, zahvaljujem se Vladimiru Beniću, dipl. ing. mat., što me naučio osnove programiranja u paketu *Wolfram Mathematica 6*.

# Sadržaj

<b>1. Uvod</b>	<b>4</b>
<b>2. Opća metoda pomaka</b>	<b>5</b>
2.1. Općenito o metodama pomaka i mehanička metafora . . . . .	5
2.2. Vrijednosti statičkih veličina u metodi pomaka . . . . .	6
2.3. Jednadžbe ravnoteže čvorova . . . . .	10
<b>3. Statička kondenzacija</b>	<b>12</b>
3.1. Teorijska podloga . . . . .	12
3.2. Kôd algoritma u paketu <i>Wolfram Mathematica 6</i> . . . . .	15
<b>4. Primjer</b>	<b>25</b>
<b>5. Zaključak</b>	<b>31</b>
<b>Literatura</b>	<b>32</b>
<b>Sažetak</b>	<b>33</b>

# 1. Uvod

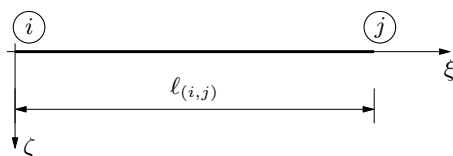
Već nekoliko stoljeća graditelji, pretežno iz ekonomskih razloga, proučavaju i primjenjuju statiku i statičke metode proračuna u procesu projektiranja građevina, radi osiguranja (tj. postizanja minimalnog rizika) stabilnosti konstrukcije. Pri tome su razvili mnoštvo metoda, koje se zasnivaju na različitim zaključcima i pretpostavkama, a koje se mogu, bolje ili lošije, primijeniti na različite konstrukcije. Odluka koju će metodu statičar primijeniti na zadanu konstrukciju ovisi o njenim parametrima – za početak je li konstrukcija statički određena ili nije, zatim kakav je oblik konstrukcije i sl. – te o točnosti koju želi postići. Većina tih metoda temelji se na inženjerskoj prosudbi i logičkom zaključivanju te je, "inteligentnim" smanjenjem broja nepoznanica u sustavima, prilagođena ručnom računanju. S druge strane, u današnje informatičko doba, korisno je "natjerati" računalo da rješava sustave jednadžbi umjesto vas, za što je pak potrebno poznavati metodu koju je razmjerno lako formalizirati i pretočiti u računalni algoritam. Jedna takva metoda je opća metoda pomaka. Ona je odgovarajuća kao osnova za programiranje statičkih proračuna i po još jednom vrlo važnom kriteriju – njome se jednako točno i lako (ako smijemo govoriti o lakoći kojom računalo rješava neki problem) rade proračuni statički određenih, kao i statički neodređenih konstrukcija – pa će ista biti i osnova ovog rada. U njemu će se iznijeti teorijska podloga (objašnjenja i svi potrebni izvodi) i računalni kôd za algoritam kojim računalo može napraviti statički proračun štapnih sistema sa zglobovima te jedan primjer na kojem će se pokazati da su programirane funkcije ispravne.

Pretpostavlja se osnovno znanje o općoj metodi pomaka, no kratki pregled, zajedno s matičnom formulacijom, dan je u točki 2. Izrazi za vrijednosti sila i momenata na krajevima elemenata, upetom i zglobnom, izvedeni su i prevedeni u matičnu formulaciju i računalni algoritam u točki 3. Cjelokupni računalni kôd pisan je u programskom paketu *Wolfram Mathematica 6* te je u posljednjoj točki rada napisan primjer proračuna pomoću navedenog programa.

## 2. Opća metoda pomaka

### 2.1. Općenito o metodama pomaka i mehanička metafora

Metode pomaka su metode proračuna štapnih sistema u kojima su nepoznanice vrijednosti translacijskih pomaka i kutovi zaokreta odabranih točaka sistema nazvanih čvorovima. Čvorovi su sve, na neki način, istaknute točke sistema, poput ležajeva, točaka u kojima se sastaje više štapova, zglobova i poluzglobova, slobodnog kraja prepusta i sl. Štapnim elementom ili linijskim elementom nazivamo dio štapa između dva čvora. Štapni element priključen u čvorove  $i$  i  $j$  označavat ćemo sa  $(i, j)$ . Lokalni koordinatni sustav odabran je tako da čvor  $i$  leži u njegovu ishodištu i da se uzdužna os štapa poklapa s osi  $\xi$  (slika 1.), dok je cijeli ravninski sistem postavljen u ravninu  $xz$  globalnog koordinatnog sustava. Svaki kruti ravninski čvor (ako nije ležajni) ima dva translacijska stupnja slobode (pomaci u smjeru osi  $x$  i  $z$ ) te jedan rotacijski stupanj slobode (zaokret za kut  $\varphi$  oko osi okomite na ravninu sistema). Statičke veličine na  $i$ -tom kraju elementa označavat ćemo parom indeksa  $i, j$ , a veličine na  $j$ -tom kraju parom  $j, i$ .



Slika 1.

Proračun metodom pomaka provodi se na osnovnom sistemu, koji se oblikuje dodavanjem zamišljenih veza koje sprječavaju translacijske pomake i zaokrete čvorova. Pri zadanom opterećenju će se u dodanim vezama pojaviti reaktivne sile i momenti kojih u izvornom sistemu nema. Tada će se sustav naći u jednom od beskonačno mnogo mogućih polja pomaka, a u stvarno stanje pomaka izvornog sistema dovest ćemo ga tako da stanju spriječenih pomaka dodamo stanje prisilnih pomaka čvorova, pri čemu ti pomaci i zaokreti moraju biti takvi da reakcije, koje se zbog njih razvijaju u zamišljenim vezama, ponište reakcije izazvane zadanim djelovanjima. Kada te reakcije iščeznu, na čvorove osnovnog sistema djelovat će samo one sile koje djeluju na čvorove izvornog sistema, pa će, budući da su u izvornom sistemu te sile u ravnoteži, u ravnoteži biti i sile u osnovnom sistemu. Slijedi da uvjete iščezavanja reakcija u zamišljenim vezama možemo izraziti kao uvjete ravnoteže sila i momenata u čvorovima, što znači da tražene vrijednosti pomaka čvorova i kuteve njihova zaokreta dobivamo rješavanjem sustava jednadžbi ravnoteže.

Poopćene sile kojima čvorovi djeluju na element nazivaju se silama na krajevima štapnog elementa. Za razliku od unutarnjih sila, one su na oba kraja štapnog elementa dogovorno pozitivne ako se njihov smisao djelovanja poklapa s pozitivnim smislom odgovarajuće osi (slika 2.).

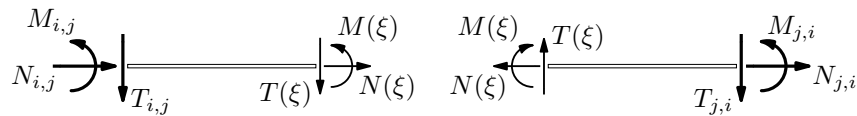
Svaka sila na kraju elementa predstavlja zbroj sila u stanju spriječenih pomaka čvorova i sila u stanju prisilnih pomaka. Tako možemo napisati:

$$N_{i,j} = n_{i,j} + \bar{N}_{i,j},$$

$$T_{i,j} = t_{i,j} + \bar{T}_{i,j},$$

$$M_{i,j} = m_{i,j} + \bar{M}_{i,j},$$

gdje su  $n_{i,j}$ ,  $t_{i,j}$  i  $m_{i,j}$  vrijednosti sila u stanju prisilnih, a  $\bar{N}_{i,j}$ ,  $\bar{T}_{i,j}$  i  $\bar{M}_{i,j}$  vrijednosti sila u stanju spriječenih pomaka. Analogno se, naravno, mogu napisati i izrazi za vrijednosti sila na kraju  $j$ .



Slika 2.

## 2.2. Vrijednosti statičkih veličina u metodi pomaka

Opterećenja na štapnim elementima ulaze u jednadžbe ravnoteže u obliku poopćenih sila upetosti (sila stanja spriječenih pomaka):

$$\bar{\mathbf{f}}_{(i,j)} = [ \bar{N}_{i,j} \quad \bar{T}_{i,j} \quad \bar{M}_{i,j} \quad \bar{N}_{j,i} \quad \bar{T}_{j,i} \quad \bar{M}_{j,i} ]^T.$$

Izrazi za njihovo izračunavanje izvode se metodom sila, a za mnoge tipove opterećenja već se mogu naći u raznim priručnicima iz građevinske statike.

Sile u stanju prisilnih pomaka čvorova posljedica su poopćenih pomaka krajeva elementa koji su jednaki pomacima čvorova sistema. To su prisilni pomaci i kutevi zaokreta ležajeva obostrano upete grede:  $u_{i,j}$ ,  $w_{i,j}$  i  $\varphi_{i,j}$  te  $u_{j,i}$ ,  $w_{j,i}$  i  $\varphi_{j,i}$ . Pretpostavimo li da su  $I(x) = I_{(i,j)}$ ,  $A(x) = A_{(i,j)}$  i  $E(x) = E_{(i,j)}$ , možemo definirati aksijalnu i fleksijsku krutost elementa  $(i, j)$ :

$$k_{(i,j)}^a = \frac{E_{(i,j)} A_{(i,j)}}{l_{i,j}} \quad i \quad k_{(i,j)}^f = \frac{E_{(i,j)} I_{(i,j)}}{l_{i,j}}.$$

Izraze za vrijednosti sila na krajevima štapnog elementa kao funkcija vrijednosti pooptćenih pomaka izvest ćemo metodom sila. Ako za osnovni sistem odaberemo lijevo upetu konzolu (slika 3.), matrica fleksibilnosti je

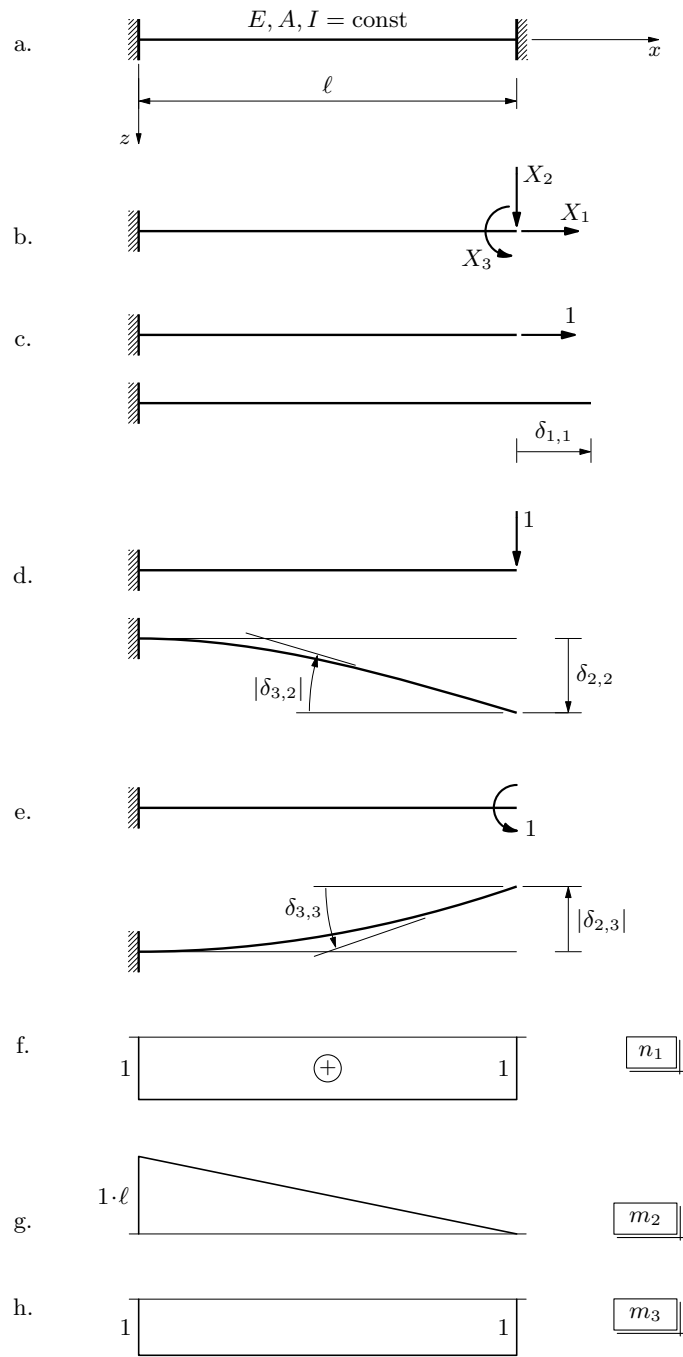
$$\mathbf{D} = \begin{bmatrix} k_{(i,j)}^a & 0 & 0 \\ 0 & \frac{l_{(i,j)}^2}{3k_{(i,j)}^f} & -\frac{l_{(i,j)}}{2k_{(i,j)}^f} \\ 0 & -\frac{l_{(i,j)}}{2k_{(i,j)}^f} & \frac{1}{k_{(i,j)}^f} \end{bmatrix},$$

pa sustav jednadžbi neprekinutosti glasi

$$\begin{bmatrix} k_{(i,j)}^a & 0 & 0 \\ 0 & \frac{l_{(i,j)}^2}{3k_{(i,j)}^f} & -\frac{l_{(i,j)}}{2k_{(i,j)}^f} \\ 0 & -\frac{l_{(i,j)}}{2k_{(i,j)}^f} & \frac{1}{k_{(i,j)}^f} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \end{bmatrix} + \begin{bmatrix} u_{i,j} \\ w_{i,j} - \varphi_{i,j} l_{(i,j)} \\ \varphi_{i,j} \end{bmatrix} = \begin{bmatrix} u_{j,i} \\ w_{j,i} \\ \varphi_{j,i} \end{bmatrix}.$$

Osnovni smo sistem odabrali tako da sile  $X_1$ ,  $X_2$  i  $X_3$  odgovaraju redom silama  $n_{j,i}$ ,  $t_{j,i}$  i  $m_{j,i}$  na kraju  $j$  štapnog elementa. Iz jednadžbi ravnoteže štapa sada možemo izvesti izraze za vrijednosti sila na kraju  $i$ :

$$\begin{aligned} n_{i,j} &= -n_{j,i}, \\ t_{i,j} &= -t_{j,i}, \\ m_{i,j} &= -m_{j,i} + l_{(i,j)} \cdot t_{j,i}. \end{aligned}$$



Slika 3.



Rješenjem sustava dobivamo izraze za sile na kraju elementa, koje možemo zapisati matrično:

$$\begin{bmatrix} n_{i,j} \\ t_{i,j} \\ m_{i,j} \\ n_{j,i} \\ t_{j,i} \\ m_{j,i} \end{bmatrix} = \begin{bmatrix} k_{(i,j)}^a & 0 & 0 & -k_{(i,j)}^a & 0 & 0 \\ 0 & \frac{12k_{(i,j)}^f}{l_{(i,j)}^2} & -\frac{6k_{(i,j)}^f}{l_{(i,j)}} & 0 & -\frac{12k_{(i,j)}^f}{l_{(i,j)}^2} & -\frac{6k_{(i,j)}^f}{l_{(i,j)}} \\ 0 & -\frac{6k_{(i,j)}^f}{l_{(i,j)}} & 4k_{(i,j)}^f & 0 & \frac{6k_{(i,j)}^f}{l_{(i,j)}} & 2k_{(i,j)}^f \\ -k_{(i,j)}^a & 0 & 0 & k_{(i,j)}^a & 0 & 0 \\ 0 & -\frac{12k_{(i,j)}^f}{l_{(i,j)}^2} & \frac{6k_{(i,j)}^f}{l_{(i,j)}} & 0 & \frac{12k_{(i,j)}^f}{l_{(i,j)}^2} & \frac{6k_{(i,j)}^f}{l_{(i,j)}} \\ 0 & -\frac{6k_{(i,j)}^f}{l_{(i,j)}} & 2k_{(i,j)}^f & 0 & \frac{6k_{(i,j)}^f}{l_{(i,j)}} & 4k_{(i,j)}^f \end{bmatrix} \begin{bmatrix} u_{i,j} \\ w_{i,j} \\ \varphi_{i,j} \\ u_{j,i} \\ w_{j,i} \\ \varphi_{j,i} \end{bmatrix},$$

ili kraće, matričnom simbolikom:

$$\mathbf{f}_{(i,j)} = \mathbf{k}_{(i,j)} \mathbf{u}_{(i,j)}.$$

Matrica  $\mathbf{k}_{(i,j)}$  je matrica krutosti elementa izražena u lokalnom koordinatnom sustavu, vektor  $\mathbf{u}_{(i,j)}$  je vektor vrijednosti popučenih pomaka krajeva elementa u smjerovima osi lokalnog koordinatnog sustava, a vektor  $\mathbf{f}_{(i,j)}$  je vektor vrijednosti sila na krajevima štapnog elementa izazvanih pomacima njegovih krajeva.

Za postavljanje jednadžbi ravnoteže čvorova, korisno je taj izraz napisati u obliku

$$\begin{bmatrix} \mathbf{f}_{(i,j)_i} \\ \mathbf{f}_{(i,j)_j} \end{bmatrix} = \begin{bmatrix} \mathbf{k}_{(i,j)_{i,i}} & \mathbf{k}_{(i,j)_{i,j}} \\ \mathbf{k}_{(i,j)_{j,i}} & \mathbf{k}_{(i,j)_{j,j}} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{(i,j)_i} \\ \mathbf{u}_{(i,j)_j} \end{bmatrix}.$$

Prijelaz iz lokalnog u globalni koordinatni sustav i obratno vršit ćemo pomoću matrice

$$\mathbf{R}_{(i,j)} = \begin{bmatrix} \mathbf{r}_{(i,j)} & \mathbf{0} \\ \mathbf{0} & \mathbf{r}_{(i,j)} \end{bmatrix},$$

gdje je  $\mathbf{r}_{(i,j)}$  matrica transformacije koordinata iz globalnog u lokalni koordinatni sustav:

$$\mathbf{r}_{(i,j)} = \begin{bmatrix} \cos \alpha_{(i,j)} & -\sin \alpha_{(i,j)} & 0 \\ \sin \alpha_{(i,j)} & \cos \alpha_{(i,j)} & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Sada su

$$\begin{aligned} \mathbf{f}_{(i,j)} &= \mathbf{R}_{(i,j)} \mathbf{f}_{(i,j)}^g, & \mathbf{f}_{(i,j)}^g &= \mathbf{R}_{(i,j)}^{-1} \mathbf{f}_{(i,j)}, \\ \bar{\mathbf{f}}_{(i,j)} &= \mathbf{R}_{(i,j)} \bar{\mathbf{f}}_{(i,j)}^g, & \bar{\mathbf{f}}_{(i,j)}^g &= \mathbf{R}_{(i,j)}^{-1} \bar{\mathbf{f}}_{(i,j)}, \\ \mathbf{u}_{(i,j)} &= \mathbf{R}_{(i,j)} \mathbf{u}_{(i,j)}^g, & \mathbf{u}_{(i,j)}^g &= \mathbf{R}_{(i,j)}^{-1} \mathbf{u}_{(i,j)}. \end{aligned}$$

### 2.3. Jednadžbe ravnoteže čvorova

Na svaki čvor  $i$  sistema djeluju sile od priključenih elemenata, a mogu djelovati i zadane ili reaktivne koncentrirane sile i momenti koje ćemo svrstati u vektor

$$\mathbf{p}_i = \begin{bmatrix} P_i^x & P_i^z & M_i \end{bmatrix}^T.$$

Vektor je ukupnih vrijednosti sila na  $i$ -tom kraju štapnog elementa  $(i, j)$

$$\hat{\mathbf{f}}_{(i,j)_i}^g = \begin{bmatrix} \mathbf{k}_{(i,j)_i,i}^g & \mathbf{k}_{(i,j)_i,j}^g \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_j \end{bmatrix} + \bar{\mathbf{f}}_{(i,j)_i}^g.$$

Kraj  $i$  elementa  $(i, j)$  djeluje na čvor  $i$  suprotno orijentiranim silama čije su vrijednosti komponente vektora  $-\hat{\mathbf{f}}_{(i,j)_i}^g$ , pa je formalni zapis uvjeta ravnoteže čvora  $i$

$$-\sum_{(i,e_i)} \hat{\mathbf{f}}_{(i,e_i)_i}^g + \mathbf{p}_i = \mathbf{0},$$

gdje je sumacija po svim elementima  $(i, e_i)$  koji su priključeni u taj čvor. Nakon uvrštavanja i sređivanja dobivamo:

$$\sum_{(i,e_i)} \left( \begin{bmatrix} \mathbf{k}_{(i,e_i)_i,i}^g & \mathbf{k}_{(i,e_i)_i,e_i}^g \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_{e_i} \end{bmatrix} \right) = -\sum_{(i,e_i)} \bar{\mathbf{f}}_{(i,e_i)_i}^g + \mathbf{p}_i.$$

Napišu li se ti izrazi za sve čvorove sistema, dobiva se sustav jednažbi (u matričnom zapisu)

$$\mathbf{K}\mathbf{u} = \mathbf{q}.$$

Vektori  $\mathbf{u}$  i  $\mathbf{q}$  imaju po  $3n$  komponenti, gdje je  $n$  broj čvorova sistema:

$$\mathbf{u} = [ \mathbf{u}_1 \quad \mathbf{u}_2 \quad \cdots \quad \mathbf{u}_n ]^T, \quad \mathbf{q} = [ \mathbf{q}_1 \quad \mathbf{q}_2 \quad \cdots \quad \mathbf{q}_n ]^T,$$

pri čemu je

$$\mathbf{q}_i = - \sum_{(i,e_i)} \bar{\mathbf{f}}_{(i,e_i)i}^g + \mathbf{p}_i.$$

Matrica  $\mathbf{K}$  sastavljena je od  $n \times n$  blokova tipa  $3 \times 3$ . Dijagonalni su blokovi

$$\mathbf{K}_{i,i} = \sum_{(i,e_i)} \mathbf{k}_{(i,e_i)i,i}^g.$$

Ako su čvorovi  $i$  i  $j$  povezani elementom  $(i, j)$ , tada će vandijagonalni blokovi  $i, j$  i  $j, i$  biti

$$\mathbf{K}_{i,j} = \mathbf{k}_{(i,j)i,j}^g \quad i \quad \mathbf{K}_{j,i} = \mathbf{k}_{(i,j)j,i}^g;$$

nisu li čvorovi povezani, ti će blokovi biti  $3 \times 3$  nul-matrice.

Rješavanjem ovog sustava jednažbi, dobivaju se tražene vrijednosti komponenta pomaka čvorova u smjerovima osi globalnog koordinatnog sustava. Potrebno je još naći njihove vrijednosti u smjerovima osi lokalnog koordinatnog sustava elemenata te iz njih, direktno, vrijednosti konačnih sila na krajevima elemenata.

### 3. Statička kondenzacija

#### 3.1. Teorijska podloga

U realnom sistemu štapovi mogu biti spojeni sa čvorovima i drugim vezama osim upetog spoja. Ovdje će se obraditi samo slučaj zglobne veze.

Kada govorimo o statičkoj kondenzaciji elementa štapnog sistema, moramo razlikovati lokalnu i globalnu statičku kondenzaciju. Lokalna se odnosi na kondenzaciju izraza za vrijednosti sila na krajevima elemenata, dok je kod globalne riječ o kondenziranju globalne matrice krutosti.



Slika 4.

**Lokalna kondenzacija.** Ako je element  $(i, j)$  sa čvorom  $i$  spojen zglobno,  $i$ -ti kraj elementa i čvor  $i$  mogu se neovisno zakretati (slika 4. lijevo), pa zaokret kraja neće biti jednak zaokretu čvora, tj.  $\varphi_{(i,j)} \neq \varphi_i$ . Međutim, znamo da je na  $i$ -tom kraju elementa  $M_{i,j} = 0$ , pa je, za jednostrano upetu gredu (slika 4. desno),

$$M_{i,j} = \frac{6k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) + 4k_{(i,j)}^f \varphi_{i,j} + 2k_{(i,j)}^f \varphi_{j,i} + \overline{M}_{i,j} = 0.$$

Sada je

$$\varphi_{i,j} = -\frac{1}{4k_{(i,j)}^f} \left( \frac{6k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) + 2k_{(i,j)}^f \varphi_{j,i} + \overline{M}_{i,j} \right).$$

Uvrstimo li to u izraz za  $M_{j,i}$ ,

$$M_{j,i} = -\frac{6k_{(i,j)}^f}{l_{(i,j)}} w_{i,j} + 2k_{(i,j)}^f \varphi_{i,j} + \frac{6k_{(i,j)}^f}{l_{(i,j)}} w_{j,i} + 4k_{(i,j)}^f \varphi_{j,i} + \overline{M}_{j,i},$$

dobivamo

$$M_{j,i} = \frac{6k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) - \frac{2k_{(i,j)}^f}{4k_{(i,j)}^f} \left( \frac{6k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) + 2k_{(i,j)}^f \varphi_{j,i} + \overline{M}_{i,j} \right) + 4k_{(i,j)}^f \varphi_{j,i} + \overline{M}_{j,i},$$

$$M_{j,i} = \frac{6k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) - \frac{3k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) - k_{(i,j)}^f \varphi_{j,i} - \frac{1}{2} \overline{M}_{i,j} + 4k_{(i,j)}^f \varphi_{j,i} + \overline{M}_{j,i},$$

odnosno, konačno,

$$M_{j,i} = \frac{3k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) + 3k_{(i,j)}^f \varphi_{j,i} + \left( \bar{M}_{j,i} - \frac{1}{2} \bar{M}_{i,j} \right).$$

Analogno, treba kondenzirati i poprečnu silu za kraj  $i$ :

$$\begin{aligned} M_{i,j} = 0 &= \frac{6k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) + 4k_{(i,j)}^f \varphi_{i,j} + 2k_{(i,j)}^f \varphi_{j,i} + \bar{M}_{i,j} \quad / \quad \times \frac{6}{4l_{(i,j)}} \\ 0 &= \left. \begin{aligned} &\frac{9k_{(i,j)}^f}{l_{(i,j)}^2} (-w_{i,j} + w_{j,i}) + \frac{6k_{(i,j)}^f}{l_{(i,j)}} \varphi_{i,j} + \frac{3k_{(i,j)}^f}{l_{(i,j)}} \varphi_{j,i} + \frac{3}{2l_{(i,j)}} \bar{M}_{i,j} \\ &T_{i,j} = \frac{12k_{(i,j)}^f}{l_{(i,j)}^2} (w_{i,j} - w_{j,i}) - \frac{6k_{(i,j)}^f}{l_{(i,j)}} \varphi_{i,j} - \frac{6k_{(i,j)}^f}{l_{(i,j)}} \varphi_{j,i} + \bar{T}_{i,j} \end{aligned} \right\} (+) \\ T_{i,j} &= \frac{3k_{(i,j)}^f}{l_{(i,j)}^2} (w_{i,j} - w_{j,i}) - \frac{3k_{(i,j)}^f}{l_{(i,j)}} \varphi_{j,i} + \left( \bar{T}_{i,j} + \frac{3}{2l_{(i,j)}} \bar{M}_{i,j} \right) \end{aligned}$$

i za kraj  $j$ :

$$\begin{aligned} M_{i,j} = 0 &= \frac{6k_{(i,j)}^f}{l_{(i,j)}} (-w_{i,j} + w_{j,i}) + 4k_{(i,j)}^f \varphi_{i,j} + 2k_{(i,j)}^f \varphi_{j,i} + \bar{M}_{i,j} \quad / \quad \times \left( -\frac{6}{4l_{(i,j)}} \right) \\ 0 &= \left. \begin{aligned} &-\frac{9k_{(i,j)}^f}{l_{(i,j)}^2} (-w_{i,j} + w_{j,i}) - \frac{6k_{(i,j)}^f}{l_{(i,j)}} \varphi_{i,j} - \frac{3k_{(i,j)}^f}{l_{(i,j)}} \varphi_{j,i} - \frac{3}{2l_{(i,j)}} \bar{M}_{i,j} \\ &T_{j,i} = \frac{12k_{(i,j)}^f}{l_{(i,j)}^2} (-w_{i,j} + w_{j,i}) + \frac{6k_{(i,j)}^f}{l_{(i,j)}} \varphi_{i,j} + \frac{6k_{(i,j)}^f}{l_{(i,j)}} \varphi_{j,i} + \bar{T}_{j,i} \end{aligned} \right\} (+) \\ T_{j,i} &= \frac{3k_{(i,j)}^f}{l_{(i,j)}^2} (-w_{i,j} + w_{j,i}) + \frac{3k_{(i,j)}^f}{l_{(i,j)}} \varphi_{j,i} + \left( \bar{T}_{j,i} - \frac{3}{2l_{(i,j)}} \bar{M}_{i,j} \right). \end{aligned}$$

Vrijednosti uzdužnih sila ostaju iste kao kod obostrano upete grede.

Sada se može napisati kondenzirana matrica krutosti jednostrano upete grede sa zglo-  
bom na kraju  $i$ :

$$\mathbf{k}_{(i,j)}^c = \begin{bmatrix} k_{(i,j)}^a & 0 & 0 & -k_{(i,j)}^a & 0 & 0 \\ 0 & \frac{3k_{(i,j)}^f}{l_{(i,j)}^2} & 0 & 0 & -\frac{3k_{(i,j)}^f}{l_{(i,j)}^2} & -\frac{3k_{(i,j)}^f}{l_{(i,j)}} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -k_{(i,j)}^a & 0 & 0 & k_{(i,j)}^a & 0 & 0 \\ 0 & -\frac{3k_{(i,j)}^f}{l_{(i,j)}^2} & 0 & 0 & \frac{3k_{(i,j)}^f}{l_{(i,j)}^2} & \frac{3k_{(i,j)}^f}{l_{(i,j)}} \\ 0 & -\frac{3k_{(i,j)}^f}{l_{(i,j)}} & 0 & 0 & \frac{3k_{(i,j)}^f}{l_{(i,j)}} & 3k_{(i,j)}^f \end{bmatrix}$$

i pripadajući joj vektor sila upetosti:

$$\bar{\mathbf{f}}_{(i,j)}^c = \begin{bmatrix} \bar{N}_{i,j} \\ \bar{T}_{i,j} + \frac{3}{2l_{(i,j)}}\bar{M}_{i,j} \\ 0 \\ \bar{N}_{j,i} \\ \bar{T}_{j,i} - \frac{3}{2l_{(i,j)}}\bar{M}_{i,j} \\ \bar{M}_{j,i} - \frac{1}{2}\bar{M}_{i,j} \end{bmatrix}.$$

Analogno se izvode i izrazi za jednostrano upetu gredu sa zglobo-  
m na kraju  $j$  te za prave štapne elemente (sa zglobo-  
m na oba kraja), pa se oni ovdje neće posebno izvoditi.  
U računalni kôd, naravno, uvrštena su sva tri slučaja.

**Globalna kondenzacija.** Komponente kondenzirane matrice krutosti elementa u retku i stupcu koji odgovaraju oslobođenom pomaku jednake su nuli. To znači da se kondenzacijom rang matrice umanjuje barem za broj raskinutih veza. Ako su svi elementi u neki čvor priključeni zglobno, globalna će matrica krutosti imati redak i stupac ispunjen nulama, pa će biti singularna. Stoga u programu još treba ukloniti taj redak i taj stupac iz matrice, što se naziva kondenzacijom globalne matrice krutosti.

S obzirom da zglob može nastati na dva načina – ili tako da se čvor eksplicitno proglasi zglobom, ili tako da se oslobode krajevi svih elemenata priključenih u čvor – program će morati provjeravati jesu li u neki od čvorova, koji nisu eksplicitno zadani kao zglobovi, svi elementi priključeni zglobno te takav čvor samostalno proglasiti punim zglobom.

### 3.2. Kôd algoritma u paketu *Wolfram Mathematica 6*

Jezgra budućeg programa za proračun ravninskih štapnih konstrukcija već je napisana u programskom paketu *Wolfram Mathematica 6*. Njzinu osnovu čine funkcije za proračun sustava sastavljenog od isključivo obostrano upetih greda, koje se ovdje neće iznositi, osim u dijelu u kojem su promijenjene, u skladu s teorijskom podlogom ovog rada. Funkcije se iznose prema redoslijedu izvođenja u programu.

Na početku programa upisuju se čvorovi – zadani koordinatama u globalnom koordinatnom sustavu – i elementi – zadani indeksima čvorova u koje su priključeni te indeksima skupina geometrijskih karakteristika. Na taj način program odmah "zna" u koje je čvorove priključen neki određeni element, no "ne zna" prepoznati koji su sve elementi priključeni u zadani čvor. Ta lista će nam trebati kod prepoznavanja punih zglobova, a možemo je generirati već ovdje. Funkcija koja to radi izgleda ovako:

```
TblElemsByNodes [n_Integer, el_List] :=
Module[
  { i, ne, nel },
  ne = Table [1, {n}];
  nel = Length [el];
  Do[
    ne[[i]] = {i},
    {i, 1, n}
  ];
  Do[
    AppendTo [ ne[[ el[[i]][[1]][[1]] ]], el[[i]][[1]] ];
    AppendTo [ ne[[ el[[i]][[1]][[2]] ]], el[[i]][[1]] ],
    {i, 1, nel}
  ];
  ne
]
```

Parametri su ove funkcije broj čvorova  $n$  i lista elemenata  $el$ , a rezultat lista listi u kojima je prvi element indeks čvora, a svi sljedeći oznake elemenata priključenih u taj čvor.

Ovdje valja napomenuti da je "oznaka" elementa lista s dva člana, od kojih svaki odgovara indeksu jednog od čvorova u koje je element priključen.

Nakon upisa geometrijskih karakteristika i ležajnih veza, slijedi upis punih zglobova – jednostavno kao liste indeksa čvorova – te otpuštenih veza na krajevima elemenata. Potonja lista ima članova koliko ima otpuštenih veza u cijelom sustavu, a svaki se član sastoji od tri dijela – dvočlane oznake elementa, indeksa čvora u kojem je veza oslobođena te liste koja govori koja veza je oslobođena (iako ti slučajevi zasad nisu obuhvaćeni u programu, predviđena je mogućnost otpuštanja i veza koje sprječavaju translacijske pomake – zasad se ograničavamo samo na zglobove). S tim podacima možemo generirati listu čvorova s pridruženim oznakama elemenata koji su u njih spojeni zglobno (zanemarujući zasad čvorove koji su direktno upisani kao zglobovi):

```
TblHngEnd [er_List, nd_List] :=
Module[
  { i, j, nr, rm, rmt, nn },
  nr = Length [er];
  rm = Table [1, {0}];
  Do[
    If[ er[[i]][[3]][[3]] == 1, AppendTo [rm, {er[[i]][[2]], er[[i]][[1]]} ]],
    {i, 1, nr}
  ];
  nn = Length [nd];
  rmt = Table [1, {nn}, {0}];
  Do[
    rmt[[i]] = {i},
    {i, 1, nn}
  ];
  Do[
    If[ i == rm[[j]][[1]], AppendTo [rmt[[i]], rm[[j]][[2]] ]],
    {i, 1, nn}, {j, 1, Length[rm]}
  ];
  rmt
]
```

*er* je lista otpuštenih veza na krajevima elemenata, a *nd* lista čvorova sistema. Program prepoznaje sve elemente koji imaju bar na jednom kraju zglobnu vezu te ih odmah pridružuje odgovarajućem čvoru. Izlazna lista ustrojena je tako da svaki element odgovara jednom čvoru sistema, a u njemu je indeks tog čvora iza kojeg slijedi popis elemenata koji su u taj čvor spojeni zglobno.



Konačan popis svih punih zglobova daje sljedeća funkcija:

```
ListFHngs [nd_List, el_List, hns_List, cns_List, er_List] :=
Module[
  { nn, nc, hs, ne, he, nel, fh },
  nn = Length [nd];
  nc = Length [cns];
  hs = Table [1, {nn}];
  Do[
    If [ i == cns[[j]][[1]], hs[[i]] = 0 ],
    {i, 1, nn}, {j, 1, nc}
  ];
  ne = TblElemsByNodes [nn, el];
  he = TblHngEnd [er, nd];
  Do[
    If [ Length [ ne[[i]] ] != Length [ he[[i]] ], hs[[i]] = 0],
    {i, 1, nn}
  ];
  fh = Table [1, {0}];
  Do[
    AppendTo [fh, hns[[i]]],
    {i, 1, Length[hns]}
  ];
  Do[
    If [ hs[[i]] == 1, AppendTo [fh, i] ],
    {i, 1, nn}
  ];
  fh
]
```

Ova funkcija kao parametre ima listu čvorova, listu elemenata, listu čvorova direktno upisanih kao zglobovi, listu ležajnih uvjeta i listu oslobođenih veza na krajevima elemenata, dok unutar sebe još dodatno poziva i dvije prethodno opisane funkcije. Rezultat je jednostavna lista indeksa punih zglobova.

Sada možemo generirati konačnu listu čvorova u koje su zglobno spojeni neki ili svi elementi (uključujući pune zglobove i oslobođene krajeve elemenata):

```
ListFERls [er_List, nd_List, el_List, hns_List, cns_List] :=
Module[
  { n, fer, i, j, fh, ne },
  n = Length [nd];
  fer = TblHngEnd [er, nd];
  fh = ListFHngs [nd, el, hns, cns, er];
  ne = TblElemsByNodes [n, el];
  Do[
    If [ fer[[i]][[1]] == fh[[j]], fer[[i]] = ne[[i]] ],
    {i, 1, n}, {j, 1, Length [fh]}
  ];
  fer
]
```

Parametri su identični onima u prethodnoj funkciji, dok je *output* ovdje lista indeksa čvorova uz koje su popisi svih elemenata koji u tom čvoru imaju oslobođen kraj.

Dosad napravljeno je dovoljno za kondenzaciju globalne matrice krutosti, no program još uvijek "ne zna" koji sve elementi i na kojim krajevima imaju zglobnu vezu. To se rješava sljedećim dvjema funkcijama:

```
ListElemsHngs [el_List, lf_List] :=
Module[
  { nel, eh, nl, br, i, j, k },
  nel = Length [el];
  eh = Table[0, {nel}, {3}];
  Do[
    eh[[i]] = {el[[i]][[1]]},
    {i, 1, nel}
  ];
  nl = Length [lf];
  br = 0;
  Do[
    br = Length [ lf[[i]] ];
    Do[
      If [ lf[[i]][[j]] == eh[[k]][[1]], AppendTo [ eh[[k]], lf [[i]][[1]] ]],
      {j, 1, br}
    ],
    {i, 1, nl}, {k, 1, nel}
  ];
  Do[
    If [ Length [ eh[[i]] ] == 1, AppendTo [eh[[i]], 0] ];
    If [ Length [ eh[[i]] ] == 2, AppendTo [eh[[i]], 0] ],
    {i, 1, nel}
  ];
  eh ]

ListElemsHngsBin [el_List, lf_List] :=
Module[
  { nel, beh, i },
  eh = ListElemsHngs [el, lf];
  nel = Length [eh];
  beh = eh;
  Do[
    beh[[i]] = ReplacePart [ beh[[i]], 2 -> 0 ];
    beh[[i]] = ReplacePart [ beh[[i]], 3 -> 0 ],
    {i, 1, nel}
  ];
  Do[
    If [ eh[[i]][[j]] == eh[[i]][[1]][[1]], beh[[i]] = ReplacePart [ beh[[i]], 2 -> 1 ]];
    If [ eh[[i]][[j]] == eh[[i]][[1]][[2]], beh[[i]] = ReplacePart [ beh[[i]], 3 -> 1 ]],
    {i, 1, nel}, {j, 2, 3}
  ];
  beh ]
```

Prva od njih generira listu elemenata *eh* u kojoj su svakom elementu pridružena dva člana čije vrijednosti odgovaraju indeksima čvorova u kojima je na tom elementu otpuštena zglobna veza (ako veza nije otpuštena, taj član, ili oba, jednaki su nuli). Druga funkcija generira (korisniju) listu u kojem ta dva člana predstavljaju redom prvi (*i*-ti) i drugi (*j*-ti) čvor u koji je element priključen i mogu poprimiti samo dvije vrijednosti – 0 (element nema zglob na tom kraju) ili 1 (element ima zglob na tom kraju). Obje funkcije pozivaju se s listom elemenata te listom čvorova s popisom elemenata koji su u njih priključeni zglobno (dobivenom pomoću funkcije *ListFERls*).

Funkcija *TblNodeDsOF* postojeća je funkcija koja određuje koliko sustav ima nepoznatih omogućenih pomaka te generira njihovu listu, pridružujući svakom takvom pomaku individualnu broječanu oznaku. U skladu s prije iznesenom teorijom, puni zglobovi moraju se maknuti iz matrice krutosti sistema, a to će se učiniti tako da se maknu s popisa nepoznatih pomaka. Funkcija *TblNodeDsOF* stoga je dorađena te sada izgleda ovako:

```
TblNodeDsOF [n_Integer, cns_List, fh_List] :=
Module[
  { i, j, hs, nc, nd, ns, di },
  di = Table [1, {n}, {3}];
  nc = Length [cns];
  ns = Length [fh];
  Do[
    di[[ cns[[i]][[1]] ]] = cns[[i]][[2]],
    {i, 1, nc}
  ];
  Do[
    di[[ fh[[i]] ]] = {1, 1, 0},
    {i, 1, ns}
  ];
  nd = 0;
  Do[
    If [ di[[i,j]] == 1,
      ++nd; di[[i,j]] = nd
    ],
    {i, 1, n}, {j, 1, 3}
  ];
  {nd, di}
]
```

Uz stare parametre, broj čvorova i listu ležajnih uvjeta, dodana je još i lista punih zglobova. Funkcija vraća programu listu kojoj je prvi član broj nepoznatih omogućenih pomaka, a drugi lista tročlanih elemenata u kojoj svaki element odgovara jednom čvoru, a svaki član tog elementa jednoj komponenti poopćenog pomaka.

Da bismo generirali matricu krutosti sustava, potrebne su nam matrice krutosti elementa. One se pak razlikuju prema tome je li element obostrano upet, jednostrano upet ili mu je zglobova veza oslododena na oba kraja. Funkcija koja generira lokalnu matricu krutosti za obostano upeti element, *ElemKLoc* ostavljena je bez izmjene, a dodane su sljedeće tri funkcije:

```
ElemKLocCnd1 [e_Real, a_Real, i_Real, l_Real] :=
Module[
  { k11, k14, k22, k23, k25, k26, k33, k35, k36, k44, k55, k56, k66, k },
  k11 = e a / l; k14 = -k11;
  k22 = 3 e i / l^3; k23 = 0; k25 = -k22; k26 = -3 e i / l^2;
  k33 = 0; k35 = 0; k36 = 0;
  k44 = k11;
  k55 = k22; k56 = -k26;
  k66 = 3 e i / l;
  k = {{ k11, 0, 0, k14, 0, 0},
        {0, k22, 0, 0, k25, k26},
        {0, 0, 0, 0, 0, 0},
        {k14, 0, 0, k44, 0, 0},
        {0, k25, 0, 0, k55, k56},
        {0, k26, 0, 0, k56, k66}};
  k ]
```

```
ElemKLocCnd2 [e_Real, a_Real, i_Real, l_Real] :=
Module[
  { k11, k14, k22, k23, k25, k26, k33, k35, k36, k44, k55, k56, k66, k },
  k11 = e a / l; k14 = -k11;
  k22 = 3 e i / l^3; k23 = -3 e i / l^2; k25 = -k22; k26 = 0;
  k33 = 3 e i / l; k35 = -k23; k36 = 0;
  k44 = k11;
  k55 = k22; k56 = -k26;
  k66 = 0;
  k = {{ k11, 0, 0, k14, 0, 0},
        {0, k22, k23, 0, k25, 0},
        {0, k23, k33, 0, k35, 0},
        {k14, 0, 0, k44, 0, 0},
        {0, k25, k35, 0, k55, 0},
        {0, 0, 0, 0, 0, 0}};
  k ]
```

```
ElemKLocCnd3 [e_Real, a_Real, i_Real, l_Real] :=
Module[
  { k11, k14, k22, k23, k25, k26, k33, k35, k36, k44, k55, k56, k66, k },
  k11 = e a / l; k14 = -k11; k44 = k11;
  k = {{ k11, 0, 0, k14, 0, 0},
        {0, 0, 0, 0, 0, 0},
        {0, 0, 0, 0, 0, 0},
        {k14, 0, 0, k44, 0, 0},
        {0, 0, 0, 0, 0, 0},
        {0, 0, 0, 0, 0, 0}};
  k ]
```

Prva od njih generira matricu krutosti elementa sa zglobom na  $i$ -tom kraju, druga matricu krutosti elementa sa zglobom na  $j$ -tom kraju, a treća matricu krutosti zglobnog štapa. Sve funkcije pozivaju se s 4 parametra: modul elastičnosti, površina poprečnog presjeka, moment tromosti poprečnog presjeka i duljina elementa.

Funkciji koja generira listu matrica krutosti elemenata dodane su *If*-naredbe unutar kojih se pozivaju gore navedene funkcije, kako bi se za elemente sa zglobnim krajevima generirale odgovarajuće, kondenzirane, matrice krutosti. Ta funkcija sada izgleda ovako:

```
ListKLoc [els_List, nds_List, echs_List, lf_List] :=
Module[
  { beh, lk, el, e, a, i, l, j },
  lk = Table [Null, {Length [els]}];
  beh = ListElemsHngsBin [els, lf];
  Do[
    el = els[[j]];
    {e, a, i} = ElemChar [el, echs];
    l = ElemLength [ElemNds [el, nds]];
    lk[[j]] = ElemKLoc [e, a, i, l];
    If [ beh[[j]][[2]] == 1,
      lk[[j]] = ElemKLocCnd1 [e, a, i, l];
      If [ beh[[j]][[3]] == beh [[j]][[2]],
        lk[[j]] = ElemKLocCnd3 [e, a, i, l] ];
    If [ beh[[j]][[3]] == 1,
      lk[[j]] = ElemKLocCnd2 [e, a, i, l];
      If [ beh[[j]][[2]] == beh [[j]][[3]],
        lk[[j]] = ElemKLocCnd3 [e, a, i, l] ],
    {j, 1, Length [els]}
  ];
  lk ]
```

*els* je lista elemenata, *nds* lista čvorova, *echs* lista geometrijskih karakteristika, a *lf* lista čvorova e elementima koji su u njih priključeni zglobno.

Nakon generiranja matrice transformacije, program je spreman napraviti matricu krutosti cijelog sistema. Navedeni postupci ostaju nepromijenjeni.

Slijedi opis opterećenja. Zasad smo opterećenja ograničili na poopćene sile koje djeluju u čvorovima i poopćene koncentrirane sile na elementima. I jedna i druga opterećenja zadaju se u obliku lista. Vanjske sile koje djeluju na čvorove direktno ulaze u jednadžbe ravnoteže čvorova i za njih vrijede isti izrazi neovisno o mogućem postojanju zglobova u čvorovima, pa se njima više nećemo baviti. Posao sa silama na elementima je nešto složeniji. One se upisuju u obliku liste s tri komponente – prva je indeks elementa na kojem opterećenje djeluje, druga je naziv funkcije koja određuje vrstu opterećenja (u našem slučaju uvijek *ConcFL* jer smo se ograničili na koncentrirane sile), a treća je lista u kojoj se nalaze vrijednosti poopćenih sila i kvocijent  $a/l$  koji govori o položaju tog opterećenja na elementu.

Funkcija *ListFxEndFsLoc* generira listu elemenata s pripadajućim silama upetosti. Njezini su parametri lista opterećenja na elementima, lista elemenata, lista čvorova i lista elemenata s označenim oslobođenim krajevima. Ona prvo poziva funkciju *FxEndFrCs* koja će u budućnosti određivati koja će se sljedeća funkcija pozvati, a u ovisnosti o zadanom opterećenju (zasad je to obavezno funkcija *ConcFL*). Nakon toga, funkcija *ListFxEndFsLoc* korigira dobivene vrijednosti da bi odgovarale elementima sa zglobnim krajevima. Napisane su promijenjena funkcija *ListFxEndFsLoc* i tri dodane funkcije (za zglob na *i*-tom kraju, za zglob na *j*-tom kraju i za zglobni štap:

```
ListFxEndFsLoc [elds_List, els_List, nds_List, beh_List] :=
Module[
  { lef, ei, f, lp, j },
  lef = Table [{0, {0, 0}, Table [0.0, {6}]}, {Length [elds]}];
  Do[
    ei = elds[[j]][[1]];
    f = elds[[j]][[2]];
    lp = elds[[j]][[3]];
    lef[[j]] = {ei,
      ElemNdIds [ els[[ei]] ],
      FxEndFrCs [ ei, f, lp, els, nds ]
    };
    If [ beh[[ei]][[2]] == 1,
      lef[[j]] = {ei,
        ElemNdIds [ els[[ei]] ],
        FxEndFsCnd1[FxEndFrCs [ei, f, lp, els, nds], ei, els, nds ] };
    If [ beh[[ei]][[3]] == beh[[ei]][[2]],
      lef[[j]] = {ei,
        ElemNdIds [ els[[ei]] ],
        FxEndFsCnd3[FxEndFrCs [ei, f, lp, els, nds], ei, els, nds ] } ]];
    If [ beh[[ei]][[3]] == 1,
      lef[[j]] = {ei,
        ElemNdIds [ els[[ei]] ],
        FxEndFsCnd2[FxEndFrCs [ei, f, lp, els, nds], ei, els, nds ] };
    If [ beh[[ei]][[2]] == beh [[ei]][[3]],
      lef[[j]] = {ei,
        ElemNdIds [ els[[ei]] ],
        FxEndFsCnd3[FxEndFrCs [ei, f, lp, els, nds], ei, els, nds ] } ]],
    {j, 1, Length [elds]}
  ];
  lef
]
```

```

FxEndFsCnd1 [lef_List, ei_Integer, els_List, nds_List] :=
Module[
  { leff, el, l },
  el = Elem [ei, els];
  l = ElemLength [ElemNds [el, nds]];
  leff = lef;
  leff[[1]] = lef[[1]];
  leff[[2]] = lef[[2]] + 3 / (2l) lef[[3]];
  leff[[3]] = 0;
  leff[[4]] = lef[[4]];
  leff[[5]] = lef[[5]] - 3 / (2l) lef[[3]];
  leff[[6]] = lef[[6]] - 1/2 lef[[3]];
  leff
]

```

```

FxEndFsCnd2 [lef_List, ei_Integer, els_List, nds_List] :=
Module[
  { leff, el, l },
  el = Elem [ei, els];
  l = ElemLength [ElemNds [el, nds]];
  leff = lef;
  leff[[1]] = lef[[1]];
  leff[[2]] = lef[[2]] + 3 / (2l) lef[[6]];
  leff[[3]] = lef[[3]] - 1/2 lef[[6]];
  leff[[4]] = lef[[4]];
  leff[[5]] = lef[[5]] - 3 / (2l) lef[[6]];
  leff[[6]] = 0;
  leff
]

```

```

FxEndFsCnd3 [lef_List, ei_Integer, els_List, nds_List] :=
Module[
  { leff, el, l },
  el = Elem [ei, els];
  l = ElemLength [ElemNds [el, nds]];
  leff = lef;
  leff[[1]] = lef[[1]];
  leff[[2]] = 0;
  leff[[3]] = 0;
  leff[[4]] = lef[[4]];
  leff[[5]] = 0;
  leff[[6]] = 0;
  leff
]

```

Potonje funkcije pozivaju se s listom vrijednosti sila upetosti dobivenih za obostrano upeti štap, indeksom elementa, listom elemenata i listom čvorova, a funkciji *ListFxEndFsLoc* vraćaju konačne vrijednosti poopćenih sila upetosti na krajevima elemenata. Nakon toga se iz svih podataka generira vektor poznatih sila (sila upetosti i sila koje djeluju u čvorovima)  $\mathbf{q}$ , te se rješava jednačba

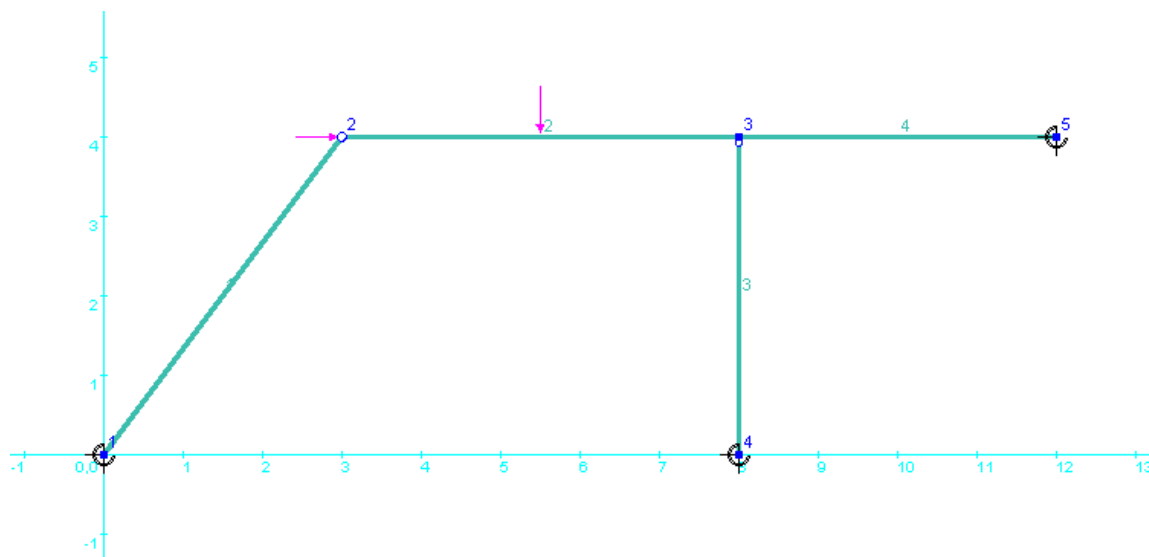
$$\mathbf{Ku} = \mathbf{q}$$

kako bi se dobile vrijednosti pomaka čvorova. Odavde se pak izračunavaju pomaci krajeva elemenata, a iz njih vrijednosti sila na krajevima elemenata. Ništa se ne mijenja u ovisnosti o postojanju zglobova, te se taj, završni, dio proračuna ovdje neće obrađivati.



## 4. Primjer

Pravilnost rada ovako dobivenog programa pokazat će se na primjeru ravninskog štap-nog sistema sa 5 čvorova i 4 štapa, prikazanog na slici 5.



Slika 5.

Upis čvorova i elemenata:

```
nds = {{0., 0.}, {3., 4.}, {8., 4.}, {8., 0.}, {12., 4.}}  
{{0., 0.}, {3., 4.}, {8., 4.}, {8., 0.}, {12., 4.}}
```

```
els = {{{1, 2}, 1}, {{2, 3}, 1}, {{3, 4}, 1}, {{3, 5}, 1}}  
{{{1, 2}, 1}, {{2, 3}, 1}, {{3, 4}, 1}, {{3, 5}, 1}}
```

Popis elemenata po čvorovima:

```
TblElemsByNodes[Length[nds], els]  
{{1, {1, 2}}, {2, {1, 2}, {2, 3}}, {3, {2, 3}, {3, 4}, {3, 5}}, {4, {3, 4}}, {5, {3, 5}}}
```

Upis geometrijski karakteristika, ležajnih uvjeta, zglobova i oslobođenih krajeva elemenata:

```

chrs = {{3. 10^7, 0.5^2, 0.5^4/12}}
{{3.*10^7, 0.25, 0.00520833}}

cnsts = {{1, {0, 0, 0}}, {4, {0, 0, 0}}, {5, {0, 0, 0}}}
{{1, {0, 0, 0}}, {4, {0, 0, 0}}, {5, {0, 0, 0}}}

hngs = {2}
{2}

endrls = {{{3, 4}, 3, {0, 0, 1}}}
{{{3, 4}, 3, {0, 0, 1}}}

```

Popis punih zglobova:

```

fh = ListFHngs[nds, els, hngs, cnsts, endrls]
{2}

```

Popis čvorova sa svim elementima koji su u njih priključeni zglobno:

```

lf = ListFERls [endrls, nds, els, hngs, cnsts]
{{1}, {2, {1, 2}}, {2, 3}}, {3, {3, 4}}, {4}, {5}}

```

Popis elemenata s označenim zglobnim krajevima:

```

beh = ListElmsHngsBin[els, lf]
{{{1, 2}, 0, 1}, {{2, 3}, 1, 0}, {{3, 4}, 1, 0}, {{3, 5}, 0, 0}}

```

Broj nepoznatih omogućenih pomaka i njihove oznake:

```

{nd, di} = TblNodeDsOF [Length[nds], cnsts, fh]
{5, {{0, 0, 0}, {1, 2, 0}, {3, 4, 5}, {0, 0, 0}, {0, 0, 0}}}

```

Lista matrica krutosti elemenata:

```

lkl = ListKLoc [els, nds, chrs, lf]
{{{1.5*10^6, 0, 0, -1.5*10^6, 0, 0}, {0, 3750., -18750., 0, -3750.,
0}, {0, -18750., 93750., 0, 18750., 0}, {-1.5*10^6, 0, 0, 1.5*10^6,
0, 0}, {0, -3750., 18750., 0, 3750., 0}, {0, 0, 0, 0, 0,
0}}, {{1.5*10^6, 0, 0, -1.5*10^6, 0, 0}, {0, 3750., 0,
0, -3750., -18750.}, {0, 0, 0, 0, 0, 0}, {-1.5*10^6, 0, 0,
1.5*10^6, 0, 0}, {0, -3750., 0, 0, 3750., 18750.}, {0, -18750., 0,
0, 18750., 93750.}}, {{1.875*10^6, 0, 0, -1.875*10^6, 0, 0}, {0,
7324.22, 0, 0, -7324.22, -29296.9}, {0, 0, 0, 0, 0,
0}, {-1.875*10^6, 0, 0, 1.875*10^6, 0, 0}, {0, -7324.22, 0, 0,
7324.22, 29296.9}, {0, -29296.9, 0, 0, 29296.9,
117188.}}, {{1.875*10^6, 0, 0, -1.875*10^6, 0, 0}, {0,
29296.9, -58593.8, 0, -29296.9, -58593.8}, {0, -58593.8, 156250.,
0, 58593.8, 78125.}, {-1.875*10^6, 0, 0, 1.875*10^6, 0,
0}, {0, -29296.9, 58593.8, 0, 29296.9, 58593.8}, {0, -58593.8,
78125., 0, 58593.8, 156250.}}}

```

Matrica transformacije:

```
lr = ListTG2L [els, nds]
{{{0.6, -0.8, 0, 0, 0, 0}, {0.8, 0.6, 0, 0, 0, 0}, {0, 0, 1, 0, 0,
0}, {0, 0, 0, 0.6, -0.8, 0}, {0, 0, 0, 0.8, 0.6, 0}, {0, 0, 0, 0,
0, 1}}, {{1., 0., 0, 0, 0, 0}, {0., 1., 0, 0, 0, 0}, {0, 0, 1, 0,
0, 0}, {0, 0, 0, 1., 0., 0}, {0, 0, 0, 0., 1., 0}, {0, 0, 0, 0, 0,
1}}, {{0., 1., 0, 0, 0, 0}, {-1., 0., 0, 0, 0, 0}, {0, 0, 1, 0, 0,
0}, {0, 0, 0, 0., 1., 0}, {0, 0, 0, -1., 0., 0}, {0, 0, 0, 0, 0,
1}}, {{1., 0., 0, 0, 0, 0}, {0., 1., 0, 0, 0, 0}, {0, 0, 1, 0, 0,
0}, {0, 0, 0, 1., 0., 0}, {0, 0, 0, 0., 1., 0}, {0, 0, 0, 0, 0,
1}}}
```

Matrica krutosti sistema:

```
K = StructK [nd, di, els, lkl, lr]
{{2.0424*10^6, -718200., -1.5*10^6, 0., 0.}, {-718200., 965100.,
0., -3750., -18750.}, {-1.5*10^6, 0., 3.38232*10^6, 0.,
0.}, {0., -3750., 0., 1.90805*10^6, -39843.8}, {0., -18750.,
0., -39843.8, 250000.}}
```

Vanjska opterećenja čvorova i elemenata:

```
lnld = {{2, {100., 0., 0.}}}
{{2, {100., 0., 0.}}}

l1d = {{2, ConcFL, {0., 100., 0., 0.5}}}
{{2, ConcFL, {0., 100., 0., 0.5}}}
```

Vrijednosti sila upetosti na svim opterećenim elementima:

```
lef = ListFxEndFsLoc [l1d, els, nds, beh]
{{2, {2, 3}, {0., -31.25, 0, 0., -68.75, -93.75}}}
```

Vektor poznatih djelovanja:

```
q = StructF [nd, di, lnld, lef, lr]
{100., 31.25, 0., 68.75, 93.75}
```

Vektor pomaka čvorova:

```
u = LinearSolve [K, q]
{0.000152926, 0.000154004, 0.0000678198, 0.0000445545, 0.000393651}
```

Vrijednosti pomaka krajeva elemenata:

```
lul = ListEDisplsLoc [u, di, els, lr]
{{0., 0., 0., -0.0000314478, 0.000214743, 0.}, {0.000152926,
0.000154004, 0., 0.0000678198, 0.0000445545,
0.000393651}, {0.0000445545, -0.0000678198, 0.000393651, 0., 0.,
0.}, {0.0000678198, 0.0000445545, 0.000393651, 0., 0., 0.}}
```

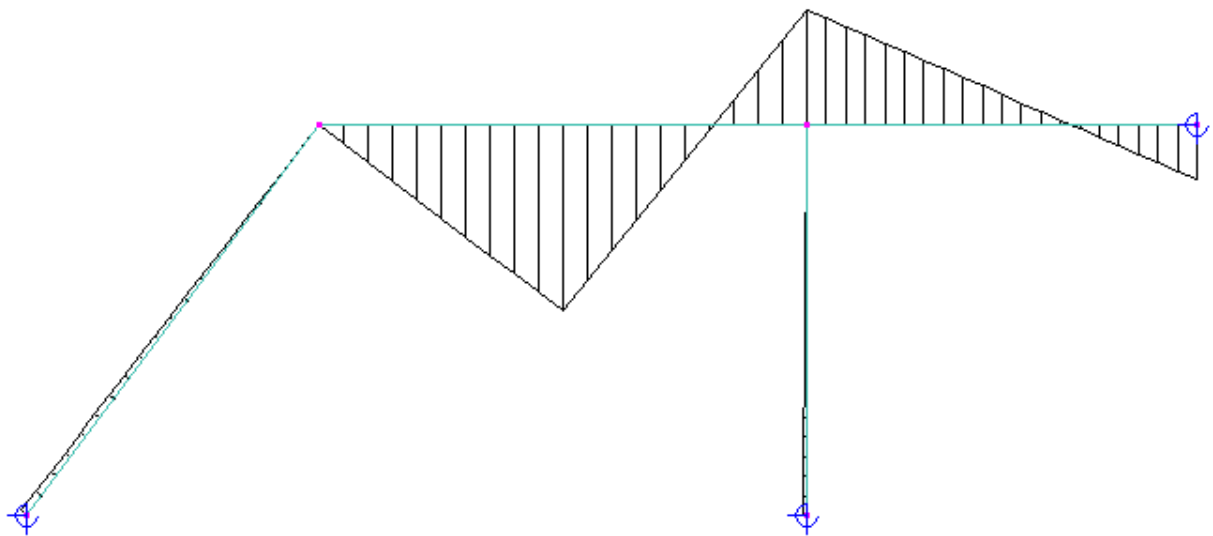
Vrijednosti ukupnih sila na krajevima elemenata:

```
lefs = ListEndForces [lkl, lul, lef]
{{47.1717, -0.805286, 4.02643, -47.1717, 0.805286,
0}, {127.659, -38.2205,
0, -127.659, -61.7795, -58.8974}, {83.5397, -0.496727, 0, -83.5397,
0.496727, 1.98691}, {127.162, -21.7602, 58.8974, -127.162, 21.7602,
28.1434}}
```

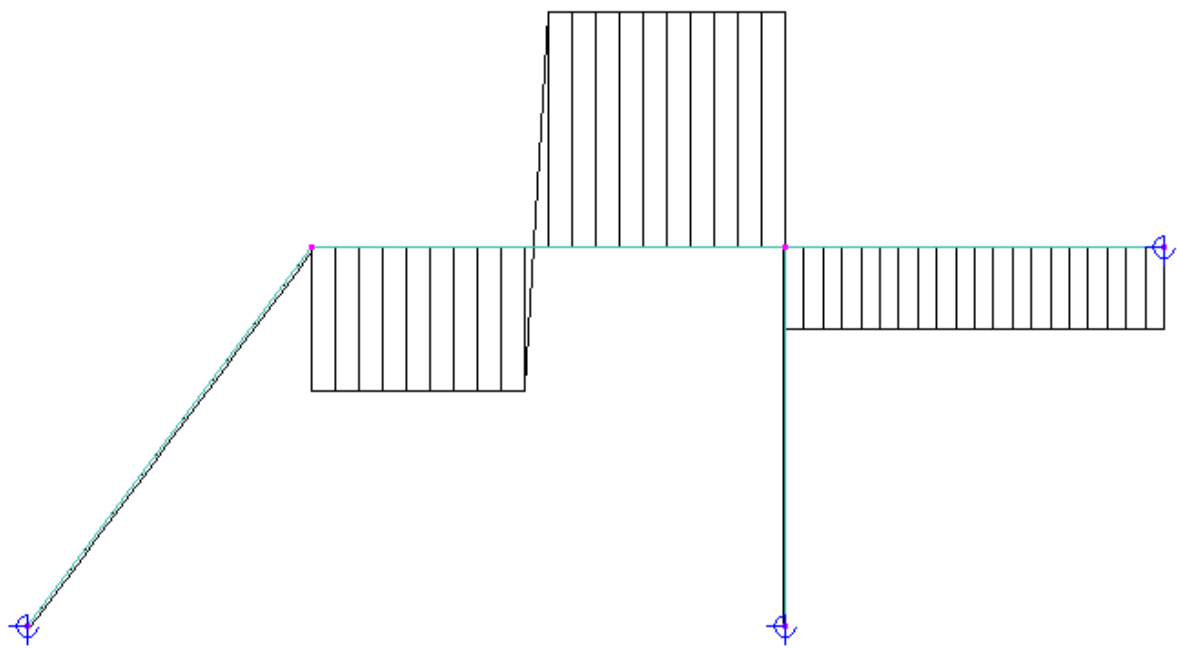
Isti sistem proračunao se Java appletom *Displacement Method* te su dobiveni identični rezultati. Popis rezultata i odgovarajući dijagrami iz *DiM*-a prikazani su na slikama 6., 7., 8. i 9.

<b>Degrees of Freedom: 5</b>						
<b>Nodal displacements:</b>						
nd	u <sub>i</sub>	v <sub>i</sub>	phi <sub>i</sub>			
1:	0	0	0			
2:	0.000152926	-0.000154004	0			
3:	6.78198e-05	-4.45545e-05	0.000393651			
4:	0	0	0			
5:	0	0	0			
<b>Element end forces:</b>						
el	H <sub>ij</sub>	T <sub>ij</sub>	M <sub>ij</sub>	H <sub>ji</sub>	T <sub>ji</sub>	M <sub>ji</sub>
1:	47.1717	0.805285	4.02643	-47.1717	-0.805285	0
2:	127.659	38.2205	0	-127.659	61.7795	-58.8974
3:	83.5397	0.496726	0	-83.5397	-0.496726	1.98691
4:	127.162	21.7602	58.8974	-127.162	-21.7602	28.1434
<b>Reactions:</b>						
nd	R <sub>x</sub>	R <sub>y</sub>	M			
1:	27.6588	38.2205	4.02643			
4:	-0.496726	83.5397	1.98691			
5:	-127.162	-21.7602	28.1434			

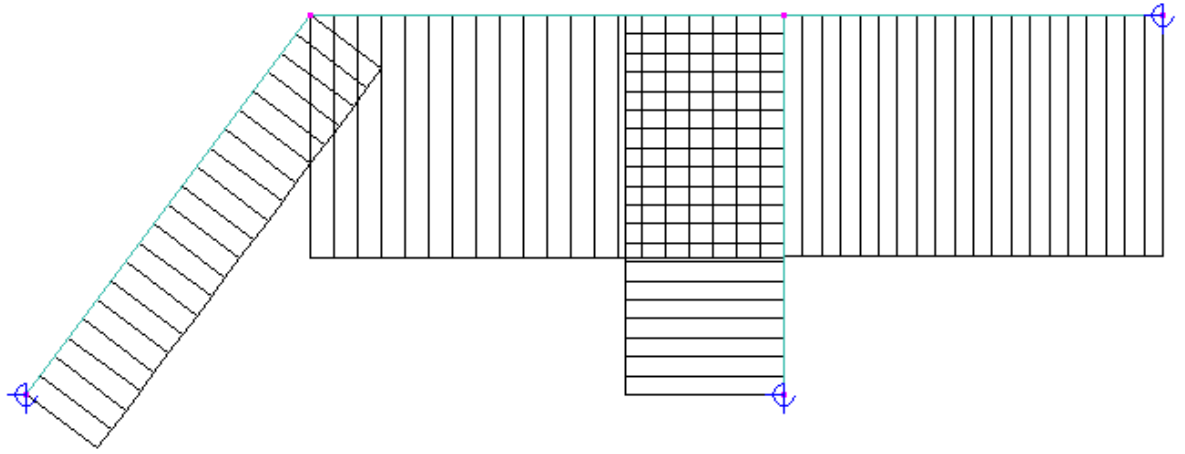
Slika 6.



Slika 7.



Slika 8.



Slika 9.

## 5. Zaključak

Za računalno rješavanje ravninskih štapnih sistema, zbog svoje jednostavne strukture i razmjerno lake formalizacije, najčešće je korištena metoda pomaka. U ovom radu je postojeći program dorađen tako što je u algoritam pretočena ideja statičke kondenzacije za zglobove na sistemu. Pokazuje se da je programski paket *Wolfram Mathematica 6* prilično jednostavan i "susretljiv" za programiranje ovakve vrste zadataka, te se očekuje daljnji razvoj programa za statički proračun ravninskih štapnih konstrukcija na dosad izgrađenim temeljima.

## Literatura

- [1] M. Anđelić: *Građevna statika II*, Građevinski fakultet Sveučilišta u Zagrebu, Zagreb, 2005.
- [2] P. Gidak: *Proširenja programa DiM*, natječajni rad, Zagreb, 2002.
- [3] D. Matković, K. Mikulić: *Kinematička kondenzacija*, natječajni rad, Zagreb, 2008.
- [4] V. Simović, K. Fresl: *Građevna statika 2.*, bilješke i skice s predavanja, <http://www.grad.hr/nastava/gs/bilj2/index.html>, pristupljeno 27. kolovoza 2008.



## Sažetak

Metoda pomaka je metoda za proračun statičkih konstrukcija koju je razmjerno lako formalizirati i pretočiti u računalni algoritam. Zato je ona često osnova računalnih programa za statičke proračune. U ovom radu se postojeći program, napisan u programskom paketu *Wolfram Mathematica 6*, koji je dosad radio samo na obostrano upetim gredama, proširio statičkom kondenzacijom te mu je dodana mogućnost računanja sistema sa zglobovima. Na početku rada je dana teorijska podloga statičke kondenzacije, a zatim je izveden računalni algoritam. Napisane su nove funkcije za generiranje matrica krutosti elemenata, listi sila upetosti elemenata i popisa nepoznatih omogućenih pomaka čvorova te mnoštvo potrebnih pomoćnih funkcija. Na kraju rada se novim programom proračunala statika jednog ravninskog štapnog sistema, a identičan se proračun proveo i u Java appletu *Displacement Method*. Dobiveni su identični rezultati, što pokazuje ispravnost programiranih funkcija. Očekuje se daljnji razvoj ovog programa za proračun ravninskih štapnih konstrukcija.