

Sveučilište u Zagrebu
Građevinski fakultet

Dinka Matković i Katja Mikulić
Kinematička kondenzacija

Zagreb, 2008.

Ovaj rad izrađen je na Katedri za statiku, dinamiku i stabilnost konstrukcija u Zavodu za tehničku mehaniku Građevinskog fakulteta Sveučilišta u Zagrebu pod vodstvom prof. dr. sc. Krešimira Fresla, dipl. ing. građ., i predan je na natječaj za dodjelu Rektorove nagrade u akademskoj godini 2007./2008.

Sadržaj

1. Uvod	1
2. Opća metoda pomakâ	4
2.1. Zamisao metode	4
2.2. Matrična formulacija	4
3. Inženjerska metoda pomakâ	22
4. Kinematička ograničenja	28
4.1. Formalni izraz pretpostavke inženjerske metode pomakâ	28
4.2. Sustav jednažbi kinematičkih ograničenja	29
4.3. Kinematička kondenzacija	37
5. Primjer	43
6. Zaključak	59
Literatura	60
Sažetak	62
Summary	62

1. Uvod

U svojim se svakodnevnim proračunima građevinski inženjeri danas služe velikim brojem različitih kompjutorskih programa. Točnost računala je neupitna, ono računa puno brže i preciznije od čovjeka. Međutim, ni proračun pomoću računala nije lišen pogrešaka koje neizbježno nastaju pri aproksimaciji konstrukcije matematičkim modelom, pri aproksimaciji matematičkog modela numeričkim, pri rješavanju tog numeričkog modela, a do grešaka naravno dolazi i zbog netočnosti izvedene konstrukcije u odnosu na projektiranu [10]. Pogreške koje se mogu izbjeći, ali su također česte i primjetne, su pogreške koje nastaju kao rezultat neznanja ili, barem, nedovoljnog i nepotpunog znanja — loša koncepcija objekta, loše modeliranje, pogreške u proračunu i slično [5].

Poznato je da nas u metodi pomakâ aproksimacija matematičkoga modela numeričkim u linearnom slučaju vodi na sustav linearnih jednadžbi ravnoteže

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

gdje su \mathbf{K} , \mathbf{f} i \mathbf{u} redom matrica krutosti, vektor sila i vektor (nepoznatih) pomaka [1, 8]. [To je globalni sustav, što znači da smo a) formirali lokalne matrice krutosti i vektore opterećenja pojedinih elemenata, b) transformirali te matrice i vektore iz lokalnih u globalni koordinatni sustav, c) uvažili ležajne uvjete i d) sastavili matricu krutosti konstrukcije.]

Znamo također da vektor pomakâ \mathbf{u} mora zadovoljiti uvjete ravnoteže, uvjete kompatibilnosti, ležajne uvjete i zakon ponašanja [10]. Međutim, ponekad možemo unaprijed odrediti neke pomake ili definirati njihove međusobne odnose. Tada govorimo o ograničenjima vektora pomakâ ili o kinematičkim ograničenjima [4, 10]. Ta ograničenja u pravilu zadajemo prema inženjerskoj procjeni uvjetâ oslanjanja, ponašanja spojeva, materijalnih i geometrijskih karakteristika elemenata ili opterećenja. U općem se slučaju kinematička ograničenja mogu opisati dodatnim sustavom nejednadžbi

$$\mathbf{u} \lesseqgtr \mathbf{C}(\mathbf{x}) \mathbf{x}.$$

Matrica $\mathbf{C}(\mathbf{x})$ opisuje vezu nepoznatih parametara \mathbf{x} , koji ne moraju biti pomaci, i nepoznatih pomaka modela bez ograničenja. Budući da u općem slučaju ograničenja ovise o nepoznatim parametrima, te su nejednadžbe nelinearne [10]. Ograničenja u obliku nejednakosti pojavljuju se u problemima optimizacije i kontaktne mehanike [17]. U linearnoj se statici ograničenja mogu prikazati linearnim jednadžbama

$$\mathbf{u} = \mathbf{C} \mathbf{x} + \mathbf{g},$$

pri čemu je najčešće \mathbf{x} vektor pomakâ. Ako je $\mathbf{g} = \mathbf{0}$, ograničenja su homogena.

Jedan je od mogućih postupaka smanjenja pogrešaka pri aproksimaciji matematičkog modela numeričkim i pri rješavanju numeričkog modela uvođenje kinematičkih ograničenja

i njihovo uvažavanje kinematičkom kondenzacijom [4, 5]. Klasična, inženjerska metoda pomakâ, razvijena u vremenima prije pojave računala, temelji se na zapažanju da je uzdužna krutost štapnih elemenata puno veća od krutosti pri savijanju:

$$\frac{EA}{\ell} \gg \frac{12EI}{\ell^3}.$$

Zbog toga su u, primjerice, okviru s vertikalnim stupovima i horizontalnim gredama uzdužne deformacije štapova puno manje od horizontalnih pomaka etaža. Prirodno je uvesti ograničenja po kojima svi čvorovi neke etaže imaju jednake horizontalne pomake, a vertikalni su pomaci svih čvorova jednaki nuli. Drugim riječima, uzdužna je krutost elemenata beskonačno velika [9, 10]. Kako se u numeričkom proračunu ne može baratati beskonačno velikim vrijednostima, one se mogu aproksimirati zadavanjem razmjerno velikih vrijednosti u odnosu na ostale vrijednosti; te se velike vrijednosti nazivaju kaznenim koeficijentima [7, 16]. Druga je mogućnost uvođenje dodatnih nepoznanica, takozvanih Lagrangeovih multiplikatora, koji su, fizikalno, nepoznate sile koje bi osigurale ograničenje kad bi se nanijele na konstrukciju u kojoj nema ograničenja [7, 16]. (Očito je riječ o mješovitoj metodi — nepoznanice su i pomaci i sile, pa se metoda pomakâ kombinira s metodom sila.) I napokon, treća je mogućnost kinematička kondenzacija, uklanjanje vezanih pomakâ iz skupa nepoznanica uz istodobno uklanjanje odgovarajućih jednadžbi [4, 7, 10].

Uvažavanje kinematičkih ograničenja često se krivo doživljava, pa se izbjegava njihova uporaba. Naime, inženjeri nedovoljno upoznati s tom problematikom smatraju, po analogiji s inženjerskom metodom pomakâ, da je primarna prednost uvažavanja ograničenja smanjenje broja nepoznanica i, stoga, skraćenje vremena potrebnog za provedbu proračuna, tako da i nema prevelikog smisla služiti se time, budući da su računala danas ionako već dovoljno brza. A u stvari te prednosti često ni nema, jer uvažavanje ograničenja kinematičkom kondenzacijom mijenja raspored elemenata matrice krutosti (,kvarimo' ga), pa je potrebno više memorije, a samim time i više vremena za provedbu proračuna [7]; Lagrangeovi su pak multiplikatori dodatne nepoznanice, pa se sustav jednadžbi povećava, a kazneni koeficijenti ne mijenjaju ni strukturu ni veličinu sustava, ali osjetno smanjuju njegovu uvjetovanost [7, 16].

Osnovna je prednost uvažavanja kinematičkih ograničenja kondenzacijom u tome što značajno poboljšavaju uvjetovanost sustava jednadžbi i učinkovito uklanjaju singularitete u odnosu na uvođenje vrlo krutih elemenata/kaznenih koeficijenata [5, 10].

Da pojasnimo: broj uvjetovanosti upotrebljavamo kako bismo procijenili točnost rješenja sustava linearnih jednadžbi [3]. Relativna pogreška u rezultatu $\|\Delta \mathbf{u}\|/\|\mathbf{u}\|$ ocjenjuje se kao umnožak broja uvjetovanosti $\text{cond}(\mathbf{K})$ matrice \mathbf{K} i norme relativne pogreške u podacima [5]:

$$\frac{\|\Delta \mathbf{u}\|}{\|\mathbf{u}\|} \leq \text{cond}(\mathbf{K}) \left(\frac{\|\Delta \mathbf{K}\|}{\|\mathbf{K}\|} + \frac{\|\Delta \mathbf{q}\|}{\|\mathbf{q}\|} \right);$$

u prethodnom su izrazu $\Delta\mathbf{K}$ i $\Delta\mathbf{q}$ apsolutne pogreške/nepouzdanosti podataka (krutosti i sila), dok je $\Delta\mathbf{u}$ apsolutna progreska u rješenju (pomacima).

Matrica tipičnog — zapravo, „skromnijeg“ — modela neke konstrukcije od nekoliko tisuća nepoznanica može imati broj uvjetovanosti 10^6 i više, što znači da je relativna pogreška u rezultatima proračuna barem milijun puta veća od relativne pogreške u podacima. S porastom broja jednadžbi sustava raste i broj uvjetovanosti, dakle možemo reći da su veliki sustavi loše uvjetovani već zbog svoje veličine [5]. Uvažavanjem kinematičkih ograničenja kondenzacijom smanjujemo broj jednadžbi i time povećavamo točnost rješenja (a ne brzinu proračuna).

Uvijek ostaje pitanje lošeg modeliranja i problem točnog odabira ograničenja, što i nije tako jednostavno već kod malo složenije konstrukcije. Ipak, vjerujemo da s malo programerske volje i taj problem može riješiti prikladni softver.

U okviru ovog rada je programskoj jezgri za proračun ravninskih štapnih sistema metodom pomakâ, napisanoj u jeziku **Mathematica**, dodana mogućnost uvažavanja kinematičkih ograničenja kinematičkom kondenzacijom. Sustav **Mathematica** upotrijebili smo ponajprije zato što omogućava simboličko „računanje“ [15, 11]. Naime, za rješavanje sistema općom metodom pomakâ baratanje simbolima nije potrebno — jednadžbe (koliko god da ih ima) mogu se nekim prikladnim programom numerički razmjerno lako riješiti [1, 8]. Međutim, sustavi jednadžbi kinematičkih ograničenja u pravilu imaju veći broj nepoznanica od samog broja jednadžbi [2, 7]. Iako se i takvi sustavi mogu riješiti Gaussovim ili Gauss–Seidelovim postupkom, dio nepoznanica, nazvanih slobodnim nepoznanicama, „prelazi“ desno od znaka jednakosti, pa se ostale, vezane, nepoznanice izražavaju kao (linearne) funkcije slobodnih nepoznanica — u rješenju ostaju simboli koji označavaju slobodne nepoznanice [6, 12]. Uz to, otvorenim ostaje pitanje izbora ili zadavanja slobodnih nepoznanica [7]. Simbolička algebra u tom je slučaju vrlo prikladan alat.

U sljedećem je odjeljku, uz primjer proračuna, detaljnije opisan već skicirani algoritam opće metode pomakâ i njegova realizacija u programskom jeziku **Mathematica**. U trećem je odjeljku isti primjer riješen „ručno“, inženjerskom metodom pomakâ. Naime, kinematičko ograničenje uvedeno u četvrtom odjeljku formalni je izraz osnovne pretpostavke inženjerske metode pomakâ. U tom je odjeljku opisano i uvažavanje uvedenih ograničenja kinematičkom kondenzacijom, uz „automatski“, strojni izbor vodećih pomaka te uz vodeće pomake koje zadaje korisnik programa. U petom je odjeljku na složenijem primjeru prikazana upotreba programa, a napravljena je i usporedba rješenja, onih dobivenih općom metodom pomakâ te onih dobivenih uvažavanjem kinematičkih ograničenja.

2. Opća metoda pomakâ

2.1. Zamisao metode

Opća (ili točna) metoda pomakâ je metoda kojom proračunavamo štapne sisteme i u kojoj su nepoznanice zaokreti i translacijski pomaci odabranih točaka sistema koje zovemo čvorovima [1, 8]. Čvorom možemo proglasiti bilo koju točku, ali obično su to ,istaknute točke' na sistemu: točke u kojima se sastaje više štapova, točke u kojima se sastaju dva štapa, ali pod nekim kutom, i slično.

Ako se ograničimo na ravninske štapne sisteme, svaki čvor i ima dva translacijska stupnja slobode (ako sistem prikažemo u ravnini xy bit će to pomaci u_i , u smjeru osi x , i v_i , u smjeru osi y) i jedan zaokret φ_i oko osi okomite na ravninu sistema.

Osnovna ideja metode pomakâ sastoji se u sljedećem:

- zadani sistem dovedemo u stanje spriječenih pomaka čvorova tako da sistemu dodamo zamišljene veze koje sprječavaju sve moguće pomake;
- time se sistem ,raspada' na niz međusobno neovisnih obostrano upetih greda na kojima računamo sile od zadanog opterećenja;
- uklanjanjem dodanih veza sistem dovodimo u stanje slobodnih pomaka čvorova, pa se računaju sile od pomaka čvorova izvornog sistema;
- konačne sile u elementima dobivamo superpozicijom.

2.2. Matrična formulacija

2.2.1. Lokalna matrica krutosti i lokalni vektor sila upetosti

Štapni element, označimo ga sa e , jednoznačno je određen indeksima i i j koji označavaju čvorove (slika 1.). Istim indeksima označavat ćemo i sile na krajevima štapnog elementa (poopćene sile kojima čvorovi djeluju na element).

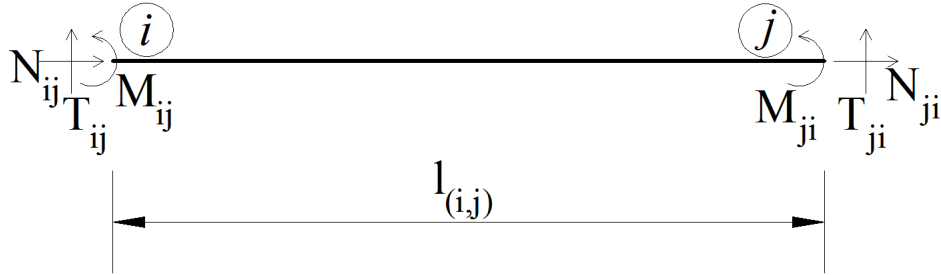
Ukupne sile na krajevima štapa zbrojevi su sila stanja spriječenih pomaka i sila stanja njihovih slobodnih pomaka, te možemo zapisati za sile na kraju i :

$$N_{i,j} = n_{i,j} + \overline{N}_{i,j}, \quad (1)$$

$$T_{i,j} = t_{i,j} + \overline{T}_{i,j}, \quad (2)$$

$$M_{i,j} = m_{i,j} + \overline{M}_{i,j}, \quad (3)$$

gdje su $n_{i,j}$, $t_{i,j}$, $m_{i,j}$ sile stanja slobodnih, a $\overline{N}_{i,j}$, $\overline{T}_{i,j}$, $\overline{M}_{i,j}$ sile stanja spriječenih pomaka.



Slika 1.

Sile oba stanja možemo izračunati metodom sila, analizom obostrano upete grede.

Sile stanja slobodnih pomaka pojavljuju se kao posljedica poopćenih pomaka krajeva elemenata (oni su jednaki pomacima čvorova sistema). Ti pomaci postaju prisilni pomaci ležajeva naše upete grede.

Za svaki element možemo za sile u stanju slobodnih pomaka napisati jednadžbu:

$$\mathbf{f}_e = \mathbf{k}_e \mathbf{u}_e, \quad (4)$$

gdje je \mathbf{f}_e vektor sila na krajevima zbog pomaka krajeva elementa, \mathbf{k}_e matrica krutosti elementa, a \mathbf{u}_e je vektor pomaka krajeva elementa, pri čemu su svi članovi izraženi u lokalnom koordinatnom sustavu.

Uz pretpostavke da su elementi zadanoga sistema konstantnog poprečnog presjeka ($I(x) = I_e$, $A(x) = A_e$) te da je modul elastičnosti također konstantan ($E(x) = E_e$), gornju jednadžbu možemo zapisati u matricnom obliku:

$$\begin{bmatrix} n_{i,j} \\ t_{i,j} \\ m_{i,j} \\ n_{j,i} \\ t_{j,i} \\ m_{j,i} \end{bmatrix} = \begin{bmatrix} \frac{E_e A_e}{l_e} & 0 & 0 & -\frac{E_e A_e}{l_e} & 0 & 0 \\ 0 & \frac{12 E_e I_e}{l_e^3} & \frac{6 E_e I_e}{l_e^2} & 0 & -\frac{12 E_e I_e}{l_e^3} & \frac{6 E_e I_e}{l_e^2} \\ 0 & \frac{6 E_e I_e}{l_e^2} & \frac{4 E_e I_e}{l_e} & 0 & -\frac{6 E_e I_e}{l_e^2} & \frac{2 E_e I_e}{l_e} \\ -\frac{E_e A_e}{l_e} & 0 & 0 & \frac{E_e A_e}{l_e} & 0 & 0 \\ 0 & -\frac{12 E_e I_e}{l_e^3} & -\frac{6 E_e I_e}{l_e^2} & 0 & \frac{12 E_e I_e}{l_e^3} & -\frac{6 E_e I_e}{l_e^2} \\ 0 & \frac{6 E_e I_e}{l_e^2} & \frac{2 E_e I_e}{l_e} & 0 & -\frac{6 E_e I_e}{l_e^2} & \frac{4 E_e I_e}{l_e} \end{bmatrix} \begin{bmatrix} u_{i,j} \\ v_{i,j} \\ \varphi_{i,j} \\ u_{j,i} \\ v_{j,i} \\ \varphi_{j,i} \end{bmatrix} \quad (5)$$

Izraze za izračunavanje komponenata lokalne matrice krutosti elementa \mathbf{k}_e možemo gotovo neposredno ugraditi u funkciju u programskom jeziku *Mathematica*, pri čemu se za sažimanje duljine kodâ može iskoristiti simetričnost matrice:

```

ElemKLoc [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 = 12 e i / l^3;
  k23 = 6 e i / l^2;
  k25 = -k22;
  k26 = k23;
  k33 = 4 e i / l;
  k35 = -k23;
  k36 = 2 e i / l;
  k44 = k11;
  k55 = k22;
  k56 = -k23;
  k66 = 4 e i / l;
  k = {{ k11, 0, 0, k14, 0, 0 },
        { 0, k22, k23, 0, k25, k26 },
        { 0, k23, k33, 0, k35, k36 },
        { k14, 0, 0, k44, 0, 0 },
        { 0, k25, k35, 0, k55, k56 },
        { 0, k26, k36, 0, k56, k66 }};
  k
]

```

Realni parametri funkcije¹ su materijalne i geometrijske karakteristike poprečnih presjeka štapnoga elementa — e (modul elastičnosti E_e), a (površina poprečnog presjeka A_e), i (moment tromosti I_e) — te l (duljina elementa ℓ_e).

Listu lokalnih matrica krutosti svih elementa sistema unutar petlje `Do[]`, koja „prolazi” listom elemenata, „slaže” funkcija

```

ListKLoc [els_List, nds_List, echs_List] :=
Module[
  { lk, el, e, a, i, l, j },
  lk = Table [Null, {Length [els]}];

```

¹ U programskom se jeziku *Mathematica* tip parametra funkcije može navesti podvlakom nadovezan na naziv parametra, u našem primjeru *naziv_Real*.

```

Do[
  e1 = els[[j]];
  {e, a, i} = ElemChar [e1, echs];
  l = ElemLength [ElemNds [e1, nds]];
  lk[[j]] = ElemKLoc [e, a, i, l],
  { j, 1, Length [els] }
];
lk
]

```

Njezini su parametri: lista `els` svih elemenata, lista `nds` svih čvorova i lista `echs` s geometrijskim i materijalnim karakteristikama elemenata. U petlji se prije funkcije `ElemKLoc[]` pozivaju funkcije

- `ElemNds [e1_, nds_List] :=`

$$\{ \text{nds}[[\text{e1}[[1]][[1]]]], \text{nds}[[\text{e1}[[1]][[2]]]] \}$$
 koja u listi čvorova `nds` „nalazi” početni i krajnji čvor elementa `e1`,
- `ElemChar [e1_, ech_List] := ech[[e1[[2]]]]`
 koja iz liste `ech` izdvaja karakteristike elementa `e1` te
- `ElemLength [xi_Real, yi_Real, xj_Real, yj_Real] :=`

$$\text{Sqrt} [(xj - xi)^2 + (yj - yi)^2]$$
 koja izračunava duljinu elementa.

Lista čvorova sadrži parove njihovih koordinata, zapisane kao dvočlane liste. Kao primjer ćemo uzeti konstrukciju sa slike 2.; za nju je lista čvorova:

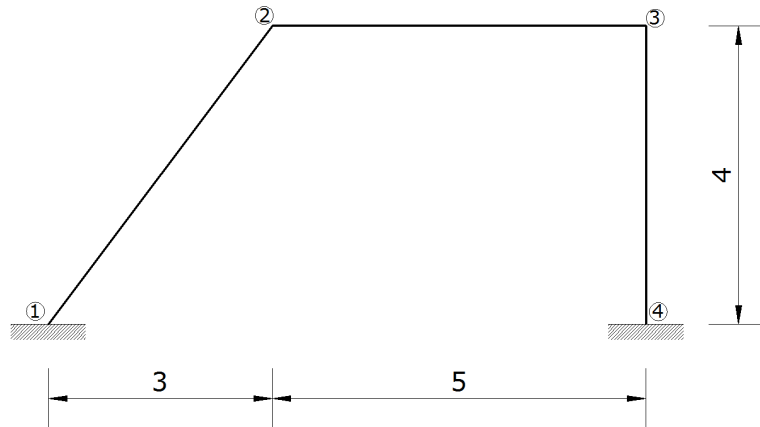
```
nds = { {0., 0.}, {3., 4.}, {8., 4.}, {8., 0.} }
```

Element se opisuje listom čiji je prvi član dvočlana lista indeksâ početnoga i krajnjeg čvora (početni je čvor ishodište lokalnoga desnog koordinatnog sustava, s osi \bar{x} orijentiranom od početnoga do krajnjeg čvora), dok je drugi član indeks uređenog skupa karakteristika elementa u listi karakteristika. Lista je elemenata za naš primjer:

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

Drugi je element, dakle, određen čvorovima 2 i 3, a njegove su karakteristike zadane prvim (i jedinim) skupom u listi

```
chrs = { {3. 10^7, 0.5^2, 0.5^4/12} }
```



Slika 2.

pri čemu su karakteristike, redom, modul elastičnosti ($E = 3 \cdot 10^7 \text{ kN/m}^2$) te površina i moment tromosti poprečnoga presjeka ($b/h = 50/50 \text{ [cm]}$).

Za konstrukciju sa slike 2. pozivom funkcije `ListKLoc []` dobivamo:

$$\mathbf{k}_{(1,2)} = \begin{bmatrix} 1,5 \cdot 10^6 & 0 & 0 & -1,5 \cdot 10^6 & 0 & 0 \\ 0 & 15\,000 & 37\,500 & 0 & -15\,000 & 37\,500 \\ 0 & 37\,500 & 125\,000 & 0 & -37\,500 & 62\,500 \\ -1,5 \cdot 10^6 & 0 & 0 & 1,5 \cdot 10^6 & 0 & 0 \\ 0 & -15\,000 & -37\,500 & 0 & 15\,000 & -37\,500 \\ 0 & 37\,500 & 62\,500 & 0 & -37\,500 & 125\,000 \end{bmatrix},$$

$$\mathbf{k}_{(2,3)} = \begin{bmatrix} 1,5 \cdot 10^6 & 0 & 0 & -1,5 \cdot 10^6 & 0 & 0 \\ 0 & 15\,000 & 37\,500 & 0 & -15\,000 & 37\,500 \\ 0 & 37\,500 & 125\,000 & 0 & -37\,500 & 62\,500 \\ -1,5 \cdot 10^6 & 0 & 0 & 1,5 \cdot 10^6 & 0 & 0 \\ 0 & -15\,000 & -37\,500 & 0 & 15\,000 & -37\,500 \\ 0 & 37\,500 & 62\,500 & 0 & -37\,500 & 125\,000 \end{bmatrix},$$

$$\mathbf{k}_{(3,4)} = \begin{bmatrix} 1,875 \cdot 10^6 & 0 & 0 & -1,875 \cdot 10^6 & 0 & 0 \\ 0 & 29\,296,9 & 58\,593,8 & 0 & -29\,296,9 & 58\,593,8 \\ 0 & 58\,593,8 & 156\,250 & 0 & -58\,593,8 & 78\,125 \\ -1,875 \cdot 10^6 & 0 & 0 & 1,875 \cdot 10^6 & 0 & 0 \\ 0 & -29\,296,9 & -58\,593,8 & 0 & 29\,296,9 & -58\,593,8 \\ 0 & 58\,593,8 & 78\,125 & 0 & -58\,593,8 & 156\,250 \end{bmatrix},$$

Kako su moduli elastičnosti, karakteristike poprečnih presjeka i duljine elemenata (1, 2) i (2, 3) međusobno jednake, jednake su i njihove lokalne matrice krutosti, iako su ti elementi u ravnini različito orijentirani.

Ako postoje opterećenja na štapovima zadanog sistema, tada ona ulaze u jednadžbe ravnoteže u obliku sila i momenata upetosti:

$$\bar{\mathbf{f}}_e = \left[\bar{N}_{i,j} \quad \bar{T}_{i,j} \quad \bar{M}_{i,j} \quad \bar{N}_{j,i} \quad \bar{T}_{j,i} \quad \bar{M}_{j,i} \right]^T. \quad (6)$$

Primjerice, za djelovanje koncentrirane sile s komponentama F_x i F_y (u lokalnom koordinatnom sustavu) i koncentriranoga momenta M u točki s (lokalnom) apscisom a primjenom metode sila dobivamo:

$$\bar{\mathbf{f}}_e = \begin{bmatrix} -\frac{F_x b}{\ell_e} \\ -\frac{F_y (3a + b) b^2}{\ell_e^3} + \frac{6 M a b}{\ell_e^3} \\ -\frac{F_y a b^2}{\ell^2} - \frac{M b (\ell_e - 3a)}{\ell^2} \\ -\frac{F_x a}{\ell_e} \\ -\frac{F_y (a + 3b) a^2}{\ell_e^3} - \frac{6 M a b}{\ell_e^3} \\ \frac{F_y a^2 b}{\ell^2} - \frac{M a (\ell_e - 3b)}{\ell^2} \end{bmatrix} \quad (7)$$

gdje je $b = \ell_e - a$.

Funkcija napisana u jeziku Mathematica sadrži manje–više doslovan prijepis komponentata tog vektora:

```
ConcFL [fx_Real, fy_Real, m_Real, al_Real, l_Real] :=
Module[
  { a, b, f },
  a = al l;
  b = l - a;
  f = Table [0, {6}];
  f[[1]] = -fx b / l;
  f[[4]] = -fx a / l;
  f[[2]] = -fy (3a + b) b^2 / l^3;
  f[[2]] += 6 m a b / l^3;
  f[[5]] = -fy (a + 3b) a^2 / l^3;
  f[[5]] += -6 m a b / l^3;
  f[[3]] = -fy a b^2 / l^2;
  f[[3]] += -m b (l - 3a) / l^2;
```

```

f[[6]] = fy a^2 b / l^2;
f[[6]] += -m a (1 - 3b) / l^2;
f
]

```

Parametri funkcije su realne vrijednosti f_x , f_y i m , tj. sile i momenti koji djeluju na promatrani štapni element. Također imamo realni parametar a_l koji označava relativni položaj sile na elementu, $a_l = a/\ell_e$, te realni parametar l koji označava duljinu elementa. Funkcija kao vrijednost ‚vraća‘ vektor \bar{f}_e .

Koncentrirano djelovanje samo je jedno od mogućih tipova opterećenja štapova. Polimorfno ponašanje, to jest, izračunavanje sila upetosti za različite tipove opterećenja omogućava funkcija

```

FxEndFs [ei_Integer, f_, lp_List, els_List, nds_List] :=
Module[
  { e1, l },
  e1 = Elem [ei, els];
  l = ElemLength [ElemNds [e1, nds]];
  f [Append [lp, l]]
]

```

Parametar f funkcija je koja izračunava sile upetosti za određeni tip opterećenja — primjerice, `ConcFL`. Funkcija `FxEndFs []` izračunava duljinu elementa ei i potom poziva funkciju f s odgovarajućim parametrima, sadržanima u listi lp , kojima je pridodana i netom izračunana duljina elementa. Lista lp , naime, sadrži podatke koji opisuju opterećenje neovisno o elementu na koji djeluje (za koncentrirano djelovanje to su vrijednosti komponenata i relativni položaj hvatišta), ali je za izračunavanje sila upetosti potrebna i duljina tog elementa.

Opterećenja štapova zadaju se u listi. (Na neki štap može, naravno, djelovati više opterećenja.) Svako je opterećenje opisano tročlanom listom: indeksom opterećenog štapa, nazivom funkcije koja izračunava sile upetosti i listom koja sadrži parametre te funkcije (osim duljine elementa).

Funkcija `ListFxEndFsLoc []` sastavlja listu vektorâ sila upetosti za sva opterećenja na elementima pozivajući funkciju `FxEndFs []` za svako opterećenje:

```

ListFxEndFsLoc [elds_List, els_List, nds_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]]],
            FxEndFs [ei, f, lp, els, nds]
            },
{ j, 1, Length [elds] }
];
lef
]

```

Sile upetosti izražene su u lokalnim koordinatnim sustavima pojedinih elemenata.

2.2.2. Prijelaz u globalni koordinatni sustav

Budući da jednadžbe ravnoteže čvorova postavljamo u smjerovima osi globalnog koordinatnog sustava, vektore sila, vektor pomaka i matricu krutosti moramo transformirati iz lokalnog u globalni koordinatni sustav.

Naime, lokalni koordinatni sustav definirali smo za svaki štapni element tako da se ishodište sustava nalazi u početnom čvoru, a lokalna os \bar{x} prolazi kroz krajnji čvor elementa. Globalni i lokalni koordinatni sustav moraju biti ekvivalentni, što znači da poklapanje osi istih oznaka možemo postići translacijom uz rotaciju oko osi z .

Globalni koordinatni sustav služi nam za definiranje geometrije konstrukcije i u odnosu na njega se mjere apsolutni pomaci točaka.

Transformaciju vektora sila na krajevima elementa i vektora pomaka krajeva izvodimo pomoću sljedeće matrice:

$$\mathbf{R}_e = \begin{bmatrix} c_e & s_e & 0 & 0 & 0 & 0 \\ -s_e & c_e & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c_e & s_e & 0 \\ 0 & 0 & 0 & -s_e & c_e & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad (8)$$

gdje su $c_e = \cos \alpha_e$ i $s_e = \sin \alpha_e$, pri čemu je α_e kut koji element zatvara s osi x . Podmatrica

$$\mathbf{r}_e = \begin{bmatrix} c_e & s_e & 0 \\ -s_e & c_e & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

prikazuje rotaciju oko osi z za kut α_e .

Za elemente sistema sa slike 2. matrice transformacije su:

$$\mathbf{R}_{(1,2)} = \begin{bmatrix} 0,6 & 0,8 & 0 & 0 & 0 & 0 \\ -0,8 & 0,6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,6 & 0,8 & 0 \\ 0 & 0 & 0 & -0,8 & 0,6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_{(2,3)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\mathbf{R}_{(3,4)} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Budući da je element (2, 3) horizontalan i da mu je lijevi čvor prvi, orijentacije osi njegovoga lokalnog koordinatnog sustava poklapaju se s orijentacijama osi globalnog sustava, tako da je matrica $\mathbf{R}_{(2,3)}$ jedinična.

Matricu \mathbf{R}_e oblikuje funkcija

```
ElemTG2L [elnds_List] :=
Module[
  { cij, sij, r },
  cij = ElemCos [elnds];
  sij = ElemSin [elnds];
  r = {{ cij, sij, 0, 0, 0, 0 },
       { -sij, cij, 0, 0, 0, 0 },
       { 0, 0, 1, 0, 0, 0 },
       { 0, 0, 0, cij, sij, 0 },
       { 0, 0, 0, -sij, cij, 0 },
       { 0, 0, 0, 0, 0, 1 }};
  r
]
```

Njezin je parametar par parova — dvočlana lista dvočlanih listi — koordinata čvorova elementa. Sinus i kosinus kuta α_e izračunavaju funkcije

```

ElemCos [xi_Real, yi_Real, xj_Real, yj_Real] :=
    (xj - xi) / ElemLength [xi, yi, xj, yj]
ElemSin [xi_Real, yi_Real, xj_Real, yj_Real] :=
    (yj - yi) / ElemLength [xi, yi, xj, yj]

```

Listu matrica transformacija za sve elemente sistema sastavlja funkcija

```

ListTG2L [els_List, nds_List] :=
Module[
    { r, j },
    r = Table [Null, {Length [els]}];
    Do[
        r[[j]] = ElemTG2L [ElemNds [els[[j]], nds]],
        { j, 1, Length [els] }
    ];
    r
]

```

Parametri su liste elemenata i čvorova. Petlja Do[] prolazi listom elemenata; za svaki se element iz liste čvorova funkcijom ElemNds[] izdvajaju njegovi čvorovi te postaju argumentima funkcije ElemTG2L[].

Prelaskom u globalni koordinatni sustav jednadžba (4) postaje

$$\mathbf{f}_e^{(g)} = \mathbf{k}_e^{(g)} \mathbf{u}_{i,j}, \quad (9)$$

gdje su $\mathbf{u}_{i,j}$ vektor koji sadrži pomake čvorova i i j , $\mathbf{f}_e^{(g)}$ vektor komponenata sila na krajevima usporednih s globalnim osima i $\mathbf{k}_e^{(g)}$ matrica krutosti elementa izražena u globalnom koordinatnom sustavu:

$$\mathbf{k}_e^{(g)} = \mathbf{R}_e^{-1} \mathbf{k}_e \mathbf{R}_e = \mathbf{R}_e^T \mathbf{k}_e \mathbf{R}_e; \quad (10)$$

$\mathbf{R}_e^{-1} = \mathbf{R}_e^T$ jer je \mathbf{R}_e ortogonalna matrica.

Globalnu matricu krutosti elementa izračunava funkcija

```

ElemKGlob [k_, r_] := Transpose [r] . k . r

```

čiji su parametri lokalna matrica krutosti i matrica transformacija. Tu funkciju u petlji poziva funkcija ListKGlob[] koja 'slaže' listu globalnih matrica krutosti svih elemenata:

```

ListKGlob [kl_List, r_List] :=
Module[
    { lkg, j },
    lkg = Table [Null, {Length [kl]}];

```



```

Do[
  lkg[[j]] = ElemKGlob [kl[[j]], r[[j]],
    { j, 1, Length [kl] }
];
lkg
]

```

Njezini su parametri, naravno, lista lokalnih matrica krutosti i lista matrica transformacija.

Matrice krutosti elemenata sistema sa slike 2. izražene u globalnom koordinatnom sustavu, su:

$$\mathbf{k}_{(1,2)}^{(g)} = \begin{bmatrix} 549\,600 & 712\,800 & -30\,000 & -549\,600 & -712\,800 & -30\,000 \\ 712\,800 & 965\,400 & 22\,500 & -712\,800 & -965\,400 & 22\,500 \\ -30\,000 & 22\,500 & 125\,000 & 30\,000 & -22\,500 & 62\,500 \\ -549\,600 & -712\,800 & 30\,000 & 549\,600 & 712\,800 & 30\,000 \\ -712\,800 & -965\,400 & -22\,500 & 712\,800 & 965\,400 & -22\,500 \\ -30\,000 & 22\,500 & 62\,500 & 30\,000 & -22\,500 & 125\,000 \end{bmatrix},$$

$$\mathbf{k}_{(2,3)}^{(g)} = \begin{bmatrix} 1,5 \cdot 10^6 & 0 & 0 & -1,5 \cdot 10^6 & 0 & 0 \\ 0 & 15\,000 & 37\,500 & 0 & -15\,000 & 37\,500 \\ 0 & 37\,500 & 125\,000 & 0 & -37\,500 & 62\,500 \\ -1,5 \cdot 10^6 & 0 & 0 & 1,5 \cdot 10^6 & 0 & 0 \\ 0 & -15\,000 & -37\,500 & 0 & 15\,000 & -37\,500 \\ 0 & 37\,500 & 62\,500 & 0 & -37\,500 & 125\,000 \end{bmatrix},$$

$$\mathbf{k}_{(3,4)}^{(g)} = \begin{bmatrix} 29\,296,9 & 0 & 58\,593,8 & -29\,296,9 & 0 & 58\,593,8 \\ 0 & 1,875 \cdot 10^6 & 0 & 0 & -1,875 \cdot 10^6 & 0 \\ 58\,593,8 & 0 & 156,250 & -58\,593,8 & 0 & 78\,125 \\ -29\,296,9 & 0 & -58\,593,8 & 29\,296,9 & 0 & -58\,593,8 \\ 0 & -1,875 \cdot 10^6 & 0 & 0 & 1,875 \cdot 10^6 & 0 \\ 58\,593,8 & 0 & 78\,125 & -58\,593,8 & 0 & 156\,250 \end{bmatrix}.$$

Matrica $\mathbf{R}_{(2,3)}$ je, vidjeli smo, jedinična, pa je $\mathbf{k}_{(2,3)}^{(g)} = \mathbf{k}_{(2,3)}$.

U globalni koordinatni sustav treba prevesti i vektore sila upetosti:

```
FxEndFsGlob [efl_, r_] := Transpose [r] . efl
```

```
ListFxEndFsGlob [efl_List, r_List] :=
```

```
Module[
```

```
{ lefg, ei, j },
```

```
lefg = Table [{0, {0, 0}, Table [0.0, {6}]}, {Length [efl]}];
```

```
Do[
```

```

    ei = ef1[[j]][[1]];
    lefg[[j]] = {ei, ef1[[j]][[2]], FxEndFsGlob [ef1[[j]][[3]], r[[ei]]},
    { j, 1, Length [ef1] }
];
lefg
]

```

2.2.3. Matrica krutosti sistema

Sada možemo komponente matrica krutosti pojedinih elemenata uklopiti u matricu krutosti cijelog sistema.

Preslikavanje indeksâ lokalnih u indekse globalnih stupnjeva slobode ostvareno je pomoću tablice koju oblikuje funkcija `TblNodeDsOF []` uvažavajući pritom ležajne uvjete:

```

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

```

Parametri funkcije su broj čvorova n i lista `cns` kojom su opisani ležajni uvjeti. Komponente te liste su parovi čiji je prvi član indeks ležajnoga čvora, a drugi lista koja sadrži tri broja: ‚0‘ označava da je odgovarajući poopćeni pomak čvora spriječen, a ‚1‘ da je slobodan; pritom su pomaci redom u_i, v_i, φ_i . U našem je primjeru:

```

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

```

Vrijednost je funkcije par koji sadrži ukupni broj stupnjeva slobode konstrukcije i spomenutu tablicu. Svakom čvoru konstrukcije odgovara redak tablice. Tri broja u retku indeksi su poopćenih pomaka čvorova u vektoru nepoznanica \mathbf{i} , ujedno, redni brojevi jednadžbi ravnoteže projekcija sila u čvoru na pravce tih pomaka; ako je neki pomak spriječen, pripadni je broj jednak nuli (odgovarajuća jednadžba ravnoteže ne ulazi u sustav jednadžbi, jer je u njoj nepoznata reakcija). Za konstrukciju sa slike 2. tablica je:

0	0	0
1	2	3
4	5	6
0	0	0

U čvorovima 1 i 4 su upeti ležajevi, pa su sva tri poopćena pomaka tih čvorova spriječena. Pomaci čvora 2, redom u_2 , v_2 i φ_2 , prva su, druga i treća nepoznanica, dok su pomaci čvora 3, u_3 , v_3 , φ_3 , četvrta, peta i šesta nepoznanica; vektor je nepoznanica, prema tome:

$$\mathbf{u} = [u_2 \ v_2 \ \varphi_2 \ u_3 \ v_3 \ \varphi_3]^T.$$

Uz to, prema brojevima u tablici, prva jednadžba sustava jednadžba je ravnoteže projekcija sila, koje djeluju na čvor 2, na os x , druga je jednadžba jednadžba ravnoteže projekcija sila, koje djeluju na isti čvor, na os y , ... , a šesta je jednadžba jednadžba ravnoteže momenata u čvoru 3.

Posljednji od nule različiti broj jednak je ukupnom broju stupnjeva slobode. Brojem stupnjeva slobode određeni su broj nepoznanica i broj jednadžbi ravnoteže.

Matricu krutosti sistema sastavlja funkcija

```
StructK [nd_Integer, di_List, els_List, kl_List, r_List] :=
Module[
  { lkg, sk, ix, i, ii, j },
  lkg = ListKGlob [kl, r];
  sk = Table [0.0, {nd}, {nd}];
  Do[
    ix = ElemDsOF [ElemNdIds [els[[j]]], di];
    Do[
      If [ix[[i]] != 0,
        Do[
          If [ix[[ii]] != 0,
            sk[[ ix[[i]], ix[[ii]] ]] += lkg[[j]][[i, ii]]
          ],
        { ii, 1, 6 }
      ]
    ]
  ]
```

```

        ],
    ],
    { i, 1, 6 }
],
{ j, 1, Length [els] }
];
sk
]

```

Njezini su parametri broj stupnjeva slobode sistema `nd`, spomenuta tablica `di`, lista elemenata `els`, lista matrica krutosti elemenata (u lokalnim koordinatnim sustavima) `kl` i lista matrica transformacije `r`.

Pozivom funkcije `ListKGlob[]` matrice krutosti elemenata izražene u lokalnom koordinatnom sustavu transformiraju se u globalni sustav, a potom se u (vanjskoj) petlji `Do[]` za svaki element (sadržan u listi `els`) pozivom funkcije `ElemDsOF[]` iz tablice `di` izdvajaju indeksi globalnih stupnjeva sloboda početnoga i krajnjeg čvora. Ti indeksi određuju retke i stupce matrice krutosti sistema u presjecišta kojih se u (unutarnjim) ugniježđenim petljama `Do[]` pribrajaju odgovarajuće komponente matrice krutosti elementa (preskačući pritom u grananju `If[]` spriječene pomake).

Primjenom funkcije `StructK[]` za primjer sa slike 2. dobivamo matricu krutosti:

$$\mathbf{K} = \begin{bmatrix} 2,0496 \cdot 10^6 & 712\,800 & 30\,000 & -1,5 \cdot 10^6 & 0 & 0 \\ 712\,800 & 980\,400 & 15\,000 & 0 & -15\,000 & 37\,500 \\ 30\,000 & 15\,000 & 250\,000 & 0 & -37\,500 & 62\,500 \\ -1,5 \cdot 10^6 & 0 & 0 & 1,5293 \cdot 10^6 & 0 & 58\,593,8 \\ 0 & -15\,000 & -37\,500 & 0 & 1,89 \cdot 10^6 & -37\,500 \\ 0 & 37\,500 & 62\,500 & 58\,593,8 & -37\,500 & 281\,250 \end{bmatrix}.$$

2.2.4. Vektor sila u čvorovima

Pomaci čvorova rješenje su sustava jednadžbi ravnoteže čvorova

$$\mathbf{K} \cdot \mathbf{u} = \mathbf{q}. \tag{11}$$

Vektor \mathbf{q} sadrži zadane sile koje djeluju neposredno na čvorove i sile upetosti priključenih elemenata (kako su sile upetosti, kao sile na krajevima štapova, po definiciji sile kojima čvorovi djeluju na štapove, prije pribrajanja komponentama vektora \mathbf{q} treba promijeniti predznake komponenata vektora $\bar{\mathbf{f}}_e^{(g)}$). Redoslijed komponenata u vektoru \mathbf{q} određen je tablicom `di` koju je oblikovala funkcija `TblNodeDsOF[]` (stranica 15).

Vektor \mathbf{q} sastavlja funkcija

```
StructF [nd_Integer, di_List, nlds_List, efsl_List, r_List] :=
Module[
  { lefg, sf, ix, d, nld, i, j },
  sf = Table [0.0, {nd}];
  Do[
    nld = nlds[[j]];
    Do[
      d = di[[nld[[1]], i]];
      If [d != 0,
        sf[[d]] += nld[[2]][[i]]
      ],
      { i, 1, 3 }
    ],
    { j, 1, Length [nlds] }
  ];
  lefg = ListFxEndFsGlob [efsl, r];
  Do[
    ix = ElemDsOF [lefg[[j]][[2]], di];
    Do[
      If [ix[[i]] != 0,
        sf[[ ix[[i]] ]] -= lefg[[j]][[3]][[i]]
      ],
      { i, 1, 6 }
    ],
    { j, 1, Length [lefg] }
  ];
  sf
]
```

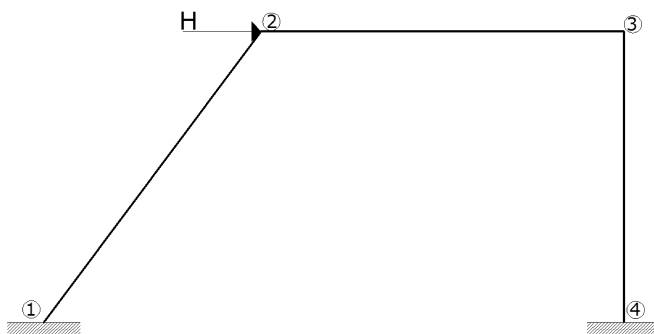
Prva su dva parametra, kao i kod funkcije `StructK[]`, broj stupnjeva slobode `nd` i tablica `di`. Slijede lista čvornih opterećenja, lista vektorâ sila upetosti (izraženih u lokalnom koordinatnom sustavu) i lista matrica transformacija.

Sile koje djeluju u čvorovima pribrajaju se u prvoj petlji `Do[]` komponentama vektora \mathbf{q} (u funkciji prikazanog listom `sf`) koje su određene indeksima u tablici `di`, dok se u drugoj petlji pribrajaju (negativne) sile upetosti prethodno transformirane u globalni koordinatni sustav.

Za primjer proračuna, pretpostavimo da na naš sistem djeluje koncentrirana sila u čvoru 2, $H = 100$ kN (slika 3.). Tada je (jednočlana) lista čvornih opterećenja:

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

Kao što primjer pokazuje, elementi liste su parovi čiji je prvi član indeks čvora u kojem sile djeluju, dok je drugi član lista koja sadrži koncentrirane sile u x i y smjeru i koncentrirani moment.



Slika 3.

Kako niti jedan štap nije opterećen, lista je vektora sila upetosti prazna (stoga pri pozivu funkcije `StructF[]` i lista matrica transformacija može biti prazna):

```
q = StructF [nd, di, lnld, {}, {}]
```

Rezultat je poziva funkcije vektor/lista:

```
{100., 0., 0., 0., 0., 0.}
```

Naime, u tablici na stranici 16 vidimo da je jednačba ravnoteže horizontalnih sila u čvoru 2 prva jednačba sustava (kao što je horizontalni pomak tog čvora prva nepoznanica), pa je sila H prva komponenta vektora \mathbf{q} . Sustav jednačbi ravnoteže je, prema tome:

$$\begin{bmatrix} 2,0496 \cdot 10^6 & 712\,800 & 30\,000 & -1,5 \cdot 10^6 & 0 & 0 \\ 712\,800 & 980\,400 & 15\,000 & 0 & -15\,000 & 37\,500 \\ 30\,000 & 15\,000 & 250\,000 & 0 & -37\,500 & 62\,500 \\ -1,5 \cdot 10^6 & 0 & 0 & 1,5293 \cdot 10^6 & 0 & 58\,593,8 \\ 0 & -15\,000 & -37\,500 & 0 & 1,89 \cdot 10^6 & -37\,500 \\ 0 & 37\,500 & 62\,500 & 58\,593,8 & -37\,500 & 281\,250 \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ \varphi_2 \\ u_3 \\ v_3 \\ \varphi_3 \end{bmatrix} = \begin{bmatrix} 100 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

2.2.5. Rješenje sustava i konačne sile

Sustav rješavamo funkcijom `LinearSolve[]`, 'ugrađenom' u jezik `Mathematica`:

```
u = LinearSolve [K, q]
```

Rješenje je sustava:

$$\mathbf{u} = \begin{bmatrix} 0,001\,785\,19 \\ -0,001\,290\,01 \\ -0,000\,095\,396\,4 \\ 0,001\,757\,7 \\ -0,000\,015\,604\,5 \\ -0,000\,175\,068 \end{bmatrix}.$$

To su pomaci čvorova — izraženi su, dakle, u globalnom koordinatnom sustavu. Za izračunavanje sila na krajevima elemenata iz vektora \mathbf{u} treba za svaki element izdvojiti pomake njegovih čvorova i potom te pomake transformirati u lokalne koordinatne sustave. Listu vektorâ pomaka čvorova elemenata u lokalnim sustavima daje funkcija

```
ListEDisplsLoc [u_List, di_List, els_List, r_List] :=
Module[
  { ldl, ug, j },
  ldl = Table [Table [0.0, {6}], {Length [els]}];
  Do[
    ug = ElemDisplsGlob [els[[j]], di, u];
    ldl[[j]] = r[[j]] . ug,
    { j, 1, Length [els] }
  ];
  ldl
]
```

Funkcija ElemDisplsGlob[] pomoću tablice di izdvaja iz vektora \mathbf{u} pomake čvorova određenoga štapa. Transformacija pomakâ iz globalnoga u lokalni koordinatni sustav je:

$$\mathbf{u}_e = \mathbf{R}_e \mathbf{u}_{i,j}. \quad (12)$$

Konačne sile na krajevima elemenata zbrojevi su sila zbog pomakâ krajeva i sila upe-
tosti:

$$\hat{\mathbf{f}}_e = \mathbf{f}_e + \bar{\mathbf{f}}_e = \mathbf{k}_e \mathbf{u}_e + \bar{\mathbf{f}}_e \quad (13)$$

(u našem je primjeru $\bar{\mathbf{f}}_e = \mathbf{0}$, pa je $\hat{\mathbf{f}}_e = \mathbf{k}_e \mathbf{u}_e$). Listu vektorâ konačnih sila izračunava funkcija

```
ListEndForces [kl_List, ul_List, efsl_List] :=
Module[
  { lefs, j },
  lefs = Table [Table [0.0, {6}], {Length [ul]}];
  Do[
```

```

    lefs[[j]] = kl[[j]] . ul[[j]],
    { j, 1, Length [kl] }
];
Do[
    lefs[[ efs1[[j]][[1]] ]] += efs1[[j]][[3]],
    { j, 1, Length [efs1] }
];
lefs
]

```

Parametri funkcije su lista lokalnih matrica krutosti, lista vektorâ pomakâ čvorova elementa (izraženih u lokalnom koordinatnom sustavu) i lista vektorâ sila upetosti. U prvoj se petlji Do[] za svaki štap izračunavaju sile na krajevima zbog pomakâ čvorova ($\mathbf{f}_e = \mathbf{k}_e \mathbf{u}_e$), a u drugoj se petlji tim silama pribrajaju sile upetosti (ako postoje).

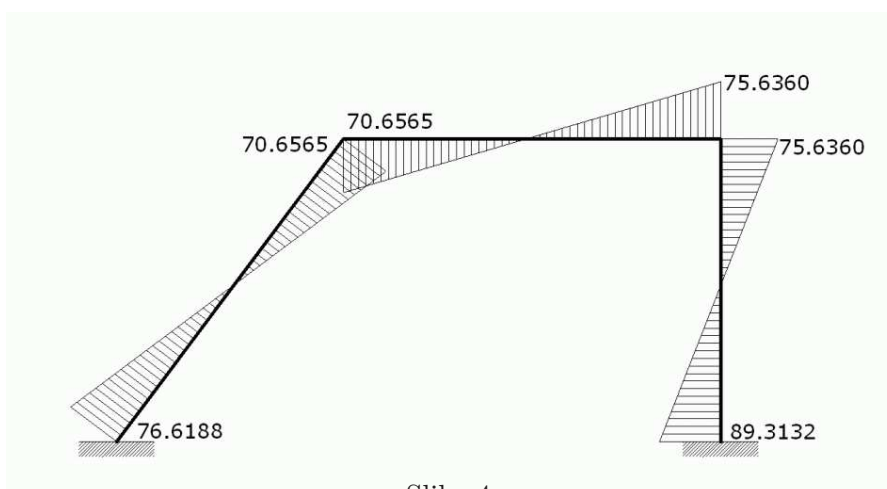
Pozivom

```
ListEndForces [lkl, ldl, {}]
```

za naš primjer dobivamo

$$\begin{bmatrix} N_{1,2} \\ T_{1,2} \\ M_{1,2} \\ N_{2,1} \\ T_{2,1} \\ M_{2,1} \end{bmatrix} = \begin{bmatrix} -58,6644 \\ 29,4551 \\ 76,6188 \\ 58,6644 \\ -29,4551 \\ 70,6565 \end{bmatrix}, \quad \begin{bmatrix} N_{2,3} \\ T_{2,3} \\ M_{2,3} \\ N_{3,2} \\ T_{3,2} \\ M_{3,2} \end{bmatrix} = \begin{bmatrix} 41,2373 \\ -29,2585 \\ -70,6565 \\ -41,2373 \\ 29,2585 \\ -75,6360 \end{bmatrix}, \quad \begin{bmatrix} N_{3,4} \\ T_{3,4} \\ M_{3,4} \\ N_{4,3} \\ T_{4,3} \\ M_{4,3} \end{bmatrix} = \begin{bmatrix} 29,2585 \\ 41,2373 \\ 75,6360 \\ -29,2585 \\ -41,2373 \\ 89,3132 \end{bmatrix},$$

pa možemo nacrtati dijagrame unutarnjih sila; primjerice, momentni je dijagram:



Slika 4.

3. Inženjerska metoda pomakâ

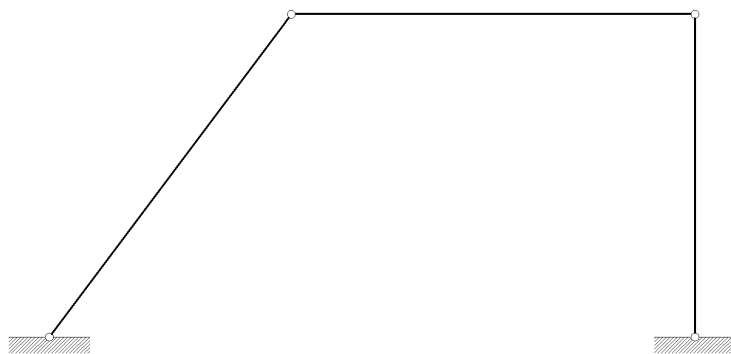
Inženjerska metoda pomakâ temelji se na pretpostavci da možemo zanemariti utjecaj uzdužnih sila, te time bitno smanjiti broj nepoznanica i, u odnosu na opću metodu pomaka i metodu sila, olakšati ‚ručno‘ rješavanje različitih sistema [1, 9].

Uzmimo za primjer sistem koji smo već riješili općom metodom pomaka (slika 2. na stranici 8).

U općoj metodi pomaka naš sistem ima šest nepoznanica: horizontalni i vertikalni pomak čvora 2 i kut zaokreta čvora 2, te horizontalni i vertikalni pomak čvora 3 i kut zaokreta čvora 3. Poslužimo li se navedenom pretpostavkom i zanemarimo uzdužne deformacije, broj nepoznanica smanjit ćemo na tri — to su zaokreti čvorova 2 i 3 te zajednički horizontalni pomak čvorova 2 i 3 (čvor 2, u stvari, ‚putuje‘ po okomici na os elementa 1–2, ali je horizontalna komponenta tog pomaka, kao što ćemo uskoro vidjeti, jednaka pomaku čvora 3).

U inženjerskoj su metodi pomaka nepoznanice kutovi zaokreta te *neovisni translacijski pomaci* čvorova, pri čemu vrijedi da je broj zaokreta jednak broju krutih čvorova, no broj neovisnih translacijskih pomaka nije jednak broju krutih i zglobnih čvorova. Njihov broj određujemo pomoću zglobne sheme sistema.

Kao što je na slici 5. prikazano, *zglobnu shemu* oblikujemo tako da sve krute čvorne veze (uključujući i ležajne) pretvorimo u zglobne. Broj neovisnih translacijskih pomaka sistema jednak je broju stupnjeva slobode zglobne sheme.



Slika 5.

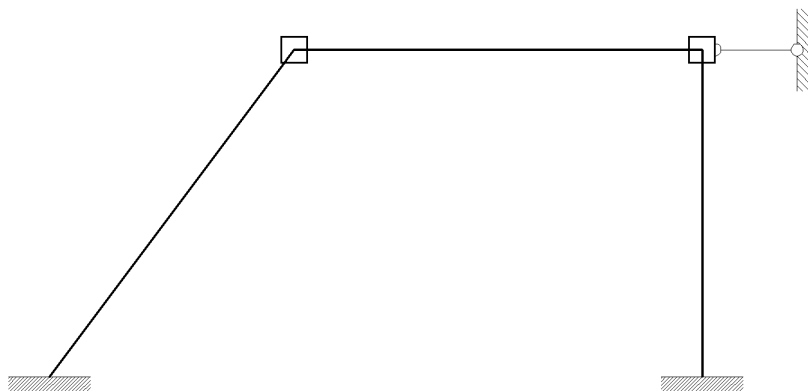
Naš sistem ima barem jedan neovisni translacijski pomak:

$$S = n_{\check{c}} \cdot 2 - n_{\check{s}} = 2 \cdot 2 - 3 = 4 - 3 = 1.$$

No, spriječimo li pomak čvora 2 (ili pomak čvora 3), dobit ćemo geometrijski nepromjenjivi sistem, pa možemo zaključiti da postoji samo jedan neovisni pomak.

Sisteme možemo razvrstati na pomične i nepomične, ovisno o tome jesu li translacijski pomaci mogući ili nemogući. Naš sistem je pomičan sistem, nepoznanice su mu kutovi zaokreta krutih čvorova (φ_2 i φ_3) te jedan neovisni translacijski pomak ($u_{2,3}$).

U prvom koraku, kao i u općoj metodi pomaka, sprječavamo moguće pomake — u čvorovima 2 i 3 dodajemo momentne spojeve s podlogom koji sprečavaju zaokrete tih čvorova, te dodajemo vezu koja sprječava odabrani translacijski pomak (slika 6.).



Slika 6.

Kao neovisni translacijski pomak odabrali smo pomak čvora 3, $u_{2,3}$, te su nam tako pomaci ostalih točaka određeni skicom pomaka (slika 7.).

Za razliku od nepomičnih sistema, kod kojih za momente na krajevima elemenata vrijede izrazi:

$$m_{i,j} = 4 k_{i,j} \varphi_i + 2 k_{i,j} \varphi_j, \quad (14)$$

$$m_{j,i} = 2 k_{i,j} \varphi_i + 4 k_{i,j} \varphi_j, \quad (15)$$

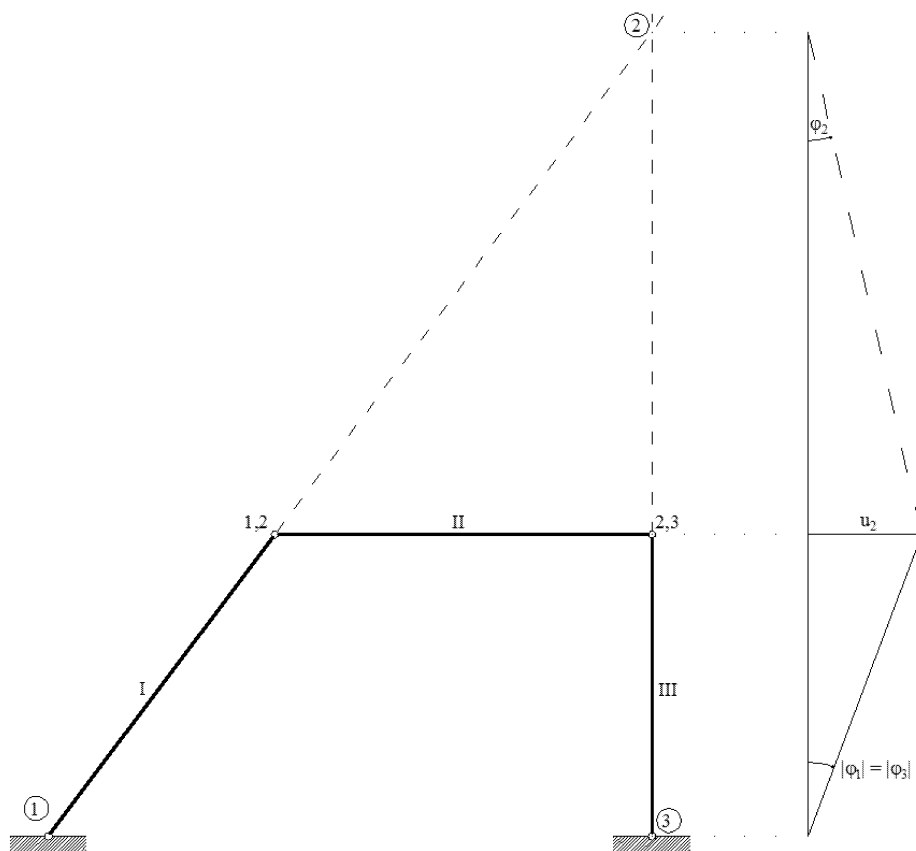
kod pomičnih sistema u obzir treba uzeti i kut zaokreta krutog tijela (slika 8.)

$$\psi_{i,j} = \frac{\Delta v_{i,j}}{l_{i,j}} \quad (16)$$

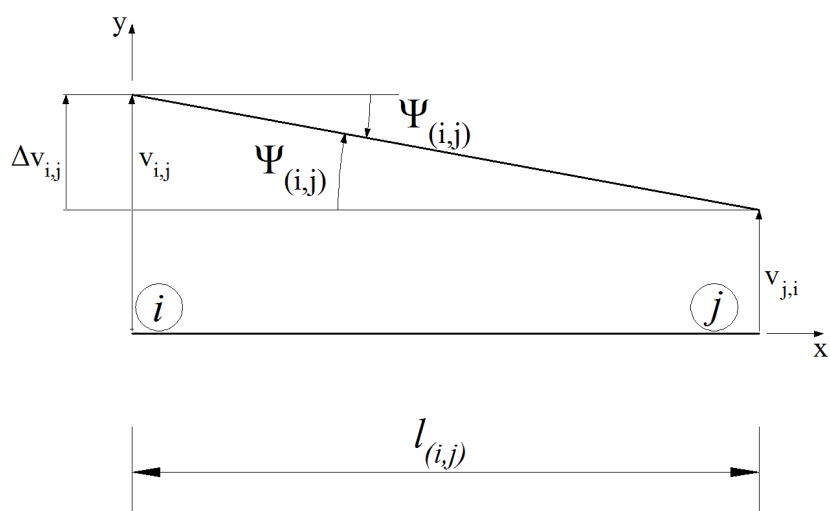
te izrazi za momente na krajevima elemenata poprimaju oblik

$$m_{i,j} = 4 k_{i,j} \varphi_i + 2 k_{i,j} \varphi_j - 6 k_{i,j} \psi_{i,j}, \quad (17)$$

$$m_{j,i} = 2 k_{i,j} \varphi_i + 4 k_{i,j} \varphi_j - 6 k_{i,j} \psi_{i,j}. \quad (18)$$



Slika 7.



Slika 8.

Zaokreti elemenata kao krutih tijela funkcije su neovisnih translacijskih pomaka, pa iz skice pomaka možemo očitati:

$$\begin{aligned}\psi_{1,2} &= \psi_I = -\frac{u_{2,3}}{4}, \\ \psi_{2,3} &= \psi_{II} = \frac{3 \cdot u_{2,3}}{20}, \\ \psi_{3,4} &= \psi_{III} = -\frac{u_{2,3}}{4}.\end{aligned}$$

Uvrstimo li to u gornje jednadžbe (17) i (18), izrazi za momente na krajevima elemenata za naš primjer glase (uz $\varphi_1 = \varphi_4 = 0$):

$$\begin{aligned}m_{1,2} &= 2 k_{1,2} \varphi_2 + \frac{3}{2} k_{1,2} u_{2,3}, \\ m_{2,1} &= 4 k_{1,2} \varphi_2 + \frac{3}{2} k_{1,2} u_{2,3}, \\ m_{2,3} &= 4 k_{2,3} \varphi_2 + 2 k_{2,3} \varphi_3 - \frac{9}{10} k_{2,3} u_{2,3}, \\ m_{3,2} &= 2 k_{2,3} \varphi_2 + 4 k_{2,3} \varphi_3 - \frac{9}{10} k_{2,3} u_{2,3}, \\ m_{3,4} &= 4 k_{3,4} \varphi_3 + \frac{3}{2} k_{3,4} u_{2,3}, \\ m_{4,3} &= 2 k_{3,4} \varphi_2 + \frac{3}{2} k_{3,4} u_{2,3}.\end{aligned}$$

Budući da su ukupne vrijednosti momenata na krajevima elemenata zbroj momenata upetosti i momenata izazvanih zaokretima čvorova i elemenata,

$$M_{i,j} = m_{i,j} + \overline{M}_{i,j}, \quad (19)$$

$$M_{j,i} = m_{j,i} + \overline{M}_{j,i}, \quad (20)$$

preostaje nam izračunati momente upetosti. Na naš sistem, međutim, djeluje samo sila $H = 100 \text{ kN}$ u čvoru 2 (slika 3. na stranici 19), pa vrijedi

$$\overline{M}_{1,2} = \overline{M}_{2,1} = \overline{M}_{2,3} = \overline{M}_{3,2} = \overline{M}_{3,4} = \overline{M}_{4,3} = 0,$$

tako da su konačni izrazi za ukupne momente na krajevima elemenata:

$$\begin{aligned}M_{1,2} &= m_{1,2} = 2 k_{1,2} \varphi_2 + \frac{3}{2} k_{1,2} u_{2,3}, \\ M_{2,1} &= m_{2,1} = 4 k_{1,2} \varphi_2 + \frac{3}{2} k_{1,2} u_{2,3}, \\ M_{2,3} &= m_{2,3} = 4 k_{2,3} \varphi_2 + 2 k_{2,3} \varphi_3 - \frac{9}{10} k_{2,3} u_{2,3},\end{aligned}$$

$$M_{3,2} = m_{3,2} = 2 k_{2,3} \varphi_2 + 4 k_{2,3} \varphi_3 - \frac{9}{10} k_{2,3} u_{2,3},$$

$$M_{3,4} = m_{3,4} = 4 k_{3,4} \varphi_3 + \frac{3}{2} k_{3,4} u_{2,3},$$

$$M_{4,3} = m_{4,3} = 2 k_{3,4} \varphi_2 + \frac{3}{2} k_{3,4} u_{2,3}.$$

Za proračun vrijednosti nepoznanica potrebne su nam tri jednačbe: jednačbe ravnoteže momenata u čvorovima 2 i 3 te jednačba teorema o virtualnim pomacima.

Jednačba ravnoteže za čvor 2 glasi:

$$-M_{2,1} - M_{2,3} = 0,$$

za čvor 3:

$$-M_{3,2} - M_{3,4} = 0,$$

a teorem o virtualnim pomacima daje jednačbu:

$$(M_{1,2} + M_{2,1}) \cdot \psi_{1,2}^{(1)} + (M_{2,3} + M_{3,2}) \cdot \psi_{2,3}^{(1)} + (M_{3,4} + M_{4,3}) \cdot \psi_{3,4}^{(1)} + 100 \cdot u_{2,3}^{(1)} = 0,$$

gdje gornjim indeksom (1) označavamo da se govori o radu na jediničnom pomaku, $u_{2,3}^{(1)} = 1$, i kutovima izazvanima tim jediničnim pomakom.

Uvrštavanjem zadanih vrijednosti i sređivanjem dobivamo sustav jednačbi:

$$\begin{aligned} 250\,000 \cdot \varphi_2 + 62\,500 \cdot \varphi_3 + 18\,750 \cdot u_{2,3} &= 0, \\ 62\,500 \cdot \varphi_2 + 281\,250 \cdot \varphi_3 + 30\,468,8 \cdot u_{2,3} &= 0, \\ 18\,750 \cdot \varphi_2 + 30\,468,8 \cdot \varphi_3 + 61\,171,9 \cdot u_{2,3} &= 100. \end{aligned} \tag{21}$$

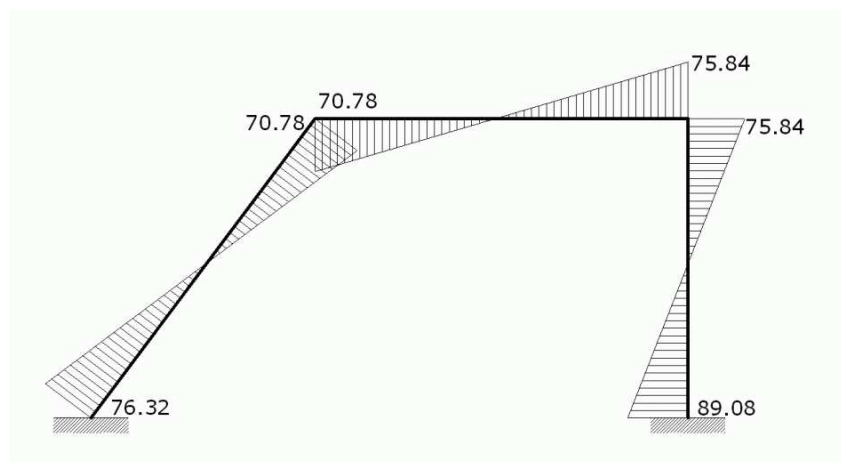
Njegova su rješenja:

$$\begin{aligned} \varphi_2 &= -0,000\,088\,6, \\ \varphi_3 &= -0,000\,169\,5, \\ u_{2,3} &= 0,001\,746\,3, \end{aligned}$$

pa su konačne vrijednosti momenata:

$$\begin{aligned} M_{1,2} &= 76,32 \text{ kNm}, \\ M_{2,1} &= 70,78 \text{ kNm}, \\ M_{2,3} &= -70,78 \text{ kNm}, \\ M_{3,2} &= -75,84 \text{ kNm}, \\ M_{3,4} &= 75,84 \text{ kNm}, \\ M_{4,3} &= 89,08 \text{ kNm}. \end{aligned}$$

Pripadni momentni dijagram ima sljedeći oblik:



Slika 9.

U tablici 1. uspoređene vrijednosti momenata dobivenih općom i inženjerskom metodom pomakâ.

Tablica 1.

	OMP [kNm]	IMP [kNm]	ε [%]
$M_{1,2}$	76,62	76,32	0,336
$M_{2,1}$	70,66	70,78	0,134
$M_{2,3}$	-70,66	-70,78	0,134
$M_{3,2}$	-75,64	-75,84	0,224
$M_{3,4}$	75,64	75,84	0,224
$M_{4,3}$	89,31	89,08	0,258

Relativne pogreške u posljednjem stupcu izračunane su prema izrazu

$$\varepsilon = \frac{|M_{i,j}^{\text{OMP}} - M_{i,j}^{\text{IMP}}|}{|M_{\text{max}}^{\text{OMP}}|} \cdot 100 [\%] = \frac{|M_{i,j}^{\text{OMP}} - M_{i,j}^{\text{IMP}}|}{89,31} \cdot 100 [\%].$$

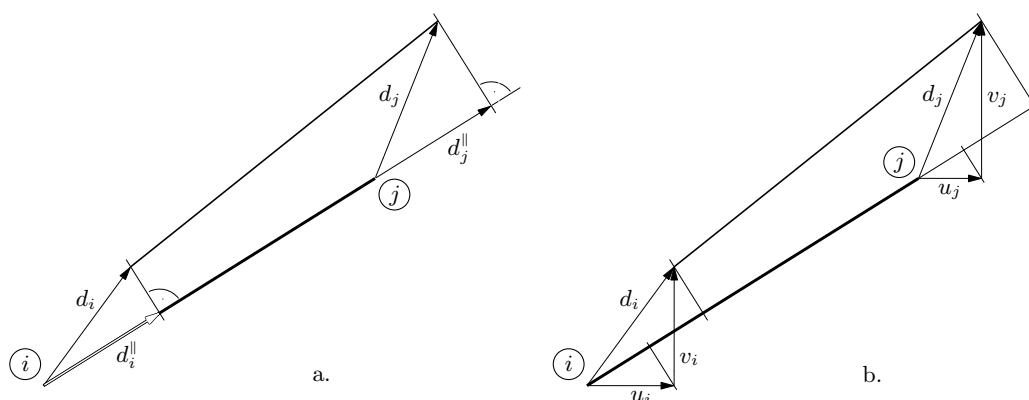
4. Kinematička ograničenja

Geometrijski uvjeti koje poopćeni pomaci, kao glavne nepoznanice u metodama pomakâ, moraju zadovoljiti, nazivaju se *kinematička ograničenja*. To mogu biti zadane komponente translacijskih pomaka ili zadani kutovi zaokreta ili pak propisani odnosi nekih pomaka i zaokreta [4, 10].

4.1. Formalni izraz pretpostavke inženjerske metode pomakâ

Uz pretpostavku da su štapni elementi u uzdužnom smjeru apsolutno kruti (što je, znamo, osnovna pretpostavka inženjerske metode pomaka), iz *osnovnog teorema kinematike nedeformabilnog tijela*, koji kaže da su projekcije vektora brzina bilo kojih dviju točaka nedeformabilnog tijela na os određenu njihovom spojnicom jednake [14], prema slici 10.a. slijedi [10]:

$$\vec{d}_i^{\parallel} = \vec{d}_j^{\parallel}. \quad (22)$$



Slika 10.

Možemo pisati

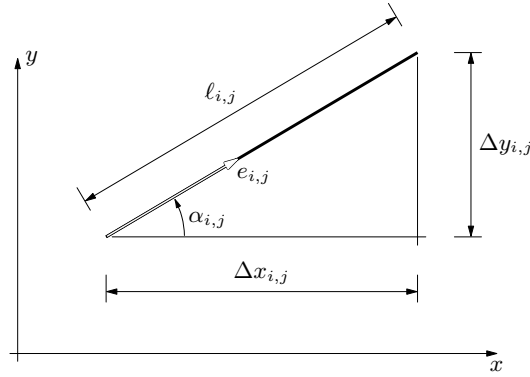
$$\vec{d}_i^{\parallel} = d_i^{\parallel} \vec{e}_{i,j} \quad \text{i} \quad \vec{d}_j^{\parallel} = d_j^{\parallel} \vec{e}_{i,j}, \quad (23)$$

gdje je $\vec{e}_{i,j}$ jedinični vektor na osi štapa (slika 11.) koji možemo prikazati u sljedećem obliku:

$$\vec{e}_{i,j} = \frac{1}{l_{i,j}} [(x_j - x_i) \vec{i} + (y_j - y_i) \vec{j}] = \frac{\Delta x_{i,j}}{l_{i,j}} \vec{i} + \frac{\Delta y_{i,j}}{l_{i,j}} \vec{j} = \cos \alpha_{i,j} \vec{i} + \sin \alpha_{i,j} \vec{j}. \quad (24)$$

Iz vektorske jednakosti (22) uz izraze (23) slijedi skalarni uvjet

$$d_i^{\parallel} = d_j^{\parallel}. \quad (25)$$



Slika 11.

Pomake krajeva možemo prikazati u komponentnom obliku

$$\vec{d}_i = \vec{u}_i + \vec{v}_i = u_i \vec{i} + v_i \vec{j} \quad \text{i} \quad \vec{d}_j = \vec{u}_j + \vec{v}_j = u_j \vec{i} + v_j \vec{j}, \quad (26)$$

pa su

$$d_i^{\parallel} = \vec{d}_i \cdot \vec{e}_{i,j} = u_i \cos \alpha_{i,j} + v_i \sin \alpha_{i,j} \quad \text{i} \quad d_j^{\parallel} = \vec{d}_j \cdot \vec{e}_{i,j} = u_j \cos \alpha_{i,j} + v_j \sin \alpha_{i,j}, \quad (27)$$

tako da uvjet (25) daje *jednadžbu kinematičkog ograničenja*

$$u_i \cos \alpha_{i,j} + v_i \sin \alpha_{i,j} - u_j \cos \alpha_{i,j} - v_j \sin \alpha_{i,j} = 0. \quad (28)$$

4.2. Sustav jednadžbi kinematičkih ograničenja

U sljedećem primjeru (slika 12.), sistem se sastoji od tri elementa, te možemo napisati tri jednadžbe kinematičkih ograničenja.

Ako uvedemo oznake

$$c_{i,j} = \cos \alpha_{i,j} = \frac{x_j - x_i}{l_{i,j}} \quad \text{i} \quad s_{i,j} = \sin \alpha_{i,j} = \frac{y_j - y_i}{l_{i,j}}, \quad (29)$$

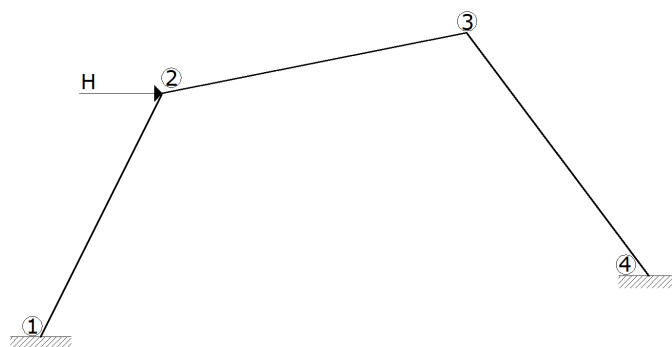
osnovni oblik jednadžbe kinematičkog ograničenja (28) poprima (sažetiji) oblik

$$c_{i,j} \cdot u_i + s_{i,j} \cdot v_i - c_{i,j} \cdot u_j - s_{i,j} \cdot v_j = 0. \quad (30)$$

Izraze (29) izračunavaju funkcije ElemCos[] i ElemSin[] koje smo uveli na stranici 12.

Jednadžbe kinematičkih ograničenja za naš primjer su:

$$\begin{aligned} c_{1,2} u_2 + s_{1,2} v_2 &= 0, \\ c_{2,3} u_2 + s_{2,3} v_2 - c_{2,3} u_3 - s_{2,3} v_3 &= 0, \\ c_{4,3} u_3 + s_{4,3} v_3 &= 0. \end{aligned} \quad (31)$$



Slika 12.

Zapišemo li ih matrično,

$$\begin{bmatrix} c_{1,2} & s_{1,2} & 0 & 0 \\ c_{2,3} & s_{2,3} & -c_{2,3} & -s_{2,3} \\ 0 & 0 & c_{3,4} & s_{3,4} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (32)$$

postaje očitim da se radi o sustavu tri jednađbe sa četiri nepoznanice. Broj redaka matrice sustava jednak je broju jednađbi, koji je pak jednak broju štapnih elemenata, dok je broj stupaca jednak broju nepoznanica — ukupnom broju translacijskih pomaka „slobodnih čvorova”.

Matricu iz izraza (32), koju možemo nazvati *matricom kinematičkih ograničenja*, oblikujemo funkcijom

```
KinCnsMtx [ntd_Integer, tdi_List, els_List, nds_List] :=
Module[
  { cnstr, elnds, c, s, ni1, ni2, i, j },
  cnstr = Table [0., {Length [els]}, {ntd}];
  Do[
    elnds = ElemNds [els[[j]], nds];
    c = ElemCos [elnds];
    s = ElemSin [elnds];
    { ni1, ni2 } = ElemNdIds [els[[j]]];
    i = tdi[[ni1, 1]];
    If [i != 0,
      cnstr [[j, i]] = c
    ];
    i = tdi[[ni1, 2]];
    If [i != 0,
      cnstr [[j, i]] = s
    ];
  ];
];
```

```

    i = tdi[[ni2, 1]];
    If [i != 0,
        cnstr [[j, i]] = -c
    ];
    i = tdi[[ni2, 2]];
    If [i != 0,
        cnstr [[j, i]] = -s
    ],
    { j, 1, Length [els] }
];
cnstr
]

```

Parametri `els` i `nds` liste su elemenata i čvorova konstrukcije. Prva dva parametra, `ntd` i `tdi`, broj su i tablica indeksa translacijskih stupnjeva slobode (bez zaokreta) u jednadžbama kinematičkih ograničenja. Tablica `tdi` analogna je na stranici 15 opisanoj tablici `di` pomoću koje u oblikovanju matrice krutosti sistema preslikavamo indekse lokalnih u indekse globalnih stupnjeva slobode. Tablicu `tdi` iz tablice `di` oblikuje funkcija

```

TranslDsOF [di_List] :=
Module[
    { ntd, tdi, nti },
    tdi = {};
    ntd = 0;
    Do[
        nti = { 0, 0 };
        If [di[[j, 1]] != 0,
            ++ntd; nti[[1]] = ntd
        ];
        If [di[[j, 2]] != 0,
            ++ntd; nti[[2]] = ntd
        ];
        AppendTo [tdi, nti],
        { j, 1, Length [di] }
    ];
    { ntd, tdi }
]

```

Uz tablicu `tdi` funkcija ,vraća' i ukupan broj translacijskih pomaka. U našem primjeru pozivom

```
{ ntd, tdi } = TranslDsOF [di]
```

dobivamo

$$\{4, \{\{0, 0\}, \{1, 2\}, \{3, 4\}, \{0, 0\}\}\}$$

Gauss–Jordanovim eliminacijskim postupkom [6, 12] sustav jednadžbi (32) prevodimo u ekvivalentni sustav

$$\begin{bmatrix} 1 & 0 & 0 & -\frac{s_{1,2}(c_{2,3}s_{3,4} - c_{3,4}s_{2,3})}{c_{3,4}(c_{1,2}s_{2,3} - c_{2,3}s_{1,2})} \\ 0 & 1 & 0 & \frac{c_{1,2}(c_{2,3}s_{3,4} - c_{3,4}s_{2,3})}{c_{3,4}(c_{1,2}s_{2,3} - c_{2,3}s_{1,2})} \\ 0 & 0 & 1 & \frac{s_{3,4}}{c_{3,4}} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix},$$

ili, u sažetijem zapisu, u kojem je istaknuta struktura matrice sustava:

$$\begin{bmatrix} 1 & 0 & 0 & h_{1,4} \\ 0 & 1 & 0 & h_{2,4} \\ 0 & 0 & 1 & h_{3,4} \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (33)$$

(pretpostavlja se da je $c_{3,4} \neq 0$). Za matricu sustava u (33) kažemo da je u *Hermiteovom normalnom obliku* ili *reduciranom obliku* [12], a često ćemo govoriti jednostavno o *reduciranoj matrici sustava*. Prva komponenta u svakom retku takve matrice, koja je različita od nule, ima vrijednost 1; tu komponentu nazivamo *uporišnom komponentom*. Stupac s uporišnom komponentom nazivamo *baznim stupcem*; u njemu sve komponente, osim uporišne, imaju vrijednost 0. U našem su primjeru prva tri stupca bazna.

Hermiteov oblik matrice izračunavamo pomoću u jezik Mathematica ‘ugrađene’ funkcije RowReduce []:

```
ReducedKinCnsMtx [mkc_List] := RowReduce [mkc]
```

Pomoću Hermiteovog oblika možemo odrediti broj neovisnih translacijskih pomaka te ih automatski ‘prepoznati’. Iz reduciranoga sustava, naime, slijedi:

$$\begin{aligned} u_2 &= -h_{1,4} \cdot v_3, \\ v_2 &= -h_{2,4} \cdot v_3, \\ u_3 &= -h_{3,4} \cdot v_3, \end{aligned}$$

pa vidimo da su pomaci u_2 , v_2 i u_3 izraženi pomoću vertikalnoga pomaka čvora 3, v_3 . Taj pomak je, prema tome, automatski ‘odabran’ kao neovisni translacijski pomak. Kako su navedenim izrazima pomaci u_2 , v_2 i u_3 određeni ako je pomak v_3 poznat, v_3 se naziva *vodećim pomakom*, a ostali su pomaci *prateći*.

Napišemo li prethodno rješenje u matricnom obliku,

$$\begin{bmatrix} u_2 \\ v_2 \\ u_3 \end{bmatrix} = -v_3 \begin{bmatrix} h_{1,4} \\ h_{2,4} \\ h_{3,4} \end{bmatrix} = -v_3 \mathbf{h}_{.,4}, \quad (34)$$

možemo zaključiti da je broj neovisnih translacijskih pomaka jednak broju stupaca reducirane matrice koji nisu bazni, da neovisni pomaci odgovaraju tim stupcima i da su komponente tih stupaca (u našem primjeru $h_{1,4}$, $h_{2,4}$ i $h_{3,4}$) koeficijenti koji određuju ovisnost pratećih pomaka (u_2 , v_2 , u_3) o vodećima.

Funkcija `Masters[]` u reduciranoj matrici kinematičkih ograničenja (parametar `rmkc`) ,pronalazi' vodeće pomake:

```
Masters [rmkc_List, tol_Real : 10.^-13] :=
Module[
  { lms, nr, nc, ei, i, j },
  { nr, nc } = Dimensions [rmkc];
  lms = {};
  i = 1;
  Do[
    If [Equal [Chop [Take [rmkc, {1, nr}, {j, j}], tol],
             ColEi [nr, i]
            ],
        If [i < nr, ++i],
        lms = { lms, j }
    ],
    { j, 1, nc }
  ];
  Flatten [lms]
]
```

Rezultat je lista indeksa vodećih pomaka (u našem primjeru lista sadrži samo jedan indeks: {4}). Drugi je parametar funkcije realna vrijednost `tol`. Naime, kako su komponente stupaca reducirane matrice rezultati niza numeričkih operacija sa zaokruživanjima, ispitivanje jednakosti s komponentama 'jediničnih stupaca' (oblikuje ih funkcija `ColEi[]`) treba provesti s određenom tolerancijom [13] (pomoću funkcije `Chop[]` jezika `Mathematica`).

Za nastavak postupka pogodno je rješenju (34) dodati neovisni pomak i napisati ga u obliku

$$\begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} -h_{1,4} \\ -h_{2,4} \\ -h_{3,4} \\ 1 \end{bmatrix} [v_3] = \mathbf{H} [v_3]. \quad (35)$$

Matrica **H** sadrži stupce reducirane matrice koji nisu bazni, s promijenjenim predznacima komponentata i jedinicima dodanim na mjestima koja odgovaraju vodećim pomacima. Tu matricu oblikuje funkcija

```
SolveFromRCnsMtx [rmkc_List, ms_List] :=
Module[
  { cc, nm, nt, ii, jj, i, j },
  nm = Length [ms];
  nt = Dimensions [rmkc][[2]];
  cc = Table [0.0, {nt}, {nm}];
  ii = 1;
  jj = 1;
  Do[
    If [j == ms[[ii]]
      ,
      cc[[j, ii]] = 1.0;
      If [ii < nm, ++ii]
      ,
      Do [
        cc[[j, i]] = -rmkc[[ jj, ms[[i]] ]],
        { i, 1, nm }
      ];
      ++jj
    ],
    { j, 1, nt }
  ];
  cc
]
```

Parametri su reducirana matrica i lista indeksa vodećih pomaka.

Kao što smo rekli, funkcija `Masters[]` određuje vodeće pomake na temelju strukture reducirane matrice. No, vodeće pomake možemo i sami izabrati. Matricu ovisnosti pratećih pomaka o vodećima (analogon matrici **H**) izračunava tada funkcija

```
SolveKinCnsEqs [eqs_List, ms_List, uu_List, tol_ : 10.^-13] :=
Module[
  { cc, nm, nu, ls, v, vv, subs, subs1, i, j },
  ls = Slaves [ms, uu];
  v = Solve [eqs, ls];
  vv = uu /. v[[1]];

```

```

nm = Length [ms];
subs = {};
Do [
  subs = {subs, uu[[ ms[[i]] ]] -> 0.},
  { i, 1, nm }
];
subs = Flatten [subs];

nu = Length [uu];
cc = Table [0.0, {nu}, {nm}];
Do [
  Do [
    cc[[j, i]] = (vv[[j]] /.
      ReplacePart [subs, uu[[ ms[[i]] ]] -> 1., i)),
    { j, 1, nu }
  ],
  { i, 1, nm }
];
Chop [cc, tol]
]

```

Njezina je ‚jezgra‘ poziv funkcije `Solve[]` jezika `Mathematica` kojoj se kao (drugi) parametar zadaje lista nepoznanica, tako da su rješenja izražena kao (linearne) funkcije ostalih simbola koji se u jednadžbama pojavljuju. Proglasimo li u našem primjeru u_2 vodećim pomakom, lista će nepoznanica biti $\{v_2, u_3, v_3\}$, dok će u_2 biti shvaćen ‚poznatim‘ simbolom, tako da će rješenje biti

$$\begin{aligned}
v_2 &= \bar{h}_2 \cdot u_2, \\
u_3 &= \bar{h}_3 \cdot u_2, \\
v_3 &= \bar{h}_4 \cdot u_2,
\end{aligned}$$

odnosno, ‚dopunimo‘ li, kao i ranije, rješenje vodećim pomakom i napišemo li ga u matričnom obliku:

$$\begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 \\ \bar{h}_2 \\ \bar{h}_3 \\ \bar{h}_4 \end{bmatrix} [u_2] = \bar{\mathbf{H}} [u_2]. \quad (36)$$

Prvi parametar funkcije `SolveKinCnsEqs[]` je sustav jednadžbi kinematičkih ograničenja eqs koji treba prethodno oblikovati funkcijom

```

KinCnsEqs [mc_List, uu_List] :=
  Thread [mc . uu == Table [0.0, {Length [mc]}]]

```

Parametri `mc` i `uu` su, redom, matrica kinematičkih ograničenja i lista simbola koji će biti upotrijebljeni kao nepoznanice. (Posebno uvođenje matrice ograničenja može se preskočiti funkcijom

```
KinCnsEqs [ntd_Integer, tdi_List, els_List, nds_List, uu_List] :=
  KinCnsEqs [KinCnsMtx [ntd, tdi, els, nds], uu]
```

čiji su parametri, osim posljednjega, isti kao u funkciji `KinCnsMtx[]`.)

Drugi je parametar funkcije `SolveKinCnsEqs[]` lista odabranih vodećih pomaka. Na temelju te liste funkcija

```
Slaves [ms_List, uu_List] := Delete [uu, Partition [ms, 1]]
```

stvora listu pratećih pomaka koja je lista nepoznanica u pozivu funkcije `Solve[]`.

Na stranici 32, uvodeći u izrazu (33) reducirani oblik matrice sustava jednadžbi kinematičkih ograničenja (32) sa stranice 30 za nosač prikazan na slici 12., istaknuli smo da pretpostavljamo da je $c_{3,4} = \cos \alpha_{3,4} \neq 0$. Ta pretpostavka znači — geometrijski — da element 3–4 nije vertikalalan. Ako je taj element vertikalalan, kao u sistemu sa slike 2. na stranici 8, njegov se gornji čvor ne može ni spustiti ni podignuti u odnosu na donji, te vrijedi da je $v_3 = 0$, iz čega je vidljivo da v_3 ne može biti vodeći pomak. Sustav jednadžbi kinematičkih ograničenja sada je

$$\begin{bmatrix} c_{1,2} & s_{1,2} & 0 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (37)$$

Kako je, naime, element 3–4 vertikalalan, $s_{3,4} = \sin \alpha_{3,4} = -1$. Kako je u našem primjeru još i element 2–3 horizontalalan, $c_{2,3} = 1$ i $s_{2,3} = 0$. (To što je u zadanoj konstrukciji 2–3 horizontalalan ne utječe na strukturu matrice ograničenja i reducirane matrice, nego samo na vrijednosti njihovih komponentata.)

Reducirani je sustav:

$$\begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & h_{2,3} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_2 \\ v_2 \\ u_3 \\ v_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (38)$$

Budući da treći stupac matrice nije bazni (sadrži koeficijente -1 i $h_{2,3}$), 'automatski' odabrani vodeći pomak ovoga nosača je u_3 :

$$\begin{aligned} u_2 &= u_3, \\ v_2 &= -h_{2,3} \cdot u_3, \\ v_3 &= 0. \end{aligned}$$

(Jasno je da i u ovom primjeru možemo sami za vodeći pomak odabrati u_2 .)

4.3. Kinematička kondenzacija

U primjeru prikazanom na slici 12. zaključili smo da duljina vodećega pomaka v_3 određuje duljine pratećih pomaka u_2 , v_2 i u_3 ili da duljina odabranoga vodećeg pomaka u_2 određuje duljine pratećih pomaka v_2 , u_3 i v_3 . Isto tako, u primjeru sa slike 2. duljina vodećega pomaka u_3 određuje prateće pomake u_2 i v_2 . Uz poznate duljine vodećih pomaka možemo pomoću izraza (35) i (36) ili analognih izraza odrediti i prateće pomake, tako da se u sustavu jednadžbi ravnoteže čvorova kao nepoznanice smiju pojaviti samo duljine vodećih (translacijskih) pomaka i kutovi zaokreta čvorova [2, 7]. Zaokreti čvorova su neovisni, te se svrstavaju u vodeće pomake.

Dakle, *kinematičku kondenzaciju* možemo definirati kao postupak ,uklanjanja' vrijednosti komponenata pomaka iz jednadžbi ravnoteže primjenom kinematičkih ograničenja [2, 7]. Time ujedno smanjujemo broj jednadžbi u sustavu.

Za primjer sa slike 12. vektor je svih nepoznatih poopćenih pomaka

$$\mathbf{u} = [u_2 \ v_2 \ \varphi_2 \ u_3 \ v_3 \ \varphi_3]^T, \quad (39)$$

dok je vektor vodećih pomaka

$$\mathbf{u}^v = [\varphi_2 \ v_3 \ \varphi_3]^T. \quad (40)$$

Vežu između vektorâ (39) i (40) možemo prikazati matrično:

$$\begin{bmatrix} u_2 \\ v_2 \\ \varphi_2 \\ u_3 \\ v_3 \\ \varphi_3 \end{bmatrix} = \begin{bmatrix} 0 & -h_{1,4} & 0 \\ 0 & -h_{2,4} & 0 \\ 1 & 0 & 0 \\ 0 & -h_{3,4} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \varphi_2 \\ v_3 \\ \varphi_3 \end{bmatrix}, \quad (41)$$

ili, sažeto:

$$\mathbf{u} = \mathbf{C} \mathbf{u}^v. \quad (42)$$

Matricu \mathbf{C} dobivamo pomoću funkcije

```
KinCondensMtx2 [nd_Integer, ntd_Integer,
                di_List, tdi_List, ms_List, h_List] :=
Module[
  { mc, d1, d2, d3, td1, td2, nm, nmr, m, lm, lt, lmc, i, j },
  nm = Length [ms];
  nmr = NumRotDsOF [di];
  mc = Table [0.0, {nd}, {nm + nmr}];
  lm = Sort [ms];
```



```

lt = {};
lmc = {};
i = 1;
nmr = 0;
Do[
  { d1, d2, d3 } = di[[j]];
  { td1, td2 } = tdi[[j]];
  If [i <= nm,
    m = lm[[i]]
  ];
  If [d1 != 0,
    lt = { lt, d1 };
    If [td1 == m,
      lmc = { lmc, ++nmr };
      If [++i <= nm,
        m = lm[[i]]
      ]
    ]
  ];
  If [d2 != 0,
    lt = { lt, d2 };
    If [td2 == m,
      lmc = { lmc, ++nmr };
      If [++i <= nm,
        m = lm[[i]]
      ]
    ]
  ];
  If [d3 != 0,
    mc[[d3, ++nmr]] = 1.0
  ],
  { j, 1, Length [di] }
];
lt = Flatten [lt];
lmc = Flatten [lmc];
Do[
  Do[
    mc[[ lt[[j]], lmc[[i]] ]] = h[[j, i]],
    { i, 1, Length [lmc] }
  ],

```

```

        { j, 1, Length [lt] }
    ];
    mc
]

```

Parametri funkcije su ukupan broj stupnjeva slobode `nd`, broj translacijskih stupnjeva slobode `ntd`, tablica indeksa svih stupnjeva slobode `di`, tablica indeksa translacijskih stupnjeva slobode `tdi`, lista vodećih pomaka `ms` i matrica `h` dobivena funkcijama `SolveFromRCnsMatrix[]` ili `SolveKinCnsEqs[]`.

Funkcija `KinCondensMtx2[]` „proširuje“ matricu \mathbf{H} tako da i zaokreti čvorova budu uvaženi. Broj redaka matrice \mathbf{C} jednak je broju svih nepoznatih pomaka `nd`, a broj je njezinih stupaca jednak zbroju broja vodećih translacijskih pomaka i broja zaokreta čvorova: `nm + nmr`. Iako funkcija to ne radi, vezu matrice \mathbf{C} i matrice \mathbf{H} jasnije ćemo vidjeti pomaknemo li u vektoru nepoznatih pomaka \mathbf{u}_s na vrh vodeće pomake (uključujući zaokrete) i promijenimo li na odgovarajući način redoslijed redaka matrice \mathbf{C} ; izraz (41) time prelazi u

$$\begin{bmatrix} \varphi_2 \\ v_3 \\ \varphi_3 \\ u_2 \\ v_2 \\ u_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -h_{1,4} & 0 \\ 0 & -h_{2,4} & 0 \\ 0 & -h_{3,4} & 0 \end{bmatrix} \begin{bmatrix} \varphi_2 \\ v_3 \\ \varphi_3 \end{bmatrix}, \quad (43)$$

odnosno,

$$\begin{bmatrix} \mathbf{u}^v \\ \mathbf{u}^p \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{3 \times 3} \\ \mathbf{C}^{(v \rightarrow p)} \end{bmatrix} \mathbf{u}^v, \quad (44)$$

gdje podmatrica $\mathbf{C}^{(v \rightarrow p)}$ sadrži one stupce reducirane matrice ograničenja koji nisu bazni i nul-stupce. Funkcija `KinCondensMtx2[]` razmjerno je složena zato što se poredak komponenata vektora svih nepoznatih pomaka ne smije promijeniti, jer odgovora redoslijedu stupaca matrice krutosti konstrukcije (drugim riječima, kad bismo promijenili redoslijed nepoznanica, morali bismo permutirati i stupce matrice sustava jednadžbi ravnoteže).

Sve u prethodnom i ovom odjeljku opisane korake potrebne za oblikovanje matrice \mathbf{C} obavljaju dvije inačice funkcije `KinCondensMtx[]`:

```

KinCondensMtx [nd_Integer, di_List, els_List, nds_List,
               tol_Real : 10.^-13] :=
Module[
  { ntd, tdi, cc, lms, hh },
  { ntd, tdi } = TranslDsOF [di];
  cc = KinCnsMtx [ntd, tdi, els, nds];

```

```

    cc = ReducedKinCnsMtx [cc];
    lms = Masters [cc, tol];
    hh = SolveFromRCnsMtx [cc, lms];
    KinCondensMtx2 [nd, ntd, di, tdi, lms, hh]
]
KinCondensMtx [nd_Integer, di_List, ms_List, els_List, nds_List,
               tol_Real : 10.^-13] :=
Module[
  { ntd, tdi, eqs, hh, unknown, uu },
  { ntd, tdi } = TranslDsOF [di];
  uu = Unknowns [ntd, unknown];
  eqs = KinCnsEqs [ntd, tdi, els, nds, uu];
  hh = SolveKinCnsEqs [eqs, ms, uu, tol];
  KinCondensMtx2 [nd, ntd, di, tdi, ms, hh]
]

```

Prva inačica, čiji su parametri broj stupnjeva slobode, tablica njihovih indeksa te liste elemenata i čvorova, „automatski“ određuje vodeće pomake reduciraajući matricu kinematičkih ograničenja. Dodatni je parametar druge inačice lista vodećih pomaka \mathbf{ms} ; tu ćemo inačicu, prema tome, upotrijebiti želimo li vodeće pomake sami zadati.

Nepoznate pomake čvorova izračunavamo rješavanjem sustava jednadžbi ravnoteže čvorova (11) na stranici 17. Uvrštavanjem izraza (42) u sustav (11) dobivamo:

$$\mathbf{K C u}^v = \mathbf{q}. \quad (45)$$

Ako vektor \mathbf{u}^v sadrži m^v vodećih komponenata pomaka (uključujući zaokrete čvorova), matrica \mathbf{C} je tipa $m \times m^v$, pa će tipa $m \times m^v$ biti i matrica $\mathbf{K C}$. Kako je $m^v < m$, sustav (45) sadrži više jednadžbi nego što je potrebno za nalaženje vrijednosti nepoznanica. (Za okvire sa slika 2. i 12. su $m = 6$ i $m^v = 3$, pa su \mathbf{K} tipa 6×6 , \mathbf{C} tipa 6×3 i $\mathbf{K C}$ tipa 6×3 ; sustav, prema tome, sadrži 6 jednadžbi sa 3 nepoznanice.) Sustav koji ima više jednadžbi negoli nepoznanica u općem je slučaju nerješiv.

Broj jednadžbi sustava možemo smanjiti tako da izraz (45) pomnožimo slijeva matricom \mathbf{C}^T , te tako dobivamo:

$$\mathbf{C}^T \mathbf{K C u}^v = \mathbf{C}^T \mathbf{q}. \quad (46)$$

Matrica \mathbf{C}^T tipa je $m^v \times m$, pa će matrica $\mathbf{C}^T \mathbf{K C}$ biti tipa $m^v \times m^v$, a vektor $\mathbf{C}^T \mathbf{q}$ imat će m^v komponenata, tako da će sustav (46) sadržavati m^v jednadžbi sa m^v nepoznanica. (U našim je primjerima matrica \mathbf{C}^T tipa 3×6 , pa je matrica $\mathbf{C}^T \mathbf{K C}$ tipa 3×3 , odnosno, sustav ima 3 jednadžbi.)

Kondenziranu matricu krutosti konstrukcije $\mathbf{C}^T \mathbf{K C}$ izračunava funkcija

```
CondensStructK [sk_, cc_] := Transpose [cc] . sk . cc
```

a kondenzirani vektor sila $\mathbf{C}^T \mathbf{q}$ funkcija

```
CondenStructF [sf_, cc_] := Transpose [cc] . sf
```

Prvom inačicom funkcije `KinCondenMtx[]`, koja 'pronalaži' vodeće pomake, za konstrukciju sa slike 2. dobivamo matricu

$$\mathbf{C} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -0,75 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix},$$

pa je kondenzirana matrica krutosti konstrukcije

$$\mathbf{C}^T \mathbf{K} \mathbf{C} = \begin{bmatrix} 250\,000 & 18\,750 & 62\,500 \\ 18\,750 & 61\,171,9 & 30\,468,7 \\ 62\,500 & 30\,468,7 & 281\,250 \end{bmatrix}.$$

Njezine su komponente jednake koeficijentima uz nepoznanice u sustavu jednadžbi inženjerske metode pomake (21) na stranici 26. Kondenzirani je vektor sila

$$\mathbf{C}^T \mathbf{q} = [0 \ 100 \ 0]^T.$$

Rješenje je kondenziranoga sustava

$$\mathbf{u}^v = [-0,000\,088\,6 \ 0,001\,746\,32 \ -0,000\,169\,496]^T.$$

Dobivene se vrijednosti osim po broju značajnih znamenaka ne razlikuju od vrijednosti dobivenih na stranici 26 inženjerskom metodom pomaka.

Za izračunavanje sila na krajevima elemenata funkcijom trebaju nam vrijednosti svih pomaka; možemo ih izračunati prema izrazu (42). Nastavak je isti kao u općoj metodi pomaka: funkcijom `ListEDisplsLoc[]` izračunavamo komponente pomaka u lokalnim koordinatnim sustavima pojedinih elemenata, a potom funkcijom `ListEndForces[]` i ukupne sile na krajevima elemenata. Vektori sila na krajevima za naš su primjer:

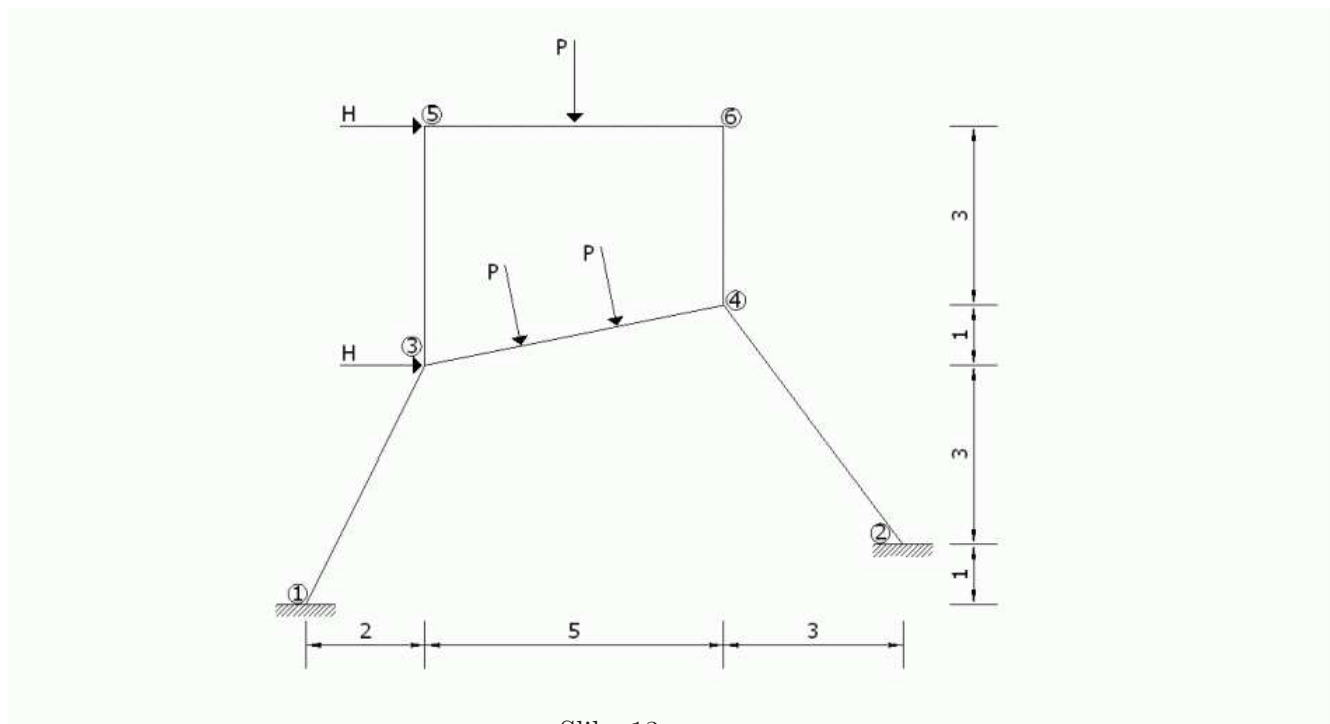
$$\begin{bmatrix} N_{1,2} \\ T_{1,2} \\ M_{1,2} \\ N_{2,1} \\ T_{2,1} \\ M_{2,1} \end{bmatrix} = \begin{bmatrix} 0 \\ 29,421\,0 \\ 76,321\,2 \\ 0 \\ -29,421\,1 \\ 70,783\,7 \end{bmatrix}, \quad \begin{bmatrix} N_{2,3} \\ T_{2,3} \\ M_{2,3} \\ N_{3,2} \\ T_{3,2} \\ M_{3,2} \end{bmatrix} = \begin{bmatrix} 0 \\ -29,324\,7 \\ -70,783\,7 \\ 0 \\ 29,324\,7 \\ -75,839\,7 \end{bmatrix}, \quad \begin{bmatrix} N_{3,4} \\ T_{3,4} \\ M_{3,4} \\ N_{4,3} \\ T_{4,3} \\ M_{4,3} \end{bmatrix} = \begin{bmatrix} 0 \\ 41,230\,3 \\ 75,839\,7 \\ 0 \\ -41,230\,3 \\ 89,081\,5 \end{bmatrix}.$$

Vrijednosti momenata se, naravno, ne razlikuju od vrijednosti koje smo izračunali inženjerskom metodom pomakâ. Dobivene uzdužne sile jednake su nuli, što je u skladu s osnovnom pretpostavkom inženjerske metode pomakâ i s uvedenim kinematičkim ograničenjima: iz $d_i^{\parallel} = d_j^{\parallel}$ slijedi $u_{i,j} = u_{j,i}$, pa su

$$n_{i,j} = \frac{E_e A_e}{\ell_e} u_{i,j} - \frac{E_e A_e}{\ell_e} u_{j,i} = 0,$$
$$n_{j,i} = -\frac{E_e A_e}{\ell_e} u_{i,j} + \frac{E_e A_e}{\ell_e} u_{j,i} = 0.$$

5. Primjer

Primjenu programa ćemo pokazati na sljedećem, nešto složenijem sistemu:



Slika 13.

Na promatrani sistem djeluju koncentrirane sile H u čvorovima 3 i 5, $H = 100$ kN, te sile $P = 50$ kN na elementima 3-4 (u trećinskim točkama) i 5-6 (u polovini raspona). Modul elastičnosti sistema iznosi $E = 3 \cdot 10^7$ kN/m², a dimenzije poprečnih presjeka su $b/h = 50/50$ [cm].

```
nds = {{0.,0.}, {10.,1.}, {2.,4.}, {7.,5.}, {2.,8.}, {7.,8.}}
```

Opis odabranog sistema započinjemo unosom koordinata čvorova u obliku liste dvočlanih listi. Pritom ishodište koordinatnog sustava postavljamo u čvor 1.

```
els = {{{1,3},1}, {{3,4},1}, {{4,2},1}, {{3,5},1}, {{5,6},1}, {{6,4},1}}
```

Elemente sistema upisujemo u listu navođenjem indeksâ početnog i krajnjeg čvora (u listi čvorova) i indeksa karakteristika elementa (u listi karakteristika).

```
chrs = {{3 10^7, 0.5^2, 0.5^4/12}}
```

Lista `chrs` sadrži karakteristike elemenata: modul elastičnosti ($E = 3 \cdot 10^7$ kN/m²), površinu poprečnog presjeka ($A = 0,5^2$ m²) i moment tromosti ($I = 0,5^4/12$ m⁴).

```
cnsts = {{1, {0, 0, 0}}, {2, {0, 0, 0}}}
```

Listom `cnsts` definiramo ležajevе sistema, tj. indekse čvorova u kojima su ležajevi, te stupnjeve slobode tih čvorova. U našem primjeru ležajevi se nalaze u čvorovima 1 i 2, a radi se o upetim ležajevima — svi su pomaci čvora spriječeni: $\{0, 0, 0\}$.

```
{nd, di} = TblNodeDsOF [Length [nds], cnsts]
```

Funkcija `TblNodeDsOF[]` vraća broj stupnjeva slobode (`nd`) i tablicu indeksa stupnjeva slobode (`di`):

```
{12, {{0,0,0},{0,0,0},{1,2,3},{4,5,6},{7,8,9},{10,11,12}}}
```

Argumenti funkcije su broj čvorova, dobiven kao broj elemenata liste `nds` (`Length [nds]`) i lista ležajeva `cnsts`.

```
lkl = ListKLoc [els, nds, chrs]
```

Parametri funkcije `ListKLoc[]` su lista elemenata `els`, lista čvorova `nds` i lista karakteristika elemenata `chrs`. Lista `lkl`, kojoj je pridružen rezultat poziva funkcije, sadrži matrice krutosti elemenata izražene u lokalnim koordinatnim sistemima u redosljedu u kojem su odgovarajući elementi navedeni u listi `els`:

$$\begin{bmatrix} 1,67705 \cdot 10^6 & 0 & 0 & -1,67705 \cdot 10^6 & 0 & 0 \\ 0 & 20963,1 & 46875 & 0 & -20963,1 & 46875 \\ 0 & 46875 & 139754 & 0 & -46875 & 69877,1 \\ -1,67705 \cdot 10^6 & 0 & 0 & 1,67705 \cdot 10^6 & 0 & 0 \\ 0 & 20963,1 & -46875 & 0 & 20963,1 & -46875 \\ 0 & 46875 & 69877,1 & 0 & -46875 & 139754 \end{bmatrix},$$

$$\begin{bmatrix} 1,47087 \cdot 10^6 & 0 & 0 & -1,47087 \cdot 10^6 & 0 & 0 \\ 0 & 14143 & 36057,7 & 0 & -14143 & 36057,7 \\ 0 & 36057,7 & 122573 & 0 & -36057,7 & 61286,3 \\ -1,47087 \cdot 10^6 & 0 & 0 & 1,47087 \cdot 10^6 & 0 & 0 \\ 0 & -14143 & -36057,7 & 0 & 14143 & -36057,7 \\ 0 & 36057,7 & 61286,3 & 0 & -36057,7 & 122573 \end{bmatrix},$$

$$\begin{bmatrix} 1,5 \cdot 10^6 & 0 & 0 & -1,5 \cdot 10^6 & 0 & 0 \\ 0 & 15000 & 37500 & 0 & -15000 & 37500 \\ 0 & 37500 & 125000 & 0 & -37500 & 62500 \\ -1,5 \cdot 10^6 & 0 & 0 & 1,5 \cdot 10^6 & 0 & 0 \\ 0 & -15000 & -37500 & 0 & 15000 & -37500 \\ 0 & 37500 & 62500 & 0 & -37500 & 125000 \end{bmatrix},$$

$$\begin{bmatrix} 1,875 \cdot 10^6 & 0 & 0 & -1,875 \cdot 10^6 & 0 & 0 \\ 0 & 29\,296,9 & 58\,593,8 & 0 & -29\,296,9 & 58\,593,8 \\ 0 & 58\,593,8 & 156\,250 & 0 & -58\,593,8 & 78\,125 \\ -1,875 \cdot 10^6 & 0 & 0 & 1,875 \cdot 10^6 & 0 & 0 \\ 0 & -29\,296,9 & -58\,593,8 & 0 & 29\,296,9 & -58\,593,8 \\ 0 & 58\,593,8 & 78\,125 & 0 & -58\,593,8 & 15\,6250 \end{bmatrix},$$

$$\begin{bmatrix} 1,5 \cdot 10^6 & 0 & 0 & -1,5 \cdot 10^6 & 0 & 0 \\ 0 & 15\,000 & 37\,500 & 0 & -15\,000 & 37\,500 \\ 0 & 37\,500 & 125\,000 & 0 & -37\,500 & 62\,500 \\ -1,5 \cdot 10^6 & 0 & 0 & 1,5 \cdot 10^6 & 0 & 0 \\ 0 & -15\,000 & -37\,500 & 0 & 15\,000 & -37\,500 \\ 0 & 37\,500 & 62\,500 & 0 & -37\,500 & 125\,000 \end{bmatrix},$$

$$\begin{bmatrix} 2,5 \cdot 10^6 & 0 & 0 & -2,5 \cdot 10^6 & 0 & 0 \\ 0 & 69\,444,4 & 104\,167 & 0 & -69\,444,4 & 104\,167 \\ 0 & 104\,167 & 208\,333 & 0 & -104\,167 & 104\,167 \\ -2,5 \cdot 10^6 & 0 & 0 & 2,5 \cdot 10^6 & 0 & 0 \\ 0 & -69\,444,4 & -104\,167 & 0 & 69\,444,4 & -104\,167 \\ 0 & 104\,167 & 104\,167 & 0 & -104\,167 & 208\,333 \end{bmatrix}.$$

`lr = ListTG2L [els, nds]`

Komponente liste `lr` matrice su transformacija iz globalnoga u lokalne koordinatne sustave pojedinih elemenata:

$$\begin{bmatrix} 0,447\,214 & 0,894\,427 & 0 & 0 & 0 & 0 \\ -0,894\,427 & 0,447\,214 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,447\,214 & 0,894\,427 & 0 \\ 0 & 0 & 0 & -0,894\,427 & 0,447\,214 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 0,980\,581 & 0,196\,116 & 0 & 0 & 0 & 0 \\ -0,196\,116 & 0,980\,581 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,980\,581 & 0,196\,116 & 0 \\ 0 & 0 & 0 & -0,196\,116 & 0,980\,581 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 0,6 & -0,8 & 0 & 0 & 0 & 0 \\ 0,8 & 0,6 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,6 & -0,8 & 0 \\ 0 & 0 & 0 & 0,8 & 0,6 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

`K = StructK [nd, di, els, lkl, lr]`

Funkcija `StructK[]` izračunava matricu krutosti sistema. Njezini su parametri broj stupnjeva slobode `nd`, tablica indeksa stupnjeva slobode `di`, lista elemenata `els`, lista lokalnih matrica krutosti `lkl` i lista matrica transformacija `lr`.

Kako je u našem primjeru broj stupnjeva slobode 12, matrica krutosti sistema bit će tipa 12×12 . Zbog njezine smo je veličine izdvojili na sliku 14. na sljedećoj stranici i ‚razlomili‘.

$1,79632 \cdot 10^6$	942575	-23739	$-1,41484 \cdot 10^6$	-280140	-7071,5
942575	$3,291 \cdot 10^6$	14394,3	-280140	-7071,5	35357,5
-23739	14394,3	418577	7071,5	-35357,5	61286,3
$-1,41484 \cdot 10^6$	-280140	7071,5	$2,03389 \cdot 10^6$	-432660	-67095,2
-280140	-7071,5	-35357,5	-432660	$3,53557 \cdot 10^6$	-12857,5
-7071,5	35357,5	61286,3	-67095,2	-12857,5	455906
-29296,9	0	58593,8	0	0	0
0	$-1,875 \cdot 10^6$	0	0	0	0
-58593,8	0	78125	0	0	0
0	0	0	-69444,4	0	104167
0	0	0	0	$-2,5 \cdot 10^6$	0
0	0	0	-104167	0	104167

-29296,9	0	-58593,8	0	0	0
0	$-1,875 \cdot 10^6$	0	0	0	0
58593,8	0	78125	0	0	0
0	0	0	-69444,4	0	-104167
0	0	0	0	$-2,5 \cdot 10^6$	0
0	0	0	104167	0	104167
$1,5293 \cdot 10^6$	0	58593,8	$-1,5 \cdot 10^6$	0	0
0	$1,89 \cdot 10^6$	37500	0	-15000	37500
58593,8	37500	281250	0	-37500	62500
$-1,5 \cdot 10^6$	0	0	$1,56944 \cdot 10^6$	0	104167
0	-15000	-37500	0	$2,515 \cdot 10^6$	-37500
0	37500	62500	104167	-37500	333333

Slika 14.

l1d = {{5, ConcFL, {0,-50,0,0.5}}, {2, ConcFL, {0,-50,0,0.333}},
{2, ConcFL, {0,-50,0,0.667}}}

Djelovanja na elemente sistema zadajemo u listi l1d. Komponente te liste su liste čiji je prvi član oznaka opterećenoga elementa (u našem primjeru 2 i 5), drugi je član naziv funkcije koja izračunava sile i momente upetosti, a treći je član lista vrijednosti komponenata sila u lokalnom koordinatnom sustavu elementa ($F_y = -50$ kN) i relativni položaj hvatišta (sila P koja djeluje na element 5, nalazi se u polovištu elementa, te njezin relativni položaj iskazujemo omjerom $a/l = 0,5$).

`lul = ListEDisplsLoc [u, di, els, lr]`

Da bismo izračunali sile na krajevima elemenata i u potpunosti riješili zadani sistem, potrebno je dobivene pomake prebaciti iz globalnog koordinatnog sustava u lokalni, te ih ‚prikazati‘ po elementima:

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ 0,000\,021\,058\,8 \\ -0,003\,114\,1 \\ -0,000\,380\,081 \end{bmatrix} \quad \begin{bmatrix} 0,002\,471\,05 \\ -0,001\,895\,25 \\ -0,000\,380\,081 \\ 0,002\,432\,31 \\ 0,000\,998\,074 \\ -0,000\,142\,242 \end{bmatrix} \quad \begin{bmatrix} 0,000\,149\,036 \\ 0,002\,624\,89 \\ -0,000\,142\,242 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
 \begin{bmatrix} -0,001\,373\,83 \\ -0,002\,794\,75 \\ -0,000\,380\,081 \\ -0,001\,367\,45 \\ -0,003\,739\,65 \\ 0,000\,181\,39 \end{bmatrix} \quad \begin{bmatrix} 0,003\,739\,65 \\ -0,001\,367\,45 \\ 0,000\,181\,39 \\ 0,003\,683\,68 \\ 0,001\,430\,92 \\ -0,000\,047\,976\,5 \end{bmatrix} \quad \begin{bmatrix} -0,001\,430\,92 \\ 0,003\,683\,68 \\ -0,000\,047\,976\,5 \\ -0,001\,455\,71 \\ 0,002\,189\,33 \\ -0,000\,142\,242 \end{bmatrix}$$

Dakle

$$[0 \ 0 \ 0 \ 0,000\,021\,058\,8 \ -0,003\,114\,1 \ -0,000\,380\,081]^T$$

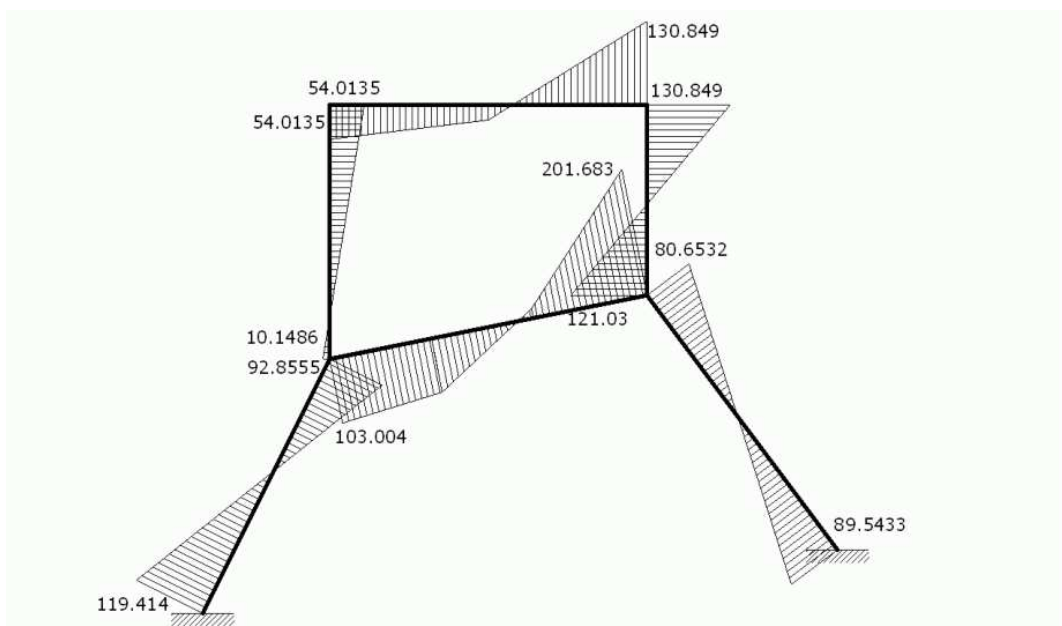
znači da su na prvom elementu, u čvoru 1, translacijski pomaci i kut zaokreta jednaki nuli, a u čvoru 3 postoje $u_3 = 0,000\,021\,058\,8$, $v_3 = -0,003\,114\,1$ te $\varphi_3 = -0,000\,380\,081$. Potrebni parametri su pomaci u globalnom koordinatnom sustavu `u`, tablica indeksa stupnjeva slobode `di`, lista elemenata `els` i lista matrica transformacija `lr`.

`lefs = ListEndForces [lkl, lul, lef]`

Funkcija `ListEndForces[]` izračunava sile i momente na krajevima elemenata za ranije dobivene parametre `lkl` (lista matricâ krutosti u lokalnom koordinatnom sustavu), `lul` (lista vektorâ pomaka krajeva elemenata) i `lef` (lista sila upetosti na krajevima elemenata):

$$\begin{bmatrix} -35,316\,6 & 47,465 & 119,414 & 35,316\,6 & -47,465 & 92,855\,5 \end{bmatrix}^T \\
 \begin{bmatrix} 56,986\,2 & -9,754\,01 & -103,004 & -56,986\,2 & 109,754 & -201,683 \end{bmatrix}^T \\
 \begin{bmatrix} 223,554 & 34,039\,3 & 80,653\,2 & -223,554 & -34,039\,3 & 89,543\,3 \end{bmatrix}^T \\
 \begin{bmatrix} -11,972\,5 & 16,040\,5 & 10,148\,6 & 11,972\,5 & -16,040\,5 & 54,013\,5 \end{bmatrix}^T \\
 \begin{bmatrix} 83,959\,5 & -11,972\,5 & -54,013\,5 & -83,959\,5 & 61,972\,5 & -130,849 \end{bmatrix}^T \\
 \begin{bmatrix} 61,972\,5 & 83,959\,5 & 130,849 & -61,972\,5 & -83,959\,5 & 121,03 \end{bmatrix}^T$$

Pripadni momentni dijagram možemo vidjeti na slici 15.



Slika 15.

Sistem je riješen općom metodom pomaka, dok ćemo u nastavku rješenje dobiti uz kinematička ograničenja i to tako da dopustimo da program sam 'izabere' vodeće pomake sistema.

```
{ntd, tdi} = TranslDsOF [di]
```

Funkcija `TranslDsOF []` za parametar `di` (tablica indeksa stupnjeva slobode) vraća broj translacijskih stupnjeva slobode (`ntd`) i tablicu indeksa translacijskih stupnjeva slobode `tdi`:

```
{8, {{0,0}, {0,0}, {1,2}, {3,4}, {5,6}, {7,8}}}
```

```
c = KinCnsMtx [ntd, tdi, els, nds]
```

Matrica `c`

$$\begin{bmatrix}
 -0,447214 & -0,894427 & 0 & 0 & 0 & 0 & 0 & 0 \\
 0,980581 & 0,196116 & -0,980581 & -0,196116 & 0 & 0 & 0 & 0 \\
 0 & 0 & 0,6 & -0,8 & 0 & 0 & 0 & 0 \\
 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 \\
 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\
 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1
 \end{bmatrix}$$

koju smo dobili pomoću funkcije `KinCnsMtx[]` je matrica sustava jednadžbi kinematičkih ograničenja

$$\begin{bmatrix} -c_{1,2} & -s_{1,3} & 0 & 0 & 0 & 0 & 0 & 0 \\ c_{3,4} & s_{3,4} & -c_{3,4} & -s_{3,4} & 0 & 0 & 0 & 0 \\ 0 & 0 & c_{2,4} & s_{2,4} & 0 & 0 & 0 & 0 \\ c_{3,5} & s_{3,5} & 0 & 0 & -c_{3,5} & -s_{3,5} & 0 & 0 \\ 0 & 0 & 0 & 0 & c_{5,6} & s_{5,6} & -c_{5,6} & -s_{5,6} \\ 0 & 0 & c_{4,6} & s_{4,6} & 0 & 0 & c_{4,6} & -s_{4,6} \end{bmatrix}.$$

Možemo vidjeti da se radi o sustavu šest jednadžbi s osam nepoznanica.

`rc = ReducedKinCnsMtx [c]`

Matrica `rc` predstavlja reduciranu matricu sustava, tj. Hermiteov oblik matrice:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & -1,7037 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0,851852 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1,33333 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1, \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0,851852 \end{bmatrix}.$$

`lm = Masters [rc]`

Pomoću funkcije `Masters[]` program sam određuje vodeće pomake `lm` iz reducirane matrice sustava `rc`. Funkcija nam vraća indekse vodećih pomaka, te se u našem primjeru radi o horizontalnom i vertikalnom pomaku čvora 6 ($\{u_6, v_6\} = \{7, 8\}$). Iz reducirane matrice, naime, prema izrazu (35) slijedi:

$$\begin{aligned} u_3 &= 1,7037 \cdot v_6, \\ v_3 &= -0,851852 \cdot v_6, \\ u_4 &= 1,33333 \cdot v_6, \\ v_4 &= v_6, \\ u_5 &= u_6, \\ v_5 &= -0,851852 \cdot v_6. \end{aligned}$$

`hh = SolveFromRCnsMtx [rc, lm]`

`hh` je matrica \mathbf{H} iz izraza (35). Ona sadrži stupce matrice `rc` koji nisu bazni s jedinicama dodanim na mjestima koja odgovaraju vodećim pomacima. Iz matrice `rc` možemo očitati da se radi o sedmom i osmom stupcu:

$$\begin{bmatrix} 0 & 1,7037 \\ 0 & -0,851852 \\ 0 & 1,33333 \\ 0 & 1 \\ 1 & 0 \\ 0 & -0,851852 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

$cc = \text{KinCondensMtx2} [\text{nd}, \text{ntd}, \text{di}, \text{tdi}, \text{lm}, \text{hh}]$

Matricu cc , odnosno C iz izraza (42), izračunava funkcija $\text{KinCondensMtx2}[]$ koja za prethodno određene parametre 'proširuje' matricu hh . Time su u matrici cc uključeni i kutovi zaokreta svih čvorova sistema:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1,7037 & 0 \\ 0 & 0 & 0 & 0 & -0,851852 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1,33333 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0,851852 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$CK = \text{Transpose}[cc] \cdot K \cdot cc$

Matrica CK je kondenzirana matrica krutosti sistema, iz izraza (46). CK je tipa 6, tj. kondenzirani sustav ima 6 jednadžbi:

$$\begin{bmatrix} 418577 & 61286,3 & 78125 & 58593,8 & -78634,8 & 0 \\ 61286,3 & 455906 & 0 & 104167 & -144485 & 104167 \\ 78125 & 0 & 281250 & 58593,8 & -169271 & 62500. \\ 58593,8 & 104167 & 58593,8 & 98741,3 & -142506 & 104167 \\ -78634,8 & -144485 & -169271 & -142506 & 428102 & -208333 \\ 0 & 104167 & 62500 & 104167 & -208333 & 333333 \end{bmatrix}.$$

$$Cq = \text{Transpose}[cc] \cdot q$$

Cq je kondenzirani vektor sila, odnosno desna strana izraza (46):

$$\left[-56,6274 \quad 56,6274 \quad -31,25 \quad 100 \quad 189,184 \quad 31,25 \right]^T.$$

$$Cu = \text{LinearSolve} [CK, Cq]$$

Rješenje kondenziranoga sustava dobivamo funkcijom `LinearSolve[]`. Vrijednosti označavaju redom $\varphi_3, \varphi_4, \varphi_5, u_6, v_6, \varphi_6$:

$$\left[-0,000353791 \quad -0,00013209 \quad 0,000206297 \quad 0,00355098 \quad 0,00158487 \quad -0,0000227914 \right]^T.$$

$$uu = cc \cdot Cu$$

Da bismo mogli izračunati sile na krajevima elemenata, potrebne su nam vrijednosti svih pomaka uu . Njih dobivamo množenjem vektora Cu matricom cc :

$$\begin{bmatrix} 0,00270015 \\ -0,00135007 \\ -0,000353791 \\ 0,00211316 \\ 0,00158487 \\ -0,00013209 \\ 0,00355098 \\ -0,00135007 \\ 0,000206297 \\ 0,00355098 \\ 0,00158487 \\ -0,0000227914 \end{bmatrix}.$$

$$luul = \text{ListEDisplsLoc} [uu, di, els, lr]$$

Nastavak proračuna isti je kao u općoj metodi bez kinematičke kondenzacije. Pomaci uu izraženi su u globalnom koordinatnom sustavu, te ih funkcijom `ListEDisplsLoc[]` ,prebacujemo' u lokalni koordinatni sustav. (Rezultat nećemo prikazivati.)

$$lefs2 = \text{ListEndForces} [lk1, luul, lef]$$

Funkcija `ListEndForces[]` izračunava sile i momente na krajevima elemenata:

$$\begin{aligned} & \left[-3,63652 \cdot 10^{-13} \quad 46,7008 \quad 116,787 \quad 3,63652 \cdot 10^{-13} \quad -46,7008 \quad 92,0652 \right]^T, \\ & \left[-4,54747 \cdot 10^{-13} \quad -9,85063 \quad -102,756 \quad 4,54747 \cdot 10^{-13} \quad 109,851 \quad -202,424 \right]^T, \\ & \left[0 \quad 34,6684 \quad 82,5431 \quad 0 \quad -34,6684 \quad 90,7987 \right]^T, \\ & \left[0 \quad 16,2846 \quad 10,6908 \quad 0 \quad -16,2846 \quad 54,4477 \right]^T, \\ & \left[0 \quad -12,1427 \quad -54,4477 \quad 0 \quad 62,1427 \quad -131,266 \right]^T, \\ & \left[0 \quad 83,7154 \quad 131,266 \quad 0 \quad -83,7154 \quad 119,88 \right]^T. \end{aligned}$$

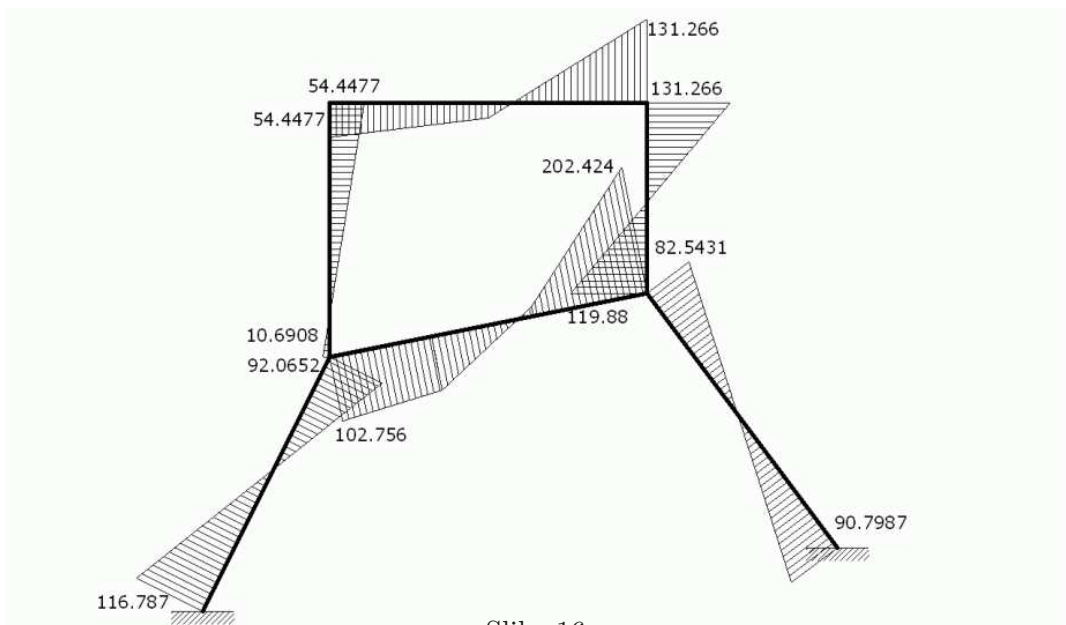
Može se dogoditi da zbog pogrešaka zaokruživanja vrijednosti uzdužnih sila u nekim elementima nisu jednake nuli—u našem primjeru uzdužne sile u prva dva elementa imaju vrlo male vrijednosti, $\sim 4 \cdot 10^{-13}$. Dosljednosti radi, funkcijom `Chop[]` jezika Mathematica možemo sve vrijednosti manje od odabrane tolerancije 'srezati' na nulu:

```
lefs2 = Chop [lefs2, 10.^-12]
```

pa prva dva vektora sila upetosti postaju:

$$\begin{aligned} & \left[0 \quad 46,7008 \quad 116,787 \quad 0 \quad -46,7008 \quad 92,0652 \right]^T, \\ & \left[0 \quad -9,85063 \quad -102,756 \quad 0 \quad 109,851 \quad -202,424 \right]^T. \end{aligned}$$

Dijagram momenata savijanja uz uvažavanje kinematičkih ograničenja prikazan je na slici 16.



Slika 16.

U ‚ručnom‘ proračunu inženjerskom metodom vjerojatno ne bismo kao neovisne translacijske pomake odabrali oba pomaka čvora 6 — može se reći da ‚automatski‘ izbor na temelju Hermiteova normalnog oblika matrice kinematičkih ograničenja nije nimalo intuitivan (iako je, naravno, s formalnoga stanovišta korektan). Prirodnije je zadati takozvane „pomake etaža” — u našem primjeru horizontalne pomake čvorova 4 i 6 (ili čvorova 3 i 5).

```
PrintTranslDsOF [tdi]
```

Ako vodeće pomake želimo izabrati sami, nakon poziva funkcije `TranslDsOF []` možemo funkcijom `PrintTranslDsOF []` pregledno ispisati translacijske pomake:

$$\begin{pmatrix} 1 & \{3, u\} \\ 2 & \{3, v\} \\ 3 & \{4, u\} \\ 4 & \{4, v\} \\ 5 & \{5, u\} \\ 6 & \{5, v\} \\ 7 & \{6, u\} \\ 8 & \{6, v\} \end{pmatrix}.$$

U dobivenoj tablici vidimo da su u_4 (tj. $\{4, u\}$) i u_6 (tj. $\{6, u\}$) pomaci s indeksima 3 i 7.

```
lm = {3, 7}
```

Odabrane ćemo pomake zadati u listi vodećih pomaka `lm`.

```
uu = Unknowns [ntd, x]
```

Funkcijom `Unknowns []` stvaramo listu naziva nepoznanica koja nam je potrebna za funkcije `KinCnsEqs []` i `SolveKinCnsEqs []`. Dobiveni nazivi su `x[1]` do `x[8]`.

```
eqs = KinCnsEqs [ntd, tdi, els, nds, uu]
```

`eqs` je sustav jednadžbi kinematičkih ograničenja (umjesto `x[i]` pisat ćemo, preglednosti radi, x_i):

$$\begin{aligned} -0,447\,214 \cdot x_1 - 0,894\,427 \cdot x_2 &= 0, \\ 0,980\,581 \cdot x_1 + 0,196\,116 \cdot x_2 - 0,980\,581 \cdot x_3 - 0,196\,116 \cdot x_4 &= 0, \\ 0,6 \cdot x_3 - 0,8 \cdot x_4 &= 0, \\ x_2 - x_6 &= 0, \\ x_5 - x_7 &= 0, \\ x_4 - x_8 &= 0. \end{aligned}$$

hh = SolveKinCnsEqs [eqs, lm, uu]

Matricu \mathbf{H} iz izraza (35) sada oblikujemo rješavajući sustav jednažbi ograničenja za nepoznanice koje nisu u listi vodećih pomaka lm:

$$\begin{bmatrix} 1,277\,78 & 0 \\ -0,638\,889 & 0 \\ 1 & 0 \\ 0,75 & 0 \\ 0 & 1 \\ -0,638\,889 & 0 \\ 0 & 1 \\ 0,75 & 0 \end{bmatrix}$$

cc = KinCondensMtx2 [nd, ntd, di, tdi, lm, hh]

Nastavak je isti kao ranije, za ,automatski' odabir vodećih pomaka. Prvo oblikujemo matricu \mathbf{C} iz izraza (42):

$$\begin{bmatrix} 0 & 1,277\,78 & 0 & 0 & 0 & 0 \\ 0 & -0,638\,889 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0,75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & -0,638\,889 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0,75 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

a potom i kondenziranu matricu krutosti sistema:

CK = Transpose[cc] . K . cc

$$\begin{bmatrix} 418\,577 & -58\,976,1 & 61\,286,3 & 78\,125 & 58\,593,8 & 0. \\ -58\,976,1 & 240\,807 & -108\,364 & -126\,953 & -106\,879 & -156\,250 \\ 61\,286,3 & -108\,364 & 455\,906 & 0 & 104\,167 & 104\,167 \\ 78\,125 & -126\,953 & 0 & 281\,250 & 58\,593,8 & 62\,500 \\ 58\,593,8 & -106\,879 & 104\,167 & 58\,593,8 & 98\,741,3 & 104\,167 \\ 0. & -156\,250 & 104\,167 & 62\,500 & 104\,167 & 333\,333 \end{bmatrix}$$

te kondenzirani vektor sila u čvorovima:

$$Cq = \text{Transpose}[cc] \cdot q$$

$$\begin{bmatrix} -56,6274 & 141,888 & 56,6274 & -31,25 & 100 & 31,25 \end{bmatrix}^T$$

$$Cu = \text{LinearSolve} [CK, Cq]$$

Rješenje kondenziranog sustava sadrži, redom, φ_3 , u_4 , φ_4 , φ_5 , u_6 i φ_6 :

$$\begin{bmatrix} -0,000353791 & 0,00211316 & -0,00013209 & 0,000206297 & 0,00355098 & -0,0000227914 \end{bmatrix}^T.$$

$$uu = cc \cdot Cu$$

$$luul = \text{ListEDisplsLoc} [uu, di, els, lr]$$

$$\text{lefs2} = \text{Chop} [\text{ListEndForces} [lkl, luul, lef], 10.^{-12}]$$

Dobivene vrijednosti pomaka i sila na krajevima elemenata jednake su vrijednostima dobivenim pri 'automatskom' odabiru vodećih pomaka.

Zanima li nas samo matrica \mathbf{C} , a ne i matrica kinematičkih ograničenja, njezin reducirani oblik, odabrani vodeći pomaci i matrica \mathbf{H} , funkcijom `KinCondensMtx[]` možemo 'zamijeniti' funkcije `KinCnsMtx[]`, `ReducedKinCnsMtx[]`, `Masters[]`, `SolveFromRCnsMtx[]` i `KinCondensMtx2[]`:

$$cc = \text{KinCondensMtx} [nd, di, els, nds]$$

Ako sami odabiremo vodeće pomake, druga inačica funkcije `KinCondensMtx[]`, s listom vodećih pomaka kao dodatnim parametrom, 'zamjenjuje' funkcije `Unknowns[]`, `KinCnsEqs[]`, `SolveKinCnsEqs[]` i `KinCondensMtx2[]`:

$$cc = \text{KinCondensMtx} [nd, di, lm, els, nds]$$

Daljnji je postupak rješavanja sistema kao i ranije:

$$CK = \text{Transpose}[cc] \cdot K \cdot cc$$

$$Cq = \text{Transpose}[cc] \cdot q$$

$$Cu = \text{LinearSolve} [CK, Cq]$$

i tako dalje.

Vrijednosti momenata na krajevima elemenata dobivenih općom metodom pomaka i uz uvažavanje kinematičkih ograničenja usporedili smo u tablici 2. na sljedećoj stranici.

Relativne pogreške ε izračunali smo prema izrazu

$$\varepsilon = \frac{|M_{i,j}^{\text{OMP}} - M_{i,j}^{\text{KO}}|}{|M_{\text{max}}^{\text{OMP}}|} \cdot 100 [\%] = \frac{|M_{i,j}^{\text{OMP}} - M_{i,j}^{\text{KO}}|}{130,849} \cdot 100 [\%].$$

Tablica 2.

	OMP [kNm]	IMP [kNm]	ε [%]
$M_{1,3}$	119,414	116,787	2,008
$M_{3,1}$	92,8555	92,0652	0,604
$M_{3,4}$	-103,004	-102,756	0,190
$M_{4,3}$	-201,683	-202,424	0,566
$M_{4,2}$	80,6532	82,5431	1,444
$M_{2,4}$	89,5433	90,7987	0,959
$M_{3,5}$	10,1486	10,6908	0,414
$M_{5,3}$	54,0135	54,4477	0,332
$M_{5,6}$	-54,0135	-54,4477	0,332
$M_{6,5}$	-130,849	-131,266	0,319
$M_{6,4}$	130,849	131,266	0,319
$M_{4,6}$	121,030	119,880	0,879

6. Zaključak

Za rješavanje građevinskih konstrukcija najčešće upotrebljavamo metodu konačnih elemenata koja nam za većinu sistema daje zadovoljavajuće rezultate. Međutim, kod vrlo velikih konstrukcijskih sistema s porastom broja jednažbi potrebnih za proračun rastu i greške u rezultatima [5].

Kinematička ograničenja su geometrijski uvjeti koje poopćeni pomaci čvorova, kao glavne nepoznanice u metodi konačnih elemenata utemeljenoj na općoj metodi pomakâ, moraju zadovoljiti. Uvažavanjem kinematičkih ograničenja koja opisuju apsolutno krute veze primjenom kinematičke kondenzacije može se osjetno smanjiti broj nepoznanica, a time i veličina sustava jednažbi, što znači da se njegova uvjetovanost poboljšava [4, 5].

U literaturi se, međutim, kao ‚manje‘ kinematičke kondenzacije navode [7, 16]:

- osim sustava jednažbi ravnoteže treba riješiti i ‚pomoćni‘ sustav jednažbi kinematičkih ograničenja oblika (32) na stranici 30 koji ima veći broj nepoznanica od broja jednažbi, pa se ne može riješiti pomoću klasičnih linearnoalgebarskih paketa poput Lapack-a;
- sustav jednažbi ravnoteže treba preurediti u skladu s izrazom (46) na stranici 40 kako bi se uklonile nepoznanice i jednažbe koje odgovaraju pratećim pomacima;
- ako se matrica sustava zapisuje u ‚štednom‘ obliku, primjena izraza (46) traži složene algoritme za osiguravanje potrebnog memorijskog prostora.

U ovom je radu programu za proračun štapnih konstrukcija općom metodom pomaka dodan modul za uvažavanje kinematičkih ograničenja kinematičkom kondenzacijom. Program je izrađen unutar programskog sustava Mathematica koji ima velike mogućnosti simboličkog računanja [15, 11]. Pokazalo se da se primjenom Mathematica-e ili nekog srodnog sustava za simboličku algebru prva od spomenutih ‚mana‘ može uspješno prevladati, neovisno o tome prepušta li se izbor vodećih pomaka programu ili ih zadaje korisnik.

Literatura

- [1] M. Anđelić: *Građevna statika II*, Građevinski fakultet Sveučilišta u Zagrebu, Zagreb, 2005.
- [2] R. D. Cook, D. S. Malkus, M. E. Plesha: *Concepts and Applications of Finite Element Analysis*, John Wiley & Sons, New York, 1989.
- [3] J. W. Demmel: *Applied Numerical Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, 1997.
- [4] J. Dvornik: *Kinematička ograničenja u metodi konačnih elemenata*, Zbornik radova C₂: Mehanika čvrstog deformabilnog tela, 18. jugoslovenski kongres teorijske i primjenjene mehanike, Jugoslovensko društvo za mehaniku, Vrnjačka banja, 1988., str. 61–64
- [5] J. Dvornik, D. Lazarević: *Manjkavosti proračunskih modela inženjerskih konstrukcija*, Građevinar **57** (2005) 4, str. 227–236.
- [6] N. Elezović: *Linearna algebra*, Element, Zagreb, 2003.
- [7] C. A. Felippa: *Introduction to Finite Element Methods*, University of Colorado, Boulder, 2004.; dostupno na <http://www.colorado.edu/engineering/CAS/courses.d/IFEM.d/Home.html>, pristupljeno: 5. svibnja 2008.
- [8] K. Fresl: *Građevna statika 1.: bilješke i skice s predavanja*, <http://www.grad.hr/nastava/gS/bilj1/index.html>, pristupljeno: 5. svibnja 2008.
- [9] K. Fresl: *Građevna statika 2.: bilješke i skice s predavanja*, <http://www.grad.hr/nastava/gS/bilj2/index.html>, pristupljeno: 5. svibnja 2008.
- [10] D. Lazarević, J. Dvornik: *Plošni nosači. Bilješke s predavanja*, rukopis, Građevinski fakultet Sveučilišta u Zagrebu, Zagreb, travanj 2008.
- [11] R. Maeder: *Programming in Mathematica*, Addison–Wesley Publishing Company, Reading, Massachusetts, 1997.
- [12] C. D. Meyer: *Matrix Analysis and Applied Linear Algebra*, Society for Industrial and Applied Mathematics, Philadelphia, 2000.
- [13] J. Stoer, R. Bulirsch: *Introduction to Numerical Analysis*, Springer–Verlag, New York, 1993.
- [14] H. Werner: *Kinematika. Predavanja—sažetak*, skripta, Građevinski fakultet, Zagreb, 2002.

- [15] S. Wolfram: *The Mathematica*, Second Edition, Addison–Wesley Publishing Company, Reading, Massachusetts, 1993.
- [16] O. C. Zienkiewicz, R. L. Taylor, J. Z. Zhu: *The Finite Element Method. It's Basis and Fundamentals*, Elsevier Butterworth–Heinemann, Oxford, 2005.
- [17] O. C. Zienkiewicz, R. L. Taylor: *The Finite Element Method for Solid and Structural Mechanics*, Elsevier Butterworth–Heinemann, Oxford, 2005.

Sažetak

Dinka Matković i Katja Mikulić

Kinematička kondenzacija

U radu je prikazan program koji služi za rješavanje ravninskih štapnih sistema uz uvažavanje kinematičkih ograničenja kinematičkom kondenzacijom. Program je izrađen unutar programskog sustava **Mathematica** te je vrlo jednostavan za upotrebu. Na primjeru je pokazan način na koji se program koristi te su uspoređena rješenja primjera dobivena metodom pomaka i rješenja dobivena uz uvažavanje kinematičkih ograničenja.

Ključne riječi: kinematička ograničenja, kinematička kondenzacija, kompjutorski program, ravninske štapne konstrukcije, metoda pomaka

Summary

Dinka Matković and Katja Mikulić

Kinematic Condensation

A computer programme for solving planar rod structures is presented in this paper. The programme incorporates kinematic constraints in the analysis by kinematic condensation. It was created within the **Mathematica** programming system and it is quite simple to use. An example has been given to show how the programme works. Results obtained by the displacement method and those obtained taking the kinematic constraints into account were compared.

Key words: kinematic constraints, kinematic condensation, computer programme, planar rod structures, displacement method