

**UNIVERZA V LJUBLJANI**

Fakulteta za strojništvo

# **Modalna analiza trirazsežnih vitkih paličnih struktur z metodo končnih elementov**

Zaključna naloga Univerzitetnega študijskega programa I. stopnje  
Strojništvo - Razvojno raziskovalni program

**Gašper Bizjan**

Ljubljana, avgust 2020



**UNIVERZA V LJUBLJANI**

Fakulteta za strojništvo

# **Modalna analiza trirazsežnih vitkih paličnih struktur z metodo končnih elementov**

Zaključna naloga Univerzitetnega študijskega programa I. stopnje  
Strojništvo - Razvojno raziskovalni program

**Gašper Bizjan**

Mentor: izr. prof. dr. Gregor Čepon, univ. dipl. inž.

Ljubljana, avgust 2020

## VLOGA ZA PREVZEM TEME ZAKLJUČNE NALOGE

Univerzitetni študijski program I. stopnje STROJNIŠTVO – Razvojno raziskovalni program

Št. zaključne naloge (izpolni Študentski referat): UNI/1394

Datum prejema vloge v ŠR: 20.8.2020

### Podatki o študentu:

Ime in priimek: GAŠPER BIZJAN Vpisna št. 23170101

Datum, kraj rojstva: 21.02.1998, LJUBLJANA

### Podatki o zaključni nalogi:

Naslov zaključne naloge (slovenski):

Modalna analiza trirazsežnih vitkih paličnih  
struktur z metodo končnih elementov

Naslov zaključne naloge (angleški):

Modal analysis of three-dimensional truss  
structures using the finite element method.

Mentor na FS: izr. prof. dr. GREGOR ČEPON

Somentor na FS: ✓

Veljavnost naslova teme je 6 mesecev od oddaje Vloge za prevzem.

Podpis študenta: Biz

Podpis mentorja: [Signature]

# Zahvala

---

Rad bi se zahvalil mentorju prof. dr. Gregorju Čeponu, brez katerega to delo ne bi nastalo. V veliko pomoč so mi bile tudi izkušnje as. Mihe Kodriča, ki mi je svetoval pri teoretični izpeljavi in programerskih trikih.

Zahvalil bi se tudi svoji družini, tako njihovi moralni kot tudi finančni podpori skozi vsa leta. Kot zadnje se zahvaljujem še partnerici, ki mi je med študijem stala ob strani.

Spodaj podpisani/-a Gašper Bizjan študent/-ka Fakultete za strojništvo Univerze v Ljubljani, z vpisno številko 23170101, avtor/-ica pisnega zaključnega dela študija z naslovom: Modalna analiza trirazsežnih vitkih paličnih struktur z metodo končnih elementov,

IZJAVLJAM,

1. \* ☒ a) da je pisno zaključno delo študija rezultat mojega samostojnega dela;  
b) da je pisno zaključno delo študija rezultat lastnega dela več kandidatov in izpolnjuje pogoje, ki jih Statut UL določa za skupna zaključna dela študija ter je v zahtevanem deležu rezultat mojega samostojnega dela;
2. da je tiskana oblika pisnega zaključnega dela študija istovetna elektronski obliki pisnega zaključnega dela študija;
3. da sem pridobil/-a vsa potrebna dovoljenja za uporabo podatkov in avtorskih del v pisnem zaključnem delu študija in jih v pisnem zaključnem delu študija jasno označil/-a;
4. da sem pri pripravi pisnega zaključnega dela študija ravnal/-a v skladu z etičnimi načeli in, kjer je to potrebno, za raziskavo pridobil/-a soglasje etične komisije;
5. da soglašam z uporabo elektronske oblike pisnega zaključnega dela študija za preverjanje podobnosti vsebine z drugimi deli s programsko opremo za preverjanje podobnosti vsebine, ki je povezana s študijskim informacijskim sistemom članice;
6. da na UL neodplačno, neizključno, prostorsko in časovno neomejeno prenašam pravico shranitve avtorskega dela v elektronski obliki, pravico reproduciranja ter pravico dajanja pisnega zaključnega dela študija na voljo javnosti na svetovnem spletu preko Repozitorija UL;
7. da dovoljujem objavo svojih osebnih podatkov, ki so navedeni v pisnem zaključnem delu študija in tej izjavi, skupaj z objavo pisnega zaključnega dela študija;
8. da dovoljujem uporabo mojega rojstnega datuma v zapisu COBISS.

V Ljubljani, 21. 8. 2020

Podpis avtorja/-ice: 

\* Obkrožite varianto a) ali b).

## **Modalna analiza trirazsežnih vitkih paličnih struktur z metodo končnih elementov**

Gašper Bizjan

Ključne besede:      metoda končnih elementov  
                         modalna analiza  
                         dinamska analiza  
                         palice  
                         python  
                         ansys mechanical APDL

Modalna analiza lastnih nihanj paličnih struktur v prostoru z metodo končnih elementov je numerična metoda reševanja realnega fizikalnega problema. Na osnovi predpostavk in omejitev fizikalne strukture izpeljemo lokalno masno in togostno matriko palice, s katerima preko transformacije iz lokalnega v globalni koordinatni sistem tvorimo sistemske matrike diskretizirane strukture žerjava. Za sistem enačb prikažemo reševanje problema lastnih vrednosti in pripadajočih lastnih vektorjev. Končni cilj naloge je izračun lastnih frekvenc in izris modalnih oblik žerjava znotraj lastnega računalniškega programa napisanega v okolju Python. Rezultate primerjamo z izračuni komercialnega programa Ansys Mechanical APDL in dokažemo pravilnost lastnega programa.

# Abstract

---

UDC 519.61:531.3:621.873(043.2)

No.: UN I/1394

## **Modal analysis of three-dimensional truss structures using the finite element method**

Gašper Bizjan

Key words:      finite element method  
                     modal analysis  
                     dynamic analysis  
                     truss  
                     python  
                     ansys mechanical APDL

The modal analysis of free vibration in truss structures in space with the finite element method is a numerical method for solving real physical problems. On the basis of certain assumptions and limitations of a physical structure we derive a local mass and stiffness matrix of a truss element. After applying a transformation from the local to global coordinate system, we assemble a system matrix of a discretized crane structure. The eigenvalue problem formulation was adopted to compute eigenvalues and the corresponding eigenvectors. The objective of the thesis is the computation of natural frequencies and to animate modal shapes of the crane using a custom Python based program. Results are compared to those generated by the commercially available program Ansys Mechanical APDL. We prove our Python program works with minimum deviation from the reference.



# Kazalo

---

Kazalo slik . . . . .	ix
Kazalo preglednic . . . . .	x
Seznam uporabljenih simbolov . . . . .	xi
Seznam uporabljenih okrajšav . . . . .	xii
<b>1 Uvod . . . . .</b>	<b>1</b>
1.1 Ozadje problema . . . . .	1
1.2 Cilji naloge . . . . .	2
<b>2 Teoretične osnove . . . . .</b>	<b>3</b>
2.1 Diskretizacija na končne elemente . . . . .	4
2.1.1 Fizikalni model - omejitve in predpostavke strukture . . . . .	4
2.1.2 Končni element . . . . .	5
2.2 Interpolacijska funkcija . . . . .	5
2.3 Izpeljava togostne in masne matrike . . . . .	6
2.3.1 Togostna in masna matrika za osno smer . . . . .	7
2.3.1.1 Diferencialna enačba . . . . .	7
2.3.1.2 Integralska variacijska formulacija . . . . .	8
2.3.1.3 Aproksimativno numerično reševanje . . . . .	9
2.3.1.4 Togostna in masna matrika za prečni smeri . . . . .	11
2.3.1.5 Diferencialna enačba za prečno smer . . . . .	11
2.3.1.6 Integralska variacijska formulacija . . . . .	13
2.3.1.7 Aproksimativno numerično reševanje . . . . .	14
2.3.2 Togostna in masna matrika v lokalnem KS . . . . .	15
2.3.3 Transformacija koordinat . . . . .	16
2.3.3.1 Pomik v glavnem koordinatnem sistemu . . . . .	17
2.3.3.2 Pospešek v glavnem koordinatnem sistemu . . . . .	18
2.3.3.3 Vozliščne sile v glavnem koordinatnem sistemu . . . . .	18
2.3.3.4 Globalna masna in togostna matrika . . . . .	19
2.4 Gibalna enačba sistema . . . . .	20
2.5 Upoštevanje robnih pogojev . . . . .	20
2.6 Analiza lastnih nihanj . . . . .	20

<b>3</b>	<b>Metodologija raziskave . . . . .</b>	<b>22</b>
3.1	Python program za modalno analizo . . . . .	22
3.1.1	Definicija sistema . . . . .	23
3.1.2	Definicija globalnih matrik končnega elementa . . . . .	24
3.1.3	Sestavljanje globalnih matrik v sistemsko matriko . . . . .	25
3.1.4	Brisanje prostostnih stopenj zaradi robnih pogojev . . . . .	27
3.1.5	Izračun lastnih vrednosti in vektorjev matrike . . . . .	27
3.2	Ansys Mechanical APDL . . . . .	28
<b>4</b>	<b>Rezultati . . . . .</b>	<b>29</b>
4.1	Lastne frekvence . . . . .	29
4.2	Lastne oblike . . . . .	30
<b>5</b>	<b>Diskusija . . . . .</b>	<b>31</b>
5.1	Število lastnih frekvenc in modalnih oblik . . . . .	31
5.2	Vrednosti lastnih frekvenc in modalnih oblik . . . . .	32
5.3	Kriterij gotovosti modalnih oblik . . . . .	33
<b>6</b>	<b>Zaključki . . . . .</b>	<b>34</b>
	<b>Literatura . . . . .</b>	<b>35</b>
	<b>Priloga A - Python programska koda . . . . .</b>	<b>36</b>

# Kazalo slik

---

Slika 1.1:	Žerjav kot primer palične konstrukcije. . . . .	1
Slika 2.1:	(a) Realna struktura žerjava, (b) diskretizirana struktura žerjava iz KE. . . . .	4
Slika 2.2:	Končni element v $x'y'z'$ koordinatnem sistemu. . . . .	5
Slika 2.3:	Lagrangevi interpolacijski funkciji. . . . .	6
Slika 2.4:	Diferencialni del končnega elementa v $x'$ -smeri. . . . .	7
Slika 2.5:	Delovanje osnih sil. . . . .	11
Slika 2.6:	(a) Diferencialni del KE v $y'$ -smeri, (b) deformacija palice v $y'$ -smeri. . . . .	12
Slika 2.7:	Potrebni rotaciji (a) okoli $z$ -osi za kot $\alpha$ in (b) okoli $y^\alpha$ -osi za kot $\beta$ za transformacijo KE (c) pri rotaciji $\alpha$ in (d) pri rotaciji $\beta$ iz lokalnega v globalni KS. . . . .	16
Slika 3.1:	Blokovni diagram algoritma Python programa. . . . .	22
Slika 3.2:	Model žerjava za katerega bomo preračunali lastne frekvence in oblike. . . . .	23
Slika 3.3:	Povezovanje vozlišč v tri končne elemente. . . . .	24
Slika 4.1:	Primerjava prvih treh modalnih oblik kjer je: (a) prva modalna oblika pri lastni frekvenci $f_1 = 0,005$ Hz; (b) druga modalna oblika pri lastni frekvenci $f_2 = 0.009$ Hz; in (c) tretja modalna oblika pri tretji lastni frekvenci $f_3 = 0.013$ Hz. . . . .	30
Slika 5.1:	Lastne frekvence $f_o$ v odvisnosti od modalne oblike. . . . .	32
Slika 5.2:	Grafični prikaz ujemanja lastnih vektorjev predstavljen kot (a) 2-D MAC vrednosti in (b) 3-D MAC vrednosti (prvih 12 modalnih oblik). . . . .	33

## Kazalo preglednic

---

Preglednica 3.1:	Materialni in geometrijski parametri. . . . .	24
Preglednica 3.2:	KE s pripadajočimi vozlišči in prostostnimi stopnjami. . .	25
Preglednica 4.1:	Lastne frekvence konstrukcije žerjava. . . . .	29

## Seznam uporabljenih simbolov

Oznaka	Enota	Pomen
$L$	m	dolžina posamezne palice
$A$	m <sup>2</sup>	površina preseka palice
$\rho$	kg/m <sup>2</sup>	gostota palic
$E$	Pa	modul elastičnosti
$[M]$	kg	masna matrika
$[K]$	Pa m <sup>2</sup> /m	togostna matrika
$[T]$	/	transformacijska matrika
$[I]$	/	identiteta
$F$	N	zunanje sile
$F_e$	N	elastične sile
$F_v$	N	vztrajnostne sile
$N$	N	notranja osna sila
$t$	s	čas
$u$	m	pomik
$\dot{u}$	m/s	hitrost
$\ddot{u}$	m/s <sup>2</sup>	pospešek
$U$	m	diskreten pomik
$x$	/	smer globalnega kartezijskega koordinatnega sistema
$y$	/	smer globalnega kartezijskega koordinatnega sistema
$z$	/	smer globalnega kartezijskega koordinatnega sistema
$x'$	/	smer lokalnega kartezijskega koordinatnega sistema
$y'$	/	smer lokalnega kartezijskega koordinatnega sistema
$z'$	/	smer lokalnega kartezijskega koordinatnega sistema
$\bar{x}$	/	smer po dolžini palice
$\Psi$	/	Lagrangev interpolacijski polinom
$\varphi$	/	kot med $x'$ in $\bar{x}$
$\alpha$	/	kot prve transformacije končnega elementa
$\beta$	/	kot druge transformacije končnega elementa
$X$	m	amplituda
$\omega$	rad/s	kotna frekvenca
$\omega_o$	rad/s	lastna kotna frekvenca
$f$	Hz	frekvenca
$f_o$	Hz	lastna frekvenca
$\lambda$	rad <sup>2</sup> /s <sup>2</sup>	lastna vrednost

### Indeksi

lok	lokalna
glob	globalna
sist	sistemska

## Seznam uporabljenih okrajšav

---

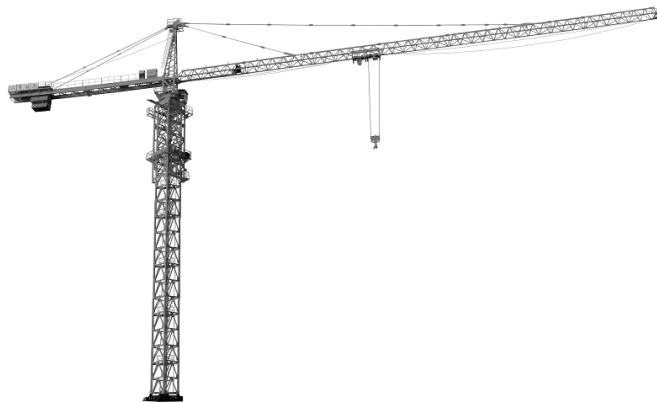
Okrajšava	Pomen
3D	tridimenzionalni (prostor)
APDL	ANSYS parametrični oblikovni jezik ( <i>ang. ANSYS Parametric Design Language</i> )
KE	končni element
KS	koordinatni sistem
MAC	kriterij gotovosti modalnih oblik ( <i>ang. modal assurance criterion</i> )
MKE	metoda končnih elementov
RP	robni pogoji

# 1 Uvod

## 1.1 Ozadje problema

Palične strukture (npr. žerjavi - slika 1.1) omogočajo doseganje velikih višin in dolžin, zato so v praksi pogost način gradnje. Zaradi njihove vitkosti vzbujanje v okolici lastnih frekvenc povzroči nihanje z velikimi amplitudami, kar lahko privede do porušitve konstrukcije. Potrebno se je izogniti obratovanju v okolici resonanc in tako zagotoviti varnost gradnje. Modalna analiza nam omogoča izračun resonančnih frekvenc in pripadajočih modalnih oblik konstrukcije.

Med dinamskimi analizami konstrukcij so analitične rešitve lastnih nihanj znane le za enostavne geometrije s specifičnimi robnimi pogoji. Eksaktne rešitve vibracij nam sicer nudijo razumevanje fizikalnega problema, toda v praksi so strukture vedno kompleksnejše. Kot močno numerično orodje pri tovrstni analizi se je izkazala metoda končnih elementov (MKE), ki kljub aproksimativnim rešitvam omogoča reševanje izredno kompleksnih večdimenzijskih sistemov.



Slika 1.1: Žerjav kot primer palične konstrukcije.

## 1.2 Cilji naloge

Osrednja problematika naloge je izdelava lastne računalniške simulacije za iskanje lastnih frekvenc in oblik trirazsežnih vitkih paličnih struktur v okolju Python, ki temelji na metodi končnih elementov.

Za izpeljavo enačbe, ki omogoča numerično reševanje problema, je potrebna definicija splošnega fizikalnega modela, na podlagi katerega izpeljemo gibalno enačbo končnega elementa palice. Na podlagi dobljenega sistema enačb lahko preko karakteristične enačbe poiščemo lastne vrednosti in pripadajoče lastne vektorje sistema.

Izdelani numerični model z metodo končnih elementov uporabimo na konkretni palični konstrukciji - žerjavu. Definiramo materialne in geometrijske lastnosti ter vozlišča paličja in medsebojne povezave. Izpeljano globalno masno matriko in togostno matriko uporabimo na vsakem končnem elementu in te nato sestavimo v sistemski matriki. Definiramo tudi robne pogoje oz. pogoje vpetja. Sledi izračun lastnih frekvenc in izris lastnih oblik žerjavne konstrukcije.

Rezultate lastnega programa primerjamo tudi s komercialnim programom Ansys znotraj okolja Ansys Mechanical APDL in verificiramo njegovo pravilnost.



## 2 Teoretične osnove

Osnovna ideja metode končnih elementov (MKE) je iskanje rešitev kompleksnega realnega problema (kompleksna geometrija, nehomogeni materiali ali obremenitve) z uporabo poenostavljenega modela, ki sestoji iz številnih delov, imenovanih končni elementi (KE) medsebojno povezanih v vozlišča. Torej zvezno območje (domeno) diskretiziramo v podobmočja, zato je MKE po naravi aproksimativna metoda in je uporabna takrat, kadar analitična rešitev ni mogoča. Metoda je tudi numerična, saj se končna oblika algebraskih enačb izrazi v sistemskih matrikah, kar zelo učinkovito omogoča računalniško reševanje sistemov velikega števila KE.

Neodvisno od geometrije in celo od vrste fizikalnega problema, ki ga želimo rešiti z MKE, so osnovni koraki formulacije vedno enaki. Preko njih bomo na osnovi pregleda literature izpeljali enačbe MKE ( [1] - [5] ) in preko njih reševali dinamsko analizo lastnih nihanj ( [6], [7] ).

**Korak 1:** Razdelitev fizikalnega modela v diskretne elemente (diskretizacija).

Na tem mestu izberemo tip, število in velikost KE. Te izbire so odvisne predvsem od narave fizikalnega problema, željene natančnosti in hitrosti numerične simulacije.

**Korak 2:** Izbira interpolacijskega modela.

Izberemo obliko poljubne interpolacijske funkcije, ki določa aproksimacijo primarne spremenljivke. Najpogosteje gre za polinomske funkcije.

**Korak 3:** Izpeljava matrik končnega elementa.

Za KE v svojem lokalnem koordinatnem sistemu (KS) diferencialno enačbo fizikalnega problema zapišemo kot integralsko variacijsko formulacijo. Uporabimo prej izbrano interpolacijsko funkcijo in dobimo značilne matrike sistema v lokalnem KS. Sledi transformacija koordinat v globalne matrike.

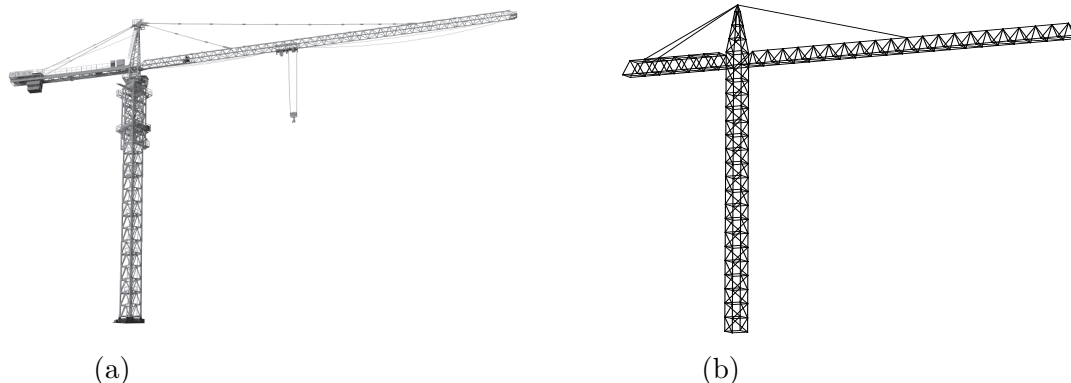
**Korak 4:** Sestavljanje matrik posameznega končnega elementa v sistemske matrike. Matrike posameznih elementov razširimo na vse prostostne stopnje in jih seštejemo.

**Korak 5:** Upoštevanje začetnih pogojev ter robnih in prehodnih pogojev.

**Korak 6:** Reševanje sistema enačb.

## 2.1 Diskretizacija na končne elemente

Obravnavana konstrukcija je v celoti sestavljena iz palic. Vsaka izmed njih predstavlja posamezni končni element.



Slika 2.1: (a) Realna struktura žerjava, (b) diskretizirana struktura žerjava iz KE.

### 2.1.1 Fizikalni model - omejitve in predpostavke strukture

Za učinkovito modeliranje struktur je potrebno opredeliti pogoje, omejitve ter tudi predpostavke našega sistema [3].

Za tak sistem velja:

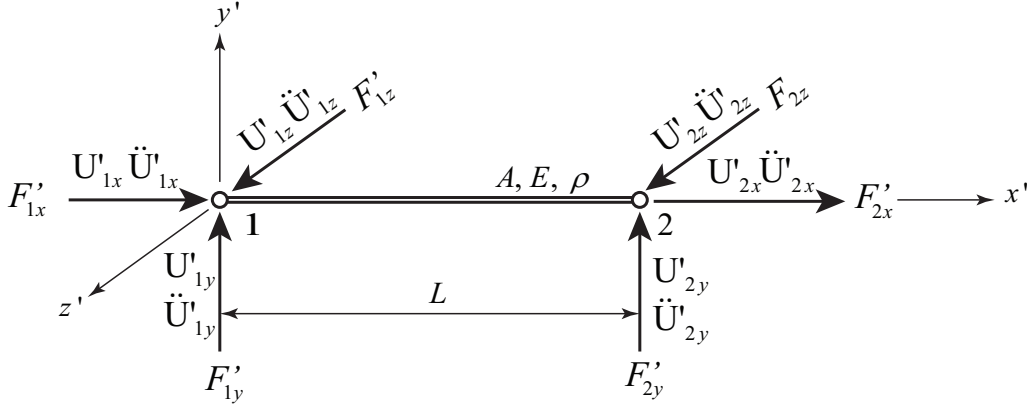
- vse palice so medsebojno členkasto vpete (prosta rotacija) z zunanjo členkasto podporo;
- zunanje sile delujejo le v členkih, torej je prisotna le notranja tlačna in natezna osna sila (notranjih strižnih sil in notranjih momentov ni); zunanji momenti niso prisotni;
- nosilni element je deformabilno telo v aksialni smeri; uklon ni prisoten in velja teorija majhnih pomikov;
- poznamo naslednje materialne lastnosti jeklenih palic: dolžino  $L$ , konstantni prerez  $A$ , gostoto  $\rho$  in modul elastičnosti  $E$ ; material je homogeno izotropen.

Poleg tega veljajo dodatne predpostavke, ki doprinesejo k poenostavitvi modela, a nimajo bistvenega vpliva na točnost rezultatov:

- obravnavamo prostorski (tridimenzionalni) problem v  $xyz$  kartezijskem koordinatnem sistemu;
- zanemarimo lastno težo palic (ni vpliva gravitacijskega pospeška);
- temperatura je po sistemu konstantna;
- sistem deluje brez energijskih izgub (ni dušenja);

### 2.1.2 Končni element

Analizirajmo posamezni element strukture (palico) izven globalnega koordinatnega sistema  $xyz$  (koord. sist. celotne strukture) v lokalnem koordinatnem sistemu  $x'y'z'$  (slika 2.2). Palica je raven enodimenzionalen element dolžine  $L$  in prereza  $A$  iz linearno-elastičnega gradiva z modulom elastičnosti  $E$  in gostote  $\rho$ . V krajiščih je obremenjena s točkovnima silama, ki ju razdelimo na komponente glede na lokalni koordinatni sistem. Sile v vozliščih povzročijo pomik  $u = u(x', y', z', t)$ , ki je v splošnem funkcija pozicije na palici in časa.



Slika 2.2: Končni element v  $x'y'z'$  koordinatnem sistemu.

## 2.2 Interpolacijska funkcija

Po vsakem podobmočju oziroma KE primarno spremenljivko  $u$  aproksimiramo z enostavno poljubno funkcijo, ki mora zadostiti določenim pogojem [2]:

- funkcija mora biti zvezna po celotnem KE,
- mora biti zvezno odvedljiva,
- računsko mora biti obvladljiva.

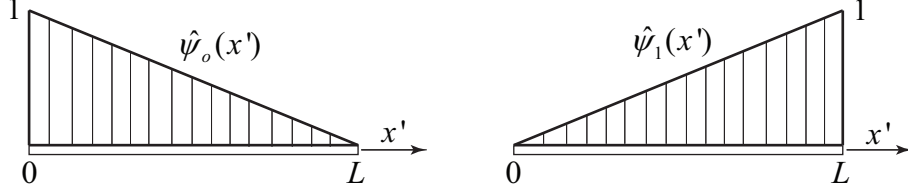
Za naš enodimenzionalni KE vse pogoje zadosti polinomska aproksimacija oblike  $\hat{u}$ . Ker so pomiki v posameznih smereh KS medsebojno neodvisni, lahko aproksimacijo zapišemo ločeno za  $x'$ -smer.

$$u(x') \approx \hat{u}(x') = \sum_{k=0}^N a_k \hat{\Psi}_k(x') \quad (2.1)$$

$$\hat{\Psi}_k(x') = \delta_{ik} = \begin{cases} 1 & ; x'_i = x'_k \\ 0 & ; x'_i \neq x'_k \end{cases}$$

$$a_i = \sum_{k=0}^N a_k \hat{\Psi}_k(x') = \hat{u}(x'_i)$$

Aproksimacija v enačbi (2.1) je grajena na diskretnih vrednostih  $\{a_k\}$  ter Lagrangevih interpolacijskih funkcijah  $\{\hat{\Psi}_k\}$ , kjer je  $\delta_{ik}$  Kroneckerjev delta tenzor. Tako dobijo koeficienti  $\{a_i\}$  fizikalni pomen, saj predstavljajo diskretne vrednosti aproksimacije pomikov  $\{\hat{u}(x'_i)\}$ , pri čemer izvedemo aproksimacijo rešitve na območju našega KE  $x' \in [0, L]$ , ki ima  $N = 2$  vozlišč. V primeru dvovozliščnega KE je aproksimacija pomika  $\hat{u}(x')$  zasnovana na dveh polinomih prvega reda v enačbi (2.2).



Slika 2.3: Lagrangevi interpolacijski funkciji.

$$\hat{u}(x') = a_0 \hat{\Psi}_0(x') + a_1 \hat{\Psi}_1(x') = a_0 \left(1 - \frac{x'}{L}\right) + a_1 \left(\frac{x'}{L}\right) \quad (2.2)$$

## 2.3 Izpeljava togostne in masne matrike

Na tej točki se prvič vprašamo o fiziki problema lastnih nihanj [6]. Lastno nihanje nastopi, kadar nek sistem vzbudimo (apliciramo silo oziroma vnesemo energijo) in nato pustimo nihati. Pri tem se energija ves čas pretvarja med kinetično energijo gibanja in potencialno energijo zaradi togosti sistema. Za uspešno analizo moramo pojav proste vibracije zapisati v tako imenovano gibalno enačbo.

Gibalno enačbo izpeljemo preko dinamično-togostne metode za analizo paličja [3]. Upoštevamo splošne aksiome principa enakosti sil ter drugi Newtonov zakon. Tako vzpostavimo ravnotežje med zunanje delujočimi silami  $\{F(t)\}$  in notranjimi silami (elastičnimi  $\{F_e(t)\}$  in vztrajnostnimi silami  $\{F_v(t)\}$ ) v vseh členkih oziroma vozliščih sistema. Dobimo sledeč sistem ravnotežnih enačb:

$$\{F_v(t)\} + \{F_e(t)\} = \{F(t)\} \dots \text{vektor velikosti } P. \quad (2.3)$$

Število enačb, ki sistem opisujejo, pogojuje število prostostnih stopenj sistema  $P$ . Ker obravnavamo prostorski model ima vsako vozlišče  $v_i$  tri prostostne stopnje  $N = 3$ . Skupno je to  $P = N \cdot v_i$ .

Posledice delovanja sil opazujemo v obliki pomikov  $u(t)$ , hitrosti  $\dot{u}(t)$  in pospeškov  $\ddot{u}(t)$  vozlišč ter preko vpeljave masne matrike  $[M]$ , ki podaja porazdelitev mas v nihajočem sistemu, in togostne matrike  $[K]$ , ki podaja velikosti sil zaradi deformacije.

$$\{F_v(t)\} = [M]\{\ddot{u}\} \quad (2.4)$$

$$\{F_e(t)\} = [K]\{u\} \quad (2.5)$$

Ko zgornje enačbe vstavimo v enačbo (2.3), dobimo matrični zapis gibalne enačbe sistema.

$$[M]\{\ddot{u}\} + [K]\{u\} = \{F(t)\} \quad (2.6)$$

Iskanje lastnih frekvenc in modalnih oblik je odvisno od reševanja dane gibalne enačbe, za katero moramo sprva formulirati ustrezni sistemski matriki  $[K]$  in  $[M]$ .

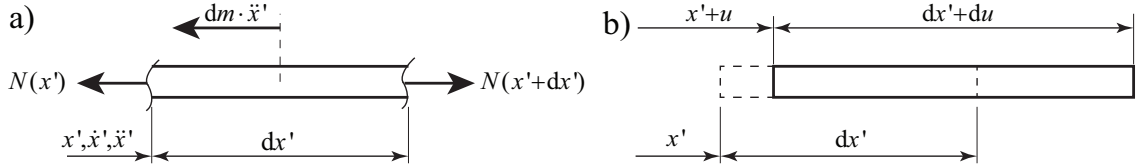
Gibalno enačbo apliciramo na KE, definiran v poglavju 2.1.2, in upoštevamo, da so pomiki in sile v  $x'$ ,  $y'$  ter  $z'$ -smeri KS medsebojno neodvisni. Tako lahko opazovanje KE ločimo na posamezni koordinatni osi. Dodatno velja tudi, da so v vsakem trenutku sile v ravnotežju in lahko posledično z vidika izpeljave časovno odvisnost pomika zanemarimo.

### 2.3.1 Togostna in masna matrika za osno smer

#### 2.3.1.1 Diferencialna enačba

Med procesom raztegovanja za smer  $x'$  obravnavamo diferencialno majhni del palice na sliki 2.4a [4].

Z upoštevanjem ravnotežja statičnih in dinamičnih sil, deformacijske konsistentnosti in konstitucijskega obnašanja elementa lahko v diferencialni enačbi definiramo elastično silo  $F_e(t)$ . Z upoštevanjem drugega časovnega odvoda pomika ter dejstva, da ima KE maso, lahko preko drugega Newtonovega zakona in D'Alembertovega načela v diferencialno enačbo vpeljemo vztrajnostno silo  $F_v(t)$ , ki deluje nasprotno smeri pomika in tako zajamemo dinamične učinke.



Slika 2.4: Diferencialni del končnega elementa v  $x'$ -smeri.

Zapišemo vsoto vseh sil, za katero velja, da je v vsakem trenutku ravnotežna.

$$\begin{aligned}
 \sum_{i=1}^N F_i &= 0 \\
 -N(x') + N(x' + dx') - dm \ddot{x}' &= 0 \\
 -N(x') + N(x') + dN(x') - dm \ddot{x}' &= 0 \\
 dN(x') - \rho A dx' \ddot{x}' &= 0 \\
 \frac{dN(x')}{dx'} - \rho A \ddot{x}' &= 0
 \end{aligned} \tag{2.7}$$

Kjer je  $N = N(x')$  notranja osna sila,  $dm = dm(x') = \rho A dx'$  diferencialni del mase in  $\ddot{x}'$  pospešek diferencialno majhnega dela.

Z upoštevanjem deformacijske konsistentnosti in konstitucijskega obnašanja lahko za majhne pomike  $u = u(x')$  upoštevamo linearno elastično zvezo med silo in deformacijo

(Hookov zakon [8]). Glede na sliko 2.4b velja  $\frac{du}{dx'} = \varepsilon_{x'x'}$ . Preko Hookovega zakona lahko izpeljemo povezavo med notranjo osno silo in pomikom.

$$\begin{aligned}\frac{du(x')}{dx'} &= \varepsilon_{x'x'} = \frac{\sigma_{x'x'}}{E} = \frac{N(x')}{EA} \\ N(x') &= EA \frac{\sigma_{x'x'}}{E} = EA \varepsilon_{x'x'} \\ N(x') &= EA \frac{du(x')}{dx'} \\ \frac{dN(x')}{dx'} &= EA \frac{d^2u(x')}{dx'^2}\end{aligned}\tag{2.8}$$

V ravnotežno enačbo (2.7) vstavimo izpeljano zvezo med sekundarno spremenljivko  $N(x')$  in primarno spremenljivko  $u(x')$  iz enačbe (2.8).

$$EA \frac{d^2u(x')}{dx'^2} - \rho A \ddot{x}' = 0\tag{2.9}$$

Upoštevamo povezavo med lokalno koordinato  $x'$  ter pomikom  $u$  in poiščimo časovne odvode.

$$\begin{aligned}x' &= x'_o + u \\ \dot{x}' &= 0 + \dot{u} \\ \ddot{x}' &= \ddot{u}\end{aligned}\tag{2.10}$$

Tako dobimo vodilno diferencialno enačbo problema.

$$EA \frac{d^2u(x')}{dx'^2} - \rho A \ddot{u}(x') = 0\tag{2.11}$$

### 2.3.1.2 Integralska variacijska formulacija

Reševanje izpeljane vodilne enačbe problema prevedemo na reševanje integralske enačbe. Povzeto po [4]. Enačbo pomnožimo s poljubno funkcijo  $v(x')$ , ki je v območju  $x' \in [0, L]$  zvezno odvedljiva, ter integriramo po celotnem območju KE. Dobimo osnovno obliko integralske formulacije.

$$\int_0^L \left[ EA \frac{d^2u(x')}{dx'^2} - \rho A \ddot{u}(x') \right] v(x') dx' = 0\tag{2.12}$$

$$\begin{aligned}\int_0^L EA \frac{d^2u(x')}{dx'^2} v(x') dx' - \int_0^L \rho A \ddot{u}(x') v(x') dx' &= 0 \\ \int_0^L EA \frac{d^2u(x')}{dx'^2} v(x') dx' &= \int_0^L \rho A \ddot{u}(x') v(x') dx'\end{aligned}\tag{2.13}$$

Levo stran enačbe (2.13) integriramo po metodi "per-partes" in nato vstavimo v celoto.

$$\begin{aligned}\int U dV &= UV - \int V dU \\ \left( \begin{array}{ll} U = v(x') & dV = EA \frac{d^2u}{dx'^2} dx' \\ dU = \frac{dv}{dx'} dx' & V = EA \frac{du}{dx'} \end{array} \right)\end{aligned}\tag{2.14}$$

$$\int_0^L EA \frac{d^2 u(x')}{dx'^2} v(x') dx' = v(x') EA \frac{du(x')}{dx'} \Big|_0^L - \int_0^L EA \frac{du(x')}{dx'} \frac{dv(x')}{dx'} dx' \quad (2.15)$$

Opazimo, da se red diferencialne enačbe zmanjša za eno stopnjo. Tako dobimo šibko obliko integralske formulacije.

$$\int_0^L EA \frac{du(x')}{dx'} \frac{dv(x')}{dx'} dx' = v(x') EA \frac{du(x')}{dx'} \Big|_0^L - \int_0^L \rho A \ddot{u}(x') v(x') dx' \quad (2.16)$$

Z ozirom na enačbo (2.7) lahko v šibki obliki integralske formulacije izrazimo notranjo osno silo.

$$\int_0^L EA \frac{du(x')}{dx'} \frac{dv(x')}{dx'} dx' = v(x') N(x') \Big|_0^L - \int_0^L \rho A \ddot{u}(x') v(x') dx' \quad (2.17)$$

$$\int_0^L EA \frac{du(x')}{dx'} \frac{dv(x')}{dx'} dx' = v(L) N(L) - v(0) N(0) - \int_0^L \rho A \ddot{u}(x') v(x') dx' \quad (2.18)$$

Vrednosti  $N(0)$  in  $N(L)$  sta notranji osni sili na robovih palice. Ob znanih robnih pogojih oziroma pogojih v vozlišču KE sta znani vrednosti. Šibka oblika integralske formulacije vključuje robne vrednosti sekundarne spremenljivke. Hkrati imamo nižji red diferencialne enačbe, ki nam zmanjša zahteve glede aproksimacijske funkcije pri aproksimativnem reševanju.

### 2.3.1.3 Aproksimativno numerično reševanje

V poglavju 2.2 smo definirali izbrano aproksimacijsko funkcijo primarne spremenljivke pomika  $\hat{u}(x')$  z uporabo dveh polinomov prvega reda v enačbi (2.2). Konstanti  $a_0$  in  $a_1$  predstavljata diskretno vrednosti pomika v vozliščih končnega elementa.

$$\hat{u}(x') = U'_{1x} \left(1 - \frac{x'}{L}\right) + U'_{2x} \left(\frac{x'}{L}\right) \quad (2.19)$$

Enačbo (2.18) zapišemo z aproksimacijo rešitve  $\hat{u}(x')$ .

$$\int_0^L EA \frac{d\hat{u}(x')}{dx'} \frac{dv(x')}{dx'} dx' = v(L) N(L) - v(0) N(0) - \int_0^L \rho A \ddot{\hat{u}}(x') v(x') dx' \quad (2.20)$$

$$\begin{aligned} \int_0^L EA \left[ \frac{-U'_{1x}}{L} + \frac{U'_{2x}}{L} \right] \frac{dv(x')}{dx'} dx' &= \dots \\ \dots &= v(L) N(L) - v(0) N(0) - \int_0^L \rho A \left[ \ddot{U}'_{1x} \left(1 - \frac{x'}{L}\right) + \ddot{U}'_{2x} \left(\frac{x'}{L}\right) \right] v(x') dx' \end{aligned} \quad (2.21)$$

V zapisani enačbi so neznane štiri vozliščne vrednosti KE:

- dve primarni vrednosti pomika  $U'_{1x}$  in  $U'_{2x}$  oziroma pospeška  $\ddot{U}'_{1x}$  in  $\ddot{U}'_{2x}$ ,
- dve sekundarni vrednosti osnih sil  $N(0) = N_1$  in  $N(L) = N_2$ .

Za določitev štirih neznank potrebujemo štiri enačbe:

- dve izhajata iz poznavanja robnega pogoja KE,
- drugi dve dobimo z ustrezno izbiro funkcije  $v(x')$ .

V skladu z Galerkinovim pristopom izbire poljubnih funkcij  $v_k(x')$  so le-te takšne kot Lagrangeve interpolacijske funkcije  $\hat{\Psi}_k(x')$ , ki so bile uporabljene pri aproksimaciji  $\hat{u}(x')$ . Torej  $v_k(x') \in \hat{\Psi}_k(x')$ .

$$\begin{aligned} v_1(x') &= 1 - \frac{x'}{L} \\ v_2(x') &= \frac{x'}{L} \end{aligned} \quad (2.22)$$

Prvo funkcijo  $v_1(x')$  vstavimo v enačbo (2.21) in rešimo integralsko enačbo.

$$\begin{aligned} \int_0^L EA \left[ \frac{-U'_{1x}}{L} + \frac{U'_{2x}}{L} \right] \frac{-1}{L} dx' &= \left( 1 - \frac{L}{L} \right) N(L) - \left( 1 - \frac{0}{L} \right) N(0) - ... \\ &... - \int_0^L \rho A \left[ \ddot{U}'_{1x} \left( 1 - \frac{x'}{L} \right) + \ddot{U}'_{2x} \left( \frac{x'}{L} \right) \right] \left( 1 - \frac{x'}{L} \right) dx' \end{aligned} \quad (2.23)$$

$$\frac{EA}{L} (U'_{1x} - U'_{2x}) = -N_1 - \left( \frac{1}{3} \rho AL \ddot{U}'_{1x} + \frac{1}{6} \rho AL \ddot{U}'_{2x} \right) \quad (2.24)$$

Kjer je masa končnega elementa  $m = \rho AL$ . Enačbo preuredimo.

$$\frac{1}{3} m \ddot{U}'_{1x} + \frac{1}{6} m \ddot{U}'_{2x} + \frac{EA}{L} (U'_{1x} - U'_{2x}) = -N_1 \quad (2.25)$$

Nato v enačbo (2.21) vstavimo še drugo funkcijo  $v_2(x')$  in rešimo integralsko enačbo.

$$\begin{aligned} \int_0^L EA \left[ \frac{-U'_{1x}}{L} + \frac{U'_{2x}}{L} \right] \frac{1}{L} dx' &= \left( \frac{L}{L} \right) N(L) - \left( \frac{0}{L} \right) N(0) - ... \\ &... - \int_0^L \rho A \left[ \ddot{U}'_{1x} \left( 1 - \frac{x'}{L} \right) + \ddot{U}'_{2x} \left( \frac{x'}{L} \right) \right] \left( \frac{x'}{L} \right) dx' \end{aligned} \quad (2.26)$$

$$\frac{EA}{L} (-U'_{1x} + U'_{2x}) = N_2 - \left( \frac{1}{6} \rho AL \ddot{U}'_{1x} + \frac{1}{3} \rho AL \ddot{U}'_{2x} \right) \quad (2.27)$$

Enačbo preuredimo.

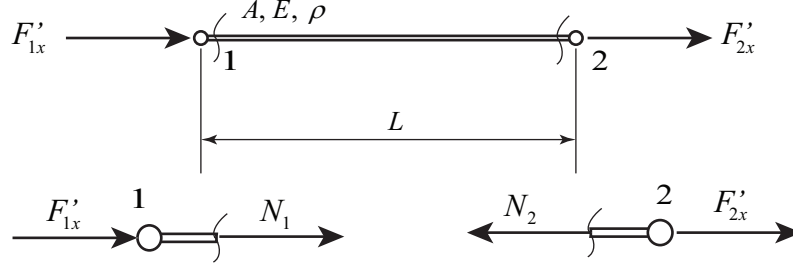
$$\frac{1}{6} m \ddot{U}'_{1x} + \frac{1}{3} m \ddot{U}'_{2x} + \frac{EA}{L} (-U'_{1x} + U'_{2x}) = N_2 \quad (2.28)$$



Po sliki (2.5) vidimo, da se osni sili  $N_1$  in  $N_2$  nahajata v vozliščih, kakor tudi sili  $F'_{1x}$  in  $F'_{2x}$ .

$$F'_{1x} + N_1 = 0$$

$$F'_{2x} - N_2 = 0$$



Slika 2.5: Delovanje osnih sil.

Zgornji enakosti vstavimo v enačbo (2.25) in enačbo (2.28). Tako smo izpeljali enačbi KE za vplive v aksialni smeri.

$$\begin{aligned} \frac{1}{3}m\ddot{U}'_{1x} + \frac{1}{6}m\ddot{U}'_{2x} + \frac{EA}{L}(U'_{1x} - U'_{2x}) &= F'_{1x} \\ \frac{1}{6}m\ddot{U}'_{1x} + \frac{1}{3}m\ddot{U}'_{2x} + \frac{EA}{L}(-U'_{1x} + U'_{2x}) &= F'_{2x} \end{aligned} \quad (2.29)$$

Matrični zapis zgornjih enačb:

$$\begin{bmatrix} \frac{m}{3} & \frac{m}{6} \\ \frac{m}{6} & \frac{m}{3} \end{bmatrix} \begin{Bmatrix} \ddot{U}'_{1x} \\ \ddot{U}'_{2x} \end{Bmatrix} + \begin{bmatrix} \frac{EA}{L} & -\frac{EA}{L} \\ -\frac{EA}{L} & \frac{EA}{L} \end{bmatrix} \begin{Bmatrix} U'_{1x} \\ U'_{2x} \end{Bmatrix} = \begin{Bmatrix} F'_{1x} \\ F'_{2x} \end{Bmatrix} \quad (2.30)$$

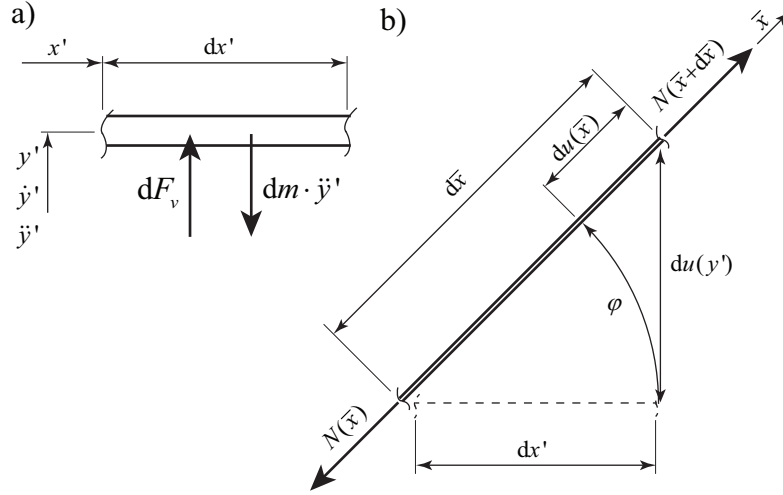
$$\frac{m}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} \ddot{U}'_{1x} \\ \ddot{U}'_{2x} \end{Bmatrix} + \frac{EA}{L} \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \begin{Bmatrix} U'_{1x} \\ U'_{2x} \end{Bmatrix} = \begin{Bmatrix} F'_{1x} \\ F'_{2x} \end{Bmatrix}$$

#### 2.3.1.4 Togostna in masna matrika za prečni smeri

Ponovno analiziramo KE v lokalnem koordinatnem sistemu  $x'y'z'$  (slika 2.2).

#### 2.3.1.5 Diferencialna enačba za prečno smer

Kot za osno  $x'$ -smer obravnavajmo  $y'$ -smer in  $z'$ -smer, ki sta pravokotni na palico in jima pripadata pomika  $u = u(y', t)$  in  $u = u(z', t)$ . Izpeljavo gibalne enačbe bomo prikazali na primeru  $y'$ -smeri, toda zaradi osne simetrije KE sta obe prečni smeri pomika enakovredni.



Slika 2.6: (a) Diferencialni del KE v  $y'$ -smeri, (b) deformacija palice v  $y'$ -smeri.

Obravnavamo diferencialno majhen del palice (slika 2.6a), vendar ga tokrat pomaknemo pravokotno na os palice. Z upoštevanjem drugega časovnega odvoda pomika ter dejstva, da ima KE maso, lahko preko drugega Newtonovim zakona in D'Alembertovega načela v diferencialno enačbo vpeljemo vztrajnostno silo  $F_v(t)$ , ki deluje v nasprotno smer gibanja. Ponovno predpostavimo neodvisnost spremenljivk od časa  $t$ , saj so opazovane sile ves čas v ravnovesju. Elastična sila  $F_e(t)$  (in notranja osna sila) zaradi pomikov, ki so na palico pravokotni, ni prisotna. To dokažemo na sledeč način: Zamislimo si primer, ko sili v členkih delujeta tako, da dobimo razteg palice v  $x'y'$ -ravnini (slika 2.6b). Ker sili delujeta v členkih, tako notranji moment kot strižna sila nista prisotna.

Zapišemo ravnotežje notranjih sil  $N = N(\bar{x})$ , ki nastanejo zaradi raztega palice v  $\bar{x}$ -smeri. V tem koraku vztrajnostne sile ne upoštevamo.

$$\begin{aligned}
 \sum_{i=1}^N F_i &= 0 \\
 -N(\bar{x}) + N(\bar{x} + d\bar{x}) &= 0 \\
 -N(\bar{x}) + N(\bar{x}) + dN(\bar{x}) &= 0 \\
 dN(\bar{x}) &= 0 \\
 \frac{dN(\bar{x})}{d\bar{x}} &= 0
 \end{aligned} \tag{2.31}$$

Enačbo (2.8) za zvezo med notranjo osno silo  $N(\bar{x})$  in za pomik  $u(\bar{x})$  vstavimo v zgornjo ravnovesno enačbo.

$$\frac{dN(\bar{x})}{d\bar{x}} = E A \frac{d^2 u(\bar{x})}{d\bar{x}^2} \tag{2.32}$$

Poglejmo odvisnost osne sile v  $\bar{x}$ -smeri od pomika vozlišč v  $y'$ -smeri. Torej iščemo odvisnost  $N(\bar{x}) = f(u(y'))$ . Pomik  $u(\bar{x})$  iz geometrije diferencialnega elementa zapišemo kot funkcijo  $u(y')$ .

$$du(\bar{x}) = d\bar{x} - dx' = \frac{du(y')}{\sin(\varphi)} - \frac{du(y')}{\tan(\varphi)} = du(y') \frac{1 - \cos(\varphi)}{\sin(\varphi)} \tag{2.33}$$

Predpostavimo teorijo majhnih pomikov, ki omejuje velikost kota  $\varphi$ . Za dovolj majhne vrednosti kota izračunamo izlimitirano funkcijo  $du(\bar{x})$ .

$$\lim_{(du(y'), \varphi) \rightarrow (0,0)} \left( du(y') \frac{1 - \cos(\varphi)}{\sin(\varphi)} \right) = 0 \quad (2.34)$$

Z ozirom na enačbo (2.32), za osno silo tako velja sledeča enačba.

$$\begin{aligned} \frac{dN(\bar{x})}{d\bar{x}} &= E A \frac{d^2}{d\bar{x}^2} \left( \cancel{du(y') \frac{1 - \cos(\varphi)}{\sin(\varphi)}}^0 \right) = 0 \\ dN(\bar{x}) &= 0 \end{aligned} \quad (2.35)$$

Za dovolj majhne pomike je torej vpliv pomikov vozlišč v  $y'$ -smeri (v primerjavi z vplivi pomikov v  $x'$ -smer) na osno silo  $N(\bar{x})$  zanemarljiv.

Vrnemo se k sliki 2.6a in ob spoznanju, da pri pomiku v  $y'$ -smeri ni aksialnih sil  $N$ , ter z upoštevanjem vztrajnostne sile  $F_v$  zapišemo ravnotežno enačbo diferencialnega elementa. Zaradi integracije po dolžini palice predpostavimo, da je  $y'$  funkcija  $x'$ , torej je  $dx' = dy'$  in je  $y' \in [0, L]$ .

$$\begin{aligned} \sum_{i=1}^N F_i &= 0 \\ dF'_y - dF_v &= 0 \\ dF'_y - dm \ddot{y}' &= 0 \\ dF'_y - \rho A dy' \ddot{y}' &= 0 \\ \frac{dF'_y}{dy'} - \rho A \ddot{y}' &= 0 \end{aligned} \quad (2.36)$$

Upošteevamo povezavo med lokalno koordinato  $y'$  in pomikom  $u$  ter poiščimo časovne odvode.

$$\begin{aligned} y' &= y'_o + u \\ \dot{y}' &= 0 + \dot{u} \\ \ddot{y}' &= \ddot{u} \end{aligned} \quad (2.37)$$

Tako dobimo vodilno diferencialno enačbo problema.

$$\frac{dF'_y}{dy'} - \rho A \ddot{u}(y') = 0 \quad (2.38)$$

### 2.3.1.6 Integralska variacijska formulacija

Reševanje izpeljane vodilne enačbe problema prevedemo na reševanje integralske enačbe, kot smo to storili v poglavju 2.3.1.2, vendar z drugo vodilno enačbo (2.38), ki jo pomnožimo s poljubno na območju  $y' \in [0, L]$  odvedljivo funkcijo  $v(y')$ , ter

integriramo po celotnem območju končnega elementa. Tako formuliramo integralsko enačbo.

$$\begin{aligned}
\int_0^L \left[ \frac{F'_y}{dy'} - \rho A \ddot{u}(y') \right] v(y') dy' &= 0 \\
\int_0^L \frac{dF'_y}{dy'} v(y') dy' - \int_0^L \rho A \ddot{u}(y') v(y') dy' &= 0 \\
\int_0^L v(y') dF'_y - \int_0^L \rho A \ddot{u}(y') v(y') dy' &= 0 \\
v(L) F'_y(L) - v(0) F'_y(0) - \int_0^L \rho A \ddot{u}(y') v(y') dy' &= 0
\end{aligned} \tag{2.39}$$

### 2.3.1.7 Aproksimativno numerično reševanje

Definiramo aproksimacijo iskane osnovne spremenljivke  $u(y')$ , saj nam to omogoča izvedenost integrala. V primeru dvovozliščnega končnega elementa je aproksimacija pomika  $\hat{u}(y')$ , tako kot v razdelku 2.3.1.3, zasnovana na dveh polinomih prvega reda. Konstanti  $a_0$  in  $a_1$  sta vrednosti pomika (v  $y'$ -smeri) v vozliščih končnega elementa. Tako je aproksimativna rešitev sledeča enačba.

$$\hat{u}(y') = U'_{1y} \left( 1 - \frac{y'}{L} \right) + U'_{2y} \left( \frac{y'}{L} \right) \tag{2.40}$$

Enačbo (2.39) zapišemo z aproksimacijo rešitve  $\hat{u}(y')$ .

$$\begin{aligned}
v(L) F'_y(L) - v(0) F'_y(0) - \int_0^L \rho A \ddot{\hat{u}}(y') v(y') dy' &= 0 \\
v(L) F'_y(L) - v(0) F'_y(0) - \int_0^L \rho A \left[ \ddot{U}'_{1y} \left( 1 - \frac{y'}{L} \right) + \ddot{U}'_{2y} \left( \frac{y'}{L} \right) \right] v(y') dy' &= 0
\end{aligned} \tag{2.41}$$

Kjer je sila  $F'_y$  v vozliščih končnega elementa enaka reakcijskim silam v  $y'$ -smeri.

$$\begin{aligned}
F'_y(0) &= -F'_{1y} \\
F'_y(L) &= F'_{2y}
\end{aligned}$$

V skladu z Galerkinovim pristopom izberemo poljubni funkciji.

$$\begin{aligned}
v_1(y') &= 1 - \frac{y'}{L} \\
v_2(y') &= \frac{y'}{L}
\end{aligned} \tag{2.42}$$

Prvo funkcijo  $v_1(y')$  vstavimo v enačbo (2.41) in rešimo integralsko enačbo.

$$\begin{aligned}
\left( 1 - \frac{L}{L} \right) F'_{2y} - \left( 1 - \frac{0}{L} \right) (-F'_{1y}) - \dots \\
\dots - \int_0^L \rho A \left[ \ddot{U}'_{1y} \left( 1 - \frac{y'}{L} \right) + \ddot{U}'_{2y} \left( \frac{y'}{L} \right) \right] \left( 1 - \frac{y'}{L} \right) dy' = 0
\end{aligned}$$

$$F'_{2y} - \left( \frac{1}{6} \rho A L \ddot{U}'_{1y} + \frac{1}{3} \rho A L \ddot{U}'_{2y} \right) = 0$$

$$\frac{1}{6} m \ddot{U}'_{1y} + \frac{1}{3} m \ddot{U}'_{2y} = F'_{2y} \quad (2.43)$$

Izpeljali smo enačbi KE za vplive izven aksialne  $y'$ -smeri.

$$\frac{1}{3} m \ddot{U}'_{1y} + \frac{1}{6} m \ddot{U}'_{2y} = F'_{1y} \quad (2.44)$$

$$\frac{1}{6} m \ddot{U}'_{1y} + \frac{1}{3} m \ddot{U}'_{2y} = F'_{2y}$$

Matrični zapis zgornjih enačb.

$$\frac{m}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} \ddot{U}'_{1y} \\ \ddot{U}'_{2y} \end{Bmatrix} = \begin{Bmatrix} F'_{1y} \\ F'_{2y} \end{Bmatrix} \quad (2.45)$$

Enako lahko storimo tudi za  $z'$ -smer in dobimo enačbo podobno enačbi (2.45).

$$\frac{m}{6} \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \begin{Bmatrix} \ddot{U}'_{1z} \\ \ddot{U}'_{2z} \end{Bmatrix} = \begin{Bmatrix} F'_{1z} \\ F'_{2z} \end{Bmatrix} \quad (2.46)$$

### 2.3.2 Togostna in masna matrika v lokalnem KS

V lokalnem KS smo izpeljali po šest enačb v osni smeri ( $x'$ -smeri) palice in na palico pravokotni smeri ( $y'$  ter  $z'$ -smeri). Če dobljene matrične zapise enačb (2.30), (2.45) in (2.46) združimo, dobimo enačbo KE v lokalnem koordinatnem sistemu, ki je enaka končni enačbi v [2].

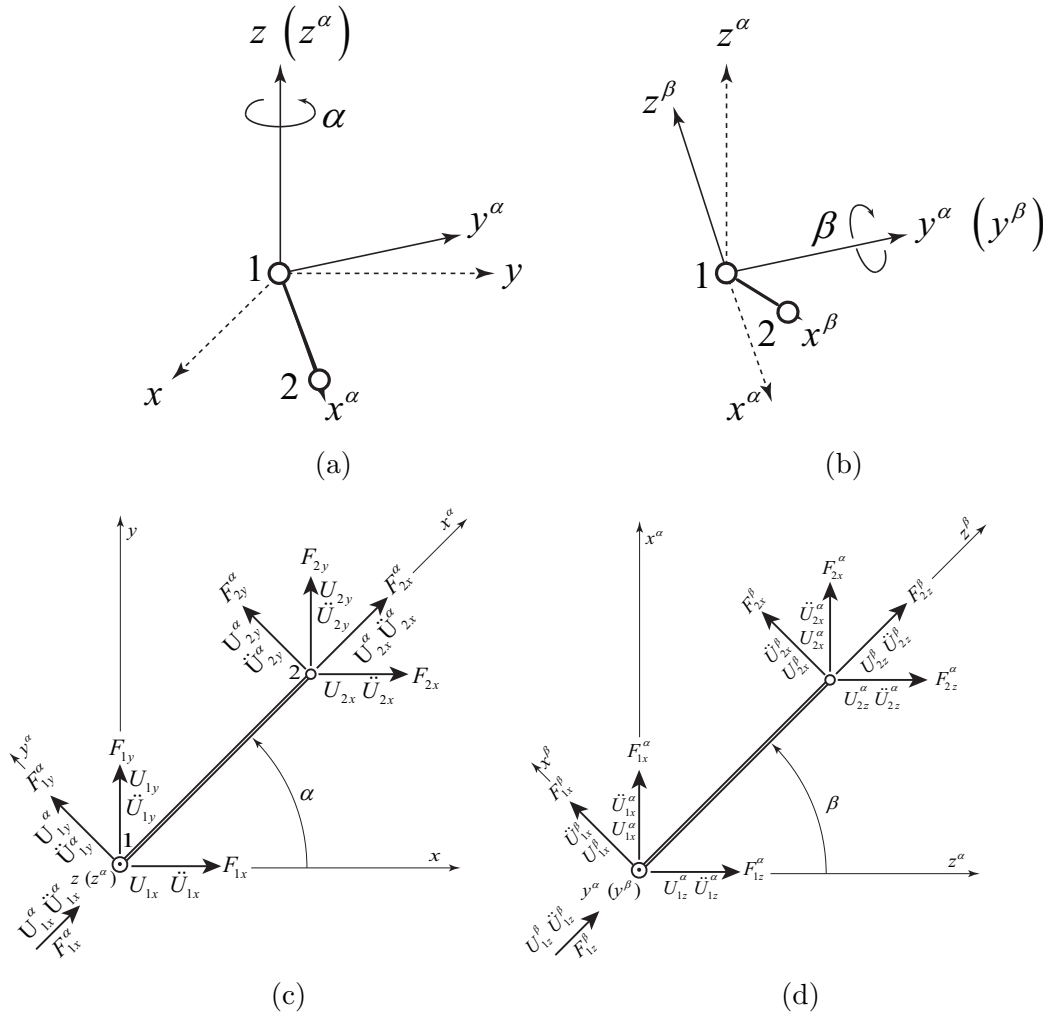
$$\frac{m}{6} \begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix} \begin{Bmatrix} \ddot{U}'_{1x} \\ \ddot{U}'_{1y} \\ \ddot{U}'_{1z} \\ \ddot{U}'_{2x} \\ \ddot{U}'_{2y} \\ \ddot{U}'_{2z} \end{Bmatrix} + \frac{EA}{L} \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{Bmatrix} U'_{1x} \\ U'_{1y} \\ U'_{1z} \\ U'_{2x} \\ U'_{2y} \\ U'_{2z} \end{Bmatrix} = \begin{Bmatrix} F'_{1x} \\ F'_{1y} \\ F'_{1z} \\ F'_{2x} \\ F'_{2y} \\ F'_{2z} \end{Bmatrix}$$

$$[M'] \{ \ddot{U}' \} + [K'] \{ U' \} = \{ F' \} \quad (2.47)$$

Enačba KE predstavlja gibalno enačbo (2.6), zapisano z diskretnimi vrednostmi pospeškov  $\{ \ddot{U}' \}$  in pomikov  $\{ U' \}$ , pri tem je  $[M']$  masna matrika in  $[K']$  togostna matrika KE v lokalnem KS.

### 2.3.3 Transformacija koordinat

Masna kot tudi togostna matrika palice sta bili izpeljani v lokalnem KS  $x'y'z'$ . Toda struktura se nahaja v globalnem KS  $xyz$ . Z ozirom na spremenljivke KE glede na globalni KS rotiramo protiurno (desnosučno pravilo) za kot  $\alpha$  ter  $\beta$  in dosežemo poljubno postavitev palice in transformacijo teh spremenljivk (slika 2.7). Teorija je povzeta po [5]. Cilj je izpeljava globalne masne matrike  $[M]$  in togostne matrike  $[K]$ .



Slika 2.7: Potrebni rotaciji (a) okoli  $z$ -osi za kot  $\alpha$  in (b) okoli  $y^\alpha$ -osi za kot  $\beta$  za transformacijo KE (c) pri rotaciji  $\alpha$  in (d) pri rotaciji  $\beta$  iz lokalnega v globalni KS.

### 2.3.3.1 Pomik v glavnem koordinatnem sistemu

Iz slike 2.7c vidimo, da so diskretne vrednosti globalnih pomikov  $\{U\}$  funkcija pomikov po prvi rotaciji  $\{U'\}$  in kota rotacije  $\alpha$ .

$$\begin{aligned} U_x &= U_x^\alpha \cos(\alpha) - U_y^\alpha \sin(\alpha) & U_y &= U_x^\alpha \sin(\alpha) + U_y^\alpha \cos(\alpha) & U_z &= U_z^\alpha \\ U_{1x} &= U_{1x}^\alpha \cos(\alpha) - U_{1y}^\alpha \sin(\alpha) & U_{1y} &= U_{1x}^\alpha \sin(\alpha) + U_{1y}^\alpha \cos(\alpha) & U_{1z} &= U_{1z}^\alpha \\ U_{2x} &= U_{2x}^\alpha \cos(\alpha) - U_{2y}^\alpha \sin(\alpha) & U_{2y} &= U_{1x}^\alpha \sin(\alpha) + U_{2y}^\alpha \cos(\alpha) & U_{2z} &= U_{2z}^\alpha \end{aligned} \quad (2.48)$$

Iz slike 2.7d razberemo diskretne vrednosti pomikov  $\{U^\alpha\}$  kot funkcijo pomikov po drugi rotaciji in kota rotacije  $\beta$ .

$$\begin{aligned} U_x^\alpha &= U_x^\beta \cos(\beta) + U_z^\beta \sin(\beta) & U_y^\alpha &= U_y^\beta & U_z^\alpha &= -U_x^\beta \sin(\beta) + U_z^\beta \cos(\beta) \\ U_{1x}^\alpha &= U_{1x}^\beta \cos(\beta) + U_{1z}^\beta \sin(\beta) & U_{1y}^\alpha &= U_{1y}^\beta & U_{1z}^\alpha &= -U_{1x}^\beta \sin(\beta) + U_{1z}^\beta \cos(\beta) \\ U_{2x}^\alpha &= U_{2x}^\beta \cos(\beta) + U_{2y}^\beta \sin(\beta) & U_{2y}^\alpha &= U_{2y}^\beta & U_{2z}^\alpha &= -U_{2x}^\beta \sin(\beta) + U_{2z}^\beta \cos(\beta) \end{aligned} \quad (2.49)$$

Iz zgornjih enačb (2.48) in (2.49) vidimo, da lahko obe transformaciji zapišemo kot eno, hkrati pa upoštevamo, da KS po drugi rotaciji sovpada z lokalnim KS KE. Sisteme enačb zapišemo matrično in definiramo matriko  $[T]$ , tako imenovano transformacijsko matriko. Velja, da je  $\sin x = sx$  ter  $\cos x = cx$ .

$$\begin{Bmatrix} U_{1x} \\ U_{1y} \\ U_{1z} \\ U_{2x} \\ U_{2y} \\ U_{2z} \end{Bmatrix} = \begin{bmatrix} c\alpha & -s\alpha & 0 & 0 & 0 & 0 \\ s\alpha & c\alpha & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha & -s\alpha & 0 \\ 0 & 0 & 0 & s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} c\beta & 0 & s\beta & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\beta & 0 & s\beta \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -s\beta & 0 & c\beta \end{bmatrix} \begin{Bmatrix} U'_{1x} \\ U'_{1y} \\ U'_{1z} \\ U'_{2x} \\ U'_{2y} \\ U'_{2z} \end{Bmatrix} \quad (2.50)$$

$$\begin{Bmatrix} U_{1x} \\ U_{1y} \\ U_{1z} \\ U_{2x} \\ U_{2y} \\ U_{2z} \end{Bmatrix} = \begin{bmatrix} c\alpha c\beta & -s\alpha & c\alpha s\beta & 0 & 0 & 0 \\ c\beta s\alpha & c\alpha & s\alpha s\beta & 0 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha c\beta & -s\alpha & c\alpha s\beta \\ 0 & 0 & 0 & c\beta s\alpha & c\alpha & s\alpha s\beta \\ 0 & 0 & 0 & -s\beta & 0 & c\beta \end{bmatrix} \begin{Bmatrix} U'_{1x} \\ U'_{1y} \\ U'_{1z} \\ U'_{2x} \\ U'_{2y} \\ U'_{2z} \end{Bmatrix} \quad (2.51)$$

$$\{U\} = [T] \{U'\} \quad (2.52)$$

Dana enačba nam v primeru znanih lokalnih pomikov končnega elementa podaja pomike v globalnem koordinatnem sistemu, kar je z vidika reševanja nekoristno. V enačbo lokalne togostne in masne matrike je treba vstaviti lokalne pomike.

$$\begin{aligned} \{U\} &= [T] \{U'\} \quad / \text{ z leve matrično množimo } [T^{-1}] \\ [T^{-1}] \{U\} &= [T^{-1}] [T] \{U'\} \\ \{U'\} &= [T^{-1}] \{U\} \end{aligned} \quad (2.53)$$

Potreben je izračun inverzne matrike  $[T^{-1}]$ . Izkaže se, da je  $[T]$  ortogonalna in pri množenju s transponirano matriko  $[T^T]$  dobimo identiteto oziroma enotsko matriko  $[I]$ .

$$[T][T^T] = [T^T][T] = [I] \quad (2.54)$$

Identiteto prav tako dobimo z množenjem matrike z njeno inverzno matriko.

$$[T][T^{-1}] = [T^{-1}][T] = [I] \quad (2.55)$$

Torej velja:

$$[T^{-1}] = [T^T] \quad (2.56)$$

in lahko enačbo (2.51) zapišemo kot:

$$\{U'\} = [T^T]\{U\} \quad (2.57)$$

### 2.3.3.2 Pospešek v glavnem koordinatnem sistemu

Diskretne vrednosti globalnih pospeškov  $\{\ddot{U}\}$  so funkcija lokalnih pospeškov  $\{\ddot{U}'\}$  in kota  $\alpha$ . Identično kot za pomik v predelu 2.3.3.1 izpeljemo za pospešek.

$$\begin{Bmatrix} \ddot{U}_{1x} \\ \ddot{U}_{1y} \\ \ddot{U}_{1z} \\ \ddot{U}_{2x} \\ \ddot{U}_{2y} \\ \ddot{U}_{2z} \end{Bmatrix} = \begin{bmatrix} c\alpha c\beta & -s\alpha & c\alpha s\beta & 0 & 0 & 0 \\ c\beta s\alpha & c\alpha & s\alpha s\beta & 0 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha c\beta & -s\alpha & c\alpha s\beta \\ 0 & 0 & 0 & c\beta s\alpha & c\alpha & s\alpha s\beta \\ 0 & 0 & 0 & -s\beta & 0 & c\beta \end{bmatrix} \begin{Bmatrix} \ddot{U}'_{1x} \\ \ddot{U}'_{1y} \\ \ddot{U}'_{1z} \\ \ddot{U}'_{2x} \\ \ddot{U}'_{2y} \\ \ddot{U}'_{2z} \end{Bmatrix}$$

$$\{\ddot{U}\} = [T]\{\ddot{U}'\}$$

$$\{\ddot{U}'\} = [T^T]\{\ddot{U}\} \quad (2.58)$$

### 2.3.3.3 Vozliščne sile v glavnem koordinatnem sistemu

Globalne vozliščne sile  $\{F\}$  so funkcija lokalnih sil  $\{F'\}$  in kota  $\alpha$ . Identično kot za pomik v podpoglavju 2.3.3.1 izpeljemo za silo.

$$\begin{Bmatrix} F_{1x} \\ F_{1y} \\ F_{1z} \\ F_{2x} \\ F_{2y} \\ F_{2z} \end{Bmatrix} = \begin{bmatrix} c\alpha c\beta & -s\alpha & c\alpha s\beta & 0 & 0 & 0 \\ c\beta s\alpha & c\alpha & s\alpha s\beta & 0 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha c\beta & -s\alpha & c\alpha s\beta \\ 0 & 0 & 0 & c\beta s\alpha & c\alpha & s\alpha s\beta \\ 0 & 0 & 0 & -s\beta & 0 & c\beta \end{bmatrix} \begin{Bmatrix} F'_{1x} \\ F'_{1y} \\ F'_{1z} \\ F'_{2x} \\ F'_{2y} \\ F'_{2z} \end{Bmatrix}$$

$$\{F\} = [T]\{F'\}$$

$$\{F'\} = [T^T]\{F\} \quad (2.59)$$



### 2.3.3.4 Globalna masna in togostna matrika

Izpeljali smo sistem enačb v matrični obliki za pomike, pospeške in sile za lokalni KS KE, kjer so vhodni podatki pomiki, pospeški in sile v globalnem KS celotne konstrukcije preko upoštevanja rotacije elementa. Vstavimo jih v gibalno enačbo (2.47).

$$\begin{aligned}
 [M']\{\ddot{U}'\} + [K']\{U'\} &= \{F'\} \\
 [M'][T^T]\{\ddot{U}\} + [K'][T^T]\{U\} &= [T^T]\{F\} \quad / \text{ z leve matrično množimo } [T] \\
 [T][M'][T^T]\{\ddot{U}\} + [T][K'][T^T]\{U\} &= \{F\} \\
 [M]\{\ddot{U}\} + [K]\{U\} &= \{F\}
 \end{aligned} \tag{2.60}$$

Masni in togostni globalni matriki v globalnem KS sta preko lokalnih matrik in transformacijske matrike definirani s sledečima matrikama.

$$\begin{aligned}
 [M] &= [T][M'][T^T] \\
 [M] &= \frac{m}{6} \begin{bmatrix} c\alpha c\beta & -s\alpha & c\alpha s\beta & 0 & 0 & 0 \\ c\beta s\alpha & c\alpha & s\alpha s\beta & 0 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha c\beta & -s\alpha & c\alpha s\beta \\ 0 & 0 & 0 & c\beta s\alpha & c\alpha & s\alpha s\beta \\ 0 & 0 & 0 & -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} c\alpha c\beta & c\beta s\alpha & -s\beta & 0 & 0 & 0 \\ -s\alpha & c\alpha & 0 & 0 & 0 & 0 \\ c\alpha s\beta & s\alpha s\beta & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha c\beta & c\beta s\alpha & -s\beta \\ 0 & 0 & 0 & -s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & c\alpha s\beta & s\alpha s\beta & c\beta \end{bmatrix} \\
 [M] &= \frac{m}{6} \begin{bmatrix} 2 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 2 & 0 \\ 0 & 0 & 1 & 0 & 0 & 2 \end{bmatrix} = [M'] \tag{2.61}
 \end{aligned}$$

$$\begin{aligned}
 [K] &= [T][K'][T^T] \\
 [K] &= \frac{EA}{L} \begin{bmatrix} c\alpha c\beta & -s\alpha & c\alpha s\beta & 0 & 0 & 0 \\ c\beta s\alpha & c\alpha & s\alpha s\beta & 0 & 0 & 0 \\ -s\beta & 0 & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha c\beta & -s\alpha & c\alpha s\beta \\ 0 & 0 & 0 & c\beta s\alpha & c\alpha & s\alpha s\beta \\ 0 & 0 & 0 & -s\beta & 0 & c\beta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} c\alpha c\beta & c\beta s\alpha & -s\beta & 0 & 0 & 0 \\ -s\alpha & c\alpha & 0 & 0 & 0 & 0 \\ c\alpha s\beta & s\alpha s\beta & c\beta & 0 & 0 & 0 \\ 0 & 0 & 0 & c\alpha c\beta & c\beta s\alpha & -s\beta \\ 0 & 0 & 0 & -s\alpha & c\alpha & 0 \\ 0 & 0 & 0 & c\alpha s\beta & s\alpha s\beta & c\beta \end{bmatrix} \\
 [K] &= \frac{EA}{L} \begin{bmatrix} c\alpha^2 c\beta^2 & s\alpha c\alpha c\beta^2 & -c\alpha s\beta c\beta & -c\alpha^2 c\beta^2 & -s\alpha c\alpha c\beta^2 & c\alpha s\beta c\beta \\ s\alpha c\alpha c\beta^2 & s\alpha^2 c\beta^2 & -s\alpha s\beta c\beta & -s\alpha c\alpha c\beta^2 & -s\alpha^2 c\beta^2 & s\alpha s\beta c\beta \\ -c\alpha s\beta c\beta & -s\alpha s\beta c\beta & s\beta^2 & c\alpha s\beta c\beta & s\alpha s\beta c\beta & -s\beta^2 \\ -c\alpha^2 c\beta^2 & -s\alpha c\alpha c\beta^2 & c\alpha s\beta c\beta & c\alpha^2 c\beta^2 & s\alpha c\alpha c\beta^2 & -c\alpha s\beta c\beta \\ -s\alpha c\alpha c\beta^2 & -s\alpha^2 c\beta^2 & s\alpha s\beta c\beta & s\alpha c\alpha c\beta^2 & s\alpha^2 c\beta^2 & -s\alpha s\beta c\beta \\ c\alpha s\beta c\beta & s\alpha s\beta c\beta & -s\beta^2 & -c\alpha s\beta c\beta & -s\alpha s\beta c\beta & s\beta^2 \end{bmatrix} \tag{2.62}
 \end{aligned}$$

Dobili smo iskani masni in togostni matriki enega samega KE, s katerima lahko preko gibalne enačbe (2.6) dinamično analiziramo v tridimenzionalnem prostoru poljubno postavljeno palico.

## 2.4 Gibalna enačba sistema

KE je za analizo definirane konstrukcije žerjava (poglavje 2.1) le osnovni sestavni del in sam po sebi ne omogoča direktne dinamske analize. Cilj metode KE je sestava osnovnih elementov v kompleksni sistem.

V istem globalnem KS lahko definiramo večje število KE, ki imajo lahko nekatera vozlišča skupna. Dobljeni masni  $[M]$  in togostni  $[K]$  matriki sta definirani glede na diskretne vrednosti pomikov vozlišč KE. Dobljene masne in togostne matrike posameznih elementov lahko z matričnim seštevanjem sestavimo v sistemski masni  $[M_{sist}]$  in togostni  $[K_{sist}]$  matriki. Mesto v sistemskih matrikah, kamor prištejemo posamezne matrike, nam definirajo pripadajoče prostostne stopnje. Dobimo eno samo sistemsko matrično gibalno enačbo.

$$[M_{sist}] \{\ddot{U}\} + [K_{sist}] \{U\} = \{F\} \quad (2.63)$$

## 2.5 Upoštevanje robnih pogojev

V dosednji izpeljavi smo predpostavili neomejeno premikanje vozlišč in posledično prosto nihanje celotne strukture. Takšen sistem imenujemo izprijen sistem (poglavje 2.6) in se izraža v obliki singularne togostne matrike  $[K]$ . Težavo togega gibanja odpravimo z vpeljavo ustreznih robnih pogojev.

Strukture so najpogostejše členkasto podprte tako, da so robna vozlišča fiksirana na mestu. Nepomična vozlišča dosežemo z eliminacijo ustrezni prostostni stopnji pripadajočih vrstic in stolpcev v sistemski masni  $[M_{sist}]$  in togostni  $[K_{sist}]$  matriki ter vektorju vozliščnih sil  $\vec{F}$ . Zapišemo lahko končno gibalno enačbo sistema z upoštevanjem robnih pogojev kot

$$\begin{matrix} [M_{sist}] & \{\ddot{U}\} & + & [K_{sist}] & \{U\} & = & \{F\} \\ N \times N & N \times 1 & & N \times N & N \times 1 & & N \times 1 \end{matrix} \quad (2.64)$$

kjer je  $N$  število neomejenih prostostnih stopenj strukture.

## 2.6 Analiza lastnih nihanj

O lastnih nihanjih lahko govorimo v primeru, ko sistemu v začetku opazovanja dodelimo neke začetne pogoje, nato pa mu prepustimo, da prosto niha. Začetni pogoji so definirani z začetnimi pomiki od ravnovesne lege, hitrostmi in pospeški koordinat vozlišč. V sistem jih lahko vnesemo na različne načine - potreben je vnos energije ali delovanje sile preko impulznega stavka. Po vnosu začetnih pogojev struktura ni pod vplivom nobene zunanje sile in pravimo, da je sistem v stanju proste vibracije [7].

Matematični model pri iskanju lastnih frekvenc (lastnih vrednosti) in pripadajočih lastnih oblik (lastnih vektorjev) predstavlja sistem  $n$  homogenih diferencialnih gibalnih enačb, kjer je  $n$  število prostostnih stopenj.

$$[M] \{\ddot{U}\} + [K] \{U\} = \{0\} \quad (2.65)$$

Za nedušene in harmonične proste vibracije iščemo rešitev kot  $n$  harmonskih funkcij.

$$U_i = X_i \sin(\omega t - \varphi), \quad ; \quad i = 1, 2, \dots, n \quad (2.66)$$

V vektorski notaciji.

$$\{U\} = \{X\} \sin(\omega t - \varphi), \quad ; \quad i = 1, 2, \dots, n \quad (2.67)$$

Kjer so iskane neznane vrednosti amplitude  $\{X\}$  in lastne frekvence  $\omega$ . Sistem ima  $n$  lastnih frekvenc ter enako število lastnih oblik. Pri nihanjih z lastnimi oblikami se vsak del sistema giblje harmonično ob hkratnem prehodu preko ravnovesnih leg. Za doseg takega gibanja moramo zagotoviti začetne pogoje, ki ustrezajo lastnim oblikam. Pri poljubnem začetnem pogoju je odziv sistema pri vsaki koordinati dan z vsoto nihanj z lastnimi oblikami.

Nastavek (2.67) uporabimo v enačbi (2.65).

$$\begin{aligned} -\omega^2 \{X\} [M] \sin(\omega t - \varphi) + [K] \{X\} \sin(\omega t - \varphi) &= \{0\} \quad / \text{ predp. } \sin(\omega t - \varphi) \neq 0 \\ -\omega^2 \{X\} [M] + [K] \{X\} &= \{0\} \\ ([K] - \omega^2 [M]) \{X\} &= \{0\} \quad / \text{ pomnožimo z } [M^{-1}] \\ ([K][M^{-1}] - \omega^2 [I]) \{X\} &= \{0\} \\ ([A] - \lambda [I]) \{X\} &= \{0\} \end{aligned} \quad (2.68)$$

Kjer je matrika  $[A] = [K][M^{-1}]$  imenovana dinamična matrika sistema.  $\lambda = \omega^2$  je parameter, ki vodi do lastnih vrednosti sistema. Netrivialna rešitev zahteva vsaj eno neničelno amplitudo v vektorju  $\{X\}$ , torej iščemo neničelnost determinante ali karakteristično enačbo sistema.

$$\det([A] - \lambda [I]) = 0 \quad (2.69)$$

Razvoj determinante pripelje do polinoma  $n$ -te stopnje:

$$\lambda^n + X_1 \lambda^{n-1} + \dots + X_n = 0, \quad (2.70)$$

iz katerega izrazimo  $n$  lastnih vrednosti sistema. Lastne vrednosti uredimo po velikosti  $\lambda_i$ ,  $i = 1, \dots, n$ , tako da najmanjše število pomeni prva lastna vrednost in hkrati prva lastna frekvenca  $f_o$ .

$$f_{oi} = \frac{\omega_{oi}}{2\pi} = \frac{\sqrt{\lambda_i}}{2\pi}; \quad i = 1, \dots, n \quad (2.71)$$

Z vstavljanjem izračunanih lastnih vrednosti  $\lambda_i$  v enačbo (2.68) izračunamo pripadajoče lastne vektorje  $X_i$ . Lastni vektorji ne predstavljajo dejanske amplitude, temveč medsebojno relativne vrednosti. Če te relativne vrednosti normiramo in formuliramo v vektor  $\{X\}$ , ta predstavlja modalno obliko pripadajoče lastne frekvence.

### 3 Metodologija raziskave

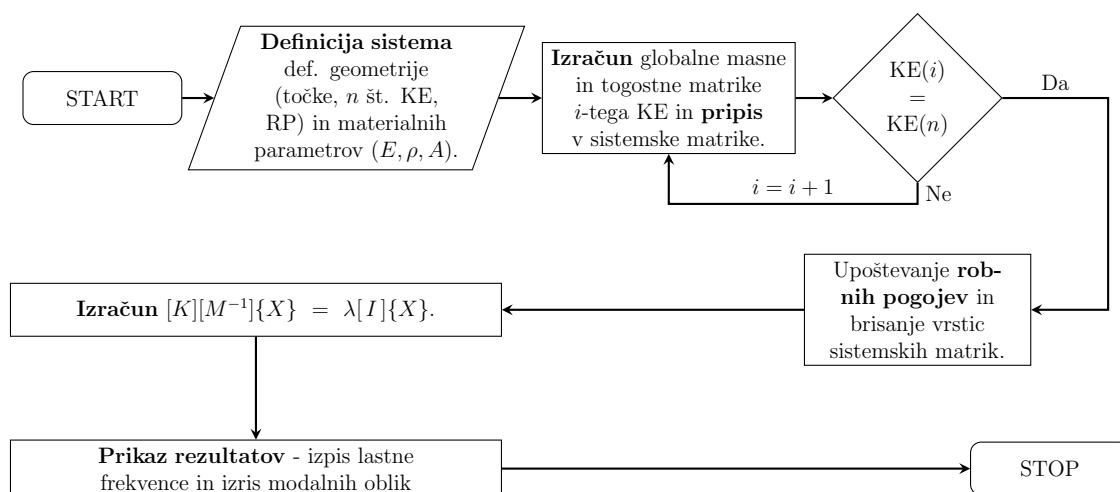
Glavni cilj sledečega poglavja je razvoj Python programa za preračun lastnih frekvenc in izris modalnih oblik poljubnega paličja, z upoštevanjem korakov v poglavju 2. Celotno znanje uporabe Pythona in njegovih paketov smo povzeli po [9].

Sledi predstavitev postopka reševanja istega problema v komercialnem MKE programu Ansys ([10], [11]), na podlagi katerega bomo verificirali pravilnost lastne kode in izračunanih rezultatov analize.

#### 3.1 Python program za modalno analizo

Python je programski jezik, ki sodi med najbolj razširjene visokonivojske programske jezike v inženirski uporabi. V njegov ekosistem sodi tudi knjižnica NumPy, ki nudi glavna orodja pri izdelavi našega programa.

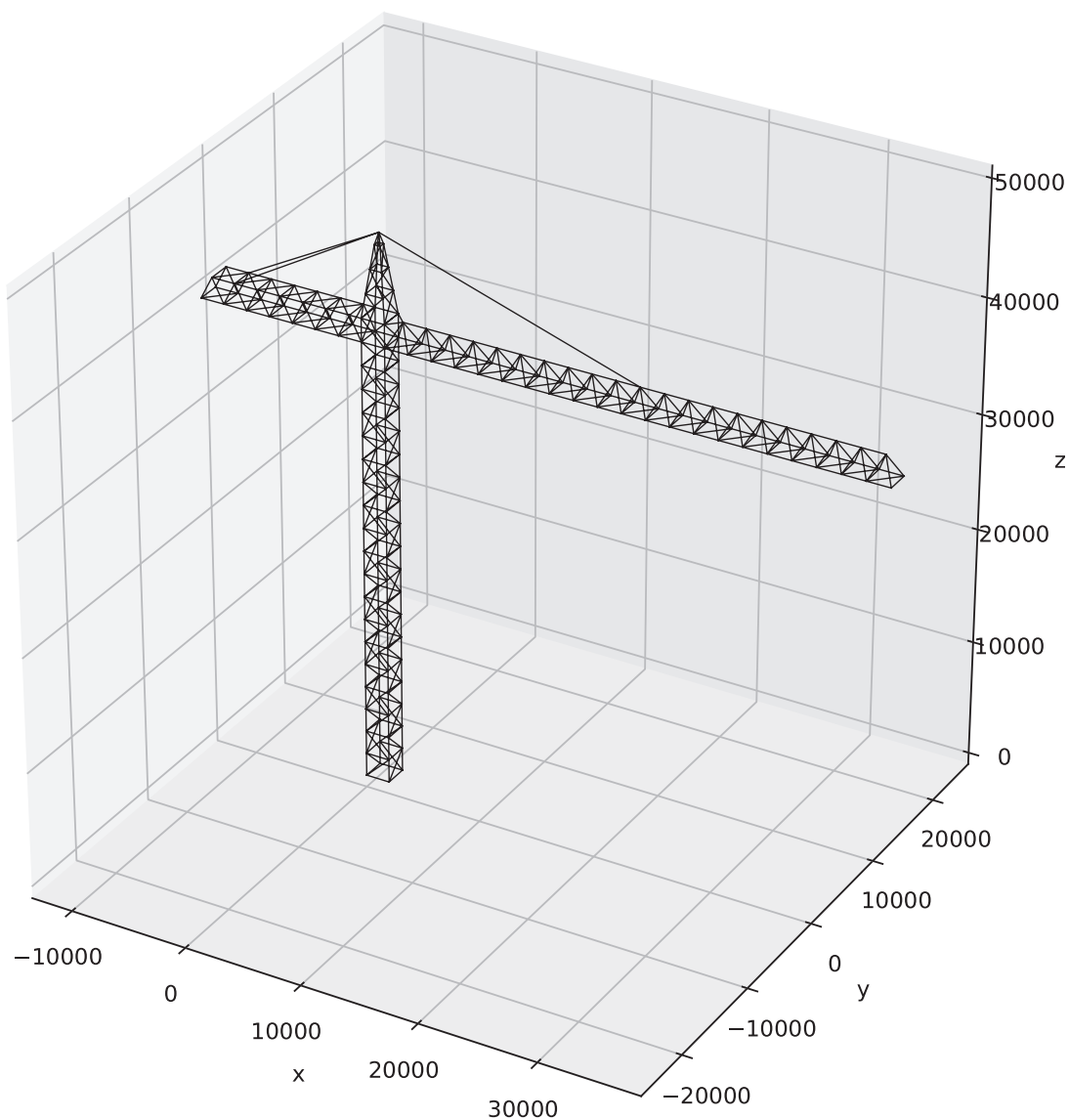
Pri snovanju naše programske kode se bomo držali sheme algoritma vidnega na sliki 3.1. Preko sheme pridemo do končnega programa, ki se nahaja v prilogi.



Slika 3.1: Blokovni diagram algoritma Python programa.

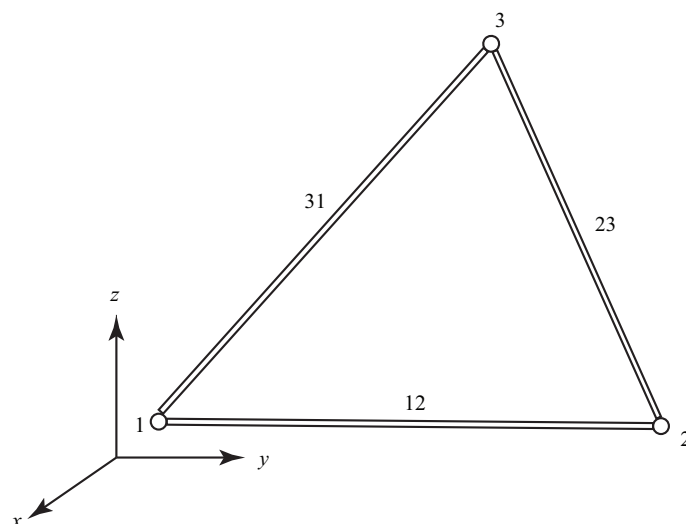
### 3.1.1 Definicija sistema

Na prvem mestu je potrebno definirati dejanski model, za katerega izvajamo modalno analizo. Kot klasičen inženirski primer definiramo jekleno konstrukcijo žerjava (slika 3.2). Zaradi velikih višin so žerjavi pogosto izpostavljeni sunkovitim vetrovom, ki lahko vzbujajo v okolici lastnih frekvenc in povzročajo nihanje z velikimi amplitudami, kar lahko privede do porušitve.



Slika 3.2: Model žerjava za katerega bomo preračunali lastne frekvence in oblike.

Naš žerjav smo diskretizirali na 188 vozlišč in 560 končnih elementov in tako začeli z reševanjem po MKE (slika 2.1). Tvorimo seznam oštevilčenih vozlišč  $V = 1, 2, 3, \dots$ , ki so definirana v globalnem kartezijevem koordinatnem sistemu s tremi prostorskimi koordinatami  $x y z$ . Nato definiramo povezave med vozlišči v drugem seznamu, kar direktno definira končne elemente  $K = 1-2, 2-3, 3-4, 4-1, \dots$  (slika 3.3). Na tej točki definiramo tudi vozlišča, ki bodo kasneje uporabljena za robne pogoje  $RP = 1, 2, 3, 4$ .



Slika 3.3: Povezovanje vozlišč v tri končne elemente.

Kot smo zapisali v poglavju 2.1, je potrebno definirati materialne lastnosti sistema ter nekatere geometrijske parametre, kar je razvidno tudi iz končne izpeljave po MKE. Potrebne materialne lastnosti so masa KE  $m$  in njegov modul elastičnosti  $E$ . Potrebno je definirati tudi geometrijsko obliko KE, ki je v osnovi palica dolžine  $L$  in s površino preseka  $A$ . Kot vidimo, lahko namesto mase preko zveze  $m = \rho AL$  definiramo gostoto materiala  $\rho$ . Dolžine palice ne pišemo eksplicitno, saj je kasneje izračunana iz pozicij vozlišč posameznega KE. Konkretni materialni in geometrijski parametri jeklenih palic so definirani v preglednici 3.1.

Preglednica 3.1: Materialni in geometrijski parametri.

	Parameter	Vrednost	Enota
1	$E$	210000	MPa
2	$\rho$	7800	kg/m <sup>3</sup>
3	$A$	100	mm <sup>2</sup>

### 3.1.2 Definicija globalnih matrik končnega elementa

Ker moramo uporabiti globalno definirana vozlišča, uporabimo izpeljavo enačb masne matrike (enačba (2.61)) in togostne matrike (enačba (2.62)) po transformaciji. Ker te matrike definiramo kot funkcije lastnosti  $E, A, L$  in transformacijskih kotov  $\alpha$  ter  $\beta$ , jih lahko kličemo z drugačnimi parametri in tako izdelamo različne KE.

Lastnosti posameznih palic smo že podali, toda na podlagi znanih podatkov moramo vstaviti vrednosti transformacijskih kotov in tako umestiti posamezni KE v globalni KS. Z upoštevanjem desnosučne transformacije (slika 2.7), kjer je pozitiven kot rotacije okoli osi proti-urno, lahko geometrijsko izpeljemo  $\alpha$  in  $\beta$  preko koordinat vozlišč vsakega končnega elementa.

Vektor  $\{V_1V_2\}$   $n$ -tega KE  $K_n$ , ki kaže od prvega vozlišča  $V_1$  do drugega  $V_2$ , je:

$$\{V_1V_2\}_n = \{V_2\}_n - \{V_1\}_n = \begin{Bmatrix} x_2 - x_1 \\ y_2 - y_1 \\ z_2 - z_1 \end{Bmatrix}_n, \text{ za vsak } n. \quad (3.1)$$

Njegova dolžina je enaka iskani dolžini KE  $L$ .

$$L = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2} \quad (3.2)$$

Preko ustreznih kotnih funkcij in komponent vektorja lahko definiramo kota

$$\alpha = \arctan\left(\frac{\{V_1V_2\}_y}{\{V_1V_2\}_x}\right) = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \text{ in} \quad (3.3)$$

$$\beta = \arcsin\left(\frac{\{V_1V_2\}_z}{|\{V_1V_2\}|}\right) = \arcsin\left(\frac{z_2 - z_1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}}\right) \quad (3.4)$$

in veljata za vsak KE.

Na tej točki smo sposobni v popolnosti zapisati  $[M]$  in  $[K]$  za vsak KE, kar pa ne moremo uporabiti za reševanje lastnih vrednosti matrike celotnega sistema. Sledi umeščanje matrik v sistemski matriki  $[M]_{sist}$  in  $[K]_{sist}$ .

### 3.1.3 Sestavljanje globalnih matrik v sistemsko matriko

Sistemski matriki sta dimenzij  $N \times N$ , kjer je  $N$  število vseh prostostnih stopenj sistema. Torej se navezujeta na vse pomike vozlišč. Razviti je potrebno algoritem, ki omogoča prištevanje matrik KE z le dvema vozlišči in tako šest prostostnimi stopnjami na ustrezna mesta sistemskih matrik. Za primer treh KE na sliki 3.3 prikažimo razvoj sistemske matrike. V preglednici 3.2 prikažemo pomike KE in pripadajoča mesta v sistemski matriki.

Preglednica 3.2: KE s pripadajočimi vozlišči in prostostnimi stopnjami.

KE	vozlišča	pomiki	mesta v sist. matriki
KE 1	1, 2	$U_{1x}, U_{1y}, U_{1z}, U_{2x}, U_{2y}, U_{2z}$	1, 2, 3, 4, 5, 6
KE 2	2, 3	$U_{2x}, U_{2y}, U_{2z}, U_{3x}, U_{3y}, U_{3z}$	4, 5, 6, 7, 8, 9
KE 3	1, 3	$U_{1x}, U_{1y}, U_{1z}, U_{3x}, U_{3y}, U_{3z}$	1, 2, 3, 7, 8, 9

Vsako matriko KE lahko razdelimo na štiri podmatrike. Vsaka od njih se preslika na ustrezno mesto sistemske matrike.

$$\begin{aligned}
[KE]^1 &= \begin{array}{c} U_{1x} \ U_{1y} \ U_{1z} \quad U_{2x} \ U_{2y} \ U_{2z} \\ \left[ \begin{array}{ccc|ccc} c_{11}^1 & c_{12}^1 & c_{13}^1 & c_{14}^1 & c_{15}^1 & c_{16}^1 \\ c_{21}^1 & c_{22}^1 & c_{23}^1 & c_{24}^1 & c_{25}^1 & c_{26}^1 \\ c_{31}^1 & c_{32}^1 & c_{33}^1 & c_{34}^1 & c_{35}^1 & c_{36}^1 \\ c_{41}^1 & c_{41}^1 & c_{41}^1 & c_{41}^1 & c_{41}^1 & c_{41}^1 \\ c_{51}^1 & c_{52}^1 & c_{53}^1 & c_{54}^1 & c_{55}^1 & c_{56}^1 \\ c_{61}^1 & c_{62}^1 & c_{63}^1 & c_{64}^1 & c_{65}^1 & c_{66}^1 \end{array} \right] \begin{array}{l} U_{1x} \\ U_{1y} \\ U_{1z} \\ U_{2x} \\ U_{2y} \\ U_{2z} \end{array} \end{array} = \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \\ \left[ \begin{array}{ccc|ccc} & & & & & \\ & A_{11} & & & A_{12} & \\ & & & & & \\ - & - & - & - & - & - \\ & A_{21} & & & A_{22} & \\ & & & & & \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \end{array} \\
[KE]^2 &= \begin{array}{c} U_{2x} \ U_{2y} \ U_{2z} \quad U_{3x} \ U_{3y} \ U_{3z} \\ \left[ \begin{array}{ccc|ccc} c_{11}^2 & c_{12}^2 & c_{13}^2 & c_{14}^2 & c_{15}^2 & c_{16}^2 \\ c_{21}^2 & c_{22}^2 & c_{23}^2 & c_{24}^2 & c_{25}^2 & c_{26}^2 \\ c_{31}^2 & c_{32}^2 & c_{33}^2 & c_{34}^2 & c_{35}^2 & c_{36}^2 \\ c_{41}^2 & c_{41}^2 & c_{41}^2 & c_{41}^2 & c_{41}^2 & c_{41}^2 \\ c_{51}^2 & c_{52}^2 & c_{53}^2 & c_{54}^2 & c_{55}^2 & c_{56}^2 \\ c_{61}^2 & c_{62}^2 & c_{63}^2 & c_{64}^2 & c_{65}^2 & c_{66}^2 \end{array} \right] \begin{array}{l} U_{2x} \\ U_{2y} \\ U_{2z} \\ U_{3x} \\ U_{3y} \\ U_{3z} \end{array} \end{array} = \begin{array}{c} 4 \ 5 \ 6 \ 7 \ 8 \ 9 \\ \left[ \begin{array}{ccc|ccc} & & & & & \\ & B_{11} & & & B_{12} & \\ & & & & & \\ - & - & - & - & - & - \\ & B_{21} & & & B_{22} & \\ & & & & & \end{array} \right] \begin{array}{l} 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \end{array} \\
[KE]^3 &= \begin{array}{c} U_{1x} \ U_{1y} \ U_{1z} \quad U_{3x} \ U_{3y} \ U_{3z} \\ \left[ \begin{array}{ccc|ccc} c_{11}^3 & c_{12}^3 & c_{13}^3 & c_{14}^3 & c_{15}^3 & c_{16}^3 \\ c_{21}^3 & c_{22}^3 & c_{23}^3 & c_{24}^3 & c_{25}^3 & c_{26}^3 \\ c_{31}^3 & c_{32}^3 & c_{33}^3 & c_{34}^3 & c_{35}^3 & c_{36}^3 \\ c_{41}^3 & c_{41}^3 & c_{41}^3 & c_{41}^3 & c_{41}^3 & c_{41}^3 \\ c_{51}^3 & c_{52}^3 & c_{53}^3 & c_{54}^3 & c_{55}^3 & c_{56}^3 \\ c_{61}^3 & c_{62}^3 & c_{63}^3 & c_{64}^3 & c_{65}^3 & c_{66}^3 \end{array} \right] \begin{array}{l} U_{1x} \\ U_{1y} \\ U_{1z} \\ U_{3x} \\ U_{3y} \\ U_{3z} \end{array} \end{array} = \begin{array}{c} 1 \ 2 \ 3 \ 7 \ 8 \ 9 \\ \left[ \begin{array}{ccc|ccc} & & & & & \\ & C_{11} & & & C_{12} & \\ & & & & & \\ - & - & - & - & - & - \\ & C_{21} & & & C_{22} & \\ & & & & & \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 3 \\ 7 \\ 8 \\ 9 \end{array} \end{array} \\
[KE]_{sist} &= \begin{array}{c} 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \\ \left[ \begin{array}{ccc|ccc|ccc} & & & & & & & & \\ & A_{11} + C_{11} & & & A_{12} & & & C_{12} & \\ & & & & & & & & \\ & & & & & & & & \\ & A_{21} & & & A_{22} + B_{11} & & & B_{12} & \\ & & & & & & & & \\ & & & & & & & & \\ & C_{21} & & & B_{21} & & & C_{12} + B_{22} & \\ & & & & & & & & \end{array} \right] \begin{array}{l} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \end{array}
\end{aligned}$$

Iz primera razvijemo splošni algoritem prištevanja podmatrik v sistemsko matriko za poljubni KE z vozliščem  $m$  in  $n$  in ga izvedemo za razvoj  $[M]_{sist}$  in  $[K]_{sist}$ . Podmatrike prištejemo v vrstice in stolpce sistemske matrike izmenično od  $3m - 2$  in do  $3m$  oziroma od  $3n - 2$  in do  $3n$ .

$$\begin{aligned}
&[KE]_{sist} \text{ (od } 3m - 2 \text{ do } 3m \text{ za vrstice; od } 3m - 2 \text{ do } 3m \text{ za stolpce) } + = [KE]_{11} \\
&[KE]_{sist} \text{ (od } 3m - 2 \text{ do } 3m \text{ za vrstice; od } 3n - 2 \text{ do } 3n \text{ za stolpce) } + = [KE]_{12} \\
&[KE]_{sist} \text{ (od } 3n - 2 \text{ do } 3n \text{ za vrstice; od } 3m - 2 \text{ do } 3m \text{ za stolpce) } + = [KE]_{21} \\
&[KE]_{sist} \text{ (od } 3n - 2 \text{ do } 3n \text{ za vrstice; od } 3n - 2 \text{ do } 3n \text{ za stolpce) } + = [KE]_{22}
\end{aligned}$$



### 3.1.4 Brisanje prostostnih stopenj zaradi robnih pogojev

Na tem mestu iz obeh sistemskih matrik  $[M]_{sist}$  in  $[K]_{sist}$  izbrišemo vrstice in stolpce, ki sovpadajo s seznamom definiranih robnih pogojev v poglavju 3.1.1. V teh točkah se struktura ne giblje, kar bomo kasneje videli v izrisanih modalnih oblikah.

Sistemske matrike, ki upoštevajo RP, poimenujemo  $[M]_{sist_{RP}}$  in  $[K]_{sist_{RP}}$ .

### 3.1.5 Izračun lastnih vrednosti in vektorjev matrike

Osnova za iskanje lastnih vrednosti in lastnih vektorjev so definirane izpeljave v predelu 2.6. Toda v sistemu z velikim številom matrik (v resnici že pri  $3 \times 3$  matriki) postane postopek analitično težko rešljiv. Lastne vrednosti in pripadajoče lastne vektorje iščemo numerično.

Uporabljena numerična metoda je implementirana na osnovi linearno algebrskih rutin v paketu *LAPACK* [12]. Sposobna je izračunati realne ali kompleksne lastne vrednosti in vektorje kvadratne matrike  $[A] = [K][M^{-1}]$  tako, da zadosti pogoju

$$[A]\{X\} = \lambda[I]\{X\}. \quad (3.5)$$

Numerična metoda hkrati vrne seznam lastnih vrednosti  $\lambda$  ter seznam desnih normiranih lastnih vektorjev  $\{X\}$ .

Sledi prikaz rezultatov, ki ga bomo izvedli v poglavju 4.

## 3.2 Ansys Mechanical APDL

Ansys nudi komercialni paket programov, ki omogočajo reševanje multifizikalnih problemov. Za izvajanje modalne analize uporabimo paket *Mechanical*, ki je namenjen analizi mehanskih struktur. Za uporabljenim paketom stoji Ansysov skriptni jezik APDL (*ang. Ansys Parametric Design Language*). Jezik nudi parameterizacijo, izvajanje logičnih funkcij in drugih kompleksnih matematičnih operacij, kar vodi v učinkoviti komercialni program za reševanje po MKE [10].

Ansys korake reševanja po MKE opredeli nekoliko drugače (vendar vseeno podobno), kot smo jih v začetku poglavja 2.

Najprej definiramo geometrijski model, uporabljen tip KE in materialne lastnosti ter nato ustvarimo iz geometrije mrežo KE. V Ansys-u se ti koraki izvedejo v predprocesorju (PREP7). Nato apliciramo obremenitve in robne pogoje sistema, navedemo tip problema in rešimo problem v rešitvenem procesorju (SOL). Ko je rešitev pripravljena, si lahko ogledamo rezultate in jih izrišemo v postprocesorju (POST1/POST26).

Konkretno za naš problem lahko prikažemo omenjene korake:

### **/PREP7**

1. definiramo točke vozlišč s  $xyz$  koordinatami in povezavami med njimi,
2. izberemo tip elementa LINK180 (3D palica),
3. definiramo materialne lastnosti  $E$ ,  $A$  in  $\rho$ ,
4. mrežimo (povezave spremenimo v KE),

### **/SOL**

5. definiramo robne pogoje in obremenitve (onemogočimo gibanje točkam),
6. tip problema nastavimo na "modal" (modalna analiza),
7. izvršimo preračun,

### **/POST1 ali /POST26**

8. izpišemo lastne frekvence in izrišemo modalne oblike.

## 4 Rezultati

V tem poglavju so predstavljeni rezultati dinamske analize po MKE.

Prikažemo izračunane **lastne frekvence** in izrisane **modalne oblike** lastnega Python programa ter jih primerjamo z rezultati analize v Ansys APDL-ju.

### 4.1 Lastne frekvence

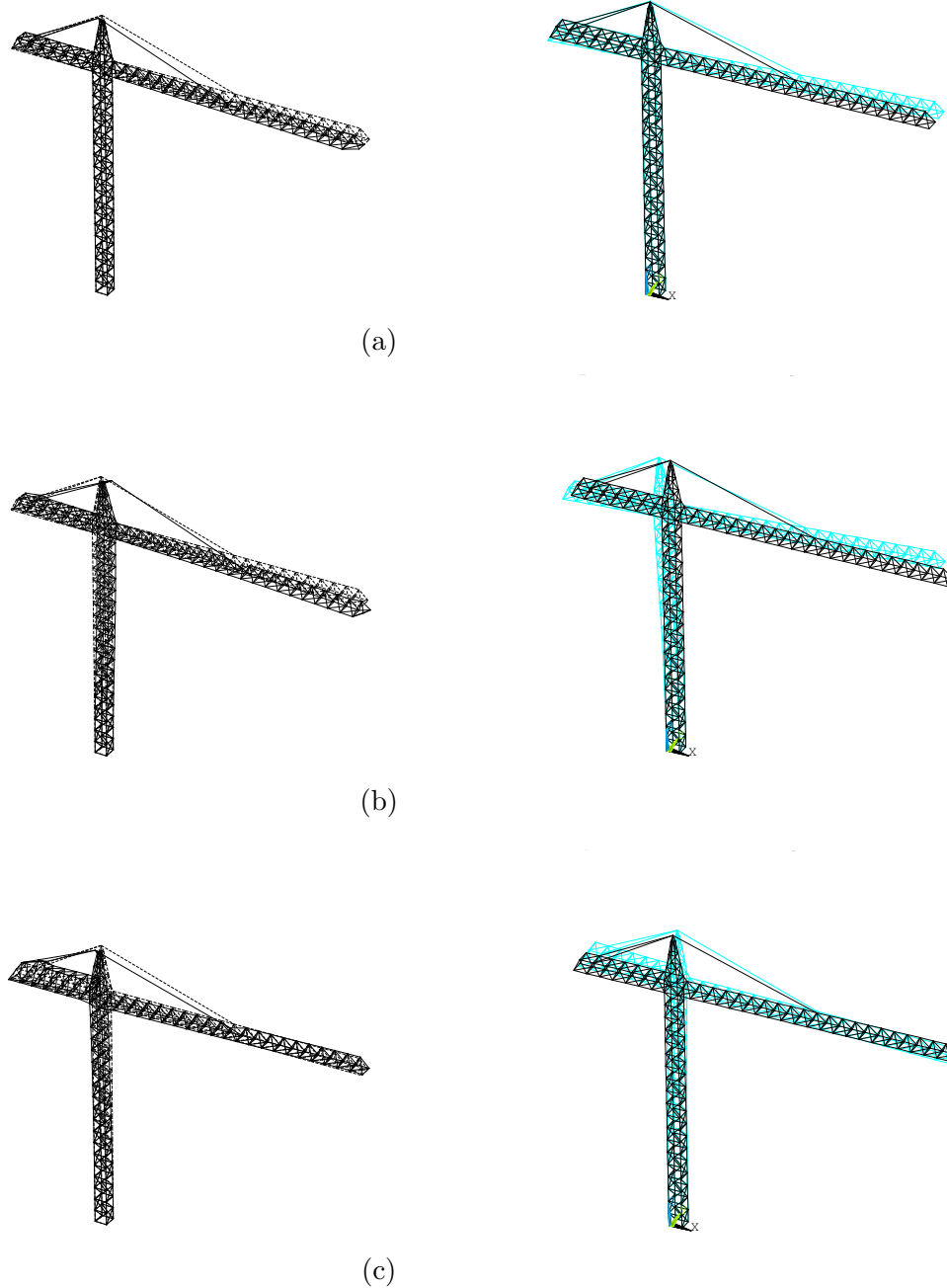
V spodnji tabeli 4.1 izpišemo nekaj frekvenc, ki smo jih izračunali v obeh programih.

Preglednica 4.1: Lastne frekvence konstrukcije žerjava.

	Python frekvence $f$ [Hz]	Ansys frekvence $f$ [Hz]
$f_1$	$4,9514 \cdot 10^{-3}$	$4,9514 \cdot 10^{-3}$
$f_2$	$9,8164 \cdot 10^{-3}$	$9,8164 \cdot 10^{-3}$
$f_3$	$1,3505 \cdot 10^{-2}$	$1,3505 \cdot 10^{-2}$
$f_4$	$2,3246 \cdot 10^{-2}$	$2,3246 \cdot 10^{-2}$
$f_5$	$3,0295 \cdot 10^{-2}$	$3,0295 \cdot 10^{-2}$
$f_{50}$	$1,6136 \cdot 10^0$	$1,6136 \cdot 10^0$
$f_{51}$	$1,6452 \cdot 10^0$	$1,6452 \cdot 10^0$
$f_{70}$	$2,5030 \cdot 10^0$	$2,5030 \cdot 10^0$
$f_{80}$	$2,9568 \cdot 10^0$	$2,9568 \cdot 10^0$
$f_{90}$	$3,3543 \cdot 10^0$	$3,3543 \cdot 10^0$
$f_{100}$	$3,7300 \cdot 10^0$	$3,7300 \cdot 10^0$
$f_{200}$	$7,4493 \cdot 10^0$	$7,4493 \cdot 10^0$
$f_{300}$	$1,2634 \cdot 10^1$	$1,2634 \cdot 10^1$
$f_{400}$	$1,5132 \cdot 10^1$	$1,5132 \cdot 10^1$
$f_{500}$	$1,9254 \cdot 10^1$	$1,9254 \cdot 10^1$
$f_{550}$	$3,8020 \cdot 10^1$	$3,8020 \cdot 10^1$
$f_{551}$	$3,8296 \cdot 10^1$	$3,8296 \cdot 10^1$
$f_{552}$	$3,8296 \cdot 10^1$	$3,8296 \cdot 10^1$

## 4.2 Lastne oblike

Na sliki 4.2 prikažimo prve tri modalne oblike obravnavane konstrukcije. Pri tem medsebojno primerjajmo izris v lastnem Python programu (levo) kot tudi izris Ansys APDL modalne analize (desno).



Slika 4.1: Primerjava prvih treh modalnih oblik kjer je: (a) prva modalna oblika pri lastni frekvenci  $f_1 = 0,005$  Hz; (b) druga modalna oblika pri lastni frekvenci  $f_2 = 0.009$  Hz; in (c) tretja modalna oblika pri tretji lastni frekvenci  $f_3 = 0.013$  Hz.

## 5 Diskusija

Na tem mestu dodatno analiziramo in komentiramo rezultate dinamske analize 3D paličja po MKE, pridobljene v poglavju 4. Sprva se osredotočimo na število in velikostni razred lastnih frekvenc. Nato analiziramo lastne oblike. Vprašamo se, zakaj smo izrisali le prve tri. Izračunamo tudi MAC vrednost lastnih vektorjev.

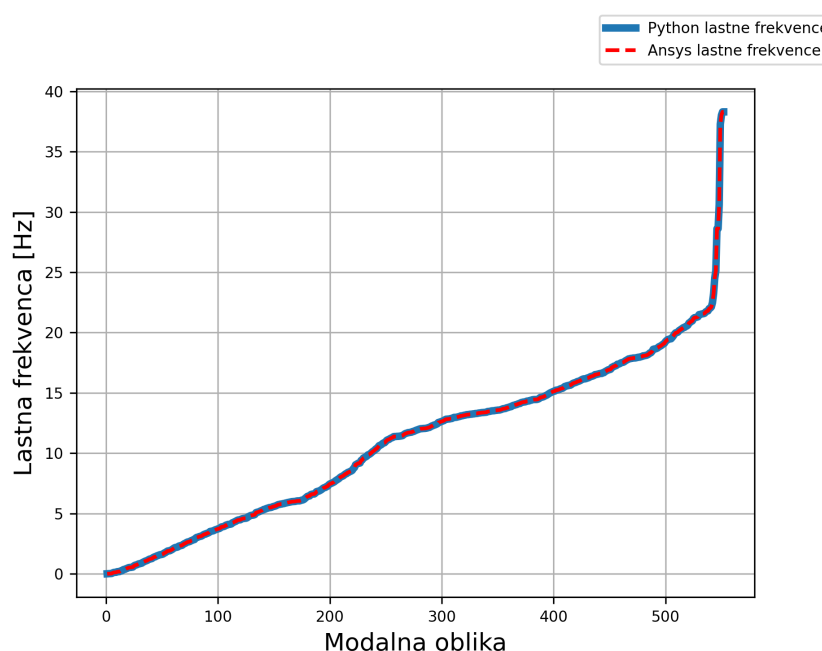
### 5.1 Število lastnih frekvenc in modalnih oblik

Število lastnih frekvenc (ter posledično tudi lastnih oblik) sovpada s številom prostostnih stopenj, ki definirajo naš sistem. Torej za diskretizirano konstrukcijo žerjava z 188 vozlišči, od katerih se lahko vsako premika v vseh treh  $x, y, z$  smereh prostora (imajo tri prostostne stopnje), velja, da imamo 564 prostostnih stopenj. Z upoštevanjem RP smo efektivno odstranili štiri vozlišča iz sistema in je potrebno odšteti 12. Ostane nam 552 prostostnih stopenj, kar je enako kot 552 izračunanih lastnih frekvenc.

Iz zgoraj navedenega lahko sklepamo, da je diskretizacija kontinuuma na KE razlog, ki vodi v končno število lastnih frekvenc. To pomeni, da ima vsak realni sistem v resnici neskončno število lastnih vrednosti in vektorjev. Nivo delitve nam lahko za isti realni objekt ne spremeni le števila, temveč tudi vrednosti rezultatov dinamske analize. V našem primeru analize paličnih konstrukcij je seveda izbira števila KE nedvoumno povezana s številom dejanskih palic v konstrukciji. Toda pri reševanju fizikalnih problemov na materialnem nivoju postane ustrezna izbira oblike in števila KE pomemben dejavnik.

## 5.2 Vrednosti lastnih frekvenc in modalnih oblik

Žerjav je relativno gledano ena večjih človekovih konstrukcij, kar se odraža tudi v velikostnem razredu vrednosti lastnih frekvenc. Na sliki 5.1 vidimo grafičen prikaz lastnih frekvenc. Njihov razpon je od 0,005 Hz (le 18 period nihanja na uro) do 38 Hz, kar je za lastne frekvence relativno malo. Toda v tem območju strukture doživijo dinamski odziv na vetrove ter potrese. Izkaže se tudi, da imajo največji vpliv začetne lastne frekvence. Razlog je v amplitudi odziva, ki nam je sicer zaradi narave naše analize nedostopna.



Slika 5.1: Lastne frekvence  $f_o$  v odvisnosti od modalne oblike.

Deformacije so pri začetnih lastnih frekvencah večje in so glavni povzročitelj porušitev struktur. Torej v realnih strukturah zelo visoke lastne frekvence pomenijo skoraj neobstoječe amplitude. To nam je pri modeliranju po MKE v pomoč, saj "neskončnega" števila modalnih oblik ne potrebujemo.

V našem primeru smo torej prikazali le prve tri modalne oblike, ki so za razumevanje nihanja žerjava najbolj pomembne. Prva modalna oblika (slika 4.1a) prikazuje nihanje žerjava okoli  $z$ -osi. Druga modalna oblika (slika 4.1b) je nihanje žerjava naprej in nazaj, kjer je največja deformacija v njegovem glavnem stebru. Zadnja prikazana modalna oblika (slika 4.1c) prikazuje vertikalno nihanje žerjava v njegovem zadnjem delu.

Na podlagi primerjave izrisa modalnih oblik težko ocenimo njihovo kakovost. Sicer v naših primerih lahko vidimo določene nekonsistentnosti v izrisih, toda primerjava vseh modalnih oblik je zamudna in premalo objektivna. Tudi direktna primerjava lastnih vektorjev, pridobljenih iz obeh metod, je zaradi velikega števila vektorjev nepraktična. Potrebno je definirati nov kriterij ocene modalnih oblik.

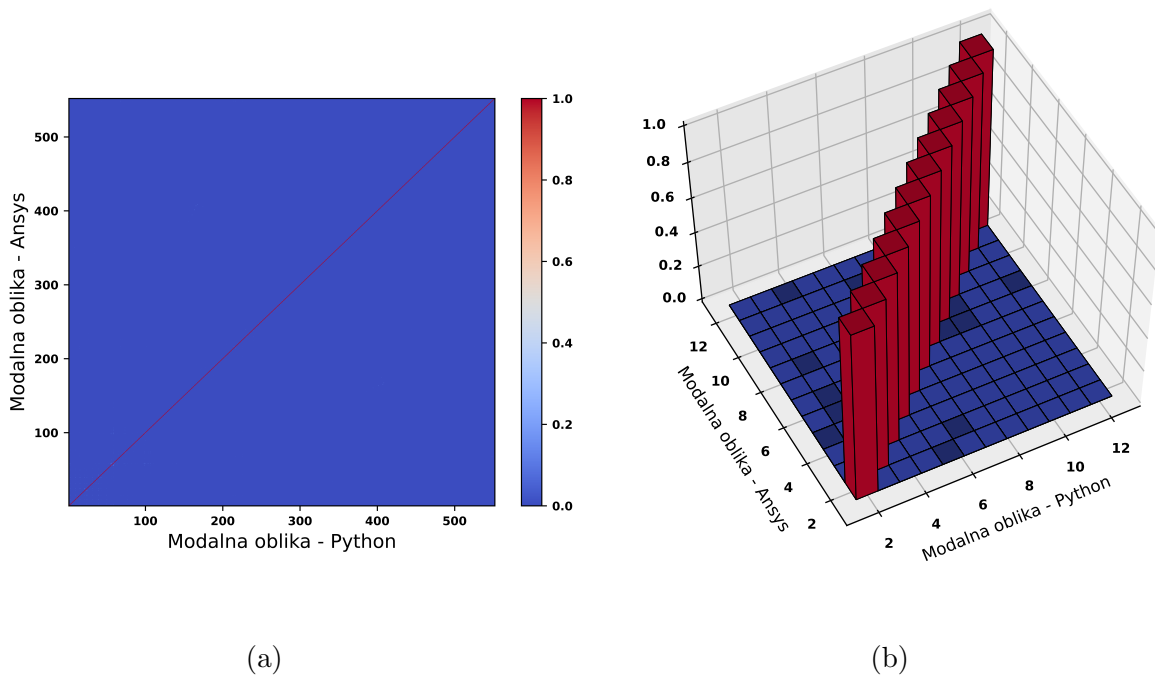
### 5.3 Kriterij gotovosti modalnih oblik

Kot konsistenten in objektiven način ocene našega izrisa vpeljemo kriterij gotovosti modalne oblike (*ang. MAC - Modal Assurance Criterion*) [13]. MAC je definiran kot normirana skalarna konstanta, ki enolično popiše konsistentnost (linearnost) med modalnim vektorjem in referenčnim modalnim vektorjem kot

$$MAC_{cdr} = \frac{|\{X_{dr}\}^H \{X_{cr}\}|^2}{\{X_{dr}\}^H \{X_{dr}\} \{X_{cr}\}^H \{X_{cr}\}}, \quad (5.1)$$

kjer je  $\{X_{dr}\}$  modalni vektor reference  $d$  in modalne oblike  $r$ ,  $\{X_{cr}\}$  modalni vektor reference  $c$  in modalne oblike  $r$  in  $\{X\}^H$  kompleksno konjugirano transponiran vektor (tudi hermitsko transponiran in hermitsko konjugiran vektor).

Izračunane MAC vrednosti po enačbi (5.1) prikažemo kot korelacijski graf (slika 5.2) med modalnimi vektorji, pridobljenimi z lastno Python kodo, in Ansys Mechanical APDL simulacijo.



Slika 5.2: Grafični prikaz ujemanja lastnih vektorjev predstavljen kot (a) 2-D MAC vrednosti in (b) 3-D MAC vrednosti (prvih 12 modalnih oblik).

Iz zgornje slike je razvidno, da je pri isti modalni obliki  $r$  medsebojno povsem linearnih vseh 552 lastnih vektorjev Python in Ansys dinamske analize, saj je  $MAC = 1$ . To nam pove, da so lastni vektorji lastnega Python programa identični tistim, pridobljenim z Ansys APDL. Opazimo tudi, da so v primeru, ko modalne oblike  $r$  vektorja referenc  $d$  in  $c$  ne sovpadajo, MAC vrednosti blizu nič. Tako lahko sklepamo, da so navidezne razlike v modalnih oblikah na sliki 4.2 posledica grafične napake.

V tem poglavju smo tako pokazali, da so rešitve lastnega Python programa za dinamsko analizo 3D paličja po MKE povsem enake kot rezultati pridobljeni v Ansys Mechanical APDL.

## 6 Zaključki

V zaključni nalogi smo obravnavali dinamsko analizo trirazsežnih vitkih paličnih struktur z metodo končnih elementov MKE. Korake in ugotovitve naloge lahko strnemo v naslednje točke:

1. Realno strukturo žerjava smo diskretizirali na model iz KE.
2. Izpeljali smo togostno in masno matriko KE 3D palice.
3. Teoretične izpeljave smo zapisali v numerični programski jezik Python in nato izvedli dinamsko analizo ter izračunali lastne frekvence in izrisali lastne oblike žerjava.
4. Pridobljene rezultate smo primerjali s komercialnim programom za reševanje po MKE Ansys Mechanical APDL.
5. Pokazali smo ujemanje v številu in vrednostih lastnih frekvenc, pridobljenih z Python in Ansys programoma.
6. Preko MAC kriterija smo dokazali ujemanje lastnih vektorjev, pridobljenih z Python in Ansys programoma.

Na primeru kompleksne realne strukture smo dokazali, da lahko z uporabo odprtokodnih programskih okolij sami izdelamo orodje za dinamsko analizo po metodi končnih elementov, ki je primerljivo s komercialnimi programi.

### **Predlogi za nadaljnje delo**

V tem delu smo le primerjali dva programa, ki temeljita na metodi končnih elementov. Kljub dobremu ujemanju rezultatov to ni dokaz o točnosti same metode v primerjavi z realnim svetom. V prihodnje bi bilo treba eksperimentalno analizirati dejanske strukture in primerjati pridobljene lastne frekvence in oblike z našim teoretičnim modelom.

Ker popolne palične strukture, ki prenašajo le osno silo, ne obstajajo, bi bilo v naš model smiselno vpeljati tudi dodatne lastnosti, ki omogočajo prenašanje upogibnih in torzijskih obremenitev. Prav tako bi mogli upoštevati dušenje. Tako bi dobili bolj realen primer palice oziroma nosilca in s tem boljše rezultate.



# Literatura

- [1] D. Logan: *A first course in the finite element method*. Cengage Learning, Boston, MA, 2017.
- [2] S. S. Rao: *The finite element method in engineering*. Butterworth-Heinemann, Oxford Elsevier, 2010.
- [3] M. Paz: *Dynamic Analysis of Trusses*. V: Chapman, Hall (ur.): *Structural Dynamics: Theory and Computation*. Springer US, Louisville KY, 1997, str. 512–537.
- [4] N. Mole: *Predavanje Metode numeričnega modeliranja - Integralska variacijska metoda in MKE*. Dostopno na: <http://lab.fs.uni-lj.si/lnms/LNMS-slo/mnm.php>, Ogled: 23. 7. 2020.
- [5] R. Deakin: *3-D Coordinate Transformations*. Surveying and Land Information Systems **58**(1999) str. 223–234.
- [6] S. Rao: *Mechanical vibrations*. Pearson Education, Inc, Hoboken, NJ, 2017.
- [7] M. Boltežar: *Nihanja sistemov z več prostostnimi stopnjami*. V: A. K. in B. Pertot (ur.): *Mehanska nihanja 1. del*. Fakulteta za strojništvo, Ljubljana, 2010, str. 165–177.
- [8] B. Kraut: *Krautov strojniški priročnik*. Littera picta, Ljubljana, 2011.
- [9] J. Slavič: *Programiranje in numerične metode v ekosistemu Pythona*. Dostopno na: <https://jankoslavice.github.io/pypinm.io/>, Ogled: 23. 7. 2020.
- [10] M. Thompson: *ANSYS Mechanical APDL for Finite Element Analysis*. Elsevier Science, San Diego, 2017.
- [11] J. Lilley: *UofA ANSYS Tutorial*. Dostopno na: <https://sites.ualberta.ca/~wmo-ussa/AnsysTutorial/index.html>, Ogled: 23. 7. 2020.
- [12] E. Anderson et al: *LAPACK - Linear Algebra PACKage*. Dostopno na: <http://www.netlib.org/lapack/>, Ogled: 08. 08. 2020.
- [13] R. Allemang: *The modal assurance criterion - Twenty years of use and abuse*. Sound & vibration **37**(2003) str. 14–23.

## Priloga A - Python programska koda

# Modalna analiza 3D vitkih paličnih struktur z MKE

Programska koda za izračun **lastnih frekvenc** in pripadajočih **modalnih oblik** po **metodi končnih elementov (MKE)**.

```
In [1]: # Programski paketi #
import numpy as np
import matplotlib as mpl
import matplotlib.pyplot as plt
import matplotlib.cm as cm
from mpl_toolkits.mplot3d import Axes3D
from matplotlib.colors import LogNorm
from ipywidgets import interact
```

```
In [2]: # Globalne nastavitve #
%matplotlib inline
plt.rcParams["figure.figsize"] = (12,12)          # Definiramo velikost izrisa grafov
plt.rcParams['figure.dpi'] = 100                 # 200 je standard
np.set_printoptions(precision=2, suppress=True) # Definiramo način izpisa matrik
```

## Postavitev fizikalnega modela

Izdelava **fizikalnega modela žerjava**. Preko definiranih **vozlišč** konstruiramo posamezne **končne elemente**.

Definiramo tudi **robne pogoje** (fiksirana vozlišča).

```
In [45]: # VOZLIŠČA - x in y koordinate
Voz = np.loadtxt(fname='tocke.txt', dtype='float', delimiter='\t', skiprows=1, usecols=(1,2,3))
# KONČNI ELEMENTI - Vozlišča oštevilčimo (od 1 naprej) in jih sestavimo v KE
KE = np.loadtxt(fname='koncni_elementi.txt', dtype='int', delimiter='\t', skiprows=1, usecols=(1,2))
# ROBNI POGOJI glede na mesta v globalni matriki
RP = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11])
```

```
In [31]: # Podatki                                # konstrukcijsko jeklo
Ej = 210000 #MPa;                                # modul elastičnosti
ro_j = 7.8E-6 #kg/mm^3                            # gostota
A1 = 100 #mm^2                                    # površina prereza
```

```

In [32]: def KE_izris(x,y,z,t1,t2):
          '''Funkcija za izrisovanje skice končnih elementov na podlagi dveh točk t1 in t2.

          Parametri:
          x - koordinata x
          y - koordinata y
          z - koordinata z
          t1 - prva točka v končnem elementu
          t2 - druga točka v končnem elementu
          '''

          ax = fig.gca(projection='3d')
          x1, x2 = x[t1], x[t2]
          y1, y2 = y[t1], y[t2]
          z1, z2 = z[t1], z[t2]
          #ax.plot([x1,x2],[y1,y2],[z1,z2], 'r.', linewidth=0.5)
          ax.plot([x1,x2],[y1,y2],[z1,z2], 'k-', linewidth=0.6)

def KE_skica(x,y,z,t1,t2):
          '''Funkcija za izrisovanje končnih elementov na podlagi dveh točk t1 in t2.

          Parametri:
          x - koordinata x
          y - koordinata y
          z - koordinata z
          t1 - prva točka v končnem elementu
          t2 - druga točka v končnem elementu
          '''

          ax = fig.gca(projection='3d')
          x1, x2 = x[t1], x[t2]
          y1, y2 = y[t1], y[t2]
          z1, z2 = z[t1], z[t2]
          ax.plot([x1,x2],[y1,y2],[z1,z2], 'k--', alpha=0.2, linewidth=0.6)

```

```

In [33]: # Definiramo graf
fig = plt.figure()
ax = plt.axes(projection='3d')
ax.set_xlabel('x'); ax.set_ylabel('y'); ax.set_zlabel('z') # oznaka osi

for i in KE: # Sprehodimo se skozi vse končne elemente in vsakega izrišemo.
    KE_izris(Voz[:,0], Voz[:,1], Voz[:,2], i[0]-1, i[1]-1);

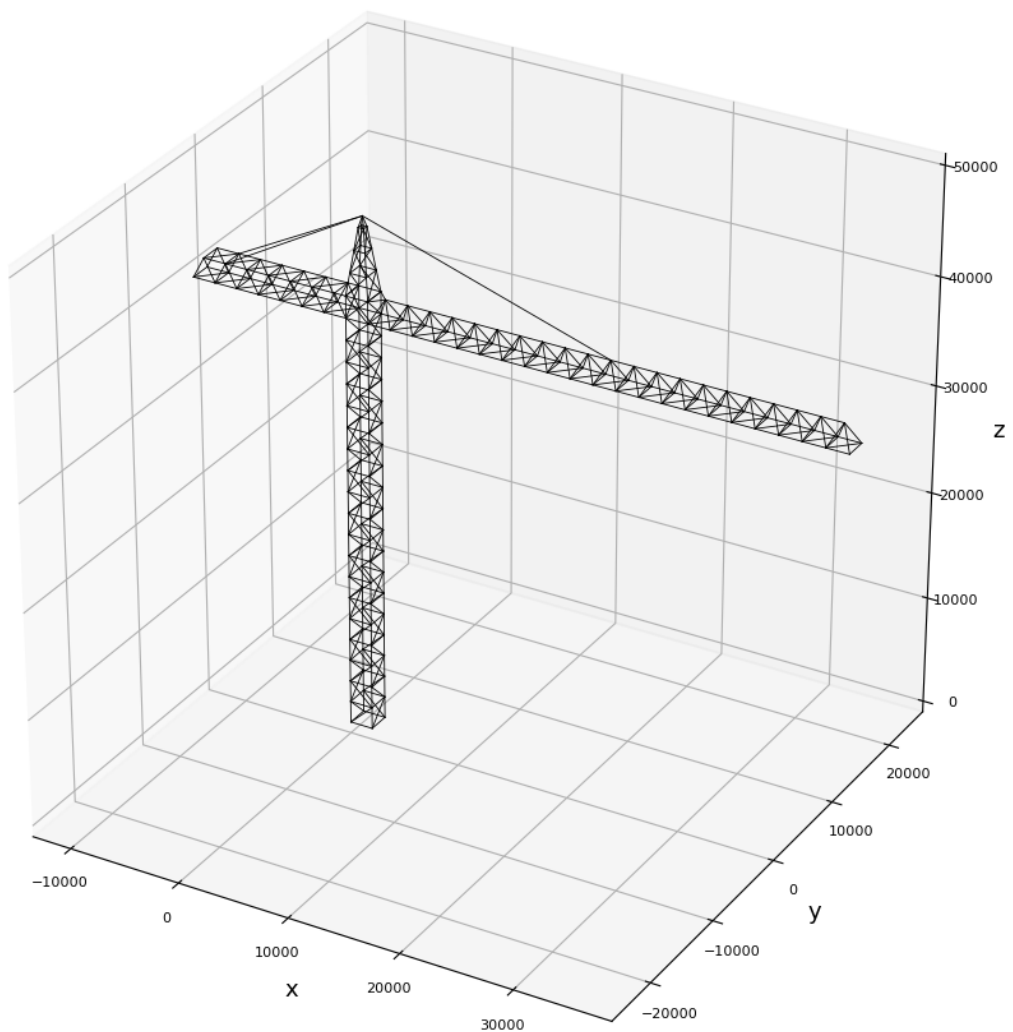
xmin, xmax = ax.get_xlim() # Določimo limite grafa
ymin, ymax = ax.get_ylim()
zmin, zmax = ax.get_zlim()

raz = 50000
ax.set_xlim(-1*raz/4, 3*raz/4); ax.set_ylim(-raz/2, raz/2); ax.set_zlim(0, raz);

plt.axis('on')
plt.savefig('model.png')

###

```



## Globalne matrike končnega elementa

Ker je potrebno uporabiti globalno definirana vozlišča, uporabimo izpeljavo masne in togostne matrike po transformaciji.

Ker te matrike definiramo splošno, jih lahko kličemo z drugačnimi parametri in tako izdelamo različne končne elemente.

### Globalna masna matrike posameznega končnega elementa

```
In [12]: ## Masna matrika ##
def M_ke(ro, A, L):
    '''Masna matrika v globalnem koordinatnem sistemu.

    Parametri:
    ro      - gostota materiala [kg/m^3]
    A       - površina prereza elementa [mm^2]
    L       - dolžina elementa [mm]
    ...

    M = ro*A*L/6*np.array([
        [2, 0, 0, 1, 0, 0],
        [0, 2, 0, 0, 1, 0],
        [0, 0, 2, 0, 0, 1],
        [1, 0, 0, 2, 0, 0],
        [0, 1, 0, 0, 2, 0],
        [0, 0, 1, 0, 0, 2]])
    return M
```

### Globalna togostna matrika posameznega končnega elementa

```
In [13]: ## Togostna matrika ##
def K_ke(alpha, beta, E, A, L):
    '''Togostna matrika v globalnem koordinatnem sistemu.

    Parametri:
    alpha - kot relativno na x os, kjer je z-os pri rotaciji vzporedna [rad]
    beta  - kot relativno na x os, kjer je y-os pri rotaciji vzporedna [rad]
    E     - modul elastičnosti [MPa]
    A     - površina prereza elementa [mm^2]
    L     - dolžina elementa [mm]
    ...


    ca = np.cos(alpha)
    sa = np.sin(alpha)
    cb = np.cos(beta)
    sb = np.sin(beta)
    K = E*A/L*np.array([
        [ca**2*cb**2, +ca*cb**2*sa, -ca*cb*sb, -ca**2*cb**2, -ca*cb**2*sa, +ca*cb*sb],
        [+ca*cb**2*sa, +cb**2*sa**2, -cb*sa*sb, -ca*cb**2*sa, -cb**2*sa**2, +cb*sa*sb],
        [-ca*cb*sb, -cb*sa*sb, +sb**2, +ca*cb*sb, +cb*sa*sb, -sb**2],
        [-ca**2*cb**2, -ca*cb**2*sa, +ca*cb*sb, +ca**2*cb**2, +ca*cb**2*sa, -ca*cb*sb],
        [-ca*cb**2*sa, -cb**2*sa**2, +cb*sa*sb, +ca*cb**2*sa, +cb**2*sa**2, -cb*sa*sb],
        [+ca*cb*sb, +cb*sa*sb, -sb**2, -ca*cb*sb, -cb*sa*sb, +sb**2]])
    return K

###
```

## Globalna masna in togostna matrika sistema

Iz posameznega končnega elementa sestavimo sistemski masni in togostni matriki.

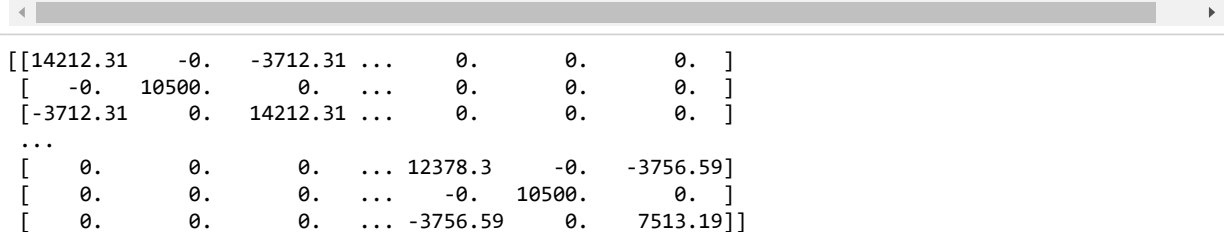
```
In [14]: M_glob_sist = np.zeros((Voz.size, Voz.size)) # definiramo prazno globalno masna matrika sistema gled
e na št. vozlišč
for ke in KE: # za vsak končni element v seznamu
    KE
    Voz1, Voz2 = Voz[ke[0]-1], Voz[ke[1]-1] # Python štetje prične z 0.
    ke_vek = Voz2 - Voz1 # vektor definiran na podlagi vozli
    šč k.e.
    ke_vek_L = np.linalg.norm(ke_vek) # velikost vektorjev oz. dolžina pa
    lic
    KE_glob = M_ke(ro_j, A1, ke_vek_L) # kličemo funkcijo masne matrike
    ## Pripis matrike k.e. v sistemsko matriko ##
    u11, u13 = 3*ke[0]-3, 3*ke[0]-1 # Definiramo neznane pomike k.e.
    u21, u23 = 3*ke[1]-3, 3*ke[1]-1
    M_glob_sist[u11:u13+1,u11:u13+1] += KE_glob[0:3,0:3]
    M_glob_sist[u11:u13+1,u21:u23+1] += KE_glob[0:3,3:6]
    M_glob_sist[u21:u23+1,u21:u23+1] += KE_glob[3:6,3:6]
    M_glob_sist[u21:u23+1,u11:u13+1] += KE_glob[3:6,0:3]
print(M_glob_sist)
```



```
[ 2.3  0.  0.  ... 0.  0.  0. ]
[ 0.  2.3 0.  ... 0.  0.  0. ]
[ 0.  0.  2.3 ... 0.  0.  0. ]
...
[ 0.  0.  0.  ... 1.62 0.  0. ]
[ 0.  0.  0.  ... 0.  1.62 0. ]
[ 0.  0.  0.  ... 0.  0.  1.62]]
```

## Skupna globalna togostna matrika

```
In [15]: K_glob_sist = np.zeros((Voz.size, Voz.size)) # definiramo prazno globalno togostno matrika sistema g
lede na št. vozlišč
for ke in KE: # za vsak končni element v seznamu
    KE
    Voz1, Voz2 = Voz[ke[0]-1], Voz[ke[1]-1] # Python štetje prične z 0.
    ke_vek = Voz2 - Voz1 # vektor definiran na podlagi vozli
    šč k.e
    ke_vek_L = np.linalg.norm(ke_vek) # velikost vektorjev oz. dolžina pa
    lic
    ke_kot_a = np.arctan2(ke_vek[1],ke_vek[0]) # kot alfa; kot med x-osjo in vektor
    jem k.e.
    ke_kot_b = np.arcsin(ke_vek[2]/ke_vek_L) # kot beta; kot med x-osjo in vektor
    jem k.e.
    KE_glob = K_ke(ke_kot_a, ke_kot_b, Ej, A1, ke_vek_L) # kličemo funkcijo togostne matrike
    ## Pripis matrike KE v sistemsko matriko ##
    u11, u13 = 3*ke[0]-3, 3*ke[0]-1 # Definiramo neznane pomike k.e.
    u21, u23 = 3*ke[1]-3, 3*ke[1]-1
    K_glob_sist[u11:u13+1,u11:u13+1] += KE_glob[0:3,0:3]
    K_glob_sist[u11:u13+1,u21:u23+1] += KE_glob[0:3,3:6]
    K_glob_sist[u21:u23+1,u21:u23+1] += KE_glob[3:6,3:6]
    K_glob_sist[u21:u23+1,u11:u13+1] += KE_glob[3:6,0:3]
print(K_glob_sist)
```



```
[ 14212.31  -0. -3712.31 ...  0.  0.  0. ]
[  -0.  10500.  0.  ...  0.  0.  0. ]
[ -3712.31  0.  14212.31 ...  0.  0.  0. ]
...
[  0.  0.  0.  ... 12378.3 -0. -3756.59]
[  0.  0.  0.  ... -0.  10500.  0. ]
[  0.  0.  0.  ... -3756.59  0.  7513.19]]
```

## Robni pogoji

Na tem mestu iz obeh sistemskih matrik izbrišemo vrste in stolpce, ki sovpadajo z robnimi pogoji.

Razlog je v tem, da robni pogoj predstavlja pomik, ki je "zaklenjena" prostostna stopnja.

```
In [16]: # Sistemska masna matrika #
M_glob_sist_RP = np.delete(M_glob_sist, RP, axis=0)
M_glob_sist_RP = np.delete(M_glob_sist_RP, RP, axis=1)

# Sistemska togostna matrika #
K_glob_sist_RP = np.delete(K_glob_sist, RP, axis=0)
K_glob_sist_RP = np.delete(K_glob_sist_RP, RP, axis=1)

print(M_glob_sist_RP)
print()
print(K_glob_sist_RP)

[[ 3.55  0.    0.    ...  0.    0.    0.   ]
 [ 0.    3.55  0.    ...  0.    0.    0.   ]
 [ 0.    0.    3.55  ...  0.    0.    0.   ]
 ...
 [ 0.    0.    0.    ...  1.62  0.    0.   ]
 [ 0.    0.    0.    ...  0.    1.62  0.   ]
 [ 0.    0.    0.    ...  0.    0.    1.62]]

[[ 10500.    -0.    -0.    ...    0.    0.    0.   ]
 [   -0.   17924.62    0.    ...    0.    0.    0.   ]
 [   -0.    0.   28424.62  ...    0.    0.    0.   ]
 ...
 [    0.    0.    0.    ... 12378.3   -0.  -3756.59]
 [    0.    0.    0.    ...   -0.  10500.    0.   ]
 [    0.    0.    0.    ... -3756.59    0.   7513.19]]
```

## Iskanje lastnih frekvenc in vektorjev

Osnova za iskanje lastnih vrednosti in lastnih vektorjev so definirane izpeljave. Toda v sistemu z velikim številom matrik (v resnici že pri  $3 \times 3$  matriki) postane postopek analitično težko rešljiv.

V *Python* paketu *numpy.linalg* imamo funkcijo *eig*, ki numerično išče tako lastne vrednosti kot pripadajočo matriko lastnih vektorjev.

```
In [17]: lastnaVrednost, lastniVektor = np.linalg.eig(K_glob_sist_RP @ np.linalg.inv(M_glob_sist_RP))

idx = lastnaVrednost.argsort()          # Lastne vrednosti uredimo po velikosti
lastnaVrednost = lastnaVrednost[idx]
lastniVektor = lastniVektor[:,idx]
lastnaFrekvenca = (np.sqrt(lastnaVrednost)/(2*np.pi)) # Lastna frekvenca iz Lastne vrednosti
```



```
In [18]: print("število lastnih frekvenc in oblik je", np.size(lastnaFrekvenca))
print()
print("lastne frekvence: ")
print(lastnaFrekvenca)
print()
print("lastni vektorji:")
print(lastniVektor)
```

število lastnih frekvenc in oblik je 552

lastne frekvence:

```
[ 0.    0.01  0.01  0.02  0.03  0.08  0.12  0.12  0.13  0.18  0.18  0.19
 0.24  0.25  0.33  0.37  0.38  0.43  0.46  0.51  0.53  0.53  0.55  0.56
 0.67  0.72  0.74  0.77  0.82  0.84  0.87  0.89  0.96  1.    1.05  1.08
 1.12  1.18  1.22  1.24  1.27  1.35  1.4   1.41  1.47  1.51  1.54  1.58
 1.59  1.61  1.65  1.73  1.75  1.83  1.89  1.91  1.92  1.96  2.01  2.09
 2.15  2.19  2.2   2.22  2.3   2.32  2.34  2.37  2.45  2.5   2.54  2.61
 2.64  2.66  2.69  2.75  2.78  2.83  2.84  2.96  3.02  3.04  3.08  3.11
 3.12  3.17  3.24  3.29  3.3   3.35  3.38  3.42  3.52  3.53  3.55  3.57
 3.63  3.66  3.69  3.73  3.75  3.79  3.89  3.9   3.92  3.95  4.03  4.04
 4.09  4.1   4.11  4.14  4.24  4.27  4.33  4.35  4.42  4.46  4.47  4.48
 4.54  4.57  4.59  4.6   4.61  4.64  4.74  4.77  4.83  4.85  4.86  4.88
 4.92  5.07  5.11  5.12  5.16  5.21  5.25  5.28  5.34  5.36  5.4   5.43
 5.47  5.49  5.49  5.52  5.57  5.59  5.61  5.65  5.7   5.74  5.75  5.78
 5.79  5.82  5.83  5.86  5.88  5.9   5.92  5.94  5.96  5.98  5.99  6.
 6.    6.03  6.05  6.05  6.05  6.08  6.11  6.13  6.18  6.27  6.34  6.42
 6.45  6.46  6.57  6.6   6.62  6.62  6.68  6.81  6.84  6.85  6.9   6.95
 7.02  7.1   7.14  7.16  7.24  7.26  7.38  7.45  7.49  7.53  7.58  7.62
 7.71  7.77  7.83  7.89  7.98  8.04  8.08  8.13  8.21  8.29  8.31  8.41
 8.42  8.48  8.52  8.58  8.72  8.81  9.04  9.09  9.16  9.17  9.19  9.41
 9.45  9.6   9.64  9.69  9.75  9.86  9.89  9.95 10.03 10.11 10.18 10.25
10.35 10.38 10.5   10.62 10.66 10.71 10.79 10.85 10.87 10.97 11.08 11.12
11.15 11.22 11.26 11.29 11.37 11.38 11.38 11.4   11.4   11.41 11.42 11.44
11.46 11.54 11.6   11.64 11.67 11.71 11.71 11.73 11.74 11.8   11.81 11.86
11.9   11.93 11.96 12.   12.03 12.03 12.04 12.05 12.05 12.07 12.09 12.12
12.14 12.19 12.25 12.28 12.33 12.35 12.39 12.44 12.55 12.55 12.61 12.63
12.67 12.7   12.78 12.82 12.82 12.84 12.84 12.87 12.88 12.92 12.97 12.97
12.98 13.    13.02 13.05 13.09 13.1   13.12 13.15 13.16 13.19 13.2   13.2
13.22 13.23 13.24 13.26 13.27 13.29 13.29 13.3   13.33 13.34 13.35 13.37
13.37 13.38 13.39 13.4   13.45 13.46 13.47 13.5   13.51 13.52 13.52 13.55
13.55 13.57 13.57 13.59 13.62 13.66 13.69 13.7   13.71 13.76 13.78 13.81
13.84 13.86 13.92 13.96 13.98 14.01 14.04 14.08 14.11 14.12 14.2   14.22
14.24 14.25 14.27 14.31 14.32 14.33 14.38 14.39 14.43 14.44 14.45 14.45
14.45 14.48 14.52 14.62 14.63 14.65 14.68 14.73 14.78 14.82 14.88 14.95
15.    15.05 15.09 15.13 15.15 15.22 15.23 15.25 15.28 15.34 15.36 15.38
15.45 15.53 15.56 15.59 15.6   15.63 15.64 15.7   15.78 15.82 15.85 15.87
15.93 15.96 15.98 16.03 16.09 16.13 16.16 16.16 16.19 16.23 16.25 16.29
16.34 16.37 16.4   16.44 16.51 16.52 16.54 16.57 16.59 16.61 16.61 16.67
16.68 16.77 16.81 16.87 16.92 16.93 16.96 17.11 17.14 17.21 17.23 17.27
17.37 17.39 17.42 17.48 17.51 17.53 17.57 17.65 17.7   17.75 17.82 17.83
17.86 17.86 17.89 17.89 17.9   17.92 17.94 17.96 17.97 18.    18.02 18.07
18.08 18.1   18.12 18.16 18.21 18.31 18.33 18.44 18.61 18.64 18.65 18.68
18.73 18.82 18.86 18.91 18.98 19.05 19.11 19.25 19.34 19.42 19.44 19.47
19.49 19.59 19.72 19.87 19.97 20.    20.03 20.15 20.22 20.28 20.34 20.41
20.43 20.5   20.54 20.61 20.79 20.84 20.94 20.97 21.17 21.24 21.25 21.28
21.31 21.49 21.5   21.53 21.55 21.57 21.6   21.72 21.77 21.8   21.99 22.02
22.13 22.49 23.25 24.52 25.12 28.61 28.61 30.62 37.28 38.02 38.3   38.3 ]
```

lastni vektorji:

```
[[-0.    0.   -0.    ...  0.   -0.   -0.   ]
 [-0.    0.    0.    ...  0.   -0.   -0.   ]
 [ 0.   -0.   -0.    ... -0.   -0.    0.   ]
 ...
 [ 0.    0.03  0.    ...  0.   -0.   -0.   ]
 [ 0.02  0.    0.05 ... -0.   -0.    0.   ]
 [ 0.   -0.02 -0.    ... -0.    0.    0.   ]]
```

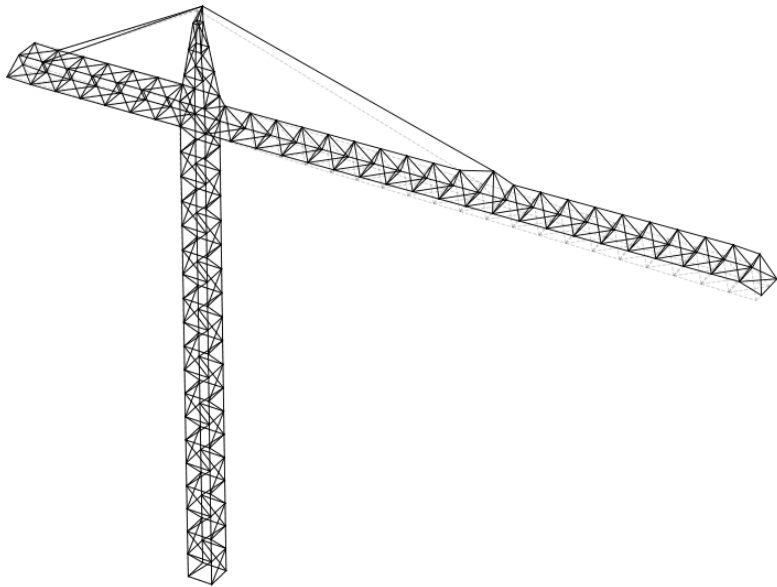
## Modalne oblike

Uporabnik lahko sam izriše in animira željeno modalno obliko.

```

In [20]: f = 1 # parametri
A = 12000
t_max = 6/lastnaFrekvenca[f-1]
t_step = 35
premik_Voz0 = Voz.copy() # začetno stanje
fig = plt.figure()
for t in np.arange(0, t_max, t_step):
    X = A*np.sin(t*lastnaFrekvenca[f-1]*2*np.pi) # "Amplituda" povečanja pomika v sinusni odvisnost
    i od t
    linVoz = np.reshape(Voz, -1).copy()
    j = 0
    for i in range(linVoz.size): # posamezni koordinati prištejemo vektor pomika (če ta ni robni pogoj)
        if i not in RP:
            linVoz[i] = linVoz[i] + X*lastniVektor[j,f-1]
            j+=1
    premik_Voz = np.reshape(linVoz, (-1, 3)) # ponovno preoblikujemo v prvotno obliko
    ax.clear() #Brišemo prejšnji graf pred prikazovanjem novega
    ax = fig.gca(projection='3d')
    ax.set_xlabel('x'); ax.set_ylabel('y'); ax.set_zlabel('z') # oznaka osi
    ax.set_xlim(-raz/4, 3*raz/4); ax.set_ylim(-raz/2, raz/2); ax.set_zlim(0, raz);
    for i in KE: # izris za vsak k.e.
        KE_izris(premik_Voz[:,0], premik_Voz[:,1], premik_Voz[:,2], i[0]-1, i[1]-1);
        KE_skica(premik_Voz0[:,0], premik_Voz0[:,1], premik_Voz0[:,2], i[0]-1, i[1]-1);
    plt.axis('off')
    plt.pause(0.05) #Pavziramo in s tem manipuliramo animacijo
    fig.canvas.draw()

```



## MAC kriterij

Izračunamo MAC vrednost.

```
In [23]: lastniVektorPython = lastniVektor
lastniVektorAnsys = np.loadtxt(fname='AnsysVektorji.txt', dtype='float', delimiter='\t')
r1=lastniVektorPython
r2=lastniVektorAnsys
```

```
In [24]: MAC_ij = np.array([])
for i in range(r1[0].size):
    for j in range(r1[0].size):
        MAC = np.abs(np.conj(r1[:, i]).T @ r2[:, j])**2/(np.conj(r1[:, i]) @ r1[:, i] * np.conj(r2
[:, j]) @ r2[:, j])
        MAC_ij = np.append(MAC_ij, MAC)
```

```
In [25]: MAC_ij=MAC_ij.reshape((r1[0].size, r2[0].size), order='C')
```

```
In [26]: plt.rcParams["figure.figsize"] = (6.5,5)           # Definiramo velikost izrisa grafov
plt.rcParams['figure.dpi'] = 100                          # 200 je standard

SMALL_SIZE = 8
MEDIUM_SIZE = 13
BIGGER_SIZE = 14

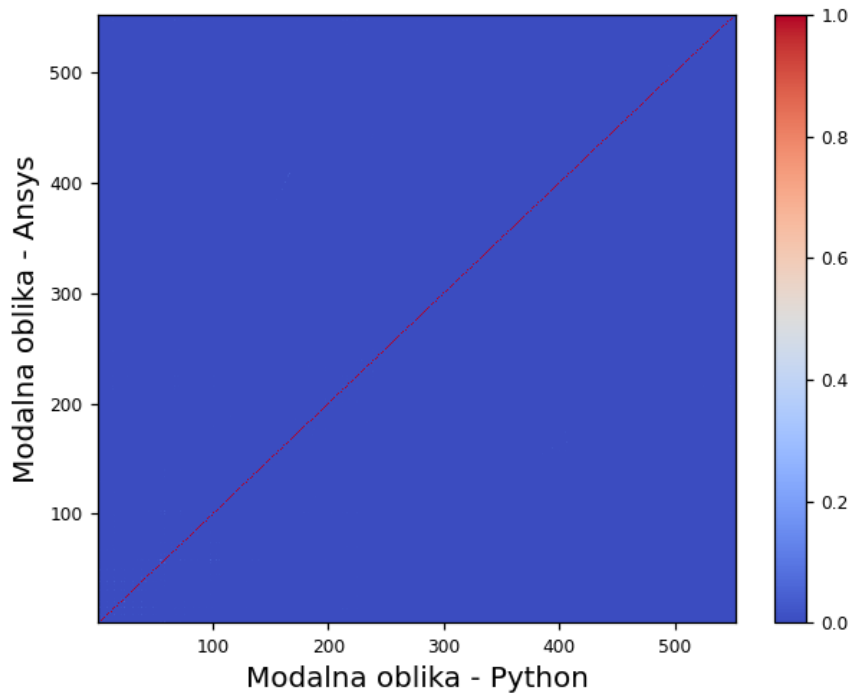
plt.rc('font', size=SMALL_SIZE)                          # controls default text sizes
plt.rc('axes', titlesize=SMALL_SIZE)                      # fontsize of the axes title
plt.rc('axes', labelsiz= MEDIUM_SIZE)                    # fontsize of the x and y labels
plt.rc('xtick', labelsiz=SMALL_SIZE)                      # fontsize of the tick labels
plt.rc('ytick', labelsiz=SMALL_SIZE)                      # fontsize of the tick labels
plt.rc('legend', fontsize=SMALL_SIZE)                     # legend fontsize
plt.rc('figure', titlesize=BIGGER_SIZE)                   # fontsize of the figure title
```

```
In [27]: X = np.linspace(1, lastnaFrekvenca.size, r1[0].size, dtype=int)
Y = X
Z = MAC_ij

fig, ax0 = plt.subplots(1, 1)

c = ax0.pcolor(X, Y, Z, cmap='coolwarm', edgecolors="none", linewidth=0)
fig.colorbar(c, ax=ax0)
ax0.set_xlabel('Modalna oblika - Python')
ax0.set_ylabel('Modalna oblika - Ansys')
#ax0.set_title('Linearno skaliran MAC')

plt.show()
```



```

In [28]: xAmplitudes = np.linspace(1, 12, 12, dtype=int) #your data here
yAmplitudes = xAmplitudes #your other data here - must be same array length

x = np.array(xAmplitudes) #turn x,y data into numpy arrays
y = np.array(yAmplitudes) #useful for regular matplotlib arrays

fig = plt.figure() #create a canvas, tell matplotlib it's 3d
ax = fig.add_subplot(111, projection='3d')

#make histogram stuff - set bins - I choose 20x20 because I have a lot of data
hist, xedges, yedges = np.histogram2d(x, y, bins=(12,12))
xpos, ypos = np.meshgrid(xedges[:-1]+xedges[1:], yedges[:-1]+yedges[1:])

xpos = xpos.flatten()/2.
ypos = ypos.flatten()/2.
zpos = np.zeros_like (xpos)

dx = xedges [1] - xedges [0]
dy = yedges [1] - yedges [0]
dz = hist.flatten()

cmap = cm.get_cmap('coolwarm') # Get desired colormap - you can change this!
max_height = 0.5 # get range of colorbars so we can normalize
min_height = 0
# scale each z to [0,1], and get their rgb values
rgba = [cmap((k-min_height)/max_height) for k in dz]

c=ax.bar3d(xpos, ypos, zpos, dx, dy, dz, color=rgba, zsort='average', edgecolor="black", linewidth=
0.5)
plt.xlabel('Modalna oblika - Python')
plt.ylabel('Modalna oblika - Ansys')
#plt.colorbar(c)
plt.show()

```

