

# Speciální numerické metody

prof. Ing. Martin Krejsa, Ph.D.

Učební materiály

Ostrava, 2022

Učební materiály předmětu **Speciální numerické metody**

Studijní program: B3607 Stavební inženýrství

Učební obor: 3607R037 Konstrukce staveb

**Klíčová slova:**

Algoritmus, MATLAB, přenosová matice, průběhy vnitřních síly a přetvoření, teorie II. řádu, geometricky nelineární výpočet, rovinná příhradová konstrukce, Hookův zákon, vzpěr, stabilita, tlačný štíhlý prut, virtuální práce, Eulerovo analytické řešení, kritické břemeno.

# Předmluva

Učební materiály předmětu „Speciální numerické metody“ jsou určeny pro studenty učebního oboru „Konstrukce staveb“ (3607R037) studijního programu „Stavební inženýrství“ (B3607). Cílem je seznámit studenty s pokročilými numerickými postupy, programátorskými technikami a tvorbou algoritmů pro řešení inženýrských úloh zaměřených na projekční činnost ve stavebnictví.

Pro aplikaci algoritmických postupů bylo zvoleno prostředí interaktivního výpočetního systému MATLAB, které umožňuje tvořit jednoduché výpočetní nástroje bez složitého programování uživatelského prostředí. MATLAB je uživatelsky příjemný nástroj pro implementaci matematických a numerických postupů, ve kterém lze kromě standardních programátorských nástrojů využít knihovnu, obsahující širokou škálu funkcí. Problémem není ani snadné zobrazení dosažených výsledků formou grafu.

Při volbě aplikačního prostředí bylo rovněž přihlédnuto k návaznosti na další předměty, vyučované na Katedře stavební mechaniky, které jsou rovněž zaměřeny na programování a tvorbu pokročilých výpočetních algoritmů.

V Ostravě, březen 2022

prof. Ing. Martin Krejsa, Ph.D.



# Obsah

Předmluva	iii
<b>1 Přenosové matice</b>	<b>1</b>
1.1 Odvození . . . . .	1
1.2 Zatížení . . . . .	5
1.2.1 Spojité zatížení . . . . .	6
1.3 Příklady na procvičení . . . . .	7
<b>2 Geometricky nelineární řešení příhradových konstrukcí</b>	<b>17</b>
2.1 Teoretické pozadí výpočtu . . . . .	17
2.2 Aplikace výpočtu podle teorie II. řádu . . . . .	18
2.2.1 Lineární výpočet rovinné příhradové konstrukce obecnou deformační metodou . . . . .	18
2.2.2 Aplikace řešení lineárního výpočtu rovinné příhradové konstrukce obecnou deformační metodou . . . . .	20
2.2.3 Aplikace řešení iteračního geometricky nelineárního výpočtu rovinné příhradové konstrukce . . . . .	30
<b>3 Stabilita tlačných prutů s využitím principu virtuálních prací</b>	<b>36</b>
3.1 Teoretické pozadí výpočtu . . . . .	36
3.2 Aplikace stabilitního řešení tlačných prutů s využitím principu virtuálních prací . . . . .	39
3.2.1 Příklady na procvičení . . . . .	40
<b>4 Vlastní čísla a vlastní vektory</b>	<b>57</b>
4.1 Úvod . . . . .	57
4.2 Numerické metody pro řešení vlastních čísel . . . . .	60
4.2.1 Částečný problém vlastních čísel . . . . .	60
4.2.2 Úplný problém vlastních čísel . . . . .	62
4.3 Praktické využití vlastních čísel a vlastních vektorů v úlohách stavební mechaniky . . . . .	64
4.3.1 Metoda inverzních iterací . . . . .	64
4.3.2 Stabilita konstrukcí . . . . .	65
4.3.3 Postup výpočtu . . . . .	66

4.3.4	Počáteční aproximace . . . . .	66
4.3.5	Aplikace řešené úlohy . . . . .	66
<b>5</b>	<b>Vlastní frekvence a tvary vlastního kmitání</b>	<b>72</b>
5.1	Úvod do problematiky . . . . .	72
5.1.1	Řešení soustavy pohybových rovnic . . . . .	72
5.1.2	Ortogonalita vlastních tvarů kmitání . . . . .	75
5.1.3	Normované vlastní tvary kmitání . . . . .	75
5.2	Určení vlastních frekvencí a tvarů vlastního kmitání u jednoduchých konstrukcí . . . . .	76
<b>6</b>	<b>Náhodné proměnné a pravděpodobnostní simulační výpočty</b>	<b>85</b>
6.1	Náhodná veličina . . . . .	85
6.1.1	Diskrétní náhodná veličina . . . . .	86
6.1.2	Spojité náhodná veličina . . . . .	86
6.1.3	Parametrické rozdělení pravděpodobnosti . . . . .	87
6.1.4	Neparametrické (empirické) rozdělení pravděpodobnosti . . . . .	88
6.1.5	Generování náhodných proměnných v Matlabu . . . . .	89
6.2	Pravděpodobnostní posouzení nosného prvku . . . . .	100
6.2.1	Účinek zatížení . . . . .	100
6.2.2	Odolnost konstrukce . . . . .	100
6.2.3	Funkce spolehlivosti . . . . .	101
6.2.4	Pravděpodobnost poruchy . . . . .	101
6.2.5	Pravděpodobnostní posudek v Matlabu . . . . .	101
	<b>Literatura</b>	<b>107</b>

# Kapitola 1

## Přenosové matice

### Cíle



Kapitola je zaměřena na:

- seznámení s přenosovými maticemi,
- teoretického pozadí přenosových matic a jejich odvození,
- jejich využití pro výpočet průběhů vnitřních sil a deformací na prutech konstrukce, řešené např. obecnou deformační metodou.

Přenosové matice se používají nejčastěji pro výpočet průběhů složek vnitřních sil a složek přetvoření na jednotlivých prutech stavebních nosných konstrukcí. Jedná se o relativně moderní výpočetní metodu, která se v praxi uplatnila až s možností využití počítačů k technickým výpočtům.

Použití přenosových matic je široké. Lze je použít s dostatečnou přesností i pro řešení prutů zakřivených nebo pro pruty s proměnlivým průřezem. Uplatnění nachází i při odvození tuhostních matic.

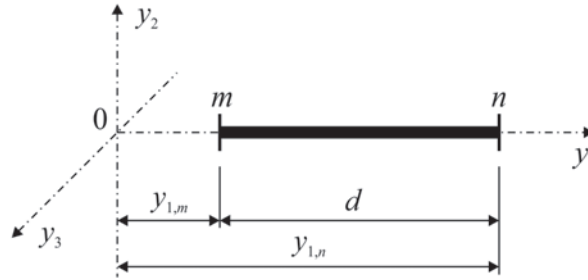
### 1.1 Odvození

Pro odvození přenosových matic je nutné nejprve definovat prizmatický dílek o nenulové délce  $d$  a stálého průřezu, který byl vyjmutý z obecně namáhaného obecného prutu. Pro tento dílek je nutné také definovat lokální kartézský a pravotočivý souřadnicový systém os  $y_i$ , kde osa  $y_1$  bude osou těžištní a osy  $y_2, y_3$  hlavními osami setrvačnosti (vizte obr. 1.1).

Na začátku tohoto dílku  $d$  necht' leží bod  $m$  a na konci bod  $n$ , přičemž platí  $y_{1,m} < y_{1,n}$ . Za předpokladu nejobecnějšího případu, t.j. prutu prostorového rámu, lze definovat v bodě  $m$  subvektor všech šesti složek vnitřních sil:

$$\{S\}_m = \{M_{1,m}, M_{2,m}, M_{3,m}, P_{1,m}, P_{2,m}, P_{3,m}\}^T, \quad (1.1)$$

kde  $M$  jsou složky momentové a  $P$  složky silové. Z možných rovnovážných dvojic



Obr. 1.1 Schéma prizmatického dílku délky  $d$  v lokálním kartézském a pravotočivém souřadnicovém systému os  $y_i$

těchto složek vnitřních sil lze vybrat na začátek i na konec dílku pouze ty, jimiž působí odňaté části prutu na vyšetřovaný dílek. Na konci dílku  $d$  necht' mají vektory kladných složek vnitřních sil shodný smysl s kladnými poloosami  $y_i$  a na začátku dílku smysl opačný.

Obdobně lze definovat v bodě  $m$  subvektor všech šesti složek přetvoření:

$$\{\delta\}_m = \{\varphi_{1,m}, \varphi_{2,m}, \varphi_{3,m}, v_{1,m}, v_{2,m}, v_{3,m}\}^T, \quad (1.2)$$

kde  $\varphi$  jsou složky rotací a  $v$  složky translací styčnicku. Kladné složky rotací a translací mají na začátku i na konci dílku  $d$  smysl shodný s kladnými poloosami  $y_i$ .

Je vhodné připomenout, že posouvající síly nepůsobí v těžišti průřezu, ale prochází středem smyku, což je bod, k němuž musí být definováno zatížení kroutícím momentem. Zároveň se jedná o bod průřezu, který se při kroucení prutu v rovině průřezu neposune.

Budou-li známy všechny prvky obou subvektorů (1.1) a (1.2), lze přesně popsat stav analyzovaného průřezu. Proto lze definovat tzv. *stavový vektor* průřezu  $p$ :

$$\{St\}_p = \{\{S\}_p^T, \{\delta\}_p^T\}^T. \quad (1.3)$$

Počáteční bod střednice dílku  $d$  je stále bod  $m$  a bod koncový  $n$ . Lze předpokládat, že na prutu, z něhož byl vybrán dílek  $d$ , existuje nějaké silové nebo deformační zatížení. V případě, že mezi body  $m$  a  $n$  nepůsobí žádné zatížení, nelze předpokládat rovnost obou vektorů  $\{St\}_m$  a  $\{St\}_n$ . Je proto nutné uvažovat závislost obou vektorů ve tvaru maticového součinu:

$$\{St\}_n = [Pr]_{m,n} \cdot \{St\}_m, \quad (1.4)$$

kde  $[Pr]_{m,n}$  je tzv. přenosovou maticí stavového vektoru bodu  $m$  do bodu  $n$ . Pro prut prostorového rámu se jedná o čtvercovou matici typu 12x12. Pro jiné typy prutových konstrukcí (rovinný rám, rošt, atd.) se počet prvků subvektorů (1.1) a (1.2) i typ přenosové matice příslušným způsobem zredukuje.



Přenosová matice  $[Pr]$  se skládá ze čtyř submatic  $[pr]_{i,j}$ :

$$[Pr] = \begin{bmatrix} [pr]_{1,1} & [pr]_{1,2} \\ [pr]_{2,1} & [pr]_{2,2} \end{bmatrix}, \quad (1.5)$$

kde submatice  $[pr]_{1,2}$  je nulová, neboť ze statických podmínek vyplývá, že na přetvoření bodu  $m$  nemohou mít obecně vliv vnitřní síly bodu  $n$ .

Rozepsáním vztahu (1.5) lze získat:

$$\begin{aligned} \{S\}_n &= [pr]_{1,1} \cdot \{S\}_m, \\ \{\delta\}_n &= [pr]_{2,1} \cdot \{S\}_m + [pr]_{2,2} \cdot \{\delta\}_m. \end{aligned} \quad (1.6)$$

Z uvedeného rozepsání (1.6) je zřejmé, že prvky submatice  $[pr]_{1,1}$  lze získat ze statických podmínek rovnováhy ve tvaru:

$$[pr]_{1,1} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & d \\ 0 & 0 & 1 & 0 & -d & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.7)$$

Prvky submatice  $[pr]_{2,2}$  jsou dány geometrií pohybu nepřetvořeného dílku jako tuhého tělesa ve tvaru:

$$[pr]_{2,2} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & d & 0 & 1 & 0 \\ 0 & -d & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (1.8)$$

Prvky submatice  $[pr]_{2,1}$  jsou dány podmínkami fyzikálními, resp. geometrickými. Lze ji odvodit ve tvaru:

$$[pr]_{2,1} = \begin{bmatrix} \frac{d}{GI_t} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{d}{EI_2} & 0 & 0 & 0 & \frac{d^2}{2EI_2} \\ 0 & 0 & \frac{d}{EI_3} & 0 & \frac{-d^2}{2EI_3} & 0 \\ 0 & 0 & 0 & \frac{d}{EA} & 0 & 0 \\ 0 & 0 & \frac{d^2}{2EI_3} & 0 & \frac{-d^3}{6EI_3} + \frac{d}{GA_2} & 0 \\ 0 & \frac{-d^2}{2EI_2} & 0 & 0 & 0 & \frac{-d^3}{6EI_2} + \frac{d}{GA_3} \end{bmatrix}, \quad (1.9)$$

kde  $E$ ,  $G$  jsou moduly pružnosti v tahu a prostém tlaku, resp. ve smyku použitého materiálu,  $d$  je délka přenosu,  $I_t$  je moment tuhosti v kroucení,  $A$  je plocha průřezu,  $I_2$  je centrální moment setrvačnosti k ose  $y_2$ ,  $I_3$  je centrální moment setrvačnosti

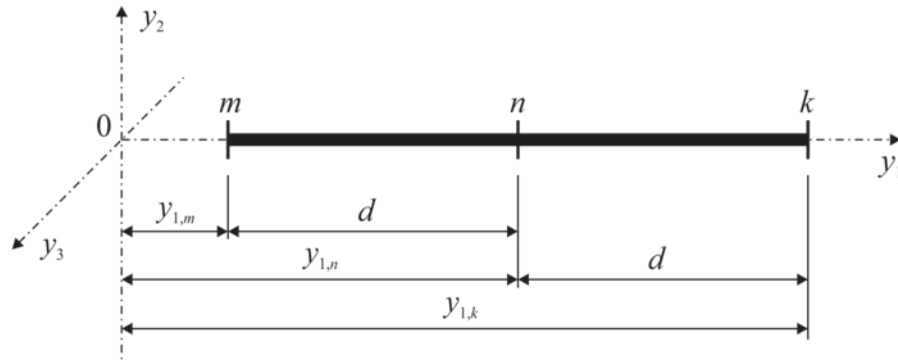
k ose  $y_3$ ,  $A_2$  je redukovaná plocha průřezu pro uplatnění práce posouvající síly  $P_2$  a  $A_3$  je redukovaná plocha průřezu pro uplatnění práce posouvající síly  $P_3$ .

Tímto je základní přenosová matice zcela definovaná.

Z vlastního odvození vyplývá, že při záměně poloh bodů  $m$  a  $n$  by v přenosové matici byla použita hodnota  $-d$  místo  $d$ , jinak se ostatní prvky matice nemění. Z tohoto důvodu je matice  $[Pr]_{n,m}$  shodná s maticí  $[Pr]_{m,n}$ , pouze prvky s lichou mocninou  $d$  mají opačné znaménko.

Pokud by za uvažovaným dílkem  $mn$  následoval dílek  $nk$  se stejným tvarem průřezu a s rovnoběžnými osami  $y_i$  (jak je schématicky zobrazeno na obr. 1.2), pak lze pokračovat v dalším přenosu, přičemž lze získat stavový vektor:

$$\{St\}_k = [Pr]_{n,k} \cdot \{St\}_n = [Pr]_{n,k} \cdot [Pr]_{m,n} \cdot \{St\}_m . \quad (1.10)$$



Obr. 1.2 Schéma dvou po sobě jdoucích prizmatických dílků délky  $d$  v lokálním kartézském a pravotočivém souřadnicovém systému os  $y_i$

Při přímém přenosu z bodu  $m$  do bodu  $k$  však platí:

$$\{St\}_k = [Pr]_{m,k} \cdot \{St\}_m \quad (1.11)$$

a z porovnání obou výrazů (1.10) a (1.11) vyplývá:

$$[Pr]_{m,k} = [Pr]_{n,k} \cdot [Pr]_{m,n} . \quad (1.12)$$

Lze tedy tvrdit, že součin dvou přenosových matic, sestavených pro dva u sebe ležící prvky se shodným (konstantním) průřezem a shodným směrem průvodního trojhranu setrvačnosti, není nutno provádět. Místo něj lze sestavit výslednou přenosovou matici pro délku, jež délkově i znaménkově odpovídá přenosu z bodu, jež je totožný s prvním indexem druhé matice do bodu, jež je totožný s druhým indexem první matice.

## 1.2 Zatížení

Pozornost je nutné věnovat i zatížení. Pro další odvození lze předpokládat, že v jistém bodě  $m$  působí jako osamělé účinky všechny složky obecného zatěžovacího momentu  $N$  a obecné zatěžovací síly  $Q$ .

Složku momentu lze považovat za kladnou tehdy, jestliže její vektor má shodný smysl s kladnou poloosou  $y_i$ . Pro složku síly necht' platí totéž. Všechny tyto složky lze seřadit do vektoru:

$$\{Zg\}_m = \{N_{m,1}, N_{m,2}, N_{m,3}, Q_{m,1}, Q_{m,2}, Q_{m,3}\}^T . \quad (1.13)$$

V témže bodě  $m$  působí současně jako osamělé deformační zatížení všechny složky obecného vektoru rotace  $R$  a obecného vektoru translace  $T$ . I tyto všechny složky lze seřadit do vektoru:

$$\{Z\delta\}_m = \{R_{m,1}, R_{m,2}, R_{m,3}, T_{m,1}, T_{m,2}, T_{m,3}\}^T . \quad (1.14)$$

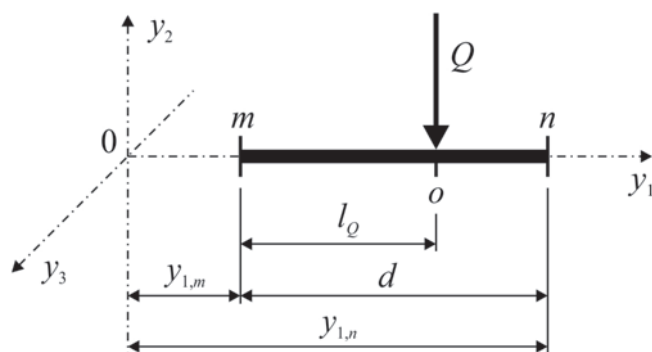
Složky vektoru rotace i složky vektoru translace jsou kladné tehdy, odpovídá-li jejich smysl směru orientace kladné poloosy.

Kompletní vektor singulárního zatížení bodu  $m$  lze definovat ve tvaru:

$$\{Z\}^T = \{-\{Zg\}_m^T, \{Z\delta\}_m^T\} . \quad (1.15)$$

Záporné znaménku u vektoru  $\{Zg\}$  je vynuceno tím, že normální stavový vektor obsahuje vnitřní síly vlastního dílku (např. síly, jimiž působí uzel na styčník), kdežto v tomto vektoru jsou obsaženy složky, působící na uzel. Tyto složky se liší znaménkem.

V dalším výpočtu necht' ve vzdálenosti  $l_Q$  v bodě  $o$  působí na vyšetřovaném dílku vektor  $\{Z\}$  singulárních zatěžovacích účinků (jako např. na schématu 1.3).



Obr. 1.3 Schéma prizmatického dílku délky  $d$  s vyznačeným bodem  $o$ , kde působí vektor  $\{Z\}$  singulárních zatěžovacích účinků

Na začátku dílku v bodě  $m$  platí stavový vektor  $\{St\}_m$ . Na konci první části (před působením zatížení) se změní na stavový vektor  $\{St\}_o$  daný výrazem  $[Pr]_{m,o} \cdot \{St\}_m$ . Těsně za působištěm zatížení se však změní jeho hodnota o  $\{Z\}$ , takže na začátku tohoto dílku má stavový vektor  $\{St\}_o$  velikost  $[Pr]_{m,o} \cdot \{St\}_m + \{Z\}_o$ . Na konci dílku v bodě  $n$  má stavový vektor hodnotu:

$$\begin{aligned} \{St\}_n &= [Pr]_{o,n} \cdot ([Pr]_{m,o} \cdot \{St\}_m + \{Z\}_o) = \\ &= [Pr]_{o,n} \cdot [Pr]_{m,o} \cdot \{St\}_m + [Pr]_{o,n} \cdot \{Z\}_o \end{aligned} \quad (1.16)$$

a s přihlédnutím ke vztahu (1.12) je možno provést úpravu:

$$\{St\}_n = [Pr]_{m,n} \cdot \{St\}_m + [Pr]_{o,n} \cdot \{Z\}_o . \quad (1.17)$$

Pokud by zatěžovací vektor ležel v bodě  $n$ , pak by přenosový vztah nabýval tvaru:

$$\{St\}_n = [Pr]_{m,n} \cdot \{St\}_m + \{Z\}_n . \quad (1.18)$$

Porovnáním vztahů (1.17) a (1.18) lze zjistit platnost  $\{Z\}_n = [Pr]_{o,n} \cdot \{Z\}_o$ . Znamená to, že zatěžovací vektor lze rovněž přenášet pomocí přenosové matice a zatěžovací vektor bodu  $o$  lze nahradit staticky a deformačně ekvivalentním zatěžovacím vektorem působícím v jiném bodě. Pro takový vektor platí např.:

$$\{Z\}_n = [Pr]_{o,n} \cdot \{Z\}_o \quad (1.19)$$

nebo

$$\{Z\}_m = [Pr]_{o,m} \cdot \{Z\}_o , \quad (1.20)$$

kde  $[Pr]_{o,m}$  je přenosová matice sestavená pro délku přenosu  $d = y_m - y_o < 0$ .

### 1.2.1 Spojité zatížení

Nyní necht' na délce  $d_z$  působí zatěžovací vektor spojitého zatížení, jehož intenzitu lze vyjádřit mocninnou řadou ve tvaru:

$$\begin{aligned} \{z\} &= \{z_0\} + y \cdot \{z_1\} + y^2 \cdot \{z_2\} + y^3 \cdot \{z_3\} + \dots + y^k \cdot \{z_k\} = \\ &= \sum_{i=0}^k y^i \cdot \{z_i\} , \end{aligned} \quad (1.21)$$

kde počátek nezávislé proměnné  $y$  necht' je totožný s levým okrajem působícího zatížení. Z tohoto spojitého zatížení lze jakékoliv diferenciální zatížení nahradit ekvivalentním zatížením na jeho počátku pomocí přenosu  $d\{z\}_0 = [Pr]_{y,0} \cdot \sum_{i=0}^k y^i \cdot \{z_i\} \cdot dy$ .

Celé toto zatížení lze tedy nahradit staticky a deformačně ekvivalentním vektorem singulárního zatížení na počátku zatěžovacího intervalu ve tvaru:

$$\{Z\}_0 = \sum_{i=0}^k \int_0^{d_z} [Pr]_{y,0} \cdot y^i \cdot dy \cdot \{z_i\} = \sum_{i=0}^k \begin{bmatrix} [pr]_{1,1}^i & [0] \\ [pr]_{2,1}^i & [pr]_{2,2}^i \end{bmatrix} \cdot \{z_i\}, \quad (1.22)$$

kde jednotlivé submatice  $[pr]^i$  nabývají tvar:

$$[pr]_{1,1}^i = \begin{bmatrix} \frac{d_z^{i+1}}{i+1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{d_z^{i+1}}{i+1} & 0 & 0 & 0 & -\frac{d_z^{i+2}}{i+2} \\ 0 & 0 & \frac{d_z^{i+1}}{i+1} & 0 & \frac{d_z^{i+2}}{i+2} & 0 \\ 0 & 0 & 0 & \frac{d_z^{i+1}}{i+1} & 0 & 0 \\ 0 & 0 & 0 & 0 & \frac{d_z^{i+1}}{i+1} & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{d_z^{i+1}}{i+1} \end{bmatrix}, \quad (1.23)$$

$$[pr]_{2,2}^i = \begin{bmatrix} \frac{d_z^{i+1}}{i+1} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{d_z^{i+1}}{i+1} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{d_z^{i+1}}{i+1} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{d_z^{i+1}}{i+1} & 0 & 0 \\ 0 & 0 & -\frac{d_z^{i+2}}{i+2} & 0 & \frac{d_z^{i+1}}{i+1} & 0 \\ 0 & \frac{d_z^{i+2}}{i+2} & 0 & 0 & 0 & \frac{d_z^{i+1}}{i+1} \end{bmatrix} \quad (1.24)$$

a

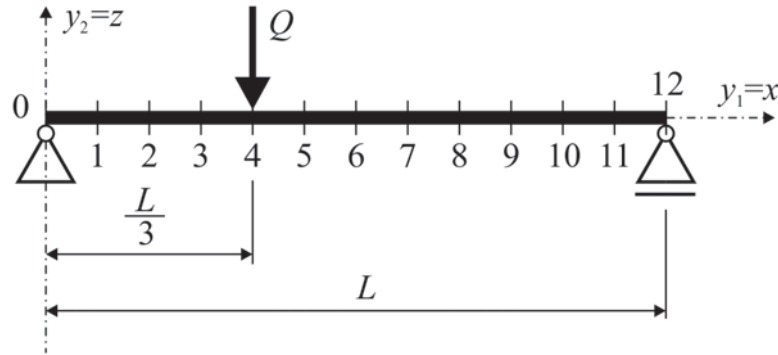
$$[pr]_{2,1}^i = \begin{bmatrix} \frac{-d_z^{i+2}}{GI_t \cdot (i+2)} & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{-d_z^{i+2}}{EI_2 \cdot (i+2)} & 0 & 0 & 0 & \frac{d_z^{i+3}}{2EI_2 \cdot (i+3)} \\ 0 & 0 & \frac{-d_z^{i+2}}{EI_3 \cdot (i+2)} & 0 & \frac{-d_z^{i+3}}{2EI_3 \cdot (i+3)} & 0 \\ 0 & 0 & 0 & \frac{-d_z^{i+2}}{EA \cdot (i+2)} & 0 & 0 \\ 0 & 0 & \frac{d_z^{i+3}}{2EI_3 \cdot (i+3)} & 0 & \frac{d_z^{i+4}}{6EI_3 \cdot (i+4)} - \frac{d_z^{i+2}}{GA_2 \cdot (i+2)} & 0 \\ 0 & \frac{-d_z^{i+3}}{2EI_2 \cdot (i+3)} & 0 & 0 & 0 & \frac{d_z^{i+4}}{6EI_2 \cdot (i+4)} - \frac{d_z^{i+2}}{GA_3 \cdot (i+2)} \end{bmatrix}. \quad (1.25)$$

## 1.3 Příklady na procvičení

Pro přiblížení použití přenosových matic následuje několik demonstračních příkladů.

**Příklad 1.1.** Určete průběh vnitřních sil a deformačních veličin na prostém nosníku délky  $L$  z obrázku 1.4, který je ve třetině rozpětí zatížen osamělým břemenem velikosti  $Q$ . Nosník rozdělte na 12 dílků délky  $d = \frac{L}{12}$ .





Obr. 1.4 Schéma řešeného nosníku z příkladu 1.1

*Řešení.* V daném případě se jedná o namáhání rovinným ohybem (např. kolem osy  $y_3$ ). Z šesti složek vnitřních sil jsou nenulové pouze ohybový moment  $M_{m,3}$  a posouvající síla  $P_{m,2}$ , ze složek přetvoření pouze pootočení  $\varphi_{m,3}$  a průhyb  $v_{m,2}$ . V obou případech je to třetí a pátý prvek subvektorů (1.1) a (1.2). U odvozených matic bude proto do výpočtu vstupovat pouze třetí a pátý sloupec a řádek (ostatní prvky jsou nulové a mohou být tedy vypuštěny).

Základní přenosové submatice budou mít tedy výsledný tvar:

$$[pr]_{1,1} = \begin{bmatrix} 1 & -d \\ 0 & 1 \end{bmatrix}, \quad (1.26)$$

$$[pr]_{2,1} = \frac{d}{EI_3} \cdot \begin{bmatrix} 1 & \frac{-d}{2} \\ \frac{d}{2} & \frac{-d^2}{6} + \frac{EI_3}{GA_2} \end{bmatrix} \quad (1.27)$$

a

$$[pr]_{2,2} = \begin{bmatrix} 1 & 0 \\ d & 1 \end{bmatrix}. \quad (1.28)$$

Při řešení lze vyjít ze skutečnosti, že stavový vektor na konci nosníku je možno získat jako funkci stavového vektoru na začátku nosníku a daného zatížení. Lze pak uplatnit okrajové podmínky (nulový ohybový moment a nulový průhyb na obou koncích) a určit nenulové hodnoty stavových vektorů na obou koncích nosníku.

Pro jednoduchost stavový vektor na začátku nosníku bude zadán:

$$\{St\}_{y_1=0} = \left\{ 0, \frac{2}{3} \cdot Q, \frac{5}{81} \cdot \frac{Q \cdot L^2}{EI_3}, 0 \right\}^T. \quad (1.29)$$

Pokud je žádoucí získat průběhy vnitřních sil a přetvárných veličin např. po dvanáctinách rozpětí, bude nutné sestavit přenosovou matici pro  $d = \frac{L}{12}$  ve tvaru:

$$[Pr]_{y_1=0, y_1=12} = \begin{bmatrix} 1 & -\frac{L}{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \\ \frac{L}{12EI_3} & -\frac{L^2}{288EI_3} & 1 & 0 \\ \frac{L^2}{288EI_3} & \frac{-L^3}{10368EI_3} + \frac{L}{12GA_2} & \frac{L}{12} & 1 \end{bmatrix}. \quad (1.30)$$

Touto maticí se vynásobí vždy zleva stávající vektor, čímž vznikne vektor v dalším bodě. Ve třetině rozpětí se stavový vektor opraví o vektor zatížení  $\{Z\}_{y_1=4} = \{0, -Q, 0, 0\}^T$  a pokračuje se v přenosu až do konce nosníku.

Celý výpočetní postup lze provést např. s využitím následujícího skriptu, kde byly zadány také konkrétní vstupní údaje. Pro názornost je v tomto skriptu uveden detailní popis algoritmu s využitím komentářů za znakem %:

```
% Příklad 1 - průběh vnitřních sil a přetvárných veličin na prostém
% nosníku od osamělého břemene, které působí ve třetině rozpětí
clc; clear;
% Vstupní údaje
L=6; % rozpětí [m]
b=0.05; % šířka obdélníkového průřezu [m]
h=0.2; % výška obdélníkového průřezu [m]
A=b*h; % průřezová plocha obdélníkového průřezu [m^2]
I=1/12*b*h^3; % moment setrvačnosti obdélníkového průřezu [m^4]
E=9.5*10^9; % modul pružnosti v tahu a prostém tlaku,
% listnaté dřeviny, třída pevnosti D18 [Pa]
G=0.59*10^9; % modul pružnosti ve smyku,
% listnaté dřeviny, třída pevnosti D18 [Pa]
Av=0.833*A; % smyková plocha obdélníkového průřezu [m^2]
Q=10*10^3; % Velikost bodového zatížení [N]
y1Q=L/3; % Souřadnice působiště zatížení [m]
n=12; % Počet dílků na prutu
d=L/n; % Šířka dílku [m]

% Přenosová matice pro d
Pr=[1 -d 0 0; 0 1 0 0;
d/(E*I) -d^2/(2*E*I) 1 0;
d^2/(2*E*I) -d^3/(6*E*I)+d/(G*Av) d 1];

% Stavový vektor v bodě y1=0
St=[0 2/3*Q 5*Q*L^2/(81*E*I) 0]';

Z=[0 -Q 0 0]'; % Vektor zatížení
kolik=1; % kolik bylo určeno hodnot průběhů
y1(kolik)=0; % aktuální souřadnice y1
M(kolik)=St(1); % velikost ohybového momentu
```

```

V(kolik)=St(2); % velikost posouvající síly
fi(kolik)=St(3); % velikost pootočení
w(kolik)=St(4); % velikost průhybu
for i=1:n
    kolik=kolik+1;
    St=Pr*St;
    y1(kolik)=y1(kolik-1)+d; % aktuální souřadnice y1
    M(kolik)=St(1); % velikost ohybového momentu
    V(kolik)=St(2); % velikost posouvající síly
    fi(kolik)=St(3); % velikost pootočení
    w(kolik)=St(4); % velikost průhybu
    if (y1Q>y1(kolik-1))&(y1Q<=y1(kolik)) % Účinek zatížení
        kolik=kolik+1;
        St=St+Z;
        y1(kolik)=y1(kolik-1); % aktuální souřadnice y1
        M(kolik)=St(1); % velikost ohybového momentu
        V(kolik)=St(2); % velikost posouvající síly
        fi(kolik)=St(3); % velikost pootočení
        w(kolik)=St(4); % velikost průhybu
    end;
end;

% Vykreslení výsledných průběhů vnitřních sil a přetvoření
figure
subplot(2,2,1) % Graf v 1.kvadrantu v rastru 2 x 2 - Vz
hold on
plot(y1,V/1000,'r-','LineWidth',2);
title('\fontsize{12}Průběh posouvající síly \itV_{z}(x)');
xlabel('x [m]');
ylabel('Vz(x) [kN]');
stem(y1,V/1000,'r-','marker','none');
hold off
subplot(2,2,2) % Graf ve 2.kvadrantu v rastru 2 x 2 - M
hold on
plot(y1,M/1000,'m-','LineWidth',2);
title('\fontsize{12}Průběh ohybových momentů \itM_{y}(x)');
xlabel('x [m]');
ylabel('My(x) [kNm]');
% set(gca,'YDir','reverse');
stem(y1,M/1000,'m-','marker','none');
hold off
subplot(2,2,3) % Graf ve 3.kvadrantu v rastru 2 x 2 - fi
hold on

```

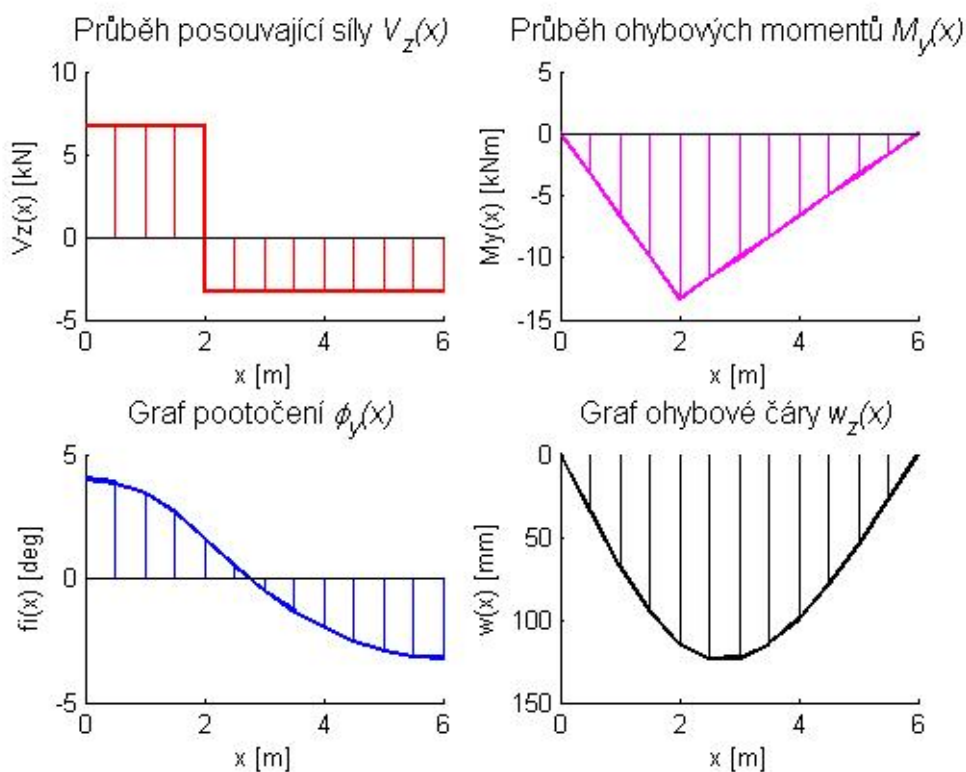


```

plot(y1,fi/pi*180,'b-','LineWidth',2);
title('\fontsize{12}Graf pootočení \it\phi_{y}(x)');
xlabel('x [m]');
ylabel('fi(x) [deg]');
stem(y1,fi/pi*180,'b-','marker','none');
hold off
subplot(2,2,4)      % Graf ve 4. kvadrantu v rastru 2 x 2 - w
hold on
plot(y1,w*1000,'k-','LineWidth',2);
title('\fontsize{12}Graf ohybové čáry \itw_{z}(x)');
xlabel('x [m]');
ylabel('w(x) [mm]');
set(gca,'YDir','reverse');
stem(y1,w*1000,'k-','marker','none');
hold off

```

Výsledné průběhy vnitřních sil a přetvoření jsou uvedeny na obrázku 1.5.



Obr. 1.5 Průběhy vnitřních sil a přetvoření na nosníku z příkladu 1.1

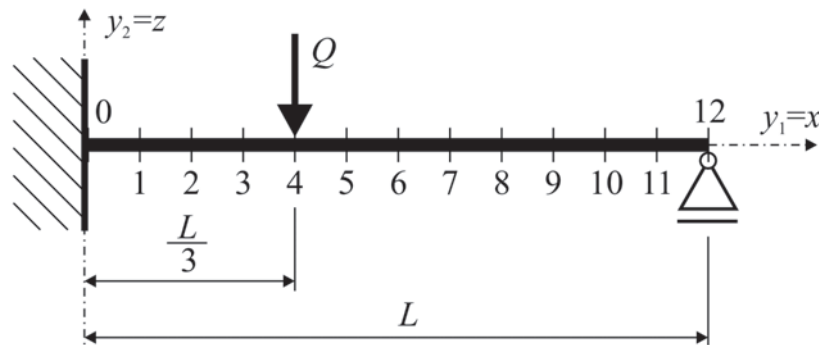




**Pro zájemce:** Při výpočtu se zanedbáním práce posouvajících sil lze v submatici (1.27) vynulovat člen  $\frac{EI_3}{GA_2} = 0$ , resp. v matici (1.30) zlomek  $\frac{L}{12GA_2} = 0$ , což se projeví na výsledných přetvořeních. Porovnejte s výsledky příkladu 1.1.



**Příklad 1.2.** Podobně jako v příkladu 1.1 určete průběh posouvajících sil, ohybových momentů, pootočení a průhybu na jednostranně vlevo vetknutém staticky neurčitěm nosníku délky  $L$ , který je ve třetině rozpětí zatížen osamělým břemenem velikosti  $Q$  v souladu se schématem na obrázku 1.6. Nosník rozdělte na 12 dílků délky  $d = \frac{L}{12}$ .



Obr. 1.6 Schéma řešeného jednostranně vlevo vetknutého staticky neurčitěho nosníku z příkladu 1.2

*Řešení.* Stavový vektor na začátku jednostranně vlevo vetknutého staticky neurčitěho nosníku se rovná:

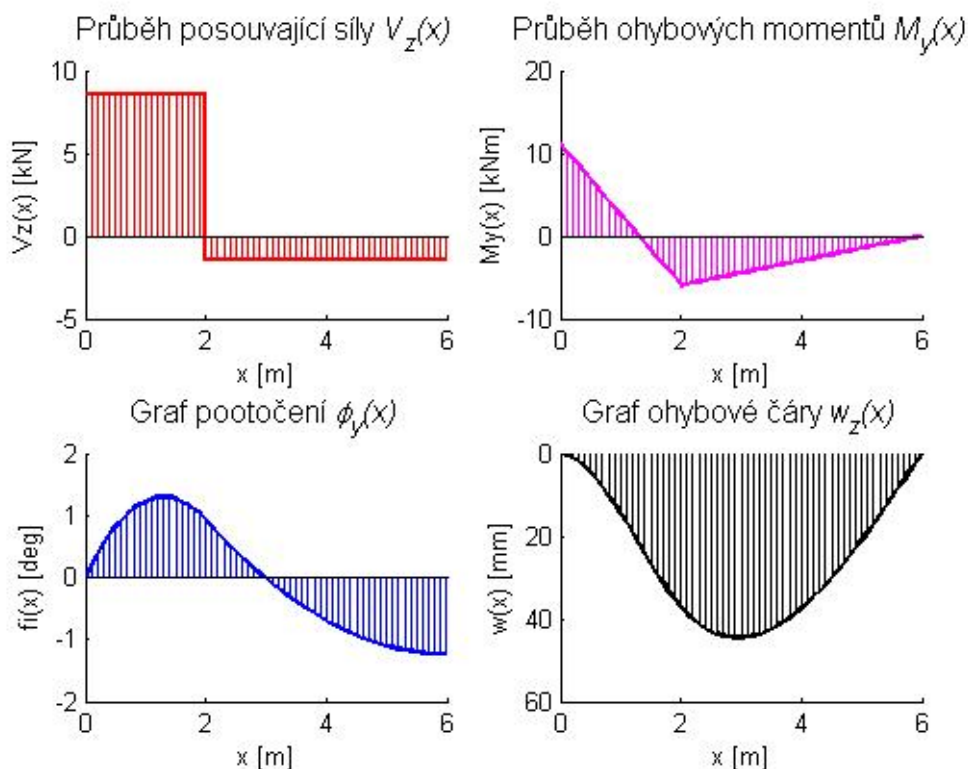
$$\{St\}_{y_1=0} = \left\{ \frac{5}{27 + 81 \cdot \frac{EI_3}{GA_2 \cdot L^2}} \cdot Q \cdot L, \frac{23 + 54 \cdot \frac{EI_3}{GA_2 \cdot L^2}}{27 + 81 \cdot \frac{EI_3}{GA_2 \cdot L^2}} \cdot Q, 0, 0 \right\}^T. \quad (1.31)$$

Zbývající výpočet je již identický s řešením příkladu 1.1.

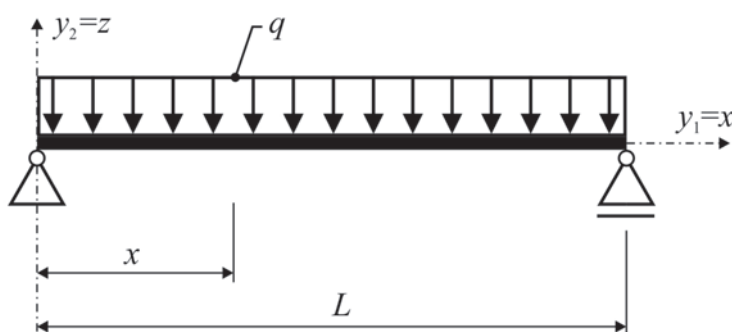
Výsledné průběhy vnitřních sil a přetvárných veličin jsou pro počet dílků  $n = 60$  ke shlédnutí na obrázku 1.7.



**Poznámka:** U staticky neurčitěho nosníku se případné zanedbání nebo nerespektování práce posouvajících sil projeví kromě velikosti přetvárných veličin také na velikosti vnitřních sil.



Obr. 1.7 Průběhy vnitřních sil a přetvoření na nosníku z příkladu 1.2



Obr. 1.8 Schéma řešeného prostě podepřeného nosníku z příkladu 1.3



**Příklad 1.3.** Určete průběh vnitřních sil i deformačních veličin na prostém nosníku z obrázku 1.8 délky  $L$ , který je zatížen rovnoměrným spojitým zatížením velikosti  $q$ . Nosník rozdělte na 30 dílků délky  $d = \frac{L}{30}$ .

*Řešení.* Stavový vektor na začátku nosníku je pro tento případ roven:

$$\{St\}_{y_1=0} = \left\{ 0, \frac{q \cdot L}{2}, \frac{q \cdot L^3}{24 \cdot EI_3}, 0 \right\}^T. \quad (1.32)$$

Účinek spojitého zatížení je do výpočtu zahrnut s využitím vztahů definovaných v kapitole 1.2.1 pro  $i = 0$ , takže zatěžovací vektor rovnoměrného spojitého zatížení  $\{z\}$  podle výrazu (1.21) je roven:

$$\{z\} = \{z_0\} = \{0, -q, 0, 0\}^T. \quad (1.33)$$

Jak již bylo dáno vztahem (1.22), spojitě zatížení lze i pro tento příklad nahradit příslušným staticky a deformačně ekvivalentním vektorem singulárního zatížení na počátku zatěžovacího intervalu šířky  $d_z$  ve tvaru:

$$\{Z\}_0 = \begin{bmatrix} d_z & \frac{d_z^2}{2} & 0 & 0 \\ 0 & d_z & 0 & 0 \\ -\frac{d_z^2}{2 \cdot EI_3} & -\frac{d_z^3}{6 \cdot EI_3} & d_z & 0 \\ \frac{d_z^3}{6 \cdot EI_3} & \frac{d_z^4}{24 \cdot EI_3} - \frac{d_z^2}{2 \cdot GA_2} & -\frac{d_z^2}{2} & d_z \end{bmatrix} \cdot \{z_0\}. \quad (1.34)$$

Celý výpočetní postup lze provést např. pomocí následujícího skriptu, kde byly zadány také konkrétní vstupní údaje a pro názornost výpočetního algoritmu je v něm opět uveden popis s využitím komentářů za znakem %:

```
% Příklad 3 - průběh vnitřních sil a přetvárných veličin na prostém
% nosníku od konstantního spojitého zatížení q
clc; clear;

% Vstupní údaje
L=6;           % rozpětí [m]
b=0.05;       % šířka obdélníkového průřezu [m]
h=0.2;        % výška obdélníkového průřezu [m]
A=b*h;        % průřezová plocha obdélníkového průřezu [m^2]
I=1/12*b*h^3; % moment setrvačnosti obdélníkového průřezu [m^4]
E=9.5*10^9;   % modul pružnosti v tahu a prostém tlaku, listnaté
              % dřeviny, třída pevnosti D18 [Pa]
G=0.59*10^9; % modul pružnosti ve smyku, listnaté dřeviny,
              % třída pevnosti D18 [Pa]
Av=0.833*A;   % smyková plocha obdélníkového průřezu [m^2]
q=2*10^3;     % velikost rovnoměrného spojitého zatížení [N/m]
n=30;        % počet dílků na prutu
d=L/n;        % šířka dílku [m]
```

```

% Přenosová matice pro d
Pr=[1 -d 0 0;
    0 1 0 0;
    d/(E*I) -d^2/(2*E*I) 1 0;
    d^2/(2*E*I) -d^3/(6*E*I)+d/(G*Av) d 1];

% Stavový vektor v bodě y1=0
St=[0 q*L/2 q*L^3/(24*E*I) 0]';

% Zatěžovací vektor rovnoměrného spojitého zatížení (i=0)
z=[0 -q 0 0]';

% Vektor singulárního zatížení na počátku zatěžovacího intervalu
% s využitím přenosové matice pro i=0
Z0=[d d^2/2 0 0;
    0 d 0 0;
    -d^2/(2*E*I) -d^3/(6*E*I) d 0;
    d^3/(6*E*I) d^4/(24*E*I)-d^2/(2*G*Av) -d^2/2 d]*z;

kolik=1;          % kolik bylo určeno hodnot průběhů
y1(kolik)=0;     % aktuální souřadnice y1
M(kolik)=St(1);  % velikost ohybového momentu
V(kolik)=St(2);  % velikost posouvající síly
fi(kolik)=St(3); % velikost pootočení
w(kolik)=St(4);  % velikost průhybu
for i=1:n
    kolik=kolik+1;
    y1(kolik)=y1(kolik-1)+d; % aktuální souřadnice y1
    St=Pr*(St+Z0);          % aktuální stavový vektor - s vlivem
                            % spojitého zatížení
    M(kolik)=St(1);        % velikost ohybového momentu
    V(kolik)=St(2);        % velikost posouvající síly
    fi(kolik)=St(3);       % velikost pootočení
    w(kolik)=St(4);        % velikost průhybu
end;

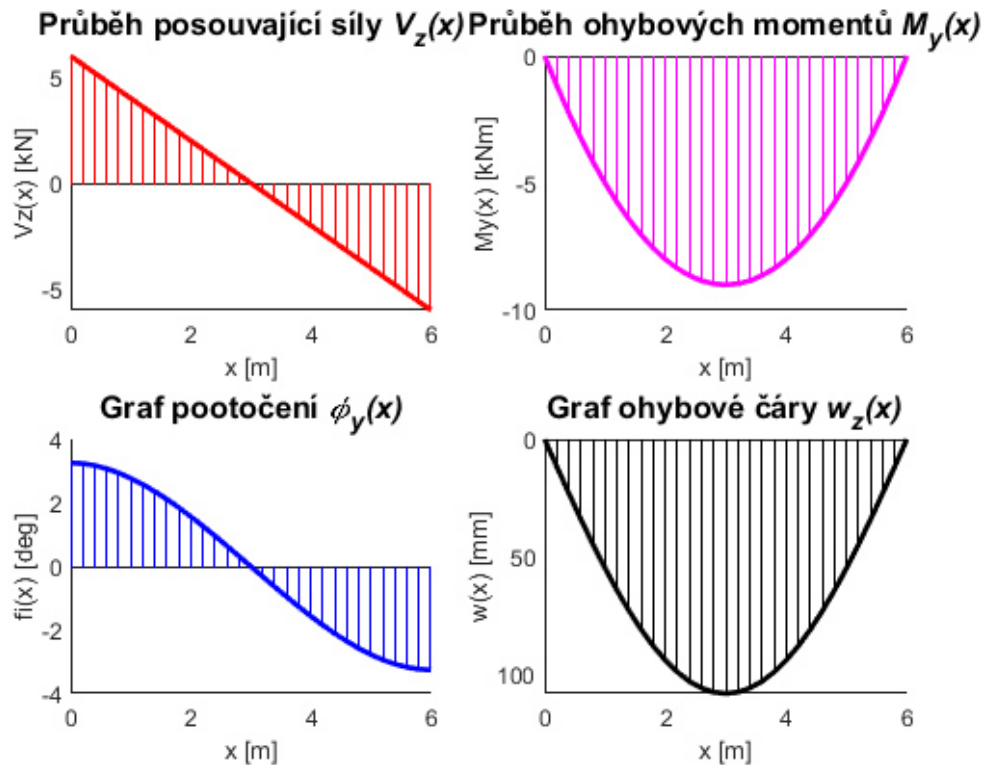
```

Zobrazení okna s výslednými průběhy vnitřních sil a přetvoření je provedeno shodně s řešením příkladu 1.1 a je uvedeno na obrázku 1.9.



**Poznámka:** Celý výpočetní postup lze uplatnit např. pro určení průběhů vnitřních sil a přetvárných veličin u prutových konstrukcí řešených obecnou deformační metodou. Stavový vektor na začátku každého z prutů pak lze sestavit na základě výsledných hodnot vnitřních sil a uzlových přetvoření.





Obr. 1.9 Průběhy vnitřních sil a přetvoření na nosníku z příkladu 1.3



**Pro zájemce:** Přenosových matic lze využít také pro řešení prutů proměnlivého průřezu. V takovém případě lze v každém dílku předpokládat konstantní průřezové charakteristiky totožné se skutečnými vlastnostmi průřezu v polovině dílku. Takové řešení sice nelze považovat za absolutně přesné, dává ale výsledky s dostatečnou přesností.



**Pro zájemce:** Podobně lze řešit i nosníky zakřivené, přičemž lze střednici nahradit polygonem velmi krátkých úsečků, jež jsou považovány za přímé. V takovém případě po zadání stavového vektoru na konci dílku je nutné provést transformaci do souřadnicového systému, v němž je vyjádřeno případné zatížení uzlů.

Po přičtení vektoru zatížení se provede další transformace do systému os průvodního trojhranu setrvačnosti dalšího dílku a takto upravený vektor se využije pro přenos na další dílku. Při této transformaci se bere v úvahu i eventuální změna polohy středu křivosti nebo excentrické připojení dalšího dílku.



**Poznámka:** Zvětšování počtu dílků (zmenšování délek přenosu) má pozitivní dopad na výsledky, protože se zvyšuje přesnost. Existuje však hranice, po níž se výsledky začnou opět zhoršovat v důsledku negativního vlivu zaokrouhlovacích chyb. Vyzkoušejte např. na skriptu uvedeném u příkladu 1.1 pro počet dílků  $n = 12, 60, 120, 600$  a 1200. Sledujte přitom hodnotu ohybového momentu a průhybu na konci nosníku.

## Kapitola 2

# Geometricky nelineární řešení příhradových konstrukcí

### Cíle



Kapitola je zaměřena na:

- seznámení s geometricky nelineárním řešením prutových konstrukcí,
- teoretického pozadí jednoho ze způsobů geometricky nelineárního řešení příhradových nosníků,
- prohloubení znalostí aplikace obecné deformační metody.

Jeden ze způsobů geometricky nelineárního výpočtu příhradových nosníků podle teorie II. řádu je založen na iteračním výpočtu s linearizací jednotlivých kroků iterace. Při lineárním výpočtu se na příhradovou konstrukci nechá působit zatížení, čímž vznikne protažení nebo stlačení jednotlivých prutů. Konstrukce se zdeformuje a nastane nerovnováha mezi výpočteným stavem a původním tvarem. Tato nerovnováha se odstraní dalším výpočtem.

### 2.1 Teoretické pozadí výpočtu

Při výpočtu rovinných i prostorových příhradových nosníků podle teorie II. řádu se osová síly určí iteračním způsobem s využitím vztahu, který vychází z Hookova zákona:

$$N_i = \frac{E_i \cdot A_i}{L_i} \cdot (L_{N,i} - L_i), \quad (2.1)$$

kde  $i$  je číslo příslušného prutu,  $E$  je modul pružnosti v tahu a prostém tlaku,  $A$  je plocha průřezu,  $L$  je původní délka prutu, určená např. ze zadané geometrie konstrukce (souřadnic uzlů), a  $L_N$  je délka prutu určená v příslušném iteračním cyklu podle teorie II. řádu. Člen  $\frac{E \cdot A}{L}$  se nazývá tahová tuhost a pro fyzikálně lineární výpočty je konstantní.

Pro názornost a jednoduchost je následující výklad zaměřen pouze na rovinnou úlohu, přičemž úloha prostorová se bude lišit pouze třetím rozměrem.

Délka  $L_N$  se pro rovinný příhradový nosník může např. určit ze vztahu:

$$L_N = \sqrt{\{d\}^T \{d\}} . \quad (2.2)$$

Vektor  $\{d\}$  je přitom definován:

$$\{d\}^T = \{\Delta y_1 + \Delta u_{y_1}, \Delta y_2 + \Delta u_{y_2}\} , \quad (2.3)$$

kde  $\Delta y_1$  a  $\Delta y_2$  jsou rozdíly souřadnic počátečního a koncového uzlu prutu,  $\Delta u_{y_1}$  a  $\Delta u_{y_2}$  pak rozdíly složek deformací uzlů ve směru dané osy.

## 2.2 Aplikace výpočtu podle teorie II. řádu

Popsaný iterační geometricky nelineární výpočetní postup podle (2.1) se skládá z jednotlivých kroků iterace, které jsou založeny na lineárním výpočtu příhradové konstrukce obecnou deformační metodou.

### 2.2.1 Lineární výpočet rovinné příhradové konstrukce obecnou deformační metodou

Neboť je řešení příhradových nosníků obecnou deformační metodou náplní předmětu Statika stavebních konstrukcí II, popis řešení se zaměří pouze na nejdůležitější vztahy a principy.

Lokální deformační vektor prutu s koncovými body  $a$ ,  $b$  má jen dva nenulové prvky:

$$r_{ab}^* = \{u_a^*, u_b^*\}^T , \quad (2.4)$$

stejně jako lokální vektor koncových sil:

$$R_{ab}^* = \{X_a^*, X_b^*\}^T . \quad (2.5)$$

V rovinných příhradových konstrukcích má globální deformační vektor prutu čtyři prvky:

$$r_{ab} = \{u_a, w_a, u_b, w_b\}^T , \quad (2.6)$$

podobně jako globální vektor koncových sil:

$$R_{ab} = \{X_a, Z_a, X_b, Z_b\}^T . \quad (2.7)$$

Pro prut rovinné příhradové konstrukce je lokální matice tuhosti typu  $2 \times 2$ :

$$k_{ab}^* = \frac{E \cdot A_{ab}}{l_{ab}} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} . \quad (2.8)$$



Pro lokální vektor koncových sil pak platí:

$$R_{ab}^* = \begin{Bmatrix} X_a^* \\ X_b^* \end{Bmatrix} = k_{ab}^* \cdot r_{ab}^* = k_{ab}^* \cdot \begin{Bmatrix} u_a^* \\ u_b^* \end{Bmatrix} = \frac{E \cdot A_{ab}}{l_{ab}} \cdot \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \cdot \begin{Bmatrix} u_a^* \\ u_b^* \end{Bmatrix}. \quad (2.9)$$

Transformační matice  $T_{ab}$  vyjadřuje geometrickou závislost lokálních parametrů deformace na globálních. Pro prut rovinné příhradové konstrukce ji lze zapsat ve tvaru:

$$T_{ab}^* = \begin{Bmatrix} \cos \gamma_{ab} & \sin \gamma_{ab} & 0 & 0 \\ 0 & 0 & \cos \gamma_{ab} & \sin \gamma_{ab} \end{Bmatrix} = \begin{Bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{Bmatrix}. \quad (2.10)$$

Pro převod z globálních parametrů deformace prutu rovinné příhradové konstrukce na lokální pak platí:

$$r_{ab}^* = \begin{Bmatrix} u_a^* \\ u_b^* \end{Bmatrix} = T_{ab} \cdot r_{ab} = \begin{Bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{Bmatrix} \cdot \begin{Bmatrix} u_a \\ w_a \\ u_b \\ w_b \end{Bmatrix}. \quad (2.11)$$

Naopak pro převod z lokálních parametrů deformace prutu rovinné příhradové konstrukce na globální lze použít vztah:

$$r_{ab} = \begin{Bmatrix} u_a \\ w_a \\ u_b \\ w_b \end{Bmatrix} = T_{ab}^T \cdot r_{ab}^* = \begin{Bmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{Bmatrix} \cdot \begin{Bmatrix} u_a^* \\ u_b^* \end{Bmatrix}. \quad (2.12)$$

Obdobně lze definovat transformační vztahy pro výpočet vektorů koncových sil prutu rovinné příhradové konstrukce  $R_{ab}^*$  v lokálním a  $R_{ab}$  v globálním souřadnicovém systému:

$$R_{ab}^* = \begin{Bmatrix} X_a^* \\ X_b^* \end{Bmatrix} = T_{ab} \cdot R_{ab} = \begin{Bmatrix} c & s & 0 & 0 \\ 0 & 0 & c & s \end{Bmatrix} \cdot \begin{Bmatrix} X_a \\ Z_a \\ X_b \\ Z_b \end{Bmatrix} \quad (2.13)$$

a

$$R_{ab} = \begin{Bmatrix} X_a \\ Z_a \\ X_b \\ Z_b \end{Bmatrix} = T_{ab}^T \cdot R_{ab}^* = \begin{Bmatrix} c & 0 \\ s & 0 \\ 0 & c \\ 0 & s \end{Bmatrix} \cdot \begin{Bmatrix} X_a^* \\ X_b^* \end{Bmatrix}. \quad (2.14)$$

Nutno připomenout, že platí  $X_a^* = -X_b^*$ .

Na základě vztahů (2.9), (2.11) a (2.14) lze odvodit:

$$R_{ab} = T_{ab}^T \cdot R_{ab}^* = T_{ab}^T \cdot k_{ab}^* \cdot r_{ab}^* = T_{ab}^T \cdot k_{ab}^* \cdot T_{ab} \cdot r_{ab} = k_{ab} \cdot r_{ab} . \quad (2.15)$$

Pro globální matici tuhosti prutu rovinné příhradové konstrukce tedy platí:

$$k_{ab} = T_{ab}^T \cdot k_{ab}^* \cdot T_{ab} . \quad (2.16)$$

Matice  $k_{ab}$  každého prutu rovinné příhradové konstrukce pak přispívá k matici tuhosti  $\mathbf{K}$  celé soustavy:

$$\mathbf{K} \cdot \mathbf{r} = \mathbf{S} , \quad (2.17)$$

kde  $\mathbf{r}$  je vektor uzlových přetvoření celé soustavy v globálním souřadnicovém systému a  $\mathbf{S}$  je vektor uzlových zatížení v globálním souřadnicovém systému.

Aby tuto soustavu bylo možné vyřešit a získat tak pro každý prut vektory  $r_{ab}$  ve vztahu (2.6), je nutné zohlednit vnější vazby konstrukce - okrajové podmínky. Jeden ze způsobů řešení je vynulování příslušného řádku a sloupce soustavy včetně pravé strany, které odpovídají zadané fixaci konstrukce v příslušném styčnicku a směru, a vložit na diagonálu celkové matice tuhosti  $\mathbf{K}$  hodnotu 1. Pak již lze soustavu lineárních rovnic úspěšně vyřešit a pomocí vztahů (2.11) a (2.9) určit i vnitřní síly na každém z prutů.

Pak už zbývá jen dopočítat reakce pomocí silových podmínek rovnováhy v uzlech s vnější vazbou. Pomocí vztahu (2.15) lze určit koncové vnitřní síly na každém prutu v řešeném styčnicku a s využitím vektoru uzlových zatížení  $\mathbf{S}$  pak lze stanovit i hodnotu reakcí.

## 2.2.2 Aplikace řešení lineárního výpočtu rovinné příhradové konstrukce obecnou deformační metodou



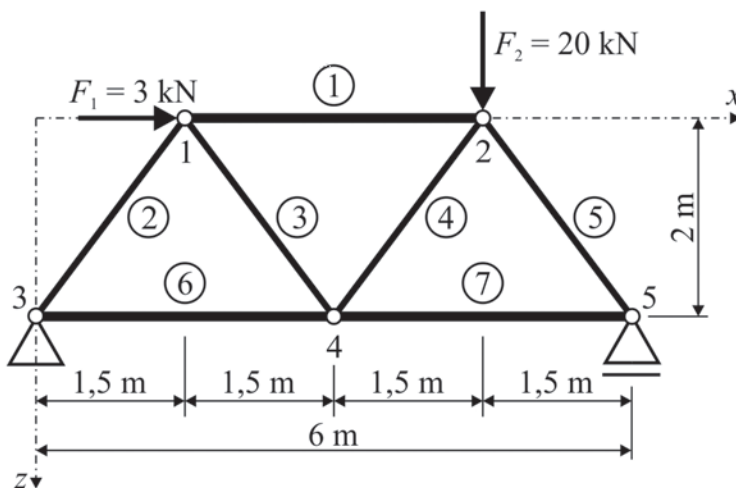
**Příklad 2.1.** Určete lineárním výpočtem s využitím obecné deformační metody posunutí styčnicků, vnitřní síly a reakce rovinného příhradového nosníku z obrázku 2.1, který je zadán s využitím následujícího skriptu (detailní popis zadání je proveden s využitím komentářů za znakem % přímo v tomto skriptu):

```
clear; clc;
format short;
% Matice souřadnic x a z jednotlivých uzlů
uzly=[1.5 0; 4.5 0; 0 2; 3 2; 6 2];
% Matice vstupních údajů prutů - číslo počátečního uzlu,
% číslo koncového uzlu a číslo řádku matice prurmat
% s průřezovými a materiálovými charakteristikami
pruty=[1 2 1; 3 1 2; 1 4 2; 4 2 2; 2 5 2; 3 4 1; 4 5 1];
% Matice průřezových a materiálových charakteristik - A, E
prurmat=[0.0015 2e11; 0.00125 2e11];
```

```

% Matice vazeb - číslo uzlu, u, w - 0 znamená, že není vazba,
% 1 znamená, že je vazba
vazby=[3 1 1; 5 0 1];
% Matice uzlových zatížení - číslo uzlu, Fx, Fz
uzelzat=[1 3e3 0; 2 0 20e3];

```



Obr. 2.1 Schéma řešené příhradové konstrukce z příkladu 2.1.

*Řešení.* Zpracování vstupních údajů a příprava na výpočet probíhá prostřednictvím skriptu s názvem `prihrada2D_1.m`:

```

% Zpracování vstupních údajů a příprava na výpočet
pocprutu=size(pruty,1); % Zjištění počtu prutů
pocuzlu=size(uzly,1); % Zjištění počtu uzlů
pocvazeb=size(vazby,1); % Zjištění počtu vazeb
pocuzelzat=size(uzelzat,1); % Zjištění počtu uzlových zatížení

% Výpočet geometrie prutů
for i=1:1:pocprutu
    % Délka prutu
    geom(i,1)=sqrt((uzly(pruty(i,2),1)-uzly(pruty(i,1),1))^2+...
        (uzly(pruty(i,2),2)-uzly(pruty(i,1),2))^2);
    % Cosinus směrového úhlu prutu
    geom(i,2)=(uzly(pruty(i,2),1)-uzly(pruty(i,1),1))/geom(i,1);
    % Sinus směrového úhlu prutu
    geom(i,3)=(uzly(pruty(i,2),2)-uzly(pruty(i,1),2))/geom(i,1);
end
clear i;

```

Zdrojový text aplikace je obsažen ve skriptu s názvem `prihrada2D_2.m` a má následující obsah:

```
% vynulování globální matice tuhosti a zetěžovacího vektoru
for i=1:1:pocuzlu*2
    for j=1:1:pocuzlu*2
        mat_tuh(i,j)=0;
    end
    zat_vektor(i)=0;
end

% Sestavení soustavy lineárních rovnic
for i=1:1:pocprutu
    % Transformační matice
    T=[geom(i,2) geom(i,3) 0 0; 0 0 geom(i,2) geom(i,3)];
    % Lokální matice tuhosti prutu
    mat_tuh_prutu=prurmat(pruty(i,3),2)*prurmat(pruty(i,3),1)/...
        geom(i,1)*[1 -1; -1 1];

    % Transformace matice tuhosti prutu
    % z lokálního do globálního souřadnicového systému
    mat_tuh_prutu_glob=T'*mat_tuh_prutu*T;

    % Přičtení matice tuhosti prutu
    % do globální matice tuhosti celé soustavy
    for j=1:1:2
        for k=1:1:2
            mat_tuh((pruty(i,1)-1)*2+j,(pruty(i,1)-1)*2+k)=...
                mat_tuh((pruty(i,1)-1)*2+j,(pruty(i,1)-1)*2+k)+...
                mat_tuh_prutu_glob(j,k);
            mat_tuh((pruty(i,2)-1)*2+j,(pruty(i,1)-1)*2+k)=...
                mat_tuh((pruty(i,2)-1)*2+j,(pruty(i,1)-1)*2+k)+...
                mat_tuh_prutu_glob(j+2,k);
            mat_tuh((pruty(i,1)-1)*2+j,(pruty(i,2)-1)*2+k)=...
                mat_tuh((pruty(i,1)-1)*2+j,(pruty(i,2)-1)*2+k)+...
                mat_tuh_prutu_glob(j,k+2);
            mat_tuh((pruty(i,2)-1)*2+j,(pruty(i,2)-1)*2+k)=...
                mat_tuh((pruty(i,2)-1)*2+j,(pruty(i,2)-1)*2+k)+...
                mat_tuh_prutu_glob(j+2,k+2);
        end
    end
end % Konec cyklu pro skládání soustavy lineárních rovnic
```

```
if ~pocuzelzat==0
% Přičtení vektoru uzlového zatížení
% do zatěžovacího vektoru celé soustavy
  for i=1:1:pocuzelzat
    for j=1:1:2
      zat_vektor((uzelzat(i,1)-1)*2+j)=...
      zat_vektor((uzelzat(i,1)-1)*2+j)+uzelzat(i,1+j);
    end
  end
end

% Respektování vazeb
for i=1:1:pocvazeb

  % V uzlu je vazba proti vodorovnému posunutí
  if vazby(i,2)==1
    for j=1:1:pocuzlu*2
      % Vynulování sloupce
      mat_tuh((vazby(i,1)-1)*2+1,j)=0;
      % Vynulování řádku
      mat_tuh(j,(vazby(i,1)-1)*2+1)=0;
    end
    % Na diagonálu přiřadit 1
    mat_tuh((vazby(i,1)-1)*2+1,(vazby(i,1)-1)*2+1)=1;
    % Vynulování zatěžovacího vektoru
    zat_vektor((vazby(i,1)-1)*2+1)=0;
  end

  % V uzlu je vazba proti svislému posunutí
  if vazby(i,3)==1
    for j=1:1:pocuzlu*2
      % Vynulování sloupce
      mat_tuh((vazby(i,1)-1)*2+2,j)=0;
      % Vynulování řádku
      mat_tuh(j,(vazby(i,1)-1)*2+2)=0;
    end
    % Na diagonálu přiřadit 1
    mat_tuh((vazby(i,1)-1)*2+2,(vazby(i,1)-1)*2+2)=1;
    % Vynulování zatěžovacího vektoru
    zat_vektor((vazby(i,1)-1)*2+2)=0;
  end
end
end
```

```

% Řešení soustavy lineárních rovnic
r=mat_tuh\zat_vektor';

% Vynulování vektoru pro sčítání koncových sil na prutech
for i=1:1:pocuzlu*2
    soucet_sil(i)=0;
end

% Globální a lokální vektory sil a deformací jednotlivých prutů
for i=1:1:pocprutu
    for j=1:1:2
        vektor_def_glob(j)=r((pruty(i,1)-1)*2+j);
        vektor_def_glob(j+2)=r((pruty(i,2)-1)*2+j);
    end

    % Transformační matice
    T=[geom(i,2) geom(i,3) 0 0; 0 0 geom(i,2) geom(i,3)];

    % Lokální matice tuhosti prutu
    mat_tuh_prutu=prurmat(pruty(i,3),2)*prurmat(pruty(i,3),1)/...
        geom(i,1)*[1 -1; -1 1];

    % Transformace matice tuhosti prutu
    % z lokálního do globálního souřadnicového systému
    mat_tuh_prutu_glob=T'*mat_tuh_prutu*T;

    % Globální koncové síly jednotlivých prutů
    konc_sily_glob=mat_tuh_prutu_glob*vektor_def_glob';

    % Přičtení globálních koncových sil jednotlivých prutů do celkového
    % součtu sil ve styčnicku
    for j=1:1:2
        soucet_sil((pruty(i,1)-1)*2+j)=...
            soucet_sil((pruty(i,1)-1)*2+j)+konc_sily_glob(j);
        soucet_sil((pruty(i,2)-1)*2+j)=...
            soucet_sil((pruty(i,2)-1)*2+j)+konc_sily_glob(j+2);
    end

    % Lokální vektory deformací jednotlivých prutů
    vek_def_lok=T*vektor_def_glob';

    % Lokální vektory koncových sil jednotlivých prutů
    vek_sil_lok(i,1:2)=mat_tuh_prutu*vek_def_lok;

```

```

end

% Výpočet reakcí
for i=1:1:pocuzlu*2
    reakce(i)=soucet_sil(i);
end

if ~pocuzelzat==0
    % Odečtení vektoru uzlového zatížení od součtu koncových sil prutů
    % v globálním souřadnicovém systému
    for i=1:1:pocuzelzat
        for j=1:1:2
            reakce((uzelzat(i,1)-1)*2+j)=...
                reakce((uzelzat(i,1)-1)*2+j)-uzelzat(i,1+j);
        end
    end
end

clear T i j k konc_sily_glob mat_tuh mat_tuh_prutu;
clear mat_tuh_prutu_glob soucet_sil vek_def_lok;
clear vektor_def_glob zat_vektor;

```

Celý lineární výpočet obecnou deformační metodou pak lze provést s využitím skriptu `prihrada2D.m`, který obsahuje instrukce:

```

% Zpracování vstupních údajů a příprava na výpočet
prihrada2D_1;
% Výpis vstupních veličin
vypis_vstup;
% Geometricky lineární výpočet
prihrada2D_2;
% Výpis výstupních veličin
vypis_vystup;

```

Výpočet pak lze doplnit o přehledný výpis vstupních, resp. výstupních údajů. Obsah skriptu označeného jako `vypis_vstup.m` umožňuje vypsát veškeré vstupní údaje výpočtu zadané příhradové konstrukce:

```

% Výpis vstupních veličin
clc;
disp(sprintf('\n'))
disp('*****')
disp('*   Souřadnice uzlů v globálním souřadnicovém systému   *')
disp('*****')
disp(' ')

```

```

disp('číslo      x      z ')
disp(' uzlu      [m]      [m]')
disp('-----')
for i=1:1:pocuzlu
    disp(sprintf('%5d %8.3f %8.3f',i,uzly(i,1),uzly(i,2)))
end

disp(sprintf('\n'))
disp('*****')
disp('*                          Vlastnosti prutů                          *')
disp('*****')
disp(' ')
disp('číslo číslo číslo      l      cos      sin      A      E')
disp('prutu uzlu1 uzlu2      [m]      [-]      [-]      [m2]      [GPa]')
disp('-----')
for i=1:1:pocprutu
    disp(sprintf('%5d %5d %5d %6.3f %7.4f %7.4f %9.2e %6.1e',i,...
        pruty(i,1),pruty(i,2),geom(i,1),geom(i,2),geom(i,3),...
        prurmat(pruty(i,3),1),prurmat(pruty(i,3),2)*1e-9))
end

disp(sprintf('\n'))
disp('*****')
disp('*                          Vnější vazby                          *')
disp('*****')
disp(' ')
disp('číslo')
disp(' uzlu      ux      uz')
disp('-----')
for i=1:1:pocvazeb
    ret='';
    for j=2:1:3
        if vazby(i,j)==1
            ret=strcat(ret,' fix');
        else
            ret=strcat(ret,' -');
        end
    end
end;
ret=strcat(sprintf('%5d ',vazby(i,1)),ret);
disp(ret);
end

```



```

if pocuzelzat>0
    disp(sprintf('\n'))
    disp('*****')
    disp('*                Uzlová zatížení                *')
    disp('*****')
    disp(' ')
    disp('číslo      Fx      Fz ')
    disp(' uzlu      [kN]    [kN]')
    disp('-----')
    for i=1:pocuzelzat
        disp(sprintf('%5d %9.3f %9.3f',uzelzat(i,1),uzelzat(i,2)*1e-3,...
            uzelzat(i,3)*1e-3));
    end
end

clear i j ret;

```

Výpis vstupních údajů zadané úlohy s řešenou příhradovou konstrukcí z příkladu 2.1 může s využitím skriptu vypis\_vstup.m vypadat následovně:

```

*****
*   Souřadnice uzlů v globálním souřadnicovém systému   *
*****

```

číslo uzlu	x [m]	z [m]
1	1.500	0.000
2	4.500	0.000
3	0.000	2.000
4	3.000	2.000
5	6.000	2.000

```

*****
*                   Vlastnosti prutů                   *
*****

```

číslo prutu	číslo uzlu1	číslo uzlu2	l [m]	cos [-]	sin [-]	A [m <sup>2</sup> ]	E [GPa]
1	1	2	3.000	1.0000	0.0000	1.50e-03	2.0e+02
2	3	1	2.500	0.6000	-0.8000	1.25e-03	2.0e+02
3	1	4	2.500	0.6000	0.8000	1.25e-03	2.0e+02
4	4	2	2.500	0.6000	-0.8000	1.25e-03	2.0e+02

5	2	5	2.500	0.6000	0.8000	1.25e-03	2.0e+02
6	3	4	3.000	1.0000	0.0000	1.50e-03	2.0e+02
7	4	5	3.000	1.0000	0.0000	1.50e-03	2.0e+02

```
*****
*                               Vnější vazby                               *
*****
```

```
číslo
uzlu   ux   uz
-----
   3   fix  fix
   5    -  fix
```

```
*****
*                               Uzlová zatížení                               *
*****
```

```
číslo      Fx      Fz
uzlu      [kN]    [kN]
-----
   1      3.000    0.000
   2      0.000    20.000
```

Skript pojmenovaný `vypis_vystup.m` pak slouží pro výpis výsledných deformací, vnitřních sil a reakcí řešené příhradové konstrukce:

```
disp(sprintf('\n'))
disp('*****')
disp('*   Deformace uzlů v globálním souřadnicovém systému   *')
disp('*****')
disp(' ')
disp('číslo      ux      uz')
disp('uzlu      [mm]    [mm]')
disp('-----')
for i=1:1:pocuzlu
    disp(sprintf('%5d %9.3f %9.3f',i,r((i-1)*2+1)*1e3,...
        r((i-1)*2+2)*1e3))
end
disp(sprintf('\n'))
disp('*****')
disp('*           Koncové vnitřní síly v prutech           *')
disp('*           v lokálním souřadnicovém systému           *')
disp('*****')
```

```

disp(' ')
disp('číslo číslo      N číslo      N ')
disp('prutu uzlu1      [kN] uzlu2      [kN]')
disp('-----')
for i=1:1:pocprutu
    disp(sprintf('%5d %5d %9.3f %5d %9.3f',i,pruty(i,1),...
        vek_sil_lok(i,1)*1e-3,pruty(i,2),vek_sil_lok(i,2)*1e-3));
end
disp(sprintf('\n'))
disp('*****')
disp('*                      Reakce                      *')
disp('*****')
disp(' ')
disp('číslo      Rx      Rz ')
disp(' uzlu      [kN]      [kN]')
disp('-----')
for i=1:1:pocuzlu
    for j=1:1:pocvazeb
        if vazby(j,1)==i
            disp(sprintf('%5d %10.3f %10.3f',i,reakce((i-1)*2+1)*1e-3,...
                reakce((i-1)*2+2)*1e-3));
        end
    end
end
clear i j;

```

Výpis výsledných veličin řešení zadané úlohy s příhradovou konstrukcí z příkladu 2.1 s využitím skriptu `vypis_vystup.m` je pak uveden v následující ukázce:

```

*****
*   Deformace uzlů v globálním souřadnicovém systému   *
*****

```

číslo uzlu	ux [mm]	uz [mm]
1	0.141	0.168
2	0.051	0.347
3	0.000	0.000
4	0.060	0.291
5	0.180	0.000

```
*****
*           Koncové vnitřní síly v prutech           *
*           v lokálním souřadnicovém systému       *
*****
```

číslo prutu	číslo uzlu1	N [kN]	číslo uzlu2	N [kN]
1	1	9.000	2	-9.000
2	3	5.000	1	-5.000
3	1	-5.000	4	5.000
4	4	5.000	2	-5.000
5	2	20.000	5	-20.000
6	3	-6.000	4	6.000
7	4	-12.000	5	12.000

```
*****
*                               Reakce                               *
*****
```

číslo uzlu	Rx [kN]	Rz [kN]
3	-3.000	-4.000
5	0.000	-16.000



### 2.2.3 Aplikace řešení iteračního geometricky nelineárního výpočtu rovinné příhradové konstrukce

*Řešení.* Následuje popis skriptu, který s využitím všech dříve popsanych výpočetních kroků umožňuje iteračně vyřešit zadanou příhradovou konstrukci podle teorie II. řádu:

```
% Záloha původních souřadnic uzlů
uzly_old=uzly;
% Zpracování vstupních údajů a příprava na výpočet
prihrada2D_1;
% Záloha původních délek prutů
delky_old=geom(:,1);
% Vypis vstupních veličin
```

```

vypis_vstup;
% Geometricky lineární výpočet
prihrada2D_2;
% Výpis výstupních veličin lineárního výpočtu
vypis_vystup;
% Záloha vypočtených vnitřních sil
N_old=-vek_sil_lok(:,1)';
N=N_old;

% Zpracování výsledků předchozího výpočtu
% a příprava na další nelineární výpočet
%
% Úprava souřadnic bodů
for i=1:1:pocuzlu
    uzly(i,1)=uzly_old(i,1)+r((i-1)*2+1);
    uzly(i,2)=uzly_old(i,2)+r((i-1)*2+2);
end

% Výpočet nové geometrie prutů
for i=1:1:pocprutu
    % Délka prutu
    geom(i,1)=sqrt((uzly(pruty(i,2),1)-uzly(pruty(i,1),1))^2+...
        (uzly(pruty(i,2),2)-uzly(pruty(i,1),2))^2);
    % Cosinus směrového úhlu prutu
    geom(i,2)=(uzly(pruty(i,2),1)-uzly(pruty(i,1),1))/geom(i,1);
    % Sinus směrového úhlu prutu
    geom(i,3)=(uzly(pruty(i,2),2)-uzly(pruty(i,1),2))/geom(i,1);
end

% Výpis výsledných veličin
num_it=1;
disp(sprintf('\n'))
disp('*****')
disp('*          Upravené délky prutů a vnitřní síly          *')
disp(sprintf(...
    '*          Iterace č.: %5d          *',...
    num_it));
disp('*****')
disp(' ')
disp('číslo          L          N')
disp('prutu          [mm]          [kN]')
disp('-----')
for i=1:1:pocprutu

```

```

N_new(i)=prurmat(pruty(i,3),2)*prurmat(pruty(i,3),1)/...
    delky_old(i)*(geom(i,1)-delky_old(i));
disp(sprintf('%5d %9.3f %9.3f',i,geom(i,1)*1e3,N_new(i)*1e-3))
end

while norm(N_old-N_new)>1e-10
    num_it=num_it+1;
    % Další výpočetní krok s upravenou geometrií
    prihrada2D_2;
    % Zpracování výsledků předchozího výpočtu
    % a příprava na další nelineární výpočet
    %
    % Úprava souřadnic bodů
    for i=1:1:poczlu
        uzly(i,1)=uzly_old(i,1)+r((i-1)*2+1);
        uzly(i,2)=uzly_old(i,2)+r((i-1)*2+2);
    end
    % Výpočet nové geometrie prutů
    for i=1:1:pocprutu
        % Délka prutu
        geom(i,1)=sqrt((uzly(pruty(i,2),1)-uzly(pruty(i,1),1))^2+...
            (uzly(pruty(i,2),2)-uzly(pruty(i,1),2))^2);
        % Cosinus směrového úhlu prutu
        geom(i,2)=(uzly(pruty(i,2),1)-uzly(pruty(i,1),1))/geom(i,1);
        % Sinus směrového úhlu prutu
        geom(i,3)=(uzly(pruty(i,2),2)-uzly(pruty(i,1),2))/geom(i,1);
    end

    % Výpis výsledných veličin
    disp(sprintf('\n'))
    disp('*****')
    disp('*          Upravené délky prutů a vnitřní síly          *')
    disp(sprintf(...
        '*          Iterace č.: %5d          *',...
        num_it));
    disp('*****')
    disp(' ')
    disp('číslo          L          N')
    disp('prutu          [mm]          [kN]')
    disp('-----')
    N_old=N_new;
    for i=1:1:pocprutu
        N_new(i)=prurmat(pruty(i,3),2)*prurmat(pruty(i,3),1)/...

```

```

        delky_old(i)*(geom(i,1)-delky_old(i));
        disp(sprintf('%5d %9.3f %9.3f',i,geom(i,1)*1e3,N_new(i)*1e-3))
    end
end

% Vypis výsledných veličin
disp(sprintf('\n'))
disp('*****')
disp('*   Srovnání výsledků lineárního a nelineárního výpočtu   *')
disp('*****')
disp(' ')
disp('číslo      Nlin      Nnelin      rozdíl')
disp('prutu      [kN]      [kN]      [%]')
disp('-----')
for i=1:1:pocprutu
    disp(sprintf('%5d %9.3f %9.3f %9.3f',i,N(i)*1e-3,N_new(i)*1e-3,...
        (abs(N_new(i))-abs(N(i)))/abs(N(i))*100))
end
uzly=uzly_old;

% Výpočet nové geometrie prutů
for i=1:1:pocprutu
    % Délka prutu
    geom(i,1)=sqrt((uzly(pruty(i,2),1)-uzly(pruty(i,1),1))^2+...
        (uzly(pruty(i,2),2)-uzly(pruty(i,1),2))^2);
    % Cosinus směrového úhlu prutu
    geom(i,2)=(uzly(pruty(i,2),1)-uzly(pruty(i,1),1))/geom(i,1);
    % Sinus směrového úhlu prutu
    geom(i,3)=(uzly(pruty(i,2),2)-uzly(pruty(i,1),2))/geom(i,1);
end

clear uzly_old delky_old N_old i r vek_sil_lok;

```

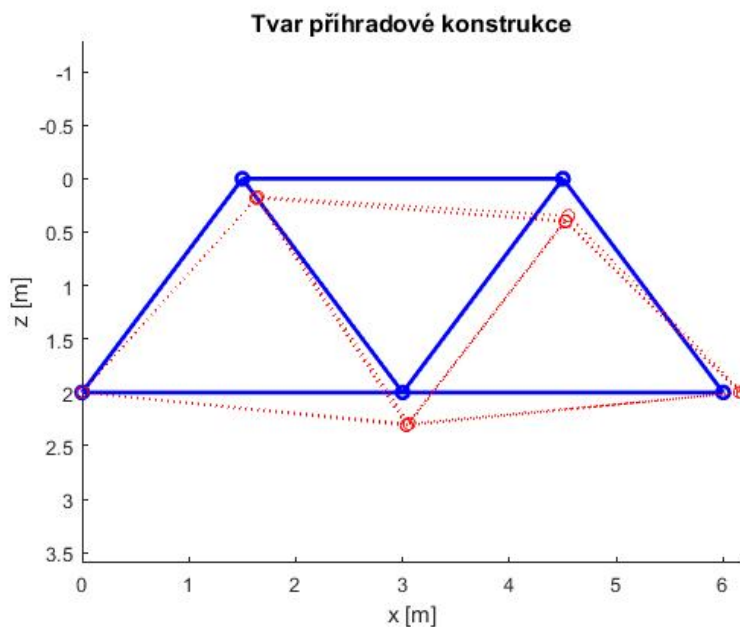


**Příklad 2.2.** Vyřešte rovinný příhradový nosník z příkladu 2.1 geometricky nelineárně podle teorie II. řádu. Sledujte rozdíl v hodnotách vypočtených vnitřních sil u lineárního a geometricky nelineárního výpočtu. V dalším výpočtu snižte tuhosti prutů řešené konstrukce (např. zmenšením hodnoty modulu pružnosti v tahu a protěm tlaku  $E$  o řád, dva a tři řády) a opět sledujte vliv na vypočtené vnitřní síly u lineárního a nelineárního výpočtu.



*Řešení.* Tvar původní příhradové konstrukce a postupná změna tvaru při iteračním geometricky nelineárním výpočtu rovinné příhradové konstrukce z příkladu 2.1 se

zadaným modulem pružnosti v tahu a prostém tlaku  $E = 2 \cdot 10^2$  MPa je uveden na obrázku 2.2.



Obr. 2.2 Tvar původní příhradové konstrukce a postupná změna tvaru při iteračním geometricky nelineárním výpočtu rovinné příhradové konstrukce z příkladu 2.1 se zadaným modulem pružnosti v tahu a prostém tlaku  $E = 2 \cdot 10^2$  MPa.

Během výpočtu této konstrukce se provede celkem 22 iteračních cyklů. Výsledné délky prutů a vnitřní síly nabývají následujících hodnot:

```
*****
*           Upravené délky prutů a vnitřní síly           *
*           Iterace č.:   22                               *
*****
```

číslo prutu	L [mm]	N [kN]
1	2900.721	-9.928
2	2441.089	-5.891
3	2550.331	5.033
4	2418.897	-8.110
5	2285.700	-21.430
6	3049.638	4.964
7	3137.345	13.735



Srovnání vnitřních sil rovinné příhradové konstrukce z příkladu 2.1 určených lineárním a geometricky nelineárním výpočtem se zadaným modulem pružnosti v tahu a prostém tlaku  $E = 2 \cdot 10^2$  MPa obsahuje následující výpis:

```
*****  
*   Srovnání výsledků lineárního a nelineárního výpočtu   *  
*****
```

číslo prutu	Nlin [kN]	Nnelin [kN]	rozdíl [%]
1	-9.000	-9.928	10.310
2	-5.000	-5.891	17.822
3	5.000	5.033	0.662
4	-5.000	-8.110	62.206
5	-20.000	-21.430	7.150
6	6.000	4.964	-17.270
7	12.000	13.735	14.454



## Kapitola 3

# Stabilita tlačných prutů s využitím principu virtuálních prací



### Cíle

Kapitola je zaměřena na:

- seznámení s numerickým řešením stability štíhlých tlačných prutů,
- teoretického pozadí jednoho ze způsobů určení vzpěrné únosnosti štíhlých tlačných prutů,
- prohloubení znalostí zaměřených na iterační výpočty s využitím zakončovací podmínky a cyklu typu `while`.

Jeden ze způsobů numerického řešení stability štíhlých tlačných prutů je založen na teorii II. řádu a principu virtuálních prací jednotkových sil.

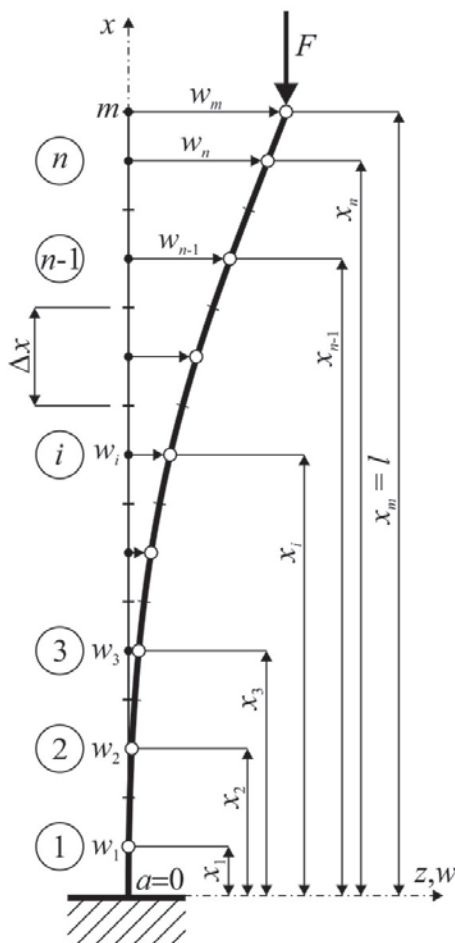
### 3.1 Teoretické pozadí výpočtu

Při výpočtu vzpěrné únosnosti štíhlých tlačných prutů lze určit posunutí obecného bodu  $i$  kolmo k ose prutu pro ohybovou tuhost  $E \cdot I_{\min} = \text{konst.}$  podle vztahu:

$$w_i = \int_0^l \frac{M_F \cdot \bar{M}}{E \cdot I_{\min}} dx = \frac{1}{E \cdot I_{\min}} \cdot \int_0^l M_F \cdot \bar{M} dx = \frac{\Delta x}{E \cdot I_{\min}} \cdot \sum_{i=1}^n (M_{F_i} \cdot M_{i,j}), \quad (3.1)$$

kde  $M_F$  je ohybový moment od skutečného zatížení na zdeformovaném prutu s uvažováním bočního vychýlení podle teorie II. řádu,  $\bar{M}$  je ohybový moment od jednotkového virtuálního zatížení,  $M_{i,j}$  je ohybový moment v bodě  $i$  od jednotkové virtuální síly působící v bodě  $j$ ,  $n$  počet diferencí v prutu o šířce  $\Delta x = \frac{l}{n}$ . Při výpočtu obou veličin  $M_{F_i}$  i  $M_{i,j}$  je nutné respektovat vazby prutu v podporách (oboustranné

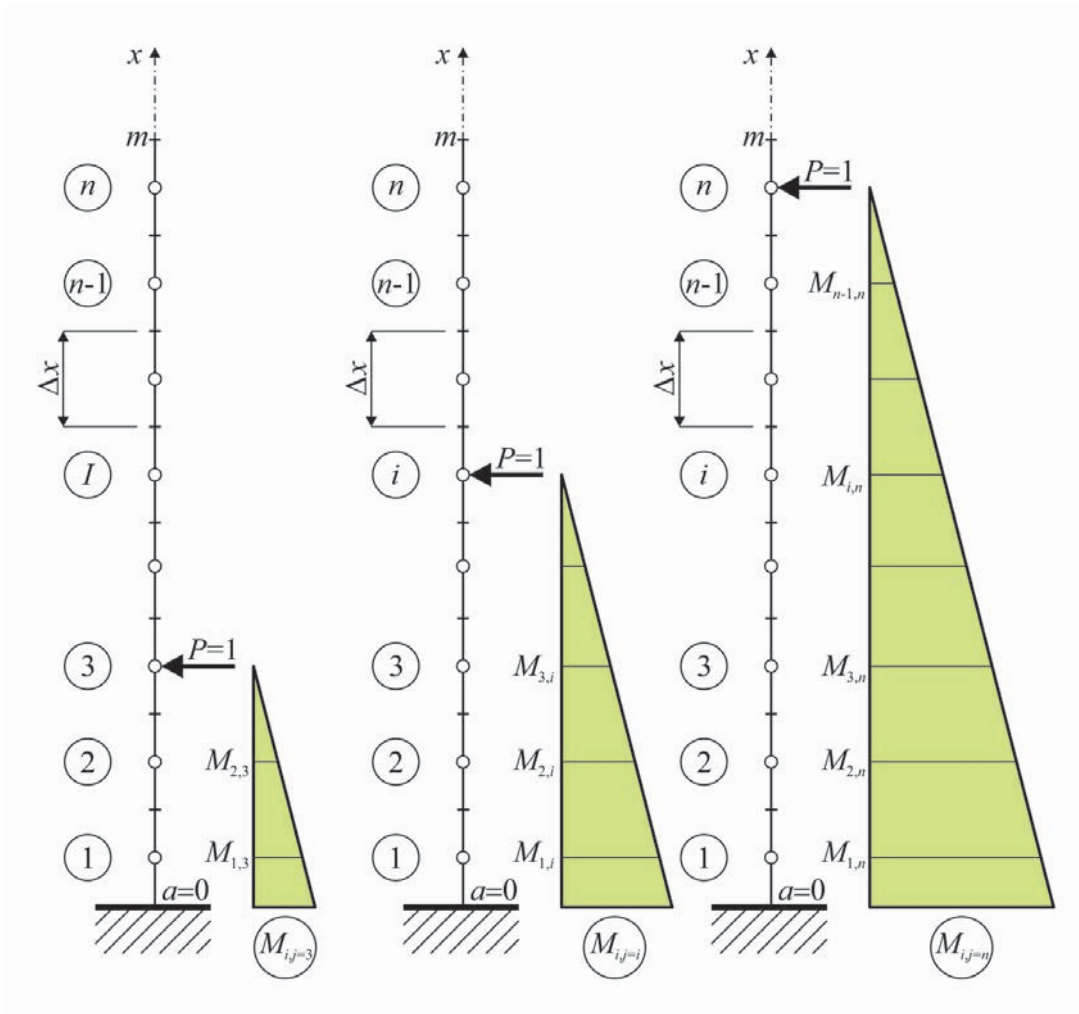
nebo jednostranné vetknutí prutu, kloubové uložení prutu). Jednotlivá posunutí  $w_i$  jsou pro případ tlačенého vetknutého sloupu, který je rozdělen na  $n$  diferencí, schématicky ukázána na obrázku 3.1.



Obr. 3.1 Schéma tlačенého vetknutého sloupu rozděleném na  $n$  diferencí

Ohybový moment  $M_{i,j}$ , který v bodě  $i$  vyjadřuje účinek jednotkové virtuální síly působící v bodě  $j$ , je pro tlačný vetknutý sloup z obrázku 3.1 schématicky znázorněn na obrázku 3.2.

Znalost posunutí  $w_i$  umožňuje určit ohybový moment  $M_{F_i}$ . Vzhledem ke skutečnosti, že celý výpočet probíhá iteračně, je nutné v 1. iteračním cyklu vyvolat počáteční impuls a posunutí  $w_i$  vhodně zvolit. V dalších iteračních cyklech se tato přetvoření budou zpřesňovat a celá úloha by měla konvergovat k ustálenému stavu zdeformované konstrukce a výsledné tlakové síle  $F$  tzv. kritického břemene na základě vztahu:



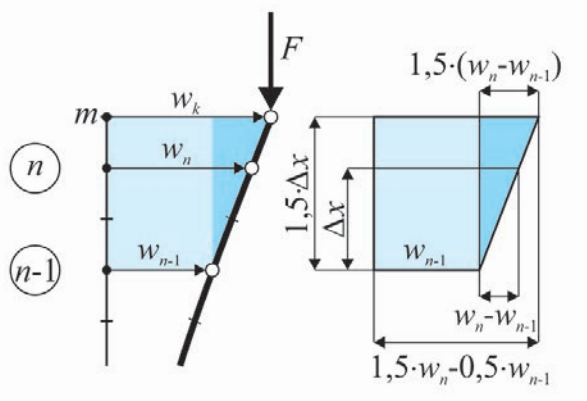
Obr. 3.2 Výpočet ohybového momentu  $M_{i,j}$  od jednotkové virtuální síly působící v bodě  $j$  tláčeného vetknutého sloupu z obrázku 3.1

$$F^k = F^{k-1} \cdot \frac{w_m^{k-1}}{w_m^k}, \quad (3.2)$$

kde  $k$  je číslo iteračního cyklu a  $w_m$  maximální posunutí prutu v příčném směru. Pro případ tláčeného vetknutého sloupu z obrázku 3.1 je tato deformace vztažena k volnému konci sloupu v bodě  $m$  a lze ji určit např. s pomocí podobnosti trojúhelníků a lineární interpolace (vizte obrázek 3.3):

$$\frac{w_n - w_{n-1}}{\Delta x} = \frac{1,5 \cdot (w_n - w_{n-1})}{1,5 \cdot \Delta x}, \quad (3.3)$$

$$w_m = 1,5 \cdot (w_n - w_{n-1}) + w_{n-1} = 1,5 \cdot w_n - 0,5 \cdot w_{n-1}. \quad (3.4)$$



Obr. 3.3 Výpočet přetvoření volného konce u tlačného vetknutého sloupu z obrázku 3.1 pomocí lineární interpolace

Iterační cyklus se ukončí po dosažení předepsané tolerance nepřesnosti  $\varepsilon$  na základě ukončovací podmínky např. ve tvaru:

$$|F^k - F^{k-1}| \leq \varepsilon. \quad (3.5)$$

Výpočet lze provést také u tlačných prutů s proměnlivou hodnotou ohybové tuhosti  $E \cdot I_{\min} \approx (E \cdot I_{\min})_i \neq \text{konst.}$ , přičemž vztah (3.1) je nutno upravit následujícím způsobem:

$$w_i = \int_0^l \frac{M_F \cdot \bar{M}}{(E \cdot I_{\min})_i} dx = \Delta x \cdot \sum_{i=1}^n \frac{M_{F_i} \cdot M_{i,j}}{(E \cdot I_{\min})_i}, \quad (3.6)$$

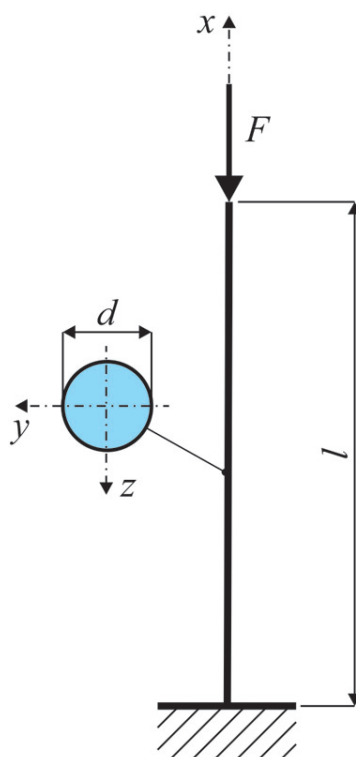
Stabilita štíhlého tlačného prutu může být ovlivněna kromě osové síly také příčným silovým zatížením, tj. kolmo k ose prutu, excentricitou působení osové síly nebo imperfekcemi. Většinu těchto vlivů lze zahrnout do výpočtu prostřednictvím ohybového momentu  $M_F$ .

## 3.2 Aplikace stabilitního řešení tlačných prutů s využitím principu virtuálních prací

Algoritmus numerického řešení stability štíhlých tlačných prutů lze sestavit na základě definovaných vztahů (3.1) až (3.6). Bližší seznámení s jeho funkčností lze uskutečnit s pomocí následujících řešeních příkladů.

### 3.2.1 Příklady na procvičení

**Příklad 3.1.** Stanovte numerickým řešením vzpěrnou únosnost tlačného vetknutého sloupu - vizte schéma na obrázku 3.4, se vstupními údaji: délka  $L = 6$  m,  $E = 2,1 \cdot 10^{11}$  Pa, průřez plný kruhový s průměrem  $d = 10$  cm. Počet diferencí volte  $n = 6, 20, 50, 100, 200, 1000$  a výsledek srovnajte s přesným analytickým řešením podle Eulera. Počáteční zatížení  $F = 100$  kN a povolená tolerance nepřesnosti  $\varepsilon = 0,001$ .



Obr. 3.4 Schéma tlačného vetknutého sloupu s kruhovým průřezem z příkladu 3.1

*Řešení.* Výpočet lze provést s využitím následujícího skriptu (detailní popis zadání je proveden s využitím komentářů za znakem % přímo ve zdrojovém textu):

```
% Příklad 1 - vetknutý sloup
% (součinitel vzpěrné délky Beta=2)
clc; clear; format short;
% Délka sloupu [m]
L=6;
```

```
% Modul pružnosti v tahu a v tlaku E [Pa]
E=2.1e11;
% Průměr kruhového průřezu d [m]
d=0.1;
% Moment setrvačnosti kruhového průřezu
I=pi/64*d^4;
% Výpočet kritického břemene podle Eulera
Fcr=pi^2*E*I/(4*L^2);
% Počet dílků/diferencí
n=6;
% Maximální dovolený počet iterací
it_max=1000;
% Počáteční zatížení [N]
F=1e5; F_old=0;
% Povolená tolerance nepřesnosti
Eps=1e-3;
% Šířka difference
dx=L/n;
% Souřadnice středu diferencí
x(1)=dx/2;
for i=2:n
    x(i)=x(i-1)+dx;
end
% Souřadnice s největší výchylkou - volný konec sloupu
xm=x(n)+dx/2;
% Počáteční výchylka - desetina x(i)
for i=1:n
    w(i)=x(i)/10;
end
wm=xm/10;
xm_old=xm;
% Určení M(i,j) od jednotkového bočního zatížení
for i=1:n
    for j=1:n
        if j>i
            M(i,j)=x(j)-x(i);
        else
            M(i,j)=0;
        end
    end
end
end
% Číslo iteračního cyklu
num_it=0;
```

```

% Výpis tiskové hlavičky
disp(sprintf('\n'))
disp('*****')
disp('* Výpočet kritického břemene pro tlačený vetknutý sloup *')
disp('*****')
disp(' ')
disp(sprintf('Fcr podle Eulera = %12.7f kN',Fcr*1e-3));
disp(' ')
disp(' číslo          F          |F-Fcr|      |F-Fcr|      |F-F_old|')
disp(' iterace        [kN]         [kN]         [%]         [kN]')
disp('-----')
disp(sprintf(' %5d %14.7f %12.4e %8.3f      - ',num_it,F*1e-3,...
    abs(F-Fcr)*1e-3,(F-Fcr)/Fcr*1e2))
while ((abs(F-F_old)>Eps)&(num_it<it_max)) % Iterační cyklus
    num_it=num_it+1;
    % Zálaha vypočtené hodnoty F a wm
    F_old=F;
    wm_old=wm;
    % Ohybový moment od síly F
    for i=1:n
        MF(i)=F*(wm-w(i));
    end
    % Určení horizontální výchylky v daném iteračním cyklu
    for i=1:n
        w(i)=0;
        for j=1:n
            w(i)=w(i)+MF(j)*M(j,i);
        end
        w(i)=w(i)*dx/(E*I);
    end
    % Určení horizontální výchylky na volném konci sloupu
    % s využitím lineární interpolace
    wm=1.5*w(n)-0.5*w(n-1);
    % Určení výsledné kritické síly v daném iteračním cyklu
    F=F*wm_old/wm;
    % Výpis mezivýsledků
    disp(sprintf(' %5d %14.7f %12.4e %8.3f %11.3e',num_it,F*1e-3,...
        abs(F-Fcr)*1e-3,(F-Fcr)/Fcr*1e2,abs(F-F_old)*1e-3))
end
end

```



Výsledky iteračního výpočtu vzpěrné únosnosti tlačného vetknutého sloupu včetně srovnání s přesným analytickým řešením jsou pro výpočet s počtem diferencií  $n = 6$  přehledně zobrazeny v následující tabulce:

```
*****
* Výpočet kritického břemene pro tlačný vetknutý sloup *
*****
```

Fcr podle Eulera = 70.6523232 kN

číslo iterace	F [kN]	F-Fcr  [kN]	F-Fcr  [%]	F-F_old  [kN]
0	100.0000000	2.9348e+01	41.538	-
1	86.8071654	1.6155e+01	22.865	1.319e+01
2	71.4551868	8.0286e-01	1.136	1.535e+01
3	70.7237270	7.1404e-02	0.101	7.315e-01
4	70.6484214	3.9018e-03	-0.006	7.531e-02
5	70.6402249	1.2098e-02	-0.017	8.196e-03
6	70.6393141	1.3009e-02	-0.018	9.108e-04
7	70.6392121	1.3111e-02	-0.019	1.020e-04
8	70.6392006	1.3123e-02	-0.019	1.145e-05
9	70.6391993	1.3124e-02	-0.019	1.287e-06
10	70.6391992	1.3124e-02	-0.019	1.448e-07

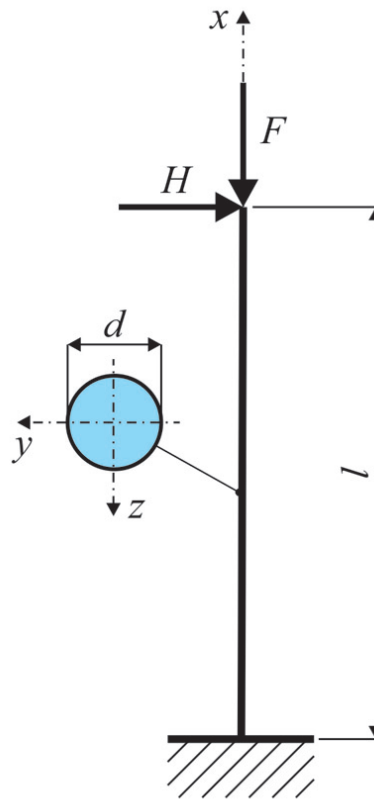


**Příklad 3.2.** Provedte numerický výpočet vzpěrné únosnosti tlačného vetknutého sloupu z příkladu 3.1, který je navíc zatížen horizontální silou  $H = 5$  kN na volném konci sloupu - vizte schéma na obrázku 3.5.



*Řešení.* Výpočet lze provést s využitím následujícího skriptu (detailní popis zadání je proveden s využitím komentářů za znakem % přímo ve zdrojovém textu):

```
% Příklad 2 - vetknutý sloup
% s bočním zatížením na volném konci sloupu
% (součinitel vzpěrné délky Beta=2)
clc; clear; format short;
% Délka sloupu [m]
L=6;
% Modul pružnosti v tahu a v tlaku E [Pa]
E=2.1e11;
% Průměr kruhového průřezu d [m]
d=0.1;
```



Obr. 3.5 Schéma tlačeného vetknutého sloupu s kruhovým průřezem z příkladu 3.2

```

% Moment setrvačnosti kruhového průřezu
I=pi/64*d^4;
% Výpočet kritického břemene podle Eulera
Fcr=pi^2*E*I/(4*L^2);
% Počet dílků/diferencí
n=6;
% Maximální dovolený počet iterací
it_max=1000;
% Boční zatížení H [N]
H=5*1e3;
% Počáteční zatížení [N]
F=100*1e3; F_old=0;
% Povolená tolerance nepřesnosti
Eps=1e-3;
% Šířka difference
dx=L/n;

```

```

% Souřadnice středu diferencí
x(1)=dx/2;
for i=2:n
    x(i)=x(i-1)+dx;
end
% Souřadnice s největší výchylkou - volný konec sloupu
xm=x(n)+dx/2;
% Počáteční výchylka - desetina x(i)
for i=1:n
    w(i)=x(i)/10;
end
wm=xm/10;
wm_old=wm;
% Určení M(i,j) od jednotkového bočního zatížení
for i=1:n
    for j=1:n
        if j>i
            M(i,j)=x(j)-x(i);
        else
            M(i,j)=0;
        end
    end
end
% Číslo iteračního cyklu
num_it=0;
% Výpis tiskové hlavičky
disp(sprintf('\n'))
disp('*****')
disp('* Výpočet kritického břemene pro tlačný vetknutý sloup *')
disp('* s bočním zatížením na volném konci sloupu *')
disp('*****')
disp(' ')
disp(sprintf('Fcr podle Eulera bez bočního zatížení = %12.7f kN',...
    Fcr*1e-3));
disp(' ')
disp(' číslo          F          |F-Fcr|      |F-Fcr|      |F-F_old|')
disp(' iterace         [kN]         [kN]         [%]         [kN]')
disp('-----')
disp(sprintf(' %5d %14.7f %12.4e %8.3f          - ',num_it,F*1e-3,...
    abs(F-Fcr)*1e-3,(F-Fcr)/Fcr*1e2))
while ((abs(F-F_old)>Eps)&(num_it<it_max))
    num_it=num_it+1;
    % Záloha vypočtené hodnoty F a wm

```

```

F_old=F;
wm_old=wm;
% Ohybový moment od síly F
for i=1:n
    MF(i)=F*(wm-w(i))+H*(xm-x(i));
end
% Určení horizontální výchylky v daném iteračním cyklu
for i=1:n
    w(i)=0;
    for j=1:n
        w(i)=w(i)+MF(j)*M(j,i);
    end
    w(i)=w(i)*dx/(E*I);
end
% Určení horizontální výchylky na volném konci sloupu
% s využitím lineární interpolace
wm=1.5*w(n)-0.5*w(n-1);
% Určení výsledné kritické síly v daném iteračním cyklu
F=F*wm_old/wm;
% Výpis mezivýsledků
disp(sprintf(' %5d %14.7f %12.4e %8.3f %11.3e', num_it, F*1e-3, ...
    abs(F-Fcr)*1e-3, (F-Fcr)/Fcr*1e2, abs(F-F_old)*1e-3))
end

```

Výsledky iteračního výpočtu vzpěrné únosnosti tlačného vetknutého sloupu jsou pro výpočet s počtem diferencí  $n = 6$  přehledně zobrazeny v následující tabulce:

```

*****
*   Výpočet kritického břemene pro tlačný vetknutý sloup   *
*       s bočním zatížením na volném konci sloupu         *
*****

```

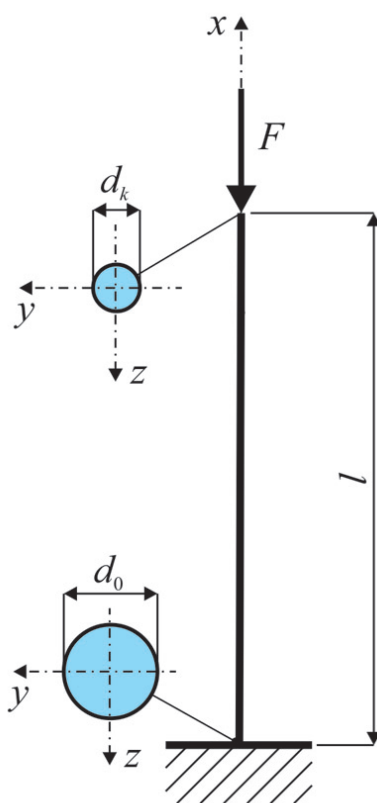
Fcr podle Eulera bez bočního zatížení = 70.6523232 kN

číslo iterace	F [kN]	F-Fcr  [kN]	F-Fcr  [%]	F-F_old  [kN]
0	100.0000000	2.9348e+01	41.538	-
1	57.8714436	1.2781e+01	-18.090	4.213e+01
2	50.6209200	2.0031e+01	-28.352	7.251e+00
3	50.3595237	2.0293e+01	-28.722	2.614e-01
4	50.3396975	2.0313e+01	-28.750	1.983e-02
5	50.3381118	2.0314e+01	-28.752	1.586e-03
6	50.3379825	2.0314e+01	-28.753	1.293e-04

7	50.3379719	2.0314e+01	-28.753	1.061e-05
8	50.3379710	2.0314e+01	-28.753	8.734e-07



**Příklad 3.3.** Proveďte numerický výpočet vzpěrné únosnosti tlačného vetknutého sloupu z příkladu 3.1 s proměnlivým průřezem - vizte schéma na obrázku 3.6, o průměru na volném konci sloupu  $d_k = 5$  cm. Ve vetknutí je průměr sloupu opět  $d_0 = 10$  cm. Změna průměru probíhá po výšce sloupu lineárně.



Obr. 3.6 Schéma tlačného vetknutého sloupu s proměnlivým kruhovým průřezem z příkladu 3.3

*Řešení.* Výpočet lze provést s využitím následujícího skriptu (detailní popis zadání je proveden s využitím komentářů za znakem % přímo ve zdrojovém textu):

```
% Příklad 3 - vetknutý sloup
% proměnlivá průřezová plocha
% (součinitel vzpěrné délky Beta=2)
clc; clear; format short;
% Délka sloupu [m]
L=6;
% Modul pružnosti v tahu a v tlaku E [Pa]
E=2.1e11;
% Průměr kruhového průřezu d ve vetknutí [m]
d0=0.1;
% Průměr kruhového průřezu d na volném konci [m]
dk=0.05;
% Počet dílků/diferencí
n=6;
% Maximální dovolený počet iterací
it_max=1000;
% Počáteční zatížení [N]
F=1e5; F_old=0;
% Povolená tolerance nepřesnosti
Eps=1e-3;
% Šířka difference
dx=L/n;
% Souřadnice středu diferencí
x(1)=dx/2;
for i=2:n
    x(i)=x(i-1)+dx;
end
% Moment setrvačnosti kruhového průřezu
% ve středu každého dílku
for i=1:n
    d=d0-(d0-dk)/L*x(i)
    I(i)=pi/64*d^4;
end
% Souřadnice s největší výchylkou - volný konec sloupu
xm=x(n)+dx/2;
% Počáteční výchylka - desetina x(i)
for i=1:n
    w(i)=x(i)/10;
end
wm=xm/10;
```

```

wm_old=wm;
% Určení M(i,j) od jednotkového bočního zatížení
for i=1:n
    for j=1:n
        if j>i
            M(i,j)=x(j)-x(i);
        else
            M(i,j)=0;
        end
    end
end
% Číslo iteračního cyklu
num_it=0;
% Výpis tiskové hlavičky
disp(sprintf('\n'))
disp('*****')
disp('* Výpočet kritického břemene pro tlačný vetknutý sloup *')
disp('* s proměnlivým průřezem *')
disp('*****')
disp(' ')
disp(' číslo          F          |F-Fcr|')
disp(' iterace          [kN]          [kN]')
disp('-----')
disp(sprintf(' %5d %14.7f          - ',num_it,F*1e-3))
while ((abs(F-F_old)>Eps)&(num_it<it_max))
    num_it=num_it+1;
    % Záloha vypočtené hodnoty F a wm
    F_old=F;
    wm_old=wm;
    % Ohybový moment od síly F
    for i=1:n
        MF(i)=F*(wm-w(i));
    end
    % Určení horizontální výchylky v daném iteračním cyklu
    for j=1:n
        w(j)=0;
        for i=1:n
            w(j)=w(j)+MF(i)*M(i,j)/I(i);
        end
        w(j)=w(j)*dx/E;
    end
    % Určení horizontální výchylky na volném konci sloupu
    % s využitím lineární interpolace

```

```

wm=1.5*w(n)-0.5*w(n-1);
% Určení výsledné kritické síly v daném iteračním cyklu
F=F*wm_old/wm;
% Výpis mezivýsledků
disp(sprintf(' %5d %14.7f %12.4e',num_it,F*1e-3,abs(F-F_old)*1e-3))
end

```

Výsledky iteračního výpočtu vzpěrné únosnosti tlačného vetknutého sloupu proměnlivého průřezu jsou pro výpočet s počtem diferencí  $n = 6$  přehledně zobrazeny v následující tabulce:

```

*****
*   Výpočet kritického břemene pro tlačný vetknutý sloup   *
*                   s proměnlivým průřezem                 *
*****

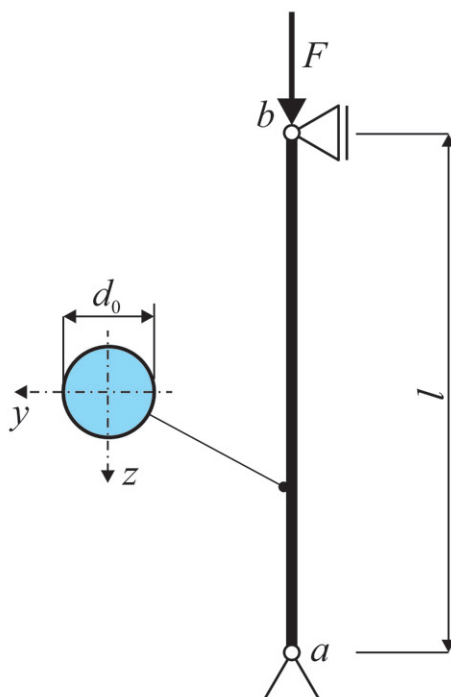
```

číslo iterace	F [kN]	F-F <sub>cr</sub>   [kN]
0	100.0000000	-
1	43.8128039	5.6187e+01
2	31.4553866	1.2357e+01
3	30.5247630	9.3062e-01
4	30.3965134	1.2825e-01
5	30.3771891	1.9324e-02
6	30.3741901	2.9991e-03
7	30.3737199	4.7023e-04
8	30.3736459	7.3998e-05
9	30.3736342	1.1660e-05
10	30.3736324	1.8383e-06
11	30.3736321	2.8987e-07



**Příklad 3.4.** Stanovte numerickým řešením vzpěrnou únosnost prostě podepřeného tlačného prutu - vizte schéma na obrázku 3.7, se vstupními údaji: délka  $L = 6$  m,  $E = 2,1 \cdot 10^{11}$  Pa, průřez plný kruhový s průměrem  $d = 10$  cm. Počet diferencí volte  $n = 6, 20, 50, 100, 200, 1000$  a výsledek srovnajte s přesným analytickým řešením podle Eulera. Počáteční zatížení  $F = 100$  kN a povolená tolerance nepřesnosti  $\varepsilon = 0,001$ .





Obr. 3.7 Schéma prostě podepřeného tlačného prutu z příkladu 3.4

*Řešení.* Výpočet lze provést s využitím následujícího skriptu (detailní popis zadání je proveden s využitím komentářů za znakem % přímo ve zdrojovém textu):

```
% Příklad 4 - prostý nosník
% (součinitel vzpěrné délky Beta=1)
clc; clear; format short;
% Délka nosníku [m]
L=6;
% Modul pružnosti v tahu a v tlaku E [Pa]
E=2.1e11;
% Průměr kruhového průřezu d [m]
d=0.1;
% Moment setrvačnosti kruhového průřezu
I=pi/64*d^4;
% Výpočet kritického břemene podle Eulera
Fcr=pi^2*E*I/(L^2);
% Počet dílků/diferencí
n=6;
% Maximální dovolený počet iterací
it_max=1000;
```

```

% Počáteční zatížení [N]
F=1e5; F_old=0;
% Povolená tolerance nepřesnosti
Eps=1e-3;
% Šířka diference
dx=L/n;
% Souřadnice středu diferencí
x(1)=dx/2;
for i=2:n
    x(i)=x(i-1)+dx;
end
% Souřadnice s největší výchylkou - L/2
if mod(n,2)==0 % je sudé
    xm=(x(n/2)+x(n/2+1))/2;
else % je liché
    xm=x((n+1)/2);
end
% Počáteční výchylka
for i=1:n
    w(i)=0;
end
if mod(n,2)==0 % je sudé
    w(n/2)=0.5;
    w(n/2+1)=0.5;
else % je liché
    w((n+1)/2)=1;
end
% Určení horizontální výchylky uprostřed nosníku
% u sudého počtu dílků s využitím lineární interpolace
if mod(n,2)==0 % je sudé
    wm=1.5*w(n/2)-0.5*w((n/2)-1);
else % je liché
    wm=w((n+1)/2);
end
wm_old=wm;
% Určení M(i,j) od jednotkového bočního zatížení
for i=1:n
    for j=1:n
        M(i,j)=(L-x(j))/L*x(i);
        if x(i)>x(j)
            M(i,j)=M(i,j)-(x(i)-x(j));
        end
    end
end

```

```

end
% Číslo iteračního cyklu
num_it=0;
% Výpis tiskové hlavičky
disp(sprintf('\n'))
disp('*****')
disp('*      Výpočet kritického břemene pro prostý nosník      *')
disp('*****')
disp(' ')
disp(sprintf('Fcr podle Eulera = %12.7f kN',Fcr*1e-3));
disp(' ')
disp(' číslo          F          |F-Fcr|      |F-Fcr|      |F-F_old|')
disp(' iterace         [kN]         [kN]         [%]         [kN]')
disp('-----')
disp(sprintf(' %5d %14.7f %12.4e %8.3f          - ',num_it,F*1e-3,...
    abs(F-Fcr)*1e-3,(F-Fcr)/Fcr*1e2))
while ((abs(F-F_old)>Eps)&(num_it<it_max))
    num_it=num_it+1;
    % Záloha vypočtené hodnoty F a wm
    F_old=F;
    wm_old=wm;
    % Ohybový moment od síly F
    for i=1:n
        MF(i)=F*w(i);
    end
    % Určení horizontální výchylky v daném iteračním cyklu
    for i=1:n
        w(i)=0;
        for j=1:n
            w(i)=w(i)+MF(j)*M(j,i);
        end
        w(i)=w(i)*dx/(E*I);
    end
    % Určení horizontální výchylky uprostřed nosníku
    % u sudého počtu dílků s využitím lineární interpolace
    if mod(n,2)==0 % je sudé
        wm=1.5*w(n/2)-0.5*w((n/2)-1);
    else % je liché
        wm=w((n+1)/2);
    end
    % Určení výsledné kritické síly v daném iteračním cyklu
    F=F*wm_old/wm;
    % Výpis mezivýsledků

```

```

disp(sprintf(' %5d %14.7f %12.4e %8.3f %11.3e', num_it, F*1e-3, ...
            abs(F-Fcr)*1e-3, (F-Fcr)/Fcr*1e2, abs(F-F_old)*1e-3))
end

```

Výsledky iteračního výpočtu vzpěrné únosnosti tlačeného prostě podepřeného prutu včetně srovnání s přesným analytickým řešením jsou pro výpočet s počtem diferencí  $n = 6$  přehledně zobrazeny v následující tabulce:

```

*****
*       Výpočet kritického břemene pro prostý nosník       *
*****

```

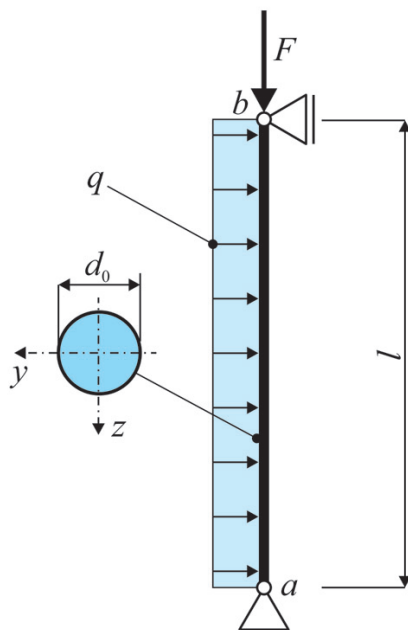
Fcr podle Eulera = 282.6092927 kN

číslo iterace	F [kN]	F-Fcr  [kN]	F-Fcr  [%]	F-F_old  [kN]
0	100.0000000	1.8261e+02	-64.615	-
1	515.4175447	2.3281e+02	82.378	4.154e+02
2	309.2505268	2.6641e+01	9.427	2.062e+02
3	280.4993441	2.1099e+00	-0.747	2.875e+01
4	276.7721610	5.8371e+00	-2.065	3.727e+00
5	276.2855031	6.3238e+00	-2.238	4.867e-01
6	276.2212772	6.3880e+00	-2.260	6.423e-02
7	276.2127435	6.3965e+00	-2.263	8.534e-03
8	276.2116054	6.3977e+00	-2.264	1.138e-03
9	276.2114533	6.3978e+00	-2.264	1.521e-04
10	276.2114329	6.3979e+00	-2.264	2.035e-05
11	276.2114302	6.3979e+00	-2.264	2.725e-06
12	276.2114298	6.3979e+00	-2.264	3.649e-07



**Příklad 3.5.** Proveďte numerický výpočet vzpěrné únosnosti tlačeného prostě podepřeného prutu z příkladu 3.4 se spojitým bočním zatížením  $q = 12$  kN/m - vizte schéma na obrázku 3.8.

*Řešení.* Výsledky iteračního výpočtu vzpěrné únosnosti tlačeného prostě podepřeného prutu včetně srovnání s přesným analytickým řešením (bez počního zatížení) jsou pro výpočet s počtem diferencí  $n = 999$  a pro počáteční hodnotu tlakové síly  $F = 300$  kN přehledně zobrazeny v následující tabulce:



Obr. 3.8 Schéma prostě podepřeného tlačného prutu z příkladu 3.5 se spojitým bočním zatížením

```
*****
*      Výpočet kritického břemene pro prostý nosník      *
*****
```

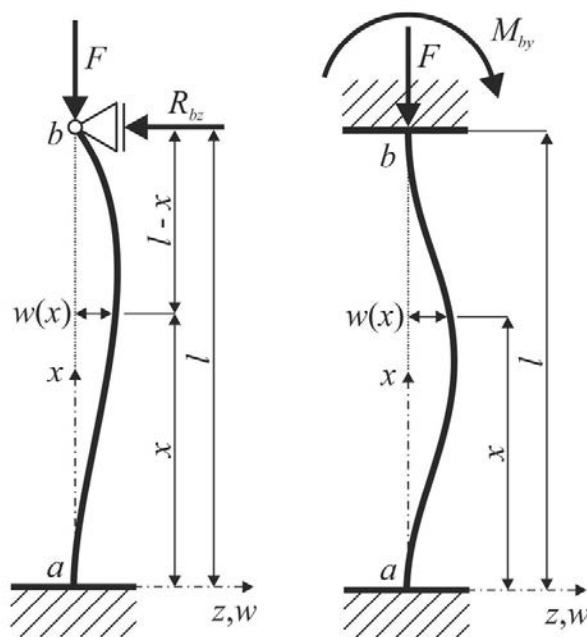
Fcr podle Eulera = 282.6092927 kN

číslo iterace	F [kN]	F-Fcr  [kN]	F-Fcr  [%]	F-F_old  [kN]
0	300.0000000	1.7391e+01	6.154	-
1	1507.0478027	1.2244e+03	433.262	1.207e+03
2	238.2711550	4.4338e+01	-15.689	1.269e+03
3	238.3281537	4.4281e+01	-15.669	5.700e-02
4	238.3556860	4.4254e+01	-15.659	2.753e-02
5	238.3586067	4.4251e+01	-15.658	2.921e-03
6	238.3588888	4.4250e+01	-15.658	2.821e-04
7	238.3589154	4.4250e+01	-15.658	2.660e-05
8	238.3589179	4.4250e+01	-15.658	2.489e-06
9	238.3589181	4.4250e+01	-15.658	2.323e-07





**Pro zájemce:** Provedte numerický výpočet vzpěrné únosnosti také pro ostatní dva případy Eulerových tlačených prutů (prut jednostranně vetknutý s posuvnou kloubovou vazbou na druhém konci a oboustranně vetknutý tlačený prut) - vizte schéma na obrázku 3.9.



Obr. 3.9 Schéma tlačeného prutu jednostranně vetknutého s posuvnou kloubovou vazbou na druhém konci a tlačeného prutu oboustranně vetknutého

## Kapitola 4

# Vlastní čísla a vlastní vektory

### Cíle



Kapitola je zaměřena na:

- seznámení s pojmy vlastní čísla a vlastní vektory,
- teoretické pozadí některých způsobů jejich výpočtu,
- ukázkou využití vlastních čísel a vektorů v inženýrské praxi.

### 4.1 Úvod

Vlastní vektor matice  $[A]$  je takový nenulový vektor  $\{u\}$ , pro který platí:

$$[A] \cdot \{u\} = \lambda \cdot \{u\}, \quad (4.1)$$

kde  $\lambda$  je vlastní (charakteristické) číslo matice  $[A]$  a  $\{u\}$  vlastní (charakteristický) vektor matice  $[A]$  příslušný vlastnímu číslu  $\lambda$ .

**Poznámka:** Množina všech vlastních čísel matice  $[A]$  se označuje jako spektrum matice  $[A]$ .



**Příklad 4.1.** Určete vlastní číslo  $\lambda$  matice  $[A] = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix}$  a vektoru  $\{u\} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}^T$ .



*Řešení.* Řešení vychází z rovnice (4.1), kdy po dosazení lze získat:

$$[A] \cdot \{u\} = \begin{bmatrix} 4 & -2 \\ 1 & 1 \end{bmatrix} \cdot \begin{Bmatrix} 2 \\ 1 \end{Bmatrix} = \begin{Bmatrix} 6 \\ 3 \end{Bmatrix} = 3 \cdot \begin{Bmatrix} 2 \\ 1 \end{Bmatrix} = 3 \cdot \{u\}. \quad (4.2)$$

Vlastním číslem matice  $[A]$  je tedy  $\lambda = 3$ . Vektor  $\{u\} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}^T$  je pak vlastní vektor matice  $[A]$  příslušný k vlastnímu číslu  $\lambda = 3$ . ▲

Rovnici (4.1) lze zapsat rovněž ve tvaru:

$$([A] - \lambda \cdot [E]) \cdot \{u\} = 0, \quad (4.3)$$

kde  $[E]$  je jednotková matice velikosti  $n$  (čtvercová matice typu  $n \times n$ , která má na hlavní diagonále jedničky a na ostatních místech nuly).

Rovnice (4.3) představuje soustavu homogenních rovnic:

$$\begin{aligned} (a_{1,1} - \lambda) \cdot x_1 + a_{1,2} \cdot x_2 + \cdots + a_{1,n} \cdot x_n &= 0 \\ a_{2,1} \cdot x_1 + (a_{2,2} - \lambda) \cdot x_2 + \cdots + a_{2,n} \cdot x_n &= 0 \\ \vdots & \\ a_{n,1} \cdot x_1 + a_{n,2} \cdot x_2 + \cdots + (a_{n,n} - \lambda) \cdot x_n &= 0 \end{aligned} \quad (4.4)$$

která má netriviální řešení pro

$$\det([A] - \lambda \cdot [E]) = 0, \quad (4.5)$$

což představuje tzv. *charakteristickou rovnici* matice  $[A]$ . Určením determinantu lze získat tzv. *charakteristický polynom*  $p(\lambda)$  stupně  $n$ . Řešením rovnice  $p(\lambda) = 0$  je pak celkem  $n$  vlastních čísel matice  $[A]$ .



**Poznámka:** Čtvercová diagonální matice  $\begin{bmatrix} \lambda_1 & 0 & 0 & 0 \\ 0 & \lambda_2 & 0 & 0 \\ \vdots & & \ddots & \\ 0 & 0 & 0 & \lambda_n \end{bmatrix}$  se označuje jako tzv. *spektrální matice* matice  $[A]$ .



**Příklad 4.2.** Určete pro matici matice  $[A] = \begin{bmatrix} 3 & -1 \\ 2 & 0 \end{bmatrix}$  vlastní čísla  $\lambda_{1,2}$  a vlastní vektory  $\{u_{1,2}\}$ .

*Řešení.* Řešení vychází z charakteristické rovnice (4.5), kdy po dosazení lze získat:

$$\det([A] - \lambda \cdot [E]) = \begin{bmatrix} 3 - \lambda & -1 \\ 2 & -\lambda \end{bmatrix} = 0. \quad (4.6)$$

Výpočet determinantu této matice vede ke kvadratické rovnici:

$$(3 - \lambda) \cdot (-\lambda) - (-1) \cdot 2 = \lambda^2 - 3 \cdot \lambda + 2 = 0, \quad (4.7)$$

jejíž řešením jsou dva kořeny - vlastní čísla  $\lambda_1 = 2$  a  $\lambda_2 = 1$ .

Vlastní vektor  $u_1$  příslušný hodnotě vlastního čísla  $\lambda_1 = 2$  lze určit řešením soustavy lineárních rovnic:

$$\begin{bmatrix} 3 - \lambda_1 & -1 \\ 2 & -\lambda_1 \end{bmatrix} \cdot u_1 = \begin{bmatrix} 1 & -1 \\ 2 & -2 \end{bmatrix} \cdot u_1 = 0. \quad (4.8)$$



Řešením soustavy lineárních rovnic (4.8) je např. vektor  $u_1 = \{ 2 \ 2 \}^T$ .

Vlastní vektor  $u_2$  příslušný hodnotě vlastního čísla  $\lambda_2 = 1$  pak lze určit obdobně řešením soustavy lineárních rovnic:

$$\begin{bmatrix} 3 - \lambda_2 & -1 \\ 2 & -\lambda_2 \end{bmatrix} \cdot u_2 = \begin{bmatrix} 2 & -1 \\ 2 & -1 \end{bmatrix} \cdot u_2 = 0. \quad (4.9)$$

Řešením soustavy lineárních rovnic (4.9) je např. vektor  $u_2 = \{ 1 \ 2 \}^T$ .



**Příklad 4.3.** Určete vlastní čísla a vlastní vektory pro matici  $[A]$  z příkladu 4.2 s využitím příkazů systému MATLAB.



*Řešení.* V případě potřeby lze určit vlastní čísla a vlastní vektory matice pomocí funkce `eig`:

```
[V,L]=eig(A);
diag(L)
V
```

kde  $L$  je spektrální matice, ve které jsou vlastní čísla na diagonále, příkaz `diag(L)` vypíše vektor s vlastními čísly a `V` vypíše matici, kde jsou jednotlivé vlastní vektory uloženy po sloupcích. Správnost výpočtu vlastních čísel a vlastních vektorů lze zkontrolovat na základě vztahu (4.3) např. příkazem:

```
A*V-V*L
```

V programovém systému MATLAB lze také určit koeficienty charakteristického polynomu a následně i jeho řešení následující posloupností příkazů:

```
c=poly(A)
l=roots(c)
```

Řešení lze opět zkontrolovat např.:

```
trace(A)-sum(l)
det(A)-l(1)*l(2)
```

nebo na základě vztahu (4.3):

```
det(A-l(1)*eye(size(A)))
det(A-l(2)*eye(size(A)))
```





**Příklad 4.4.** Určete vlastní čísla a vlastní vektory pro následující matice:  $[A_1] =$   
 $= \begin{bmatrix} 3 & 2 \\ 4 & 1 \end{bmatrix}$ ,  $[A_2] = \begin{bmatrix} 2 & 0 & 0 \\ 2 & 2 & 1 \\ 1 & 1 & 2 \end{bmatrix}$ ,  $[A_3] = \begin{bmatrix} 2 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$  a  $[A_4] = \begin{bmatrix} 3 & 0 & 0 & 0 \\ 4 & 1 & 0 & 0 \\ 0 & 0 & 2 & 1 \\ 0 & 0 & 0 & 2 \end{bmatrix}$ .

## 4.2 Numerické metody pro řešení vlastních čísel

Matematické úlohy související s nalezením vlastních čísel lze rozdělit do dvou skupin:

- *částečný problém vlastních čísel*, kdy se v úloze hledají pouze některá vlastní čísla - obvykle s nejmenší nebo největší absolutní hodnotou,
- *úplný problém vlastních čísel*, kdy se v úloze určují všechna vlastní čísla,

### 4.2.1 Částečný problém vlastních čísel

V případě částečného problému vlastních čísel se určuje pouze některé vlastní číslo. Většinou je řešením tzv. dominantní vlastní číslo s největší hodnotou.

#### Mocninná metoda

Zde je uveden skript, který využívá Mocninnou metodu (originální označení je *Power Iteration*) pro určení dominantního vlastního čísla čtvercové matice:

```
function [x,lam]=powerit(A,x,k)
% Vstup: matice A, počáteční (nenulový) vektor x, počet kroků cyklu k
% Výstup: dominantní vlastní číslo lam, příslušný vlastní vektor x
for j=1:k
    u=x/norm(x);           % normalizace vektoru
    x=A*u;                 % mocninný krok
    lam=u'*x;              % Rayleighův aproximační koeficient
end
```



**Příklad 4.5.** Určete Mocninnou metodou hodnotu dominantního vlastního čísla matice  $[A] = \begin{bmatrix} 1 & 2 & 0 \\ 0 & 0 & 1 \\ 3 & 6 & 2 \end{bmatrix}$ . Počáteční vektor  $x$  zvolte  $\{x\} = \{1, 1, 1\}^T$  a  $k = 10$ .

Výsledek zkontrolujte postupem z příkladu 4.3.

### Inverzní mocninná metoda

Modifikovaná Mocninná metoda pro určení vlastního čísla, které je nejbližší zadané hodnotě  $s$ , se nazývá *Inverzní mocninná metoda* (originální označení je *Inverse Power Iteration*). Následuje skript pro řešení touto metodou:

```
function [x,lam]=invpowerit(A,x,s,k)
% Vstup: matice A, počáteční (nenulový) vektor x
% tzv. posun s, počet kroků cyklu k
% Výstup: vlastní číslo lam, dominantní vlastní vektor inv(A-sI)
As=A-s*eye(size(A));
for j=1:k
    u=x/norm(x);           % normalizace
    x=As\u;                % inverzní mocninný krok
    lam=u'*x;              % Rayleighův aproximační koeficient
end
lam=1/lam+s;
```

**Příklad 4.6.** Určete Inverzní mocninnou metodou hodnotu vlastního čísla matice  $[A]$  z příkladu 4.5, které leží nejbližší hodnotě  $s = -2$ . Výsledek zkontrolujte postupem z příkladu 4.3.



### Metoda Rayleighova podílu

Modifikací mocninné metody je metoda Rayleighova podílu (Rayleigh Quotient Iteration), která je implementována v následujícím skriptu:

```
function [x,lam]=rqi(A,x,k)
% Vstup: matice A, počáteční (nenulový) vektor x, počet kroků cyklu k
% Výstup: dominantní vlastní číslo lam, příslušný vlastní vektor x
for j=1:k
    u=x/norm(x);           % normalizace
    lam=u'*A*u;            % Rayleighův koeficient
    x=(A-lam*eye(size(A)))\u; % inverzní mocninná iterace
end
x=x/norm(x);
lam=x'*A*x;                % Rayleighův koeficient
```

**Příklad 4.7.** Určete metodou Rayleighova podílu hodnotu dominantního vlastního

čísla matice  $[A] = \begin{bmatrix} 4 & 1 & 0 \\ 1 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix}$ . Počáteční vektor  $x$  zvolte  $\{x\} = \{1, 1, 1\}^T$  a  $k = 10$ .

Výsledek zkontrolujte postupem z příkladu 4.3.



### 4.2.2 Úplný problém vlastních čísel

Metody na řešení úplného problému se dělí do tří základních skupin:

1. Metody založené na výpočtech vlastních čísel pomocí charakteristického polynomu. Jsou nevýhodné pro řešení matic vyššího řádu  $n$ , protože v takových případech je obtížné určit kořeny charakteristického polynomu pomocí determinantu.
2. Metody využívající podobnosti matic, které jsou založeny na principu stejných vlastních čísel pro podobné matice. Patří zde např. *metoda LU-rozkladu* nebo *metoda ortogonálních transformací*.
3. Smíšené metody založené na převodu obecné matice na matici třídiagonální (patří zde např. (např. metoda Givensova, Householderova nebo Lanczosova) a následný efektivní výpočet kořenů charakteristického polynomu této upravené matice.

#### Metoda simultánní iterace

Metoda simultánní iterace (v originále Normalized Simultaneous Iteration) je implementována v následujícím skriptu:

```
function [Q,lam]=nsi(A,k)
% Vstup: matice A, počet iteračních kroků k
% Výstup: vlastní čísla ve vektoru lam a matice vlastních vektorů Q
[m,n]=size(A);
Q = eye(m,m);
for j = 1:k
    [Q,R] = qr(A*Q); % QR faktorizace
end
lam=diag(Q'*A*Q); % Rayleighův koeficient
```



**Příklad 4.8.** Určete metodou simultánní iterace hodnoty všech vlastních čísel matice  $[A] = \begin{bmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{bmatrix}$  a příslušné vlastní vektory. Výsledek zkontrolujte postupem z příkladu 4.3.

### Neposuvný QR algoritmus

Neposuvný QR algoritmus (v originále Unshifted QR Algorithm) je implementován v následujícím skriptu:

```
function [Qbar,lam]=unshiftedqr(A,k)
% Vstup: matice A, počet iteračních kroků k
% Výstup: vlastní čísla ve vektoru lam a
% matice vlastních vektorů Qbar
[m,n]=size(A);
Q=eye(m,m);
Qbar=Q; R=A;
for j = 1:k
    [Q,R] = qr(R*Q); % QR factorizace
    Qbar=Qbar*Q;    % akumulace Q'
end
lam=diag(R*Q);    % diagonální konvergence k vlastním číslům
```

**Příklad 4.9.** Určete s pomocí neposuvného QR algoritmu hodnoty všech vlastních čísel matice  $[A]$  z příkladu 4.8 a příslušné vlastní vektory. Výsledek zkontrolujte postupem z příkladu 4.3.



**Příklad 4.10.** Všechny popisované algoritmy pro výpočet vlastních čísel a vlastních vektorů pracovaly s využitím cyklu typu `for`. Upravte tyto algoritmy tak, aby využívaly cyklus typu `while` s vhodnou zakončovací podmínkou se zadanou tolerancí nepřesnosti  $\varepsilon$ .



### Posuvný QR algoritmus

Posuvný QR algoritmus (v originále Shifted QR Algorithm) určuje reálná a komplexní vlastní čísla čtvercové matice a je implementován v následujícím skriptu s využitím cyklu typu `while`:

```
% Vstup: matice A
% Výstup: vlastní čísla ve vektoru lam
function lam=shiftedqr(a)
tol=1e-14;
kounttol=500;
m=size(a,1);
lam=zeros(m,1);
n=m;
while n>1
    kount=0;
```

```

while max(abs(a(n,1:n-1)))>tol&kount<kounttol
    kount=kount+1;    % sleduje počet qr's
    mu=a(n,n);        % posun je mu
    [q,r]=qr(a-mu*eye(n));
    a=r*q+mu*eye(n);
end
if kount<kounttol    % byl izolován blok 1x1
    lam(n)=a(n,n);    % určení vlastního čísla
    n=n-1;
    a=a(1:n,1:n);    % zmenšení řádu matice o 1
else                % byl izolován blok 2x2
    disc=(a(n-1,n-1)-a(n,n))^2+4*a(n,n-1)*a(n-1,n);
    lam(n)=(a(n-1,n-1)+a(n,n)+sqrt(disc))/2;
    lam(n-1)=(a(n-1,n-1)+a(n,n)-sqrt(disc))/2;
    n=n-2;
    a=a(1:n,1:n);    % zmenšení řádu matice o 2
end
end
if n>0;lam(1)=a(1,1);end % zbyl jen blok 1x1

```



**Příklad 4.11.** Určete s pomocí posuvného QR algoritmu hodnoty všech vlastních čísel matice  $[A]$  z příkladu 4.8. Výsledek zkontrolujte postupem z příkladu 4.3.

## 4.3 Praktické využití vlastních čísel a vlastních vektorů v úlohách stavební mechaniky

Vlastní čísla a jejich příslušné vektory mají své uplatnění při řešení technických problémů včetně úloh stavební mechaniky. V následujícím výkladu bude stručně popsána aplikace metody inverzních iterací při řešení stability prutových konstrukcí a naznačeno obdobné uplatnění v dynamice stavebních konstrukcí při řešení vlastních frekvencí prutové konstrukce.

### 4.3.1 Metoda inverzních iterací

Metoda slouží k výpočtu vlastních čísel a příslušných vlastních vektorů svazku matic  $\mathbf{K}$  a  $\mathbf{M}$ . V úlohách stavební mechaniky matice  $\mathbf{K}$  obvykle představuje matici tuhosti konstrukce, zatímco matice  $\mathbf{M}$  může být buď geometrickou maticí (v úlohách stability konstrukcí) nebo maticí hmotnosti v dynamice (při výpočtu vlastních frekvencí).

V případě netlumeného kmitání platí:

$$\mathbf{M} \ddot{\mathbf{u}} + \mathbf{K} \mathbf{u} = \mathbf{0} . \quad (4.10)$$

Řešení rovnice (4.10) lze hledat ve tvaru:

$$\mathbf{u}(\mathbf{t}) = \mathbf{y} \sin \omega t . \quad (4.11)$$

Určením  $\ddot{\mathbf{u}}$  a dosazením do (4.10) lze získat rovnici pro výpočet netlumeného kmitání:

$$(-\omega^2 \mathbf{M} + \mathbf{K}) \mathbf{y} = \mathbf{0} . \quad (4.12)$$

Rovnice (4.12) se může přepsat do tvaru:

$$\mathbf{K} \mathbf{y} = \omega^2 \mathbf{M} \mathbf{y} . \quad (4.13)$$

Rovnice (4.13) je splněna pokud  $\omega$  je vlastní frekvencí konstrukce a  $\mathbf{y}$  je příslušným vlastním tvarem konstrukce. Protože tyto veličiny nejsou předem známy a je potřebné je vypočítat, místo  $\mathbf{y}$  se může použít aproximace  $\mathbf{x}_i$ .

**Poznámka 4.12.** Při volbě aproximací  $\mathbf{x}_i$  je nutno dbát v úvahu také skutečnost, že zvolené aproximace musí rovněž splňovat okrajové podmínky úlohy.

Nyní je možné určit amplitudy setrvačných sil:

$$\mathbf{S} = \mathbf{M} \mathbf{x}_i . \quad (4.14)$$

Následuje výpočet vektoru posunutí  $\mathbf{x}_{i+1}$  pomocí:

$$\mathbf{K} \mathbf{x}_{i+1} = \mathbf{S} = \mathbf{M} \mathbf{x}_i . \quad (4.15)$$

Proces (4.15) je možné opakovat tak dlouho, až je rozdíl mezi  $\mathbf{x}_i$  a  $\mathbf{x}_{i+1}$  dostatečně malý. Pak je možné předpokládat, že byla nalezena aproximace vektoru výchylek  $\mathbf{y}$  s požadovanou přesností.

Samotnou hodnotu vlastní kruhové frekvence  $\omega$  je možné najít pomocí vztahu pro Rayleighův koeficient:

$$\omega_2 = \rho(i+1) = \frac{\mathbf{x}_{i+1}^T \mathbf{K} \mathbf{x}_{i+1}}{\mathbf{x}_{i+1}^T \mathbf{M} \mathbf{x}_{i+1}} . \quad (4.16)$$

### 4.3.2 Stabilita konstrukcí

Stejným způsobem je možné řešit úlohy stability konstrukcí – jen matice  $\mathbf{M}$  bude nahrazena geometrickou maticí tuhosti  $\mathbf{K}_G$  podle [2] a koeficient  $\rho$  bude mít význam násobitele zatížení. Velikost kritických zatížení  $\mathbf{F}_{cr}$  lze získat výpočtem:

$$\mathbf{F}_{cr} = \omega \mathbf{F} , \quad (4.17)$$

kde  $\mathbf{F}$  je vektor zadaných zatížení.

### 4.3.3 Postup výpočtu

Pro praktický výpočet je potřebné hodnoty  $\mathbf{x}_i$  normovat, aby nebyly členy tohoto vektoru příliš malé. Nejvhodnější je normování pomocí matice tuhosti:

$$\mathbf{x}_i = \frac{\mathbf{x}_i}{\sqrt{\mathbf{x}_i^T \mathbf{M} \mathbf{x}_i}}. \quad (4.18)$$

### 4.3.4 Počáteční aproximace

Počáteční vektor neznámých výchylek  $\mathbf{x}_1$  lze zvolit libovolně. Musí však plnit stejné okrajové podmínky, jako pro vektor výchylek konstrukce.

V dalším výpočtu se výpočet opakuje pro  $i = 1 \dots n$ :

$$\mathbf{K} \mathbf{x}_{i+1} = \mathbf{M} \mathbf{x}_i, \quad (4.19)$$

$$\mathbf{x}_{i+1} = \frac{\mathbf{x}_{i+1}}{\sqrt{\mathbf{x}_{i+1}^T \mathbf{M} \mathbf{x}_{i+1}}}. \quad (4.20)$$

$$\rho(i+1) = \frac{\mathbf{x}_{i+1}^T \mathbf{K} \mathbf{x}_{i+1}}{\mathbf{x}_{i+1}^T \mathbf{M} \mathbf{x}_{i+1}}. \quad (4.21)$$

Nakonec se provede test konvergence:

$$\frac{|\rho_{k+1} - \rho_k|}{|\rho_{k+1}|} \leq 10^{-2s}, \quad (4.22)$$

kde hodnota  $-2s$  vyjadřuje míru požadované přesnosti (doporučuje se volit  $s \leq 4$ ).

Pokud je rovnice (4.22) splněna, může být výpočet ukončen a  $\rho_{i+1}$  je hledaným vlastním číslem svazku matic (a tedy  $\omega = \sqrt{r h o_{i+1}}$  je hledanou vlastní kruhovou frekvencí). Potom je možno také vypočítat vektor výchylek  $\mathbf{y}_{i+1}$  odpovídající vlastnímu číslu  $\rho_{i+1}$ :

$$\mathbf{y}_{k+1} = \mathbf{M} \mathbf{x}_{i+1}. \quad (4.23)$$

### 4.3.5 Aplikace řešené úlohy

Celý problém lze aplikovat prostřednictvím následujícího skriptu:

```
% Lineární stabilita: Eulerův prostě uložený nosník
% Autor: doc. Ing. Jiří Brožovský, Ph.D.
clear; clc;
% Dimenze úlohy (2 stupně volnosti: u,v; 2 uzly na prutu):
ndof=3;
puzlu=2;
% Vstupní údaje: materiálové a průřezové charakteristiky
```



```

E=2.10e11;
% Průměr kruhového průřezu d [m]
d=0.1;
% Moment setrvačnosti kruhového průřezu
I=pi/64*d^4;
A=pi/4*d^2;
% Vstupní údaje: geometrie konstrukce
L=6;
uzly=[0 0; 0.25 0; 0.5 0; 0.75 0;
1.0 0; 1.25 0; 1.5 0; 1.75 0; 2.0 0];
uzly=L*uzly;
pruty=[1 2; 2 3; 3 4; 4 5; 5 6; 6 7; 7 8; 8 9];
nuzlu=size(uzly,1);
nprutu=size(pruty,1);
% Vstupní údaje: podpory (uzel, směr, velikost):
podpory=[1 1 0; 1 2 0; 9 2 0];
npodpor=size(podpory,1);
% Vstupní údaje: zatížení - síly (uzel, směr, velikost):
sily=[9 1 -1000];
nsil=size(sily,1);
% Výpočet
% Výpočet kódových čísel
kcis=zeros(nprutu,ndof*puzlu);
for i=1:nprutu
    for j=1:puzlu
        for k=1:ndof
            kcis(i,((j-1)*ndof+k))=(pruty(i,j)-1)*ndof+k;
        end
    end
end
% Nulování matic a vektorů
velikost = nuzlu*ndof;
K = zeros(velikost);
M = zeros(velikost);
u = zeros(velikost,1);
F = zeros(velikost,1);
% Sestavení a lokalizace matic tuhosti
for i=1:nprutu
    Keg=zeros(puzlu*ndof);
    % Délka prutu
    dx2 = (uzly(pruty(i,1),1)-uzly(pruty(i,2),1))^2;
    dy2 = (uzly(pruty(i,1),2)-uzly(pruty(i,2),2))^2;
    L = sqrt(dx2 + dy2);

```

```

% Matice tuhosti
Keg=[E*A/L 0 0 -E*A/L 0 0;
0 12*E*I/L^3 6*E*I/L^2 0 -12*E*I/L^3 6*E*I/L^2;
0 6*E*I/L^2 4*E*I/L 0 -6*E*I/L^2 2*E*I/L;
-E*A/L 0 0 E*A/L 0 0;
0 -12*E*I/L^3 -6*E*I/L^2 0 12*E*I/L^3 -6*E*I/L^2;
0 6*E*I/L^2 2*E*I/L 0 -6*E*I/L^2 4*E*I/L];
% Transformace
s=(uzly(pruty(i,2),2)-uzly(pruty(i,1),2))/L;
c=(uzly(pruty(i,2),1)-uzly(pruty(i,1),1))/L;
T=[ c  s  0  0  0  0;
   -s  c  0  0  0  0;
    0  0  1  0  0  0;
    0  0  0  c  s  0;
    0  0  0 -s  c  0;
    0  0  0  0  0  1];
Ke=T'*Keg*T;
% Lokalizace
for j=1:(puzlu*ndof)
    for k=1:(puzlu*ndof)
        K(kcis(i,j),kcis(i,k))=K(kcis(i,j),kcis(i,k))+Ke(j,k);
    end
end
end
% Zatížení
for i=1:nsil
    iuz=sily(i,1);
    ismer=sily(i,2);
    pos=(ndof*(iuz-1))+ismer;
    F(pos) = F(pos)+sily(i,3);
end
% Podpory
for i=1:npodpor
    iuz=podpory(i,1);
    ismer=podpory(i,2);
    pos=(ndof*(iuz-1))+ismer;
    u(pos)=u(pos)+podpory(i,3);
    for j=1:velikost
        K(pos,j)=0.0;
        K(j,pos)=0.0;
    end
    K(pos,pos)=1.0;
end
end

```

```

% Řešení soustavy rovnic
u=K\F;
% Výpočet výsledků na prutech
for i=1:nprutu
    ue=zeros(puzlu*ndof,1);
    uel=zeros(puzlu*ndof,1);
    % Získání lokálních vektorů deformací
    for j=1:(puzlu*ndof)
ue(j)=u(kcis(i,j));
    end
    % Transformace
    dx2=(uzly(pruty(i,1),1)-uzly(pruty(i,2),1))^2;
    dy2=(uzly(pruty(i,1),2)-uzly(pruty(i,2),2))^2;
    L=sqrt(dx2 + dy2);
    s=(uzly(pruty(i,2),2)-uzly(pruty(i,1),2))/L;
    c=(uzly(pruty(i,2),1)-uzly(pruty(i,1),1))/L;
    T=[c s 0 0 0 0;
    -s c 0 0 0 0;
    0 0 1 0 0 0;
    0 0 0 c s 0;
    0 0 0 -s c 0;
    0 0 0 0 0 1];
    uel=T*ue;
    % Matice tuhosti (jen lokální)
    Kelb=[E*A/L 0 0 -E*A/L 0 0;
    0 12*E*I/L^3 6*E*I/L^2 0 -12*E*I/L^3 6*E*I/L^2;
    0 6*E*I/L^2 4*E*I/L 0 -6*E*I/L^2 2*E*I/L;
    -E*A/L 0 0 E*A/L 0 0;
    0 -12*E*I/L^3 -6*E*I/L^2 0 12*E*I/L^3 -6*E*I/L^2;
    0 6*E*I/L^2 2*E*I/L 0 -6*E*I/L^2 4*E*I/L];
    % síly v prutech
    Fe=Kelb*uel;
    N=Fe(1);
    % geometrická matice
    Me= N/L * [0 0 0 0 0 0;
    0 6/5 L/10 0 -6/5 L/10;
    0 L/10 2*L^2/15 0 -L/10 -L^2/30;
    0 0 0 0 0 0;
    0 -6/5 -L/10 0 6/5 -L/10;
    0 L/10 -L^2/30 0 -L/10 2*L^2/15];
    % Transformace
    Mg=T'*Me*T;
    for j=1:(puzlu*ndof)

```

```

    for k=1:(puzlu*ndof)
        M(kcis(i,j),kcis(i,k)) = M(kcis(i,j),kcis(i,k))+Mg(j,k);
    end
end
end
% Podpory na matici hmotnosti
for i=1:npodpor
    iuz=podpory(i,1);
    ismer=podpory(i,2);
    pos=(ndof*(iuz-1))+ismer;
    for j=1:velikost
        M(pos,j)=0.0;
        M(j,pos)=0.0;
    end
    M(pos,pos)=1.0;
end
% Metoda inverzních iterací
xk=zeros(velikost,1);
for i=1:velikost
    xk(i)=1;
end
% v aproximaci musí být splněny okrajové podmínky
for i=1:npodpor
    iuz=podpory(i,1);
    ismer=podpory(i,2);
    pos=(ndof*(iuz-1))+ismer;
    xk(pos)=0.0;
end
% Iterační cyklus
for i=1:10
    xk1 = K\(M*xk);
    xk=xk1/sqrt(xk1'*M*xk1); % "nové" zk
    % Násobitel - počítá se jako Rayleighuv koeficient
    ro=(xk'*K*xk)/(xk'*M*xk);
    % podmínka konvergence
    if (i>1)
        if (((abs(ro-ro0))/abs(ro))<(1e-8))
            break
        end
    end
    ro0=ro;
end
% Výchylky odpovídající ro

```

```

y=M*xk;
% -----
% Následující výpočet může stanovit
% vlastní frekvenci a tvar pomocí funkce EIG
% -----
[v,d]=eig(K, M);
D=diag(d);
disp('Násobitel zatížení je: ')
dd=D(1);
di=1;
for iy=1:size(D,1)
    if D(iy)>1.00001
        dd=D(iy);
        di=iy;
        break;
    end
end
for iy=di:size(D,1)
    if D(iy)<dd
        if D(iy)>0
            if D(iy)~=1.0
                dd=D(iy);
                di=iy;
            end
        end
    end
end
end
dd
% Deformovaný tvar (pro přímý nosník)
a=zeros(nuzlu,1);
b=zeros(nuzlu,1);
for i=1:nuzlu
    a(i,1)=i;
    pos=3*(i-1)+2;
    b(i,1)=y(pos);
end
plot(a,b);
disp('Inverzními iteracemi spočteno [kN]: ')
Fnu=ro
disp('Eulerovo řešení [kN]: ')
L=uzly(nuzlu,1)-uzly(1,1);
Frc=((pi^2)*E*I)/(L^2)/1000

```

## Kapitola 5

# Vlastní frekvence a tvary vlastního kmitání



### Cíle

Kapitola je zaměřena na:

- seznámení s jednoduchými úlohami dynamiky stavebních konstrukcí,
- teoretické pozadí výpočtu vlastních frekvencí a tvarů vlastního kmitání jednoduchých prutových konstrukcí,
- ukázkou využití vlastních čísel a vektorů v úlohách stavební dynamiky.

## 5.1 Úvod do problematiky

Při řešení úloh stavební dynamiky zaměřených na dynamické systémy s  $n$  stupni volnosti lze využít metodu konstant tuhosti, která vychází z podmínek dynamické rovnováhy sil působících na hmotné body vyjádřených pomocí konstant tuhosti.

### 5.1.1 Řešení soustavy pohybových rovnic

Pro určení vlastního kmitání při zanedbání útlumu se musí sestavit soustava  $n$  pohybových rovnic, kterou lze vyjádřit maticově:

$$[m]_D \cdot \{\ddot{v}(t)\} + [k] \cdot \{v(t)\} = 0, \quad (5.1)$$

kde  $[m]_D$  je diagonální matice hmotnosti,  $[k]$  matice tuhosti,  $\{\ddot{v}(t)\}$  vektor zrychlení a  $\{v(t)\}$  vektor přemístění v čase  $t$ .

Partikulární řešení homogenní diferenciální rovnice 2. řádu (5.1) nabývá tvar:

$$\{v(t)\} = \{v_{(j)}^0\} \cdot \sin(\omega_{(j)} \cdot t) \quad (5.2)$$

a

$$\{v(t)\} = \{v_{(j)}^0\} \cdot \cos(\omega_{(j)} \cdot t), \quad (5.3)$$

kde  $\omega_{(j)}$  je  $j$ -tá vlastní kruhová frekvence systému.

Dosažením rovnic (5.2) a (5.3) do pohybových rovnic (5.1) lze získat systém lineárních rovnic, tzv. frekvenční rovnice:

$$([k] - \omega_{(j)}^2 \cdot [m]_D) \cdot \{v_{(j)}^0\} = 0 \quad (5.4)$$

nebo

$$[A(\omega_{(j)})] \cdot \{v_{(j)}^0\} = 0, \quad (5.5)$$

kde  $[A(\omega_{(j)})] = ([k] - \omega_{(j)}^2 \cdot [m]_D)$  je tzv. matice dynamické tuhosti.

Netriviální řešení rovnic (5.4) nebo (5.5) je dáno charakteristickou rovnicí stupně  $n$ :

$$\det([k] - \omega_{(j)}^2 \cdot [m]_D) = \det[A(\omega_{(j)})] = 0. \quad (5.6)$$

Pokud jsou matice  $[m]_D$  a  $[k]$  pozitivně definitní, řešením charakteristické rovnice (5.6) je  $n$  reálných a pozitivních kořenů  $\lambda_{(j)} = \omega_{(j)}^2$ .

Ke každému vlastnímu číslu  $\lambda_{(j)}$  je také přiřazen příslušný vlastní vektor  $\{v_{(j)}^0\}$ , jenž odpovídá nenulovému řešení rovnic (5.4) nebo (5.5). Soustava těchto vlastních vektorů popisuje tvary vlastního kmitání konstrukce.

Pro každý kořen  $\lambda_{(j)} = \omega_{(j)}^2$  je splněna charakteristická rovnice (5.6) a rovnice (5.4) a (5.5) mají řešení:

$$\{v_{(j)}^0\} = \{v_{1(j)}^0, v_{2(j)}^0, \dots, v_{n(j)}^0\}^T = \{v_{i(j)}^0\}^T, \quad (5.7)$$

pro  $i, j = 1, \dots, n$ , které splňují rovnici (5.3).

Rovnici (5.4) lze upravit násobením maticí  $[m]_D^{-1}$  zleva:

$$([m]_D^{-1} \cdot [k] - \omega_{(j)}^2 \cdot [m]_D^{-1} \cdot [m]_D) \cdot \{v_{(j)}^0\} = [m]_D^{-1} \cdot 0, \quad (5.8)$$

čímž lze získat:

$$([D]^{-1} - \omega_{(j)}^2 \cdot [E]_D) \cdot \{v_{(j)}^0\} = 0, \quad (5.9)$$

kde

$$[D]^{-1} = [m]_D^{-1} \cdot [k] = [k_r]. \quad (5.10)$$

Matice  $[k_r]$  se nazývá inverzní dynamická matice nebo redukovaná matice tuhosti, ze které lze také určit vlastní čísla  $\lambda_{(j)}$  a příslušné vlastní vektory  $\{v_{(j)}^0\}$  (např. pomocí funkce `eig` v programovém systému MATLAB). Výsledné vlastní kruhové frekvence kmitání  $\omega_{(j)}$  se řadí do vektoru vlastních kruhových frekvencí kmitání podle velikosti:

$$\{\omega_{(j)}\} = \{\omega_{(1)}, \omega_{(2)}, \dots, \omega_{(n)}\}^T, \quad (5.11)$$

přičemž platí  $0 < \omega_{(1)} < \omega_{(2)} < \dots < \omega_{(n)}$ .

Při řešení homogenní soustavy rovnic (5.4) nebo (5.5) nelze pro každé  $\omega_{(j)}$  stanovit veškeré výchylky  $v_{i(j)}^0$ , ale pouze jejich vzájemný poměr:

$$\begin{aligned} v_{1(j)}^0 \div v_{2(j)}^0 \div \dots \div v_{n(j)}^0 &= A_{i1(j)} \div A_{i2(j)} \div \dots \div A_{in(j)} = \dots = \\ &= A_{1k(j)} \div A_{2k(j)} \div \dots \div A_{nk(j)}, \end{aligned} \quad (5.12)$$

kde poměr výchylek jednotlivých hmotných bodů pro každý tvar kmitání se rovná vzájemnému poměru algebraických doplňků subdeterminantů matice  $[A(\omega_{(j)})]$  určených vzhledem k prvkům libovolného řádku (index  $i$ ) nebo sloupce (index  $k$ ) matice  $[A(\omega_{(j)})]$ . Znamená to také, že soustava může kmitat v každém vlastním tvaru s výchylkami  $\{v_{(j)}^0\}$ , ale také s libovolným výkmitem s výchylkami  $q_{(j)}^0 \cdot \{v_{(j)}^0\}$ , kde  $q_{(j)}^0$  představuje libovolnou konstantu.

Pro  $n$  vlastních kruhových frekvencí lze určit také tzv. tvarové součinitele, které charakterizují vlastní tvary kmitání:

$$a_{2(j)} = \frac{v_{2(j)}^0}{v_{1(j)}^0}, \quad a_{3(j)} = \frac{v_{3(j)}^0}{v_{1(j)}^0}, \quad \dots \quad a_{n(j)} = \frac{v_{n(j)}^0}{v_{1(j)}^0}. \quad (5.13)$$

Všechny tvary vlastního kmitání obsažené ve vektorech (5.7) lze rovněž seřadit do tzv. modální matice, ve které prvky každého sloupce  $j$  určují  $j$ -tý tvar kmitání:

$$[v^0] = [\{v_{(1)}^0\}, \{v_{(2)}^0\}, \dots, \{v_{(j)}^0\}, \dots, \{v_{(n)}^0\}] . \quad (5.14)$$

Obecné řešení soustavy  $n$  pohybových rovnic (5.1) pro určení vlastního kmitání při zanedbání útlumu - tzv. volného kmitání, lze v maticovém tvaru popsat rovnicemi:

$$\{v_0(t)\} = \sum_{j=1}^n [A_{(j)}^0 \cdot \sin(\omega_{(j)} \cdot t) + B_{(j)}^0 \cdot \cos(\omega_{(j)} \cdot t)] \cdot \{v_{(j)}^0\} \quad (5.15)$$

a

$$\{v_0(t)\} = \sum_{j=1}^n q_{(j)}^0 \cdot \{v_{(j)}^0\} \cdot \sin(\omega_{(j)} \cdot t + \varphi_{(j)}^0) = \sum_{j=1}^n q_{(j)}(t) \cdot \{v_{(j)}^0\}, \quad (5.16)$$

kde konstanta  $q_{(j)}^0$  vyjadřuje podíl každého vlastního tvaru na složeném kmitání a fázové úhly  $\varphi_{(j)}^0$  vyjadřují fázové posunutí každého vlastního tvaru v čase.

Pro rychlost a zrychlení platí následující maticové rovnice:

$$\{\dot{v}_0(t)\} = \sum_{j=1}^n \omega_{(j)} \cdot [A_{(j)}^0 \cdot \cos(\omega_{(j)} \cdot t) + B_{(j)}^0 \cdot \sin(\omega_{(j)} \cdot t)] \cdot \{v_{(j)}^0\} \quad (5.17)$$



a

$$\{\ddot{v}_0(t)\} = - \sum_{j=1}^n \omega_{(j)}^2 \cdot [A_{(j)}^0 \cdot \sin(\omega_{(j)} \cdot t) + B_{(j)}^0 \cdot \cos(\omega_{(j)} \cdot t)] \cdot \{v_{(j)}^0\}. \quad (5.18)$$

### 5.1.2 Ortogonalita vlastních tvarů kmitání

Tvary vlastního kmitání jsou ortogonální (nezávislé), což lze dokázat pro  $j \neq h$  s využitím matice hmotnosti  $[m]_D$ :

$$\{v_{(j)}^0\}^T \cdot [m]_D \cdot \{v_{(h)}^0\} = 0 \quad (5.19)$$

nebo matice tuhosti  $[k]$ :

$$\{v_{(j)}^0\}^T \cdot [k] \cdot \{v_{(h)}^0\} = 0. \quad (5.20)$$

### 5.1.3 Normované vlastní tvary kmitání

Pro tvary vlastního kmitání se mohou amplitudy zvolit tak, aby pro výchylky  $q_{(j)}^0 \cdot \{v_{i(j)}^0\} = V_{i(j)}$  platilo:

$$\{V_{(j)}\}^T \cdot [m]_D \cdot \{V_{(j)}\} = 1. \quad (5.21)$$

Vektory  $\{V_{(j)}\}$  ve vztahu (5.21) vyjadřují normované vlastní tvary - tzv. ortonormované vlastní tvary, které lze určit:

$$\{V_{(j)}\} = \frac{\{v_{(j)}^0\}}{\sqrt{\{v_{(j)}^0\}^T \cdot [m]_D \cdot \{v_{(j)}^0\}}}. \quad (5.22)$$

Ortonormovaná modální matice se pak skládá z ortonormovaných vektorů:

$$[V] = [\{V_{(1)}\}, \{V_{(2)}\}, \dots, \{V_{(n)}\}] \quad (5.23)$$

Podmínky ortogonality lze definovat s využitím ortonormované modální matice  $[V]$  pomocí rovnic:

$$[V]^T \cdot [m]_D \cdot [V] = [E]_D \quad (5.24)$$

a

$$[V]^T \cdot [k] \cdot [V] = [\omega_{(j)}^2]_D, \quad (5.25)$$

kde  $[E]$  je diagonální jednotková matice a  $[\omega_{(j)}^2]_D$  je diagonální matice druhých mocnin vlastních kruhových frekvencí kmitání.

## 5.2 Určení vlastních frekvencí a tvarů vlastního kmitání u jednoduchých konstrukcí

Mechanická rezonance, tedy shoda vlastních mechanických kmitočtů stavební konstrukce s kmitočtem přicházejících impulsů (kmitů, rázů) je nežádoucím jevem, neboť může způsobit poškození kmitající části i částí, na kterou se kmitání přenáší. Síly buzené dynamickým zatížením (u mostů např. chodci nebo dopravou) ve stejné frekvenci totožné s některou vlastní frekvencí konstrukce může zapříčinit vznik tzv. rezonance. Z tohoto důvodu se u dynamicky namáhaných konstrukcí zjišťují dynamické charakteristiky, ke kterým patří např. vlastní frekvence (příslušející svislému, vodorovnému a kroutivému kmitání).



**Poznámka:** Kmitání lávek může mít různé příčiny, např. chodce, kteří mohou jít, běžet, poskakovat nebo tančit, vítr apod. Účinky pěší dopravy na lávkách přitom závisí na různých faktorech - např. na počtu a rozmístění chodců. Pokud nedochází k významné odezvě lávky, obvykle se pohybující chodci budí na lávce periodické současně působící síly ve svislém směru s frekvencí v rozmezí 1 až 3 Hz a ve vodorovném směru s frekvencí v rozmezí 0,5 až 1,5 Hz. Skupina klusajících chodců se může pohybovat po lávce s frekvencí 3 Hz.



**Poznámka:** Pokud je některá vlastní frekvence svislého kmitání nosné konstrukce lávky v oblasti frekvencí 1,0 až 5,0 Hz a/nebo vlastní frekvence vodorovného kmitání hlavní nosné konstrukce lávky v oblasti frekvencí 0,5 až 2,5 Hz, je podle ČSN EN 1991-2 Zatížení mostů dopravou nutné provést podrobnou analýzu dynamického chování lávky.

Analýza musí obsahovat výpočet vlastních frekvencí a tvarů kmitání lávky na vhodném výpočetním modelu, odhad tlumení konstrukce, výpočet vynuceného kmitání lávky od skupiny chodců nebo proudu chodců.



**Poznámka:** V případě lehkých výztužných prvků, lanových závěsů a podobných konstrukčních prvků by se podle ČSN EN 1993-2 Navrhování ocelových konstrukcí - Ocelové mosty, mělo rovněž uvažovat se vznikem rezonance konstrukce. Pokud mají tyto konstrukční prvky vlastní frekvence blízké frekvenci libovolného mechanického buzení od pravidelného přejíždění vozidel přes spoje mostovky, má se buď zvýšit tuhost nebo se mají do konstrukce vložit tlumiče kmitání.

V následujícím učebním textu bude výklad zaměřen na stanovení vlastních tvarů kmitání a vlastních frekvencí jednoduchých stavebních konstrukcí.



**Příklad 5.1.** S využitím výpočetních postupů popsaných v kapitole 5.1 určete vlastní frekvence a tvary vlastního kmitání na prostém nosníku délky  $L = 6$  metrů, který je tvořen ocelovým válcovaným profilem IPE300. Výpočet proveďte pro počet stupňů volnosti  $n = 3, 6, 10, 20, 50, 100$ , přičemž sledujte změny hodnot výsledných veličin i vlastních tvarů.

*Řešení.* Požadovaný výpočet lze provést s využitím následujícího skriptu vytvořeného v prostředí programového systému MATLAB:

```
% Příklad 1:
% Určuje vlastní frekvence a tvary vlastního kmitání
% na prostém nosníku
clc; clear;
% Zadání
L=6;           % Délka nosníku [m]
E=2.1e11;     % Modul pružnosti v tahu a prostém tlaku [Pa]
I=8.356e-5;   % Moment setrvačnosti [m^4] (IPE300)
mi=42.2;      % Hmotnost [kg/m] (IPE300)
n=30;         % Počet stupňů volnosti
ndraw=6;      % Kolik vlastních tvarů vykreslit (ndraw<=n)
d=L/(n+1);    % Nosník se rozdělí na n+1 dílků stejné délky d
m=mi*d;       % Velikost soustředěné hmoty v bodě i
mr=m*ones(1,n); % Matice hmotnosti v řádkovém tvaru
md=diag(mr,0); % Diagonální matice hmotnosti [m]
% Inverzní matice k matici hmotnosti
mdinv=diag(ones(n,1)./diag(md,0),0);
% Výpočet matice konstant tuhosti [k] pro prostý nosník
for i=1:n
    pb=i/(n+1);
    ppb=1-pb;
    for j=i:n;
        xb=j/(n+1);
        xpb=1-xb;
        Delb(j,i)=pb*xpb*(1-(pb^2)-(xpb^2));
        Delb(i,j)=Delb(j,i);
    end
end
Delb=(1/6)*Delb;
Suc=(L^3)/(E*I);
Del=Suc*Delb;
kb=inv(Delb);
Suci=1/Suc;
k=Suci*kb;
% Redukovaná matice tuhosti kr
kr=mdinv*k;
% Výpočet vlastních čísel
[v,om2]=eig(kr); % Vlastní vektory a vlastní čísla matice kr
om2v=diag(om2,0); % Vektor 2. mocnin vlastních úhlových frekvencí
omv=sqrt(om2v); % Vektor vlastních úhlových frekvencí
```

```

fv=omv./(2*pi); % Vektor vlastních frekvencí [Hz]
% Frekvence a tvary kmitání uspořádané podle pořadí
omv=sort(omv); % Uspořádá prvky vektoru omv podle velikosti
[fv,i]=sort(fv); % Uspořádá prvky vektoru {fv} podle velikosti
% a pořadí indexů uloží do vektoru {i}
v=v(:,i); % Vymění pořadí sloupců matice v
% podle pořadí vlastních tvarů
% Výpočet ortonormované modální matice V
V=zeros(n);
for j=1:n
    vj=v(:,j);
    VJ=vj./sqrt(vj'*md*vj);
    V(:,j)=VJ;
end
% Kontrola ortonormovaných tvarů kmitání
Ekont=V'*md*V; % Musí vyjít [E]
norm(Ekont-eye(n))% norma
% Vykreslení výsledků
x=linspace(0,L,n+2);
xp=[0,L]; yp=[0,0];
xv=x(2:n+1); % Vnitřní body
figure
for j=1:ndraw % j-tý tvar kmitání
    VJ=[0;V(:,j);0];
    jr=int2str(j);
    fr=num2str(fv(j,1),8);
    out=[jr,' tvar kmitání, f(' ,jr,' ) = ',fr,' Hz'];
    if ndraw<4 % Graf v j-tém kvadrantu v rastru s 1 sloupcem
        subplot(ndraw,1,j)
    else % Graf v j-tém kvadrantu v rastru s 2 sloupci
        if mod(ndraw,2)==1
            subplot(floor(ndraw/2)+1,2,j)
        else
            subplot(ndraw/2,2,j)
        end
    end
    hold on
    plot(x,VJ,'b-',xp,yp,'r^--',xv,V(:,j),'bo')
    title(out);
    stem(x,VJ,'b-', 'marker', 'none');
    hold off
end
end

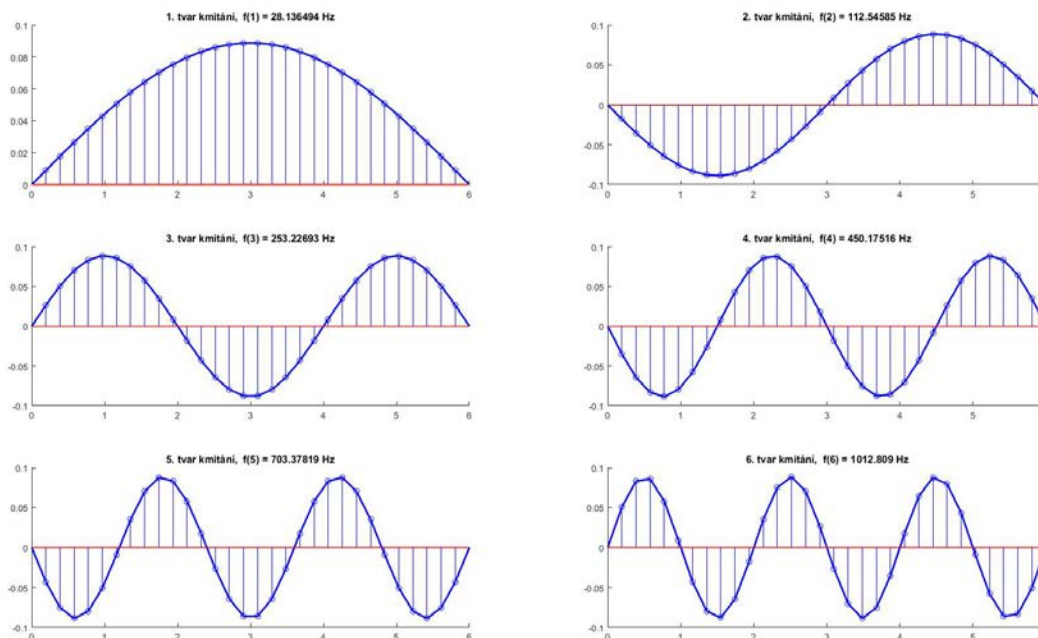
```

Výsledné vlastní frekvence jsou pro řešení kmitání prostého nosníku s 30 stupni volnosti z příkladu 5.1 uvedeny pro prvních 30 vlastních tvarů v následujícím výpisu:

```
*****
*       Výpočet vlastních frekvencí - prostý nosník       *
*****
Počet stupňů volnosti dynamické soustavy =          30
Kontrola ortonormovaných tvarů kmitání   = 1.925e-15
```

číslo vlastního tvaru	vlastní frekvence [Hz]
1	28.136
2	112.546
3	253.227
4	450.175
5	703.378
6	1012.809
7	1378.416
8	1800.106
9	2277.729
10	2811.040
11	3399.668
12	4043.055
13	4740.382
14	5490.472
15	6291.661
16	7141.624
17	8037.163
18	8973.935
19	9946.126
20	10946.069
21	11963.821
22	12986.727
23	13999.051
24	14981.738
25	15912.460
26	16766.092
27	17515.736
28	18134.386
29	18597.143
30	18883.716

Vlastní frekvence a prvních 6 tvarů vlastního kmitání prostého nosníku z příkladu 5.1 jsou také graficky zobrazeny na obrázku 5.1.



Obr. 5.1 Vlastní frekvence a prvních 6 tvarů vlastního kmitání prostého nosníku z příkladu 5.1 se zadanými 30 stupni volnosti.



**Příklad 5.2.** S využitím výpočetních postupů popsaných v kapitole 5.1 určete vlastní frekvence a tvary vlastního kmitání na konzolovém nosníku délky  $L = 2$  metry, který je tvořen ocelovým válcovaným profilem IPE300. Výpočet proveďte pro počet stupňů volnosti  $n = 3, 6, 10, 20, 50, 100$ , přičemž sledujte změny hodnot výsledných veličin i vlastních tvarů.

*Řešení.* Požadovaný výpočet lze provést s využitím následujícího skriptu vytvořeného v prostředí programového systému MATLAB:

```
% Příklad 2:
% Určuje vlastní frekvence a tvary vlastního kmitání
% na konzole
clc; clear;
% Zadání
L=2;           % Délka nosníku [m]
E=2.1e11;     % Modul pružnosti v tahu a prostém tlaku [Pa]
I=8.356e-5;   % Moment setrvačnosti [m^4] (IPE300)
```

```

mi=42.2;          % Hmotnost [kg/m] (IPE300)
n=24;            % Počet stupňů volnosti
ndraw=6;        % Kolik vlastních tvarů vykreslit (ndraw<=n)
d=L/n;          % Nosník se rozdělí na n dílků stejné délky d
m=mi*d;         % Velikost soustředěné hmoty v bodě i
mn=m/2;         % Velikost hmoty v koncovém bodě
mrv=m*ones(1,n-1); % Matice hmotnosti vnitřních bodů v řádkovém tvaru
mr=[mrv,mn];    % Matice hmotnosti v řádkovém tvaru
md=diag(mr,0);  % Diagonální matice hmotnosti [m]
% Inverzní matice k matici hmotnosti
mdinv=diag(ones(n,1)./diag(md,0),0);
% Výpočet matice konstant tuhosti [k] pro konzolu
for i=1:n
    pb=i/n;
    ppb=1-pb;
    for j=i:n;
        xb=j/n;
        xpb=1-xb;
        Delb(j,i)=3*(pb^2)*xb-(pb^3);
        Delb(i,j)=Delb(j,i);
    end
end
Delb=(1/6)*Delb;
Suc=(L^3)/(E*I);
Del=Suc*Delb;
kb=inv(Delb);
Suci=1/Suc;
k=Suci*kb;
% Redukovaná matice tuhosti kr
kr=mdinv*k;
% Výpočet vlastních čísel
[v,om2]=eig(kr); % Vlastní vektory a vlastní čísla matice kr
om2v=diag(om2,0); % Vektor 2. mocnin vlastních úhlových frekvencí
omv=sqrt(om2v); % Vektor vlastních úhlových frekvencí
fv=omv./(2*pi); % Vektor vlastních frekvencí [Hz]
% Frekvence a tvary kmitání uspořádané podle pořadí
omv=sort(omv); % Uspořádá prvky vektoru omv podle velikosti
[fv,i]=sort(fv); % Uspořádá prvky vektoru {fv} podle velikosti
% a pořadí indexů uloží do vektoru {i}
v=v(:,i); % Vymění pořadí sloupců matice v
% podle pořadí vlastních tvarů
% Výpočet ortonormované modální matice V
V=zeros(n);

```

```

for j=1:n
    vj=v(:,j);
    VJ=vj./sqrt(vj'*md*vj);
    V(:,j)=VJ;
end
% Kontrola ortonormovaných tvarů kmitání
Ekont=V'*md*V;      % Musí vyjít [E]
norm(Ekont-eye(n)) % norma
% Vykreslení výsledků
x=linspace(0,L,n+1);
y=zeros(1,n+1);
xp=[0,0];           % Vetknutí
yp=[-0.1,+0.1];    % Vetknutí
xv=x(2:n+1);       % Všechny body mimo vetknutí
figure
for j=1:ndraw      % j-tý tvar kmitání
    VJ=[0;V(:,j)];
    jr=int2str(j);
    fr=num2str(fv(j,1),8);
    out=[jr,'. tvar kmitání, f(',jr,') = ',fr,' Hz'];
    if ndraw<4 % Graf v j-tém kvadrantu v rastru s 1 sloupcem
        subplot(ndraw,1,j)
    else % Graf v j-tém kvadrantu v rastru s 2 sloupci
        if mod(ndraw,2)==1
            subplot(floor(ndraw/2)+1,2,j)
        else
            subplot(ndraw/2,2,j)
        end
    end
end
hold on
plot(x,y,'r--',x,VJ,'b-',xp,yp,'r-',xv,V(:,j),'bo')
title(out);
stem(x,VJ,'b-', 'marker', 'none');
hold off
end

```

Výsledné vlastní frekvence jsou uvedeny v následujícím výpisu:

```

*****
*      Výpočet vlastních frekvencí - konzolový nosník      *
*****

```

```

Počet stupňů volnosti dynamické soustavy =          24
Kontrola ortonormovaných tvarů kmitání   = 7.491e-13

```



číslo vlastního tvaru	vlastní frekvence [Hz]
1	90.140
2	563.789
3	1575.829
4	3082.360
5	5085.961
6	7583.208
7	10570.704
8	14044.370
9	17999.016
10	22427.594
11	27320.027
12	32661.405
13	38429.330
14	44590.142
15	51093.856
16	57867.764
17	64808.999
18	71777.031
19	78588.048
20	85014.464
21	90793.674
22	95649.282
23	99323.604
24	101612.584

Vlastní frekvence a prvních 6 tvarů vlastního kmitání konzolového nosníku z příkladu 5.2 jsou pak graficky zobrazeny na obrázku 5.2.

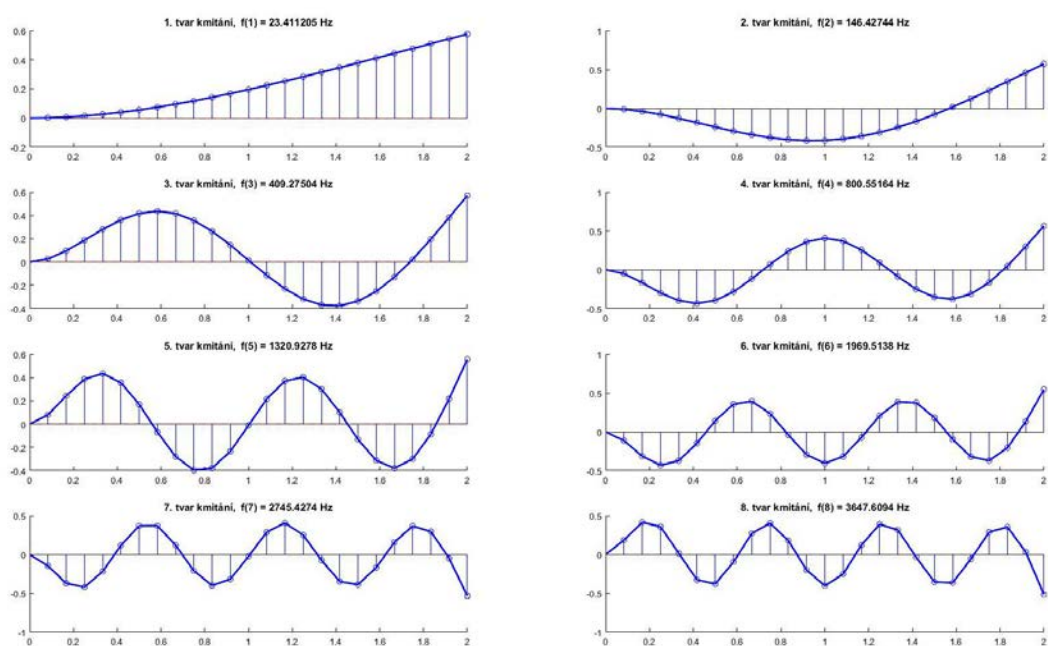


**Příklad 5.3.** U obou obou předchozích nosníků změňte průřez ocelového nosníku např. na IPE200 a IPE140 a sledujte změny v hodnotách vlastních frekvencí.



**Pro zájemce:** S využitím výpočetních postupů popsaných v kapitole 5.1 a vztahů pro určení příslušné matice tuhosti  $[k]$  v [10, 11] určete vlastní frekvence a tvary vlastního kmitání na jednostranně či oboustranně vetknutém nosníku konstantního průřezu.





Obr. 5.2 Vlastní frekvence a prvních 8 tvarů vlastního kmitání konzolového nosníku z příkladu 5.2 se zadanými 24 stupni volnosti.

## Kapitola 6

# Náhodné proměnné a pravděpodobnostní simulační výpočty

### Cíle



Kapitola je zaměřena na:

- způsoby pravděpodobnostního vyjádření náhodných proměnných,
- jednoduché pravděpodobnostní úlohy využívající simulační metody,
- ukázkou využití statistických nástrojů programového systému MATLAB.

### 6.1 Náhodná veličina

Náhodná veličina (používá se i přívlastek stochastická nebo proměnná) je libovolná veličina, kterou je možné opakovaně měřit v čase, a její hodnoty podrobit zpracování metodami teorie pravděpodobnosti nebo matematické statistiky. Náhodné veličiny jsou proměnné, jejichž hodnoty při konstantních podmínkách závisí na náhodě, přičemž každá z těchto hodnot vystupuje s určitou pravděpodobností. Náhodné veličiny mohou být diskrétní nebo spojité.

**Definice 6.1.** Náhodným jevem se rozumí opakovatelná činnost prováděná za stejných (nebo přibližně stejných) podmínek, jejíž výsledek je nejistý a závisí na náhodě.

**Definice 6.2.** Pravděpodobnost náhodného jevu je číslo, které je mírou očekávatelnosti výskytu náhodného jevu (s jakou jistotou lze náhodný jev očekávat). Pravděpodobnost náhodného jevu se obecně označuje reálným číslem od 0 do 1. Náhodný jev, který nemůže nastat, má pravděpodobnost 0. Naopak náhodný jev jistý má pravděpodobnost 1. Pravděpodobnost lze také uvádět v procentech (0 až 100 %).

**Definice 6.3.** Rozdělení pravděpodobnosti je funkce, která přiřazuje pravděpodobnosti náhodným jevům.

### 6.1.1 Diskrétní náhodná veličina

Náhodná veličina  $X$  je diskrétní, jestliže se prvky výběrového prostoru  $\Omega$  zobrazí na osu reálných čísel jako izolované body  $x_1, x_2, \dots, x_k$ , přičemž každý z těchto bodů má nenulovou pravděpodobnost.

Pravděpodobnostní funkce (anglicky *probability mass function – PMF*) je funkce v teorii pravděpodobnosti a statistice, která udává pravděpodobnost, že diskrétní náhodná veličina se rovná příslušné hodnotě:

$$f_X(x) = P(X = x) . \quad (6.1)$$

Rovněž platí:

$$\sum_{x \in \Omega} f_X(x) = 1 . \quad (6.2)$$

Distribuční funkce  $F$  (kumulovaná pravděpodobnost, anglicky *Cumulative Distribution Function – CDF*) je funkce, která udává pravděpodobnost, že hodnota náhodné proměnné  $X$  je menší než zadaná hodnota  $x$ :

$$F_X(x) = P(X \leq x) . \quad (6.3)$$

### 6.1.2 Spojitá náhodná veličina

Náhodná veličina  $X$  je spojitá, jestliže existuje nezáporná funkce  $f$ , pro kterou za předpokladu  $a < b$  platí:

$$P(a < X < b) = \int_a^b f_X(x) dx . \quad (6.4)$$

Funkce  $f$  se nazývá hustotou rozdělení pravděpodobnosti náhodné veličiny  $X$  (hustota pravděpodobnosti, anglicky *probability density function – PDF*, také se používá označení frekvenční funkce). K jejím vlastnostem patří:

$$f_X(x) \geq 0 \quad (6.5)$$

a

$$\int_{\Omega} f_X(x) dx = 1, \quad (6.6)$$

kde  $\Omega$  je definiční obor náhodné proměnné  $X$ .

Distribuční funkce  $F_X(x)$  spojité náhodné veličiny  $X$  je definovaná jako pravděpodobnost, že realizace této náhodné veličiny  $X$  nepřekročí hodnotu  $x$ :

$$F_X(x) = P(X \leq x). \quad (6.7)$$

Distribuční funkce je neklesající a její hodnota  $F(-\infty) = 0$  a  $F(\infty) = 1$ .

### 6.1.3 Parametrické rozdělení pravděpodobnosti

Náhodná veličina  $X$  je jednoznačně určena rozdělením pravděpodobnosti pomocí *pravděpodobnostní funkce* u diskrétních náhodných veličin, *hustoty pravděpodobnosti* u spojitých náhodných veličin nebo *distribuční funkce*. K jejich určení často slouží číselné charakteristiky, označované jako statistické momenty. Patří k nim střední hodnota  $\mu$  (anglicky *mean* nebo *expected value*), rozptyl  $\sigma^2$  (anglicky *variance*; resp. směrodatná odchylka  $\sigma$ , anglicky *standard deviation*), šikmost (koeficient asymetrie, anglicky *skewness*) a špičatost (koeficient excesu, anglicky *kurtosis*).

Tyto charakteristiky jsou často používány jako vstupní parametry tzv. parametrických rozdělení pravděpodobnosti. Mezi nejpoužívanější parametrická rozdělení pravděpodobnosti diskrétních náhodných veličin patří:

- Alternativní rozdělení (náhodná veličina  $X$  nabývá pouze hodnot 0 nebo 1, lze použít např. pro simulaci hodu mincí),
- Binomické rozdělení ( $n$  náhodných pokusů se stejnou pravděpodobností, lze použít např. pro simulaci hodu hrací kostkou),
- Poissonovo rozdělení,
- Negativně binomické rozdělení,
- Pascalovo rozdělení (speciální případ negativně binomického rozdělení),
- Geometrické rozdělení (speciální případ Pascalova rozdělení),
- Hypergeometrické rozdělení,
- Logaritmické rozdělení.

K často používaným parametrickým rozdělením pravděpodobnosti spojitých náhodných veličin patří:

- Rovnoměrné rozdělení,
- Normální rozdělení (označované také jako Gaussovo rozdělení),
- Logaritmicke-normální rozdělení (také log-normální rozdělení),

- Exponenciální rozdělení,
- Cauchyho rozdělení,
- Gama rozdělení,
- Laplaceovo rozdělení (nebo také dvojitě exponenciální rozdělení),
- Logistické rozdělení,
- Maxwelllovo-Boltzmannovo rozdělení,
- Studentovo rozdělení,
- Fischerovo-Snedecorovo rozdělení,
- $\chi^2$  rozdělení (Chí kvadrát).

### 6.1.4 Neparametrické (empirické) rozdělení pravděpodobnosti

Další z možností jak vyjádřit rozdělení pravděpodobnosti náhodné veličiny  $X$  je použití neparametrického (empirického) rozdělení pravděpodobnosti, které je založeno na výběrovém souboru dat získaných např. měřením nebo monitoringem. Neparametrické (empirické) rozdělení pravděpodobnosti náhodné veličiny  $X$  může být vyjádřeno histogramem o  $n$  intervalech (třídách) se stejnou šířkou  $\Delta x$ .

**Definice 6.4.** Histogram je grafické znázornění distribuce dat pomocí sloupcového grafu se sloupci stejné šířky, vyjadřující šířku intervalů (tříd), přičemž výška sloupců vyjadřuje četnost (nebo pravděpodobnost) sledované veličiny v daném intervalu.

Jedním ze způsobů, jak lze neparametrické (empirické) rozdělení pravděpodobnosti ve formě histogramu uložit do datového souboru s příponou \*.dis, je následující zápis textového souboru:

```
[Description] (1. oddíl datového souboru)
Identification= volitelný popis datového souboru
Type= Pure Discrete | Discrete | Continuous (typ rozdělení)
```

```
[Parameters] (2. oddíl datového souboru)
Min= minimální hodnota náhodné proměnné
Max= maximální hodnota náhodné proměnné
Bins= celkový počet intervalů (tříd) daného histogramu
Total= součet četností ve všech intervalech
```

```
[Bins] (3. oddíl datového souboru)
... (četnosti v jednotlivých intervalech)
```

kde položka *Identification* obsahuje popis daného histogramu, např. o jaká data se jedná, příp. jejich původ. V oddíle **[Bins]** jsou na každém dalším řádku uvedeny četnosti jednotlivých intervalů (tříd). Jejich součet musí odpovídat uvedené hodnotě *Total* v předchozím oddílu datového souboru.

### 6.1.5 Generování náhodných proměnných v Matlabu

**Příklad 6.5.** S využitím funkcí programového systému MATLAB vygenerujte  $N = 1 \cdot 10^6$  hodnot náhodné proměnné  $A$  s normálním rozdělením pravděpodobnosti s parametry: střední hodnota= 200 a směrodatná odchylka= 15. Určete u ní statistické parametry a zobrazte ve formě histogramu s 64 intervaly.



*Řešení.* Požadovaný výpočet lze provést s využitím následujícího skriptu vytvořeného v prostředí programového systému MATLAB:

```

clc; clear;
N=1e6;      % Počet simulací
str_hod=200; % Střední hodnota
sm_odch=15; % Směrodatná odchylka
Bins=64;    % Počet intervalů v histogramu
A = random('Normal',str_hod,sm_odch,1,N);
disp(sprintf('\n'))
disp('*****')
disp('*          Generování (pseudo)náhodné veličiny          *')
disp('*          s parametrickým rozdělením pravděpodobnosti    *')
disp('*          Normalní / Gaussovo                             *')
disp('*****')
disp(' ')
disp('Vstupní údaje:')
disp('-----')
disp(sprintf('Počet simulací N          = %12d',N));
disp(sprintf('Střední hodnota          = %12.3f',str_hod));
disp(sprintf('Směrodatná odchylka        = %12.3f',sm_odch));
disp(' ')
disp('Parametry vygenerované náhodné veličiny:')
disp('-----')
disp(sprintf('Minimální hodnota          = %12.3f',min(A)));
disp(sprintf('Maximální hodnota          = %12.3f',max(A)));
disp(sprintf('Rozpětí                      = %12.3f',range(A)));
disp(sprintf('Střední hodnota            = %12.3f',mean(A)));
disp(sprintf('Směrodatná odchylka        = %12.3f',std(A)));
disp(sprintf('Rozptyl                     = %12.3f',var(A)));
disp(sprintf('Variační koeficient         = %11.2f%%',...
    std(A)/mean(A)*100));
disp(sprintf('Šikmost                    = %12.3f',skewness(A)));
disp(sprintf('Špičatost                  = %12.3f',kurtosis(A)));
disp(sprintf('Medián                     = %12.3f',median(A)));
disp(sprintf('Kvantil 5%%                = %12.3f',quantile(A,0.05)));
disp(sprintf('Percentil 5%%              = %12.3f',prctile(A,5)));

```

```

disp(sprintf('Pravděpodobnost(X<200) = %11.2f%%',sum(A<200)/N*100));
figure(1);
histfit(A,Bins);
% Mění barvu histogramu - červené sloupce s bílým okrajem
set(findobj(gca,'Type','patch'),'FaceColor','r',...
    'EdgeColor','w','FaceColor','b');
title('Normalní parametrické rozdělení pravděpodobnosti');
xlabel('Náhodná proměnná x');
ylabel('Četnost');

```

Textový výstup úlohy pak může vypadat následovně:

```

*****
*           Generování (pseudo)náhodné veličiny           *
*           s parametrickým rozdělením pravděpodobnosti *
*           Normalní / Gaussovo                          *
*****

```

Vstupní údaje:

```

-----
Počet simulací N      =      1000000
Střední hodnota      =      200.000
Směrodatná odchylka =      15.000

```

Parametry vygenerované náhodné veličiny:

```

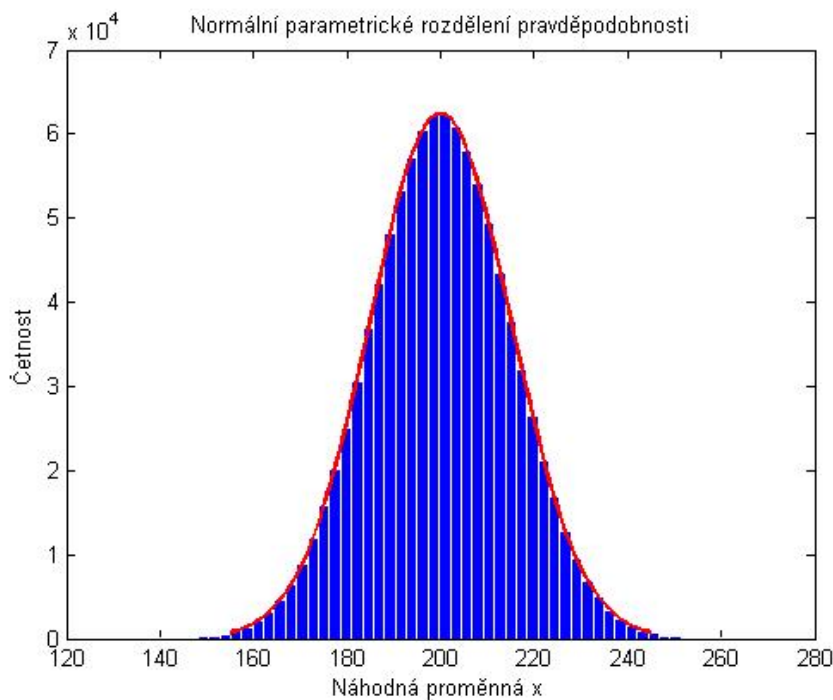
-----
Minimální hodnota    =      124.205
Maximální hodnota    =      269.629
Rozpětí              =      145.423
Střední hodnota      =      200.014
Směrodatná odchylka =      14.995
Rozptyl              =      224.858
Variační koeficient  =      7.50%
Šikmost              =      0.000
Špičatost            =      3.004
Medián                =      200.015
Kvantil 5%           =      175.359
Percentil 5%         =      175.359
Pravděpodobnost(X<200) =      49.96%

```

Grafické zobrazení normalního parametrického rozdělení pravděpodobnosti z příkladu 6.5 je uvedeno na obrázku 6.1.







Obr. 6.1 Normální parametrické rozdělení pravděpodobnosti z příkladu 6.5.

**Příklad 6.6.** Podobně jako v příkladu 6.5 vygenerujte  $N = 1 \cdot 10^6$  hodnot náhodné proměnné  $A$  s log-normálním rozdělením pravděpodobnosti.



*Řešení.* Požadovaný výpočet lze provést s využitím následujícího skriptu:

```
clc; clear;
N=1e6;      % Počet simulací
str_hod=200; % Střední hodnota
sm_odch=15; % Směrodatná odchylka
Bins=64;    % Počet intervalů v histogramu
A=random('Lognormal',log((str_hod^2)/sqrt(sm_odch^2+str_hod^2)),...
        sqrt(log(((sm_odch^2)/(str_hod^2))+1))),1,N);
disp(sprintf('\n'))
disp('*****')
disp('*          Generování (pseudo)náhodné veličiny          *')
disp('*          s parametrickým rozdělením pravděpodobnosti    *')
disp('*          Lognormální                                       *')
disp('*****')
disp(' ')
disp('Vstupní údaje:')
```

```

disp('-----')
disp(sprintf('Počet simulací N      = %12.1e',N));
disp(sprintf('Střední hodnota      = %12.3f',str_hod));
disp(sprintf('Směrodatná odchylka   = %12.3f',sm_odch));
disp(' ')
disp('Parametry vygenerované náhodné veličiny:')
disp('-----')
disp(sprintf('Minimální hodnota      = %12.3f',min(A)));
disp(sprintf('Maximální hodnota      = %12.3f',max(A)));
disp(sprintf('Rozpětí                = %12.3f',range(A)));
disp(sprintf('Střední hodnota        = %12.3f',mean(A)));
disp(sprintf('Směrodatná odchylka     = %12.3f',std(A)));
disp(sprintf('Rozptyl                = %12.3f',var(A)));
disp(sprintf('Variační koeficient         = %11.2f%%',...
    std(A)/mean(A)*100));
disp(sprintf('Šikmost                = %12.3f',skewness(A)));
disp(sprintf('Špičatost                = %12.3f',kurtosis(A)));
disp(sprintf('Medián                  = %12.3f',median(A)));
disp(sprintf('Kvantil 5%%                = %12.3f',quantile(A,0.05)));
disp(sprintf('Percentil 5%%              = %12.3f',prctile(A,5)));
disp(sprintf('Pravděpodobnost(X<200) = %11.2f%%',sum(A<200)/N*100));
figure(1);
histfit(A,Bins,'lognormal');
% Mění barvu histogramu - červené sloupce s bílým okrajem
set(findobj(gca,'Type','patch'),'FaceColor','r',...
    'EdgeColor','w','FaceColor','b');
title('Lognormalní parametrické rozdělení pravděpodobnosti');
xlabel('Náhodná proměnná x');
ylabel('Četnost');

```

Textový výstup úlohy pak může vypadat následovně:

```

*****
*           Generování (pseudo)náhodné veličiny           *
*           s parametrickým rozdělením pravděpodobnosti *
*           Lognormální                                    *
*****

```

Vstupní údaje:

```

-----
Počet simulací N      =      1.0e+006
Střední hodnota      =      200.000
Směrodatná odchylka  =      15.000

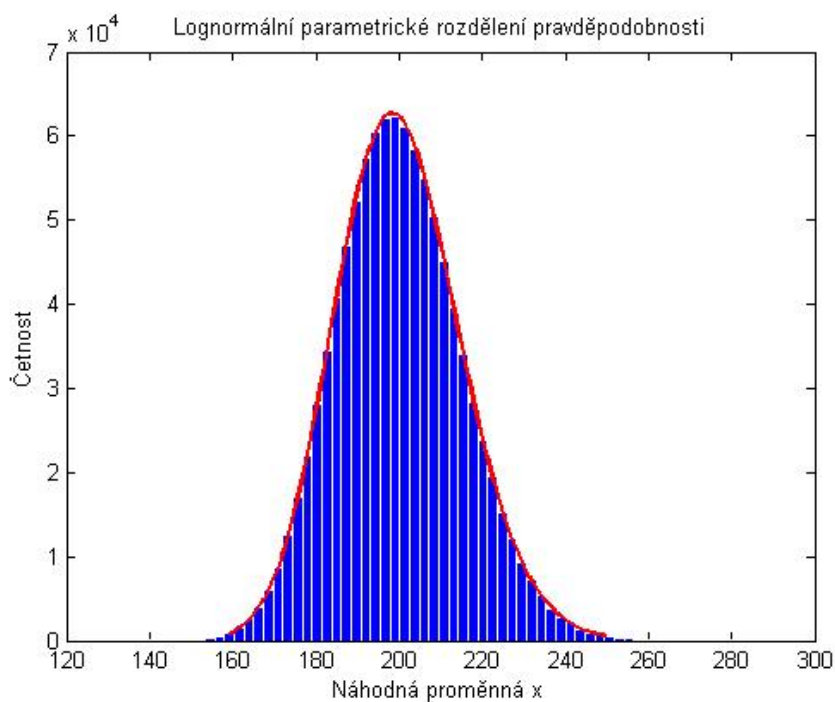
```

Parametry vygenerované náhodné veličiny:

---

Minimální hodnota	=	140.916
Maximální hodnota	=	285.838
Rozpětí	=	144.922
Střední hodnota	=	199.987
Směrodatná odchylka	=	14.993
Rozptyl	=	224.777
Variační koeficient	=	7.50%
Šikmost	=	0.226
Špičatost	=	3.096
Medián	=	199.432
Kvantil 5%	=	176.323
Percentil 5%	=	176.323
Pravděpodobnost ( $X < 200$ )	=	51.50%

Grafické zobrazení logaritmicko-normalního parametrického rozdělení pravděpodobnosti z příkladu 6.6 je uvedeno na obrázku 6.2.



Obr. 6.2 Logaritmicko-normalní parametrické rozdělení pravděpodobnosti z příkladu 6.6.





**Příklad 6.7.** Podobně jako v příkladu 6.5 a 6.6 vygenerujte  $N = 1 \cdot 10^6$  hodnot náhodné proměnné  $A$  s neparametrickým rozdělením pravděpodobnosti, definovaným v souboru T235FY01.DIS z databáze programu ProbCalc [4].

*Řešení.* Požadovaný výpočet lze provést s využitím následujícího skriptu:

```

clc; clear;
N=1e6;          % Počet simulací
filename='T235FY01.DIS';
[his,prob,d_val,type]=crea_his(filename);
A=crea_sa2(his,rand(1,N),d_val,type);
disp(sprintf('\n'))
disp('*****')
disp('*          Generování (pseudo)náhodné veličiny          *')
disp('*      s neparametrickým rozdělením pravděpodobnosti      *')
disp('*****')
disp(' ')
disp('Vstupní údaje:')
disp('-----')
disp(sprintf('Počet simulací N          = %12.1e',N));
disp(sprintf('Název souboru *.dis        = %12s',filename));
disp(' ')
disp('Parametry vygenerované náhodné veličiny:')
disp('-----')
disp(sprintf('Minimální hodnota          = %12.3f',min(A)));
disp(sprintf('Maximální hodnota          = %12.3f',max(A)));
disp(sprintf('Rozpětí                          = %12.3f',range(A)));
disp(sprintf('Střední hodnota                = %12.3f',mean(A)));
disp(sprintf('Směrodatná odchylka           = %12.3f',std(A)));
disp(sprintf('Rozptyl                          = %12.3f',var(A)));
disp(sprintf('Variační koeficient            = %11.2f%%',...
    std(A)/mean(A)*100));
disp(sprintf('Šikmost                          = %12.3f',skewness(A)));
disp(sprintf('Špičatost                        = %12.3f',kurtosis(A)));
disp(sprintf('Medián                          = %12.3f',median(A)));
disp(sprintf('Kvantil 5%%                       = %12.3f',quantile(A,0.05)));
disp(sprintf('Percentil 5%%                     = %12.3f',prctile(A,5)));
disp(sprintf('Pravděpodobnost(X<235) = %11.2f%%',sum(A<235)/N*100));
Bins=length(d_val);
figure(1);
hist(A,Bins);
% Mění barvu histogramu - červené sloupce s bílým okrajem
set(findobj(gca,'Type','patch'),'FaceColor','r',...

```

```

    'EdgeColor','w','FaceColor','b')
title('Neparametrické rozdělení pravděpodobnosti');
xlabel('Náhodná proměnná x');
ylabel('Četnost');
clear i;

```

Ve skriptu je obsažena funkce pro načtení parametrů histogramu `crea_his`, která je naprogramována v souboru `crea_his.m` s následujícím obsahem:

```

function [D_DISTR, PROB_F, D_VALUE, TYPE] =crea_his(FILENAME)
% [D_DISTR, PROB_F, D_VALUE, TYPE] =crea_his(FILENAME)
%
% Create the discrete or continuous histogram and prob.function,
% i.e. the pairs pi(x),xi, i=1:Bins
% length(D_VALUE) ...
% ... the number of discrete values of the random variable FILENAME
% sum(PROB_F)==1
%
% INPUT
%=====
% FILENAME - Histogram name
%
% OUTPUT
%=====
% D_DISTR - Distribution Function
% PROB_F - Probability Density of Histogram
% D_VALUE - Value of Particular Histogram Bin
% TYPE - Histogram TYPE: 0 - Discrete, 1 - Piece-wise uniform

% Author: Pavel PRAKS, pavel.praks@vsb.cz
% Edited by: Petr KONECNY, petr.konecny@vsb.cz
% VSB - Technical University Ostrava, www.vsb.cz, Czech Republic
% 04-11-2011

[Min, Max, Bins, Total, freq, TYPE]=read_his(FILENAME);

%CLEARING VARIABLES
PROB_F=[];
D_DISTR=[];
D_VALUE=[];

if TYPE == 1
%CONTINUOUS HISTOGRAM
d_step = (Max-Min)/(Bins);

```

```

    for i=1:Bins
        D_VALUE(i)= Min + (i-0.5)*d_step;
    end;
else
    %DISCRETE HISTOGRAM
    TYPE = 0; % if ther is no result from read_his
    d_step = (Max-Min)/(Bins-1);
    for i=1:Bins
        D_VALUE(i)= Min + (i-1)*d_step;
    end;
end; % if TYPE
%D_VALUE involves discrete values between Min and Max
%for each freq create the histogram
for i = 1:length(freq)
    HOW_MANY_TIMES=freq(i);
    D_DISTR=[D_DISTR, D_VALUE(i) * ones(1, HOW_MANY_TIMES)];
end;
PROB_F=freq/Total;

```

Ve skriptu `crea_his.m` je obsažen příkaz `read_his` naprogramovaný ve skriptu souboru `read_his.m` s následujícím obsahem:

```

function [Min, Max, Bins, Total, freq, Type]=read_his(FILENAME)
% Read a histogram from file
% function [Min, Max, Bins, Total, freq, Type]=read_his(FILENAME)
%
% INPUT
%=====
% FILENAME - Histogram name
%
% OUTPUT
%=====
% Min - Histogram minimal value
% Max - Histogram maximal value
% Bins - Number of histogram bins
% Total - Total number of samples
% freq - Frequencies of particular bin
% Type - Histogram type: 0 - Discrete, 1 - Piece-wise uniform

% Author: Pavel PRAKS, pavel.praks@vsb.cz
% Edited by: Petr KONECNY, petr.konecny@vsb.cz
% VSB - Technical University Ostrava, www.vsb.cz, Czech Republic
% 04-11-2011

```

```

PERMISSION='r';
fid = fopen(FILENAME,PERMISSION);

Type = 0; %

while 1
    line = fgets(fid);
    co=findstr('Type=Continuous',line);
    if length(co)~0;
        Type=1;
    else
        co=findstr('ontinuous',line);
        if length(co)~0;
            Type=1;
        end,
    end;
    co=findstr('Min',line);
    if length(co)~0; eval(strcat(line,',')); end;
    co=findstr('Max',line);
    if length(co)~0; eval(strcat(line,',')); end;
    co=findstr('Bins',line);
    if length(co)~0; eval(strcat(line,',')); end;
    co=findstr('Total',line);
    if length(co)~0; eval(strcat(line,',')); end;
    if findstr('[Bins]',line); break; end;
end;
freq=[];
while ~feof(fid)
    line = setstr(fgets(fid));
    freq=[freq, sscanf(line,'%d')];
end;
for i = 2 : Bins
    freq(i);
end
fclose(fid);

```

Skript řešení úlohy obsahuje rovněž příkaz `crea_sa2`, který je naprogramován v následujícím souboru `crea_sa2.m`:

```

function D_RAND=crea_sa2(D_DISTR,RND,D_VALUE,TYPE)
% Create random realization of D_DISTR
%
% INPUT
%=====

```

```

% D_DISTR - Distribution Function
% RND - input random number in interval <0;1>
% TYPE - Histogram TYPE: 0 - Discrete, 1 - Piece-wise uniform

% OUTPUT
%=====
% D_RAND - random realisation from D_DISTR using RND

% Author: Pavel PRAKS, pavel.praks@vsb.cz
% Edited by: Petr KONECNY, petr.konecny@vsb.cz
% VSB - Technical University Ostrava, www.vsb.cz, Czech Republic
% 04-11-2005

D_RAND=[];
LEN_DISTR=length(D_DISTR);
D_STEP = D_VALUE(2) - D_VALUE(1);

%CONTINUOUS HISTOGRAM
%-----
if TYPE == 1
    ORDER = 1+ RND *LEN_DISTR; % Order in Distribution Function
    ORDER_F = fix(ORDER); % Rounded order
    % Reminder after rounding utilised
    % in generation of Piece-Wise histogram (PSEUDOGENERATOR);
    ORDER_R = ORDER - ORDER_F;
    D_RAND = D_DISTR(ORDER_F) +(ORDER_R - 0.5) * D_STEP;

%DISCRETE HISTOGRAM
%-----
else
    D_RAND=D_DISTR(fix( 1+ RND *LEN_DISTR ) );
end % if TYPE

```

Textový výstup řešené úlohy pak může nabývat následující podobu:

```

*****
*           Generování (pseudo)náhodné veličiny           *
*       s neparametrickým rozdělením pravděpodobnosti       *
*****

```

Vstupní údaje:

```

-----
Počet simulací N          =      1.0e+006
Název souboru *.dis       = T235FY01.DIS

```

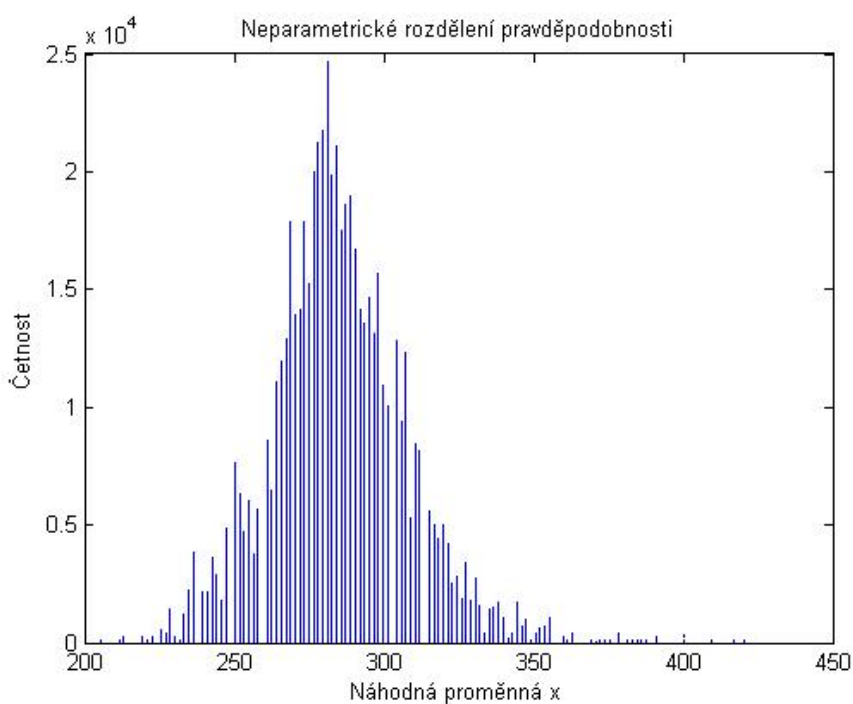


Parametry vygenerované náhodné veličiny:

---

Minimální hodnota	=	205.000
Maximální hodnota	=	421.000
Rozpětí	=	216.000
Střední hodnota	=	285.883
Směrodatná odchylka	=	23.540
Rozptyl	=	554.147
Variační koeficient	=	8.23%
Šikmost	=	0.577
Špičatost	=	4.685
Medián	=	284.000
Kvantil 5%	=	249.000
Percentil 5%	=	249.000
Pravděpodobnost ( $X < 235$ )	=	0.90%

Grafické zobrazení neparametrického rozdělení pravděpodobnosti z příkladu 6.7 je uvedeno na obrázku 6.3.



Obr. 6.3 Neparametrické rozdělení pravděpodobnosti z příkladu 6.7.



## 6.2 Praviděpodobnostní posouzení nosného prvku

V procesu návrhu konstrukce se provádí řada výpočetních operací, souvisejících s posudkem spolehlivosti jednotlivých konstrukčních částí nebo konstrukce jako celku. Musí být splněna různá kritéria spolehlivosti, definovaná příslušnými normovými předpisy.

### 6.2.1 Účinek zatížení

Zatížení je jednou z nejvýznamnějších veličin, která vstupuje do procesu posudku spolehlivosti konstrukcí. Při stanovení účinků zatížení v pravděpodobnostních výpočtech je nutno brát v úvahu zdroj zatížení, způsob působení na konstrukci, intenzitu, směr, dobu trvání, ale i vliv prostředí - např. změnu teploty či vlhkosti.

Účinek zatížení  $E$  je nutno považovat za náhodnou veličinu zejména vzhledem k náhodné proměnlivosti zatížení v čase a prostoru. Veličina, vyjadřující účinek zatížení, se váže na mezní stav, podle něhož se daný pravděpodobnostní posudek provádí. V případě mezního stavu únosnosti tak může účinek zatížení představovat skutečnou velikost dané vnitřní síly, příp. napětí. U mezního stavu použitelnosti je účinek zatížení zpravidla dán skutečným přetvořením konstrukce.

### 6.2.2 Odolnost konstrukce

Odolnost konstrukce  $R$  je schopnost konstrukce odolávat v daných podmínkách účinkům zatížení. Je závislá zejména na výpočetním modelu, materiálových vlastnostech konstrukce (pevnostní a tuhostní charakteristiky použitých materiálů) a jejich geometrických charakteristikách (tvar, rozměr nosných prvků, průřezové charakteristiky, výrobní a montážní nepřesnosti).

Veličina, vyjadřující odolnost konstrukce, se váže na mezní stav, podle něhož se daný pravděpodobnostní posudek spolehlivosti provádí. V případě mezního stavu únosnosti tak může odolnost konstrukce představovat únosnost v daném namáhání, kterou lze určit na úrovni vnitřní síly nebo daného napětí. U mezního stavu použitelnosti je odolnost konstrukce dána mezním přetvořením konstrukce, příp. přípustnou frekvencí kmitání.

S mezním stavem, v rámci něhož se pravděpodobnostní výpočet provádí, souvisí i samotná tvorba výpočetního modelu. Svou roli přitom hraje použitá metodika výpočtu (teorie 1. či 2. řádu) nebo matematický popis chování materiálu konstrukce (pružné chování materiálu, kdy je limitním stavem dosažení napětí na mezi kluzu, nebo využití plastických vlastností, kdy je limitní plastická únosnost, přípustná velikost trvalé deformace, případně tažnost materiálu).

Na výpočetní model může mít rozhodující vliv i skutečnost, zda je předmětem pravděpodobnostního výpočtu posudek spolehlivosti pouze části nosné konstrukce (prvek, průřez) nebo celý nosný systém.

### 6.2.3 Funkce spolehlivosti

Konstrukce musí být navržena tak, aby odolnost konstrukce  $R$  byla větší než je účinek zatížení  $E$ . S přihlédnutím ke všem nahodilostem v zatížení, výrobním a montážním nepřesnostem a prostředí, v němž konstrukce plní svou funkci, odolnost konstrukce  $R$  i účinek zatížení  $E$  je nutno považovat za veličiny náhodné. Je nezbytné, aby obě veličiny vykazovaly stejný fyzikální rozměr.

Pravděpodobnostní posudek spolehlivosti je založen na podmínce spolehlivosti, kterou lze vyjádřit ve tvaru:

$$R - S > 0, \quad (6.8)$$

kde  $R$  je odolnost konstrukce a  $E$  účinek zatížení. Levá strana nerovnice (6.8) se nazývá funkce spolehlivosti  $RF$  (*reliability function*), ale bývá také označována jako funkce poruchy  $G$  či rezerva spolehlivosti  $Z$ .

Podmínku spolehlivosti (6.8) lze rovněž vyjádřit vztahy:

$$\frac{R}{S} > 1, \quad (6.9)$$

nebo

$$\frac{R}{S} - 1 > 0. \quad (6.10)$$

Nesplnění kterékoliv podmínky spolehlivosti (6.8), (6.9) nebo (6.10) představuje z hlediska spolehlivosti nepříznivý, tzn. poruchový stav, kdy účinek zatížení  $E$  převyšuje velikost odolnosti konstrukce  $R$ .

### 6.2.4 Pravděpodobnost poruchy

Pravděpodobnost poruchy  $p_f$  je možno určit ze vztahu:

$$p_f = P(Z \leq 0) = P(R - S \leq 0). \quad (6.11)$$

Její velikost je ovlivněna částí histogramu rezervy spolehlivosti  $Z$ , pro kterou platí  $Z \leq 0$ . Pravděpodobnost bezporuchového stavu  $p_s$  je pak rovna  $1 - p_f$ .

### 6.2.5 Pravděpodobnostní posudek v Matlabu

**Příklad 6.8.** S využitím funkcí programového systému MATLAB proveďte posudek spolehlivosti ohýbaného nosníku z ocelového profilu IPE120, který je zatížen po celé své délce spojitým zatížením stálým s extrémní hodnotou 1,6 kN/m a dlouhodobým proměnným s extrémní hodnotou 1,2 kN/m. Vygenerujte  $N = 1 \cdot 10^6$  hodnot funkce spolehlivosti  $RF = R - E$ , kde  $R$  je odolnost konstrukce (únosnost za ohybu) a  $E$  je účinek zatížení (největší ohybový moment). Sestrojte histogram funkce spolehlivosti  $RF$  a určete pravděpodobnost poruchy  $p_f$ .



*Řešení.* Požadovaný výpočet lze provést s využitím následujícího skriptu vytvořeného v prostředí programového systému MATLAB:

```
% Posudek spolehlivosti prostého nosníku zatíženého spojitě
clc; clear;
% Zadání vstupních údajů
N=1e6;          % Počet simulací
L=6;           % Délka nosníku
Wy=0.5296e-4;  % Průřezový modul IPE120
filename_fy='T235FY01.DIS';
filename_DL='DEAD1.DIS';
filename_LL='LONG2.DIS';
% Načtení histogramů
[his_fy,prob_fy,d_val_fy,type_fy]=crea_his(filename_fy);
[his_DL,prob_DL,d_val_DL,type_DL]=crea_his(filename_DL);
[his_LL,prob_LL,d_val_LL,type_LL]=crea_his(filename_LL);
% Generování (pseudo)náhodných veličin
Fy=crea_sa2(his_fy,rand(1,N),d_val_fy,type_fy);
DL=crea_sa2(his_DL,rand(1,N),d_val_DL,type_DL);
LL=crea_sa2(his_LL,rand(1,N),d_val_LL,type_LL);
% Výpočet účinku zatížení - Mmax
E=1/8*(1.6*DL+1.2*LL)*L^2;
% Výpočet odolnosti konstrukce - únosnosti za ohybu
R=Fy*Wy*1000;
% Výpočet funkce spolehlivosti
RF=R-E;
% Vykreslení výsledků
[n1,ctr1]=hist(E,100);
[n2,ctr2]=hist(R,100);
[n3,ctr3]=hist(RF,100);
figure(1);
hold on;
subplot(2,2,2);
plot(E,R,'.', 'MarkerSize',1);
hold on;
plot([min(min(E),min(R)) max(max(E),max(R))],...
      [min(min(E),min(R)) max(max(E),max(R))], 'r-');
axis([min(E)*0.9 max(E)*1.1 min(R)*0.9 max(R)*1.1]);
title('Pravděpodobnostní posudek spolehlivosti nosníku');
xlabel('Účinek zatížení E');
ylabel('Odolnost konstrukce R');
subplot(2,2,4);
bar(ctr1,-n1,1);
```

```

axis([0.9*min(ctr1) 1.1*max(ctr1) -max(n1)*1.1 0]);
axis('off');
subplot(2,2,1);
barh(ctr2,-n2,1);
axis([-max(n2)*1.1 0 0.9*min(ctr2) 1.1*max(ctr2)]);
axis('off');
subplot(2,2,3);
bar(ctr3,n3,1);
axis([min(ctr3) max(ctr3) min(n3) max(n3)]);
title('Histogram funkce spolehlivosti')
xlabel('RF=R-E')
ylabel('Četnost')
colormap([.8 .8 1]);
hold off;
% Vypis výsledků
disp(sprintf('\n'))
disp('*****')
disp('*          Pravděpodobnostní posudek spolehlivosti          *')
disp('*****')
disp(' ')
disp('Vstupní údaje:')
disp('-----')
disp(sprintf('Počet simulací N          = %12.1e',N));
disp(' ')
disp('Parametry funkce spolehlivosti RF:')
disp('-----')
disp(sprintf('Minimální hodnota          = %12.3f',min(RF)));
disp(sprintf('Maximální hodnota          = %12.3f',max(RF)));
disp(sprintf('Rozpětí                      = %12.3f',range(RF)));
disp(sprintf('Střední hodnota             = %12.3f',mean(RF)));
disp(sprintf('Směrodatná odchylka        = %12.3f',std(RF)));
disp(sprintf('Rozptyl                     = %12.3f',var(RF)));
disp(sprintf('Variační koeficient         = %11.2f%%',...
    std(RF)/mean(RF)*100));
disp(sprintf('Šikmost                      = %12.3f',skewness(RF)));
disp(sprintf('Špičatost                   = %12.3f',kurtosis(RF)));
disp(sprintf('Medián                      = %12.3f',median(RF)));
disp(sprintf('Kvantil 5%%                 = %12.3f',...
    quantile(RF,0.05)));
disp(sprintf('Percentil 5%%              = %12.3f',prctile(RF,5)));
Pf=sum(RF<0)/N;          % Pravděpodobnost poruchy
disp(sprintf('Pravděpodobnost poruchy = %12.4e',Pf));
Beta=mean(RF)/std(RF); % Index spolehlivosti

```

```

disp(sprintf('Index spolehlivosti Beta = %12.4f',Beta));
disp(' ')
disp('*****')
if Pf<=8.4e-6
    disp(sprintf('*          Pf    = %12.4e <= Pd    = %12.4e          *',...
        Pf,8.4e-6));
    disp(sprintf('*          Beta = %12.4f >= Betad = %12.4f          *',...
        Beta,4.3));
    disp('*          Konstrukce vyhoví podle ČSN EN 1990          *');
    disp('*    pro třídu spolehlivosti RC3/CC3 s velkými následky    *');
elseif Pf<=7.2e-5
    disp(sprintf('*          Pf    = %12.4e <= Pd    = %12.4e          *',...
        Pf,7.2e-5));
    disp(sprintf('*          Beta = %12.4f >= Betad = %12.4f          *',...
        Beta,3.8));
    disp('*          Konstrukce vyhoví podle ČSN EN 1990          *');
    disp('*    pro třídu spolehlivosti RC2/CC2 se středními následky *');
elseif Pf<=4.8e-4
    disp(sprintf('*          Pf    = %12.4e <= Pd    = %12.4e          *',...
        Pf,4.8e-4));
    disp(sprintf('*          Beta = %12.4f >= Betad = %12.4f          *',...
        Beta,3.3));
    disp('*          Konstrukce vyhoví podle ČSN EN 1990          *');
    disp('*    pro třídu spolehlivosti RC1/CC1 s malými následky    *');
else
    disp(sprintf('*          Pf    = %12.4e > Pd    = %12.4e          *',...
        Pf,4.8e-4));
    disp(sprintf('*          Beta = %12.4f < Betad = %12.4f          *',...
        Beta,3.3));
    disp('*          Konstrukce nevyhoví podle ČSN EN 1990          *');
end
disp('*****')

```

Textový výstup úlohy pak může vypadat následovně:

```

*****
*          Pravděpodobnostní posudek spolehlivosti          *
*****

```

Vstupní údaje:

-----  
Počet simulací N                    =            1.0e+006

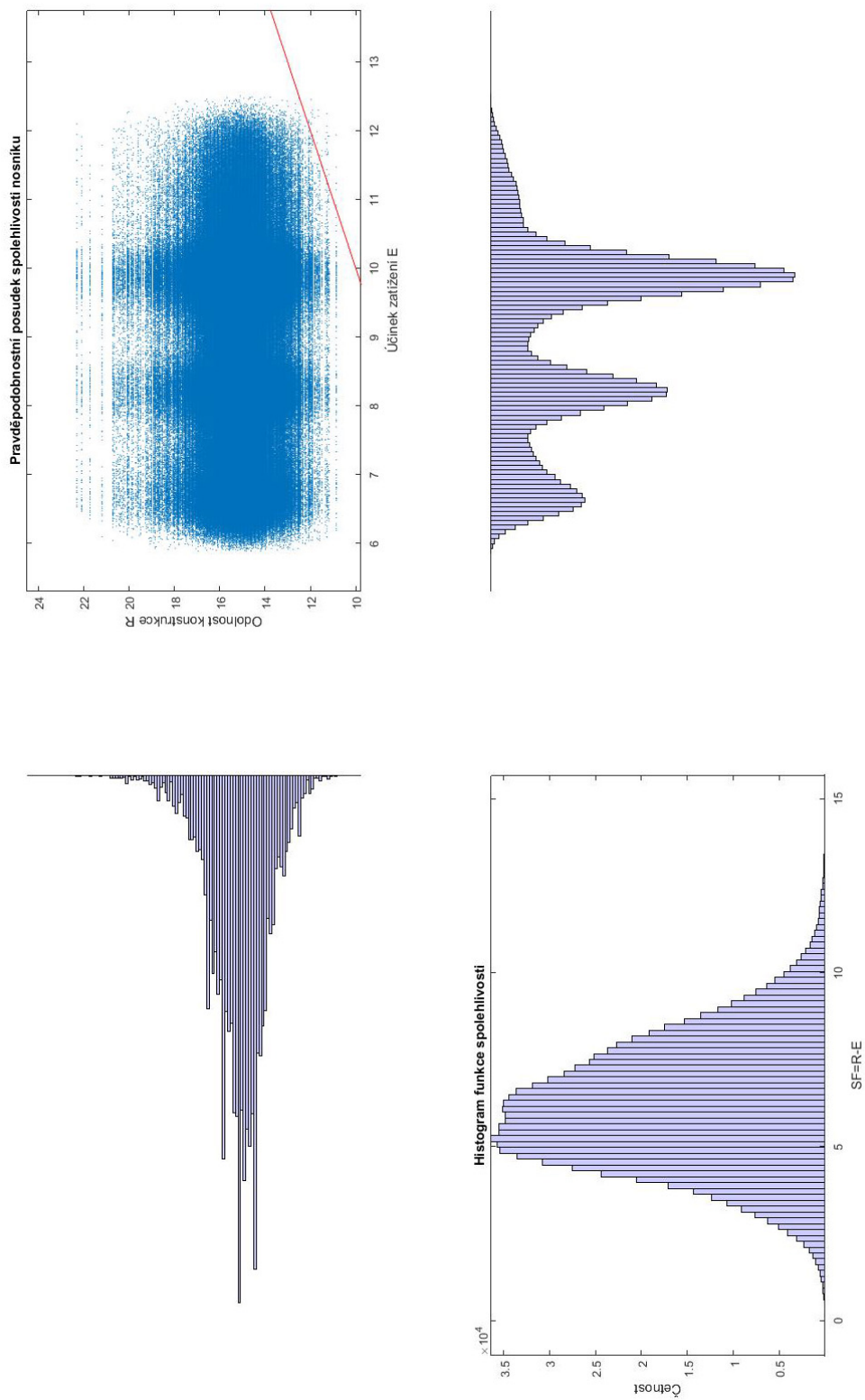
Parametry funkce spolehlivosti RF:

```
-----
Minimální hodnota      =      -0.983
Maximální hodnota      =      16.118
Rozpětí                 =      17.100
Střední hodnota        =       6.180
Směrodatná odchylka    =       1.820
Rozptyl                 =       3.313
Variační koeficient     =      29.45%
Šikmost                 =       0.305
Špičatost              =       3.149
Medián                  =       6.065
Kvantil 5%             =       3.384
Percentil 5%           =       3.384
Pravděpodobnost poruchy = 5.3000e-005
Index spolehlivosti Beta =       3.3950
```

```
*****
*      Pf   = 5.3000e-005 <= Pd   = 7.2000e-005      *
*      Beta =      3.3950 >= Betad =      3.8000      *
*      Konstrukce vyhoví podle ČSN EN 1990          *
*      pro třídu spolehlivosti RC2/CC2 se středními následky *
*****
```

Grafické zobrazení pravděpodobnostního posouzení nosníku z příkladu 6.8 je uvedeno na obrázku 6.4, který obsahuje histogramy odolnosti konstrukce  $R$ , účinku zatížení  $E$  a funkce spolehlivosti  $SF$  včetně prostorové interpretace s hranicí poruchy.

▲



Obr. 6.4 Grafické zobrazení pravděpodobnostního posouzení nosníku z příkladu 6.8.



# Literatura



- [1] *Algoritmus*. Webové stránky zaměřené na tvorbu algoritmů. [on-line]. <<http://www.algoritmy.net>>.
- [2] Cook, R.D. – Malkus, D.S. – Plesha, M.E. – Witt, R.J. *Concepts and Applications of Finite Element Analysis*. 4. vydání. Wiley, 2007. (736 s).
- [3] Eaton, J.W. *Octave*. Programový systém pro provádění matematických výpočtů. Freeware, verze 3.8.2. [on-line]. <<http://www.gnu.org/software/octave>>. University of Wisconsin, Department of Chemical Engineering, 1998-2014.
- [4] Janas, P. – Krejsa, M. – Krejsa, V. *Přímý Optimalizovaný Pravděpodobnostní Výpočet*. [on-line]. <<https://dspace.vsb.cz/bitstream/handle/10084/110988/monografie.pdf>>. Monografie, 1. vydání. VŠB-TU Ostrava, 2015. (191 s). ISBN 978-80-248-3798-7.
- [5] Just, M. *Octave — český průvodce programem*. Elektronický manuál programového systému Octave. [on-line]. <<http://www.octave.cz>>. Univerzita Tomáše Bati ve Zlíně, Fakulta aplikované informatiky, 2006.
- [6] Krejsa, M. – Konečný, P. *Spolehlivost a bezpečnost staveb*. [on-line]. <<http://mi21.vsb.cz/modul/spolehlivost-bezpecnost-staveb>>. Učební texty a interaktivní materiály. VŠB-TU Ostrava, 2012.
- [7] Kučera, R. *Numerické metody*. Učební texty. VŠB-TU Ostrava. (152 s). ISBN 80-248-1198-7.
- [8] Materna, A. — Štěpánek, P. — Teplý, B. *Automatizace inženýrských úloh*. Skriptum. Vysoké učení technické v Brně, 1985. (132 s).
- [9] MATLAB. Programový systém pro provádění matematických výpočtů. Komerční software, verze R2014b. [on-line]. <<http://www.mathworks.com>>. The MathWorks, únor 2015.
- [10] Melcer, J. – Kuchárová, D. *Dynamika stavebních konstrukcí – Příklady*. 1. vydání. Žilinská univerzita v Žilině, 2004. (300 s). ISBN 80-8070-326-4.

- [11] Melcer, J. – Lajčáková, G. *Aplikácie programového systému Matlab pri riešení úloh dynamiky stavebných konštrukcií*. 1. vydání. Žilinská univerzita v Žilině, 2011. (166 s). ISBN 978-80-554-0308-3.
- [12] Ralston, A. *Základy numerické matematiky*. 1. vydání. Academia, Praha, 1973. (635 s).
- [13] Rektorys, K. *Přehled užité matematiky*. 4. vydání. SNTL - Nakladatelství technické literatury, Praha, 1981. (1140 s).
- [14] Sauer T. *Numerical Analysis*. George Mason University. Pearson Education, Inc., 2006. (669 s). ISBN 0-321-26898-9.
- [15] Sigmon K. *MATLAB Primer CZ*. Elektronický manuál programového systému MATLAB. Druhé vydání. [on-line]. <<https://artax.karlin.mff.cuni.cz/~beda/cz/matlab/primer.cz/matlab-primer.html>>. Department of Mathematics, University of Florida, 1989, 1992. Z anglického originálu přeložil Petr Klášterecký.
- [16] *Wikipedia*. Otevřená internetová encyklopedie. Webové stránky. [on-line]. <<http://cs.wikipedia.org>>.