VYSOKÉ UČENÍ TECHNICKÉ
V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA STAVEBNÍ
FACULTY OF CIVIL ENGINEERING

ÚSTAV STAVEBNÍ MECHANIKY
INSTITUTE OF STRUCTURAL MECHANICS

METODY ANALÝZY STATICKÉ STABILITY
METHODS OF STATIC BUCKLING ANALYSIS

DIPLOMOVÁ PRÁCE
DIPLOMA THESIS

AUTOR PRÁCE — Bc. Filip Svoboda
AUTHOR

VEDOUCÍ PRÁCE — doc. Ing. PETR FRANTÍK, Ph.D.
SUPERVISOR

BRNO 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA STAVEBNÍ

Studijní program	N3607 Stavební inženýrství
Typ studijního programu	Navazující magisterský studijní program s prezenční formou studia
Studijní obor	3608T001 Pozemní stavby
Pracoviště	Ústav stavební mechaniky

ZADÁNÍ DIPLOMOVÉ PRÁCE

Student	Bc. Filip Svoboda
Název	Metody analýzy statické stability
Vedoucí práce	doc. Ing. Petr Frantík, Ph.D.
Datum zadání	31. 3. 2016
Datum odevzdání	13. 1. 2017

V Brně dne 31. 3. 2016

prof. Ing. Drahomír Novák, DrSc.
Vedoucí ústavu

prof. Ing. Rostislav Drochytka, CSc., MBA
Děkan Fakulty stavební VUT

PODKLADY A LITERATURA

Literatura dle pokynů vedoucího práce.

Macur, J., Úvod do teorie dynamických systémů a jejich simulace, skripta, nakladatelství PC-DIR, Brno, 1995

Brepta, R., Půst, L., Turek, F.: Mechanické kmitání, Technický průvodce 71, nakladatelství Sobotáles,
Praha, 1994.

ZÁSADY PRO VYPRACOVÁNÍ

Nastudování potřebných znalostí dle pokynů vedoucího práce. Zorientování se v problematice. Vytvoření numerických modelů, jejich ověření a aplikace.

STRUKTURA DIPLOMOVÉ PRÁCE

VŠKP vypracujte a rozčleňte podle dále uvedené struktury:

1. Textová část VŠKP zpracovaná podle Směrnice rektora "Úprava, odevzdávání, zveřejňování a uchovávání vysokoškolských kvalifikačních prací" a Směrnice děkana "Úprava, odevzdávání, zveřejňování a uchovávání vysokoškolských kvalifikačních prací na FAST VUT" (povinná součást VŠKP).
2. Přílohy textové části VŠKP zpracované podle Směrnice rektora "Úprava, odevzdávání, zveřejňování a uchovávání vysokoškolských kvalifikačních prací" a Směrnice děkana "Úprava, odevzdávání, zveřejňování a uchovávání vysokoškolských kvalifikačních prací na FAST VUT" (nepovinná součást VŠKP v případě, že přílohy nejsou součástí textové části VŠKP, ale textovou část doplňují).

_____

doc. Ing. Petr Frantík, Ph.D.
Vedoucí diplomové práce

Abstrakt

Cílem předkládané práce je vytvoření počítačové aplikace, která spočítá kritické zatížení pro ztrátu stability rovinných prutových konstrukcí pomocí metody konečných prvků. Úvod je věnován seznámení se s problematikou a odvození nezbytných vztahů. Poté jsou popsány všechny důležité kroky a numerické metody nezbytné pro správný chod aplikace. Nakonec je provedena analýza několika vybraných úloh a výsledky jsou porovnány se známými analytickými řešeními a s dalšími dostupnými aplikacemi.

Klíčová slova

Lineární stabilita, vzpěr, metoda konečných prvků, c#, programování, vlastní čísla

Abstract

The aim of this theses is to create application, which is able to calculate buckling load of structure made from 1D bar elements, using finite element method. introduction is devoted to basic principles of buckling and derivation of necessary formulas. Then are described all operations and numerical methods needed for the application. At the and is in detail analyzed few structures and results are compared with known solutions or with other applications.

Keywords

Linear stability, buckling, finite element method, c#, programing, eigenvalue

BIBLIOGRAFICKÁ CITACE VŠKP

Bc. Filip Svoboda Metody analýzy statické stability. Brno, 2017. 39 s., 1 s. příl. Diplomová práce. Vysoké učení technické v Brně, Fakulta stavební, Ústav stavební mechaniky. Vedoucí práce doc. Ing. Petr Frantík, Ph.D.

PROHLÁŠENÍ

Prohlašuji, že jsem diplomovou práci zpracoval samostatně a že jsem uvedl všechny použité informační zdroje.

V Brně dne 12. 1. 2017

.....
Bc. Filip Svoboda

PROHLÁŠENÍ

Prohlašuji, že elektronická forma odevzdané diplomové práce je shodná s odevzdanou listinnou formou.

V Brně dne 12. 1. 2017

.....
Bc. Filip Svoboda

Poděkování

Na tomto místě bych rád poděkoval všem, kteří mě podporovali při psaní této práce, zejména pak doc. Ing. Petru Frantíkovi, Ph.D. za výborné vedení, odborné rady a skvělý osobní přístup.

Obsah

Kapitola 1

Úvod

Konstrukce navrhujeme tak, aby byly schopné přenést veškeré zatížení, které na ně působí. Pokud dojde k poruše konstrukce, tak to bývá ze dvou důvodů. Buď kvůli poruše materiálu, ať už se jedná o překročení meze pevnosti nebo meze kluzu a nebo kvůli ztrátě stability. V prvním případě v konstrukci vznikají taková napětí, která vedou k nenávratným poruchám materiálu. Zatímco ve druhém případě mohou být napětí v přípustných hodnotách, ale konstrukce není schopna setrvat v rovnovážném stavu, dojde ke ztrátě stability a tedy ke kolapsu konstrukce.

1.1 Stabilita konstrukce

Ztráta stability nejčastěji nastává při namáhání tlakem, ohybem nebo krutem. S úlohami ztráty stability se setkáváme u štíhlých konstrukcí, tenkostěnných nosníků nebo u otevřených, či uzavřených profilů.

1.2 Cíle práce

V následující práci se zaměříme na hledání kritického zatížení pro ztrátu stability rovinných prutových konstrukcí, ke které typicky dochází při vzpěrném namáhání. Pro tento účel má být vyvinuta počítačová aplikace, která bude řešit tento problém pomocí metody konečných prvků v rámci malých deformací s využitím teorie druhého řádu. Výsledky řešení budou porovnány se známými analytickými výrazy, popřípadě s výsledky výpočtu pomocí jiné metody či aplikace.

Kapitola 2

Stabilita

2.1 Vzpěr Prutu

Nejběžnější inženýrskou úlohou se ztrátou stability je vzpěr prutu. Jeho řešení v rámci teorie druhého řádu s uvažováním malých deformací spočívá v hledání tlakového zatížení, při kterém původní tvar prutu pozbývá stabilitu a výslednou deformaci nelze z podmínek rovnováhy určit. Při tomto zatížení P, které se nazývá kritické, těleso setrvává buďto v původním přímém nebo deformovaném stavu. Pokud je tedy těleso zatíženo kritickým zatížením P_{cr} existují současně dvě rovnovážné konfigurace. Tento stav se nazývá bifurkace.

Jako první odvodil vztah pro kritickou sílu kloubově uloženého prutu s konstantním průřezem švýcarský matematik a fyzik Leonhard Euler v roce 1757, viz [11]. Tento velmi dobře známý vztah má podobu:

$$P_{cr} = \frac{\pi^2 EI}{L^2}, \tag{2.1}$$

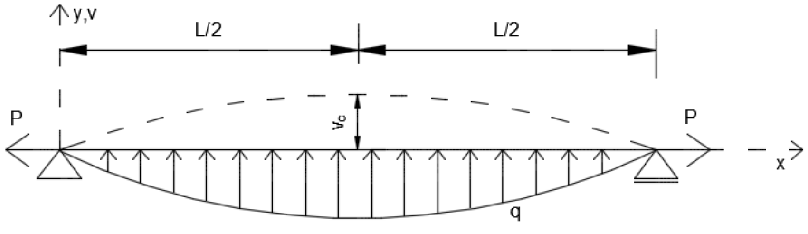
kde E je modul pružnosti materiálu prutu, I je moment setrvačnosti průřezu a L je délka prutu. Existuje několik způsobů, jak k tomuto vztahu dojít. Například pomocí energetického principu, ze kterého jde následně odvodit výpočet pomocí metody konečných prvků, viz [2].

2.2 Energetický princip

Z hlediska energetického principu můžeme na problematiku pohlížet tak, že ke vzpěru dojde, když těleso přemění osovou deformační energii na ohybovou beze změny působícího zatížení. Běžné stavební konstrukce mají osovou tuhost daleko větší než ohybovou, velká deformační energie tedy může být akumulována i při malých deformacích a když dojde ke vzpěru, tak poměrně velké ohybové deformace jsou potřebné k absorbování uvolněné energie.

2.2.1 Odvození

Pro ilustraci si odvodíme kritickou sílu na prostém nosníku pomocí minima potenciální energie, viz [2].

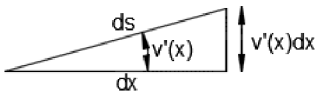


Obrázek 2.1: Prostý nosník

Představme si prostý nosník zatížený na koncích osovou silou P a příčným zatížením q, viz obr. 2.1. Pro malá příčné deformace $v = v(x)$ můžeme potenciální energii deformace U_b vyjádřit pomocí vztahu:

$$U_b = \frac{1}{2}\int_0^L EI(v''(x))^2 dx \tag{2.2}$$

Představme si, že přetvoření $v = v(x)$ nastává bez jakékoliv osové deformace u, každý diferenciální element délky dx se prodlouží na délku ds, kde $ds > dx$. Výraz pro ds a jeho aproximace založena na prvních dvou členech binomického rozvoje jsou uvedeny na obrázku 2.2.



$$ds = \sqrt{1 + (v'(x))^2}dx$$
$$ds \approx 1 + \frac{1}{2}(v'(x))^2 dx$$

Obrázek 2.2: Prostý nosník

Poměrné přetvoření ϵ_m v ose prutu tedy můžeme vyjádřit jako:

$$\begin{aligned} \epsilon_m &= \frac{ds - dx}{dx} = \frac{ds}{dx} - 1 \\ \epsilon_m &\approx (1 + \frac{1}{2}(v'(x))^2) - 1 = \frac{1}{2}(v'(x))^2 \end{aligned} \tag{2.3}$$

Pro malé deformace zůstává síla P konstantní. Každý diferenciální element dx se prodlouží o délku $\epsilon_m dx$, síla P tedy koná práci a v prutu se akumuluje deformační energie o velikosti $P\epsilon_m dx$. Změnu osové potenciální energie deformace U_m tedy můžeme vyjádřit jako:

$$\begin{aligned} U_m &= \int_0^L P\epsilon_m dx \\ U_m &= \frac{1}{2}\int_0^L P(v'(x))^2 dx \end{aligned} \tag{2.4}$$

Když budeme uvažovat, že přetvoření v je půlka sinusové vlny, tedy $v = v_c \sin(\pi x/L)$, kde v_c je přetvoření uprostřed prutu, platí:

$$U_b = \frac{\pi^4 EI}{4L^3} v_c^2 \tag{2.5}$$

$$U_m = \frac{\pi^2 P}{4L} v_c^2 \tag{2.6}$$

Pokud i příčné zatížení je půlka sinusové vlny, tedy $q = q_c \sin(\pi x/L)$, pro celkovou potenciální energii platí:

$$\Pi_p = U_m + U_b + \Omega \qquad kde \qquad \Omega = -\int_0^L vq dx = -\frac{q_c L}{2} v_c \tag{2.7}$$

Pro rovnovážný stav soustavy platí $d\Pi/dv = 0$, po zderivování vztahu (2.6) tedy získáme:

$$v_c = \frac{q_c L}{2(k + k_\sigma)} \qquad kde \qquad k = \frac{\pi^4 EI}{2L^3} \qquad a \qquad k_\sigma = \frac{\pi^2 P}{2L} \tag{2.8}$$

Ke vzpěru dojde při takovém zatížení P, když v_c je nenulové, zatímco $q_c = 0$. Jinak řečeno ohybová tuhost je zredukovaná na nulu, když $P = P_{cr}$. A tedy z rovnice $k + k_\sigma = 0$ získáme Eulerův vztah pro kritickou sílu:

$$P_{cr} = -\frac{\pi^2 EI}{L^2} \tag{2.9}$$

Matematicky jsme vypočítali vlastní číslo soustavy:

$$(k + k_\sigma)v_c = 0 \tag{2.10}$$

2.2.2 Vliv napjatosti na tuhost

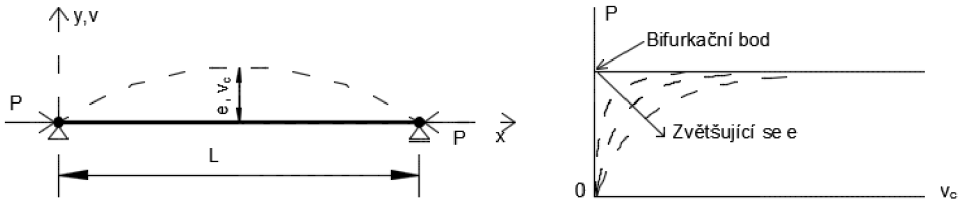
Za povšimnutí jistě stojí vliv normálových sil na průhyb. Jak můžeme vidět ve vztahu (2.8). Když je síla P kladná, tak nám zmenšuje průhyb q_c a zvětšuje tedy ohybovou tuhost prutu. Pokud je záporná, tak je tomu přesně naopak.

2.2.3 Předpoklady

Metoda výpočtu lineární stability má ovšem jisté předpoklady a omezení, které musíme dodržet:

- Elastické chování materiálu prutu - zatěžování probíhá v oblasti platnosti Hookova zákona
- Prut je dokonale přímý a síla působí v ose prutu bez jakýchkoliv excentricit.

Uvažme obr. 2.3, kritická síla $P_{cr} = \pi^2 EI/L^2$ je vypočtená pouze pro $e = 0$. Se zvětšující se počáteční excentricitou e, síla P vyvolává ohybový moment a dochází tedy k deformaci prutu, spíše než ke vzpěru. V takovém případě je daleko výstižnější nelineární stabilitní analýza.



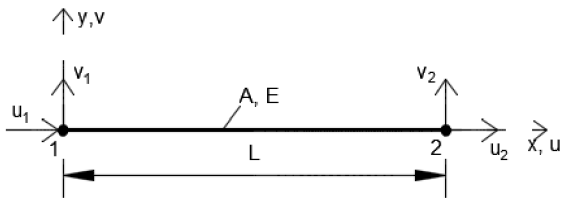
Obrázek 2.3: Prut s počáteční imperfekcí a její vliv stabilitu

2.3 Implementace lineární stability v rámci MKP

2.3.1 Geometrická matice tuhosti

V kapitole 2.2.2 jsme si naznačili, že osová napjatost může mít velký vliv na celkovou tuhost konstrukce. Při výpočtech pomocí metody konečných prvků je tento vliv vyjádřen pomocí geometrické matice $[K_\sigma]$. Členy této matice jsou nezávislé na materiálových vlastnostech a jsou tedy funkcemi pouze geometrie prvku, přetvoření a osových napjatostí.

Pro názornost si ukážeme odvození geometrické matice tuhosti příhradového prutu viz obr. 2.4.



Obrázek 2.4: Příhradový prut a jeho stupně volnosti

Pokud je prut vystaven příčnému posunu v_1 a v_2, tak na koncích prutu musí také působit síly F_1 a F_2, aby nastala rovnováha, jak je ukázáno na obr. 2.5.

Z momentové podmínky rovnováhy tedy platí:

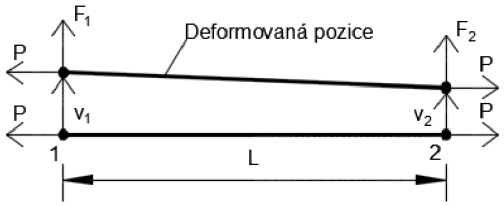
$$F_1 = \frac{P}{L}(v_1 - v_2) \tag{2.11}$$

Ve vertikálním směru platí rovnováha:

$$F_2 = -F_1 \tag{2.12}$$

Kombinací vztahů 2.11 a 2.12 můžeme síly F_1 a F_2 vyjádřit maticovou rovnicí:

$$\begin{Bmatrix} F_1 \\ F_2 \end{Bmatrix} = \frac{P}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}\begin{Bmatrix} v_1 \\ v_2 \end{Bmatrix} \qquad kde \qquad \frac{P}{L}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} = [K_\sigma] \tag{2.13}$$



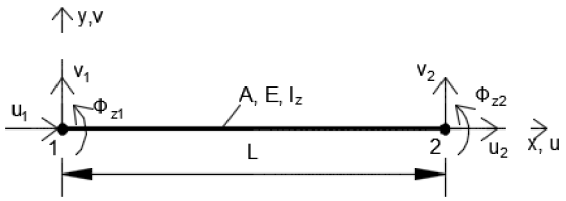
Obrázek 2.5: Síly působící na příhradový prut

Celková matice tuhosti $[K]$ je poté daná součtem materiálové $[K_M]$ a geometrické matice tuhosti $[K_\sigma]$:

$$[K] = [K_M] + [K_\sigma] \tag{2.14}$$

Pro ohýbaný prut ve 2D, viz obr. 2.6, platí geometrická matice tuhosti, viz [2]:

$$[K_\sigma] = \frac{P}{30L}\begin{bmatrix} 36 & 3L & -36 & 3L \\ 3L & 4L^2 & -3L & -L^2 \\ -36 & -3L & 36 & -3L \\ 3L & -L^2 & -3L & 4L^2 \end{bmatrix} \qquad d = \begin{Bmatrix} v_1 \\ \Phi_{z1} \\ v_2 \\ \Phi_{z2} \end{Bmatrix} \tag{2.15}$$



Obrázek 2.6: Ohýbaný prut a jeho stupně volnosti

Běžně neznáme, jaké normálové síly působí na konstrukci. Je tedy zapotřebí výpočet provést ve dvou krocích. Nejprve lineární analýzou spočítáme vnitřní síly, abychom mohli sestavit geometrickou matici a následně spočítáme součinitel kritického zatížení a příslušný vektor vybočení.

2.3.2 Problém vlastních čísel

Jak bylo naznačeno ve vztahu (2.10) výpočet kritického zatížení je z matematického hlediska výpočet vlastních čísel soustavy, viz [7]:

$$([K_M] + \lambda[K_\sigma])\{u\} = 0, \tag{2.16}$$

kde $[K_M]$ je materiálová matice tuhosti, $[K_\sigma]$ je geometrická matice tuhosti, λ je vlastní číslo, tedy součinitel kritického zatížení a $\{u\}$ je vlastní vektor, tedy vektor tvaru ztráty stability.

Netriviální řešení má tato soustava tehdy, když je matice $([K]+\lambda[K_\sigma])$ singulární, tzn. když

$$det|[K_M] + \lambda[K_\sigma]| = 0 \tag{2.17}$$

Po získání λ následně vypočteme příslušný vlastní vektor $\{u\}$ zpětným dosazením do (2.16). Je důležité si uvědomit, že řešení vyhovuje jakýkoliv vektor, který je násobkem vypočítaného vlastního vektoru $\{u\}$, jedná se tedy pouze o tvar, který následně vhodným způsobem normalizujeme.

Rovnice (2.16) představuje řešení polygonu n-tého řádu, kde n je řád matic $[K]$ a $[K_\sigma]$. Existuje tedy n řešení, u stabilitních úloh má zpravidla největší význam to nejmenší vlastní číslo, ale jsou samozřejmě případy, kdy nás zajímají i vyšší vlastní čísla, např. když vyšetřujeme globální stabilitu složitější konstrukce, ale při nejmenším kritickém zatížení vybočuje pouze jeden lokální prut.

Jedna z možností, jak vyřešit rovnici (2.16) je tedy najít kořeny polygonu n-tého řádu. Tato metoda však není velmi efektivní a proto se využívají rychlejší a spolehlivější iterační metody jako je např. mocninná metoda, iterace podprostoru, či Lanczosova metoda, viz [2]. Pomocí těchto metod lze vypočítat požadovaný počet nejmenších vlastních čísel a vektorů i na soustavách s velkým počtem rovnic.

Kapitola 3

Program

V kapitole 2 jsme si ukázali, jak se vypočítá stabilita prutové konstrukce. Nyní přejdeme k vytvoření programu, který nám metodou konečných prvků spočítá kritické zatížení konstrukce a tvar ztráty stability. Ukážeme si tedy některé kroky a metody, které jsou stěžejní pro výpočet.

3.1 Využití jazyka C#

Program je vytvořen v programovacím jazyce C#, viz [5]. Je to moderní vysokoúrovňový objektově orientovaný programovací jazyk vyvinutý firmou Microsoft a jeho první verze byla vydána v roce 2002. C# lze využít k tvorbě databázových programů, webových aplikací a stránek, webových služeb, formulářových aplikací ve Windows atd.

Jazyk C# je tedy, jako většina moderních programovacích jazyků, objektově orientovaný. To znamená, že program je poskládán z jednotlivých objektů, díky kterým je zdrojový kód přehlednější a jednodušší na údržbu.

3.2 Model

Nejprve si definujeme veškeré potřebné parametry pro sestavení modelu:

1. Body

```
// X souradnice ve 2D prostoru [m]
public double X { get; set; }
// Z souradnice ve 2D prostoru [m]
public double Z { get; set; }

// Promenna support, ktera nam rika jestli je bod podepren
// a pokud ano, tak ktere stupne volnosti jsou mu odebrany
public Support2D Support { get; set; }
```

2. Podpory

```
// Kazde podpore muze byt odebran kterykoliv stupen volnosti
public SupportType Fx { get; set; }
public SupportType Fz { get; set; }
public SupportType Ry { get; set; }


public enum SupportType
{
    Free,
    Rigid
}
```

3. Material

```
// Jmeno
public string Name { get; set; }
// Modul pruznosti [Pa]
public double E { get; set; }
```

4. Průřez

```
// Jmeno
public string Name { get; set; }
// Plocha [m2]
public double A { get; set; }
// Moment setrvacnosti [m4]
public double Iy { get; set; }
// Material
public Material Material { get; set; }
```

5. Prut

```
// Pocatecni bod prutu
public Node2D Node1 { get; set; }
// Koncovy bod prutu
public Node2D Node2 { get; set; }
// Prurez prutu
public CrossSection CrossSections { get; set; }
// True pokud je na zacatku prutu kloub
public bool HingeBegin { get; set; }
// True pokud je na konci prutu kloub
public bool HingeEnd { get; set; }
```

6. Zatížení

```
// Zatizeni muze byt definovano pouze v uzlu jako sila nebo
// moment

// Smer sily/moment
public Direction Direcion { get; set; }
// Velikost sily/momentu
public double Magnitude { get; set; }
// Bod ve kterem pusobi
public Node2D Node2D { get; set; }

public enum Direction
{
    X,
    Z,
    My
}
```

3.3 Výpočet

Když máme sestaven model, můžeme přejít k výpočtu.

Nejdříve provedeme diskretizaci modelu. Pruty si tedy nadělíme na jednotlivé konečné prvky podle požadované velikosti.

3.3.1 Lineární analýza

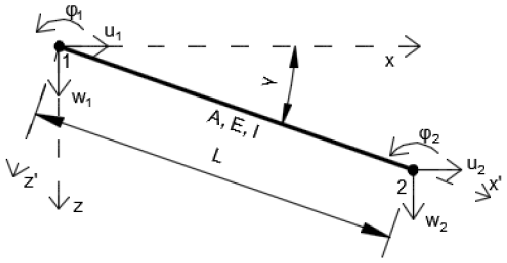
Pro každý konečný prvek si vypočítáme jeho délku. Jelikož známe jeho počáteční a koncový bod $A[x_1, y_1]$ a $B[x_2, y_2]$, můžeme si spočítat souřadnice vektoru $\overrightarrow{u}$:

$$\overrightarrow{u} = B - A = [x_2 - x_1, y_2 - y_1] \tag{3.1}$$

Následně délku L pomocí Pythagorovy věty:

$$L = |\overrightarrow{u}| = \sqrt{u_1^2 + u_2^2} \tag{3.2}$$

A pomocí goniometrických funkcí si vypočítáme úhel γ, který nám říká, jak moc je prut pootočen od vodorovné polohy viz obr. 3.1.



Obrázek 3.1: Oboustranně monoliticky připojený prut

Nyní si vypočítáme globální a lokální matice tuhosti jednotlivých prvků. Pro oboustranně monoliticky připojený prut, viz obr. 3.1, platí následující lokální matice tuhosti:

$$[K_{1,2}] = \begin{bmatrix} \frac{EA}{L} & 0 & 0 & -\frac{EA}{L} & 0 & 0 \\ 0 & \frac{12EI}{L^3} & -\frac{6EI}{L^2} & 0 & -\frac{12EI}{L^3} & -\frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{4EI}{L} & 0 & \frac{6EI}{L^2} & \frac{2EI}{L} \\ -\frac{EA}{L} & 0 & 0 & \frac{EA}{L} & 0 & 0 \\ 0 & -\frac{12EI}{L^3} & \frac{6EI}{L^2} & 0 & \frac{12EI}{L^3} & \frac{6EI}{L^2} \\ 0 & -\frac{6EI}{L^2} & \frac{2EI}{L} & 0 & \frac{6EI}{L^2} & \frac{4EI}{L} \end{bmatrix} \tag{3.3}$$

Sestavíme globální matici $[K]$ a globální zatěžovací vektor $\{F\}$ a vypočítáme vektor globálních deformací $\{u\}$ z rovnice:

$$[K]\{u\} = \{F\} \tag{3.4}$$

Jedná se o nehomogenní soustavu lineárních algebraických rovnic. Dle Frobeniovy věty, viz [13], platí:

Nehomogenní soustava lineárních algebraických rovnic $A\overrightarrow{x} = \overrightarrow{b}$ má řešení pouze v případě, že hodnost matice soustavy $h(A)$ je rovna hodnosti rozšířené matice soustavy $h(A|\overrightarrow{b})$. Pokud je $h(A)$ rovno počtu neznámých, má soustava jedno řešení; pokud je $h(A)$ menší než počet neznámých, je řešení nekonečně mnoho (je-li větší než počet neznámých, nemůže být splněna předchozí podmínka a soustava tedy nemá řešení).

Když sestavíme rovnici (3.4), tak má nekonečně mnoho řešení, protože díky okrajovým podmínkám víme, že některé posuny, či pootočení jsou nulové a tedy hodnost matice $h(A)$ je menší než počet neznámých. Této situaci předejdeme, když z matice odstraníme příslušné nulové řádky a sloupce, poté bude mít soustava právě jedno řešení.

Existuje mnoho numerických metod, jak vyřešit soustavu lineárních algebraických metod např. Gaussova eliminační metoda nebo LU rozklad, viz [13]. Vzhledem k tomu, že matice tuhosti $[K]$ je symetrická můžeme využít Choleského rozkladu, viz [9].

Choleského rozklad

Základním předpokladem pro Choleského rozklad tedy je, že matice $[A]$ je symetrická a poté platí:

$$[A] = [L][L]^T, \tag{3.5}$$

kde $[L]$ je trojúhelníková matice a soustavu $[A]\{x\} = \{b\}$ lze vypočítat vyřešením dvou soustav rovnic:

$$\begin{aligned}[L]\{y\} &= \{b\} \\ [L]^T\{x\} &= \{y\}\end{aligned} \tag{3.6}$$

Prvky matice $[L]$ je možné počítat po sloupcích zleva a v každém sloupci odshora dolů. Při rozepsání několika členů je vidět opakující se vzor, který lze využít pro naprogramování celého rozkladu.

Pro první sloupec platí následující:

$$\begin{aligned} a_{11} = l_{11}l_{11} &\longrightarrow l_{11} = \sqrt{a_{11}} \\ a_{21} = l_{21}l_{11} &\longrightarrow l_{21} = a_{21}/l_{11} \\ &\vdots \\ a_{n1} = l_{n1}l_{11} &\longrightarrow l_{n1} = a_{n1}/l_{11} \end{aligned} \tag{3.7}$$

Pro druhý sloupec platí:

$$\begin{aligned} a_{22} = l_{21}l_{21} + l_{22}l_{22} &\longrightarrow l_{22} = \sqrt{a_{22} - l_{21}^2} \\ a_{32} = l_{31}l_{21} + l_{32}l_{22} &\longrightarrow l_{32} = (a_{32} - l_{31}l_{21})/l_{22} \\ &\vdots \\ a_{n2} = l_{n1}l_{21} + l_{n2}l_{22} &\longrightarrow l_{n2} = (a_{n2} - l_{n1}l_{21})/l_{22}, \end{aligned} \tag{3.8}$$

kde a_{ij} jsou prvky matice $[A]$ a l_{ij} jsou prvky matice $[L]$. Celé řešení výpočtu deformací zapsané v programovacím jazyce c# vypadá následovně:

```
// Staticka metoda do ktere vstupuje glbalni matice tuhosti a globalni
    vektor zatizeni
public static double[] Decomposition(double[,] GlobalMatrixCalc,
    double[] GlobalVectorCalc)
{

// Globalni matice tuhosti
double[,] A = GlobalMatrixCalc;

//Globalni zatezovaci vektor
double[] b = GlobalVectorCalc;

// Trojuhelnikova matice, ktera vznikne Choleskeho rozkladem
double[,] chol = new double[GlobalMatrixCalc.GetLength(0),
    GlobalMatrixCalc.GetLength(0)];

// Pomocny vektor pro vypocet deformaci
double[] d = new double[GlobalMatrixCalc.GetLength(0)];

// Vektor spocitanych deformaci
double[] resultChol = new double[GlobalMatrixCalc.GetLength(0)];

// pomocna promenna
int xx = 0;


// Choleskeho rozklad
for (int i = 0; i < A.GetLength(0); i++)
{
    xx++;
    for (int j = 0; j < xx; j++)
```

```
    {
        double sum = 0;

        for (int k = j - 1; k >= 0; k--)
        {
                sum = sum + chol[i, k] * chol[j, k];
        }

        if (i == j)
        {
                chol[i, j] = Math.Sqrt(A[i, j] - sum);
        }
        else
        {
                chol[i, j] = (A[i, j] - sum) / chol[j, j];
        }
    }
}

// Kdyz mame vypocitanou trojuhelnikovou matici double[,] chol, zbyva
    vypocitat vysledny vektor deformace double[] resultChol
// primy chod
for (int i = 0; i < A.GetLength(0); i++)
{
    double sum = 0;

    for (int k = i - 1; k >= 0; k--)
    {
        sum = sum + chol[i, k] * d[k];
    }
    d[i] = (b[i] - sum) / chol[i, i];
}

// zpetny chod
for (int i = A.GetLength(0) - 1; i >= 0; i--)
{
    double sum = 0;

    for (int k = i + 1; k < A.GetLength(0); k++)
    {
        sum = sum + chol[k, i] * resultChol[k];
    }
    resultChol[i] = (d[i] - sum) / chol[i, i];
}

// Vratime vysledny vektor
return resultChol;
}
```

Náročnost výpočtu Choleského rozkladu je úměrná n^3, což může trvat relativně dlouhou dobu při úlohách s velkým počtem stupňů volnosti. Velká výhoda ale spočívá v tom, že když máme např. mnoho zatěžovacích stavů, tak nám stačí udělat rozklad pouze jednou a pak pouze počítáme vektory deformace s různými zatěžovacími vektory podle vztahu (3.6), náročnost výpočtu je úměrná n^2

Pokud by matice tuhosti nebyla pozitivně definitní, viz [12], tak dojde v průběhu výpočtu k dělení nulou nebo výpočtu odmocniny ze záporného čísla. Říkáme tedy, že Choleského rozklad je bezpodmínečně zpětně stabilní. Matice tuhosti jsou zpravidla pozitivně definitní, takže tuto podmínku splňujeme.

Nyní, když máme deformace, tak si stačí spočítat jednotlivé vektory deformací v lokálních souřadnicích $\{u^*_{i,j}\}$, ty vynásobíme s příslušnými lokálními maticemi tuhosti $[K^*_{i,j}]$ a získáme vektory koncových sil $\{R^*_{i,j}\}$ jednotlivých prutů:

$$\{R^*_{i,j}\} = [K^*_{i,j}]\{u^*_{i,j}\} \tag{3.9}$$

3.3.2 Lineární stabilita

V kapitole 2.3.2 jsme si ukázali, že výpočet lineární stability spočívá v nalezení vlastních čísel matice a k tomu odpovídající vlastí vektor. Nyní již také známe vnitřní síly, můžeme si tedy spočítat geometrickou matici tuhosti viz. kapitola 2.3.1.

Pro výpočet rovnice (2.16) použijeme mocninnou metodu, viz [1] Jedná se o iterační metodu, kterou vypočítáme nejmenší vlastní číslo podle následujícího předpisu:

- Výpečet pomocné matice $[E] = -[K]^{-1}[K_\sigma]$
- Počáteční odhad vlastního vektoru $\{y\}^0$
- Nastavení počátku iterace $i = 0$
- Výpočet nového vlastního vektoru $\{y\}^{i+1} = [E]\{y\}^i$
- Normování vlastního vektoru $\{y\}^{i+1} = \{y\}^{i+1}/max(y^{i+1})$
- Výpočet vlastního čísla $\lambda = 1/max(y^{i+1})$
- Určení chyby $\lambda^{i+1} - \lambda^i < \epsilon$
- Ukončení výpočtu nebo pokračování v iteraci

Před samotnou iterací si tedy musíme připravit matici $[E]$

Inverzní matice

Inverzní matice, viz [10], se může počítat pouze z matice čtvercové, na obdélníkové matici není inverzní matice definována. Inverzní matice k matici $[A]$ (značíme $[A]^{-1}$) dále existuje jen tehdy, je-li matice regulární (nemá lineárně závislé řádky). Tato matice je pak určena jednoznačně. Dvě hlavní vlastnosti inverzní matice jsou:

$$([A]^{-1})^{-1} = [A] \tag{3.10}$$

A hlavně nejdůležitější vlastnost inverzní matice (což je zároveň také definice inverzní matice):

$$[A][A]^{-1} = [I], \tag{3.11}$$

kde $[I]$ je jednotková matice

Výpočet inverzní matice je opět velmi náročný na délku výpočtu. Zase existuje spousta metod, jak ji získat. My využijeme toho, že je matice symetrická a že již máme Choleského rozklad.

Výpočet inverzní matice v programovacím jazyce c# vypadá následovně:

```
// V tomto pripade je metoda pro vypocet soucasti nestatickeho objektu,
    nema tedy zadne parametry, ani nic nevraci, vsechny potrebne
    promenne jsou soucasti celeho objektu
public void InverseCholesky()
{
    // Globalni matice tuhosti - pro jednoduchost oznaceno pouze jako A
    double[,] A = matrixFactoryLin.GlobalMatrixCalc;
    // Pomocny vektor
    double[] d = new double[A.GetLength(0)];


    // Vytvoreni jednotkove matice
```

```
    double[,] identityMatrix = new double[A.GetLength(0),
        A.GetLength(0)];

    for (int i = 0; i < identityMatrix.GetLength(0); i++)
    {
        identityMatrix[i, i] = 1;
    }

    // chol[i, i] je trojuhelnikova matice z Choleskeho rozkladu
    for (int h = 0; h < A.GetLength(0); h++)
    {
        // primy chod
        for (int i = 0; i < A.GetLength(0); i++)
        {
            double sum = 0;

            for (int k = i - 1; k >= 0; k--)
            {
                sum = sum + chol[i, k] * d[k];
            }
            d[i] = (identityMatrix[h, i] - sum) / chol[i, i];
        }


        // zpetny chod
        for (int i = A.GetLength(0) - 1; i >= 0; i--)
        {
            double sum = 0;

            for (int k = i + 1; k < A.GetLength(0); k++)
            {
                sum = sum + chol[k, i] * identityMatrix[h, k];
            }
            identityMatrix[h, i] = (d[i] - sum) / chol[i, i];
        }
    }

    // Vynasobeni cele matice *-1
    for (int i = 0; i < identityMatrix.GetLength(0); i++)
    {
        for (int j = 0; j < identityMatrix.GetLength(0); j++)
        {
                identityMatrix[i, j] = identityMatrix[i, j] * -1;
        }
    }

    // Puvodni jednotkovou matici jsme postupne promenili na inverzni,
        takze ji priradime spravne promenne
    InverseGlobalMatrixCalc = identityMatrix;
}
```

Násobení matic

Vynásobení matic je poměrně jednoduchý algoritmus:

```
public void multiMatrix()
{
    // Inverzni matice
    double[,] A = InverseGlobalMatrixCalc;
    // Geometricka maice
    double[,] B = GlobalGeomMatrixCalc;

    // Nova matice vznikla vynasobenim
```

```
    MultiMatrixCalc = new double[A.GetLength(0), A.GetLength(0)];

    for (int i = 0; i < MultiMatrixCalc.GetLength(0); i++)
    {
        for (int j = 0; j < MultiMatrixCalc.GetLength(0); j++)
        {
            for (int k = 0; k < MultiMatrixCalc.GetLength(0); k++)
            {
                MultiMatrixCalc[i, j] = MultiMatrixCalc[i, j] + A[i, k]
                    * B[k, j];
            }
        }
    }
}
```

Vlastní čísla

Nyní máme vše přichystané pro výpočet nejmenšího vlastního čísla:

```
// Metoda, ktera ma jako parametr pripravenou matici E a vraci vlastni
    cislo a prislusny vlastni vektor
public static double[] PowerMethod(double[,] E, out double
    criricalEigenValue)
{
    // Lokalni promenne potrebne pro vypocet
    double max = 0;
    double eigenValue = 0;
    double maxLoop;

    // Pomocny vektor y
    double[] y = new double[E.GetLength(0)];
    // Pocatecni odhad valstniho vektoru
    for (int i = 0; i < y.Length; i++)
    {
        y[i] = 1;
    }

    // Zacatek iteracniho cyklu
    while (true)
    {
        // Vytvoreni noveho vlastniho vektoru
        double[] eigenVector = new double[E.GetLength(0)];

        // Nastaveni lokalni promenne na 0
        maxLoop = 0;

        // Vypocitani vlastniho vektoru - vynasobeni vektoru a matice
        for (int k = 0; k < eigenVector.Length; k++)
        {
            for (int l = 0; l < eigenVector.Length; l++)
            {
                eigenVector[k] = eigenVector[k] + (y[l] * E[k, l]);
            }
        }

        // Nalezeni maximalniho cisla ve vektoru
        for (int i = 0; i < eigenVector.Length; i++)
        {
            if (Math.Abs(eigenVector[i]) > maxLoop)
            {
                maxLoop = Math.Abs(eigenVector[i]);
                max = eigenVector[i];
```

```
            }
        }

        // Normovani vlastniho vektoru
        for (int m = 0; m < eigenVector.Length; m++)
        {
            eigenVector[m] = eigenVector[m] / max;
        }

        // Vypocet clastniho cisla
        max = 1 / max;

        // Zaokrouhleni vlastnich cisel
        double compareMax = Math.Round(max, 4);
        double compareEigenValue = Math.Round(eigenValue, 4);

        // Porovnani, jestli se vlastni cisla rovnaji
        if (compareMax.Equals(compareEigenValue))
        {
            // Pokud ano, tak vratime vlastni vektor a vlastni cislo
            criricalEigenValue = max;
            return y;
        }
        // Pokud ne, tak prenastavime lokalni promenne a pokracujemme v
            iteraci
        eigenValue = max;
        z = eigenVector;
    }
}
```

Nyní již máme hotové veškeré výpočty. Zbývá tedy sestavit vhodné algoritmy pro zobrazení výsledků.

Kapitola 4

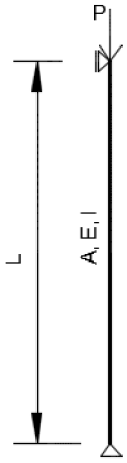
Konvergenční analýza

V následující kapitole se zaměříme na některé vybrané úlohy výpočtu lineární stability. Ukážeme si správnost výsledků v závislosti na počtu konečných prvků, porovnáme je se známými analytickými řešeními a s výsledky z jiných aplikací.

4.1 Přímý prut

Jako první se budeme věnovat přímému, prostě uloženému prutu o délce $L = 5$ m, modulu pružnosti materiálu $E = 210 \cdot 10^9$ Pa, momentu setrvačnosti průřezu $I = 1 \cdot 10^{-5}$ m^4 a průřezové ploše $A = 0.1$ m^2 viz obr. 4.1, pro který známe exaktní řešení viz. vztah (2.9):

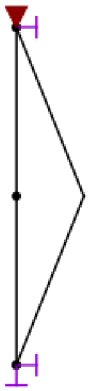
$$P_{cr} = \frac{\pi^2 EI}{L^2} = \frac{\pi^2 \cdot 210\text{e}9 \cdot 1\text{e}{-5}}{5^2} = 829,0468kN$$



Obrázek 4.1: Přímý prut

4.1.1 Výsledky

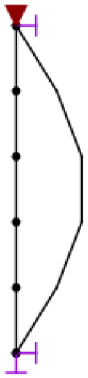
Na obrázcích 4.2 - 4.9 jsou zobrazeny tvary ztráty stability a kritické síla pro různé počty konečných prvků.



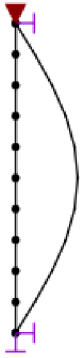
Obrázek 4.2: 2 prvky - $P_{cr} = 835.2831kN$

Obrázek 4.3: 3 prvky - $P_{cr} = 830.9578kN$

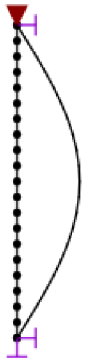
Obrázek 4.4: 4 prvky - $P_{cr} = 829.4714kN$



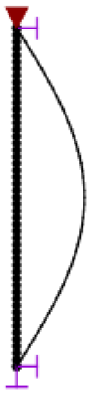
Obrázek 4.5: 5 prvků - $P_{cr} = 829.2226kN$



Obrázek 4.6: 10 prvků - $P_{cr} = 829.0579kN$



Obrázek 4.7: 20 prvků - $P_{cr} = 829.0475kN$



Obrázek 4.8: 50 prvků - $P_{cr} = 829.0468kN$

Obrázek 4.9: 100 prvků - $P_{cr} = 829.0468kN$

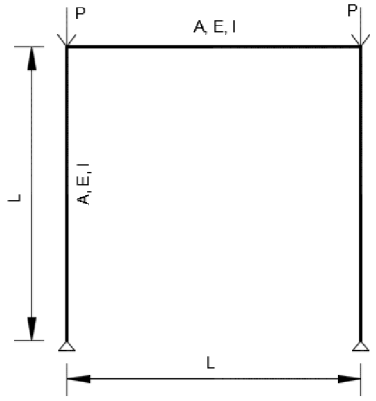
V tabulce 4.1 jsou uvedeny výsledné kritické síly a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu. Jak je vidět, tak již při dvou prvcích dostáváme relativně přesné řešení a při 50 prvcích je odchylka téměř nulová.

Počet prvků	Kritická síla [kN]	Odchylka [%]
2	835.2831	0.752
3	830.3578	0.158
4	829.4714	0.051
5	829.2226	0.021
10	829.0579	0.001
20	829.0475	$8.44 \cdot 10^{-05}$
50	829.0468	$< 1 \cdot 10^{-05}$
100	829.0468	$< 1 \cdot 10^{-05}$

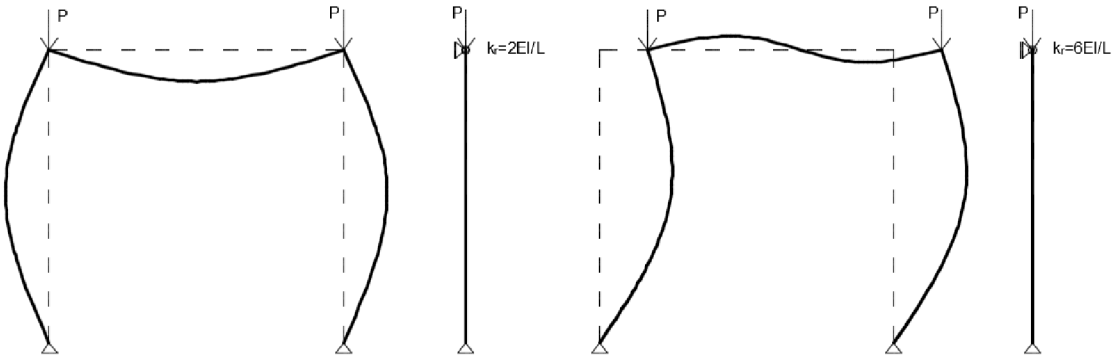
Tabulka 4.1: Vypočtené kritické síly přímého prutu a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu.

4.2 Rovinný rám

Nyní uvažme rovinný kloubově podepřený rám viz obr. 4.10. Tvar ztráty stability může být buď symetrický nebo nesymetrický, viz [3]. Při analytickém výpočtu můžeme na problém pohlížet tak, že se jedná o sloup, jehož jeden konec má určitou rotační tuhost v závislosti na ohybové tuhosti příčle rámu viz obr. 4.11.



Obrázek 4.10: Kloubově podepřený rám



Obrázek 4.11: Kloubově podepřený rám - Symetrický a nesymetrický tvar ztráty stability

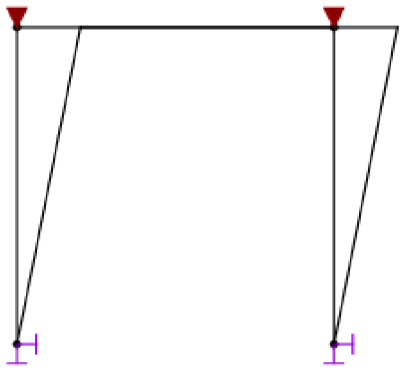
4.2.1 Nesymetrický tvar ztráty stability

Rám o délce a šířce $L = 5$ m, modulu pružnosti materiálu $E = 210 \cdot 10^9$ Pa, momentu setrvačnosti průřezu $I = 1{\cdot}10^{-5}$ m^4 a průřezové ploše $A = 0.1$ m^2 viz obr. 4.10. Pro tento nesymetrický tvar platí vztah, viz [3]:

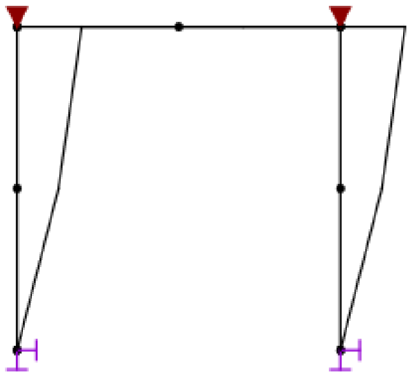
$$P_{cr} = \frac{1.34955^2 EI}{L^2} = \frac{1.34955^2 \cdot 210e9 \cdot 1e{-}5}{5^2} = 152.98859kN$$

4.2.2 Výsledky

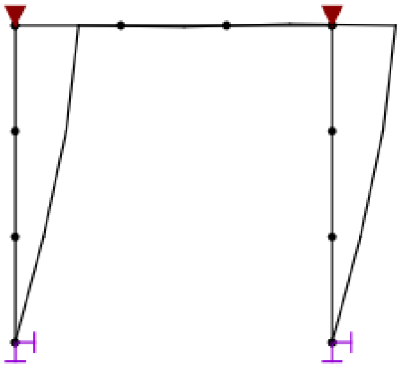
Na obrázcích 4.12 - 4.18 jsou zobrazeny tvary ztráty stability a kritická síla pro různý počet konečných prvků.



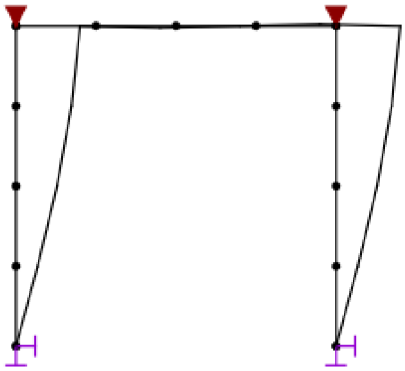
Obrázek 4.12: 3 prvky - $P_{cr} = 153.42066kN$



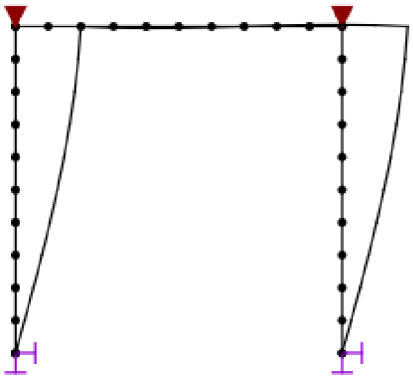
Obrázek 4.13: 6 prvků - $P_{cr} = 153.01538kN$



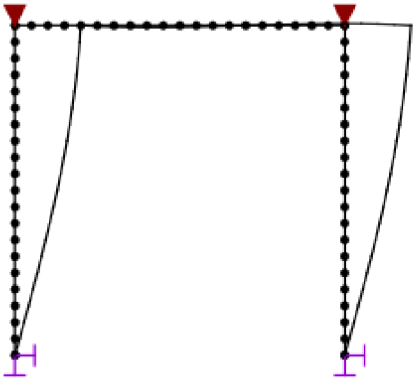
Obrázek 4.14: 9 prvků - $P_{cr} = 152.99079kN$



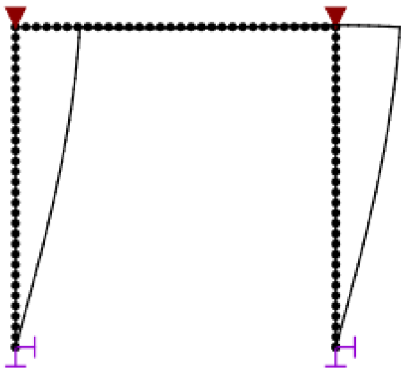
Obrázek 4.15: 12 prvků - $P_{cr} = 152.98655kN$



Obrázek 4.16: 30 prvků - $P_{cr} = 152.98463kN$



Obrázek 4.17: 60 prvků - $P_{cr} = 152.98457kN$



Obrázek 4.18: 90 prvků - $P_{cr} = 152.98457kN$

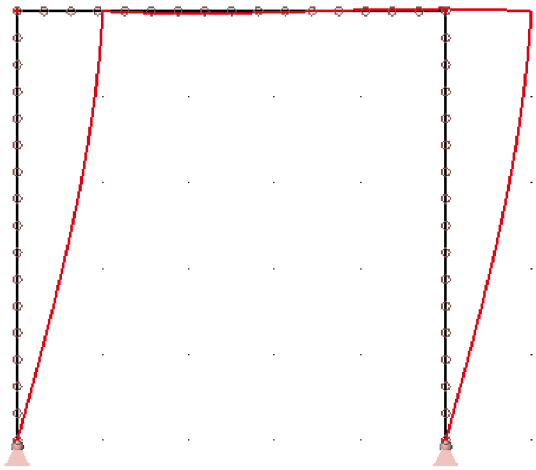
V tabulce 4.2 jsou uvedeny výsledné kritické síly a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu. Výpočet opět konverguje velmi rychle, při nejmenším možném počtu konečných prvků, tedy při třech, máme odchylku pouze $0,282\%$. Řešení ovšem konvergovalo k výsledku, který je o $0,003\%$ rozdílný od analytického. Tato odchylka je způsobená pravděpodobně nedokonalostí numerických metod nebo nepřesností analytického výpočtu.

Počet prvků	Kritická síla [kN]	Odchylka [%]
3	153.42066	−0.282
6	153.01538	−0.017
9	152.99079	−0.001
12	152.98655	0.001
30	152.98463	0.003
60	152.98457	0.003
90	152.98457	0.003

Tabulka 4.2: Vypočtené kritické síly rovinného rámu a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu.

Scia Engineer

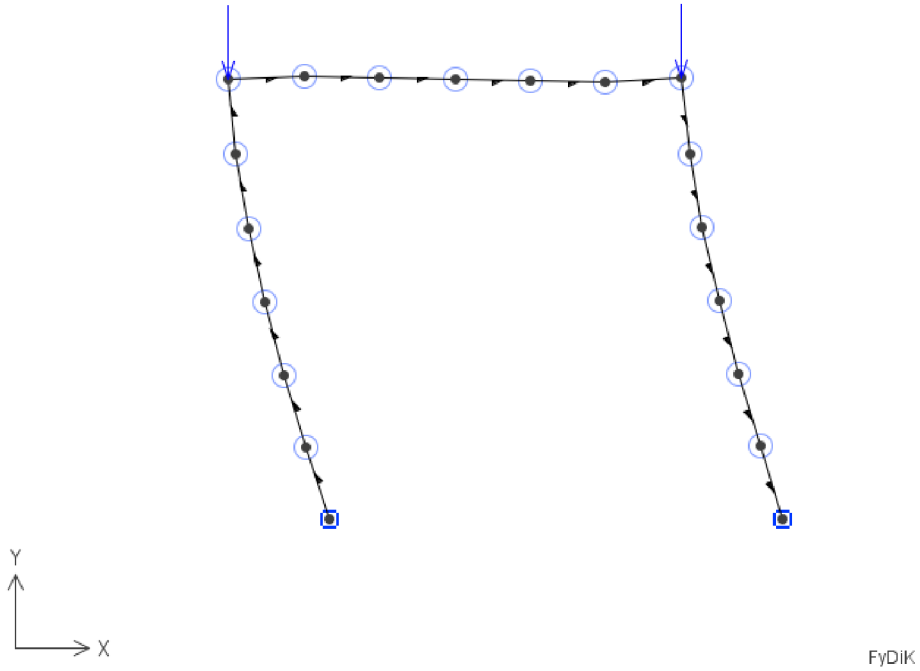
Pro porovnání si ještě uvedeme výsledek za softwaru Scia Engineer, viz [8]. Při 30-ti prvcích dostáváme výsledek $P_{cr} = 152.98kN$, viz obr. 4.19, což se shoduje s našimi výsledky. Ve Scia Engineer bylo P_{cr} spočítáno Laczosovou metodou.



Obrázek 4.19: Scia Engineer - $P_{cr} = 152.98kN$

FyDiK

Další aplikace ve které jsme si ověřili správnost výpočtu je FyDiK, viz [14], kde jsme geometricky nelineárním výpočtem dospěli k řešení $P_{cr} = 150.1kN$, při 18-ti prvcích viz obr. 4.20.



Obrázek 4.20: FyDiK - $P_{cr} = 150.1kN$

4.2.3 Symetrický tvar ztráty stability

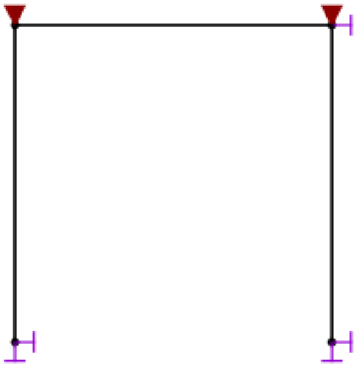
Protože dokážeme spočítat pouze jeden stav ztráty stability, tak pro vyvolání tohoto případu přidáme do modelu ještě jednu posuvnou podporu.

Rám o délce a šířce $L = 5$ m, modulu pružnosti materiálu $E = 210 \cdot 10^9$ Pa, momentu setrvačnosti průřezu $I = 1{\cdot}10^{-5}$ m^4 a průřezové ploše $A = 0.1$ m^2 viz obr. 4.10. Pro tento symetrický tvar platí vztah, viz [3]:

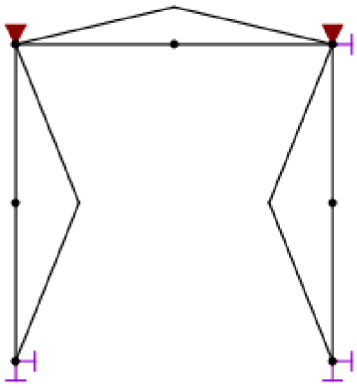
$$P_{cr} = \frac{3.59088^2 EI}{L^2} = \frac{3.59088^2 \cdot 210e9 \cdot 1e{-}5}{5^2} = 152.98859kN$$

4.2.4 Výsledky

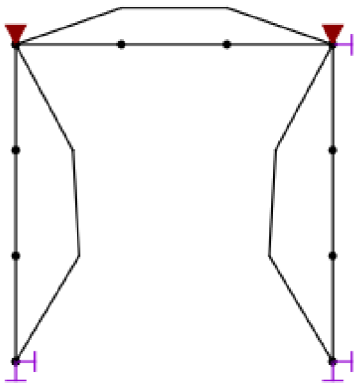
Na obrázcích 4.21 - 4.27 jsou zobrazeny tvary ztráty stability a kritická síla pro různý počet konečných prvků.



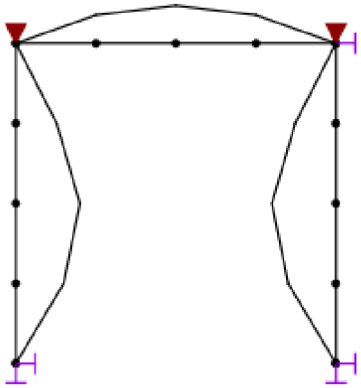
Obrázek 4.21: 3 prvky - $P_{cr} = 1417.13603kN$



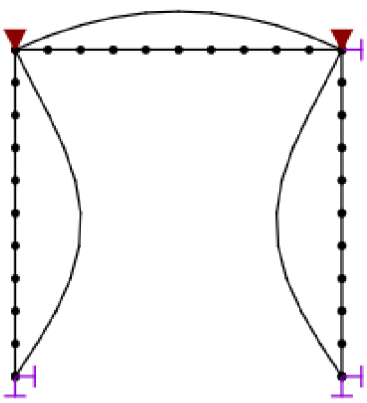
Obrázek 4.22: 6 prvků - $P_{cr} = 1093.15239kN$



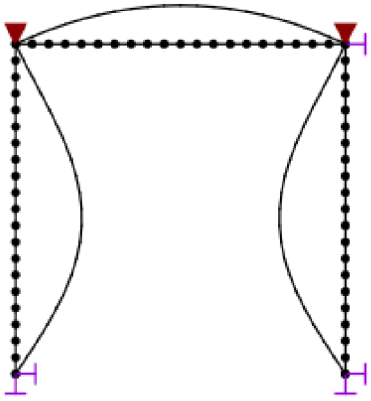
Obrázek 4.23: 9 prvků - $P_{cr} = 1085.42789kN$



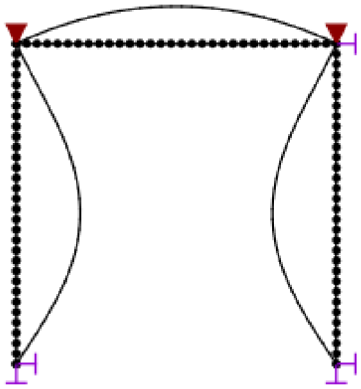
Obrázek 4.24: 12 prvků - $P_{cr} = 1083.89197kN$



Obrázek 4.25: 30 prvků - $P_{cr} = 1083.15136kN$



Obrázek 4.26: 60 prvků - $P_{cr} = 1083.13218kN$



Obrázek 4.27: 90 prvků - $P_{cr} = 1083.13107kN$

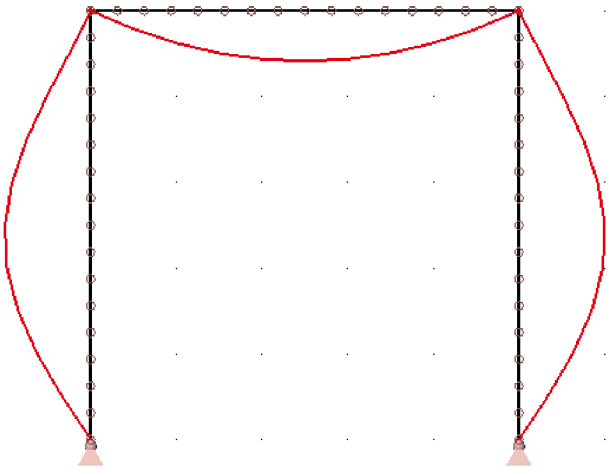
V tabulce 4.3 jsou uvedeny výsledné kritické síly a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu. Tento případ konverguje pomaleji, než úloha s nesymetrickým tvarem vybočení. Při třech prvcích dostáváme odchylku 30.836%. Se zvětšujícím se dělením ale opět dostáváme adekvátní výsledky.

Počet prvků	Kritická síla [kN]	Odchylka [%]
3	1417.13603	-30.836
6	1093.15239	-0.925
9	1085.42789	-0.212
12	1083.89197	-0.070
30	1083.15136	-0.002
60	1083.13218	$-2 \cdot 10^{-05}$
90	1083.13107	$7 \cdot 10^{-05}$

Tabulka 4.3: Vypočtené kritické síly rovinného rámu a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu.

Scia Engineer

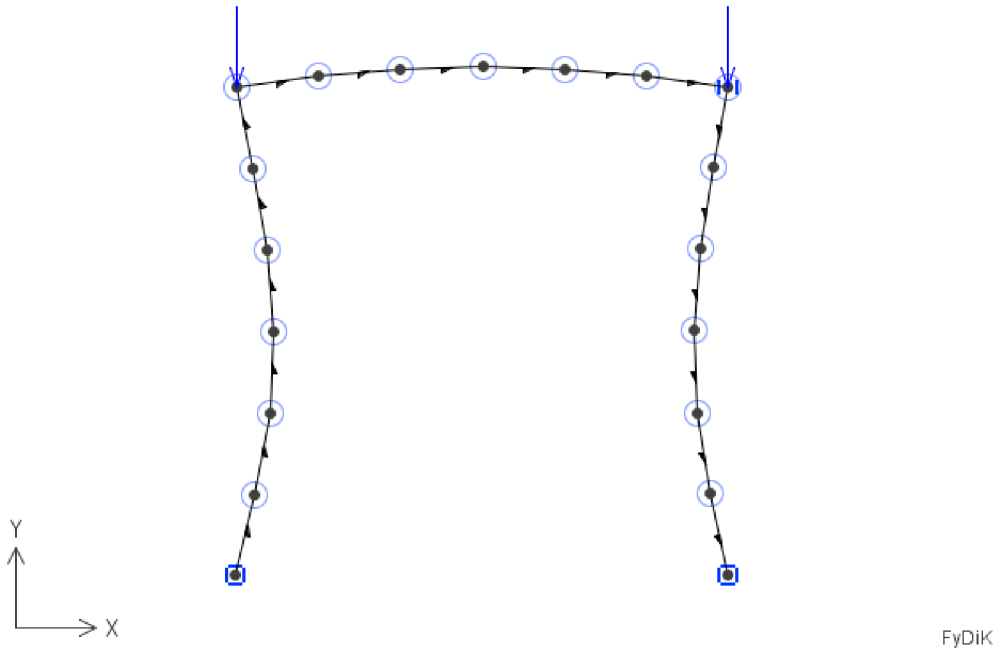
Pro porovnání si opět uvedeme výsledek za softwaru Scia Engineer, viz [8]. Při 30-ti prvcích dostáváme výsledek $P_{cr} = 1083.00kN$, viz obr. 4.28, což se shoduje s našimi výsledky. Nyní jsme již nemuseli přidávat vodorovnou podporu, protože tento symetrický tvar nastane při vypočítání druhého nejmenšího vlastního čísla (ve Scia Engineer si můžeme nastavit kolik jich chceme spočítat).



Obrázek 4.28: Scia Engineer - $P_{cr} = 1083.00kN$

FyDiK

Také jsme opět použili FyDiK, viz [14] a výsledná kritická vyšla $P_{cr} = 1044.3kN$, což je opět velmi blízko našemu řešení viz obr. 4.29.



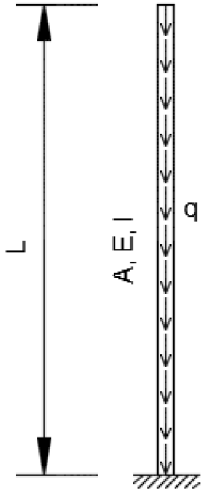
Obrázek 4.29: FyDiK - $P_{cr} = 1044.3kN$

4.3 Konzola zatížená spojitým zatížením

Další zajímavou stabilitní úlohou je konzola zatížená spojitým zatížením. Vztah pro kritické spojité zatížení q_{cr} definoval Stepan Prokopovič Timošenko na začátku 20. století, viz [6]:

$$q_{cr} = 7.837\frac{EI}{L^3} \tag{4.1}$$

Mějme konzolu o délce $L = 5$ m, modulu pružnosti materiálu $E = 210 \cdot 10^9$ Pa, momentu setrvačnosti průřezu $I = 1{\cdot}10^{-5}$ m^4 a průřezové ploše $A = 0.1$ m^2 viz obr. 4.30.



Obrázek 4.30: Konzola zatížená vlastní tíhou

Pro kritickou sílu podle vztahu (4.1) platí:

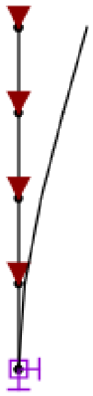
$$q_{cr} = 7.837\frac{EI}{L^3} = 7.837\frac{210e9 \cdot 1e{-}5}{5^3} = 131.662kN/m$$

4.3.1 Výsledky

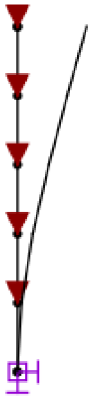
Na obrázcích 4.31 - 4.37 jsou zobrazeny tvary ztráty stability a kritická síla pro různý počet konečných prvků.

Obrázek 4.31: 2 prvky - $q_{cr} = 118.256kN/m$

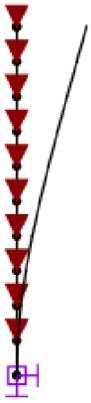
Obrázek 4.32: 3 prvky - $q_{cr} = 125.669kN/m$



Obrázek 4.33: 4 prvky - $q_{cr} = 128.286 kN/m$

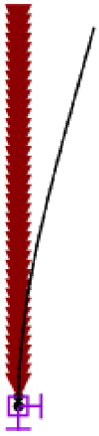


Obrázek 4.34: 5 prvků - $q_{cr} = 129.501 kN/m$



Obrázek 4.35: 10 prvků - $q_{cr} = 130.829 kN/m$

Obrázek 4.36: 20 prvků - $q_{cr} = 131.532 kN/m$



Obrázek 4.37: 40 prvků - $q_{cr} = 131.634 kN/m$

V tabulce 4.4 jsou uvedeny výsledné kritické síly a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu. Opět vidíme, že s jemnějším dělením dostáváme přesnější řešení. Poměrně velké odchylky při menším počtu prvků jsou způsobeny rozdělením síly do uzlů, polovina zatížení jednoho prvku vždy působí přímo v podpoře a nevsupuje tedy do výpočtu.

Počet prvků	Kritická síla [kN/m]	Odchylka [%]
2	118.256	10.182
3	125.669	4.552
4	128.286	2.564
5	129.501	1.641
10	130.829	0.632
20	131.532	0.099
40	131.634	0.021

Tabulka 4.4: Vypočtené kritické síly konzoly zatížené spojitým zatížením a jejich odchylky od analytického řešení v závislosti na jemnosti dělení modelu.

Kapitola 5

Závěr

Předkládaná diplomová práce se věnovala hledání kritického zatížení pro ztrátu stability rovinných prutových konstrukcí. V úvodu byl ukázán výpočet pro přímý prut a následně byly, pomocí energetických principů, odvozeny vztahy pro řešení tohoto problému v rámci metody konečných prvků. Za účelem ověření těchto vztahů byla vyvinuta aplikace v programovacím jazyce c#, která spočítá kritické zatížení na libovolné rovinné prutové konstrukci. Byly popsány všechny důležité kroky a numerické metody, včetně zdrojového kódu, nezbytné pro výpočet. Nakonec byla provedena analýza na několika vybraných úlohách. Při této analýze se porovnávala přesnost výpočtu, při různém počtu konečných prvků, s analytickým řešením a s dalšími dostupnými aplikacemi. Získané výsledky dokládají správnou funkci aplikace.

Literatura

[1] MATHEWS, John a Kurtis Fink. *Numerical methods using Matlab*. Prentice Hall, 1999.

[2] COOK, Robert. *Concepts and applications of finite element analysis*. Johm Wiley & Sons, INC., 2002. ISBN: 9780471356059.

[3] GAMBHIR, Murari. *Stability analysis and design of structures*. Springer-Verlag Berlin Heidelberg, 2004. ISBN: 9783642058660.

[4] NĚMEC, Ivan. *Nelineární mechanika*. Brno, 2006.

[5] ALBAHARI, Joseph a Ben Albahari. *C# 5.0 in nutshell*. O'Reilly Media, Inc., 2012. ISBN: 78-1-449-32010-2.

[6] MIT OpenCourseWare. *Advanced Topic in Column Buckling*. Dostupné z: http://ocw.mit.edu. 2013.

[7] Eigenvalues a eigenvectors [online]. *Wikipedie*. Dostupné z: https://en.wikipedia.org/wiki/Eigenvalues_and_eigenvectors.

[8] SCIA Engineer. Dostupné z: http://www.scia.net.

[9] Cholesky decomposition [online]. *Wikipedie*. Dostupné z: http://en.wikipedia.org/wiki/Cholesky_decomposition.

[10] Inverzní matice [online]. *Wikipedie*. Dostupné z: http://www.matematika.cz/inverzni-matice.

[11] Leonhard Euler [online]. *Wikipedie*. Dostupné z: http://en.wikipedia.org/wiki/Leonhard_Euler.

[12] Positive-definite matrix [online]. *Wikipedie*. Dostupné z: http://en.wikipedia.org/wiki/Positive-definite_matrix.

[13] Soustava lineárních rovnic [online]. *Wikipedie*. Dostupné z: http://cs.wikipedia.org/wiki/Soustava_lineárních_rovnic.

[14] FRANTÍK, Petr. *FyDiK*. Dostupné z: http://fydik.kitnarf.cz/.

Příloha

Na přiloženém kompaktním disku se nachází:

- Vyvinutá aplikace
- Návod na použití aplikace