



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ  
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA STROJNÍHO INŽENÝRSTVÍ  
ÚSTAV AUTOMATIZACE A INFORMATIKY

FACULTY OF MECHANICAL ENGINEERING  
INSTITUTE OF AUTOMATION AND COMPUTER SCIENCE

## ŘEŠENÍ ÚLOH DYNAMIKY TĚLES POMOCÍ MATEMATICKÝCH SOFTWARE

APPLICATION OF NUMERICAL PROCEDURES FOR SOLUTION OF DYNAMICS PROBLEMS

DIPLOMOVÁ PRÁCE  
MASTER'S THESIS

AUTOR PRÁCE  
AUTHOR

Bc. PAVEL PUČEGL

VEDOUCÍ PRÁCE  
SUPERVISOR

doc. RNDr. KAREL PELLANT, CSc.

BRNO 2010



Vysoké učení technické v Brně, Fakulta strojního inženýrství

Ústav automatizace a informatiky

Akademický rok: 2009/2010

## **ZADÁNÍ DIPLOMOVÉ PRÁCE**

student(ka): Bc. Pavel Pučegl

který/která studuje v **magisterském navazujícím studijním programu**

obor: **Aplikovaná informatika a řízení (3902T001)**

Ředitel ústavu Vám v souladu se zákonem č.111/1998 o vysokých školách a se Studijním a zkušebním řádem VUT v Brně určuje následující téma diplomové práce:

### **Řešení úloh dynamiky těles pomocí matematických softwarů**

v anglickém jazyce:

### **Application of numerical procedures for solution of dynamics problems**

Stručná charakteristika problematiky úkolu:

Při výuce kinematiky a dynamiky u řady příkladů není možné nalézt analytické vyjádření výsledku. Při hledání numerického řešení úloh mechaniky popř. pro účely konstrukce robotů jsou proto v současné době v rámci stávajících softwarů vyvíjeny a aplikovány speciální matematické procedury (např. SimMechanics v rámci Matlabu apod.)

Cíle diplomové práce:

Cílem práce je aplikace stávajících matematických softwarů (Maple, Matlab apod.) pro výukové a demonstrační řešení příkladů z kinematiky a dynamiky. Bude provedeno testování možnosti použití těchto systémů pro zjišťování frekvenčních charakteristik nelineárních mechanických systémů

Seznam odborné literatury:

T. Zemánek, Aplikace matematických softwarů při numerickém řešení úloh dynamiky

Vedoucí diplomové práce: doc. RNDr. Karel Pellant, CSc.

Termín odevzdání diplomové práce je stanoven časovým plánem akademického roku 2009/10.

V Brně, dne 20.11.2009



---

prof. RNDr. Ing. Miloš Šeda, Ph.D.  
Ředitel ústavu

---

doc. RNDr. Miroslav Doupovec, CSc.  
Děkan fakulty

## **PODĚKOVÁNÍ**

Na tomto místě bych rád poděkoval všem, kteří mi poskytli svůj čas a svými cennými radami přispěli k vypracování této diplomové práce. Především pak vedoucímu mé diplomové práce panu doc. RNDr. Karlu Pellantovi, CSc.

Dále bych rád poděkoval svému zaměstnavateli, který mě po celou dobu studia podporoval.



### **BIBLIOGRAFICKÁ CITACE DLE ČSN ISO 690**

PUČEGL, P. *Řešení úloh dynamiky těles pomocí matematických softwarů*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2010. 61 s. Vedoucí diplomové práce doc. RNDr. Karel Pellant, CSc.

### **ČESTNÉ PROHLÁŠENÍ**

Prohlašuji, že předložená diplomová práce je mou vlastní autorskou prací, kterou jsem vypracoval pod vedením vedoucího mé diplomové práce a s použitím literatury uvedené v seznamu literatury.

V Brně dne 28. 5. 2010

Pavel Pučegl

.....





**ABSTRAKT**

Rozbor možnosti použití stávajících matematických softwarů pro zjišťování frekvenčních charakteristik nelineárních mechanických systémů.

**ABSTRACT**

Discussion of the application of mathematical softwares for the determination of frequency characteristics of mechanical systems.

**KLÍČOVÁ SLOVA**

Nelineární systémy, mechanika těles, software, frekvenční charakteristika, numerické procedury.

**KEYWORDS**

Nonlinear systems, mechanics of solids, software, frequency characteristics, numerical procedures.



## OBSAH

1. ÚVOD.	13
2. KMITÁNÍ.	15
2.1 Volné lineární kmitání mechanického oscilátoru.	15
2.2 Nucené lineární kmitání mechanického oscilátoru.	17
2.2.1 Amplitudo-frekvenční charakteristika lineárního systému.	18
2.3 Nelineární kmitání.	19
2.3.1 Nelinearita vratné síly pružiny.	19
2.3.2 Nelinearita tlumení.	20
3. KONSTRUKCE AMPLITUDO-FREKVENČNÍ CHARAKTERISTIKY.	23
3.1 Metody Runge-Kutta.	23
3.2 Nalezení ustálené výchylky kmitů.	23
4. IMPLEMENTACE METODY DO MATEMATICKÝCH SOFTWARE.	27
4.1 Implementace v programu MATLAB.	28
4.1.1 MATLAB.	28
4.1.2 Program pro MATLAB.	28
4.2 Implementace v programu MAPLE.	33
4.2.1 MAPLE.	33
4.2.2 Program pro MAPLE.	34
5. ŘEŠENÉ PŘÍKLADY.	41
5.1 Příklad č.1 - lineární systém.	41
5.2 Příklad č.2 - systém s měknoucí charakteristikou vratné síly pružiny.	44
5.3 Příklad č.2 - systém s tuhnoucí charakteristikou vratné síly pružiny.	47
5.4 Příklad č.2 - systém s nelineárním tlumením.	51
6. ZÁVĚR.	55
7. SEZNAM OBRÁZKŮ.	57
8. SEZNAM POUŽITÉ LITERATURY.	59
9. SEZNAM PŘÍLOH.	61



## 1. ÚVOD

Úlohy dynamiky patří z pohledu mechaniky těles k nejzajímavějším úlohám. Je to proto, že se tyto úlohy velmi často vyskytují v praxi. Řešení úloh dynamiky vede z matematického pohledu na řešení diferenciálních rovnic, mnohdy navíc nelineárních. Řešení takovýchto rovnic je velmi obtížné.

Při současném stavu výpočetní techniky se jeví jako nejjednodušší numerické řešení takovýchto úloh. Existuje řada matematických softwarů, které nabízí celou škálu možností řešení diferenciálních rovnic včetně rovnic nelineárních.

Cílem této práce je aplikace stávajících matematických softwarů MATLAB a MAPLE, které patří bezesporu k nejpoužívanějším, na řešení úloh dynamiky tak, aby bylo možné pomocí těchto softwarů demonstrovat chování nelineárních kmitavých dynamických systémů. Řešenou problematiku lze rozdělit na tyto dílčí části:

- Analýzu problematiky kmitání
- Výběr vhodné metody pro řešení dané problematiky
- Implementaci vybrané metody do prostředí MATLAB a MAPLE
- Testování metody a její implementace na příkladech



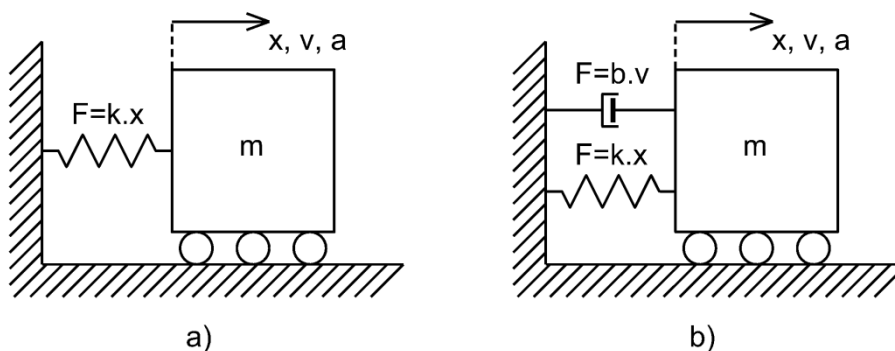
## 2. KMITÁNÍ

Mechanické kmitání je periodicky se opakující pohyb vyvolaný interakcí setrvačné síly tělesa a odporové deformační síly pružiny spřažené s tímto tělesem. Z pohledu dynamiky patří kmitání k nejzajímavějším druhům pohybu. Kmitavý pohyb bývá většinou symetrický podle rovnovážné polohy. Výjimku tvoří nelinearity závislé na směru pohybu. Podle různých hledisek rozlišujeme několik druhů mechanického kmitání:

- podle tlumení
  - o netlumené - kmitající těleso není v průběhu kmitání brzděno
  - o tlumené - kmitající těleso je v průběhu kmitání brzděno speciálním brzdícím prvkem (tlumič) nebo odporem prostředí, v němž kmitá
- podle charakteru vnější síly působící na dynamickou soustavu
  - o volné - na kmitající těleso nepůsobí v průběhu kmitání žádná vnější harmonická síla
  - o nucené - na kmitající těleso působí v průběhu kmitání vnější harmonická síla s tělesem vhodně spřažená
- podle charakteru prvků dynamické soustavy
  - o lineární - chování všech prvků dynamické soustavy lze popsat lineárními rovnicemi
  - o nelineární - v dynamické soustavě se vyskytuje alespoň jeden prvek, jehož chování nelze popsat lineární rovnicí

### 2.1 Volné lineární kmitání mechanického oscilátoru

Schematické zobrazení netlumeného mechanického oscilátoru s 1° volnosti je znázorněno na obrázku 1a, tlumeného na obrázku 1b.



Obr. 1.: Mechanický oscilátor

Při sestavování pohybové rovnice vyjdeme z vyjádření setrvačné síly tělesa dle II. Newtonova pohybového zákona

$$a) \quad F_s + F_k = 0, \quad b) \quad F_s + F_b + F_k = 0 \quad (1)$$

kde  $F_s$  je setrvačná síla,  $F_k$  je vratná síla pružiny a  $F_b$  je brzdná síla tlumení. Působení valivého odporu neuvažujeme, příp. je zohledněn v brzdné síle tlumení. Po dosazení potom dostaneme pohybové rovnice ve tvaru

$$a) \quad m \cdot a(t) + k \cdot x(t) = 0, \quad b) \quad m \cdot a(t) + b \cdot v(t) + k \cdot x(t) = 0 \quad (2)$$

kde  $m$  je hmotnost tělesa,  $k$  je tuhost pružiny,  $b$  je koeficient tlumení,  $x(t)$  je okamžitá hodnota výchylky,  $v(t)$  je okamžitá hodnota rychlosti a  $a(t)$  je okamžitá hodnota zrychlení v čase  $t$ . Po úpravě a dosazení obecných pohybových rovnic

$$v(t) = \frac{dx}{dt} = \dot{x}(t), \quad a(t) = \frac{dv}{dt} = \frac{d^2x}{dt^2} = \ddot{x}(t) \quad (3)$$

dostaneme pohybovou rovnici mechanického oscilátoru s 1° volnosti v základním tvaru

$$a) \quad m \cdot \ddot{x}(t) + k \cdot x(t) = 0, \quad b) \quad m \cdot \ddot{x}(t) + b \cdot \dot{x}(t) + k \cdot x(t) = 0 \quad (4)$$

Řešení lineární diferenciální rovnice 4a (oscilátor bez tlumení) pro výchylku  $x$  z rovnovážné polohy v libovolném čase  $t$  má tvar

$$x(t) = A \cdot \sin(\Omega \cdot t) + B \cdot \cos(\Omega \cdot t) \quad (5)$$

kde  $A$  a  $B$  jsou integrační konstanty ve tvaru

$$A = \frac{\dot{x}(0)}{\Omega}, \quad B = x(0), \quad C = \sqrt{A^2 + B^2} = \sqrt{\left(\frac{\dot{x}(0)}{\Omega}\right)^2 + x(0)^2} \quad (6)$$

$\Omega$  je vlastní frekvence mechanického oscilátoru ve tvaru

$$\Omega^2 = \frac{k}{m}, \quad \Omega = \sqrt{\frac{k}{m}} \quad (7)$$

vyjadřující úhlovou frekvenci volných netlumených kmitů a  $C$  pak jejich amplitudu. [2]

V případě oscilátoru s tlumením dostaneme řešení lineární diferenciální rovnice 4b pro výchylku  $x$  z rovnovážné polohy v libovolném čase  $t$  ve tvaru

$$x(t) = C_1 \cdot e^{\lambda_1 t} + C_2 \cdot e^{\lambda_2 t} \quad (8)$$

kde  $C_1$  a  $C_2$  jsou integrační konstanty, které se určí z počátečních podmínek  $x(0)$ ,  $\dot{x}(0)$  a  $\lambda_1$ ,  $\lambda_2$  jsou tzv. vlastní čísla soustavy, což jsou kořeny charakteristické rovnice

$$m \cdot I^2 + b \cdot I + k = 0 \quad (9)$$

jejichž řešení je dáno vztahem

$$I_{1,2} = \frac{-b \pm \sqrt{b^2 - 4km}}{2m} \quad (10)$$

Z rovnice 10 je zřejmé, že vlastní čísla soustavy  $\lambda_1$  a  $\lambda_2$  mohou být buďto reálná (shodná nebo různá), výsledný pohyb po dosazení vlastních čísel do rovnice 8 bude exponenciální, nebo komplexně sdružená, výsledný pohyb bude kmitavý. O charakteru vlastních čísel rozhoduje znaménko výrazu pod odmocninou, tedy výraz  $b^2 - 4km$ . Podle charakteru vlastních čísel dělíme tlumení na podkritické  $b^2 - 4km < 0$ , kritické  $b^2 - 4km = 0$  a nadkritické  $b^2 - 4km > 0$ . Jak již bylo zmíněno, v případě kritického a nadkritického tlumení se nejedná o kmitavý pohyb, z pohledu vyšetřování kmitavého pohybu je tedy zajímavý pouze případ podkritického tlumení. Pro případ podkritického tlumení lze vztah (rovnice 10) pro výpočet vlastních čísel soustavy upravit do následující podoby

$$I_{1,2} = -\frac{b}{2m} \pm i \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2} \quad (11)$$

Vzhledem k tomu, že reálná část vztahu je záporná, má charakter tlumení a bývá proto označována jako koeficient doznívání  $\delta$ .

$$d = \frac{b}{2m} \quad (12)$$



Imaginární část vztahu (rovnice 11) má charakter kmitání, lze z ní vyjádřit vztah pro vlastní frekvenci tlumeného mechanického oscilátoru ve tvaru

$$\Omega_{tl} = \sqrt{\frac{k}{m} - \left(\frac{b}{2m}\right)^2} = \sqrt{\Omega^2 - d^2} \quad (13)$$

kde  $\Omega$  je vlastní frekvence netlumeného mechanického oscilátoru. Vlastní frekvence mechanického oscilátoru je jeho nejdůležitějším parametrem a v případě nuceného kmitání buzeného harmonickou silou má významný vliv na jeho amplitudo-frekvenční charakteristiku. [3]

Pohybovou rovnici 4b tlumeného mechanického oscilátoru s 1° volnosti lze po dosazení vztahů pro koeficient dozínání  $\delta$  (rovnice 12) a vlastní frekvenci tlumeného mechanického oscilátoru  $\Omega_{tl}$  (rovnice 13) přepsat do tvaru

$$\ddot{x}(t) + 2d \cdot \dot{x}(t) + (\Omega_{tl}^2 + d^2) \cdot x(t) = 0 \quad (14)$$

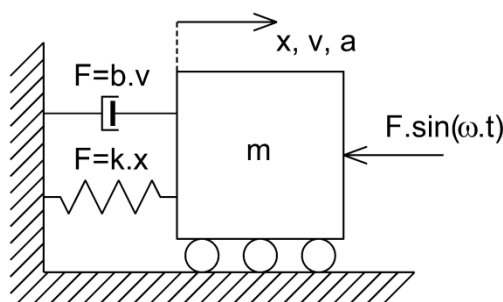
Řešení této rovnice pak bude vypadat

$$x(t) = C \cdot e^{-d \cdot t} \cdot \sin(\Omega_{tl} \cdot t + j) \quad (15)$$

kde  $C$  a  $\varphi$  jsou integrační konstanty vyplývající z počátečních podmínek  $x(0)$  a  $\dot{x}(0)$ . [2]

## 2.2 Nucené lineární kmitání mechanického oscilátoru

V případě nuceného kmitání je mechanický oscilátor na rozdíl od volného kmitání buzen vnější harmonickou silou působící na kmitající těleso. Schematické znázornění tohoto případu je na obrázku 2.



Obr. 2.: Harmonicky buzený mechanický oscilátor

Vzhledem k tomu, že netlumený mechanický oscilátor je pouze teoretický případ, v praxi je kmitající těleso tlumeno vždy alespoň odporem prostředí, budeme nadále uvažovat pouze tlumený oscilátor.

V případě buzení harmonickou silou v pohybové rovnici přibude vztah pro tuto vnější sílu. Pohybová rovnice nuceného kmitání bude tedy vypadat

$$m \cdot \ddot{x}(t) + b \cdot \dot{x}(t) + k \cdot x(t) = F \cdot \sin(\omega \cdot t) \quad (16)$$

Pohybovou rovnici 16 harmonicky buzeného mechanického oscilátoru lze upravit do podobného tvaru jako pohybovou rovnici 14 oscilátoru bez buzení

$$\ddot{x}(t) + 2d \cdot \dot{x}(t) + (\Omega_{tl}^2 + d^2) \cdot x(t) = \frac{F}{m} \cdot \sin(\omega \cdot t) \quad (17)$$

Řešení pohybové rovnice 17 je dáno součtem řešení homogenní části, stejné jako u oscilátoru bez buzení, vyjadřující tlumené odeznívající kmitání oscilátoru bez buzení - přechodový děj při změně buzení, a partikulárního integrálu budící složky pohybové rovnice 17, který bude mít tvar ustálených kmitů s frekvencí budící síly  $\omega$ . Řešení tedy bude mít tvar

$$x(t) = C \cdot e^{-d \cdot t} \cdot \sin(\Omega_{tl} \cdot t + j) + A_{ust} \cdot \sin(\omega \cdot t + g) \quad (18)$$

kde  $A_{ust}$  je amplituda ustálených kmitů a  $\gamma$  je fázový posun ustálených kmitů vůči kmitům budící síly  $F$ . Hodnota složky netlumených kmitů v řešení pohybové rovnice je exponenciálně klesající, po odeznění přechodového děje tedy klesne k nule a řešení pohybové rovnice bude mít tvar

$$x(t) = A_{ust} \cdot \sin(w \cdot t + g) \quad (19)$$

Dosazením řešení ustáleného kmitání (rovnice 19) do pohybové rovnice 17 získáme vztah

$$\begin{aligned} & -m \cdot A_{ust} \cdot w^2 \cdot \sin(w \cdot t + g) + 2d \cdot m \cdot A_{ust} \cdot w \cdot \cos(w \cdot t + g) + \\ & + m \cdot (\Omega_{tl}^2 + d^2) \cdot A_{ust} \cdot \sin(w \cdot t + g) = F \cdot \sin(w \cdot t) \end{aligned} \quad (20)$$

Pohybová rovnice 20 musí být splněna pro libovolný čas  $t$ . V řešení rovnice 20 lze tedy pokračovat vhodnou volbou času  $t$  a jeho dosazením do rovnice 20. Pro zjednodušení výpočtů zvolíme  $t$  takové, že  $w \cdot t + g = 0$  a  $w \cdot t + g = \pi/2$ . Dosazením takto zvolených hodnot času  $t$  do pohybové rovnice 20 dostaneme následující soustavu rovnic

$$\begin{aligned} 2d \cdot m \cdot A_{ust} \cdot w &= -F \cdot \sin(g) \\ -m \cdot A_{ust} \cdot w^2 + m \cdot A_{ust} \cdot (\Omega_{tl}^2 + d^2) &= F \cdot \cos(g) \end{aligned} \quad (21)$$

Vyjádřením  $\gamma$  z první rovnice a následným dosazením do druhé rovnice získáme vztah pro výpočet ustálené výchylky nuceného kmitání  $A_{ust}$  ve tvaru

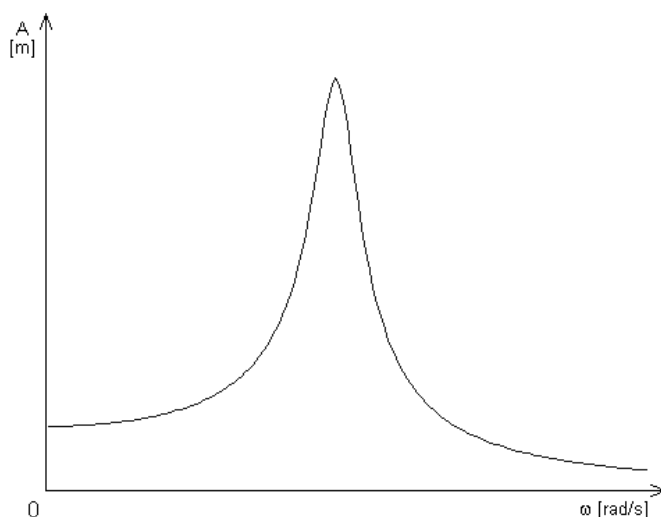
$$A_{ust} = \frac{F}{m \cdot \sqrt{(\Omega_{tl}^2 + d^2 - w^2)^2 + 4d^2 \cdot w^2}} = \frac{F}{\sqrt{(k - m \cdot w^2)^2 + b^2 \cdot w^2}} \quad (19)$$

[4]

Tento vztah je analytickým vyjádřením amplitudo-frekvenční charakteristiky lineárního tlumeného mechanického oscilátoru s 1° volnosti buzeného harmonickou silou  $F$  s úhlovou frekvencí  $\omega$ .

### 2.2.1 Amplitudo-frekvenční charakteristika lineárního systému

Tvar amplitudo-frekvenční charakteristiky lineárního tlumeného mechanického oscilátoru s 1° volnosti je znázorněn na obrázku 3.



Obr. 3.: Amplitudo-frekvenční charakteristika lineárního systému

Z vyjádření amplitudo-frekvenční charakteristiky (rovnice 19) lze vyjádřit hodnotu amplitudy pro nulovou budící frekvenci, což je hodnota výchylky za předpokladu působení konstantní síly o velikosti  $F$  jako

$$A_{ust}(0) = \frac{F}{k} \quad (20)$$

a hodnotu amplitudy pro budící frekvenci jdoucí k nekonečnu, tj. hodnotu, k níž se amplitudo-frekvenční charakteristika blíží se vzrůstající frekvencí jako

$$A_{ust}(\infty) = \lim_{w \rightarrow \infty} \frac{F}{\sqrt{(k - m \cdot w^2)^2 + b^2 \cdot w^2}} = 0 \quad (21)$$

Hodnotu budící frekvence, při níž je amplituda kmitů největší, lze nalézt jako extrém funkce, tedy jako hodnotu, při níž je první derivace vyjádření amplitudo-frekvenční charakteristiky (rovnice 19) podle budící frekvence  $\omega$  rovna nule

$$-\frac{1}{2} \cdot \frac{F \cdot (-4k \cdot m \cdot w + 4m^2 \cdot w^3 + 2b^2 \cdot w)}{(k^2 - 2k \cdot m \cdot w^2 + m^2 \cdot w^4 + b^2 \cdot w^2)^{3/2}} = 0 \quad (22)$$

Rovnice 22 má tři řešení ve tvaru

$$0, \quad \pm \sqrt{\frac{k}{m} - \frac{b^2}{2m^2}} \quad (23)$$

z nichž pouze jedno vyhovuje oblasti, v níž se hledaný extrém nachází. Hledaná hodnota je tedy

$$\sqrt{\frac{k}{m} - \frac{b^2}{2m^2}} \quad (24)$$

což je hodnota blízká vlastní frekvenci tlumeného mechanického oscilátoru s 1° volnosti. Při hodnotě budící frekvence shodné s vlastní frekvencí jsou vlastní kmity mechanického oscilátoru bez buzení shodné s kmity vnější budící frekvence a jejich složením dojde k rezonanci, což má za následek značný nárůst amplitudy výsledných ustálených kmitů.

## 2.3 Nelineární kmitání

Nelineární kmitání nastane v případě, že alespoň jeden z parametrů pohybové rovnice mechanického oscilátoru má nelineární průběh, tj. je vyjádřen nelineární rovnicí. Tím pádem se celá pohybová rovnice stane nelineární. U takovéto nelineární diferenciální rovnice nelze nalézt analytické vyjádření amplitudo-frekvenční charakteristiky tak jako v případě lineární rovnice, protože v nelineárních systémech neplatí princip superpozice - jednotlivé složky vlnění tedy nelze vzájemně sčítat. Nelineární diferenciální pohybovou rovnicí lze řešit pouze přibližnými metodami, příslušnými k danému typu úlohy. Neexistuje zde obecné řešení, které lze použít na všechny typy úloh.

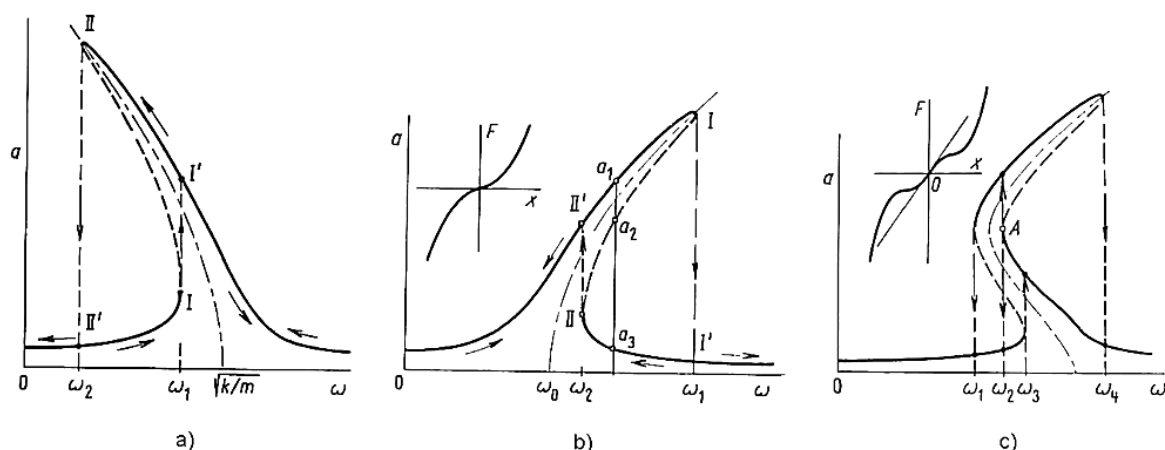
Zpravidla se mohou vyskytnout následující druhy nelinearity:

- nelinearita vratné síly pružiny (odlišnost od Hookova zákona)
- nelinearita tlumení
- kombinace obou

### 2.3.1 Nelinearita vratné síly pružiny

Rozlišujeme více druhů nelinearity vratné síly pružiny. Podle závislosti vratné síly na stlačení pružiny může mít pružina charakteristiku:

- měknoucí (degresivní) - tuhost pružiny  $k$  se při vzrůstajícím stlačení zmenšuje (např.  $F = k_1 \cdot x - k_3 \cdot x^3 \Rightarrow k = k_1 - k_3 \cdot x^2$ , obrázek 4a)
- tuhnoucí (progresivní) - tuhost pružiny  $k$  se při vzrůstajícím stlačení zvětšuje (např.  $F = k_1 \cdot x + k_3 \cdot x^3 \Rightarrow k = k_1 + k_3 \cdot x^2$ , obrázek 4b)
- obecně nelineární - kombinace obou, tuhost pružiny  $k$  se při vzrůstajícím stlačení zmenšuje až do určité hodnoty, od níž se při nadále vzrůstajícím stlačení začne zvětšovat (např.  $F = k_1 \cdot x - k_3 \cdot x^3 + k_5 \cdot x^5 \Rightarrow k = k_1 - k_3 \cdot x^2 + k_5 \cdot x^4$ , obrázek 4c)



Obr. 4.: Amplitudo-frekvenční charakteristiky systémů s nelineární pružinou [2]

Jak je patrné z obrázku 4, nelineární charakteristika pružiny způsobuje naklonění rezonančního vrcholu amplitudo-frekvenční charakteristiky, čímž vzniká oblast víceznačné hodnoty amplitudy ustálených kmitů. [2]

V případě měknoucí (degresivní) charakteristiky pružiny dochází k naklonění rezonančního vrcholu směrem k nižším frekvencím, což ukazuje obrázek 4a, v případě tuhnoucí charakteristiky pak k naklonění směrem k vyšším frekvencím, obrázek 4b. Naklonění rezonančního vrcholu kopíruje skeletovou křivku vyjadřující závislost vlastní úhlové frekvence netlumeného systému na amplitudě kmitů. Pro uvedené příklady tuhnoucí a měknoucí charakteristiky lze skeletovou křivku vyjádřit ve tvaru

$$\omega = \Omega = \sqrt{\frac{k_1}{m} \pm \frac{3}{4} \cdot \frac{k_3}{m} \cdot a^2} \quad (25)$$

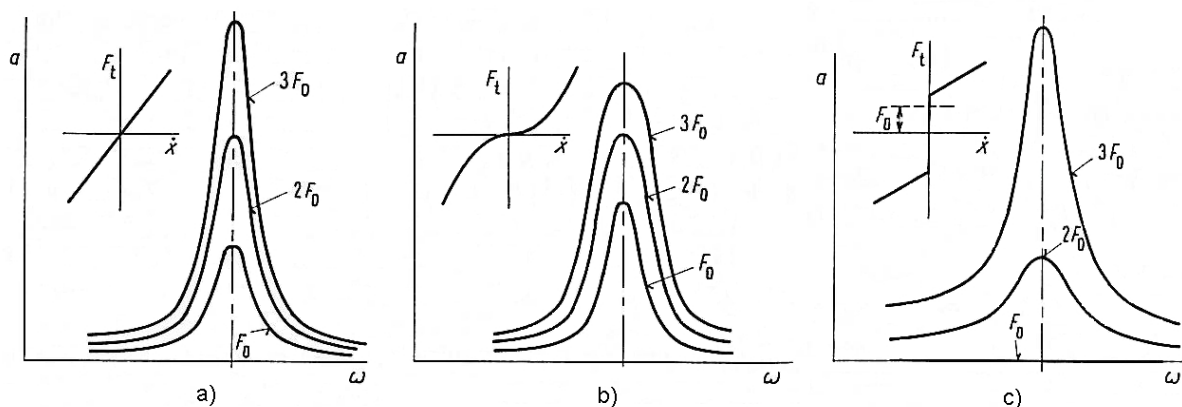
Znaménko ve vztahu určuje charakteristika pružiny. Příklad znaménka „+“ je pro tuhnoucí charakteristiku, znaménka „-“ potom pro charakteristiku měknoucí. Z amplitudo-frekvenčních charakteristik je zřejmé, že řešení nelineární diferenciální pohybové rovnice bude vzhledem k víceznačnosti závislé nejen na parametrech soustavy, ale i na počátečních podmínkách. [2]

### 2.3.2 Nelinearita tlumení

Nelineární tlumení ovlivňuje tvar rezonančního vrcholu i u jinak lineárního systému [2]. Jak již bylo zmíněno, je to způsobeno tím, že vlivem nelineárního členu vyjadřující buzení se celá pohybová rovnice soustavy stává nelineární.

Obdobně jako u nelinearity vratné síly pružiny rozlišujeme nelinearitu odporové síly tlumení podle charakteristiky velikosti odporové síly na rychlosti pohybu na:

- tuhnoucí (progresivní) - velikost odporové síly tlumení se při vzrůstající rychlosti zvětšuje (např.  $F = b \cdot \dot{x}$ , uvedený vztah vzhledem k sudé mocnině nerespektuje znaménko, ve výpočtech a simulacích je proto nutno použít výraz  $F = b \cdot \dot{x} |\dot{x}|$ )
- měknoucí (degresivní) - velikost odporové síly tlumení se při vzrůstající rychlosti zmenšuje (např.  $F = b_1 \cdot \dot{x} - b_2 \cdot \dot{x}^2$ )
- obecně nelineární - kombinace obou, velikost odporové síly tlumení se při vzrůstající rychlosti zvětšuje až do určité hodnoty, od níž se při nadále vzrůstající rychlosti začne zmenšovat (např.  $F = \frac{b_1 \cdot \dot{x}}{1 + b_2 \cdot \dot{x}^2}$ )



Obr. 5.: Amplitudo-frekvenční charakteristiky – nelineární tlumení [2]

Obrázek 5 zobrazuje průběhy rezonančních vrcholů pro tři velikosti budící síly  $F_0$ . Obrázek 5a znázorňuje lineární systém, obrázek 5b a 5c pak systém nelineární. V případě lineárního systému je výška rezonančních vrcholů přímo úměrná amplitudě budící síly  $F_0$ , v případě nelineárního nikoli. Pokud jde o progresivní tlumení, výška rezonančních vrcholů roste s amplitudou budící síly  $F_0$  pomaleji, v případě degresivního tlumení potom rychleji. [2]

Z výše uvedeného je zřejmé, že konstrukce amplitudo-frekvenční charakteristiky nelineárního systému je vzhledem k nemožnosti jejího analytického vyjádření značně problematická. Nabízí se tedy možnost její konstrukce pomocí vhodné metody simulace, což je jedním z cílů této práce.



### 3. KONSTRUKCE AMPLITUDO-FREKVENČNÍ CHARAKTERISTIKY

Pro konstrukci amplitudo-frekvenční charakteristiky pomocí simulace chování systému je nutné nejprve vytvořit matematický model soustavy pomocí pohybové rovnice. Tuto rovnici je následně nutno vyřešit pomocí vhodné metody numerické integrace za pomoci vhodného matematického softwaru pro patřičný počet úhlových frekvencí budící síly  $F$  tak, abychom s dostatečnou hustotou pokryli zájmovou oblast amplitudo-frekvenční charakteristiky, kterou chceme zkonstruovat.

Většina používaných matematických softwarů obsahuje numerickou metodu řešení diferenciálních rovnic Runge-Kutta 4. řádu s odhadem chyby 5. řádu.

#### 3.1 Metody Runge-Kutta

Metody Runge-Kutta patří mezi nejpoužívanější numerické metody, protože vynikají svou jednoduchostí a v případě vyšších řádů (nejpoužívanější 4. řád) i relativně vysokou přesností. Jedná se o jednokrokové metody vycházející z Taylorova rozvoje. Využívají aproximace derivace funkce  $f(x, y)$  hodnotou funkce  $f(x, y)$ . Řád přesnosti metody je určen počtem bodů, ve kterých je nutné hodnotu funkce vyčíslit.

Nejjednodušší Runge-Kuttovou metodou je metoda Eulerova. Jedná se o metodu Runge-Kutta prvního řádu přesnosti. Řešení  $y_{n+1}$  je hledáno pouze za pomoci předchozí hodnoty funkce  $y_n$ , vzdálené od řešení o přírůstek  $h$ , podle explicitního vztahu

$$y_{n+1} = h \cdot f(x_n, y_n) \quad (26)$$

jehož geometrickou interpretací je bod  $(x_{n+1}, y_{n+1})$  ležící na přímce procházející bodem  $(x_n, y_n)$  se směrnici  $f(x_n, y_n)$ . Eulerova metoda je tedy lineární extrapolací. [6]

Z výše uvedeného je zřejmé, že chyba metody má kumulativní charakter, tedy že s počtem užití metody neustále vzrůstá. Zmenšení přírůstku  $h$  bohužel na kumulativnost chyby nemá žádný vliv.

Pro odstranění kumulativnosti chyby se využívají metody vyšších řádů. Tyto metody kromě předchozí hodnoty využívají navíc hodnotu funkce v několika dalších blízkých bodech, které obecně nemusí být předchozími body řešení. Nejpoužívanější Runge-Kuttova metoda čtvrtého řádu využívá pro vyjádření následujícího stavu hodnoty na začátku, uprostřed a na konci intervalu  $\langle x_n, x_{n+1} \rangle$  a je dána pomocí vztahů

$$\begin{aligned} k_1 &= f(x_n, y_n) \\ k_2 &= f\left(x_n + \frac{h}{2}, y_n + k_1 \cdot \frac{h}{2}\right) \\ k_3 &= f\left(x_n + \frac{h}{2}, y_n + k_2 \cdot \frac{h}{2}\right) \\ k_4 &= f(x_n + h, y_n + k_3 \cdot h) \\ y_{n+1} &= y_n + \frac{h}{6} \cdot (k_1 + 2k_2 + 2k_3 + k_4) \end{aligned} \quad (27)$$

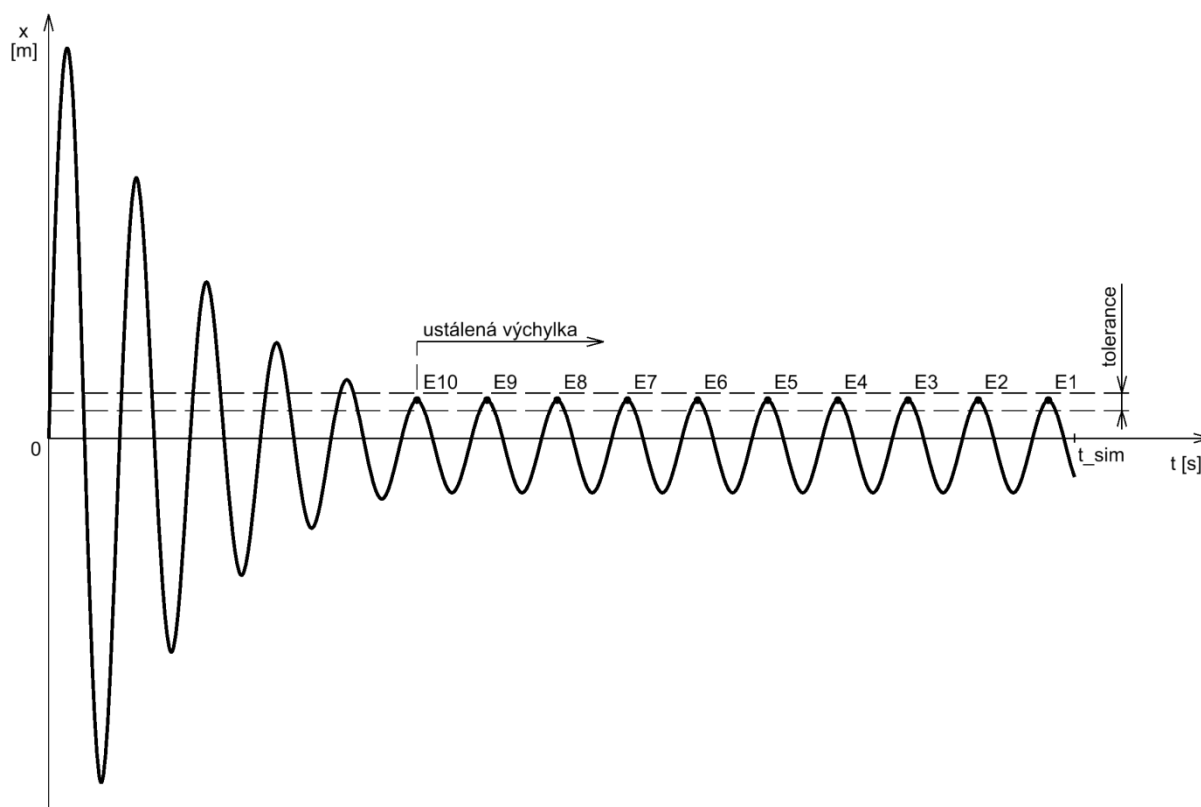
Je vidět, že s rostoucím řádem Runge-Kuttovy metody se zvyšuje počet výpočtů nutných pro vyjádření následujícího stavu. Vzhledem k tomu, že však není nutné vyčíslit hodnotu derivace funkce, jsou výpočty jednoduché a v případě zpracování pomocí výpočetní techniky se Runge-Kuttovy metody vyznačují relativně vysokou rychlostí.

#### 3.2 Nalezení ustálené výchylky kmitů

Pro nalezení ustálené výchylky kmitů je nutné nejprve numericky vyřešit pohybovou rovnici soustavy, tj. získat hodnoty výchylky a rychlosti pro diskrétní časové hodnoty v určitém časovém úseku. V našem případě od nuly do doby simulace označené jako  $t_{sim}$ . Tyto hodnoty společně s příslušnými diskrétními hodnotami času je nutno vhodným

způsobem reprezentovat, pro následnou práci s nimi. Vypočtené hodnoty lze reprezentovat například jako prvky vektorů, s nimiž umí všechny matematické softwary bez problémů pracovat, navíc je ve většině případů dovedou mezi sebou sdílet.

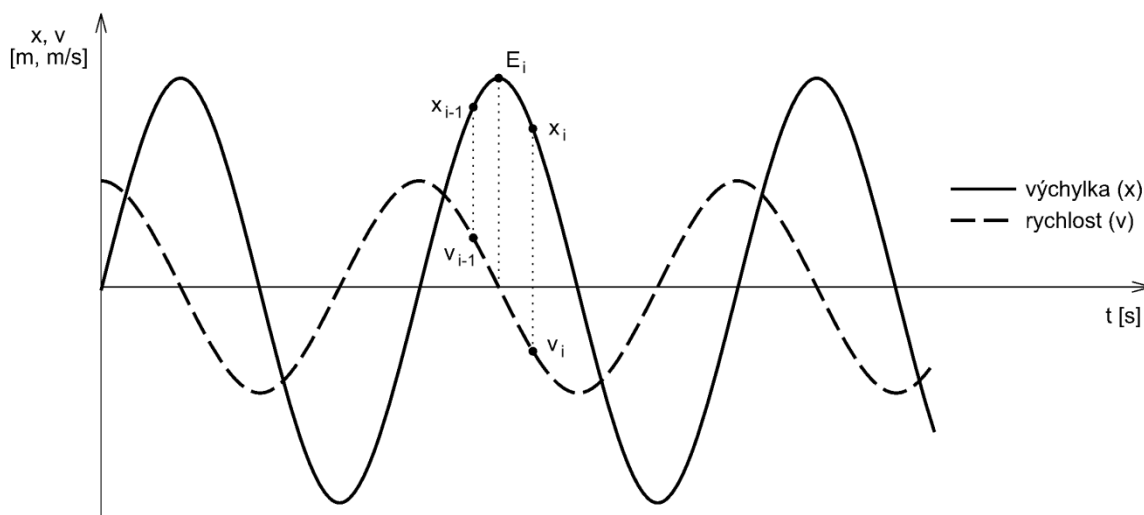
Získané hodnoty je nutné vyšetřit s cílem zjistit, zda se již výchylka kmitů za dobu simulace ustálila, a v případě, že ano, na jaké hodnotě. V případě, že se hodnota výchylky za dobu simulace neustálila, tj. ještě neodezněl přechodový děj, je nutné zvětšit dobu simulace. Zda je výchylka ustálená zjistíme tak, že vzájemně porovnáme deset posledních hodnot maximální výchylky a ověříme, zda se všechny vyskytují v předem definované toleranci, jak znázorňuje obrázek 6.



Obr. 6.: Hledání ustálené výchylky kmitů

Maximální hodnoty výchylky nalezneme jako místa lokálních extrémů tak, že nalezneme místa, kde hodnota derivace funkce kterou reprezentují, je rovna nule, tj. kde je nulová hodnota rychlosti. Vzhledem k tomu, že máme k dispozici hodnoty rychlosti pouze v diskrétních časových okamžicích, nalezneme dvě hodnoty takové, aby hodnota předchozí měla opačné znaménko než hodnota následující. Protože kmitání nemusí být z pravidla symetrické, je lepší vzájemně porovnávat pouze příslušné půlvlny. Zaměříme se tedy pouze na analýzu kladných půlvln. Jak je patrné z obrázku 7, kladná půlvlna výchylky nastává v momentě průchodu hodnoty rychlosti nulou ve směru od kladných hodnot k záporným. Budeme tedy hledat takové dvě sousední hodnoty rychlosti, aby předcházející hodnota měla kladné znaménko a následující hodnota měla znaménko záporné. Když tyto hodnoty nalezneme, přiřadíme k nim příslušné hodnoty výchylky, tj. hodnoty výchylky ve stejných diskrétních časových okamžicích, jako jsou nalezené hodnoty rychlosti. Následně pomocí vhodné metody interpolace nalezneme hodnotu maximální výchylky, tedy hodnotu výchylky pro případ, kdy je hodnota rychlosti nulová. Abychom zvýšili přesnost interpolace, je vhodné pro interpolaci využít širší okolí zájmové oblasti, než pouhé dvě hodnoty.





Obr. 7.: Nalezení maximální výchylky

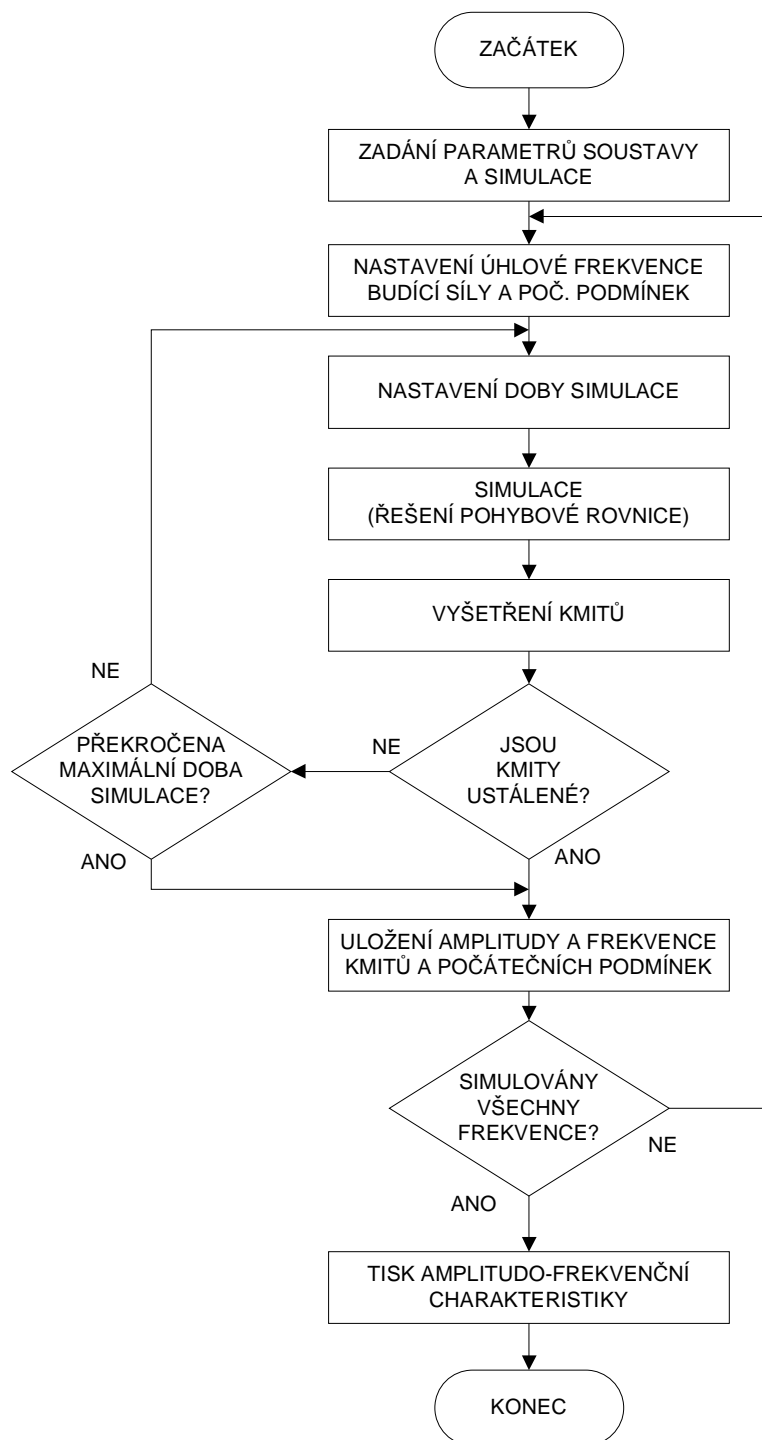
Porovnání, zda leží takto získané hodnoty maximální výchylky v dané toleranci, provedeme tak, že z hodnot uděláme aritmetický průměr a od hodnoty tohoto průměru odečteme a přičteme polovinu tolerance. Tak získáme minimální a maximální hodnotu vymezující toleranci a pro všechny hodnoty maximální výchylky ověříme, zda leží mezi těmito hodnotami. V případě, že ano, označíme aritmetický průměr hodnot jako hodnotu ustálené výchylky pro danou úhlovou frekvenci  $\omega$  budící síly  $F$  a počáteční podmínky  $\dot{x}(0), x(0)$ .

Protože však, jak již bylo zmíněno, ustálená výchylka nelineárních kmitavých systémů nezávisí pouze na budící frekvenci, nýbrž i na počátečních podmínkách, je nutné kromě hodnoty ustálené výchylky pro každou úhlovou frekvenci  $\omega$  zjistit i ony počáteční podmínky. Protože chceme zachovat kontinuitu při průchodu daného frekvenčního pásma, budeme zároveň se zjišťováním výchylky ustálených kmitů pro každou vyšetřovanou úhlovou frekvenci budící síly zjišťovat i hodnotu výchylky a rychlosti v určitém čase. Tyto hodnoty jsou právě totiž hodnotami počátečních podmínek pro následující vyšetřovanou úhlovou frekvenci budící síly. Vzhledem k tomu, že řešení pohybové rovnice začínáme vždy v čase  $t = 0$ , kde je amplituda budící síly  $F \cdot \sin(\omega \cdot t) = 0$  s následnou kladnou půlvlnou, musíme pro zachování kontinuity nalézt hodnoty výchylky a rychlosti v takovém čase, aby se v něm budící síla chovala shodně. Pro nalezení těchto hodnot použijeme obdobný způsob jako pro nalezení hodnoty maximální výchylky.



#### 4. IMPLEMENTACE METODY DO MATEMATICKÝCH SOFTWARE

Ze zmíněného postupu konstrukce amplitudo-frekvenční charakteristiky lze sestavit základní vývojový diagram implementace metody do matematických softwarů. Tento diagram je znázorněn na obrázku 8.



Obr. 8.: Základní vývojový diagram

## 4.1 Implementace v programu MATLAB

### 4.1.1 MATLAB

MATLAB je asi nejrozšířenější softwarové prostředí pro numerické výpočty, analýzu a vizualizaci dat, vyvíjený a distribuovaný spol. The Mathworks. Název je odvozen ze slov Matrix Laboratory – laboratoř s maticemi. Jak již sám název napovídá, celé prostředí je přizpůsobeno práci s maticemi.

Programové prostředí MATLAB využívá vlastní programovací jazyk. Jedná se o jazyk interpretovaný. Příkazy jazyka se zapisují přímo do příkazové řádky interpretru, jejich vykonávání se spustí po stisknutí klávesy *Enter*. V případě potřeby psát větší množství příkazů, které je potřeba spustit současně, je možné tyto příkazy napsat do souboru s příponou *\*.m*, tzv. *m-file*. Příkazy z tohoto souboru se pak spustí zapsáním jména tohoto souboru do příkazové řádky interpretru a stiskem klávesy *Enter*. Programové prostředí MATLAB obsahuje editor jazyka MATLAB, v němž lze jednotlivé *m-file* vytvářet a zvýrazňuje příkazy jazyka MATLAB. Novější verze obsahují i jednoduchý debugger s jehož pomocí lze program snadněji ladit. Funkce se zapisují do zvláštních samostatných *m-file* ve formátu „*název funkce*“.m a musí být uloženy ve stejném adresáři, jako je hlavní *m-file*, z něž jsou volány, příp. v tzv. MATLAB path, což je výchozí složka pro umístění funkcí MATLABu. V této složce jsou umístěny všechny předdefinované funkce dodávané společně s programem. Jazyk MATLAB obsahuje většinu běžných datových typů a rozhodovacích struktur, svou syntaxí se nejvíce podobá jazyku C. Podrobné informace o datových typech a příkazech programovacího prostředí MAPLE lze nalézt v nápovědě, případně v některé z mnoha učebnic, které byly o tomto prostředí publikovány. Podrobný rozbor jednotlivých příkazů by byl tedy pouhým opakováním a není ani cílem této práce.

### 4.1.2 Program pro MATLAB

Dle vývojového diagramu program začíná zadáním parametrů soustavy. V této části je současně zajištěno vymazání předchozích hodnot z paměti, vyčištění obrazovky a nastavení simulovaných hodnot (amplituda, rozsah a krok frekvencí budící síly, počáteční podmínky) a parametrů simulace (základní doba simulace, maximální doba simulace, a tolerance hodnoty ustálených kmitů). Uvedená tolerance odpovídá rozkmitu o  $\pm 0,5\%$ , tedy 1% toleranci.

```
%mazání
clc
clear
%nastavení parametrů soustavy
m = 5;
k = 300;
k3 = 25;
b = 6;
%nastavení simulovaných hodnot
F = 100;
omega_min = 0.1;
omega_max = 16;
krok_omega = 0.1;
poc_vychylka = 0;
poc_rychlost = 0;
%nastavení parametrů simulace
doba_sim = 500;
max_t_sim = 2000;
tolerance = 0.005;
```

Následuje cyklus opakování simulace a vyšetřování průběhů hodnot výsledků pro všechny úhlové frekvence budící síly *F*. Jedná se o nepodmíněný cyklus s počtem operací takovým, aby odpovídal počtu vyšetřovaných úhlových frekvencí budící síly *F*.

```

for i4 = 1:1:(omega_max-omega_min)/krok_omega+1
    .
    .
    .
end

```

Uvnitř tohoto cyklu se nachází části pro nastavení úhlové frekvence budící síly  $F$ , počátečních podmínek a doby simulace. Pro nastavení úhlové frekvence budící síly jsou uvedeny dvě možnosti, z nichž jedna je pro průchod simulace frekvenčním pásmem ve směru od vyšších frekvencí k nižším a druhá naopak. Z těchto možností je nutné si vybrat jen jednu.

```

%výpočet omega pro případ průchodu od nižších frekvencí směrem k vyšším
omega = omega_min-krok_omega+i4*krok_omega
%pro případ průchodu směrem od vyšších k nižším
%omega = omega_max+krok_omega-i4*krok_omega
%nastavení počátečních podmínek
y0 = [poc_vychylka; poc_rychlost];
%nastavení doby simulace
tspan = [0, doba_sim];
%nastavení parametrů numerické metody
options = '';

```

Následuje nejdůležitější část, která zajišťuje běh simulace a následné vyšetřování kmitů. Jedná se o vnořený cyklus s několika dalšími dílčími částmi uvnitř. Kostra cyklu pouze kontroluje, zda jsou kmity již ustálené a zda nebyla překročena maximální doba simulace. Zároveň v případě, že kmity na konci simulace nejsou ustálené, zdvojnásobuje dobu simulace.

```

while mimo_toleranci == 1 && tspan(2) <= max_t_sim
    .
    .
    .
    tspan(2)= tspan(2)*2;
end

```

Uvnitř cyklu zajišťujícího simulaci a vyšetřování kmitů se dále nachází dvě části pro vlastní simulaci a pro následné vyšetření výsledných kmitů. Simulaci zajišťuje funkce `ode45`, což je implementace Runge-Kuttovy metody 4. řádu s odhadem chyby pro 5. řád, která je součástí MATLABu. Funkci `ode45` se při volání předává několik parametrů. Nejprve název souboru, v němž je definována diferenciální rovnice  $n$ -tého řádu převedená na  $n$  rovnic prvního řádu ve formě funkce. V našem případě se jedná o definici dvou rovnic prvního řádu. Tento soubor musí ležet ve stejném adresáři jako soubor, z něhož jej voláme, případně v adresáři, kde leží funkce `ode45`. Dále se funkci `ode45` předávají dva vektory. Vektor času, obsahující čas počátku konce intervalu, pro nějž chceme rovnici řešit a vektor počátečních podmínek. Za těmito vektory následují parametry řešení a za nimi vstupní proměnné figurující v diferenciálních rovnicích.

```

%volání funkce
[T,Y] = ode45('dif_rce_oscil',tspan,y0,options,k,k3,m,b,F,omega);

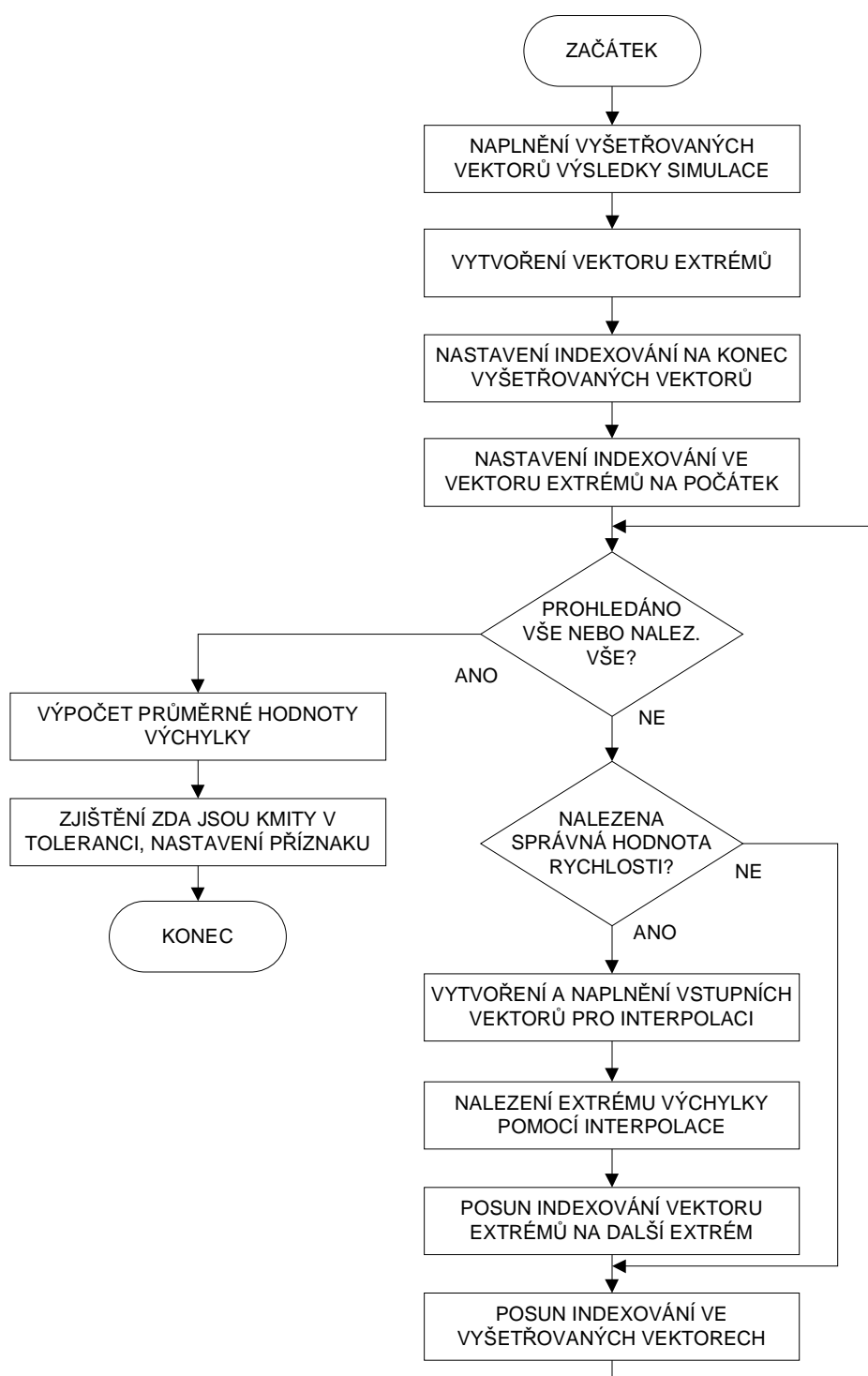
```

Vyjádření pohybové diferenciální rovnice v souboru *dif\_rce\_oscil.m*:

```

function dy = dif_rce_oscil(t,y,options,k,k3,m,b,F,omega)
dy(1,1) = y(2,1);
dy(2,1) = -1/m*(b*y(2,1)+k*y(1,1)+k3*y(1,1)^3-F*sin(omega*t));
end

```



Obr. 9.: Vývojový diagram části pro vyšetření kmitů v prostředí MATLAB

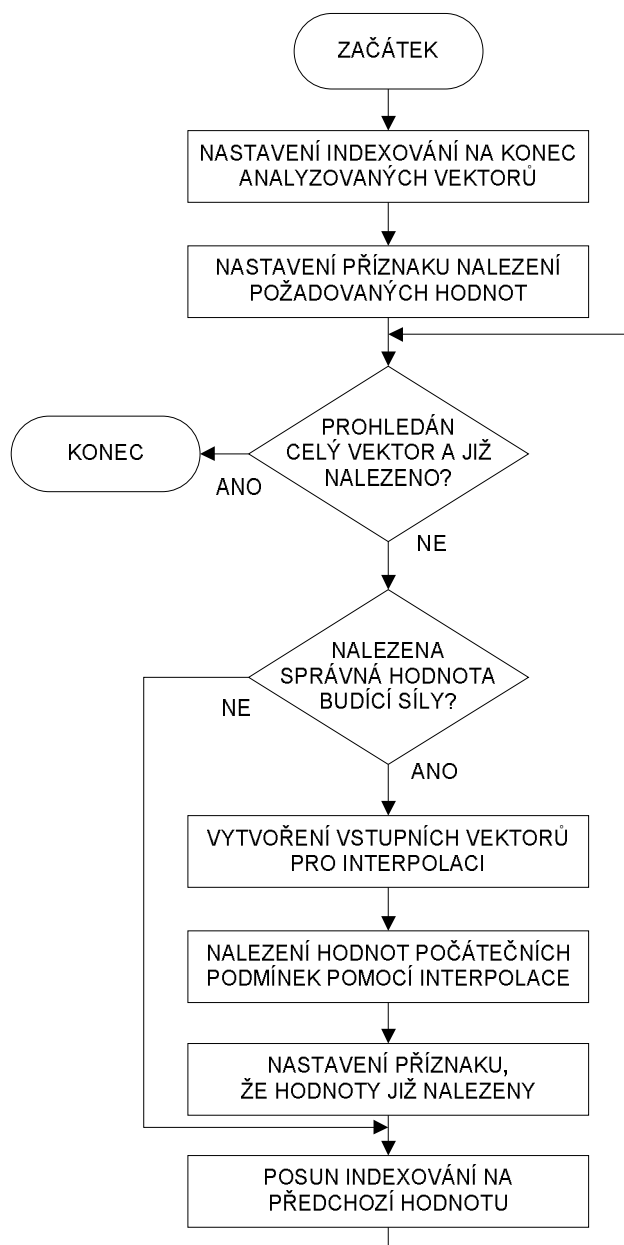
Část kódu zajišťující vyšetření kmitů zachycuje vývojový diagram na obrázku 9. Nejprve se naplní patřičné vyšetřovací vektory výslednými hodnotami simulace. Mohl bychom pracovat přímo s maticí výsledku  $Y$ , práce s vektory je však pohodlnější a přehlednější. Pokračuje se vytvořením vektoru extrémů, tento vektor se vytváří naplněný nulami pro případ, že bychom nenalezli dostatečný počet extrémů. Nuly by nám pro tento

případ zajistily, že kmity budou vyhodnoceny jako doposud neustálené. Vektor by mohl být vytvořen pomocí cyklu, vzhledem k jeho délce je však kratší a jednodušší jej nadefinovat přímo. Po nastavení počátečního indexování ve vektorech následuje cyklus, v němž se opakovaně hledají požadované hodnoty rychlosti, resp. jejich indexy ve vektoru takové, aby bylo nalezeno místo, kde rychlost přechází z kladných hodnot do záporných, tedy místo kladného extrému výchylky. Pokud jsou tyto indexy nalezeny, vytvoří se vektory pro následnou interpolaci. Pro zvýšení přesnosti interpolace tyto vektory naplníme deseti hodnotami na každou stranu od hledaného místa pro každou hodnotu a interpolujeme je pomocí metody Hermitova interpolačního polynomu (parametr *'pchip'* ve volání interpolace). Po provedení interpolace její výsledek uložíme na patřičné místo ve vektoru extrémů a posuneme jeho indexování. V případě, že již bylo nalezeno všech deset extrémů, vypočteme z těchto hodnot aritmetický průměr tak, že pomocí cyklu provedeme jejich součet a podělíme jej deseti. Nakonec pomocí cyklu porovnáme všechny extrémy, zda se nacházejí uvnitř tolerančního pásma, a nastavíme patřičný příznak pro rozhodování nadřazeného cyklu.

```
%vyprázdnění vektorů proto aby nové nebyly kratší než pro předchozí omegu
vychylka = 0;
rychlost = 0;
%naplnění vektorů k vyšetření výsledky simulace
for i = 1:1:length(Y)
    vychylka(i) = Y(i,1);
    rychlost(i) = Y(i,2);
end
%vytvoření vektoru extrémů
extrem = [0 0 0 0 0 0 0 0 0 0];
%nastavení indexování na počátek, -10 z důvodu interpolace, tou 10 přibude
i = length(vychylka)-10;
i2 = 1;
    while i >= 10 && i2 <= 10
        if rychlost(i) < abs(rychlost(i)) &&
            rychlost(i-1) == abs(rychlost(i-1))
            %naplnění vyšetřovaných prom. 10 hodnotami okolo extrému
            for i3 = 1:1:20
                x_interp(i3) = vychylka(i-11+i3);
                v_interp(i3) = rychlost(i-11+i3);
            end
            %nalezení extrému pomocí interpolace
            extrem(i2) = interp1(v_interp,x_interp,0,'pchip');
            %posun indexování vektoru extrémů
            i2 = i2+1;
        end
        %posun indexování vyšetřovaných vektorů
        i = i-1;
    end
    %výpočet průměrné hodnoty výchylky
    prumerna_vychylka = 0;
    for i = 1:1:10
        prumerna_vychylka = prumerna_vychylka+extrem(i);
    end
    prumerna_vychylka = prumerna_vychylka/10;
    %kontrola, zda jsou kmity v toleranci
    mimo_toleranci = 0;
    for i = 1:1:10
        if extrem(i) > prumerna_vychylka+prumerna_vychylka*tolerance ||
            extrem(i) < prumerna_vychylka-prumerna_vychylka*tolerance
            mimo_toleranci = 1;
        end
    end
end
```

Po vyšetření kmitů následuje část zajišťující uložení amplitudy a úhlové frekvence ustálených kmitů a počátečních podmínek. Amplituda a úhlová frekvence se ukládají, jak již bylo zmíněno, do příslušných vektorů.

```
AMPLITUDA(i4) = prumer_na_vychylka;  
OMEGA(i4) = omega;
```



Obr. 10.: Vývojový diagram části pro nalezení počátečních podmínek v prostředí MATLAB

Nalezení počátečních podmínek znázorňuje vývojový diagram na obrázku 10. Z vývojového diagramu je zřejmé, jak bude kód této části programu v jazyce MATLAB vypadat. Po provedení počátečního nastavení se postupně procházejí vektory výsledků simulace a hledá se taková hodnota indexů, aby budící síla v čase daném prvkem vektoru diskrétního času  $T$  na pozici  $i$  byla kladná a v čase předchozím, tj. na pozici  $i+1$  byla záporná. Následně si tyto hodnoty, resp. hodnoty budící síly pro tyto hodnoty času společně



s příslušnými hodnotami výchylky a rychlosti uložíme do vektorů, které použijeme jako vstupy pro funkci interpolace. Pomocí interpolace získáme hodnoty výchylky a rychlosti v místě, kde budící síla prochází nulou ve směru od záporných hodnot do kladných.

```
%nastavení indexování na počátek
i = length(vychylka);
%nastavení na 1 proto, aby cyklus while proběhl minimálně 1x
opak = 1;
while i > 1 && opak == 1
    if sin(omega*T(i)) == abs(sin(omega*T(i))) &&
       sin(omega*T(i-1)) < abs(sin(omega*T(i-1)))
        %příprava vektorů pro interpolaci
        sin_interp = [sin(omega*T(i-1)),sin(omega*T(i))];
        x_interp = [vychylka(i-1),vychylka(i)];
        v_interp = [rychlost(i-1),rychlost(i)];
        %nalezení hodnot pomocí interpolace
        poc_vychylka = interp1(sin_interp,x_interp,0);
        poc_rychlost = interp1(sin_interp,v_interp,0);
        %již nalezeno, dále neopakuj
        opak = 0;
    end
    %posun indexování
    i=i-1;
end
```

Pokud jsou zjištěny hodnoty výchylek pro všechny požadované úhlové frekvence budící síly, následuje tisk grafu amplitudo-frekvenční charakteristiky, včetně popisu.

```
%tisk amplitudofrekvenční charakteristiky
plot(OMEGA,AMPLITUDA);
title('Amplitudo-frekvenční charakteristika');
xlabel('\omega [rad/s]');
ylabel('A [m]');
```

V případě použití pro řešení systémů s nelineárním tlumením a lineární vratnou silou pružiny je možné z kódu vypustit část pro nalezení počátečních podmínek, protože jak již bylo zmíněno, nelinearita tlumení nemá vliv na naklánění rezonančního vrcholu. Řešení pohybové rovnice tedy není závislé na počátečních podmínkách, mohou tedy zůstat stále nulové.

## 4.2 Implementace v programu MAPLE

### 4.2.1 MAPLE

MAPLE je softwarové prostředí vyvinuté na univerzitě Waterloo v Kanadě, v současné době distribuované společností Waterloo Maple Inc. Jeho název zřejmě vyplývá ze spojení slov Mathematics pleasure – matematika potěšením. MAPLE náleží do systémů počítačové algebry, označovaných zkratkou CAS (Computer Algebra Systems). Tyto systémy se vyznačují tím, že na rozdíl od standardních programů pro matematické výpočty modelují matematické operace se symbolickými výrazy. Proměnné tedy představují neznámé bez přiřazení numerické hodnoty. MAPLE tak uchovává čísla v přesném tvaru (např.  $1/3$  místo 0.3333,  $\pi$  místo 3.1415, atd.). Tím pádem dává MAPLE odpovědi s podstatně vyšší přesností, než běžné numerické softwary pracující pouze s čísly v pohyblivé řádové čárce. [8]

MAPLE využívá pro své programování speciální programovací jazyk MPJ. Tento jazyk se nijak významně neliší od jiných programovacích jazyků. Stejně jako jazyk MATLAB i jazyk MPJ obsahuje většinu běžných datových typů a rozhodovacích struktur. Nejvíce se jazyk MPJ podobá jazyku Pascal. Na rozdíl od běžných programovacích jazyků používá

MAPLE automatické datové typy, tj. automaticky přidělí proměnné datový typ na základě přiřazované hodnoty. Podrobné informace o datových typech a příkazech programu MAPLE lze nalézt v nápovědě programu vyvolané po stisku klávesy *F1* v programovém okně, případně zadáním *help(předmět dotazu)*; do příkazové řádky. Zároveň stejně jako v případě MATLABu existuje značné množství učebnic, kde jsou jednotlivé příkazy dopodrobna rozebrány. Podrobný rozbor jednotlivých příkazů by byl opět pouhým opakováním.

#### 4.1.2 Program pro MAPLE

Dle základního vývojového diagramu program začíná stejně jako v případě MATLABu zadáním parametrů soustavy. V této části je taktéž zajištěno vymazání předchozích hodnot z paměti a nastavení simulovaných hodnot (amplituda, rozsah a krok frekvencí budící síly, počáteční podmínky) a parametrů simulace (maximální doba simulace, tolerance hodnoty ustálených kmitů, tolerance nulové hodnoty). Zároveň je zde definována pohybová rovnice systému a jsou zde inicializovány vektory pro uložení hodnot ustálených výchylek a příslušných úhlových frekvencí  $\omega$  budící síly  $F$ . Příkazu *trunc* je u výpočtu délky vektorů použito z důvodu typové kompatibility. MAPLE používá automatické datové typy a v případě dělení ve výpočtu je výsledek automaticky typu *real*, který nelze použít pro indexování ve vektorech. Výsledek operace *trunc* je automaticky typu *integer* a ten pro indexování ve vektorech použít lze, přitom výsledek není operací *trunc* ovlivněn, protože se stejně jedná o celé číslo.

```
#vymazání předchozích hodnot
restart;
#nastavení parametrů soustavy
m := 5;
k := 300;
k3 := 200;
b := 6;
#nastavení simulovaných hodnot
F := 100;
omega_min := 0.5;
omega_max := 15.0;
krok_omega := 0.1;
x_poc := 0;
v_poc := 0;
#definice pohybové rovnice systému
rce_syst := diff(diff(x(t),t),t) = -1/m*(b*diff(x(t),t)+k*x(t)+k3*x(t)^3
-F*sin(omega*t));
#nastavení parametrů simulace
tolerance_x := 0.005;
tolerance_v := 0.00001;
max_t_sim := 2000;
#inicializace vektorů ustálené výchylky a frekvence budící síly
pocet := trunc((omega_max-omega_min)/krok_omega+1);
AMPLITUDA := Vector(1..pocet);
OMEGA := Vector(1..pocet);
```

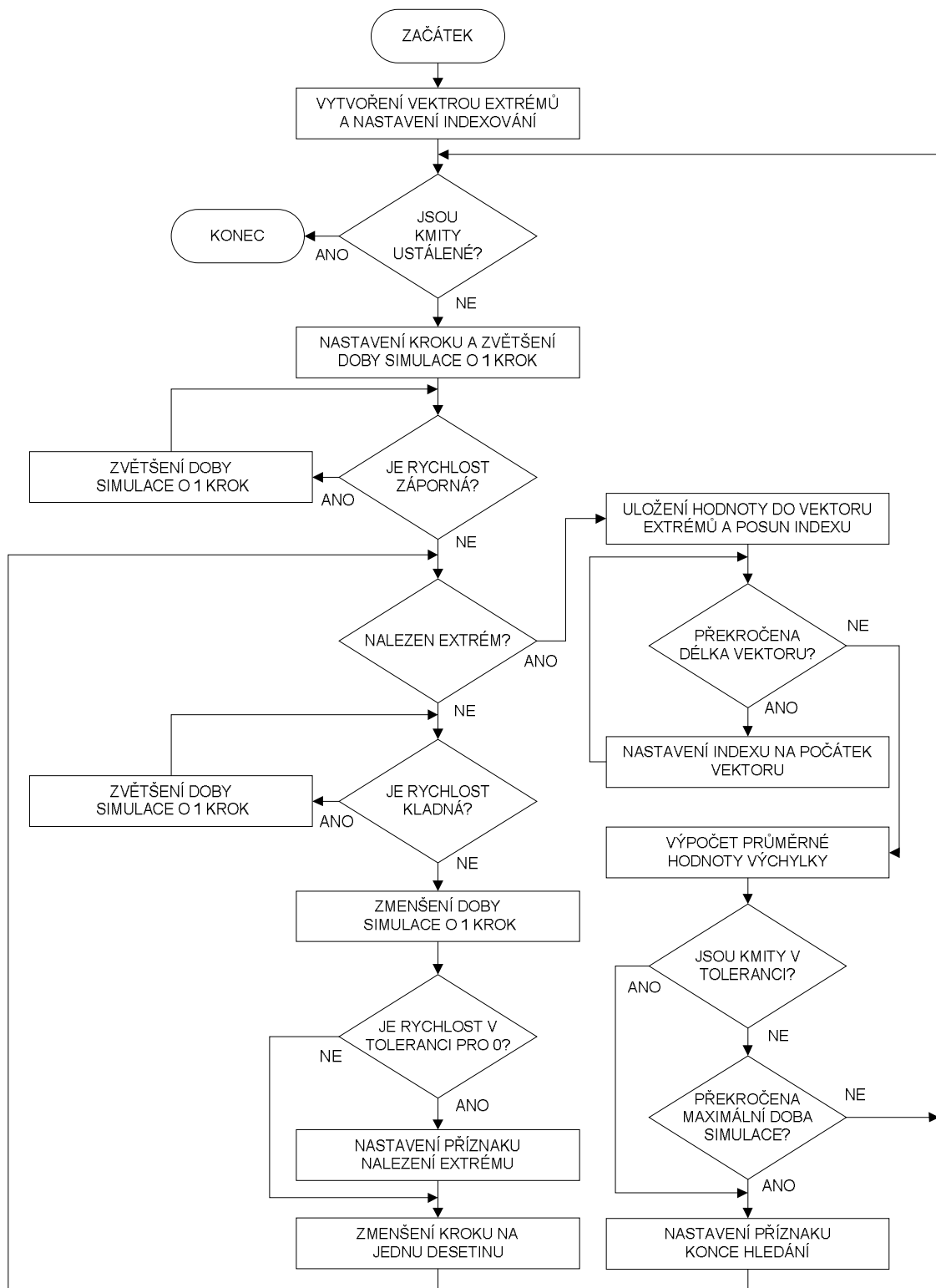
Následuje cyklus opakování simulace a vyšetřování průběhů hodnot výsledků pro všechny úhlové frekvence budící síly  $F$ . Jedná se o nepodmíněný cyklus s počtem operací takovým, aby odpovídal počtu vyšetřovaných úhlových frekvencí budící síly  $F$ .

```
for i4 from 1 to pocet do
.
.
.
end do;
```

Uvnitř tohoto cyklu se nachází části pro nastavení úhlové frekvence budící síly  $F$ , počátečních podmínek a doby simulace. Pro nastavení úhlové frekvence budící síly jsou opět uvedeny dvě možnosti, z nichž jedna je pro průchod simulace frekvenčním pásmem ve směru od vyšších frekvencí k nižším a druhá naopak. Z těchto možností je nutné si opět vybrat pouze jednu, stejně jako v případě MATLABu. MAPLE na rozdíl od MATLABu umožňuje řešit diferenciální rovnici pro předem nedefinovaný časový rozsah. Toto je zajištěno parametrem  $\text{maxfun}=0$ . Při zadání příkazu *dsolve* se provede pouze obecné „předřešení“ diferenciální rovnice a řešení v konkrétním čase je vypočítáno až v momentě, kdy se na něho dotazuje. Z toho důvodu není nutné nastavovat dobu simulace. Stejně tak není nutné vytvářet vektor počátečních podmínek, protože se do funkce *dsolve* dosazují jednotlivě.

```
#výpočet omega pro případ průchodu od nižších frekvencí směrem k vyšším
omega := omega_min-krok_omega+i4*krok_omega:
#výpočet omega pro případ průchodu směrem od vyšších k nižším
#omega := omega_max+krok_omega-i4*krok_omega:
#řešení diferenciální rovnice pro předem neurčený časový úsek (maxfun=0)
reseni := dsolve([rce_syst,x(0) = x_poc,D(x)(0) = v_poc],numeric,maxfun=0):
```

Po zmíněném „předřešení“ pohybové rovnice následuje vyšetření kmitů. Toto je zajištěno několika vnořenými cykly, jak je patrné z vývojového diagramu na obrázku 11. Nejprve se vytvoří vektor extrémů a nastaví se jeho indexování na počátek. Následuje cyklus, který se opakuje do doby, než je nalezena ustálená výchylka nebo je překročena maximální doba simulace. Uvnitř cyklu se nejprve nastaví základní krok jako přibližně jedna desetina periody kmitu. Touto hodnotou je zajištěno dostatečně rychlé vyšetřování kmitů a zároveň je zajištěno, aby nebyly přeskokovány celé půlperiody. Po definování základního kroku se tento krok přičte k času simulace a zjistí se hodnota rychlosti v tomto čase. Následuje cyklus, který neustále zvyšuje čas simulace do doby, než je nalezena kladná půlvlna kmitu rychlosti. Poté, co je kladná půlvlna nalezena, následuje hledání jejího konce, pomocí postupného dělení kroku. Hledání končí v momentě, kdy je hodnota rychlosti v toleranci pro nulovou hodnotu definované na počátku programu. Když je extrém nalezen, pokračuje se jeho uložením do vektoru extrémů s následným zvýšením indexování ve vektoru. Dále se provede kontrola, zda nebyla překročena délka vektoru. Pokud ano, nastaví se indexování na počátek. Na počátek se nastaví pro případ, že hodnoty ve vektoru nejsou v toleranci, tj. výchylka ještě není ustálená, potom je potřeba hledat další extrém a tímto přepsat nejstarší nalezený extrém, tj. první. Za předpokladu, že výchylka opět nebude ustálená, přepíše se opět nejstarší extrém, tj. druhý atd. Následuje výpočet průměrné hodnoty výchylky a porovnání, zda jsou kmity v toleranci. Tyto části jsou řešeny pomocí cyklů obdobně jako v případě MATLABu. Na konci cyklu se provede kontrola, zda nebyla překročena maximální doba simulace a případně se nastaví patřičný příznak.



Obr. 11.: Vývojový diagram části pro vyšetření kmitů v prostředí MAPLE

```

#cyklus pro vyšetření kmitů
#nastavení času simulace na počátek
t_sim := 0:
#vytvoření vektoru extrémů a nastavení indexování
EXTREM := Vector(1..10):

```

```

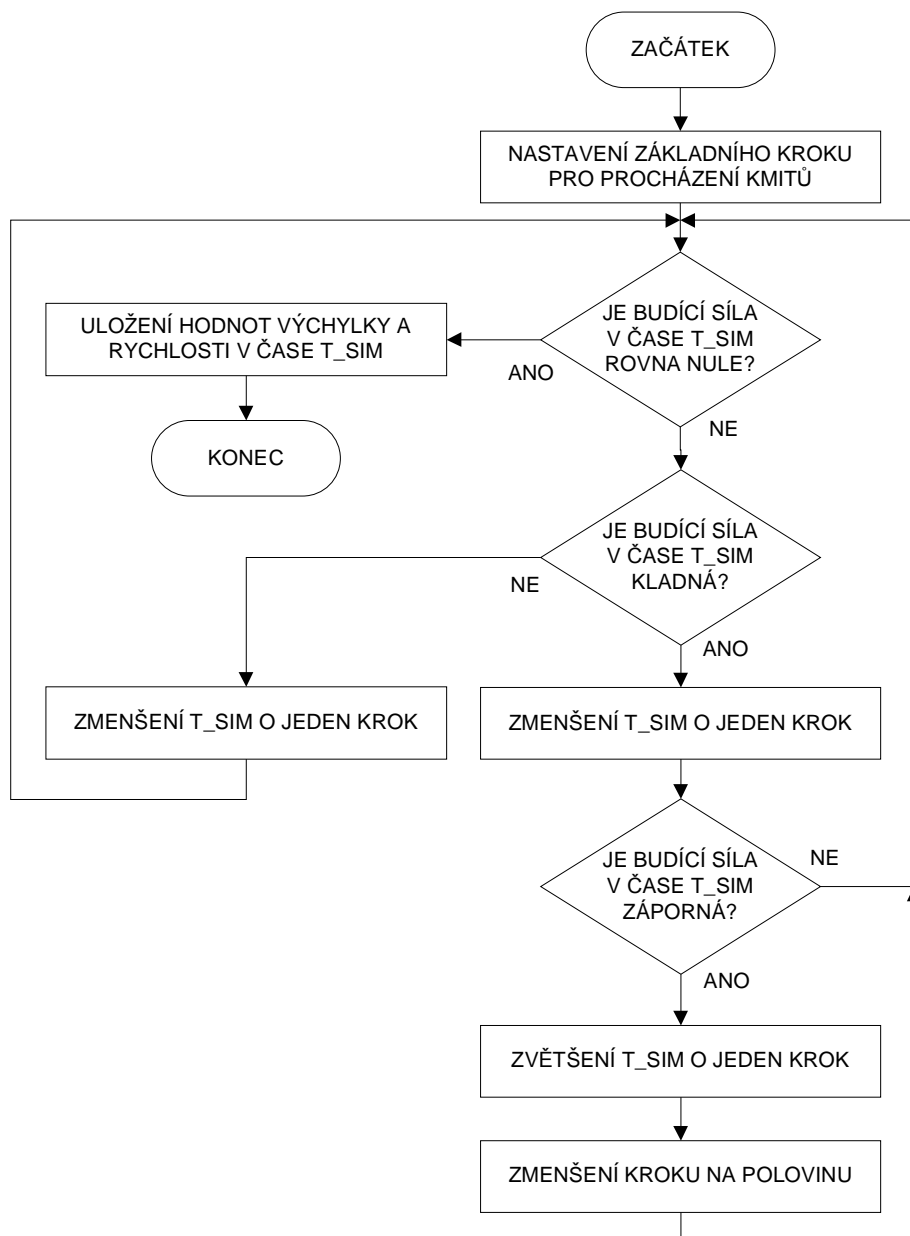
i := 1:
#nastavení příznaku opakování
mimo_toleranci := 1:
while mimo_toleranci = 1 do
    #definování základního kroku jako cca 0.1 násobek periody kmitu
    krok := 0.6/omega:
    #definování příznaku nalezení
    nalezeno := 0:
    #hledání kladné půlvlny
    t_sim := t_sim+krok:
    rychlost := op(2,reseni(t_sim)[3]):
    while rychlost < 0 do
        t_sim := t_sim+krok:
        rychlost := op(2,reseni(t_sim)[3])
    end do:
    #hledání konce kladné půlvlny pomocí postupného dělení kroku
    while nalezeno < 1 do
        while rychlost > 0 do
            t_sim := t_sim+krok:
            rychlost := op(2,reseni(t_sim)[3]):
        end do:
        #konec přeskočen, návrat o krok zpět
        t_sim := t_sim-krok:
        rychlost := op(2,reseni(t_sim)[3]):
        #kontrola zda je již rychlost v toleranci, nastavení příznaku
        if rychlost < tolerance_v then
            nalezeno := 1:
        end if:
        krok := krok/10:
    end do:
    #uložení nalezeného extrému
    EXTREM[i] := op(2,reseni(t_sim)[2]):
    i := i+1:
    #kontrola překročení délky vektoru
    if i > 10 then
        i := i-10:
    end if:
    #výpočet průměrné hodnoty výchylky
    prumer := 0:
    for i2 from 1 to 10 do
        prumer := prumer+EXTREM[i2]:
    end do:
    prumer := prumer/10:
    #kontrola zda jsou kmity v toleranci
    mimo_toleranci := 0:
    for i3 from 1 to 10 do
        if abs(EXTREM[i3]-prumer) > prumer*tolerance_x then
            mimo_toleranci := 1:
        end if:
    end do:
    #kontrola zda byl překročen maximální čas
    mc := max_t_sim-trunc(t_sim):
    if mc < 1 then
        mimo_toleranci := 0:
    end if:
end do:

```

Po vyšetření kmitů následuje uložení nalezených hodnot ustálené výchylky včetně příslušné úhlové frekvence budící síly  $F$  do příslušných vektorů.

```
#uložení nalezených hodnot do příslušných vektorů
AMPLITUDE[i4] := prumer:
OMEGA[i4] := omega:
```

Poslední část uvnitř hlavního cyklu programu řeší nalezení počátečních podmínek. Vývojový diagram této části je na obrázku 12.



Obr. 12.: Vývojový diagram části pro nalezení počátečních podmínek v prostředí MAPLE

Hledání počátečních podmínek je řešeno podobně jako hledání extrému postupným snižováním kroku a přibližováním k hodnotě, kde hodnota budící síly prochází nulou směrem ze záporných hodnot ke kladným v čase, kde je již výchylka kmitů soustavy ustálená. Pro tento čas se hodnoty výchylky a rychlosti uloží do patřičných proměnných.

```

#nalezení počátečních podmínek
krok := 0.6/omega:
while abs(evalf(sin(omega*t_sim))) > tolerance_v do
  if evalf(sin(omega*t_sim)) > 0 then
    t_sim := t_sim-krok:
    if evalf(sin(omega*t_sim)) < 0 then
      t_sim := t_sim+krok:
      krok := krok/2:
    end if:
  else
    t_sim := t_sim-krok:
  end if:
end do:
x_poc := op(2,reseni(t_sim)[2]):
v_poc := op(2,reseni(t_sim)[3]):

```

Na konci celého programu následuje stejně jako v MATLABu tisk grafu, včetně popisu.

```

plot(OMEGA,AMPLITUDA,title='Amplitudo-frekvenční charakteristika',
      labels=['omega [rad/s]','A [m]']);

```





## 5. ŘEŠENÉ PŘÍKLADY

### 5.1 Příklad č.1 - lineární systém

Uvedené programy byly nejprve otestovány na lineárním systému z důvodu možnosti porovnání výsledků s analytickým vyjádřením. Vzájemným porovnáním je možné získat představu o přesnosti popisované metody.

Předpokládejme příklad z obrázku 2 s následujícími parametry:

$$m = 5\text{kg}, \quad k = 300\text{N} \cdot \text{m}^{-1}, \quad b = 6\text{N} \cdot \text{s} \cdot \text{m}^{-1}, \quad F = 100\text{N}$$

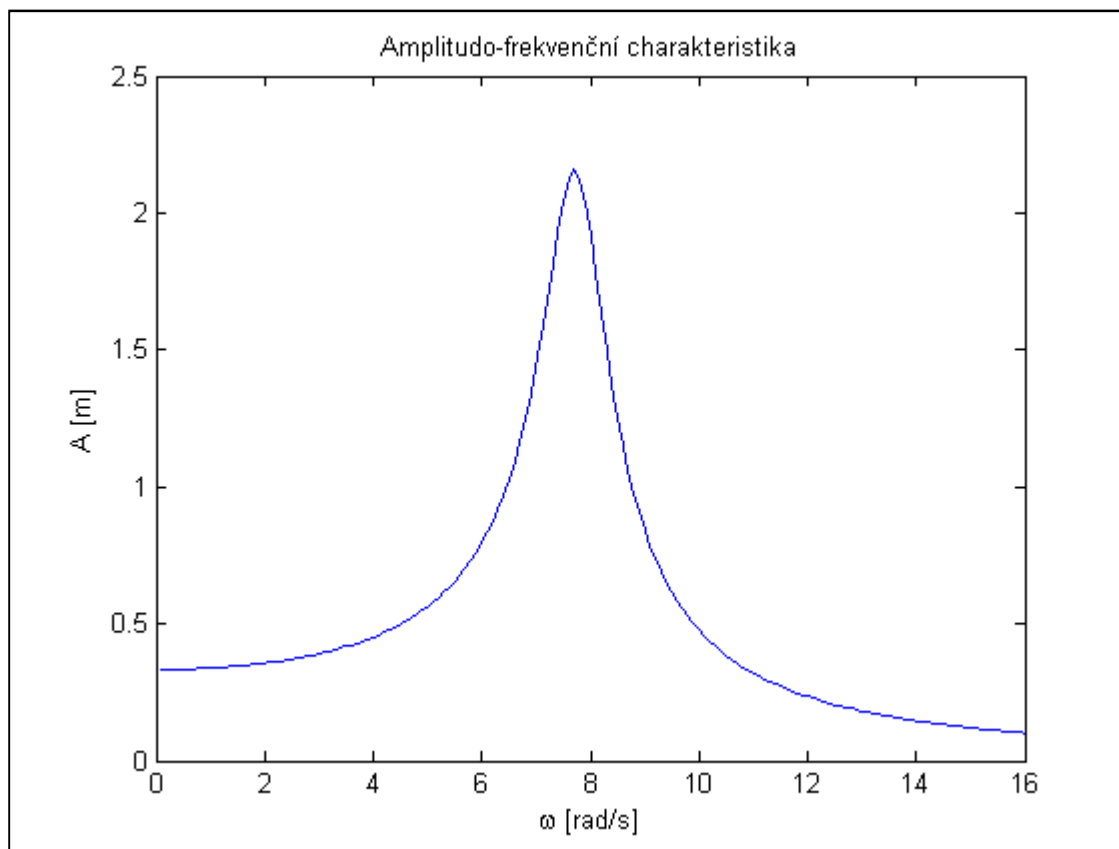
V případě lineárního systému můžeme ponechat pohybovou rovnici stejnou, jako pro systém s měknoucí, příp. tuhnoucí charakteristikou vratné síly pružiny, pouze parametr  $k$  položit roven nule. Nebo můžeme definici pohybové rovnice upravit. V prostředí MATLAB ji upravíme v souboru *dif\_rce\_oscil.m* podle skutečné pohybové rovnice systému do následujícího tvaru:

```
function dy = dif_rce_oscil(t,y,options,k,m,b,F,omega)
dy(1,1) = y(2,1);
dy(2,1) = -1/m*(b*y(2,1)+k*y(1,1)-F*sin(omega*t));
end
```

Zároveň je nutné vypustit parametr  $k$  z volání funkce ode45, které bude potom vypadat:

```
[T,Y] = ode45('dif_rce_oscil',tspan,y0,options,m,b,F,omega);
```

Výslednou amplitudo-frekvenční charakteristiku v systému MATLAB zobrazuje obrázek 13.

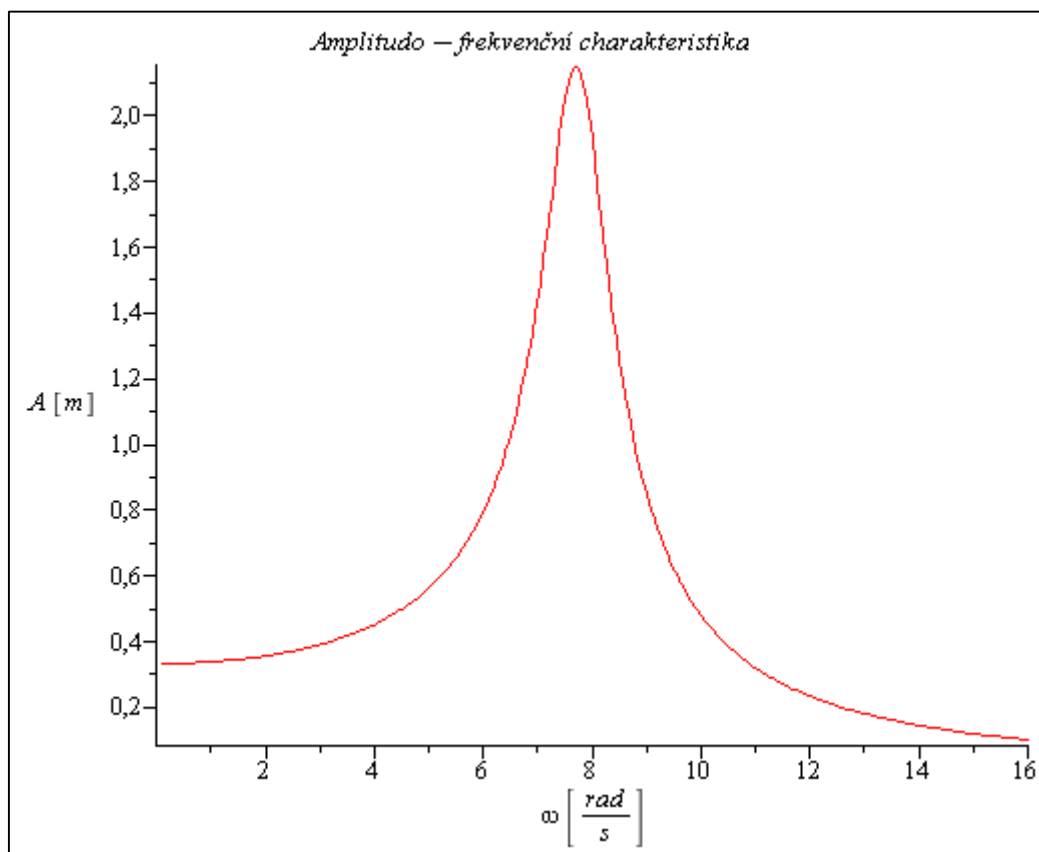


Obr. 13.: Amplitudo-frekvenční charakteristika lineárního systému v prostředí MATLAB

Upravená pohybová rovnice v systému MAPLE bude vypadat následovně:

```
rce_syst := diff(diff(x(t),t),t) = -1/m*(b*diff(x(t),t)+k*x(t)-
F*sin(omega*t));
```

Obrázek 14 potom zobrazuje výslednou amplitudo-frekvenční charakteristiku zkonstruovanou pomocí programu MAPLE.

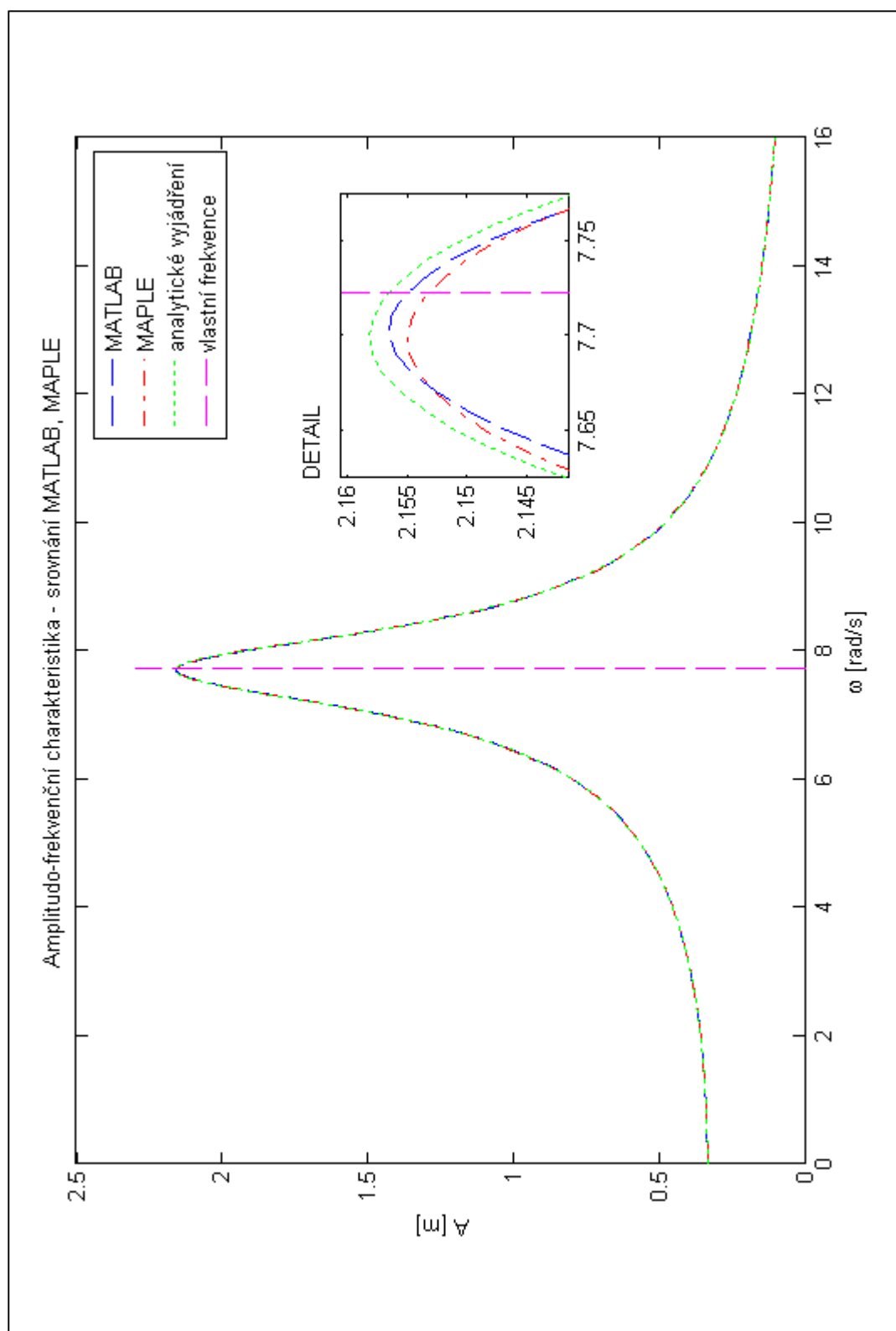


Obr. 14.: Amplitudo-frekvenční charakteristika lineárního systému v prostředí MAPLE

Analytické vyjádření v programu MATLAB vypadá následovně:

```
A_AN=0
O_AN=0
O_max=16
for i=1:1:100*O_max
    O_AN(i)=0.01*i;
    A_AN(i)=F/sqrt((k-m*O_AN(i)^2)^2+b^2*O_AN(i)^2);
end
```

Srovnání výsledků obou prostředí MATLAB i MAPLE společně s analytickým vyjádřením a vyznačením vlastní frekvence soustavy zachycuje obrázek 15. Z tohoto obrázku je patrný minimální rozdíl mezi použitými prostředím a relativně vysoká přesnost simulační metody. Absolutní chyba činí v případě prostředí MATLAB  $0,002m$  a v případě prostředí MAPLE  $0,003m$ , což v případě výchylky  $2,158m$  vede k relativní chybě  $0.09\%$  a  $0.14\%$ . Z obrázku je zároveň patrné, že k maximální výchylce skutečně nedochází přesně při hodnotě shodné s vlastní frekvencí soustavy, nýbrž při hodnotě blízké, o něco nižší, jak je zmíněno v kapitole 2.



Obr. 15.: Amplitudo-frekvenční charakteristika lineárního systému - srovnání MATLAB a MAPLE

## 5.2 Příklad č.2 - systém s měknoucí charakteristikou vratné síly pružiny

Předpokládejme příklad z obrázku 2 s následujícími parametry:

$$m = 5\text{kg}, \quad k = 300\text{N} \cdot \text{m}^{-1}, \quad k_3 = 25\text{N} \cdot \text{m}^{-3}, \quad b = 6\text{N} \cdot \text{s} \cdot \text{m}^{-1}, \quad F = 100\text{N}$$

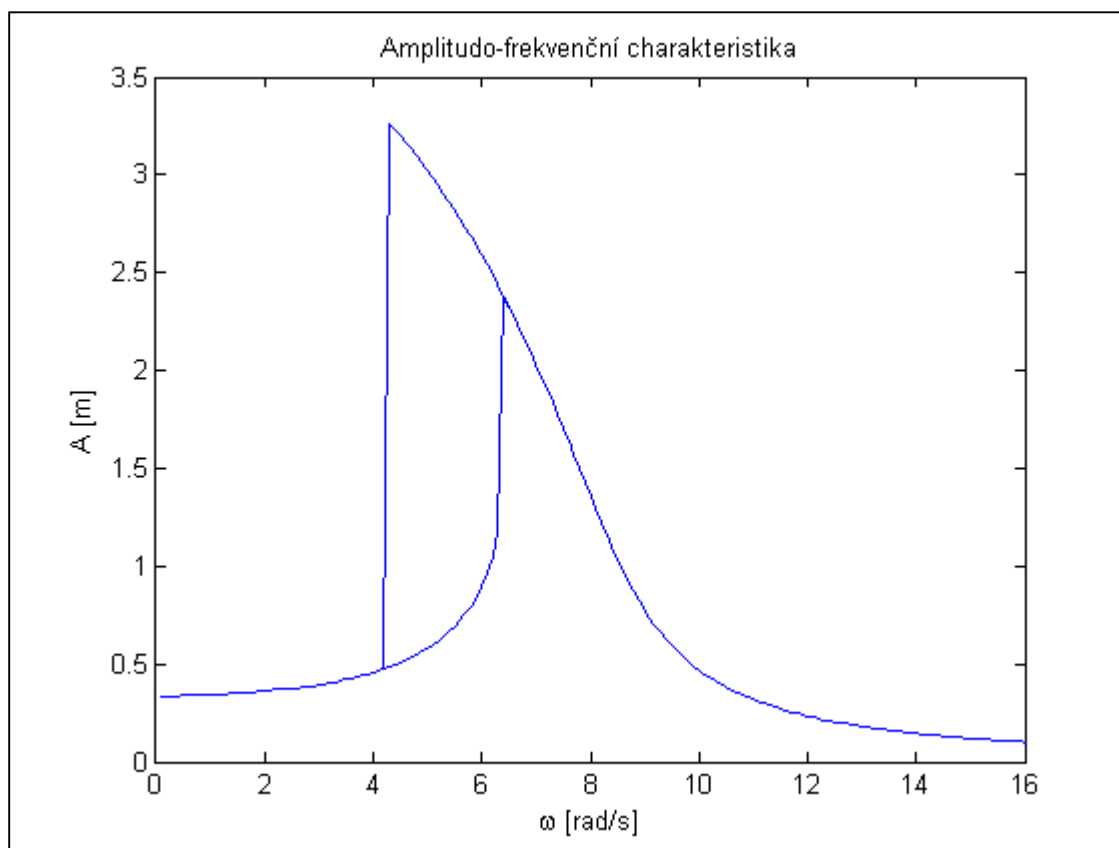
a vratnou silou pružiny definovanou jako:

$$F = k \cdot x - k_3 \cdot x^3$$

Pohybová rovnice v souboru dif\_rce\_oscil.m bude pro tento případ vypadat

```
function dy = dif_rce_oscil(t,y,options,k,k3,m,b,F,omega)
dy(1,1) = y(2,1);
dy(2,1) = -1/m*(b*y(2,1)+k*y(1,1)- k*y(1,1)^3-F*sin(omega*t));
end
```

Výslednou amplitudo-frekvenční charakteristiku v systému MATLAB zobrazuje obrázek 16.

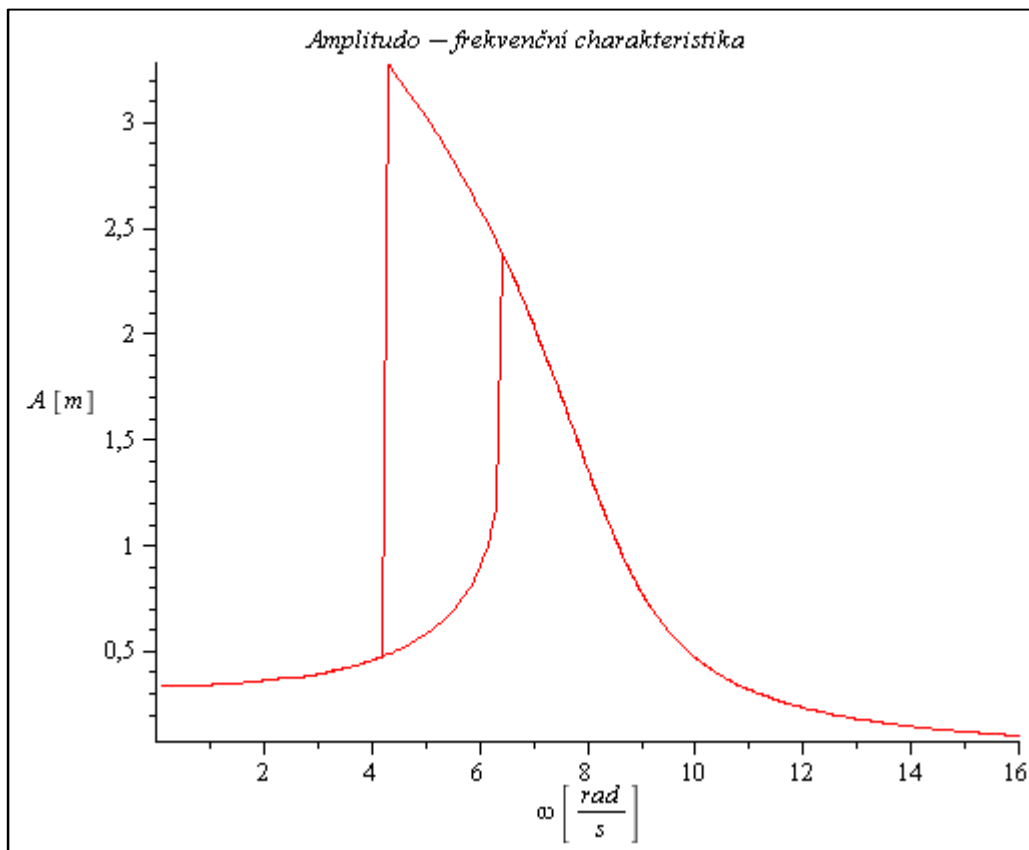


Obr. 16.: Amplitudo-frekvenční charakteristika systému s měknoucí pružinou v prostředí MATLAB

Upravená pohybová rovnice v systému MAPLE bude vypadat následovně:

```
rce_syst := diff(diff(x(t),t),t) = -1/m*(b*diff(x(t),t)+k*x(t)-k3*x(t)^3
-F*sin(omega*t));
```

Obrázek 17 potom zobrazuje výslednou amplitudo-frekvenční charakteristiku zkonstruovanou pomocí programu MAPLE.

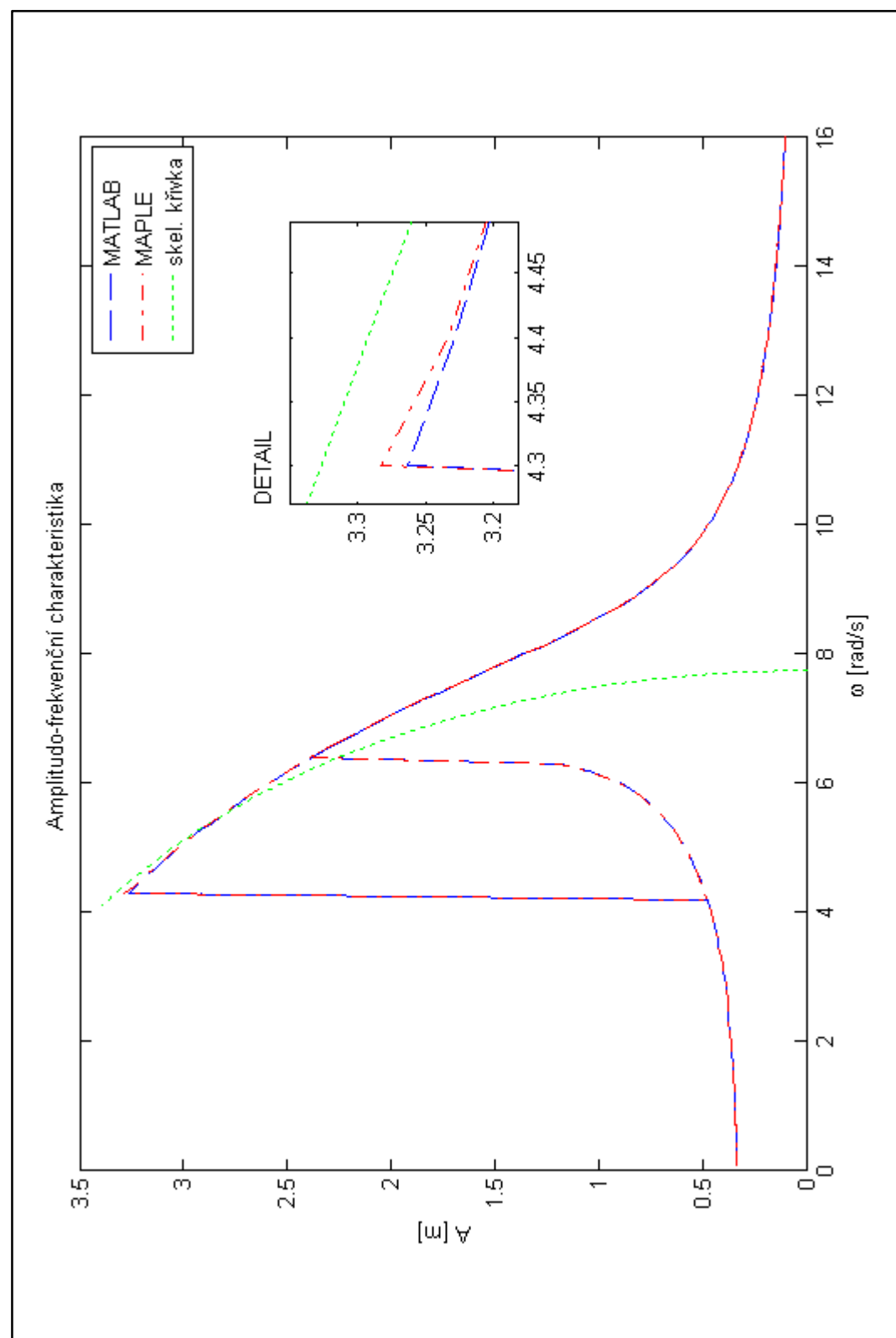


Obr. 17.: Amplitudo-frekvenční charakteristika systému s měknoucí pružinou v prostředí MAPLE

Srovnání výsledků obou prostředí MATLAB i MAPLE společně se skeletovou křivkou zachycuje obrázek 18. Výpočet skeletové křivky v programu MATLAB vypadá následovně:

```
A_SK=0
O_SK=0
A_max=3.4
for i=0:1:100*A_max
    A_SK(i+1)=0.01*i;
    O_SK(i+1)=sqrt(k/m-(3/4)*(k3/m)*A_SK(i+1)^2);
end
```

Z obrázku 18 je patrný opět minimální rozdíl mezi použitými prostředím, avšak značná odchylka od skeletové křivky. V případě prostředí MATLAB se jedná o absolutní chybu  $0,07m$  a v případě MAPLE  $0,05m$ , což v případě výchylky  $3,33m$  vede k relativní chybě  $2,1\%$  a  $1,5\%$ . Přesnost řešení tohoto typu nelineárního systému je tedy přibližně o řád menší. Chyba zřejmě vyplývá z toho, že se jedná o téměř krajní případ nelinearity, jelikož při vratné síle pružiny definované jako  $F = 300 \cdot x - 25 \cdot x^3$  se pro hodnotu maximální výchylky tato síla blíží nule.



Obr. 18.: Amplitudo-frekvenční charakteristika systému s měknoucí pružinou - srovnání MATLAB a MAPLE

### 5.3 Příklad č.3 - systém s tuhnoucí charakteristikou vratné síly pružiny

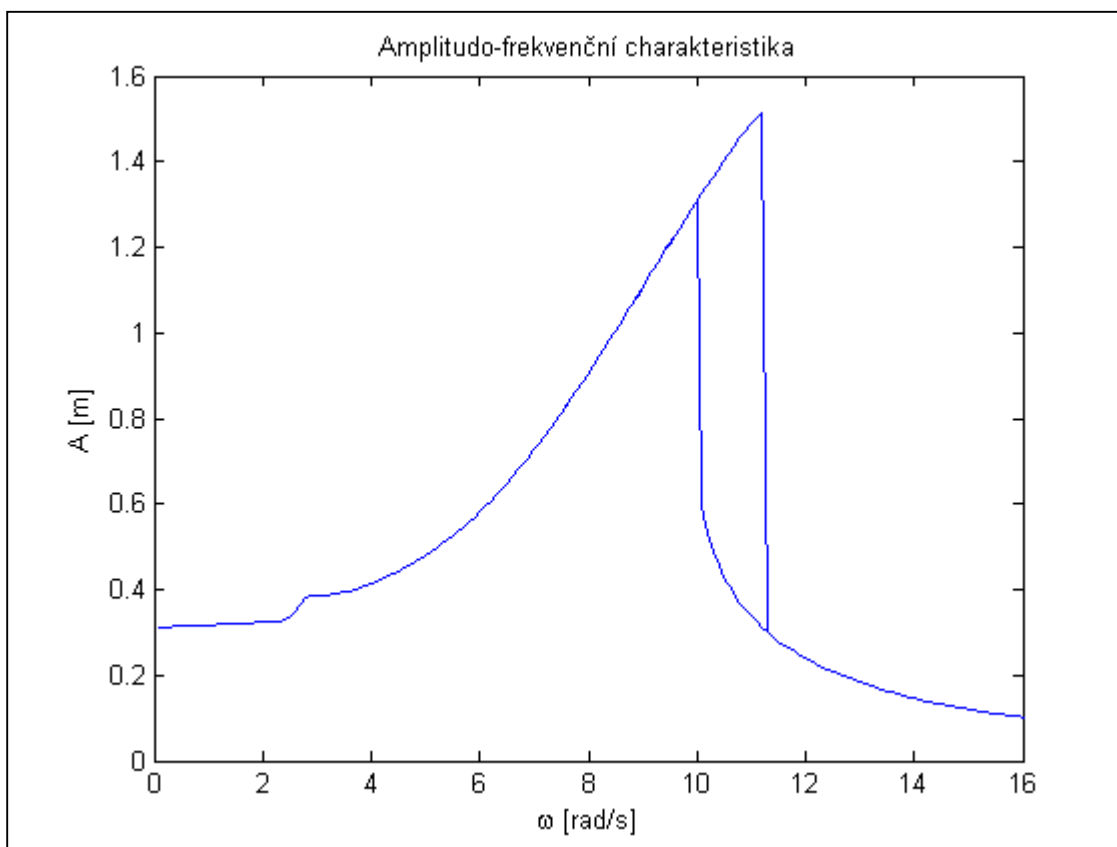
Předpokládejme příklad z obrázku 2 s následujícími parametry:

$$m = 5\text{kg}, \quad k = 300\text{N} \cdot \text{m}^{-1}, \quad k_3 = 200\text{N} \cdot \text{m}^{-3}, \quad b = 6\text{N} \cdot \text{s} \cdot \text{m}^{-1}, \quad F = 100\text{N}$$

a vratnou silou pružiny definovanou jako:

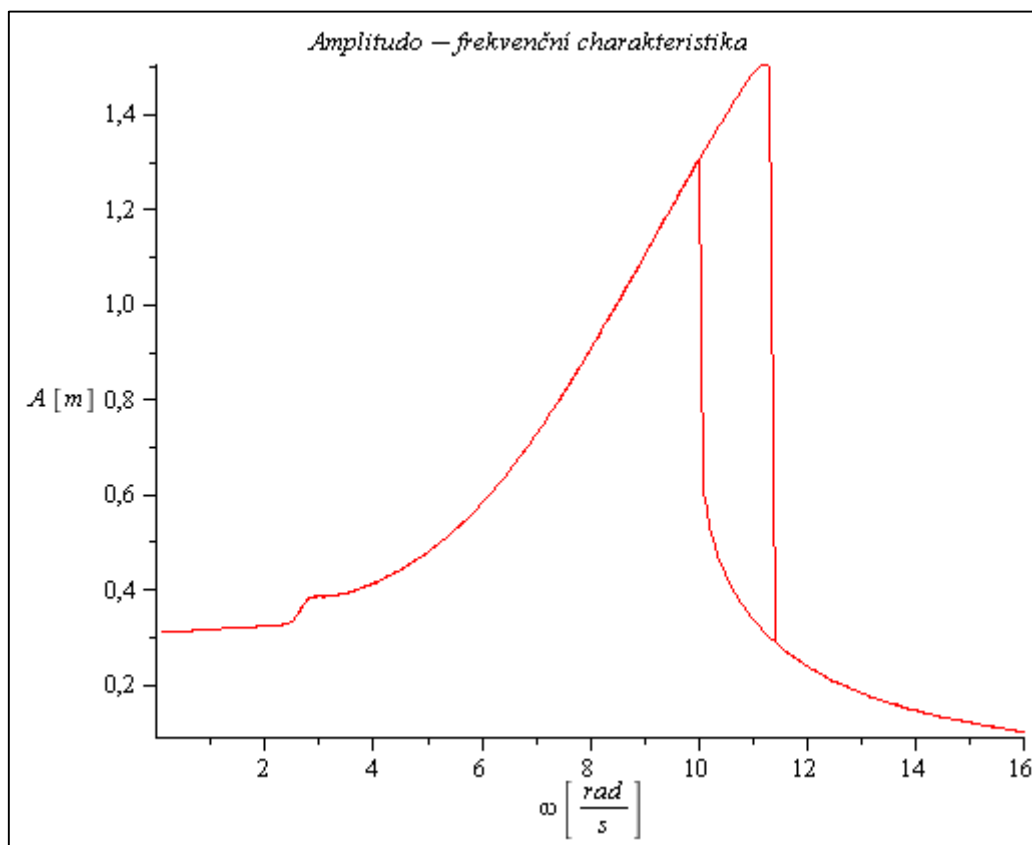
$$F = k \cdot x + k_3 \cdot x^3$$

Na tomto příkladu byla popisována implementace metody v obou softwarových prostředích. Na pohybových rovnicích se tedy pro tento případ nic nemění. Výslednou amplitudo-frekvenční charakteristiku v systému MATLAB zobrazuje obrázek 19.



Obr. 19.: Amplitudo-frekvenční charakteristika systému s tuhnoucí pružinou v prostředí MATLAB

Obrázek 20 potom zobrazuje výslednou amplitudo-frekvenční charakteristiku zkonstruovanou pomocí programu MAPLE.



Obr. 20.: Amplitudo-frekvenční charakteristika systému s tuhnoucí pružinou v prostředí MAPLE

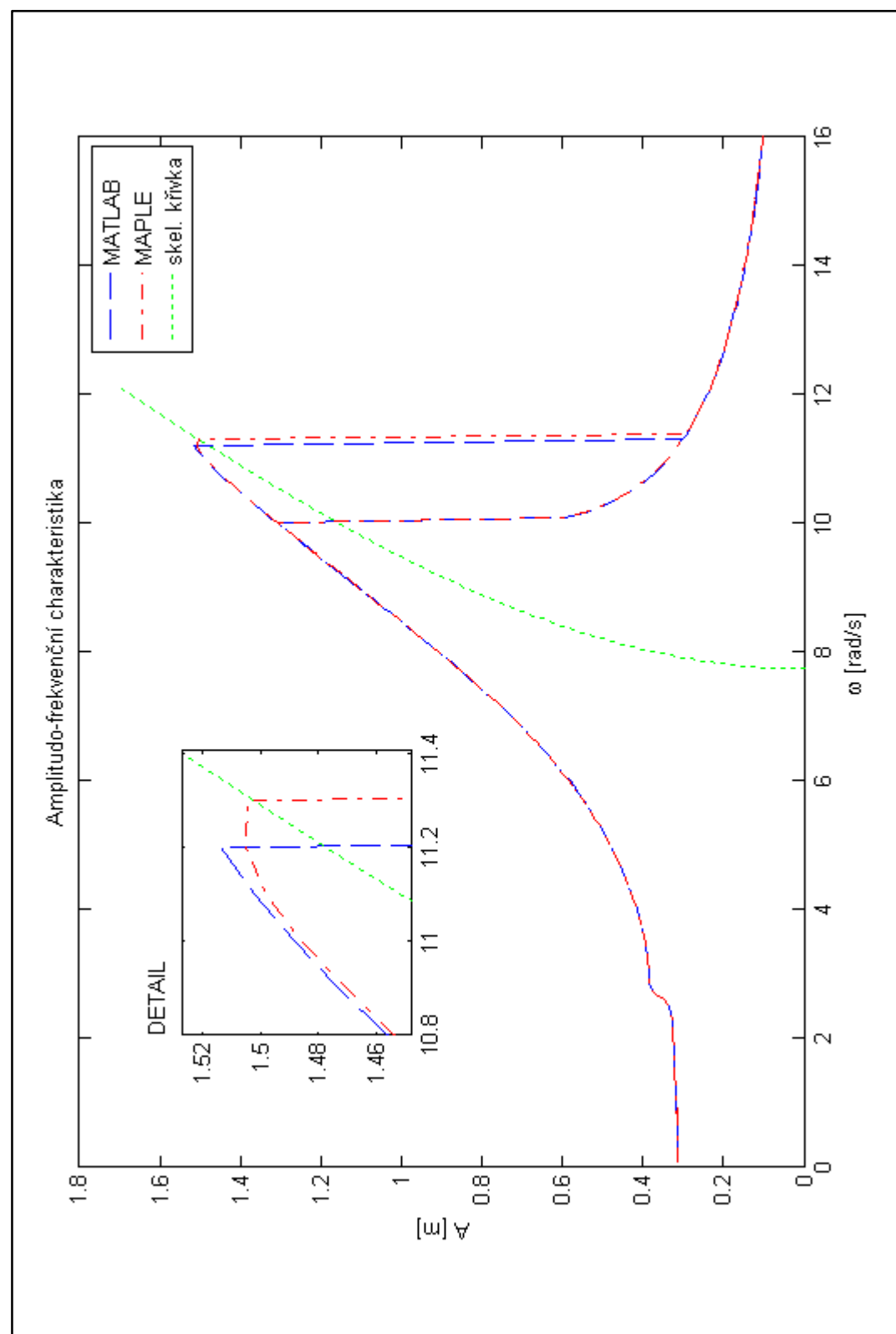
Srovnání výsledků obou prostředí MATLAB i MAPLE společně se skeletovou křivkou zachycuje obrázek 21. Výpočet skeletové křivky v programu MATLAB vypadá následovně:

```
A_SK=0
O_SK=0
A_max=3.4
for i=0:1:100*A_max
    A_SK(i+1)=0.01*i;
    O_SK(i+1)=sqrt(k/m+(3/4)*(k3/m)*A_SK(i+1)^2);
end
```

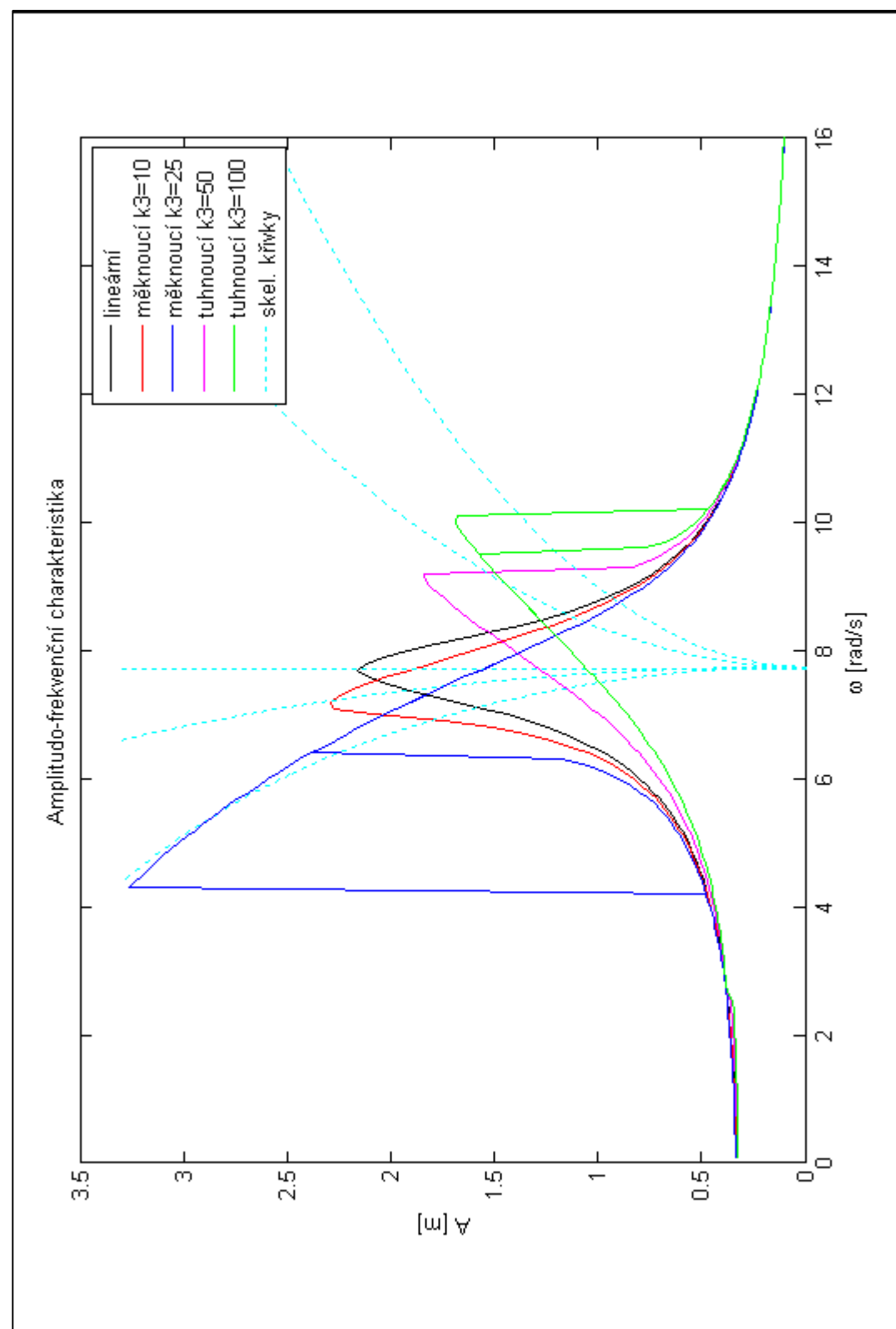
Na obrázku 21 je vidět, že v případě tuhnoucí charakteristiky systém MAPLE dokáže déle sledovat horní část amplitudo-frekvenční charakteristiky. V tomto případě je dokonce patrná velmi přesná konvergence obou výsledků ke skeletové křivce, systém MAPLE dokonce dovedl nasimulovat „prohnutí“ na samotném vrcholu amplitudo-frekvenční charakteristiky.

Srovnání vlivu nelinearity pro oba druhy nelinearity vratné síly pružiny znázorňuje obrázek 22, který porovnává shodné systémy s různými charakteristikami a velikostmi vratné síly pružiny.





Obr. 21.: Amplitudo-frekvenční charakteristika systému s tuhnoucí pružinou - srovnání MATLAB a MAPLE



Obr. 22.: Srovnání systémů s různými charakteristikami a velikostí vratné síly pružiny

#### 5.4 Příklad č.4 - systém s nelineárním tlumením

Předpokládejme příklad z obrázku 2 s následujícími parametry:

$$m = 5\text{kg}, \quad k = 300\text{N} \cdot \text{m}^{-1}, \quad b = 6\text{N} \cdot \text{s} \cdot \text{m}^{-1}, \quad F = 100\text{N}$$

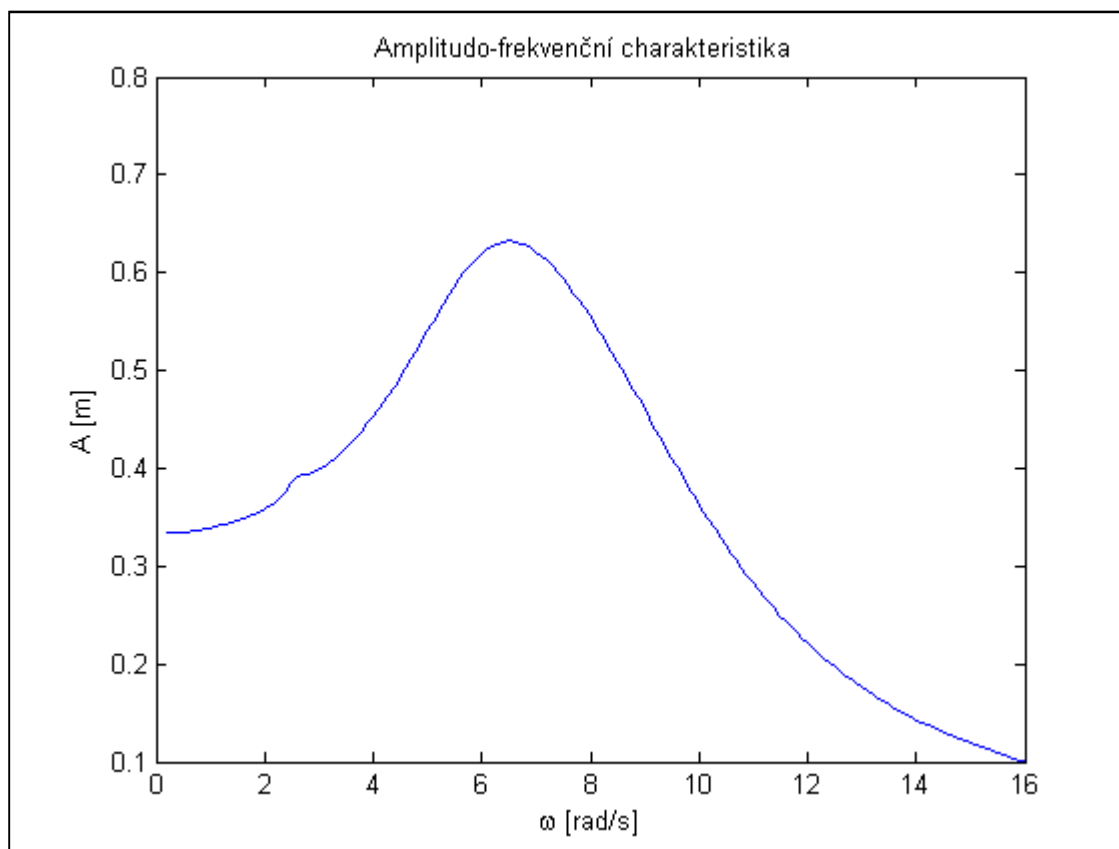
a hodnotou tlumení definovanou jako:

$$F = b \cdot v^2$$

Pohybová rovnice v souboru dif\_rce\_oscil.m bude pro tento případ vypadat

```
function dy = dif_rce_oscil(t,y,options,k,m,b,F,omega)
dy(1,1) = y(2,1);
dy(2,1) = -1/m*(b*y(2,1)*abs(y(2,1))+k*y(1,1)-F*sin(omega*t));
end
```

Stejně jako v případě lineárního systému není nutné v tomto případě předávat funkci ode45 hodnotu  $k3$ . Tato proměnná může být tedy z programu vypuštěna. Výslednou amplitudo-frekvenční charakteristiku v systému MATLAB zobrazuje obrázek 23.

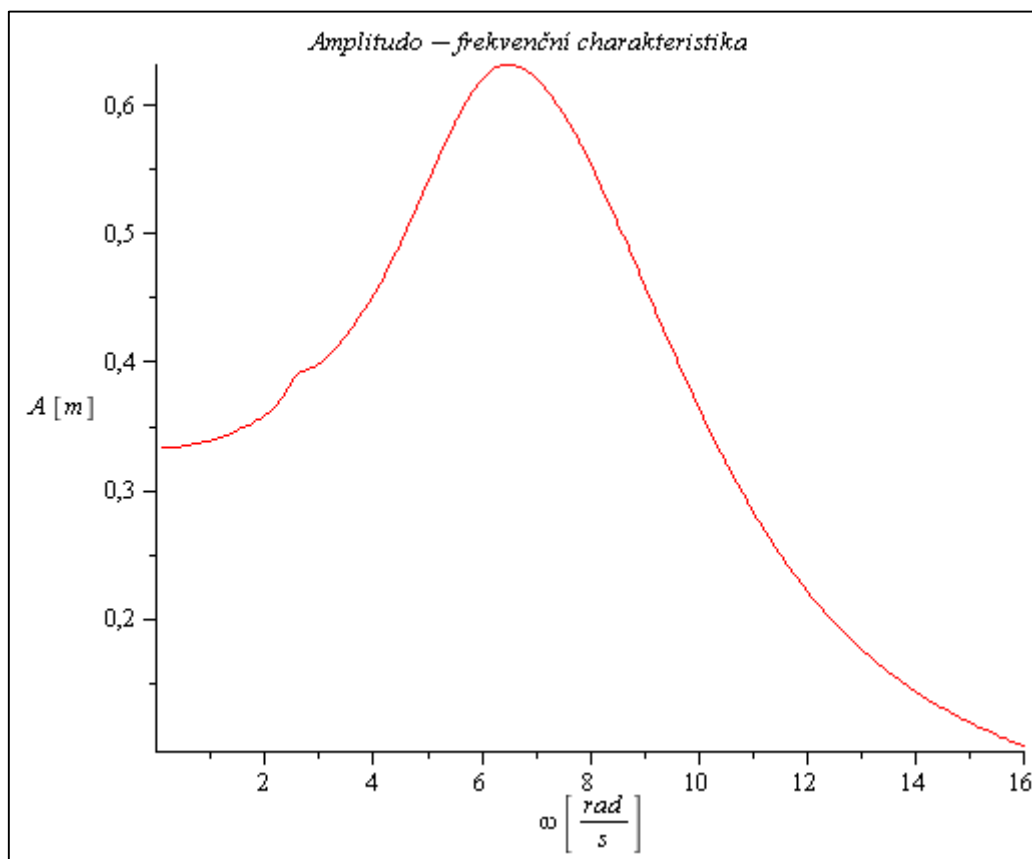


Obr. 23.: Amplitudo-frekvenční charakteristika systému s nelineárním tlumením v prostředí MATLAB

Upravená pohybová rovnice v systému MAPLE bude vypadat následovně:

```
rce_syst := diff(diff(x(t),t),t) = -1/m*(b*diff(x(t),t)*abs(diff(x(t),t))
+k*x(t)-F*sin(omega*t));
```

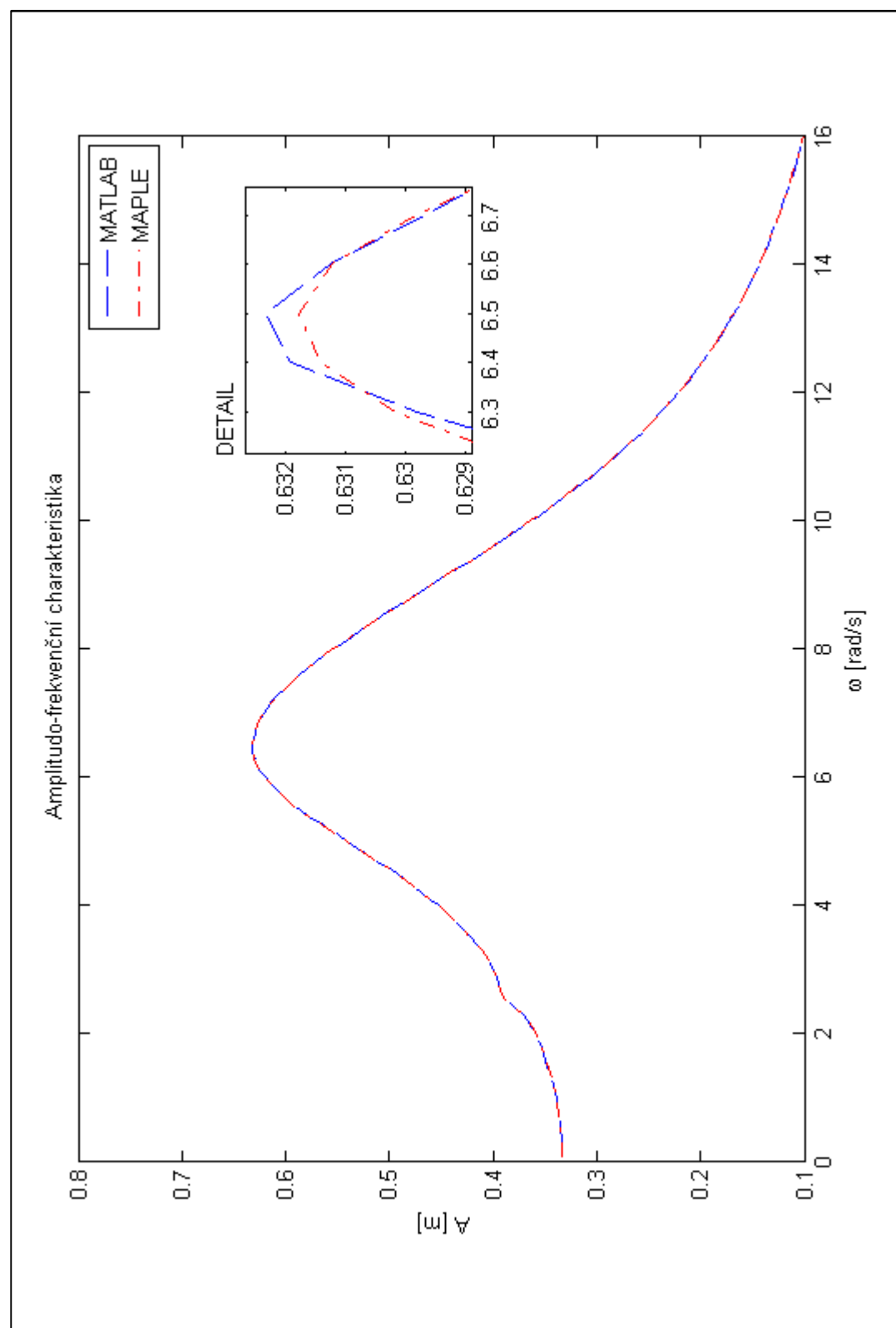
Obrázek 24 potom zobrazuje výslednou amplitudo-frekvenční charakteristiku zkonstruovanou pomocí programu MAPLE.



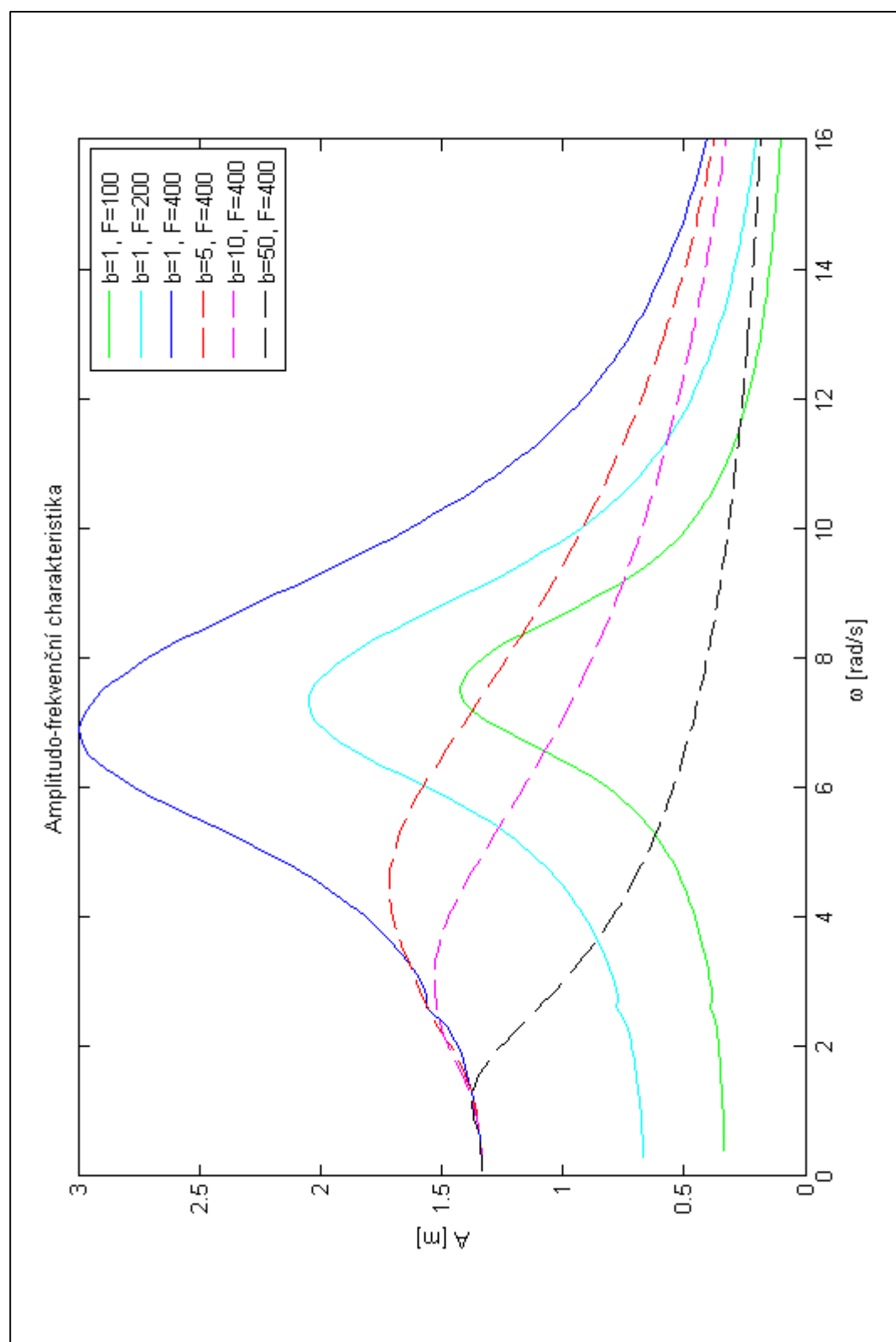
*Obr. 24.: Amplitudo-frekvenční charakteristika systému s nelineárním tlumením v prostředí MAPLE*

Srovnání výsledků obou prostředí MATLAB i MAPLE zachycuje obrázek 25. Ze srovnání je v tomto případě patrný opět minimální rozdíl mezi použitými prostředími.

Na obrázku 26 je potom možné pozorovat vliv nelinearity tlumení na vrchol amplitudo-frekvenční charakteristiky. Na tomto obrázku je porovnáno několik charakteristik pro různé velikosti parametru kvadratického tlumení  $b$ .



Obr. 25.: Amplitudo-frekvenční charakteristika systému s nelineárním tlumením - srovnání MATLAB a MAPLE



Obr. 26.: Srovnání systémů s různými velikostmi kvadratického tlumení

## 6. ZÁVĚR

Z uvedených výsledků je zřejmé, že zmíněná metoda implementovaná ve vhodném matematickém softwaru je schopna nelineární systémy řešit bez výrazných problémů. Určitě ji tedy lze použít pro demonstrační a výukové účely.

Pro tyto účely považuji za vhodnější využití prostředí MATLAB. Poskytuje totiž příjemnější uživatelské rozhraní a díky jeho orientaci na maticový počet i snazší práci s výsledky. Naproti tomu v prostředí MAPLE byla simulace přibližně dvojnásobně rychlejší. To však může být způsobeno trochu odlišnou konstrukcí programu. Jak je patrné z programů, MATLAB využívá pevnou dobu simulace, kdežto MAPLE proměnlivou. Délku běhu programu v prostředí MATLAB lze značně ovlivnit vhodným výběrem základní doby simulace. Pokud se doba simulace zvolí příliš dlouhá, řeší se pohybová rovnice „zbytečně“ pro příliš dlouhý úsek. Naopak, zvolí-li se příliš krátká a kmity nebudou po této době ještě ustálené, program bude muset dobu zdvojnásobit a řešit pohybovou rovnici opět od počátku. Tím pádem se tedy bude pohybová rovnice v určitém časovém úseku řešit dvakrát, v případě několikanásobného prodlužování doby simulace i vícekrát.

V případě implementace do nějaké grafické nadstavby matematického softwaru, např. Simulink v prostředí MATLAB, bychom obdrželi obdobné výsledky, protože tyto nadstavby využívají pro své řešení předdefinované funkce ze svého mateřského prostředí.





## 7. SEZNAM OBRÁZKŮ

Obr. 1. Mechanický oscilátor. . . . .	15
Obr. 2. Harmonicky buzený mechanický oscilátor. . . . .	17
Obr. 3. Amplitudo-frekvenční charakteristika lineárního systému. . . . .	18
Obr. 4. Amplitudo-frekvenční charakteristiky systémů s nelineární pružinou. . . . .	20
Obr. 5. Amplitudo-frekvenční charakteristiky - nelineární tlumení. . . . .	21
Obr. 6. Hledání ustálené výchylky kmitů. . . . .	24
Obr. 7. Nalezení maximální výchylky. . . . .	25
Obr. 8. Základní vývojový diagram. . . . .	27
Obr. 9. Vývojový diagram části pro vyšetření kmitů v prostředí MATLAB. . . . .	30
Obr. 10. Vývojový diag. části pro nalezení počátečních podmínek v prostředí MATLAB. . . . .	32
Obr. 11. Vývojový diagram části pro vyšetření kmitů v prostředí MAPLE. . . . .	36
Obr. 12. Vývojový diag. části pro nalezení počátečních podmínek v prostředí MAPLE. . . . .	38
Obr. 13. Amplitudo-frekvenční char. lineárního systému v prostředí MATLAB. . . . .	41
Obr. 14. Amplitudo-frekvenční char. lineárního systému v prostředí MAPLE. . . . .	42
Obr. 15. Amplitudo-frekvenční char. lineárního syst. - srovnání MATLAB a MAPLE. . . . .	43
Obr. 16. Amplitudo-frekvenční char. syst. s měknoucí pružinou v prostředí MATLAB . . . . .	44
Obr. 17. Amplitudo-frekvenční char. syst. s měknoucí pružinou v prostředí MAPLE. . . . .	45
Obr. 18. Amplitudo-frekvenční charakteristika systému s měknoucí pružinou - srovnání MATLAB a MAPLE. . . . .	46
Obr. 19. Amplitudo-frekvenční char. syst. s tuhnoucí pružinou v prostředí MATLAB. . . . .	47
Obr. 20. Amplitudo-frekvenční char. syst. s tuhnoucí pružinou v prostředí MAPLE. . . . .	48
Obr. 21. Amplitudo-frekvenční charakteristika systému s tuhnoucí pružinou - srovnání MATLAB a MAPLE. . . . .	49
Obr. 22. Srovnání systémů s různými charakteristikami a velikostmi vratné síly pružiny. . . . .	50
Obr. 23. Amplitudo-frekvenční char. syst. s nelineárním tlumením v prostředí MATLAB. . . . .	51
Obr. 24. Amplitudo-frekvenční char. syst. s nelineárním tlumením v prostředí MAPLE. . . . .	52
Obr. 21. Amplitudo-frekvenční charakteristika systému s nelineárním tlumením - srovnání MATLAB a MAPLE. . . . .	53
Obr. 26. Srovnání systémů s různými velikostmi kvadratického tlumení. . . . .	54



## 8. SEZNAM POUŽITÉ LITERATURY

- [1] ZEMÁNEK, T. *Aplikace matematických softwarů při řešení úloh kinematiky a dynamiky*. Brno: Vysoké učení technické v Brně, Fakulta strojního inženýrství, 2009. 41 s. Vedoucí bakalářské práce doc. RNDr. Karel Pellant, CSc.
- [2] JULIŠ, K., BREPTA, R. *MECHANIKA II. DÍL Dynamika*. Praha: SNTL - Nakladatelství technické literatury, 1987. 673 s.
- [3] *Mechanika těles – Dynamika* [online]. [cit. 2010-04-20]. Dostupný z WWW: [http://www.umat.fme.vutbr.cz/~pkrejci/opory/dynamika/kapitola\\_7.html](http://www.umat.fme.vutbr.cz/~pkrejci/opory/dynamika/kapitola_7.html).
- [4] *Nucené kmitání – Wikipedia* [online]. [cit. 2010-04-20]. Dostupný z WWW: [http://cs.wikipedia.org/wiki/Nucené\\_kmitání](http://cs.wikipedia.org/wiki/Nucené_kmitání).
- [5] SLAVÍK, J. *Mechanika III-Dynamika*. Brno: SNTL - Nakladatelství technické literatury, 1980. Skripta FS VUT v Brně.
- [6] PRETLOVÁ, V., ZAHRADNÍK, J. *Numerické metody v geofyzice*. Praha: Matematicko-fyzikální fakulta University Karlovy v Praze, 1976. 97 s. Postgraduální kurs zpracování geofyzikálních dat a číslicové seismiky.
- [7] HALLIDAY, D., RESNICK, R., WALKER J. *Fyzika*. Brno: Vysoké učení technické v Brně – Nakladatelství VUTIUM, Nakladatelství PROMETHEUS, 2000. 1254 s. ISBN: 80-214-1868-0
- [8] BUCHAR, J. *Úvod do programového souboru MAPLE V*. Brno: Vysoká škola zemědělská v Brně, 1994. 83 s. ISBN: 80-7157-117-2



## **9. SEZNAM PŘÍLOH**

Následující přílohy jsou pouze na CD, jedná se o soubory:

- Pucegl\_DP\_2010.pdf (vlastní text práce)
- program\_matlab.m (program v prostředí MATLAB)
- dif\_rce\_oscil.m (definice pohybové rovnice soustavy v prostředí MATLAB)
- program\_maple.mpl (program v prostředí MAPLE)