

# **Testování a dokumentace bloku DAE**

**Výzkumná zpráva projektu TA04010364, výsledek 2014V005**

Jana Königsmarková

# Obsah

<b>1</b>	<b>Simulace modelů systémů popsaných soustavou diferenciálních algebraických rovnic</b>	<b>3</b>
<b>2</b>	<b>Soustava diferenciálních algebraických rovnic</b>	<b>3</b>
2.1	Typy diferenciálních algebraických rovnic . . . . .	5
2.2	Index diferenciální algebraické soustavy . . . . .	6
2.3	Příklady systémů vedoucích na soustavu diferenciálních algebraických rovnic	7
<b>3</b>	<b>Numerické metody pro obyčejné diferenciální rovnice</b>	<b>8</b>
<b>4</b>	<b>Numerické metody pro řešení diferenciálních algebraických rovnic</b>	<b>9</b>
4.1	Metoda BDF pro soustavu DAE . . . . .	12
4.2	Metoda RADAUIIA pro soustavu DAE . . . . .	13
<b>5</b>	<b>Simulace a testování</b>	<b>14</b>
5.1	Simulace a testování metod na základních příkladech . . . . .	14
5.2	Rozbor simulace na příkladě descriptor systému . . . . .	16
5.3	Možnosti simulace . . . . .	20
5.4	Blok pro řešení soustavy diferenciálních algebraických rovnic . . . . .	22

# 1 Simulace modelů systémů popsaných soustavou diferenciálních algebraických rovnic

Při modelování dynamických vlastností fyzikálních systémů vznikají soustavy obyčejných diferenciálních rovnic (ODE - Ordinary Differential Equations), nebo ještě obecněji soustavy diferenciálních algebraických rovnic (DAE - Differential Algebraic Equations). Chceme-li vytvořené modely systémů využívat při návrhu řídicích systémů namísto reálného prototypu, potřebujeme umět řešit výslednou soustavu dostatečně rychle a s vhodnou přesností, aby bylo možné provádět simulaci v reálném čase a navrhnout tak algoritmus řízení, který bude možné obdobným způsobem využít pro reálný systém. Vytvoření vhodných numerických metod pro řešení soustav DAE je stále otevřeným zkoumaným problémem, kterému je v současné době věnována značná pozornost. Výsledná metoda pro řešení tohoto problému musí být navíc vhodná pro simulaci v reálném čase, musí být efektivní, dostatečně rychlá a rozumně přesná.

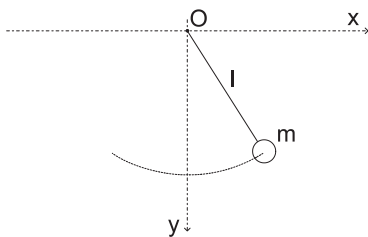
## 2 Soustava diferenciálních algebraických rovnic

Soustava diferenciálních algebraických rovnic (DAE) spojuje v rovnicích dohromady jednak některé z neznámých funkcí i s jejich derivacemi a jednak také neznámé funkce, jejichž derivace se v soustavě nevyskytují. Tyto neznámé funkce se zde mohou vyskytovat například následkem nějaké vazební vedlejší podmínky a nemusí pro ně ani v některých bodech existovat derivace. V uvedených soustavách tedy rozlišujeme diferenciální proměnné a algebraické proměnné. Pokud by systém algebraické neznámé neobsahoval, získali bychom soustavu obyčejných diferenciálních rovnic (ODE). Algebraickou část můžeme získat například z různých vedlejších podmínek, které vzájemně svazují diferenciální a algebraické proměnné. Systém diferenciálních algebraických rovnic lze popsat obecně v tzv. plně implicitním tvaru

$$F(t, y, \dot{y}) = 0, \quad (1)$$

kde Jacobiho matice  $\frac{\partial F}{\partial \dot{y}}$  představuje obecně singulární matici (pro obyčejnou diferenciální soustavu by muselo být  $\frac{\partial F}{\partial \dot{y}}$  regulární). Zmíněnou singularitou tak kromě diferenciálních proměnných povolujeme i algebraickou část.

Jedním ze základních modelů vedoucím na systém diferenciálních algebraických rovnic je model jednoduchého kyvadla (obr. 1). Do Lagrangeova vztahu



Obrázek 1: Jednoduché kyvadlo.

$$L(\dot{q}, q) = T - V - \lambda g \quad (2)$$

kde  $q$  představují zobecněné souřadnice,  $\lambda$  Lagrangeův multiplikátor,  $T$  kinetickou energii,  $V$  potenciální energii, dosadíme kinetickou energii ve tvaru

$$T = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2),$$

potenciální energii

$$V = mgl - mgy$$

a vazební podmínku, která nám bude svazovat pevně souřadnice bodu  $m$  vzhledem k délce závěsu  $l$

$$x^2 + y^2 = l^2, \quad (3)$$

Potom pro

$$\frac{d}{dt} \frac{\partial L}{\partial \dot{q}_k} - \frac{\partial L}{\partial q_k} = Q_k, \quad (4)$$

kde  $q_1 = x$ ,  $q_2 = y$ ,  $q_3 = \lambda$  a nulové zobecněné síly  $Q_k$ , dostaneme postupně

$$\begin{aligned} m\ddot{x} + 2\lambda x &= 0 \\ m\ddot{y} + 2\lambda y - mg &= 0 \\ x^2 + y^2 - l^2 &= 0 \end{aligned}$$

a po úpravách

$$\begin{aligned} \ddot{x} &= -\frac{2}{m}\lambda x \\ \ddot{y} &= -\frac{2}{m}\lambda y + g \\ 0 &= x^2 + y^2 - l^2. \end{aligned}$$

Nyní můžeme vyjádřit uvedenou soustavu jako soustavu diferenciálních algebraických rovnic prvního řádu, využijeme označení  $x_1 = x$ ,  $x_2 = \dot{x}$ ,  $x_3 = y$ ,  $x_4 = \dot{y}$  a získáme

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{2}{m}\lambda x_1 \\ \dot{x}_3 &= x_4 \\ \dot{x}_4 &= -\frac{2}{m}\lambda x_3 + g \\ 0 &= x_1^2 + x_3^2 - l^2. \end{aligned} \quad (5)$$

Poslední rovnice soustavy (5) představuje vazební podmínku, která reprezentuje algebraickou rovnici v daném systému. Pokud bychom řešili soustavu bez této vazební podmínky, nepohyboval by se bod  $m$  po předpokládané trajektorii, ale postupně by se jeho vzdálenost od pevného závěsu zvětšovala.

## 2.1 Typy diferenciálních algebraických rovnic

Diferenciální a algebraická část mohou být v soustavě propojeny nebo oddělené, může se dále jednat o lineární nebo nelineární rovnice, s koeficienty závislými na čase nebo nezávislými. Podle toho lze rozlišovat některé speciální typy diferenciálních algebraických soustav.

- **Plně implicitní** systém diferenciálních algebraických nelineárních rovnic

$$F(t, y, \dot{y}) = 0, \quad (6)$$

kdy  $F : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$  je spojitě diferencovatelná funkce pro  $y$ .

- **Semi-implicitní** systém diferenciálních algebraických nelineárních rovnic

$$\begin{aligned} F(t, y, \dot{y}) &= 0, \\ g(t, y, \dot{y}) &= 0, \end{aligned} \quad (7)$$

kde Jacobián  $\frac{\partial F}{\partial \dot{y}}$  je nesingulární matice.

- **Semi-explicitní** systém diferenciálních algebraických nelineárních rovnic

$$\begin{aligned} \dot{y} &= f(t, y, z), \\ 0 &= g(t, y, z). \end{aligned} \quad (8)$$

- Systém nelineárních DAE v **Hessenbergově tvaru** (index 1)

$$\begin{aligned} \dot{y} &= f(t, y, z), \\ 0 &= g(t, y, z), \end{aligned} \quad (9)$$

pro nesingulární  $g_z$  pro všechna  $t$ .

- Systém nelineárních DAE rovnic v **Hessenbergově tvaru** (index 2)

$$\begin{aligned} \dot{y} &= f(y, z), \\ 0 &= g(y), \end{aligned} \quad (10)$$

pro nesingulární  $g_y f_z$  pro všechna  $t$ .

- Systém nelineárních DAE rovnic v **Hessenbergově tvaru** (index 3)

$$\begin{aligned} \dot{x} &= f(t, x, y, z), \\ \dot{y} &= g(x, y), \\ 0 &= h(y), \end{aligned} \quad (11)$$

pro nesingulární  $h_y g_x f_z$  pro všechna  $t$ .

- **Lineární** soustava DAE s časově **proměnnými** koeficienty

$$A(t)\dot{y}(t) + B(t)y(t) = f(t), \quad (12)$$

kde matice  $A(t)$  je obecně singulární.

- **Lineární** soustava DAE s **konstantními** koeficienty

$$A\dot{y}(t) + By(t) = f(t), \quad (13)$$

kde matice  $A$  je obecně singulární.

## 2.2 Index diferenciální algebraické soustavy

Významným znakem charakterizujícím soustavu DAE rovnic je **index** DAE soustavy. Existuje několik druhů indexů popisujících jisté vlastnosti DAE systému, ale uvedený index soustavy DAE, nazývaný též diferenciální index, je jedním ze základních.

**Definice 1.** Diferenciální index je minimální číslo, kolikrát je nutné zderivovat podle času soustavu (6) nebo jen některé její rovnice, abychom určili  $\dot{y}$  jako spojitou funkci v čase a  $y$ .

Index DAE soustavy tedy vlastně také říká, zda a jak obtížně je soustava DAE rovnic převeditelná na ODE. Existují soustavy, které nelze na soustavu ODE vůbec převést. Navíc, derivování algebraických proměnných nemusí mít fyzikální opodstatnění a operace derivování představuje špatně podmíněnou operaci. Dále také algebraické neznámé nemusí mít derivaci definovanou v každém bodě. Z těchto důvodů bývá lepší řešit systém DAE rovnic s indexem 1, než převádět soustavu na ODE. Pro tyto DAE již existují vhodné numerické metody, které vedou na rychlejší a efektivnější výpočet, než kdybychom převáděli a řešili systém jako soustavu obyčejných diferenciálních rovnic.

Diferenciální index také udává, jak je daný systém složitý na řešení, čím je vyšší, je obtížnější danou soustavu řešit. Soustava obyčejných diferenciálních rovnic má tedy index 0. Pro obyčejné diferenciální soustavy existuje mnoho numerických metod, zatímco diferenciální algebraické soustavy s vyšším indexem jsou hůře řešitelné.

Například diferenciální index jednoduchého kyvadla (5) je 3. Budeme-li postupně derivovat algebraickou rovnici  $x_1^2 + x_3^2 - l^2 = 0$ , budeme získávat

$$\begin{aligned} 2x_1\dot{x}_1 + 2x_3\dot{x}_3 &= 0, & (1.\text{der.}) \\ x_1x_2 + x_3x_4 &= 0, \end{aligned}$$

$$\begin{aligned} \dot{x}_1x_2 + x_1\dot{x}_2 + \dot{x}_3x_4 + x_3\dot{x}_4 &= 0, & (2.\text{der.}) \\ x_2^2 + x_1\left(-\frac{2}{m}\lambda x_1\right) + x_4^2 + x_3\left(-\frac{2}{m}\lambda x_3 + g\right) &= 0, \\ x_2^2 - \frac{2}{m}\lambda(x_1^2 + x_3^2) + x_4^2 + gx_3 &= 0, \\ x_2^2 - \frac{2}{m}\lambda l^2 + x_4^2 + gx_3 &= 0, \end{aligned}$$

$$\begin{aligned} 2x_2\dot{x}_2 - \frac{2}{m}\dot{\lambda}l^2 + 2x_4\dot{x}_4 + g\dot{x}_3 &= 0, & (3.\text{ der.}) \\ -\frac{4}{m}\lambda x_1x_2 - \frac{2}{m}l^2\dot{\lambda} - \frac{4}{m}\lambda x_3x_4 + 3gx_4 &= 0. \end{aligned}$$

Protože v poslední rovnici je již koeficient u  $\dot{\lambda}$  nenulový, můžeme vyjádřit soustavu jed-

noduchého kyvadla jako soustavu obyčejných diferenciálních rovnic.

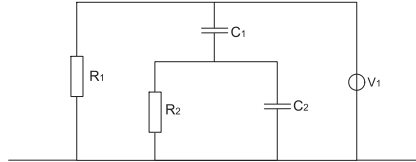
$$\begin{aligned}
\dot{x}_1 &= x_2 \\
\dot{x}_2 &= -\frac{2}{m}\lambda x_1 \\
\dot{x}_3 &= x_4 \\
\dot{x}_4 &= -\frac{2}{m}\lambda x_3 + g \\
\dot{\lambda} &= \frac{1}{2l^2}(-4\lambda x_1 x_2 - 4\lambda x_3 x_4 + 3mgx_4).
\end{aligned} \tag{14}$$

**Poznámka 1.** Pro převod systému DAE na ODE lze postupně derivovat rovnice, dokud se nám nepovede buď dostatečně snížit index soustavy DAE nebo převést je přímo na ODE. Pro některé systémy lze také využít různé symbolické algoritmy, jako např. Pantelideův algoritmus [4].

## 2.3 Příklady systémů vedoucích na soustavu diferenciálních algebraických rovnic

Na systémy DAE vedou problémy z mechaniky vícehmotových pohyblivých těles, mechatroniky, elektrických obvodů, optimálního řízení, chemické kinetiky, či například molekulární dynamiky. Jedním z takových systémů je model jednoduchého kyvadla zmíněný v odstavci 2.

Dalším příkladem může být *návrh elektrických obvodů*, viz. [1, 3]. Na základě Kirchhoffových zákonů pro napětí a proud, společně s konstitutivními vztahy se odvodí soustava diferenciálních algebraických rovnic. Uvedeme si příklad vedoucí na lineární soustavu indexu 2. Pro obvod na obrázku 2 dostaneme



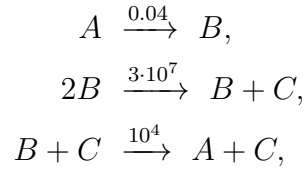
Obrázek 2: Elektrický obvod.

$$\begin{bmatrix} -1 & 1 & 0 \\ 1 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{j}_V \end{bmatrix} + \begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ j_V \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -v(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{15}$$

V práci [1] se také zabývají vyjádřením téhož obvodu s virtuálními prvky, aby do soustavy zahrnuli i skryté vazby. Řešení takového systému bude ekvivalentní s původním a nově odvozená soustava bude mít index 1. Získáme následující rovnice

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & -2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{e}_1 \\ \dot{e}_2 \\ \dot{j}_V \end{bmatrix} + \begin{bmatrix} -1 & 0 & -1 \\ 0 & -1 & 0 \\ -1 & 0 & 0 \end{bmatrix} \begin{bmatrix} e_1 \\ e_2 \\ j_V \end{bmatrix} + \begin{bmatrix} \dot{v}(t) \\ -\dot{v}(t) \\ -v(t) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \tag{16}$$

Příkladem z oblasti chemických procesů je tzv. *Robertsonův problém* [7], který modeluje kvantitativní úbytek a vznik zúčastněných chemických reaktantů. Robertsonův příklad popisuje kinetiku autokatalytické reakce s reaktanty  $A, B, C$



kde uvedené konstanty charakterizují rychlost reakce. Následkem značně rozdílných konstant vede tento problém na docela stiff soustavu, kterou můžeme vyjádřit buď jako soustavu obyčejných diferenciálních rovnic nebo jako soustavu diferenciálních algebraických rovnic v Hessenbergově tvaru s indexem 1 (9) pro neznámé funkce koncentrací reaktantů  $A, B$  a  $C$ .

$$\begin{aligned} \dot{y}_A &= -0.04y_A + 10^4y_By_C, \\ \dot{y}_B &= 0.04y_A - 10^4y_By_C - 3 \cdot 10^7y_B^2, \\ 1 &= y_A + y_B + y_C, \end{aligned} \tag{17}$$

$$\begin{aligned} \dot{y}_A &= -0.04y_A + 10^4y_By_C, \\ \dot{y}_B &= 0.04y_A - 10^4y_By_C - 3 \cdot 10^7y_B^2, \\ \dot{y}_C &= 3 \cdot 10^7y_B^2. \end{aligned} \tag{18}$$

### 3 Numerické metody pro obyčejné diferenciální rovnice

Pro řešení obyčejných diferenciálních rovnic existuje několik vhodných numerických metod. Tyto metody můžeme rozlišovat například podle způsobu jejich odvození, podle složitosti, přesnosti nebo také stability. Rozlišujeme tak explicitní, či implicitní metody, jednokrokové či vícekové metody, metody prvního, druhého apod. řádu přesnosti. Podle vlastností daného systému a požadavků na efektivitu výpočtu si vybereme nějakou vhodnou numerickou metodu. Implicitní metody je možné oproti explicitním metodám s jednodušším předpisem aplikovat na tzv. stiff (tuhé) systémy. Stiff systémem se rozumí systém, který má velké rozpětí mezi svými časovými konstantami. Mezi vlastními čísly příslušného linearizovaného systému je velká vzdálenost na reálné ose.

Uvažujme základní, obecně nelineární, obyčejnou diferenciální rovnici prvního řádu

$$\dot{y}(t) = f(t, y(t)), \tag{19}$$

kde  $f : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R}$ ,  $y : \mathbb{R} \rightarrow \mathbb{R}$  je neznámá funkce definovaná na  $I \subset \mathbb{R}$ ,  $t \in I$ .



Implicitní metody jsou takové metody, v jejichž schéma závisí funkce  $f$  na hledané hodnotě v novém řešeném čase,  $(y_{n+1})$ . U těchto metod je důležitá oblast absolutní stability. Oblastí absolutní stability nazýváme oblast v komplexní polorovině, která splňuje jistý požadavek pro testovací rovnici. Touto rovnicí je základní obyčejná diferenciální rovnice prvního řádu  $\dot{y} = \lambda y$ , pro  $t > 0$  a  $y(t_0) = y_0$  a  $\operatorname{Re}(\lambda) < 0$ , u níž známe řešení  $y(t) = e^{\lambda t} y_0$  a víme, že splňuje podmínku  $|y(t)| \leq |y(0)|$ . Pro numerické řešení bychom chtěli docílit stejného chování a tedy požadujeme, aby platilo  $|y_{n+1}| \leq |y_n|$ . Pokud numerické schéma vyjádříme ve tvaru  $y_{n+1} = R(z)y_n$ , kde  $R(z)$  je pro explicitní metody polynomiální funkce, pro implicitní racionální lomená funkce, a pro  $z = \lambda h$ , dostaneme požadavek, aby  $|R(z)| \leq 1$ . Oblast v komplexní rovině, která toto splňuje, nazýváme oblastí absolutní stability. Metody, pro které oblast absolutní stability inciduje s celou zápornou polorovinou komplexní roviny, nazýváme A-stabilní.

Implicitní metodou s velmi jednoduchým předpisem je implicitní Eulerova metoda. Jedná se o metodu prvního řádu, jejíž tvar je dán vztahem

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}).$$

Zajímavými metodami vhodnými pro stiff systémy jsou víceukrokové metody založené na aproximaci neznámé funkce polynomem daného stupně procházejícím několika předchozími body diskretizace. Metoda  $k$ -ukroková používá pro výpočet hodnoty v novém časovém bodě  $y_{n+1}$  předchozí vypočtené hodnoty  $y_n, y_{n-1}, \dots, y_{n+1-k}$ . Díky využití těchto hodnot se zvyšuje přesnost aproximace. Jedná se o tzv. BDF (Backward Differentiation Formula) metody. Metoda  $k$ -ukroková vyžaduje tedy již i pro začátek výpočtu znalost  $k$  hodnot (počáteční podmínku  $y_0$  a dalších  $k - 1$  hodnot). Tyto hodnoty lze určit například jednokrokovými metodami, nebo víceukrokovými metodami avšak nižšího řádu (neboť ty vyžadují znalost méně předpočtených hodnot). Přesnost výpočtu hodnoty  $y_{n+1}$  u víceukrokových metod je tak ovlivněna přesností počátečních  $k$  hodnot. Příkladem víceukrokové metody druhého řádu je metoda, která má pro rovnici (19) předpis

$$y_{n+1} = \frac{4}{3}y_n - \frac{1}{3}y_{n-1} + \frac{2}{3}hf(t_{n+1}, y_{n+1}).$$

BDF metody jsou implicitní metody, do druhého řádu A-stabilní, poté se jejich oblast absolutní stability zužuje, což může například ovlivňovat výpočet systémů, jejichž vlastní čísla se nachází u imaginární osy. Od sedmého řádu již nejsou vhodné pro stiff systémy.

Důležitým typem numerických metod pro výpočet řešení diferenciální rovnice jsou implicitní metody Runge-Kuttova typu. Do nich zahrnujeme například tzv. Radauovy, Lobattovy, Gauss-Legendrovy metody, které se liší podle užitého pravidla pro numerickou kvadraturu. Radauovy metody jsou plně implicitní, A-stabilní, s stupňová metoda z této skupiny dosahuje řádu přesnosti  $2s - 1$ , ale zároveň jsou náročnější na implementaci. Mezi uvedené metody zahrnujeme RADAUIA a dále zkoumanou RADAUIIA metodu.

## 4 Numerické metody pro řešení diferenciálních algebraických rovnic

Numerické metody pro řešení soustavy DAE se stále vyvíjejí a stávající navržené metody jsou testovány a zlepšovány. Současné metody, které se aplikují na systémy DAE vycházejí

z metod odvozených pro ODE a upravují se pro použití i na algebraickou část. Snáze lze řešit systémy DAE s indexem 1 a existují již i metody pro soustavy DAE s indexem do 3. Pokud provedeme jednoduchou úvahu a místo systému ve tvaru

$$\begin{aligned}\dot{y} &= f(t, y, z), \\ 0 &= g(t, y, z)\end{aligned}\tag{20}$$

budeme uvažovat tzv. regularizovaný systém

$$\begin{aligned}\dot{y} &= f(t, y, z), \\ \xi \dot{z} &= g(t, y, z), \quad \xi \in (0, 1), \quad \xi \ll 1,\end{aligned}\tag{21}$$

přejde původní systém na soustavu obyčejných diferenciálních rovnic prvního řádu, ale vzniklý systém bude velmi stiff. Příslušná soustava ODE bude obtížněji řešitelná, než soustava DAE. Přesto tato úvaha vede k myšlence, že numerické metody pro řešení diferenciálních algebraických rovnic by měly být implicitní, tedy takové, které se využívají na stiff ODE.

Stěžejními metodami, kterými se řeší soustavy DAE jsou buď vícekové metody typu BDF (Backward Differentiation Formula) [2], nebo implicitní metody Runge-Kuttova typu, jakou je metoda RADAUIIA. Jednokrokové metody bývají výpočetně více náročné, protože vyžadují vyčíslení proměnných a jejich derivací i uvnitř kroku, ne jen v mezních diskretizovaných bodech daného intervalu. Pro metody s vyšším řádem tak bude schéma stále složitější. Vícekové metody vyžadují vyčíslení jen v aktuálním čase, ale oproti tomu je nutné pro  $k$ -krokovou metodu určit  $k$  počátečních hodnot. Tyto hodnoty potom dále ovlivňují i přesnost výsledného řešení.

Důležitým aspektem při řešení diferenciálních algebraických metod je celková konzistence počátečních podmínek. Například pro soustavu (9) a počáteční podmínku  $y(t_0) = y_0$ , můžeme získat hodnoty  $z(t_0)$  řešením nelineární soustavy  $g(t, y, z) = 0$ .

Další významnou součástí uvedených algoritmů je řešení vzniklé nelineární soustavy, k němuž se nejčastěji používá Newtonova metoda, případně její modifikace. Newtonova metoda pro řešení nelineární soustavy  $H(x) = 0$ , kde zobrazení  $H : \mathbb{R}^n \rightarrow \mathbb{R}^n$  a předpokládáme, že  $\exists x^* \in \mathbb{R}^n : H(x^*) = 0$ ,  $H$  je spojitě diferencovatelné v okolí  $x^*$  a Jacobián  $H'(x^*) = J(x^*)$  není singulární v okolí  $x^*$ . Newtonova metoda tedy vychází z Taylorova rozvoje,

$$H(\xi) = H(x^k) + H'(x^k)(\xi - x^k) + \dots$$

a následkem zanedbání vyšších členů dospějeme k rovnici

$$\begin{aligned}x^{k+1} &= x^k - [H'(x^k)]^{-1} H(x^k), \\ H(x^k) (x^{k+1} - x^k) &= -H(x^k), \\ H(x^k) s^k &= -H(x^k), \quad x^{k+1} = s^k + x^k,\end{aligned}$$

kde  $k$  označuje pořadí iterace. V každém kroku je vyžadováno řešení soustavy lineárních rovnic. Newtonova metoda je tedy časově náročná a pracná. Pokud se ale nacházíme se startovacím bodem dostatečně blízko hledanému bodu  $x^*$ , konverguje obecně rychle. Pro urychlení a zjednodušení se využívají některé modifikace, kdy například pro několik iterací

necháváme pevný Jacobián vypočtený v bodě  $x^0$ . Další alternativou je tzv. inexaktní Newtonova metoda [6], kdy se volí v každé iteraci člen  $\xi_k \in [0, 1)$  a nepřesně se řeší soustava  $J(x^k)s^k = -H(x^k)$ , ( $J(x^k) = H'(x^k)$ ), čímž se získá vektor  $s^k$  takový, že  $\|H(x^k) + J(x^k)s^k\| \leq \xi_k \|H(x^k)\|$ .

**Poznámka 2.** Mezi nejjednodušší formu diferenciálních algebraických rovnic patří semi-explicitní soustava (??). Předpokládejme dále, že máme konzistentní počáteční podmínky. Pro rovnici  $0 = g(t, y, z)$  existuje podle věty o implicitní funkci taková funkce  $\tilde{g}$ , že platí  $z = \tilde{g}(t, y)$ . DAE soustava potom přejde na soustavu ODE pro funkci  $\dot{y} = f(t, y, \tilde{g}(t, y))$ .

Existuje několik stávajících solverů pro řešení diferenciálních algebraických rovnic, jako např. DASSL (Differential Algebraic System Solver), algoritmus ve Fortranu navržený Lindou Petzold, založený na metodě BDF, s proměnným krokem a řádem. Využívá ho například software *Modelica*, je defaultní metodou v *Dymole* - komerčním prostředí pro modelování a simulace. Není ovšem vhodný pro systémy nelineární, s nespojitostmi a změny kroku jsou obecně ve víceukrokových metodách drahé. Lze jej aplikovat na systémy, které jsou stiff a s indexem 0 a 1. V Modelice lze také nalézt některé alternativy této metody, jako řešič DASSLSTEPS a DASSLWORT. Dalšími metodami v Modelice jsou ODASSL, modifikace metody DASSL pro přeurčené soustavy a metoda MEXX, která je uvažovaná pro speciální systémy DAE s indexem 2, a není určená pro stiff systémy ani pro obyčejné diferenciální soustavy. Ve volně šiřitelném programu *Scilab* pro numerické výpočty jsou od verze 5.5.0 přidány řešiče DASSL, DASRT, DASKR pro implicitní formu diferenciálních algebraickým rovnic. V prostředí *Matlab* jsou implementovány metody ode15s pro stiff soustavy ODE a DAE, proměnného řádu, využívající metody BDF, ode23t pro mírně stiff soustavy založená na lichoběžníkové metodě a od verze R2014 ode15i - metoda pro plně implicitní DAE rovnice s indexem 1.

Jiným typem metody, která se vyskytuje například v [8] je jednokroková metoda RADAUIIA vyvinutá E. Hairerem a G. Wannerem, využívaná pro některé soustavy DAE i s indexem až 3. Algoritmus RADAUIIA je již nově také zařazen v balíčku prostředí Dymola, avšak není defaultní metodou pro nedostatečnou robustnost a také z důvodu, že implementace není dostatečně otestována, podle publikace [5].

John Pryce a Ned Nedialkov v práci [11] navrhuje pro řešení soustavy DAE metodu DAETS založenou na Taylorově rozvoji. Tato metoda vykazuje dobré výsledky i pro soustavy s vyšším indexem, pokud požadujeme vysokou přesnost. Jestliže však upřednostňujeme efektivní rychlou metodu, je DAETS v porovnání s metodami DASSL a RADAU pomalejší.

**Poznámka 3.** Například výstup pro systém (18) s počáteční podmínkou  $x(0) = 1$ ,  $y(0) = 0$ ,  $z(0) = 0$  ze softwaru *Modelica* vypadá následovně

```
model Robertson
  Real x(start=1),y(start=0),z(start=0);
equation
  der(x) = -0.04x + 10e4*y*z;
  der(y) = 0.04x - 10e4*y*z - 3*10e7*y*y;
  der(z) = 3*10e7*y*y;
end Robertson
```

Součástí této práce je implementace numerických metod pro řešení diferenciálních algebraických rovnic typu BDF pro index 1 řádu 2 a 3, a pro vyšší indexy metody RADAUIIA se dvěma a třemi stupni, tedy řádu 3 a 5. Vybrané metody musí být vhodně přesné a dostatečně rychlé, aby byly schopné řešit soustavu DAE v reálném čase. Podle vlastností, typu a velikosti systému zvolíme vhodnou z nich.

Pro diferenciální algebraickou soustavu s indexem 1 budeme volit převážně metody typu BDF, a jestliže soustava nebude mít vlastní čísla v blízkosti imaginární osy vybereme metodu řádu 3, jinak buď RADAUIIA metodu nebo BDF nižšího řádu. Pro diferenciální algebraickou soustavu s vyšším indexem budeme volit metody RADAUIIA, a pokud soustava bude mít malý počet rovnic, můžeme uvažovat metodu řádu 5.

## 4.1 Metoda BDF pro soustavu DAE

Metoda BDF - Backward Differentiation Formula - je jednou ze základních metod pro řešení diferenciální algebraické soustavy s indexem 1 (plně implicitní typ) a někdy i s indexem 2 (pro případ semi-implicitních (7) a semi-explicitních tvarů (8)). Patří do skupiny více krokových metod, není tedy samostartující, pro zahájení výpočtu potřebuje předem určit  $k$  hodnot. Oblast  $A$ -stability inciduje pro  $k = 1, 2$  s celou zápornou polorovinou komplexní roviny, pro vyšší řády se tato oblast zmenšuje a vhodně použitelné pro stiff systémy jsou jen metody do 6. řádu.

Obecná BDF metoda má pro soustavu (1) DAE tvar

$$F\left(t_{n+1}, y_{n+1}, \frac{1}{\beta h} \sum_{j=0}^k \alpha_j y_{n-j+1}\right) = 0, \quad (22)$$

$\alpha_j \in \mathbb{R}, j = 1, \dots, k, \beta \in \mathbb{R}$ .

Mezi nejjednodušší metody patří již zmiňovaná implicitní Eulerova metoda, která po aplikaci na plně implicitní diferenciální algebraické soustavy vede na schéma

$$F\left(t_{n+1}, y_{n+1}, \frac{y_{n+1} - y_n}{h}\right) = 0.$$

Pro efektivní výpočet se budeme soustředit na formule řádu 2 a 3, které vedou na předpis

$$F\left(t_{n+1}, y_{n+1}, \frac{1}{h} \left(\frac{3}{2}y_{n+1} - 2y_n + \frac{1}{2}y_{n-1}\right)\right) = 0 \quad (23)$$

a

$$F\left(t_{n+1}, y_{n+1}, \frac{1}{h} \left(\frac{11}{6}y_{n+1} - 3y_n + \frac{3}{2}y_{n-1} - \frac{1}{3}y_{n-2}\right)\right) = 0. \quad (24)$$

Hodnota  $y_0$  musí být konzistentní počáteční podmínka a hodnoty  $y_1$ , resp.  $y_1$  a  $y_2$  získáme z jednokrokové RADAUIIA metody třetího, resp. pátého řádu.

## 4.2 Metoda RADAUIIA pro soustavu DAE

Metoda RADAUIIA patří mezi implicitní Runge-Kuttovy metody. Jedná se tedy o jednokrokovou metodu, která využívá aproximace neznámé funkce ve vnitřních bodech jednotlivých intervalů a následně Radauovu-Lobbatovu kvadraturu. Podle [10] je možné ji aplikovat i na soustavy s indexem 3. Obecné schéma metody RADAUIIA pro soustavu DAE (1) má následující předpis

$$\begin{aligned} F(t_n + c_i h, Y_{n+1,i}, K_{n+1,i}) &= 0, \\ Y_{n+1,i} &= y_n + h \sum_{j=1}^s a_{i,j} K_{n+1,j}, \quad i = 1, 2, \dots, s \\ y_{n+1} &= y_n + h \sum_{i=1}^s b_i K_{n+1,i}, \end{aligned}$$

kde pro stupeň  $s \in N$  je metoda řádu  $2s - 1$  a hodnoty  $c_i, b_i, a_{i,j}$ , pro  $j, i = 1, \dots, s$  jsou uváděny pro jednotlivé řády v tzv. Butcherovo tabulce.

$c_1$	$a_{1,1}$	$\dots$	$a_{1,s}$
$\vdots$	$\vdots$		$\vdots$
$c_s$	$a_{s,1}$	$\vdots$	$a_{s,s}$
	$b_1$	$\dots$	$b_s$

Tabulka 1: Butcherova tabulka pro implicitní Runge-Kuttovy metody.

$\frac{1}{3}$	$\frac{5}{12}$	$-\frac{1}{12}$
1	$\frac{3}{4}$	$\frac{1}{4}$
	$\frac{3}{4}$	$\frac{1}{4}$

Tabulka 2: Butcherova tabulka koeficientů RadauIIA metody pro  $s = 2$ .

$\frac{4-\sqrt{6}}{10}$	$\frac{88-7\sqrt{6}}{360}$	$\frac{296-169\sqrt{6}}{1800}$	$\frac{-2+3\sqrt{6}}{225}$
$\frac{4+\sqrt{6}}{10}$	$\frac{296+169\sqrt{6}}{1800}$	$\frac{88+7\sqrt{6}}{360}$	$\frac{-2-3\sqrt{6}}{225}$
1	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$
	$\frac{16-\sqrt{6}}{36}$	$\frac{16+\sqrt{6}}{36}$	$\frac{1}{9}$

Tabulka 3: Butcherova tabulka koeficientů RadauIIA metody pro  $s = 3$ .

Uvažujeme-li konkrétně metodu 3. řádu (pro  $s = 2$ ), získáme

$$\begin{aligned} F\left(t_n + \frac{1}{3}h, y_n + h\left(\frac{5}{12}K_{n+1,1} - \frac{1}{12}K_{n+1,2}\right), K_{n+1,1}\right) &= 0, \\ F\left(t_n + h, y_n + h\left(\frac{3}{4}K_{n+1,1} + \frac{1}{4}K_{n+1,2}\right), K_{n+1,2}\right) &= 0, \\ y_{n+1} &= y_n + h\left(\frac{3}{4}K_{n+1,1} + \frac{1}{4}K_{n+1,2}\right), \end{aligned}$$

a pro metodu pátého řádu ( $s = 3$ ) získáme

$$\begin{aligned}
F\left(t_n + \frac{4 - \sqrt{6}}{10}h, y_n + h\left(\frac{88 - 7\sqrt{6}}{360}K_{n+1,1} + \frac{296 - 169\sqrt{6}}{1800}K_{n+1,2} + \frac{-2 - 3\sqrt{6}}{225}K_{n+1,3}\right), K_{n+1,1}\right) &= 0, \\
F\left(t_n + \frac{4 + \sqrt{6}}{10}h, y_n + h\left(\frac{296 + 169\sqrt{6}}{1800}K_{n+1,1} + \frac{88 + 7\sqrt{6}}{360}K_{n+1,2} + \frac{-2 - 3\sqrt{6}}{225}K_{n+1,3}\right), K_{n+1,2}\right) &= 0, \\
F\left(t_n + h, y_n + h\left(\frac{16 - \sqrt{6}}{36}K_{n+1,1} + \frac{16 + \sqrt{6}}{36}K_{n+1,2} + \frac{1}{9}K_{n+1,3}\right), K_{n+1,3}\right) &= 0, \\
y_{n+1} = y_n + h\left(\frac{16 - \sqrt{6}}{36}K_{n+1,1} + \frac{16 + \sqrt{6}}{36}K_{n+1,2} + \frac{1}{9}K_{n+1,3}\right).
\end{aligned}$$

Implementovaný software uvažuje prioritně nejobecnější tvar diferenciálních algebraických rovnic - plně implicitní soustavu (6). Jako nejvíce univerzální metoda se jeví metoda RADAUIIA třetího řádu, která je dostatečně přesná i rychlá a je vhodná i pro metody s indexem větším než 1. Pro řešení nelineární soustavy využíváme metodu Newtonovu, klasického typu, protože budeme uvažovat diferenciální algebraické soustavy s menším počtem rovnic, a omezený počet iterací, aby se výpočet stihl v kratším čase, než je uvažovaná perioda. Pro přesnost a konvergenci Newtonovy metody je důležitý počáteční odhad, pro dobře volené hodnoty konverguje metoda kvadraticky. Jako počáteční odhad v jednotlivých iteracích volíme buď hodnotu  $y_n$  pro případ více krokových metod, a nebo hodnotu  $K_n$  pro Runge-Kuttovy metody.

V následujícím výzkumu se budeme snažit o zlepšení stávajících vybraných metod, případně sestavení jiné metody pro simulaci diferenciálních algebraických rovnic v reálném čase, dále o zrychlení výpočtu, určení dostatečně malého počtu iterací Newtonovy metody, resp. modifikaci Newtonovy metody pro zlepšení výpočetního výkonu a snížení počtu operací.

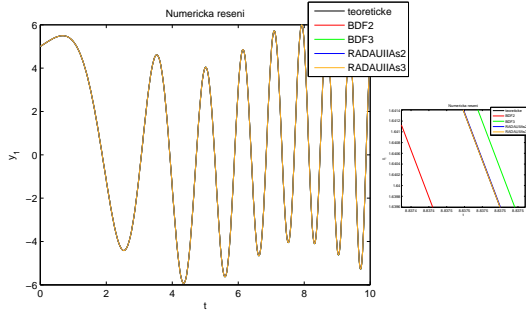
## 5 Simulace a testování

### 5.1 Simulace a testování metod na základních příkladech

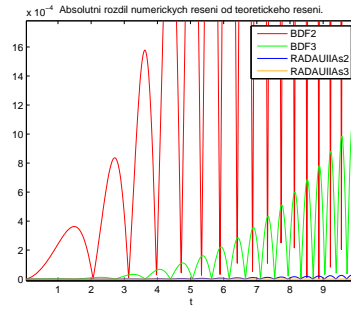
**Příklad 1.** Mějme soustavu

$$\begin{aligned}
\dot{y}_1 &= -ty_2 - (1 + t)z_1 \\
\dot{y}_2 &= ty_1 - (1 + t)z_2 \\
\frac{y_1 - z_2}{5} &= \cos\left(\frac{t^2}{2}\right) \\
\frac{y_2 + z_1}{5} &= \sin\left(\frac{t^2}{2}\right)
\end{aligned} \tag{25}$$

a  $y_1(0) = 5$ ,  $y_2(0) = 1$ ,  $z_1(0) = -1$ ,  $z_2(0) = 0$ ,  $t \in [0, 10]$  a aplikujme na ni metodu založenou na BDF i na RADAUIIA vztazích. Řešení soustavy představují funkce  $y_1 = \sin(t) + 5 \cos\left(\frac{t^2}{2}\right)$ ,  $y_2 = \cos(t) + 5 \sin\left(\frac{t^2}{2}\right)$ ,  $z_1 = -\cos(t)$ ,  $z_2 = \sin(t)$ . Výsledná numerická řešení a porovnání oproti teoretickému řešení jsou uvedeny na obrázcích 3,4.

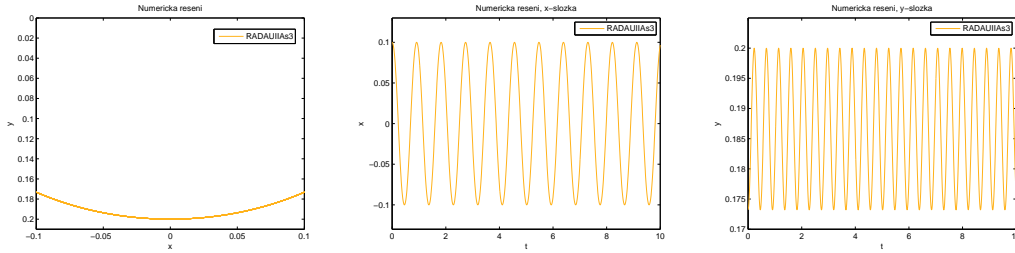


Obrázek 3: Numerická řešení soustavy (25), první složka  $y_1(t)$ .



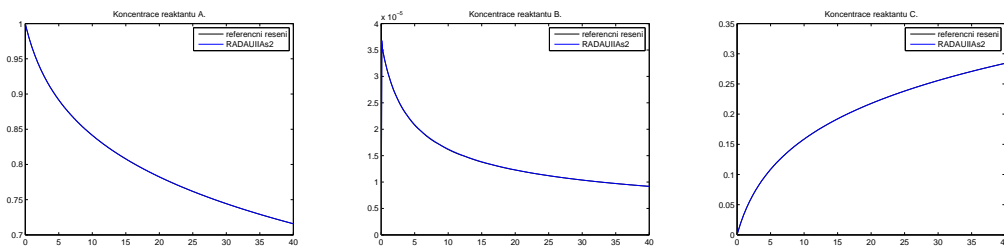
Obrázek 4: Rozdíl numerických řešení od teoretického, první složka  $y_1(t)$ .

**Příklad 2.** Uvažujme model jednoduchého kyvadla (5) a využijme metody RADAUIIA. Výsledný průběh vypočtený metodou třetího a pátého řádu je velmi podobný, stejně tak obě metody výborně splňují algebraickou rovnici. Průběh pro  $s = 3$  je znázorněn na obrázku 5.



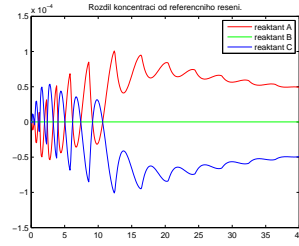
Obrázek 5: Průběh jednoduchého kyvadla pro diferenciální algebraickou soustavu (14).

**Příklad 3.** Soustavu popisující proces chemické reakce budeme řešit jednak jako systém DAE (17) příslušnými metodami a jednak jako soustavu ODE (18). Pro tento případ obdržíme v podstatě též řešení. Hodnoty koncentrací reaktantů  $A$ ,  $B$  a  $C$  pro počáteční podmínky  $[1 \ 0 \ 0]^T$  a interval  $[0, 40]$  jsou znázorněny na obrázku 6, za referenční řešení budeme považovat řešení vypočtené metodou ode23s s velmi velkou přesností.

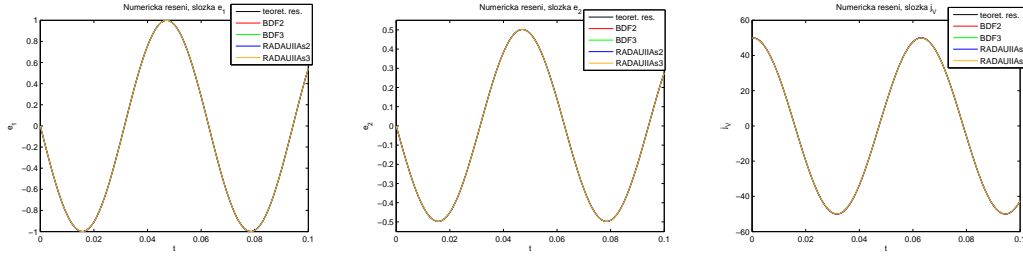


Obrázek 6: Průběh koncentrací reaktantů  $A$ ,  $B$ ,  $C$  pro soustavu (17).

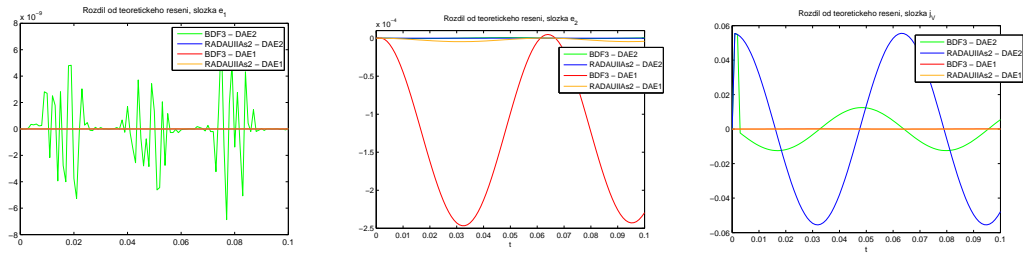
**Příklad 4.** Pro elektrický obvod (15) a (16) uvažujme počáteční podmínky  $[0, 0, 50]^T$ . Odpovídající simulační výsledky jsou ukázány na obrázcích 8 a 9. V tomto případě můžeme porovnat numerické řešení s řešením vypočteným analyticky.



Obrázek 7: Rozdíl koncentrací od referenčního řešení pro soustavu (17).



Obrázek 8: Časový průběh složek pro soustavu (15).



Obrázek 9: Rozdíl hodnot vypočtených pro soustavu (16) a (15) od teoretického řešení.

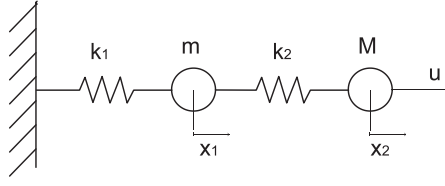
## 5.2 Rozbor simulace na příkladě descriptor systému

Uvažujme nejprve hmotový model s pružinami podle obrázku 10. Pro něj můžeme odvodit pohybové rovnice

$$\begin{aligned} F_1 &= -k_1 x_1 + k_2 (x_2 - x_1) = m \ddot{x}_1, \\ F_2 &= -k_2 (x_2 - x_1) + u = M \ddot{x}_2, \end{aligned}$$

$$\begin{aligned} \ddot{x}_1 &= -\frac{k_1}{m} x_1 + \frac{k_2}{m} (x_2 - x_1) = -\frac{k_1 + k_2}{m} x_1 + \frac{k_2}{m} x_2, \\ \ddot{x}_2 &= \frac{k_2}{M} x_1 - \frac{k_2}{M} x_2 + \frac{1}{M} u \end{aligned}$$





Obrázek 10: Model s pružinami.

$$\begin{aligned}
 \dot{x}_1 &= v_1 \\
 \dot{v}_1 &= -\frac{k_1 + k_2}{m}x_1 + \frac{k_2}{m}x_2 \\
 \dot{x}_2 &= v_2 \\
 \dot{v}_2 &= \frac{k_2}{M}x_1 - \frac{k_2}{M}x_2 + \frac{1}{M}u.
 \end{aligned} \tag{26}$$

Soustavu (26) můžeme dále upravit do tvaru

$$\begin{aligned}
 \dot{x}_1 &= v_1 \\
 m\dot{v}_1 &= -(k_1 + k_2)x_1 + k_2x_2 \\
 \dot{x}_2 &= v_2 \\
 \dot{v}_2 &= \frac{k_2}{M}x_1 - \frac{k_2}{M}x_2 + \frac{1}{M}u.
 \end{aligned}$$

Nyní uvažujme hmotu  $m \rightarrow 0$ , čímž získáme

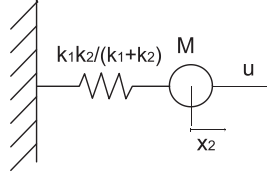
$$\begin{aligned}
 \dot{x}_1 &= v_1 \\
 0 &= -(k_1 + k_2)x_1 + k_2x_2 \\
 \dot{x}_2 &= v_2 \\
 \dot{v}_2 &= \frac{k_2}{M}x_1 - \frac{k_2}{M}x_2 + \frac{1}{M}u.
 \end{aligned}$$

Z této soustavy můžeme snadno určit polohu  $x_1$  a dosadit ji do vztahu pro rychlost druhého bodu

$$\begin{aligned}
 x_1 &= \frac{k_2}{k_1 + k_2}x_2 \\
 \dot{x}_2 &= v_2 \\
 \dot{v}_2 &= \frac{k_2}{M}x_1 - \frac{k_2}{M}x_2 + \frac{1}{M}u = -\frac{k_1 k_2}{(k_1 + k_2)M}x_2 + \frac{1}{M}u.
 \end{aligned}$$

Ekvivalentní systém pro  $m \rightarrow 0$  je znázorněn na obrázku 11. Tato úvaha nás vede k soustavě diferenciálních algebraických rovnic

$$\begin{aligned}
 \dot{x}_2 &= v_2 \\
 \dot{v}_2 &= \frac{k_2}{M}x_1 - \frac{k_2}{M}x_2 + \frac{1}{M}u \\
 0 &= -(k_1 + k_2)x_1 + k_2x_2.
 \end{aligned} \tag{27}$$

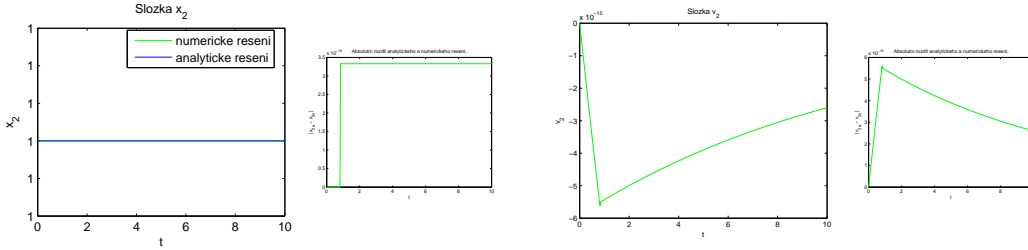


Obrázek 11: Ekvivalentní model s pružinami.

Řešení soustavy DAE by potom mělo odpovídat řešení soustavy ODE

$$\begin{aligned}\dot{x}_2 &= v_2 \\ \dot{v}_2 &= -\frac{k_1 k_2}{(k_1 + k_2)M} x_2 + \frac{1}{M} u.\end{aligned}\quad (28)$$

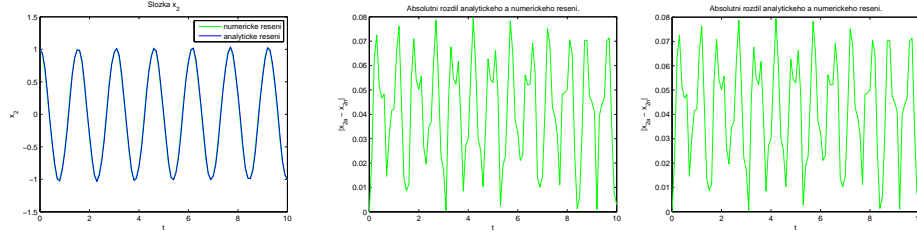
Soustavu obyčejných diferenciálních rovnic můžeme vyřešit analyticky pro daný vstup a následně tak budeme porovnávat numerické řešení soustavy DAE oproti teoretickému. Počáteční hodnoty budeme uvažovat  $x_2(0) = 1$  a  $v_2(0) = 0$ , parametry modelu jsme zvolili  $k_1 = 5$ ,  $k_2 = 10$ ,  $M = \frac{1}{5}$ . Pro soustavu DAE je nutné dopočítat konzistentní počáteční podmínku i pro neznámou polohu  $x_1$ , tj.  $x_1(0) = \frac{2}{3}$ . Obrázky 12 demonstrují numerické simulace pro výsledné systémy. Nejprve vezmeme pro jednoduchost vstup ve tvaru  $u = \frac{k_1 k_2}{k_1 + k_2} \cdot 1$ , pro který přímo známe jednoduchý tvar výstupu, neboť bude platit  $v_2(t) = 0$ ,  $x_2(t) = 1$ . získáme a můžeme tak oproti němu porovnat numerické řešení.



Obrázek 12: Porovnání numerického řešení vůči analytickému,  $u = \frac{k_1 k_2}{k_1 + k_2}$ .

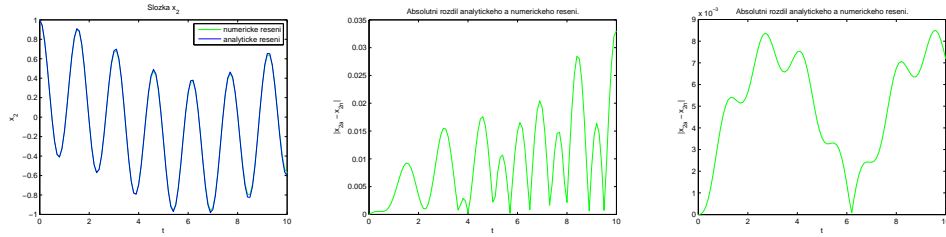
Nyní uvažujme obecnější vstup  $u = \cos\left(\frac{k_1 k_2}{(k_1 + k_2)M} t\right)$ . Analytické řešení soustavy (28) má tvar  $x_2(t) = \frac{479}{470} \cos\left(\frac{5}{3} \sqrt{6} t\right) - \frac{9}{470} \cos\left(\frac{50}{3} t\right)$  a  $v_2(t) = -\frac{479}{282} \sin\left(\frac{5}{3} \sqrt{6} t\right) \sqrt{6} + \frac{15}{47} \sin\left(\frac{50}{3} t\right)$ . Výsledná řešení jsou uvedena na obrázku 13. Pro tento případ je zajímavé sledovat časovou náročnost numerického algoritmu. Pro hodnotu kroku  $h = 0.1$  při simulačních testech bylo potřeba průměrně nejhůře  $1.3 \cdot 10^{-4}$  simulačního času. Pokud bychom zjemnili krok a samotnou výpočetní část metody pro získání hodnoty  $y_{n+1}$  volali vícekrát, můžeme obecně předpokládat prodloužení doby výpočtu a zlepšení přesnosti vypočteného řešení. Protože je ale vstupní funkce velice kmitavá a v jednotlivých iteracích ji neaproximujeme dostatečně přesně, bude se řešení přesností podobat předchozímu s hrubším krokem. Časová náročnost vychází po opakovaných výpočtech také podobně, (řádově přibližně  $10^{-4}$ ).

Vezmeme nyní vstup ve tvaru  $u = \cos\left(\frac{1}{2} t\right)$ . Analytické řešení soustavy (28) je  $x_2(t) = \frac{137}{197} \cos\left(\frac{5}{3} \sqrt{6} t\right) + \frac{60}{197} \cos\left(\frac{1}{2} t\right)$  a  $v_2(t) = -\frac{685}{591} \sin\left(\frac{5}{3} \sqrt{6} t\right) \sqrt{6} - \frac{30}{197} \sin\left(\frac{1}{2} t\right)$ . Výsledné časové průběhy jsou znázorněny na obrázku 14. Pokud zvolíme krok  $h = 0.1$ , trvá výpočet



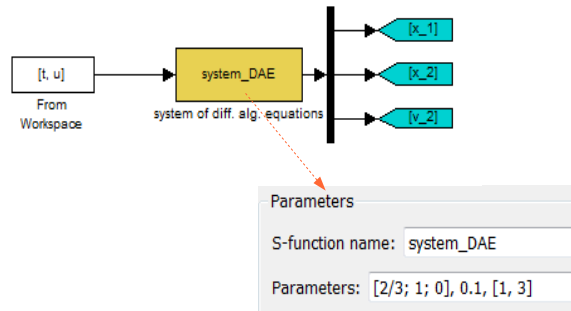
Obrázek 13: Porovnání numerického řešení vůči analytickému,  $u = \cos\left(\frac{k_1 k_2}{(k_1 + k_2)M} t\right)$ , uprostřed řešení s hrubším krokem, vpravo s desetkrát opakovanou iterací.

jedné iterace řádově opět asi  $10^{-4}$  simulačního času. Pro případ opakovaného volání s desetkrát jemnějším krokem během iterace dostaneme v absolutním porovnání s teoretickým řešením o řád přesnější výsledek a časová náročnost se nezhorší. U tohoto příkladu jsou postačující pro rozumnou přesnost dvě (případně pro výbornou tři) iterace Newtonovy metody.



Obrázek 14: Porovnání numerického řešení vůči analytickému,  $u = \cos\left(\frac{1}{2}t\right)$ , vpravo desetkrát opakované volání vnitřní části metody

Na obrázku 15 je znázorněn blok DAE při simulaci hmotového modelu s pružinami popsaného soustavou diferenciálních algebraických rovnic.



Obrázek 15: Blok DAE pro řešení soustavy DAE modelu (27).

### 5.3 Možnosti simulace

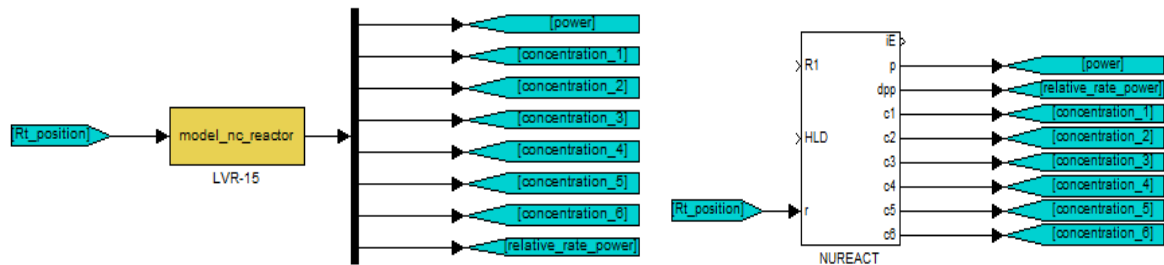
Budeme-li se specializovat na některé vybrané problémy - energetická zařízení, je rozumné oprostit se od obecnosti simulace a příslušná numerická schémata aplikovat přímo na rovnice popisující speciální systém, například model vodní turbíny, parní turbíny, parního kotle nebo jaderného reaktoru. Daný výpočet se jednak zrychlí a také můžeme přímo využít některých znalostí o systému, jako například index soustavy DAE, případně nepočítat Jacobián numericky, ale definovat je pro tento konkrétní příklad přímo, nebo bychom na základě složitosti a typu systému mohli snížit počet iterací Newtonovy metody. Výpočet urychlíme dále analýzou výsledného numerického předpisu. Tímto by vznikly konkrétní bloky pro vybraná energetická zařízení.

Příkladem může být blok NUREACT, v němž je na soustavu obyčejných diferenciálních rovnic přímo implementován algoritmus RADAUIIA druhého stupně. Jaderný reaktor typu LR-0 můžeme popsat bodovým modelem jako soustavu sedmi obyčejných diferenciálních rovnic pro neznámé funkce výkon neutronového toku  $P(t)$  a koncentrace šesti skupin zpožděným neutronů  $c_i(t)$ , pro  $i = 1, \dots, 6$ . Blíže o tomto modelu pojednává zpráva [9].

$$\begin{aligned} \frac{dP(t)}{dt} &= \frac{k_{\text{ef}}(t)(1 - \beta) - 1}{\tau} P(t) + \sum_{i=1}^m \lambda_i c_i(t) \\ \frac{dc_i(t)}{dt} &= \beta_i \frac{k_{\text{ef}}(t)}{\tau} P(t) - \lambda_i c_i(t), \quad i = 1, \dots, m, \quad m = 6 \\ k_{\text{ef}}(t) &= \frac{1}{1 - \rho(t)}, \end{aligned} \quad (29)$$

kde  $k_{\text{ef}}(t)$  je efektivní multiplikační faktor,  $\rho(t)$  reaktivita, která je řízena regulačními tyčemi, hodnoty  $\beta_i$ ,  $\lambda_i$  odpovídají parametrům zpožděných neutronů,  $\beta = \sum_i \beta_i$  a  $\tau$  době života jedné generace neutronů. Kinetické parametry reaktoru LVR-15 pro model a vývoj regulátoru budeme uvažovat podle [9].

Soustava popisující dynamické vlastnosti jaderného reaktoru nulového výkonu (29) je tvořena obyčejnými diferenciálními rovnicemi. Její numerický výpočet je proto snazší, než u soustavy diferenciálních algebraických rovnic. Blok modelu jaderného reaktoru NUREACT je znázorněn na obrázku 16.



Obrázek 16: Blok NUREACT, vlevo implementace v nástroji Matlab/Simulink, vpravo implementace v řídicím systému REX.

Odpovídající rovnice pro soustavu (29) a aplikaci implicitní 2-stupňové Rungeho-Kuttovy metody mají tvar:

$$g_i = y_n + h \sum_{j=1}^s a_{i,j} f(x_n + c_j h, g_j) \quad i = 1, 2 \quad (30)$$

$$y_{n+1} = y_n + h \sum_{j=1}^2 b_j f(x_n + c_j h, g_j),$$

kde  $y$  je vektor stavových proměnných  $y = [P, c_1, c_2, c_3, c_4, c_5, c_6]^T$  a  $g_1, g_2$  jsou pomocné vektory,  $g_1$  se složkami  $\gamma_i$ , pro  $i = 1, \dots, 7$  a  $g_2$  se složkami  $\gamma_i$ , pro  $i = 8, \dots, 14$ , neboli  $g_1 = [\gamma_1, \gamma_2, \dots, \gamma_7]^T$ ,  $g_2 = [\gamma_8, \gamma_9, \dots, \gamma_{14}]^T$ , a jejich složky jsou dány vztahy

$$\begin{aligned} \gamma_1 = & \left( \frac{1 - \beta}{1 - \rho(t + 1/3 h)} - 1 \right) \left( P + \frac{5}{12} h \gamma_1 - 1/12 h \gamma_8 \right) \tau^{-1} + \\ & + \lambda_1 \left( c_1 + \frac{5}{12} h \gamma_2 - 1/12 h \gamma_9 \right) + \lambda_2 \left( c_2 + \frac{5}{12} h \gamma_3 - 1/12 h \gamma_{10} \right) + \\ & + \lambda_3 \left( c_3 + \frac{5}{12} h \gamma_4 - 1/12 h \gamma_{11} \right) + \lambda_4 \left( c_5 + \frac{5}{12} h \gamma_5 - 1/12 h \gamma_{12} \right) + \\ & + \lambda_5 \left( c_5 + \frac{5}{12} h \gamma_6 - 1/12 h \gamma_{13} \right) + \lambda_6 \left( c_6 + \frac{5}{12} h \gamma_7 - 1/12 h \gamma_{14} \right) \end{aligned}$$

$$\gamma_2 = \beta_1 \left( P + \frac{5}{12} h \gamma_1 - 1/12 h \gamma_8 \right) (1 - \rho(t + 1/3 h))^{-1} \tau^{-1} - \lambda_1 \left( c_1 + \frac{5}{12} h \gamma_2 - 1/12 h \gamma_9 \right)$$

$$\gamma_3 = \beta_2 \left( P + \frac{5}{12} h \gamma_1 - 1/12 h \gamma_8 \right) (1 - \rho(t + 1/3 h))^{-1} \tau^{-1} - \lambda_2 \left( c_2 + \frac{5}{12} h \gamma_3 - 1/12 h \gamma_{10} \right)$$

$$\gamma_4 = \beta_3 \left( P + \frac{5}{12} h \gamma_1 - 1/12 h \gamma_8 \right) (1 - \rho(t + 1/3 h))^{-1} \tau^{-1} - \lambda_3 \left( c_3 + \frac{5}{12} h \gamma_4 - 1/12 h \gamma_{11} \right)$$

$$\gamma_5 = \beta_4 \left( P + \frac{5}{12} h \gamma_1 - 1/12 h \gamma_8 \right) (1 - \rho(t + 1/3 h))^{-1} \tau^{-1} - \lambda_4 \left( c_5 + \frac{5}{12} h \gamma_5 - 1/12 h \gamma_{12} \right)$$

$$\gamma_6 = \beta_5 \left( P + \frac{5}{12} h \gamma_1 - 1/12 h \gamma_8 \right) (1 - \rho(t + 1/3 h))^{-1} \tau^{-1} - \lambda_5 \left( c_5 + \frac{5}{12} h \gamma_6 - 1/12 h \gamma_{13} \right)$$

$$\gamma_7 = \beta_6 \left( P + \frac{5}{12} h \gamma_1 - 1/12 h \gamma_8 \right) (1 - \rho(t + 1/3 h))^{-1} \tau^{-1} - \lambda_6 \left( c_6 + \frac{5}{12} h \gamma_7 - 1/12 h \gamma_{14} \right)$$

$$\begin{aligned}\gamma_8 = & \left( \frac{1-\beta}{1-\rho(t+h)} - 1 \right) (P + 3/4 h \gamma_1 + 1/4 h \gamma_8) \tau^{-1} + \\ & + \lambda_1 (c_1 + 3/4 h \gamma_2 + 1/4 h \gamma_9) + \lambda_2 (c_2 + 3/4 h \gamma_3 + 1/4 h \gamma_{10}) + \\ & + \lambda_3 (c_3 + 3/4 h \gamma_4 + 1/4 h \gamma_{11}) + \lambda_4 (c_4 + 3/4 h \gamma_5 + 1/4 h \gamma_{12}) + \\ & + \lambda_5 (c_5 + 3/4 h \gamma_6 + 1/4 h \gamma_{13}) + \lambda_6 (c_6 + 3/4 h \gamma_7 + 1/4 h \gamma_{14})\end{aligned}$$

$$\gamma_9 = \frac{\beta_1 (P + 3/4 h \gamma_1 + 1/4 h \gamma_8)}{(1 - \rho(t+h)) \tau} - \lambda_1 (c_1 + 3/4 h \gamma_2 + 1/4 h \gamma_9)$$

$$\gamma_{10} = \frac{\beta_2 (P + 3/4 h \gamma_1 + 1/4 h \gamma_8)}{(1 - \rho(t+h)) \tau} - \lambda_2 (c_2 + 3/4 h \gamma_3 + 1/4 h \gamma_{10})$$

$$\gamma_{11} = \frac{\beta_3 (P + 3/4 h \gamma_1 + 1/4 h \gamma_8)}{(1 - \rho(t+h)) \tau} - \lambda_3 (c_3 + 3/4 h \gamma_4 + 1/4 h \gamma_{11})$$

$$\gamma_{12} = \frac{\beta_4 (P + 3/4 h \gamma_1 + 1/4 h \gamma_8)}{(1 - \rho(t+h)) \tau} - \lambda_4 (c_4 + 3/4 h \gamma_5 + 1/4 h \gamma_{12})$$

$$\gamma_{13} = \frac{\beta_5 (P + 3/4 h \gamma_1 + 1/4 h \gamma_8)}{(1 - \rho(t+h)) \tau} - \lambda_5 (c_5 + 3/4 h \gamma_6 + 1/4 h \gamma_{13})$$

$$\gamma_{14} = \frac{\beta_6 (P + 3/4 h \gamma_1 + 1/4 h \gamma_8)}{(1 - \rho(t+h)) \tau} - \lambda_6 (c_6 + 3/4 h \gamma_7 + 1/4 h \gamma_{14}).$$

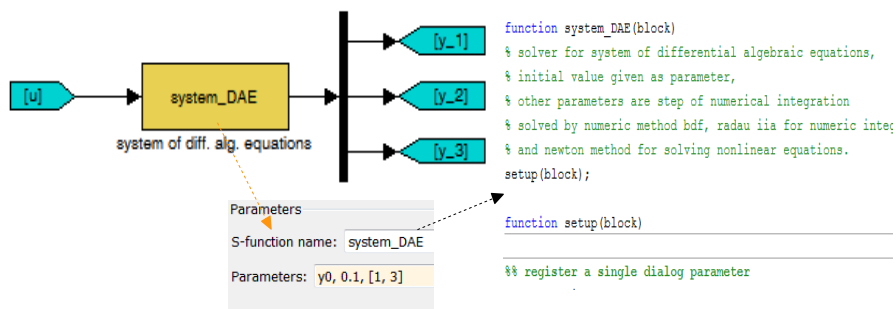
Reaktivitu v čase  $t + \frac{1}{3}h$  a  $t+h$  jsme aproximovali konstantní funkcí, tj.  $\rho(t) = \rho(t + \frac{1}{3}) = \rho(t+h)$ . Výslednou hodnotu  $y_{n+1}$  získáme pro vyřešená  $g_1, g_2$  podle předpisu

$$y_{n+1} = y_n + h \left( \frac{3}{4} g_1 + \frac{1}{4} g_2 \right).$$

## 5.4 Blok pro řešení soustavy diferenciálních algebraických rovnic

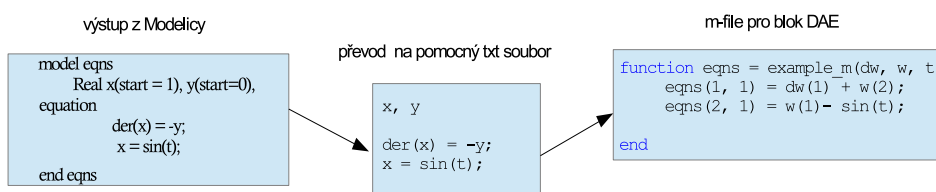
Blok pro řešení soustavy diferenciálních algebraických rovnic (případně obyčejných diferenciálních rovnic) bude spouštěn s konstantním krokem, který bude definován jako jeden z parametrů pracovního bloku. Dalšími budou počáteční hodnota vektoru neznámých a dimenze vektoru vstupů a výstupů. Ukázka bloku je znázorněna na obrázku 17.

Blok je vytvořen jako S-funkce v *Matlabu* a je převoditelný do jazyka *C*. Soustavu, kterou budeme řešit, můžeme interpretovat v m-file (*example\_m*) nástroji *Matlab*, který



Obrázek 17: Blok DAE pro řešení soustavy diferenciálních algebraických rovnic.

pro zjednodušené nastínění představy lze sestavit podle principu uvedeného na obrázku 18.

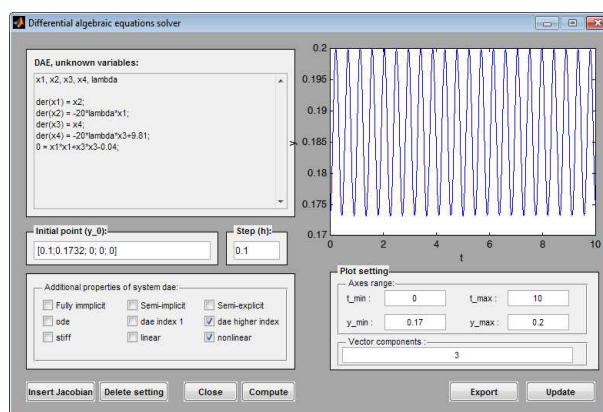


Obrázek 18: Zjednodušená představa interpretace rovnic pro vytvoření m-file řešeného blokem DAE.

**Poznámka 4.** Příslušné numerické metody pro testování můžeme volat klasicky následujícími příkazy

`radauiia2_m(pocatecni_podminky, vstup(i), t(i-1), h)`

a analogicky pro `radauiia3_m`, `bdf2_m`, `bdf3_m`, pro jednoduché použití lze využít GUI, viz obrázek 19.



Obrázek 19: GUI sestavené v Matlabu pro jednoduchá testování.

## Reference

- [1] Bächle, S., and F. Ebert. *Element-based topological index reduction for differential-algebraic equations in circuit simulation*, 2005.
- [2] Brenan, Kathryn Eleda, Stephen L. Campbell, and Linda Ruth Petzold. Numerical solution of initial-value problems in differential-algebraic equations. Vol. 14. Siam, 1996.
- [3] Cellier E. F.; *Mathematical Modeling of Physical Systems, Electrical Circuit I*, 2012.
- [4] Cellier E. F.; *Numerical Simulation of Dynamic Systems*, Lectures, Department of Computer Science, ETH Zurich, 2013.
- [5] Cellier E. F., Kofman E.; *Continuous System Simulation*, Springer-Verlag New York, Inc., Secaucus, NJ, 2006.
- [6] Drag P., M. Kwiatkowska; *Inexact Newton Matrix-free Methods For Solving Complex Biotechnological Systems*, Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, pp. 597-602, DOI: 10.15439/2014F216, ACSIS, Vol. 2, 2014.
- [7] Gobbert M. K.; *Robertson's Example for Stiff Differential Equations*. Technical Report, Arizona State University, 1996.
- [8] Hairer, E. and Nørsett; S.P. and Wanner, G., *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*, ISBN: 9783540604525, 1996, Springer.
- [9] Königsmarková J., M. Schlegel; *Numerické řešení a analýza bodového modelu jaderného reaktoru nulového výkonu*, Výzkumná zpráva NTIS P1b/2013 CANUT PB3 TE01020455DV001, 2013.
- [10] Per Grove T., C. Bendtsen; *Numerical Solution of Differential Algebraic Equations*, Technical University of Denmark, 1999.
- [11] Pryce, J., Nedialkov N.; *The DAETS Differential-Algebraic Equation Solver*, University of Louisiana at Lafayette, 2008.