

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ANALÝZA STIFF SOUSTAV DIFERENCIÁLNÍCH ROVNIC

DISERTAČNÍ PRÁCE

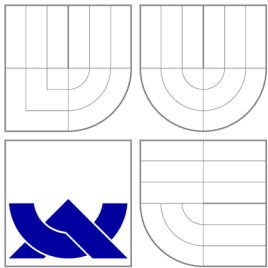
PHD THESIS

AUTOR PRÁCE

AUTHOR

Ing. VÁCLAV ŠÁTEK

BRNO 2011



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV INTELIGENTNÍCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF INTELLIGENT SYSTEMS

ANALÝZA STIFF SOUSTAV DIFERENCIÁLNÍCH ROVNIC

STIFF SYSTEMS ANALYSIS

DISERTAČNÍ PRÁCE

PHD THESIS

AUTOR PRÁCE

AUTHOR

Ing. VÁCLAV ŠÁTEK

VEDOUCÍ PRÁCE

SUPERVISOR

doc. Ing. JIŘÍ KUNOVSKÝ, CSc.

BRNO 2011

Abstrakt

Řešení tuhých (“stiff”) soustav diferenciálních rovnic patří i v současné době stále mezi komplikované úlohy. Základním problémem je přesná definice tuhých systémů. Jednoznačná definice tuhých systémů stále neexistuje. Problémem je dále i detekce tuhých systémů diferenciálních rovnic. K řešení se v praxi využívají implicitní numerické metody nižších řádů, jejichž oblasti stability jsou relativně velké.

Ve své práci se zabývám numerickým řešením obyčejných diferenciálních rovnic především numerickým výpočtem využívajícím metody Taylorovy řady. Analyzuji vznik tuhosti jednotlivých systémů diferenciálních rovnic, ukazuji možnou náhradu některých tuhých systémů ekvivalentními systémy bez tuhosti. Dále zkoumám možnost detekce tuhých systémů pomocí členů explicitní Taylorovy řady, zaměřuji se na stabilitu explicitní a implicitní Taylorovy řady. V závěru práce experimentálně ověřuji možnosti řešení tuhých systémů s využitím implicitní Taylorovy řady, zkoumám vhodnost použití víceslovní aritmetiky a navrhuji vhodný paralelizovatelný algoritmus implicitní Taylorovy řady s rekurentním výpočtem členů a Newtonovou iterační metodou (ITMRN).

Klíčová slova

tuhé soustavy diferenciálních rovnic, Taylorova řada, obyčejné diferenciální rovnice, numerické řešení diferenciálních rovnic

Abstract

The solving of stiff systems is still a contemporary sophisticated problem. The basic problem is the absence of precise definition of stiff systems. A question is also how to detect the stiffness in a given system of differential equations. Implicit numerical methods are commonly used for solving stiff systems. The stability domains of these methods are relatively large but the order of them is low.

The thesis deals with numerical solution of ordinary differential equations, especially numerical calculations using Taylor series methods. The source of stiffness is analyzed and the possibility how to reduce stiffness in systems of ordinary differential equations (ODEs) is introduced. The possibility of detection stiff systems using explicit Taylor series terms is analyzed. The stability domains of explicit and implicit Taylor series are presented. The solutions of stiff systems using implicit Taylor series method are presented in many examples. The multiple arithmetic must be used in many cases. The new suitable parallel algorithm based on implicit Taylor series method with recurrent calculation of Taylor series terms and Newton iteration method (ITMRN) is proposed.

Keywords

Stiff Systems of Differential Equations, Taylor Series, Ordinary Differential Equations, Numeric Solution of Differential Equations

Citace

Václav Šátek: Analýza stiff soustav diferenciálních rovnic, disertační práce, Brno, FIT VUT v Brně, 2011

Analýza stiff soustav diferenciálních rovnic

Prohlášení

Prohlašuji, že jsem tuto disertační práci vypracoval samostatně pod vedením pana doc. Ing. Jiřího Kunovského CSc. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....
Václav Šátek
3. listopadu 2011

Poděkování

V průběhu svého studia jsem byl podporován následujícími granty: FRVŠ MŠMT FR2681/2010/G1 a FIT-S-10-1. Dále bych chtěl poděkovat svému školiteli doc. Ing. Jiřímu Kunovskému, CSc. za jeho podporu při studiu a za odbornou pomoc při zpracování této práce. Můj dík patří také rodině, Hance a všem ostatním, kteří mě po dobu studia jakkoliv podporovali.

© Václav Šátek, 2011.

Tato práce vznikla jako školní dílo na Vysokém učení technickém v Brně, Fakultě informačních technologií. Práce je chráněna autorským zákonem a její užití bez udělení oprávnění autorem je nezákonné, s výjimkou zákonem definovaných případů.

Obsah

1	Úvod	8
1.1	Motivace, cíle disertační práce	8
1.2	Struktura práce	9
2	Řešení diferenciálních rovnic	11
2.1	Matematické základy	11
2.1.1	Jacobiova matice	11
2.1.2	Vlastní čísla	12
2.1.3	Numerické řešení lineárních rovnic	13
2.1.4	Numerické řešení nelineárních rovnic	14
2.2	Numerické řešení obyčejných diferenciálních rovnic	16
2.2.1	Základní pojmy	16
2.2.2	Jednokrokové metody	18
2.2.3	Více krokové metody	24
2.3	Shrnutí	27
3	Tuhé systémy	29
3.1	Stabilita	29
3.1.1	Oblasti stabilit numerických metod	31
3.2	Padé aproximace exponenciální funkce	34
3.2.1	Oblast stability řádu hvězd	35
3.3	Shrnutí	37
4	Analýza a řešení tuhých systémů	38
4.1	Detekce tuhých systémů	38
4.2	Řešení tuhých systémů s využitím ekvivalentního systému	42
4.2.1	Problém 1	42
4.2.2	Problém 2	43
4.3	Příklady v elektrotechnice	47
4.3.1	RC elektrický obvod	47
4.3.2	RLC elektrický obvod	49
4.3.3	Van der Polův oscilátor	51
4.4	Převod SLAR na SDR	52
4.5	Newtonova metoda	54
4.6	Shrnutí	57

5	Moderní metody Taylorovy řady	58
5.1	Úvod	58
5.2	Současný stav	59
5.3	Rekurentní výpočet	59
5.4	Tvořící rovnice	60
5.5	Stabilita, konvergence a konzistence	62
5.5.1	Dahlquistův problém	63
5.6	Příklady výpočtu	73
5.6.1	Semi-analytický výpočet	73
5.6.2	Speciální testovací soustava diferenciálních rovnic	77
5.6.3	Problém stability	83
5.6.4	Problém stiff exponenciály	89
5.6.5	Problém parazitních prvků v RLC obvodu	96
5.6.6	Problém Van Der Pol	99
5.7	Shrnutí	102
6	Závěr	103
6.1	Zvolený přístup k řešení	103
6.2	Dosažené výsledky	103
6.3	Možnosti dalšího výzkumu	104
	Literatura	104
	Přehled publikací autora	110
	Přílohy	112
A	RLC parazitní	113
A.1	Analytické řešení	113
A.2	Numerické řešení	120
A.2.1	Explicitní Taylorova metoda	120
A.2.2	Implicitní Taylorova metoda	121
A.2.3	ITMRN	126
B	Obrázky a grafy	131

Seznam obrázků

3.1	Oblast absolutní stability - explicitní Eulerova metoda	31
3.2	Oblast stability A-stabilní numerické metody	31
3.3	Oblast absolutní stability - lichoběžníková metoda	32
3.4	Oblast stability $A(\alpha)$ -stabilní numerické metody	33
3.5	Oblast absolutní stability - implicitní Eulerova metoda	34
3.6	Oblast řádu hvězd - (0,1) Padé aproximace	36
3.7	Oblast řádu hvězd - (0,2) Padé aproximace	36
3.8	Oblast řádu hvězd - (1,1) Padé aproximace	36
3.9	Oblast řádu hvězd - (2,3) Padé aproximace	36
4.1	Členy Taylorovy řady, $a = 1$	40
4.2	Členy Taylorovy řady, $a = 10$	40
4.3	Členy Taylorovy řady po automatickém snížení integračního kroku	41
4.4	Příklad 1 - TKSL/C	44
4.5	RC obvod	47
4.6	RC obvod - TKSL/386: $a = 5, \omega = 1$	48
4.7	Elektrický obvod RLC bez parazitních prvků (vlevo) s parazitními prvky (vpravo)	49
4.8	Elektrický RLC obvod bez parazitních prvků - TKSL/386, časový průběh napětí u_C	50
4.9	Elektrický obvod s parazitními prvky - TKSL/386, časový průběh napětí u_1, u_2	50
4.10	Van der Pol - TKSL/C, $\mu = 0.01$	51
4.11	Van der Pol - TKSL/C, $\mu = 10$	52
4.12	Řešení SDR (4.35) pomocí TKSL/386	54
5.1	Oblast stability explicitní Taylorova řada $ORD = 1, 2, 3, 4$	64
5.2	Oblast stability explicitní Taylorova řada $ORD = 5, 6$	64
5.3	Oblast stability implicitní Taylorova řada $ORD = 2$	65
5.4	Oblast stability implicitní Taylorova řada $ORD = 3$	65
5.5	Oblast stability implicitní Taylorovy řady $ORD = 4$	65
5.6	Oblast stability implicitní Taylorovy řady $ORD = 5$	65
5.7	Oblast stability impl. Taylorovy řady $ORD = 11$ v blízkosti osy $Im(z)$	66
5.8	Oblast stability impl. Taylorovy řady $ORD = 14$ v blízkosti osy $Im(z)$	66
5.9	Absolutní chyba výpočtu, explicitní Taylorova řada $ORD = 1, 2, 3$	66
5.10	Absolutní chyba výpočtu, implicitní Taylorova řada $ORD = 1, 2, 3$	66
5.11	Absolutní chyba výpočtu, explicitní Taylorova řada $ORD = 63$	67
5.12	Absolutní chyba výpočtu, implicitní Taylorova řada $ORD = 63$	67
5.13	$R_I(z)$ (5.17)	68
5.14	$R_I(z)$ (5.17) - detail	68

5.15	Řešení problému (5.20) - implicitní numerické metody, $h = 0.1$	70
5.16	Členy explicitní Taylorovy řady DY ($\lambda = -1, h = 1$)	72
5.17	Členy explicitní Taylorovy řady DY ($\lambda = -10, h = 1$)	72
5.18	Členy explicitní Taylorovy řady DY ($\lambda = -100, h = 1$)	73
5.19	Explicitní Eulerova metoda - absolutní chyba výpočtu v $TMAX = 2$	74
5.20	Implicitní Eulerova metoda - absolutní chyba výpočtu v $TMAX = 2$	74
5.21	Explicitní Taylorova metoda, absolutní chyba výpočtu v $TMAX = 2$	76
5.22	Řešení problému (5.40)	85
5.23	Absolutní chyba výpočtu explicitní Taylorovy metody v $TMAX = 10$	100
5.24	Absolutní chyba výpočtu ITMRN v $TMAX = 10$	100
5.25	ITMRN - počet Newt. iterací pro $ORD = 1, 2, 3$	101
5.26	ITMRN - počet Newt. iterací v závislosti na h_N	101
A.1	Řešení TKSL/386	118
A.2	Řešení TKSL/386	120
B.1	RC elektrický obvod, napětí u_C - MATLAB (ODE45): $a = 10^5, \omega = 1$	131
B.2	RLC elektrický obvod - TKSL/386: $ERR = u_C - u_2, TMAX = 10$	132
B.3	RLC elektrický obvod - TKSL/386: $ERR = u_C - u_2, TMAX = 20$	132
B.4	RLC elektrický obvod - TKSL/386: $ERR = u_C - u_2, TMAX = 50$	133
B.5	Oblast stability explicitní Taylorovy řady $ORD = 63$	133
B.6	Oblast stability implicitní Taylorovy řady $ORD = 63$	133

Seznam tabulek

2.1	Koeficienty Adams-Bashfortovy metody	25
2.2	Koeficienty Adams-Moultonovy metody	26
2.3	Koeficienty metod zpětného derivování	27
3.1	Velikosti úhlu α u metod zpětného derivování	33
3.2	Funkce stabilit: $R(z) = y_{i+1}/y_i$	34
3.3	Padé aproximace pro funkci e^z	35
4.1	Průměrná absolutní chyba výpočtu - explicitní Eulerova metoda	46
4.2	Maximální velikost h ($EPS = 10^{-20}$, $a = 2000$) - explicitní numerické metody	48
4.3	Impl. lichoběžníková metoda (4.43), $h = 0.25$, $TOL = 10^{-6}$	56
4.4	Impl. lichoběžníková metoda (4.44), $h = 0.25$, $TOL = 10^{-6}$, $h_N = 0.01$. . .	56
4.5	Impl. lichoběžníková metoda (4.44), $h = 0.25$, $TOL = 10^{-6}$, $h_N = 0.001$. .	57
4.6	Impl. lichoběžníková metoda (4.44), $h = 0.25$, $TOL = 10^{-6}$, $h_N = 0.0001$. .	57
5.1	Implicitní Taylorova řada - absolutní chyba výpočtu, $h = 0.1$, $z = -2.5$. .	67
5.2	Implicitní Taylorova řada - absolutní chyba výpočtu v prvním kroku $\varepsilon_I(z)$.	68
5.3	Maximální velikost h , expl. numerické metody, chyba výpočtu $EPS = 10^{-20}$	69
5.4	Implicitní metody - absolutní chyba výpočtu v prvním kroku v závislosti na h	69
5.5	Impl. Taylorova řada a Newtonova metoda - absolutní chyba výpočtu v prvním kroku $h = 1$, $\lambda = -100$, $h_N = 0.01$, $TOL = 10^{-15}$	71
5.6	Absolutní chyba výpočtu - expl. Eulerova metoda, $h = 0.1$	75
5.7	Absolutní chyba výpočtu - explicitní Taylorova metoda, $h = 0.1$, $L = -10000$	77
5.8	Absolutní chyba výpočtu, explicitní Taylorova metoda, $h = 0.1$, $ORD = 8$.	78
5.9	Absolutní chyba výpočtu, explicitní Taylorova metoda, $h = 0.5$, $ORD = 8$.	78
5.10	Absolutní chyba výpočtu, implicitní Taylorova metoda, $h = 0.1$, $ORD = 8$.	79
5.11	Absolutní chyba výpočtu, implicitní Taylorova metoda, $h = 0.5$, $ORD = 8$.	79
5.12	Absolutní chyba výpočtu, implicitní Taylorova metoda (analyticky vyjádřená Jacobiova matice), $h = 0.1$, $ORD = 8$, $TOL = 10^{-20}$	81
5.13	Absolutní chyba výpočtu, implicitní Taylorova metoda (analyticky vyjádřená Jacobiova matice), $h = 0.5$, $ORD = 8$, $TOL = 10^{-10}$	81
5.14	Absolutní chyba výpočtu, implicitní Taylorova metoda (numericky vypočtená Jacobiova matice), $h = 0.1$, $h_N = 0.1$, $ORD = 8$, $TOL = 10^{-20}$	83
5.15	Absolutní chyba výpočtu, implicitní Taylorova metoda (numericky vypočtená Jacobiova matice), $h = 0.5$, $h_N = 0.1$, $ORD = 8$, $TOL = 10^{-10}$	83
5.16	Explicitní numerické metody - maximální velikost h , $EPS = 1 \times 10^{-15}$. . .	84
5.17	Absolutní chyba výpočtu v prvním kroku - implicitní Taylorova metoda . .	85
5.18	Abs. chyba výpočtu ITMRN: $h = 0.25$, $h_N = 0.1$, $ORD = 5$, $TOL = 10^{-10}$	89
5.19	Abs. chyba výpočtu ITMRN: $h = 0.5$, $h_N = 0.1$, $ORD = 5$, $TOL = 10^{-10}$.	89

5.20	Abs. chyba výpočtu ITMRN: $h = 0.5, h_N = 0.1, ORD = 10, TOL = 10^{-10}$	89
5.21	Absolutní chyba výpočtu ITMRN: $h = 1.5, h_N = 0.1, TOL = 10^{-10}$	90
5.22	Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 0.1, ORD = 1$	91
5.23	Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 0.1, a = 100$.	91
5.24	Explicitní Taylorova řada - víceslovní aritmetika	92
5.25	Absolutní chyba výpočtu - implicitní Taylorova metoda: $h = 0.1$	93
5.26	Absolutní chyba výpočtu ITMRN: $h = 0.1, ORD = 5, 6, 7, 8, TOL = 10^{-10},$ $a = 10^4$	95
5.27	Absolutní chyba výpočtu ITMRN: $h = 0.05, ORD = 8, TOL = 10^{-10}, a = 10^4$	96
5.28	Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 10^{-12}$	97
5.29	Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 10^{-11}$	97
5.30	Absolutní chyba výpočtu - impl. Taylorova metoda: $h = 10^{-5}, TMAX = 0.1$	97
5.31	Absolutní chyba výpočtu - impl. Taylorova metoda: $h = 10^{-4}, TMAX = 0.1$	97
5.32	Absolutní chyba výpočtu ITMRN: $h = 10^{-5}, TMAX = 0.1, h_N = 0.1,$ $TOL = 10^{-10}$	98
5.33	Absolutní chyba výpočtu ITMRN: $h = 10^{-4}, TMAX = 0.1, h_N = 0.1,$ $TOL = 10^{-10}$	98
5.34	Absolutní chyba výpočtu ITMRN s víceslovní aritmetikou: $h = 10^{-2}, TMAX =$ $1, h_N = 0.1, TOL = 10^{-10}$	99
5.35	Maximální velikost h , explicitní numerické metody, $EPS = 1 \times 10^{-5}$	100
5.36	Absolutní chyba výpočtu v prvním kroku ITMRN: $h = 1, TOL = 10^{-10}$. .	102

Symbols and abbreviations

\mathcal{J}	Jacobiova matice
h	integrační krok
h_N	krok v diferenční formuli použitý u Newtonovy metody
ITMRN	kombinace implicitní Taylorovy metody s rekurentním výpočtem členů a Newtonovou metodou
<i>EPS</i>	“ <i>Error per step</i> ” nastavitelná chyba výpočtu počítána rozdílem řádu metody
<i>ORD</i>	řád Taylorovy řady
ODR	obyčejná diferenciální rovnice
SDR	soustava diferenciálních rovnic, jedná se o počáteční úlohy
SLAR	soustava lineárních algebraických rovnic
<i>TOL</i>	nastavitelná chyba výpočtu u Newtonovy metody
<i>TMAX</i>	maximální čas udávající délku intervalu hledaného numerického řešení
⊙	konec příkladu

Kapitola 1

Úvod

Historický vznik diferenciálního počtu sahá až do 17. století, kdy jej definovali Isaac Newton (pomocí geometrické interpolace) a Gottfried Leibnitz (pomocí limity). V minulém století století především s rozvojem výpočetní techniky se vyvíjelo mnoho numerických metod využívajících prostředků diferenciálního počtu.

Celý život se snažíme porozumět chování různých systémů. Často používáme k poznání systému metodu pokusů a omylů, která je však časově i finančně náročná. Volí se tedy cesta popisu systému jiným (často zjednodušeným) systémem. Ve 20. století se s rozšířením počítačů začalo simulovat mnoho spojitých modelů (chemické reakce, elektrické obvody, biologie, ...). Většinou tyto dynamické systémy lze popsat soustavou obyčejných diferenciálních rovnic, která je jako jediná schopna přesně definovat chování systému v libovolném časovém okamžiku.

Soustavy diferenciálních rovnic jsou v praxi řešeny běžnými numerickými metodami. Nejčastěji se používají metody Runge-Kutta, Eulerova metoda, nebo metody typu prediktor-korektor. V práci prezentovaná metoda využívající Taylorovy řady se od běžných metod liší možnostmi výpočtu vyšších derivací a jejich následným využitím při výpočtu. Tato metoda umožňuje změnu integračního kroku během výpočtu, zadání přesnosti výpočtu, možnost využití víceslovní aritmetiky a další strategie.

Mnoho technických problémů svojí specifikou vyžaduje zvolení vhodné metody řešení diferenciálních rovnic. Vyžadují většinou přesné a rychlé řešení. U některých problémů jsme však nuceni upřednostnit rychlost výpočtu oproti přesnosti řešení.

Některé zvláštní soustavy diferenciálních rovnic nelze řešit pomocí běžných (zejména explicitních) numerických metod. Těmto soustavám říkáme *tuhé soustavy* diferenciálních rovnic (“Stiff systems of differential equations”).

Předkládaná práce se zabývá analýzou a řešením tuhých soustav diferenciálních rovnic, tedy problémy, které často vyžadují velmi malý integrační krok a výpočet je i přes využití nejmodernější výpočetní techniky časově velmi náročný.

1.1 Motivace, cíle disertační práce

Řešení tuhých (“stiff”) soustav diferenciálních rovnic patří i v současné době stále mezi problémové. Základním problémem je **přesná definice tuhých systémů**, která stále neexistuje.

Problémem je dále i **detekce tuhých systémů diferenciálních rovnic**. K detekci tuhých systémů se často v praxi využívá jedna z definic tuhých systémů pomocí *koeficientu*

tuhosti soustavy, který je dán poměrem maximální a minimální absolutní hodnoty reálné části vlastního čísla Jacobiovy matice řešené soustavy. Výpočet Jacobiovy matice u rozsáhlých soustav diferenciálních rovnic je výpočetně i časově náročný. K řešení tuhých systémů diferenciálních rovnic se v praxi často využívají implicitní numerické metody nižších řádů, které nám poskytují větší oblasti stability řešení (tedy možnost volby většího integračního kroku) na úkor přesnosti numerického řešení.

Ve své práci se budu zabývat numerickým řešením obyčejných diferenciálních rovnic, především numerickým výpočtem využívajícím metody Taylorovy řady, která poskytuje řešení s vysokou přesností výpočtu. Budu analyzovat vznik tuhosti na konkrétních systémech diferenciálních rovnic. Budu zkoumat nové možnosti detekce tuhých soustav diferenciálních rovnic (SDR) pomocí Taylorovy řady. Zaměřím se na nalezení nových přístupů vedoucích k odstranění tuhosti ze soustav diferenciálních rovnic. Prozkoumám stabilitu explicitní a implicitní Taylorovy řady a ukáži vhodnost použití obou metod na konkrétních problémech. V praxi se často u implicitních numerických metod využívá k výpočtu Newtonova metoda, a proto se v závěru práce zaměřím na možnost řešení vybraných problémů pomocí implicitní Taylorovy řady využívající Newtonovu metodu.

Hlavní cíle předložené práce tak můžeme shrnout do dvou bodů:

- Analyzovat konkrétní problémy a vznik tuhosti v těchto soustavách, zkoumat možnost nalezení ekvivalentního systému “bez tuhosti” k původní tuhé soustavě diferenciálních rovnic, ukázat možnou detekci tuhých systémů pomocí členů explicitní Taylorovy metody.
- Zkoumat na vybraných úlohách možnosti výpočtu pomocí explicitní a implicitní Taylorovy metody se zaměřením na konvergenci a stabilitu uvedených metod. Při implementaci se zaměřit na vliv víceslovní aritmetiky a navrhnout vhodný algoritmus (výpočet členů Taylorovy řady) pro budoucí možnou paralelizaci výpočtu.

1.2 Struktura práce

Práce je strukturována do šesti kapitol. Úlohou první kapitoly je uvést stručně do problematiky, vysvětlit motivaci a cíle disertační práce.

Druhá kapitola se věnuje především problematice numerického řešení soustav diferenciálních rovnic (SDR)¹. V úvodu je provedeno seznámení s matematickými pojmy, které se v práci budou následně vyskytovat. Důraz je kladen na přehled numerických metod, které se v praxi využívají k řešení soustav obyčejných diferenciálních rovnic. Cílem této kapitoly je uvést do problematiky numerického řešení diferenciálních rovnic.

Třetí kapitola se zaměřuje na problematiku tuhých systémů diferenciálních rovnic ve spojení se stabilitou numerických metod. Cílem této kapitoly je vysvětlit rozdělení numerických metod z hlediska stability a vhodnosti jejich následného použití k numerickému řešení tuhých soustav diferenciálních rovnic.

Čtvrtá kapitola se věnuje analýze a řešení tuhých systémů. Úvodem je prezentována nová možnost detekce tuhých systémů pomocí členů explicitní Taylorovy řady. Dále zde navrhuji nový možný způsob odstraňování tuhosti v “tuhých” soustavách diferenciálních

¹Pokud se v práci budeme následně setkávat s pojmem “SDR”, jedná se přesněji o pojem *počáteční úlohy pro diferenciální rovnice*.

rovnice převodem tuhých systémů diferenciálních rovnic na nové ekvivalentní systémy “bez tuhosti”, které lze řešit běžnými explicitními numerickými metodami. Je proveden rozbor vybraných tuhých systémů v elektrotechnice. Následuje ukázka možnosti řešení soustav lineárních algebraických rovnic pomocí transformace na soustavu diferenciálních rovnic a s ní spojený možný vznik tuhosti. Závěrem je demonstrována aplikace Newtonovy metody na konkrétním příkladu s porovnáním možného analytického a numerického řešení derivace. Cílem této kapitoly je seznámit čtenáře pomocí vybraných problémů se vznikem tuhých systémů diferenciálních rovnic a možnostmi jejich detekce a možnostmi jejich řešení.

Pátá kapitola se soustřeďuje na řešení tuhých systému diferenciálních rovnic metodou Taylorovy řady. Úvodem je provedeno seznámení s možnostmi rekurentního výpočtu členů Taylorovy řady pomocí tvořících rovnic. Následně je zkoumána stabilita a konvergence explicitní a implicitní Taylorovy řady. Na vybraných problémech (soustavách diferenciálních rovnic) je pak demonstrována možnost řešení pomocí nově implementované implicitní Taylorovy metody s rekurentním výpočtem členů a Newtonovou metodou (ITMRN).

Šestá kapitola shrnuje dosažené výsledky disertační práce, hodnotí splnění cílů a diskutuje možnosti dalšího výzkumu.

Kapitola 2

Řešení diferenciálních rovnic

Úlohou této kapitoly je uvést do problematiky numerického výpočtu diferenciálních rovnic. Nejedná se o podrobné proniknutí do problematiky, pouze o stručný přehled základních pojmů se kterými se budeme setkávat v následujících částech práce.

2.1 Matematické základy

Seznámíme se stručně se základními pojmy Jacobiova matice a vlastní čísla, které si demonstrováme na jednoduchém příkladě. Následně se zaměříme na možnost numerického řešení soustav lineárních algebraických rovnic (SLAR) pomocí převodu na soustavu diferenciálních rovnic (SDR). Zmíníme možnost numerického řešení soustav nelineárních rovnic pomocí Newtonovy metody.

Jde pouze o stručný přehled, podrobnější vysvětlení a formální definice všech pojmů nalezneme např. v [KDJ05, Č04, vH05, Dal97, Vit94, Bar06, BF04, Mat10a].

2.1.1 Jacobiova matice

Jacobiova matice (resp. její determinant) se využívá k určení, zda je dané zobrazení regulární nebo singulární. Jacobiova matice však také spolu s vlastními čísly umožňuje určit “tuhost” systému diferenciálních rovnic (viz Def. 4.1.2).

Definice 2.1.1 *Mějme funkce $f_i(x_1, x_2, \dots, x_k)$ pro $i = 1, 2, \dots, n$, které mají parciální derivace $\frac{\partial f_i}{\partial x_j}$. Pak Jacobiova matice je definována jako*

$$\frac{D(f_1, f_2, \dots, f_n)}{D(x_1, x_2, \dots, x_k)} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \frac{\partial f_1}{\partial x_k} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \frac{\partial f_2}{\partial x_k} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \frac{\partial f_n}{\partial x_k} \end{pmatrix}. \quad (2.1)$$

Jacobiova matice se obvykle značí zápisem $\frac{D(f_1, f_2, \dots, f_n)}{D(x_1, x_2, \dots, x_k)}$. My však budeme v následujícím textu pro jednoduchost používat značení \mathcal{J} . Je-li matice \mathcal{J} čtvercová, tedy $n = k$, nazýváme její determinant ($\det \mathcal{J}$) - Jacobiho determinantem (zkráceně *Jakobiánem*).

Příklad výpočtu Jacobiovy matice

Mějme soustavu dvou diferenciálních rovnic prvního řádu

$$\begin{aligned}y' &= z, & y(0) &= 1, \\z' &= -a \cdot y - (a+1) \cdot z, & z(0) &= -1,\end{aligned}\tag{2.2}$$

z této soustavy chceme zjistit Jacobiovu matici \mathcal{J} .

Uřčíme si potřebné parciální derivace

$$\begin{aligned}\frac{\partial y'}{\partial y} &= 0, & \frac{\partial y'}{\partial z} &= 1, \\ \frac{\partial z'}{\partial y} &= -a, & \frac{\partial z'}{\partial z} &= -a-1,\end{aligned}$$

a sestrojíme Jacobiovu matici

$$\mathcal{J} = \begin{pmatrix} 0 & 1 \\ -a & -a-1 \end{pmatrix}.$$

◊

2.1.2 Vlastní čísla

S pojmem *vlastní číslo* se v matematice setkáme např. při řešení soustav lineárních diferenciálních rovnic 1. řádu s konstantními koeficienty, při testování stability nelineárních autonomních systémů atd. Vlastní čísla se ovšem také využívají k detekci tuhých systémů diferenciálních rovnic (viz Def. 4.1.2).

Definice 2.1.2 *Vlastním číslem čtvercové matice \mathbf{A} se nazývá každé (obecně komplexní) číslo λ , splňující pro některý nenulový vektor \vec{x} rovnost*

$$\mathbf{A} \cdot \vec{x} = \lambda \cdot \vec{x}.\tag{2.3}$$

Nenulový vektor \vec{x} , který splňuje rovnici (2.3), nazýváme vlastním vektorem matice \mathbf{A} příslušným k vlastnímu číslu λ .

Rovnici (2.3) lze postupnými úpravami převést do tvaru (2.4).

$$\begin{aligned}\mathbf{A} \cdot \vec{x} - \lambda \cdot \vec{x} &= 0, \\ \mathbf{A} \cdot \vec{x} - \lambda \cdot \mathbf{E} \cdot \vec{x} &= 0, \\ (\mathbf{A} - \lambda \cdot \mathbf{E}) \cdot \vec{x} &= 0.\end{aligned}\tag{2.4}$$

Jedná se tedy o homogenní soustavu lineárních algebraických rovnic, která má nenulové řešení právě tehdy, když $\det(\lambda \cdot \mathbf{E} - \mathbf{A}) = 0$, což lze rozepsat

$$\begin{vmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{vmatrix} = 0.\tag{2.5}$$

Rovnice (2.5) se nazývá *charakteristická rovnice*. Polynom proměnných stupně n na levé straně se nazývá *charakteristický polynom* matice \mathbf{A} a jeho kořeny jsou vlastními čísly.

Každá čtvercová matice n -tého řádu má tedy právě n vlastních čísel, z nichž některá se mohou opakovat jakožto *násobná vlastní čísla*.

Příklad výpočtu vlastních čísel matice

Mějme Jacobiovu matici \mathcal{J} ze soustavy diferenciálních rovnic (2.2)

$$\mathcal{J} = \begin{pmatrix} 0 & 1 \\ -a & -a-1 \end{pmatrix}, \quad (2.6)$$

pro tuto matici chceme zjistit vlastní čísla.

Vypočítáme kořeny charakteristického polynomu

$$\det(\mathbf{A} - \lambda \cdot \mathbf{E}) = 0,$$

$$\begin{aligned} \begin{vmatrix} -\lambda & 1 \\ -a & -a-1-\lambda \end{vmatrix} &= -\lambda(-a-1-\lambda) - (-a) = \\ &= a\lambda + \lambda + \lambda^2 + a = \lambda^2 + (a+1)\lambda + a = 0, \end{aligned}$$

spočítáme kořeny kvadratické rovnice

$$\lambda^2 + (a+1)\lambda + a = 0,$$

$$\begin{aligned} D &= (a+1)^2 - 4 \cdot a = a^2 + 2 \cdot a + 1 - 4 \cdot a = \\ &= a^2 - 2 \cdot a + 1 = (a-1)^2, \\ \lambda_1 &= \frac{-(a+1)+a-1}{2} = \underline{\underline{-1}}, \\ \lambda_2 &= \frac{-(a+1)-(a-1)}{2} = \underline{\underline{-a}}, \end{aligned}$$

vlastní čísla matice \mathcal{J} jsou tedy $\lambda_1 = -1$, $\lambda_2 = -a$.

⊙

2.1.3 Numerické řešení lineárních rovnic

Jedním z netradičních způsobů řešení soustav lineárních algebraických rovnic (SLAR) je numerické řešení pomocí převodu na soustavu obyčejných diferenciálních rovnic (SDR).

Mějme SLAR ve tvaru

$$\mathbf{A} \cdot \vec{x} = \vec{b}, \quad (2.7)$$

k ní vytvoříme odpovídající soustavu diferenciálních rovnic (počáteční úlohu) ve tvaru

$$\mathbf{A} \cdot \vec{x} - \vec{b} = \vec{x}', \quad (2.8)$$

kde počáteční podmínky definujeme $\vec{x}(0) = \vec{0}$, derivace \vec{x}' je počítána k jedné proměnné (času t).

Hledáme tedy řešení SLAR (2.7) pomocí SDR (2.8). Provádíme numerický výpočet SDR (2.8), dokud nenalezneme ustálený stav, tedy dvě následující hodnoty řešení jsou shodné $\vec{x}_i = \vec{x}_{i+1}$ (tzn. derivace jsou nulové $\vec{x}'_i = \vec{x}'_{i+1} = \vec{x}'_{i+2} = \dots = \vec{0}$).

U výše popsaného elementárního převodu SLAR na SDR se však můžeme setkat s nestabilitou systému (tzn. alespoň pro jedno vlastní číslo systému platí $\Re(\lambda) > 0$). Jednou z možností zajištění stability výpočtu vzniklé SDR je využití transponované matice.

Definice 2.1.3 Má-li matice $\mathbf{A} \in \mathbb{R}^{m \times n}$ prvky a_{ij} , $i = 1, \dots, m$, $j = 1, \dots, n$, pak matici $\mathbf{A}^T \in \mathbb{R}^{n \times m}$ takovou, že

$$a_{ji}^T = a_{ij} \text{ pro } j = 1, \dots, n, i = 1, \dots, m, \quad (2.9)$$

nazýváme matici transponovanou k matici \mathbf{A} .

Mějme nestabilní SDR ve tvaru (2.8). Pro dosažení stabilního řešení vynásobíme původní SLAR (2.7) zápornou transponovanou maticí zleva a vytvoříme tak nový systém diferenciálních rovnic ve tvaru

$$-\mathbf{A}^T \cdot \mathbf{A} \cdot \vec{x} + \mathbf{A}^T \cdot \vec{b} = \vec{x}'. \quad (2.10)$$

Je-li matice \mathbf{A} regulární (což je základní podmínka řešení soustavy rovnic), pak matice $\mathbf{A}^T \cdot \mathbf{A}$ je pozitivně definitní. Kořeny charakteristické rovnice (vlastní čísla) takto upraveného systému se počítají podle vztahu

$$|-\mathbf{A}^T \cdot \mathbf{A} - \lambda \cdot \mathbf{E}| = 0. \quad (2.11)$$

Systém (2.10) je tedy pozitivně definitní a následkem této vlastnosti má také záporná vlastní čísla (2.11). Je tedy vždy stabilní. Následkem této transformace se ale zvyšuje “tuhost” systému. Více informací o konvergenci numerických metod a stabilitě řešení SDR s využitím transponované matice nalezneme v [Seh07]. Příklad vzniku tuhosti při převodu SLAR na SDR rozebereme v kapitole 4.4.

2.1.4 Numerické řešení nelineárních rovnic

Při řešení tuhých systémů diferenciálních rovnic se využívají převážně implicitní numerické metody. Tyto metody ze své implicitní podstaty musí navíc numericky (iteračně) řešit soustavu nelineárních rovnic.

Pro zjednodušení se zaměříme na obecnou iterační metodu pro řešení pouze jedné rovnice o jedné neznámé. Hledáme tedy řešení rovnice

$$f(x) = 0. \quad (2.12)$$

Předpokládáme, že v dostatečně malém intervalu $\langle a, b \rangle$ leží jediný kořen $x = x^*$ rovnice (2.12). Zvolíme počáteční bod x_0 (blízký x^*) ležící v daném intervalu. Dále obecně postupujeme podle rekurentního předpisu

$$x_k = \phi_k(x_0, x_1, x_2, \dots, x_{k-1}), \quad (2.13)$$

kde posloupnost x_0, x_1, \dots díky vhodným funkcím ϕ_k (závisí na $f(x)$) konverguje k x^* . Obvykle volíme tento “nejjednodušší” rekurentní předpis

$$x_k = \phi(x_{k-1}), \quad k = 1, 2, \dots, \quad (2.14)$$

tak, aby $x^* = \lim_{k \rightarrow \infty} (x_k)$.

Často bývá funkce $\phi(x)$ diferencovatelná. Platí-li

$$|\phi'(x^*)| \leq K < 1 \quad (2.15)$$

a je-li ϕ' spojitá, pak postupné aproximace (2.14) budou konvergovat, zvolíme-li x_0 v blízkosti x^* . Konvergence řady x_n je tím rychlejší, čím je menší K .

Chceme-li získat výsledek x^* s danou chybou výpočtu ϵ , můžeme použít tohoto kritéria pro ukončení iteračního procesu (2.14):

$$\frac{K}{1-K}|x_k - x_{k-1}| < \epsilon \quad (2.16)$$

Iterace (2.14) je řádu m pokud platí

$$\phi'(x^*) = \phi''(x^*) = \dots = \phi^{(m-1)}(x^*) = 0, \quad \phi^{(m)}(x^*) \neq 0. \quad (2.17)$$

Existuje-li m spojitých derivací funkce $\phi(x)$ v okolí bodu x^* , pak z Taylorova rozvoje

$$\begin{aligned} \phi(x) - x^* &= (x - x^*)\phi'(x^*) + \frac{1}{2!}(x - x^*)^2\phi''(x^*) + \dots + \\ &+ \frac{1}{m-1!}(x - x^*)^{m-1}\phi^{(m-1)}(x^*) + \frac{1}{m!}(x - x^*)^m\phi^{(m)}(\xi), \end{aligned} \quad (2.18)$$

pro iteraci řádu m a pro $x = x_{k-1}$ platí tedy

$$x_k - x^* = \frac{1}{m!}(x_{k-1} - x^*)^m\phi^{(m)}(\xi_k). \quad (2.19)$$

Newtonova metoda

Jednou z nejčastěji používaných metod pro řešení nelineárních rovnic je Newtonova metoda. Řešíme iterační rovnici

$$x = x - \frac{f(x)}{f'(x)} = \phi(x), \quad (2.20)$$

která má stejný kořen x^* jako rovnice (2.12). Předpokládáme, že v dostatečně malém intervalu $\langle a, b \rangle$ leží jediný kořen $x = x^*$ rovnice (2.20) a funkce f má spojitě nenulové derivace f', f'' . Lze odvodit

$$\phi'(x) = \frac{f(x)f''(x)}{(f'(x))^2}. \quad (2.21)$$

Z $f(x^*) = 0$, plyne $\phi'(x^*) = 0$, takže Newtonova metoda je druhého řádu. Jelikož platí $|\phi'(x^*)| < 1$, platí v některém okolí bodu x^* , že budou postupně aproximace

$$x_{j+1} = x_j - \frac{f(x_j)}{f'(x_j)}, \quad j = 0, 1, 2, \dots, \quad (2.22)$$

konvergovat k x^* , zvolíme-li x_0 z dostatečně malého okolí řešení x^* .

Pro mnohé případy je analytické vyjádření $f'(x)$ složité případně nemožné. Proto lze použít diferenční formule, kdy např. aproximujeme

$$f'(x_j) \doteq \frac{f(x_j + h_N) - f(x_j)}{h_N}, \quad (2.23)$$

pro vhodně malé h_N .

Při řešení soustavy nelineárních rovnic pomocí Newtonovy metody je nutno vždy vy počítat Jacobiovu matici soustavy a řešit soustavu rovnic

$$\mathcal{J}(\mathbf{x}_j)(\mathbf{x}_{j+1} - \mathbf{x}_j) + \mathbf{f}(\mathbf{x}_j) = \mathbf{0}. \quad (2.24)$$

Rovnici (2.24) lze přepsat do tvaru

$$\mathbf{x}_{j+1} = \mathbf{x}_j - \mathcal{J}^{-1}(\mathbf{x}_j)\mathbf{f}(\mathbf{x}_j). \quad (2.25)$$

Newtonova metoda se často využívá u implicitních numerických metod řešení diferenciálních rovnic. Ukázkou použití Newtonovy metody při řešení tuhých systémů diferenciálních rovnic nalezneme v podkapitole 4.5 a v podkapitole 5.6.

2.2 Numerické řešení obyčejných diferenciálních rovnic

Nalezení analytického řešení rozsáhlých soustav diferenciálních rovnic je složité, mnohdy nemožné. V současné době se s rozmachem výpočetní techniky čím dál častěji využívá numerického řešení. Zaměříme se tedy na problém numerického řešení obyčejných diferenciálních rovnic s počáteční podmínkou.

2.2.1 Základní pojmy

Počáteční problém pro ODR. *Obyčejná diferenciální rovnice* (ODR) se nazývá rovnice *n-tého řádu* (ODR_n), jestliže neznámá funkce je funkcí jedné proměnné a její nejvyšší derivace neznámé funkce je *n-tého řádu*. V práci se budeme zabývat především diferenciálními rovnicemi prvního řádu. *Obecný tvar* diferenciální rovnice prvního řádu je

$$g(t, y(t), y'(t)) = 0. \quad (2.26)$$

Předpokládejme, že rovnici (2.26) lze také vyjádřit explicitně ve tvaru

$$y'(t) = f(t, y(t)). \quad (2.27)$$

Takovéto obecné řešení diferenciální rovnice obsahuje integrační konstantu, která může nabývat libovolné hodnoty. Proto k jednoznačnému určení $y(t)$ musíme ještě doplnit hodnotu funkce v určitém bodě $t = t_0$, tedy *počáteční podmínku*

$$y(t_0) = y_0. \quad (2.28)$$

Rovnice (2.27) spolu s počáteční podmínkou (2.28) se nazývá *počáteční úloha*, nebo také *Cauchyova úloha*.

Numerické řešení počáteční úlohy. Pod tímto pojmem si představíme nalezení přibližných hodnot funkce $y(t)$ v bodech t_i pokrývajících dostatečně hustě určitý interval $\langle a, b \rangle$. Hledáme tedy řešení v jednotlivých bodech $a = t_0 < t_1 < \dots < t_k = b$. Množinu bodů t_i , $i \in \langle 0, k \rangle$ nazýváme *síť*, její prvky *uzly sítě* a vzdálenost dvou sousedních uzlů $t_{i+1} - t_i = h_i$ nazýváme *délkou kroku sítě* v uzlu t_i , nebo také *integračním krokem*. Mají-li navíc všechny kroky stejnou délku $h = \frac{b-a}{k}$, mluvíme o *pravidelné síti*.

Budeme předpokládat, že každá Cauchyova úloha, o jejímž numerickém řešení budeme uvažovat, má jediné řešení. Je dobře známo, že tato podmínka **existence a jednoznačnosti řešení** je splněna, když je funkce $f(t, y(t))$ spojitá, ohraničená a splňuje Lipschitzovu podmínku.

Definice 2.2.1 *O funkci f řekneme, že splňuje v bodě (t_0, y_0) Lipschitzovu podmínku, pokud existují konstanta L a okolí U bodu (t_0, y_0) tak, že*

$$|f(t, u) - f(t, v)| \leq L|u - v|, \quad \forall (t, u), (t, v) \in U. \quad (2.29)$$

Vypočtenou aproximaci hodnoty přesného řešení $y(t_i)$ v uzlu t_i budeme značit y_i . Má-li být **numerická metoda** řešení použitelná, je nutné, aby vypočtené hodnoty y_i konvergovaly k hodnotám $y(t_i)$ pro $h \rightarrow 0$. Konvergencí zde rozumíme skutečnost, že $\lim_{i \rightarrow \infty} (y_i) = y(t)$ pro $h \rightarrow 0$ a $i \rightarrow \infty$, přičemž $h \cdot i = T$. Konverguje-li řešení získané

numerickou metodou pro danou diferenciální rovnici (2.27) s počáteční podmínkou (2.28), nazýváme tuto metodu *konvergentní*.

Numerická metoda pro řešení počáteční úlohy (2.27), (2.28) je tedy předpis pro postupný výpočet aproximací $y_1, y_2, \dots, y_k, y_0 = konst.$ Metody můžeme rozlišit podle způsobu výpočtu na

1. **metody jednokrokové** využívají k výpočtu aproximace y_{i+1} pouze informaci y_i pro $i = 0, 1, \dots$
2. **metody vícezkrokové** (k -zkrokové), používají pro výpočet aproximace y_{i+1} předchozí vypočtené hodnoty $y_i, y_{i-1}, \dots, y_{i+1-k}$.

Při zkoumání jednotlivých numerických metod pro řešení konkrétních soustav diferenciálních rovnic se setkáme s pojmem **chyba numerických metod**. Většinou se mluví o lokálních diskretizačních chybách (“local truncation error”) a globálních diskretizačních chybách (“global truncation error”).

Lokální diskretizační chyba je chyba, které se dopustíme v jednom kroku metody. Vzniká následkem zanedbání vyšších členů Taylorovy řady viz Def. 2.2.2. Většina známých numerických metod využívá k výpočtu pouze první derivace.

Globální diskretizační chyba (celková chyba) je důsledkem hromadění účinků vzniklých chyb lokálních. Lze jí vyjádřit vztahem $\varepsilon_n = y(t_n) - y_n$. Kromě chyb metody musíme brát v úvahu i **zaokrouhlovací chyby**, které jsou zapříčiněny omezenou délkou slova v počítači, ve kterém je uložena hodnota čísla.

Dalším pojmem, který je úzce spjat s chybou metody, je **stabilita metody**. Řekneme, že metoda je *absolutně stabilní*, pokud pro daný integrační krok h a danou diferenciální rovnici chyba vzniklá při výpočtu y_n se nezvětší v následujících hodnotách $y_k, k > n$. Tedy je zaručeno, že aproximace y_i konverguje k hodnotě přesného řešení $y(t_i)$. Podrobněji se stabilitou numerických metod budeme zabývat v podkapitole 3.1.

Pro jednoduchost analýzy se v následujících kapitolách uvádí jedna diferenciální rovnice prvního řádu s jednou neznámou funkcí. Všechny pojmy však samozřejmě platí také pro **soustavy diferenciálních rovnic 1. řádu**, tedy n rovnic o n neznámých funkcích $y_1(t), y_2(t), \dots, y_n(t)$ tvaru

$$y'_j(t) = f_j(t, y_1(t), y_2(t), \dots, y_n(t)), \quad j = 1, 2, \dots, n. \quad (2.30)$$

Počáteční podmínky jsou tvaru

$$y_j(t_0) = y_{j,0}, \quad j = 1, 2, \dots, n. \quad (2.31)$$

Existují ovšem také **diferenciální rovnice n-tého řádu** ve tvaru

$$y^{(n)}(t) = F(t, y(t), y'(t), \dots, y^{(n-1)}(t)). \quad (2.32)$$

s počátečními podmínkami

$$y(t_0) = y_{1,0}, \quad y'(t_0) = y_{2,0}, \quad \dots, \quad y^{(n-1)}(t_0) = y_{n,0}. \quad (2.33)$$

Tyto rovnice lze však převést na soustavu diferenciálních rovnic 1. řádu (2.30) s počátečními podmínkami (2.31). Stačí položit $y_1 = y, y_2 = y', \dots, y_n = y^{(n-1)}$, poté $f_j = y_{j+1}, j = 1, 2, \dots, n-1$ a $f_n = F(t, y_1(t), y_2(t), \dots, y_n(t))$.

2.2.2 Jednokrokové metody

Jednokrokové metody se vyznačují tím, že přibližné řešení y_{i+1} v uzlu t_{i+1} se počítá ze vztahu, v němž kromě neznámé y_{i+1} vystupuje předchozí uzel t_i , vypočtená hodnota y_i v tomto uzlu a samozřejmě pravá strana diferenciální rovnice $f(t, y)$. Formálně lze tedy jednokrokovou metodu zapsat

$$y_{i+1} = y_i + h\Phi(t_i, y_i, y_{i+1}, h; f). \quad (2.34)$$

Funkce Φ je funkcí čtyř proměnných t_i, y_i, y_{i+1} a h , závislá na funkci $f(t, y)$. Pokud funkce Φ na y_{i+1} nezávisí, jde o *metodu explicitní*, pokud Φ na y_{i+1} závisí, jde o *metodu implicitní*. V některých literaturách se přiřazují implicitní numerické metody s výskytem y_i, y_{i+1} v (2.34) mezi vícekrokové metody (např. Lichoběžníková metoda v [MUK05]).

Metody Taylorova rozvoje

Mějme funkci $f(t)$, která má v bodě $t = \eta$ derivace až do n -tého řádu. Hledejme nyní polynom $P(t)$ stupně n ve tvaru

$$P(t) = c_0 + c_1(t - \eta) + c_2(t - \eta)^2 + \dots + c_n(t - \eta)^n, \quad (2.35)$$

se středem v bodě η takový, aby byly splněny podmínky

$$f(\eta) = P(\eta), \quad f'(\eta) = P'(\eta), \quad \dots, \quad f^{(n)}(\eta) = P^{(n)}(\eta). \quad (2.36)$$

Vyjádříme nyní vyšší derivace polynomu (2.35), tak aby byly splněny podmínky (2.36)

$$\begin{aligned} f(\eta) &= P(\eta) &= [c_0 + c_1(t - \eta) + c_2(t - \eta)^2 + \dots + c_n(t - \eta)^n]_{t=\eta} &= c_0, \\ f'(\eta) &= P'(\eta) &= [c_1 + 2 \cdot c_2(t - \eta) + \dots + n \cdot c_n(t - \eta)^{n-1}]_{t=\eta} &= c_1, \\ f''(\eta) &= P''(\eta) &= [2 \cdot c_2 + \dots + n \cdot (n-1) \cdot c_n(t - \eta)^{n-2}]_{t=\eta} &= 2 \cdot c_2, \\ &\vdots && \\ f^{(n)}(\eta) &= P^{(n)}(\eta) &= [n \cdot (n-1) \cdot \dots \cdot (n - (n-1)) \cdot c_n]_{t=\eta} &= n! \cdot c_n. \end{aligned}$$

Podmínky (2.36) vedou tedy na soustavu rovnic

$$\begin{aligned} c_0 &= f(\eta), \\ c_1 &= f'(\eta), \\ c_2 &= \frac{f''(\eta)}{2!}, \\ &\vdots \\ c_n &= \frac{f^{(n)}(\eta)}{n!}. \end{aligned} \quad (2.37)$$

Po dosazení rovnic (2.37) do původního polynomu (2.35) obdržíme polynom

$$P(t) = f(\eta) + \frac{f'(\eta)}{1!}(t - \eta) + \frac{f''(\eta)}{2!}(t - \eta)^2 + \dots + \frac{f^{(n)}(\eta)}{n!}(t - \eta)^n, \quad (2.38)$$

což je známý Taylorův polynom. Dosadíme-li do (2.38) $t = t_i + h = t_{i+1}$ a $\eta = t_i$, získáme známý *explicitní tvar* Taylorova polynomu

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \dots + \frac{h^n}{n!}y_i^{(n)}. \quad (2.39)$$

Pokud však dosadíme do (2.38) $t = t_i$ a $\eta = t_i + h = t_{i+1}$, obdržíme méně používaný Taylorův polynom v *implicitním tvaru*

$$y_i = y_{i+1} + (-h)y'_{i+1} + \frac{(-h)^2}{2!}y''_{i+1} + \cdots + \frac{(-h)^n}{n!}y^{(n)}_{i+1},$$

což je ekvivalentní s

$$y_{i+1} = y_i + hy'_{i+1} - \frac{h^2}{2!}y''_{i+1} + \cdots - \frac{(-h)^n}{n!}y^{(n)}_{i+1}. \quad (2.40)$$

Taylorův polynom (2.38) je základem veškerých jednokrokových numerických metod. Poskytuje nejpřesnější aproximaci funkce. Při výpočtu řešení je potřeba brát v úvahu větší počet členů (řádově alespoň desítky). Problémem je ovšem získávání vyšších derivací. Jsou-li metody Taylorova rozvoje dobře implementovány, mohou být pro řešení obecných problémů velmi efektivní.

Chyba výpočtu vzniklá zanedbáním vyšších členů Taylorovy řady

Lokální diskretizační chyba “*local truncation error*” vzniká obecně u numerických metod v jednom kroku výpočtu zanedbáním vyšších členů Taylorovy řady.

Definice 2.2.2 Obecná diferenční metoda

$$\begin{aligned} y_0 &= y(t_0), \\ y_{i+1} &= y_i + h\varphi(t_i, y_i), \quad i = 0, 1, \dots, k, \end{aligned}$$

kde $\varphi(t_i, y_i)$ je funkce závislá na konkrétní zvolené diferenční metodě, má v každém kroku lokální diskretizační chybu vzniklou zkrácením

$$\tau_{i+1}(h) = \frac{y_{i+1} - (y_i + h\varphi(t_i, y_i))}{h} = \frac{y_{i+1} - y_i}{h} - \varphi(t_i, y_i), \quad i = 0, 1, \dots, k,$$

kde uvažujeme, že $y_i = y(t_i)$ značí v tomto případě přesné (analytické) řešení.¹

Předpokládejme, že máme $n + 1$ spojitých derivací řešení $y(t)$ počáteční úlohy

$$y'(t) = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha.$$

Hodnotu v čase t_{i+1} můžeme vyjádřit Taylorovou řadou pomocí derivací v bodě t_i

$$y(t_{i+1}) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \cdots + \frac{h^n}{n!}y^{(n)}(t_i) + \frac{h^{n+1}}{(n+1)!}y^{(n+1)}(\xi_i), \quad (2.41)$$

kde $\xi_i \in (t_i, t_{i+1})$. Derivováním řešení $y(t)$ dostaneme

$$\begin{aligned} y'(t) &= f(t, y(t)), \\ y''(t) &= f'(t, y(t)), \\ &\vdots \\ y^{(n)}(t) &= f^{(n-1)}(t, y(t)). \end{aligned}$$

¹Neuvažujeme chybu vzniklou akumulací chyb z předchozích kroků.

Dosazením do rovnice (2.41) obdržíme

$$y(t_{i+1}) = y(t_i) + hf(t_i, y(t_i)) + \dots + \frac{h^n}{n!} f^{(n-1)}(t_i, y(t_i)) + \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)). \quad (2.42)$$

Numerickou metodu získanou z rovnice (2.42) zanedbáním posledního členu, který obsahuje neznámý parametr ξ_i ležící mezi t_i a t_{i+1} , obdržíme **Taylorovu metodu řádu n**

$$\begin{aligned} y_0 &= \alpha, \\ y_{i+1} &= y_i + hT^{(n)}(t_i, y_i), \quad i = 0, 1, \dots, k, \end{aligned} \quad (2.43)$$

kde

$$T^{(n)}(t_i, y_i) = f(t_i, y_i) + \frac{h}{2} f'(t_i, y_i) + \dots + \frac{h^{n-1}}{n!} f^{(n-1)}(t_i, y_i).$$

Poznamenejme, že Eulerova metoda je Taylorova metoda řádu 1.

Věta 2.2.3 *Použijeme-li Taylorovu metodu řádu n k aproximaci řešení rovnice*

$$y'(t) = f(t, y), \quad a \leq t \leq b, \quad y(a) = \alpha,$$

s integračním krokem h a pokud $y \in C^{n+1}[a, b]$, pak lokální diskretizační chyba je $O(h^n)$.

Důkaz *Rovnice (2.42) může být přepsána do tvaru*

$$y_{i+1} - y_i - hf(t_i, y_i) - \frac{h^2}{2} f'(t_i, y_i) - \dots - \frac{h^n}{n!} f^{(n-1)}(t_i, y_i) = \frac{h^{n+1}}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)),$$

kde $\xi \in (t_i, t_{i+1})$. Lokální diskretizační chyba je tedy

$$\tau_{i+1}(h) = \frac{y_{i+1} - y_i}{h} - T^{(n)}(t_i, y_i) = \frac{h^n}{(n+1)!} f^{(n)}(\xi_i, y(\xi_i)), \quad i = 0, 1, \dots, k.$$

Jelikož $y \in C^{n+1}[a, b]$ máme $y^{(n+1)}(t) = f^{(n)}(t, y(t))$ ohraničenu v intervalu $[a, b]$ a $\tau_{i+1} = O(h^n)$ pro $i = 1, 2, \dots, k+1$.

Podrobněji se pojmem “lokální diskretizační chyba” zabývá např. [BF04].

Metoda Eulerova

Explicitní Eulerova metoda (stručně EE metoda), je nejjednodušší metoda pro řešení počáteční úlohy (2.27), (2.28). Tato metoda se snadno odvodí z Taylorova rozvoje

$$y(t_{i+1}) = y(t_i + h) = y(t_i) + hy'(t_i) + \frac{h^2}{2} y''(\xi_i), \quad \xi_i \in (t_i, t_{i+1}). \quad (2.44)$$

Bereme-li v úvahu pouze první dva členy (výraz $\frac{h^2}{2} y''(\xi_i)$ je lokální chyba v daném kroku) a za výraz $y'(t_i)$ dosadíme $f(t_i, y_i)$, dostaneme předpis pro EE metodu

$$y_{i+1} = y_i + hf(t_i, y_i). \quad (2.45)$$

Implicitní Eulerova metoda (stručně IE metoda), je dána předpisem

$$y_{i+1} = y_i + hf(t_{i+1}, y_{i+1}). \quad (2.46)$$

Všimněme si neznámé y_{i+1} , která je argumentem funkce $f(t, y)$, takže z rovnice (2.46) nelze explicitně vyjádřit y_{i+1} , jak tomu bylo u EE metody. Musíme tedy navíc řešit rovnici

$$g(z) := z - y_i - hf(t_{i+1}, z) = 0, \quad (2.47)$$

odkud určíme y_{i+1} jako řešení z . Obecně se tato rovnice řeší pomocí vhodné iterační metody (např. Newtonovou metodou viz podkapitola 2.1.4). To je však ve srovnání s EE metodou problém navíc. IE metody se využívají u problémů, které vyžadují **větší oblast absolutní stability** (tuhé systémy).

Eulerovy metody řadíme do metod 1. řádu (využívají se pouze první dva členy Taylorova rozvoje). Eulerovy metody proto patří mezi méně přesné metody. Při zkracování kroku lze zpřesňovat řešení, ovšem od jisté hranice začne převládat vliv zaokrouhlovací chyby a celková chyba výpočtu při dalším zjemňování kroku poroste. Z hlediska stability vykazuje IE metoda větší oblast absolutní stability, při srovnání s EE metodou (podrobnější rozbor stability nalezneme v podkapitole 3.1).

Metody Runge-Kutta

Dalšími často používanými jednokrokovými metodami vyšších řádů, využívající pouze první derivace zadané funkce, jsou metody založené na výpočtu funkce $f(t, y)$ i mezi jednotlivými uzly $[t_i, y_i]$. Takovéto metody nazýváme *s-stupňové Runge-Kutta metody*. Obecný tvar těchto metod je

$$y_{i+1} = y_i + h \sum_{j=1}^s b_j k_j, \quad (2.48)$$

kde koeficienty k_j jsou určeny předpisem

$$k_j = f(t_i + hc_j, y_i + h \sum_{l=1}^s a_{j,l} k_l), \quad j = 1, 2, \dots, s. \quad (2.49)$$

Abychom získali konkrétní metodu, musíme určit stupeň s a reálné konstanty b_j , c_j a $a_{j,l}$ pro $j, l = 1, 2, \dots, s$. Tak např. pro dříve zmíněnou EE metodu musíme zvolit $s = 1$, $b_1 = 1$, $c_1 = 0$, $a_{1,1} = 0$ a pro IE metodu musíme zvolit $s = 1$, $b_1 = 1$, $c_1 = 1$, $a_{1,1} = 1$.

Obecně definovaná Runge-Kutta metoda s koeficienty (2.49) je *implicitní*. Pro určení koeficientu k_j musíme řešit soustavu s obecně nezávislých lineárních rovnic. Neznámé jsou složky všech koeficientů k_1, k_2, \dots, k_s .

Podstatně známější a v praxi častěji používané jsou metody *explicitní*, pro které $a_{j,l} = 0$ pro $l \geq j$ a pro koeficienty k_j platí tedy vztah

$$k_1 = f(t_i, y_i), \quad k_j = f(t_i + hc_j, y_i + h \sum_{l=1}^{j-1} a_{j,l} k_l), \quad j = 2, 3, \dots, s. \quad (2.50)$$

U těchto metod se žádné soustavy rovnic řešit nemusí. Stačí postupně počítat explicitně zadané koeficienty k_1, k_2, \dots, k_s .

Koeficienty metod Runge-Kutta je zvykem zapisovat do tabulky známé jako *Butcherova tabulka*

$$\begin{array}{c|cccc}
 c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\
 c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\
 \hline
 & b_1 & b_2 & \cdots & b_s
 \end{array} \tag{2.51}$$

U explicitních metod se do tabulky zapisují koeficienty $a_{k,l}$ jen pro $l < k$. Dále budeme uvažovat podmínku

$$c_j = \sum_{l=1}^s a_{j,l}, \quad j = 1, 2, \dots, s, \tag{2.52}$$

kteřou splňují všechny prakticky používané metody.

Obecně platí, že maximální dosažitelný řád $p(s)$ s -stupňové Runge-Kutta metody závisí na s následovně

$$\begin{array}{ll}
 p(s) = s & \text{pro } s = 1, 2, 3, 4, \\
 p(5) = 4, & \\
 p(6) = 5, & \\
 p(7) = 6, & \\
 p(8) = 6, & \\
 p(9) = 7, & \\
 p(s) \leq s - 2 & \text{pro } s = 10, 11, \dots
 \end{array} \tag{2.53}$$

V praxi se běžně setkáme s metodami čtvrtého řádu, které mají dostatečnou stabilitu i přesnost. Uveďme si nyní nejznámější explicitní s -stupňové Runge-Kutta metody řádu $1 \leq p \leq 4$. Pouze pro tyto řády existují metody, které počítají pravou stranu právě p -krát.

Metoda Runge-Kutta 1. řádu $s = p = 1$ existuje tedy jediná explicitní metoda a tou je již zmíněná explicitní Eulerova metoda $y_{i+1} = y_i + hf(t_i, y_i)$.

Metoda Runge-Kutta 2. řádu $s = p = 2$ Butcherova tabulka po základních úpravách má tvar

$$\begin{array}{c|c}
 0 & \\
 a & a \\
 \hline
 & 1 - b \quad b
 \end{array} \tag{2.54}$$

kde platí vztah $ab = \frac{1}{2}$. Parametry a, b jsou tedy svázány jednou podmínkou. Zvolíme-li $a \neq 0$, je $b = \frac{1}{2a}$. Nejčastěji se setkáme s dvěma variantami metody Runge-Kutta 2.řádu:

1. pro $a = \frac{1}{2}, b = 1$ dostaneme metodu

$$y_{i+1} = y_i + hk_2, \quad k_1 = f(t_i, y_i), \quad k_2 = f\left(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1\right), \tag{2.55}$$

známou pod názvem *modifikovaná Eulerova metoda* (v anglické literatuře “*midpoint Euler formula*”).

2. pro $a = 1, b = \frac{1}{2}$ dostaneme metodu

$$y_{i+1} = y_i + \frac{1}{2}h(k_1 + k_2), \quad k_1 = f(t_i, y_i), \quad k_2 = f(t_i + h, y_i + hk_1), \tag{2.56}$$

uváděnou také pod názvem *modifikovaná Eulerova metoda*, ale též pod názvem *Heunova metoda*.

Metoda Runge-Kutta 3. řádu $s = p = 3$ Butcherova tabulka má tvar

$$\begin{array}{c|ccc}
 0 & & & \\
 c_2 & c_2 & & \\
 c_3 & c_3 - a_{3,2} & a_{3,2} & \\
 \hline
 & b_1 & b_2 & b_3
 \end{array} \tag{2.57}$$

Metod Runge-Kutta 3. řádu je již více. Velice populární je *Ralstonova metoda* s vyladěnou chybou, která má tvar

$$\begin{array}{c|ccc}
 0 & & & \\
 \frac{1}{2} & \frac{1}{2} & & \\
 \frac{3}{4} & 0 & \frac{3}{4} & \\
 \hline
 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9}
 \end{array} \quad \begin{array}{l}
 y_{i+1} = y_i + \frac{1}{9}h(2k_1 + 3k_2 + 4k_3), \\
 k_1 = f(t_i, y_i), \\
 k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1), \\
 k_3 = f(t_i + \frac{3}{4}h, y_i + \frac{3}{4}hk_2).
 \end{array} \tag{2.58}$$

Metody Runge-Kutta 4. řádu $s = p = 4$ jsou nejpoužívanější ze všech dříve zmíněných metod. Nejznámější je *klasická Runge-Kutta metoda*

$$\begin{array}{c|cccc}
 0 & & & & \\
 \frac{1}{2} & \frac{1}{2} & & & \\
 \frac{1}{2} & 0 & \frac{1}{2} & & \\
 1 & 0 & 0 & 1 & \\
 \hline
 & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6}
 \end{array} \quad \begin{array}{l}
 y_{i+1} = y_i + \frac{1}{6}h(k_1 + 2k_2 + 2k_3 + k_4), \\
 k_1 = f(t_i, y_i), \\
 k_2 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_1), \\
 k_3 = f(t_i + \frac{1}{2}h, y_i + \frac{1}{2}hk_2), \\
 k_4 = f(t_i + h, y_i + hk_3).
 \end{array} \tag{2.59}$$

Klasické Runge-Kutta metody jsou velice populární pro svou jednoduchost, dobrou stabilitu a přesnost. Díky tomu metody Runge-Kutta patří stále mezi nejpoužívanější numerické metody. Jejich modifikacemi se celý život zabývá prof. Butcher [But08b].

V dnešní době výkonných počítačů se však příliš často “klasické” Runge-Kutta metody nepoužívají. Jejimi nástupci jsou vylepšené *dvojice vnořených Runge-Kutta metod*. Základní myšlenka těchto metod je jednoduchá. Použijí se dvě metody z nichž jedna je řádu p a druhá řádu $p + 1$. Z výchozí hodnoty y_i spočteme y_{i+1}^{**} přesnější metodou a y_{i+1}^* méně přesnou metodou. Tyto dvě metody rozdílných řádů se využívají k odhadu lokální chyby $\varepsilon_i = y_{i+1}^{**} - y_{i+1}^*$, kterou pak využívají k řízení délky kroku.

Dvojice Runge-Kutta metod se popisují pomocí *rozšířené Butcherovy tabulky*

$$\begin{array}{c|cccc}
 c_1 & a_{1,1} & a_{1,2} & \cdots & a_{1,s} \\
 c_2 & a_{2,1} & a_{2,2} & \cdots & a_{2,s} \\
 \vdots & \vdots & \vdots & \ddots & \vdots \\
 c_s & a_{s,1} & a_{s,2} & \cdots & a_{s,s} \\
 \hline
 & b_1^* & b_2^* & \cdots & b_s^* \\
 & b_1^{**} & b_2^{**} & \cdots & b_s^{**} \\
 \hline
 & E_1 & E_2 & \cdots & E_s
 \end{array} \quad \begin{array}{l}
 k_j = f(t_i + hc_j, y_i + h \sum_{l=1}^s a_{j,l}k_l), \quad j = 1, 2, \dots, s, \\
 y_{i+1}^* = y_i + h \sum_{j=1}^s b_j^* k_j, \\
 y_{i+1}^{**} = y_i + h \sum_{j=1}^s b_j^{**} k_j, \\
 \varepsilon_i = h \sum_{j=1}^s E_j k_j, \\
 E_j = b_j^{**} - b_j^*, \quad j = 1, 2, \dots, s.
 \end{array}$$

Zmíníme se jen o dvou nejznámějších metodách, které se úspěšně používají např. v programovém produktu MATLAB [Mat10a].

První metodou je **Bogacki-Shampine (3,2) metoda**, která využívá jako přesnější metodu Ralstonovu metodu řádu 3 s vyladěnou chybou $y_{i+1}^{**} = y_i + h(\frac{2}{9}k_1 + \frac{1}{3}k_2 + \frac{4}{9}k_3) = y_{i+1}$ a pomocnou metodu $y_{i+1}^* = y_i + h(\frac{7}{24}k_1 + \frac{1}{4}k_2 + \frac{1}{3}k_3 + \frac{1}{8}k_4)$ řádu 2. Koeficient $k_4 = f(t_{i+1}, y_{i+1})$ v pomocné metodě je roven koeficientu k_1 v následujícím kroku, proto nezvyšuje výpočetní složitost. V MATLABu tuto metodu najdeme pod označením ODE23.

Druhou často používanou metodou je **Dormand-Price (5,4) metoda**. Tato metoda je řádu 5, počítá však se sedmi koeficienty k_i . Ve skutečnosti se provádí o jeden výpočet méně (tedy při každém kroku počítáme pravou stranu 6-krát). Podobně jako u předchozí metody $k_7 = f(t_{i+1}, y_{i+1})$ se použije opět v následujícím kroku pro k_1 . V MATLABu tuto metodu najdeme jako funkci ODE45.

Dalšími velice populárními metodami jsou **lichoběžníkové metody** (v anglické literatuře *trapezoidal rule*) dané předpisem

$$y_{i+1} = y_i + \frac{1}{2}h[f(t_i, y_i) + f(t_{i+1}, y_{i+1})]. \quad (2.60)$$

Jedná se o jednokrokové implicitní Runge-Kutta metody řádu 2, které mají velkou absolutní stabilitu (jsou A-stabilní, L-stabilní však nejsou²), používají se prakticky výhradně pro řešení *tuhých problémů*. V MATLABu je klasická implicitní lichoběžníková metoda základem funkce ODE23T.

Pro složitější tuhé problémy vyžadující vyšší stabilitu (tedy L-stabilitu) existuje vylepšená implicitní *dvojitá metoda lichoběžníková* řádu 2 využívající metodu zpětného derivování řádu 2. Metoda je implementována v MATLABu jako funkce ODE23TB.

2.2.3 Vícekrokové metody

Pojem vícekové metody (k -krokové metody) zahrnuje metody, které používají k výpočtu hodnoty y_{i+1} v uzlu t_{i+1} , dříve spočtené hodnoty $y_i, y_{i-1}, \dots, y_{i+1-k}$ a odpovídající hodnoty $f(t_i), f(t_{i-1}), \dots, f(t_{i+1-k})$ pravé strany diferenciální rovnice. Obecný tvar předpisu pro výpočet hodnoty y_{i+1} pomocí k -krokové metody je

$$y_{i+1} = h \sum_{j=0}^k b_j f(t_{i+1-k}, y_{i+1-k}) - \sum_{j=1}^k a_j y_{i+1-j}. \quad (2.61)$$

Pokud je ve vztahu (2.61) koeficient $b_0 = 0$ mluvíme o *explicitních metodách*, jinak pro $b_0 \neq 0$ hovoříme o *implicitních metodách*, kde musíme řešit navíc rovnici

$$y_{i+1} = \varphi(y_{i+1}), \quad \text{kde} \quad \varphi(z) = hb_0 f(t_{i+1}, z) + \sum_{j=1}^k (hb_j f(t_{i+1-j}, y_{i+1-j}) - a_j y_{i+1-j}), \quad (2.62)$$

abychom určili y_{i+1} . Problémem vícekové metody je získání prvních k *startovacích hodnot* y_0, y_1, \dots, y_{k-1} při začátku výpočtu. Hodnotu y_0 určíme z počáteční podmínky a ostatní hodnoty $y_j, j = 1, \dots, k-1$ můžeme spočítat pomocí jiné j -krokové metody nebo vhodné jednokrokové metody.

Nejznámějšími vícekové metodami jsou *Adamsovy metody* a *metody zpětného derivování*. Obě skupiny metod patří mezi *lineární vícekové metody*.

²Podrobněji se s pojmy *A-stabilita*, *L-stabilita* seznámíme v podkapitole 3.1.

Adamsovy metody

Adamsovy metody jsou charakteristické tím, že koeficient a_1 ze vztahu (2.61) je jako jediný nenulový. Podle koeficientu b_0 můžeme rozdělit metody na *explicitní Adams-Bashfortovy metody* (pro $b_0 = 0$) a *implicitní Adams-Moultonovy metody* (pro $b_0 \neq 0$). Z hlediska stability (většinou D-stabilní) se tyto metody příliš nehodí pro řešení *tuhých systémů*. Pro mnoho systémů, které nejeví známky “tuhosti”, se však hojně využívají.

Adams-Bashfortovy metody nalezneme ve tvaru

$$y_{i+1} = y_i + h \sum_{j=1}^k b_j f(t_{i+1-j}, y_{i+1-j}). \quad (2.63)$$

Nejčastěji se setkáme s metodami, které pracují s počtem kroků $k = 1, \dots, 6$. Jednotlivé koeficienty pro tyto metody s pevnou délkou kroku h jsou uvedeny v Tab. 2.1.

Tabulka 2.1: Koeficienty Adams-Bashfortovy metody

k	b_1	b_2	b_3	b_4	b_5	b_6
1	1					
2	$\frac{3}{2}$	$-\frac{1}{2}$				
3	$\frac{23}{12}$	$-\frac{16}{12}$	$\frac{5}{12}$			
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{9}{24}$		
5	$\frac{1901}{720}$	$-\frac{2774}{720}$	$\frac{2616}{720}$	$-\frac{1274}{720}$	$\frac{251}{720}$	
6	$\frac{4277}{1440}$	$-\frac{7923}{1440}$	$\frac{9982}{1440}$	$-\frac{7298}{1440}$	$\frac{2877}{1440}$	$-\frac{475}{1440}$

Pro metodu řádu $k = 1$ dostaneme již známou explicitní Eulerovu metodu (2.45). První vícekrokovou metodu získáme při $k = 2$

$$y_{i+1} = y_i + h \frac{3}{2} f(t_i, y_i) - \frac{1}{2} f(t_{i-1}, y_{i-1}). \quad (2.64)$$

Adams-Moultonovy metody nalezneme ve tvaru

$$y_{i+1} = y_i + h \sum_{j=0}^k b_j f(t_{i+1-j}, y_{i+1-j}). \quad (2.65)$$

Následkem členu $hb_0 f(t_{i+1}, y_{i+1})$, který se vyskytuje na pravé straně výrazu, mluvíme o implicitních Adams-Moultonových metodách. V Tab. 2.2 nalezneme koeficienty pro metody řádu $k = 1, 2, \dots, 6$ s konstantní délkou kroku.

Pro $k = 1$ obdržíme implicitní Eulerovu metodu (2.46), pro $k = 2$ dostaneme lichoběžníkovou metodu (2.60). Obě tyto metody jsou jednokrokové. Vícekrokovou metodu získáme až při $k \geq 3$.

Jak jsme se již dříve zmínili, problémem u vícekrokových metod je získání počátečních k startovacích hodnot y_0, y_1, \dots, y_{k-1} . V moderních programech se používají tzv. *samostartující algoritmy*. Startovací hodnoty např. pro k -krokovou Adams-Bashfortovu metodu se

Tabulka 2.2: Koeficienty Adams-Moultonovy metody

k	b_0	b_1	b_2	b_3	b_4	b_5
1	1					
2	$\frac{1}{2}$	$\frac{1}{2}$				
3	$\frac{5}{12}$	$\frac{8}{12}$	$-\frac{1}{12}$			
4	$\frac{9}{24}$	$\frac{19}{24}$	$-\frac{5}{24}$	$\frac{1}{24}$		
5	$\frac{251}{720}$	$\frac{646}{720}$	$-\frac{264}{720}$	$\frac{106}{720}$	$-\frac{19}{720}$	
6	$\frac{475}{1440}$	$\frac{1427}{1440}$	$-\frac{798}{1440}$	$\frac{482}{1440}$	$-\frac{173}{1440}$	$\frac{27}{1440}$

počítají opět pomocí Adams-Bashfortových metod, y_1 pomocí Adams-Bashfortovy jednokrokové metody, y_2 pomocí Adams-Bashfortovy dvou-krokové metody atd. , až nakonec y_{k-1} pomocí Adams-Bashfortovy $(k-1)$ -krokové metody.

Adams-Bashfortovy a Adams-Moultonovy metody se samostatně téměř nepoužívají. Jejich kombinace (prediktor-korektor metody) však patří k základnímu programovému vybavení pro řešení obyčejných diferenciálních rovnic.

Kombinací explicitní a implicitní Adamsovy metody získáme tzv. **prediktor-korektor metody**. Explicitní formule zde slouží jako prediktor a předpovídá se s její pomocí hodnota řešení pro implicitní metodu. Schematicky značíme tento výpočet PECE. P označuje výpočet aproximace počáteční explicitní Adams-Bashfortovou metodou, C je korekce pomocí implicitní Adams-Moultonovy metody a E je vyhodnocení pravé strany rovnice, tedy funkce $f(t_{i+1}, y_{i+1})$. Konkrétně pro Adamsovy metody druhého řádu počítáme PECE³ schéma následovně

$$\begin{array}{ll}
 P & \text{Adams-Bashfort řádu 2} & y_{i+1}^* = y_i + \frac{1}{2}h(3f_i - f_{i-1}), \\
 E & & f_{i+1}^* = f(t_{i+1}, y_{i+1}^*), \\
 C & \text{Adams-Moulton řádu 2} & y_{i+1} = y_i + \frac{1}{2}h(f_{i+1}^* + f_i), \\
 E & & f_{i+1} = f(t_{i+1}, y_{i+1}).
 \end{array} \tag{2.66}$$

Kvalitní programy založené na metodách prediktor-korektor mění během výpočtu jak délku kroku, tak i řád metody. Příkladem může být funkce ODE113 v MATLABu, využívající schéma PECE Adamsovy metody prediktor-korektor. Změnu řádu provádí současně se změnou délky kroku. Takovéto algoritmy označujeme VSVO algoritmy (*variable step variable order*).

Oblasti absolutní stability PECE prediktor-korektor metod nejsou příliš velké a s rostoucím řádem se dále zmenšují. Proto tyto metody nejsou příliš vhodné k řešení tuhých systémů.

³V poslední fázi E vypočtenou hodnotu funkce f_{i+1} dosadíme opět do rovnice (2.65) a tím získáme výslednou hodnotu y_{i+1} . Při vynechání posledního fáze E (vyhodnocení pravé strany) se jedná o schéma PEC metody prediktor-korektor.

Metody zpětného derivování

Pro řešení tuhých systémů jsou potřebné metody, které se vyznačují velkou oblastí absolutní stability. Metody zpětného derivování (*backward differentiation formulas*) tuto vlastnost mají. Obecný předpis pro k -krokové metody zpětného derivování je

$$\alpha_{k,0}y_{i+1} + \alpha_{k,1}y_i + \cdots + \alpha_{k,k}y_{i+1-k} = hf(t_{i+1}, y_{i+1}),$$

$$\alpha_{k,j} = hl'_j(t_{i+1}), \quad l_j(t) = \prod_{\substack{m=1, \\ m \neq j}}^k \frac{t-t_{i+1-m}}{t_{i+1-j}-t_{i+1-m}}. \quad (2.67)$$

Metody zpětného derivování tedy patří mezi k -krokové implicitní metody. Pro $k = 1$ obdržíme opět již známou implicitní Eulerovu metodu (2.46).

V Tab. 2.3 nalezneme koeficienty pro k -krokové metody zpětného derivování s konstantní délkou kroku (pro $k = 1, 2, \dots, 6$).

Tabulka 2.3: Koeficienty metod zpětného derivování

k	$\alpha_{k,0}$	$\alpha_{k,1}$	$\alpha_{k,2}$	$\alpha_{k,3}$	$\alpha_{k,4}$	$\alpha_{k,5}$	$\alpha_{k,6}$
1	1	-1					
2	$\frac{3}{2}$	-2	$\frac{1}{2}$				
3	$\frac{11}{6}$	-3	$\frac{3}{2}$	$-\frac{1}{3}$			
4	$\frac{25}{12}$	-4	3	$-\frac{4}{3}$	$\frac{1}{4}$		
5	$\frac{137}{60}$	-5	5	$-\frac{10}{3}$	$\frac{5}{4}$	$-\frac{1}{5}$	
6	$\frac{147}{60}$	-6	$\frac{15}{12}$	$-\frac{20}{3}$	$\frac{15}{4}$	$-\frac{6}{5}$	$\frac{1}{6}$

Metody zpětného derivování jsou oproti Adamsovým metodám dosti nepřesné. Mají však velkou přednost a z tohoto důvodu jsou často používány - metody zpětného derivování mají mnohem větší oblast absolutní stability. Pro k -krokové metody, kde $k = 1, 2$, pokrývá oblast absolutní stability celou zápornou komplexní rovinu, jsou tedy A-stabilní. Metody, kde $k = 3, 4, 5, 6$ se oblast absolutní stability postupně zmenšuje, jsou A(α)-stabilní. Obecně však můžeme říct, že všechny metody zpětného derivování jsou A_0 -stabilní. Podrobněji se seznámíme se stabilitou numerických metod v podkapitole 3.1.

V praxi se nejčastěji setkáme s implementací k -krokových metod zpětného derivování pro $k = 1, 2, \dots, 5$. V programu MATLAB je implementována funkce ODE15s, která počítá diferenciální rovnice pomocí metod zpětného derivování řádů 1 až 5. Funkce ODE15s je typu VSVO, během výpočtu tedy vybírá optimální délku kroku a řád metody.

2.3 Shrnutí

Existuje značné množství numerických metod sloužících k řešení soustav obyčejných diferenciálních rovnic. Při volbě vhodné numerické metody pro daný problém (soustavu ODR) je vhodné znát možné chování systému. Uživatel je předem nucen se rozhodnout, kterou numerickou metodu pro daný problém použije. Při řešení tuhých soustav diferenciálních

rovníc jsme nuceni využívat numerických metod s velkou oblastí absolutní stability, tedy především implicitních numerických metod. U běžných implicitních numerických metod je velká oblast stability pouze u metod nižších řádů. V odborné literatuře se tedy často setkáme s možnou definicí tuhých systémů, kdy při řešení tuhých soustav diferenciálních rovnic se upřednostňuje **rychlost výpočtu na úkor přesnosti výpočtu**.

Stále není vyřešen problém přesné definice tuhých systémů a tedy jejich detekce. Detekce se v mnoha programech provádí pomocí Jacobiovy matice a vlastních čísel. V další části práce si ukážeme nový možný způsob detekce využívající členů Taylorovy řady (viz podkapitola 4.1).

Kapitola 3

Tuhé systémy

... Around 1960, things became completely different and everyone became aware that the world was full of stiff problems. (G. Dahlquist in Aiken 1985)

Náplní této práce je analyzovat tuhé systémy diferenciálních rovnic a možnosti jejich řešení pomocí “moderní metody Taylorovy řady”.

Vysvětleme si nejprve pojem “tuhé systémy (problémy)”. Při řešení mnoha praktických úloh vznikají soustavy diferenciálních rovnic, které vykazují jev označovaný v anglické literatuře “stiffness”. Soustavy se pak nazývají “stiff systems”. V některých česky psaných publikacích se můžeme setkat také s pojmem “úlohy se silným tlumením” [Vit94].

Tuhé systémy nemají přesnou definici. Obvykle se uvádí, že jsou to soustavy diferenciálních rovnic, pro které je řešení pomocí explicitních metod neefektivní [HW02]. Častěji se však vychází z tvrzení, že se jedná o soustavy se značně rozdílnými časovými koeficienty [Vit94], tedy vlastními čísly (podkapitola 2.1.2) Jacobiva matice soustavy diferenciálních rovnic (podkapitola 2.1.1). Podrobněji se budeme touto definicí zabývat v podkapitole 4.1.

Tuhé soustavy diferenciálních rovnic se vyznačují tím, že při jejich numerickém řešení omezuje délku kroku především stabilita použité numerické metody, což se projevuje na přesnosti metody. Stabilita je jedním z limitujících faktorů efektivního řešení tuhých soustav diferenciálních rovnic a je různá podle použité konkrétní numerické metody. Cílem této kapitoly je čtenáře seznámit s rozdělením běžných numerických metod z pohledu jejich oblastí absolutní stability.

3.1 Stabilita

Pojmem stabilita si definujeme jako stabilitu numerické metody. Obecně můžeme říct, že numerická metoda je stabilní, jestliže při výpočtu libovolného stabilního počátečního problému má následující vlastnost: malá změna startovacích hodnot způsobí malou změnu numerického řešení (pro jednokrokovou metodu jsou startovací hodnoty počáteční podmínky, pro k -krokovou metodu jsou startovací hodnoty y_0, y_1, \dots, y_{k-1}). Důkazy a podrobnější rozbor nalezneme např. v [Č04].

U většiny numerických metod vyžaduje stabilita metody omezení integračního kroku h tak, aby platil vztah

$$|h\lambda| \leq K, \quad (3.1)$$

kde λ je v absolutní hodnotě maximální vlastní číslo Jacobiovy matice, K je pak konstanta závislá na konkrétní numerické metodě a diferenciální rovnici.

Stabilitu numerického řešení dané diferenciální rovnice si ukážeme na následujícím příkladu.

Příklad: oblasti absolutní stability pro explicitní Eulerovu metodu

Mějme zadanou počáteční úlohu

$$y' = \lambda y, \tag{3.2}$$

s počáteční podmínkou

$$y(0) = 1, \tag{3.3}$$

kde λ je libovolná konstanta (obecně i komplexní číslo) splňující podmínku $\lambda < 0$ (resp. $\Re(\lambda) < 0$).

Známe analytické řešení rovnice (3.2) je

$$y(t) = e^{\lambda t}. \tag{3.4}$$

Jedná se tedy o klesající funkci, kde $e^{\lambda t} \rightarrow 0$ pro $t \rightarrow +\infty$. Počáteční hodnotu (3.3) lze považovat za jednotkovou poruchu a proměnnou t za čas. Úlohu (3.2) s počáteční podmínkou (3.3) lze tedy interpretovat jako proces stabilizace: s rostoucím časem dochází k útlumu počáteční poruchy neustálým přibližováním se k nulovému stacionárnímu stavu. Takové chování vykazuje celá řada dějů reálného světa.

Od numerické metody (v našem případě explicitní Eulerovy metody) vyžadujeme, aby se numerické řešení blížilo ke stacionárnímu stavu, tedy $y_i \rightarrow 0$ pro $i \rightarrow \infty$. Aby bylo tlumení nepřetržité, požadujeme splnění podmínky

$$|y_{i+1}| \leq |y_i|. \tag{3.5}$$

Explicitní Eulerova metoda pro danou úlohu (3.2) je zadána vztahem

$$y_{i+1} = y_i + h y_i' \tag{3.6}$$

po dosazení je

$$y_{i+1} = y_i + h \lambda y_i = (1 + h \lambda) y_i = (1 + h \lambda)^i y_0, \tag{3.7}$$

kde $y_0 = y(0) = 1$ podle počáteční podmínky (3.3).

Aby platila podmínka konvergence (3.5), musí platit

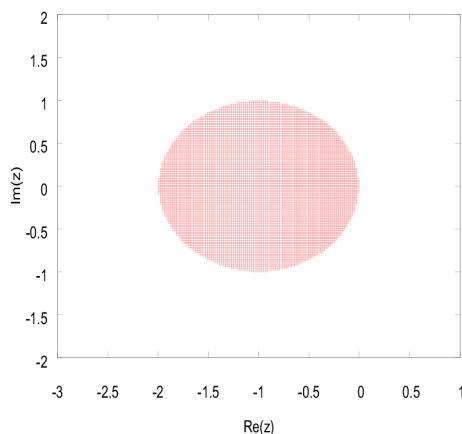
$$|1 + h \lambda| \leq 1. \tag{3.8}$$

⊙

Hodnocení explicitní Eulerovy metody z pohledu stability

Explicitní Eulerova metoda je absolutně stabilní pro λ a h takové, pro které je splněna podmínka (3.8). Položme nyní $z = h \lambda$ a řekneme, že oblastí absolutní stability je v našem případě jednotkový kruh $|z + 1| \leq 1$ Obr. 3.1 (zvýrazněná část) komplexní roviny se středem $(-1, 0)$.

Explicitní Eulerova metoda nemá tedy příliš velkou oblast absolutní stability a z tohoto hlediska ji nelze použít k řešení tuhých diferenciálních rovnic (pro které je obecně $|\lambda| \gg 1$).



Obrázek 3.1: Oblast absolutní stability - explicitní Eulerova metoda

3.1.1 Oblasti stabilit numerických metod

Z hlediska velikosti oblasti absolutní stability řadíme numerické metody do několika základních skupin.

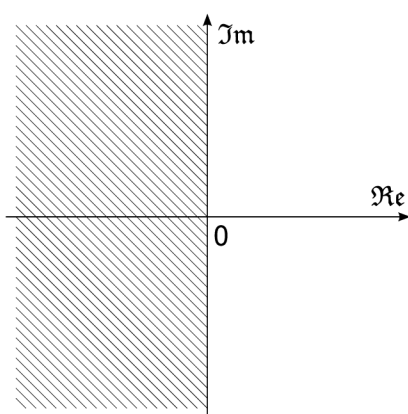
První skupinou jsou **A-stabilní metody**. Jedná se o metody s největší oblastí absolutní stability.

Definice 3.1.1 (Dahlquist 1963). *Numerické metody, u kterých oblast absolutní stability \mathcal{R}_A splňuje podmínku*

$$\mathcal{R}_A \supseteq \mathcal{C}^-, \quad (3.9)$$

se nazývají A-stabilní.

Oblastí absolutní stability A-stabilní numerické metody je tedy nadmnožina celé záporné poloroviny $\mathcal{C}^- = \{z \in \mathcal{C}; \Re(z) < 0\}$ viz Obr. 3.2.



Obrázek 3.2: Oblast stability A-stabilní numerické metody

Jedná se o silný požadavek, který splňuje jen málo numerických metod. Z dříve zmiňova-

ných metod zde můžeme zařadit lichoběžníkovou metodu, která je dána vztahem

$$y_{i+1} = y_i + \frac{h}{2}(y'_i + y'_{i+1}). \quad (3.10)$$

Její aplikace na úlohu (3.2) poskytne

$$y_{i+1} = y_i + \frac{h}{2}(\lambda y_i + \lambda y_{i+1}) = \frac{y_i(1 + h\lambda/2)}{1 - h\lambda/2}, \quad (3.11)$$

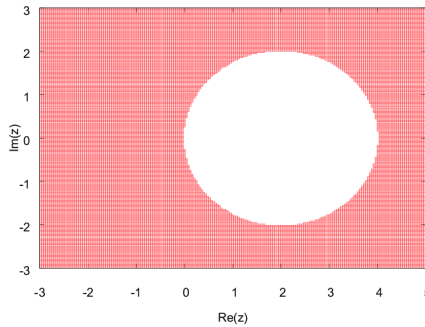
požadujeme splnění podmínky nepřetržitého tlumení

$$|y_{i+1}| \leq |y_i|, \quad \text{když } \Re(h\lambda) < 0,$$

dále položíme $z = h\lambda$ a obdržíme

$$\frac{y_{i+1}}{y_i} = \frac{1 + z/2}{1 - z/2}. \quad (3.12)$$

Oblast absolutní stability tedy musí splňovat podmínku $(1 + z/2)/(1 - z/2) \leq 1$ viz Obr.3.3.



Obrázek 3.3: Oblast absolutní stability - lichoběžníková metoda

Z vícekových metod jsou dále A-stabilní metody Adamsovy do druhého řádu a metody zpětného derivování rovněž do druhého řádu.

Tyto metody jsou pro svou velkou stabilitu vhodné k řešení tuhých systémů. Jedná se však o metody maximálně druhého řádu.

U metod vyšších řádů pokrývá oblast absolutní stability často jen určitou část záporné poloviny určené úhlem α způsobem z Obr. 3.4. Mluvíme o **A(α)-stabilních** numerických metodách.

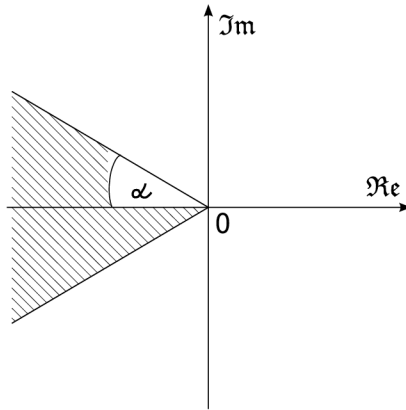
Definice 3.1.2 (Widlund 1967). *O numerické metodě řekneme, že je A(α)-stabilní, jestliže její oblast absolutní stability \mathfrak{R}_A splňuje podmínku*

$$\mathfrak{R}_A \supseteq \{z | (\Re(z) < 0) \wedge (|\frac{\Im m(z)}{\Re(z)}| < \tan \alpha)\}. \quad (3.13)$$

Jako příklad A(α)-stabilních metod můžeme uvést k -krokové metody zpětného derivování (viz podkapitola 2.2.3), pro které jsou hodnoty úhlu α uvedeny v Tab. 3.1.

Zvláštní podmnožinu tvoří **A₀-stabilní metody**. Jsou to metody, jejichž interval absolutní stability pokrývá interval $(-\infty, 0)$. Definice oblasti absolutní stability \mathfrak{R}_A je tedy

$$\mathfrak{R}_A \supseteq \{z | \Re(z) < 0, \Im(z) = 0\}. \quad (3.14)$$



Obrázek 3.4: Oblast stability $A(\alpha)$ -stabilní numerické metody

Tabulka 3.1: Velikosti úhlu α u metod zpětného derivování

k	1	2	3	4	5	6
α	90°	90°	88°	73°	52°	18°

$A(\alpha)$ -stabilní metoda je A_0 -stabilní metoda pro libovolné $\alpha \in (0, \pi/2)$. Do této skupiny metod patří všechny k -krokové metody zpětného derivování.

Na některé “velmi tuhé” problémy A -stabilní metody “nestačí”. Tyto problémy vyžadují metody s ještě přísnější stabilitou a konvergencí, tzv. **L-stabilní metody**. Vraťme se k našemu příkladu (3.2) s počáteční podmínkou (3.3). Při obecném řešení numerickou metodou získáme rovnici

$$y_{i+1} = R(z)y_i, \quad \text{kde } z = h\lambda. \quad (3.15)$$

Funkce $R(z) = y_{i+1}/y_i$ se nazývá *funkce stability*. Pro správné řešení musí zřejmě platit $|y_{i+1}| \leq |y_i|$, to samozřejmě platí když

$$|R(z)| \leq 1. \quad (3.16)$$

Nyní si můžeme definovat L -stabilní metody.

Definice 3.1.3 (Ehle 1969). *Metodu nazýváme L -stabilní, pokud je A -stabilní a existuje limita*

$$\lim_{\Re(z) \rightarrow -\infty} R(z) = 0. \quad (3.17)$$

Mezi L -stabilní metody můžeme zařadit implicitní Eulerovu metodu. Ukážeme si pro názornost odvození funkce stability $R(z)$ pro náš příklad (3.2).

Mějme implicitní Eulerovu metodu ve tvaru

$$y_{i+1} = y_i + hf(x_{i+1}, y_{i+1}). \quad (3.18)$$

Dosadíme-li dříve zmíněný příklad (3.2) do implicitní Eulerovy formule (3.18), obdržíme

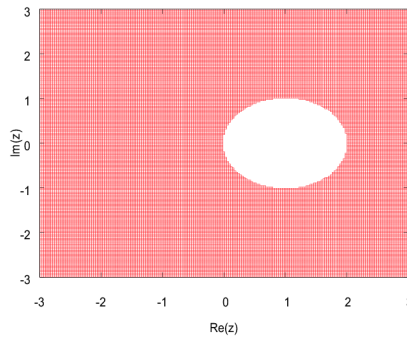
$$y_{i+1} = y_i + h\lambda y_{i+1}. \quad (3.19)$$

Ze vztahů (3.15) a (3.19) již snadno odvodíme funkci stability pro implicitní Eulerovu metodu

$$R(z) = \frac{1}{1-z}, \quad (3.20)$$

obecně tedy pro $z = a + ib$, kde $a < 0$, je $R(z) = \frac{1}{1-a-ib}$.

Platí $|R(z)| \leq 1$ a $\lim_{a \rightarrow -\infty} R(z) = 0$. Implicitní Eulerova metoda je tedy L-stabilní. Její oblast absolutní stability je znázorněna na Obr. 3.5.



Obrázek 3.5: Oblast absolutní stability - implicitní Eulerova metoda

V Tab. 3.2 jsou uvedeny funkce stabilit některých často používaných numerických metod.

Tabulka 3.2: Funkce stabilit: $R(z) = y_{i+1}/y_i$

metoda	$R(z)$
explicitní Eulerova metoda	$1 + z$
implicitní Eulerova metoda	$\frac{1}{1-z}$
Lichoběžníková metoda	$\frac{1+z/2}{1-z/2}$

3.2 Padé aproximace exponenciální funkce

Padé aproximace exponenciální funkce (Padé 1892) jsou racionální funkce, které poskytují největší řád chyby aproximace funkce e^z pro daný stupeň jmenovatele a čitatele.

Teorém 3.2.1 (k,j) -Padé aproximace k funkci e^z je dána vztahem

$$R_{kj}(z) = \frac{P_{kj}(z)}{Q_{kj}(z)}, \quad (3.21)$$

kde

$$P_{kj}(z) = 1 + \frac{k}{j+k} \cdot z + \frac{k(k-1)}{(j+k)(j+k-1)} \cdot \frac{z^2}{2!} + \cdots + \frac{k(k-1)\cdots 1}{(j+k)\cdots(j+1)} \cdot \frac{z^k}{k!},$$

$$Q_{kj}(z) = 1 - \frac{j}{k+j} \cdot z + \frac{j(j-1)}{(k+j)(k+j-1)} \cdot \frac{z^2}{2!} - \cdots + (-1)^j \frac{j(j-1)\cdots 1}{(k+j)\cdots(k+1)} \cdot \frac{z^j}{j!},$$

Řád aproximace funkce e^z je tedy roven $j+k$, kde k značí stupeň polynomu čitatele, j značí stupeň polynomu jmenovatele. Chyba aproximace je rovna $O(z^{j+k+1})$.

V Tab. 3.3 nalezneme některé Padé aproximace exponenciální funkce e^z . Připomeňme, že pro $j = 0$ v prvním řádku Tab. 3.3 obdržíme explicitní Taylorův rozvoj řádu k a ve sloupci pro $k = 0$ implicitní Taylorův rozvoj řádu j .

Tabulka 3.3: Padé aproximace pro funkci e^z

$R_{kj}(z)$	$k = 0$	$k = 1$	$k = 2$	$k = 3$
$j = 0$	$\frac{1}{1}$	$\frac{1+z}{1}$	$\frac{1+z+\frac{z^2}{2!}}{1}$	$\frac{1+z+\frac{z^2}{2!}+\frac{z^3}{3!}}{1}$
$j = 1$	$\frac{1}{1-z}$	$\frac{1+\frac{1}{2}z}{1-\frac{1}{2}z}$	$\frac{1+\frac{2}{3}z+\frac{1}{3}\frac{z^2}{2!}}{1-\frac{1}{3}z}$	$\frac{1+\frac{3}{4}z+\frac{1}{2}\frac{z^2}{2!}+\frac{1}{4}\frac{z^3}{3!}}{1-\frac{1}{4}z}$
$j = 2$	$\frac{1}{1-z+\frac{z^2}{2!}}$	$\frac{1+\frac{1}{3}z}{1-\frac{2}{3}z+\frac{1}{3}\frac{z^2}{2!}}$	$\frac{1+\frac{1}{2}z+\frac{1}{6}\frac{z^2}{2!}}{1-\frac{1}{2}z+\frac{1}{6}\frac{z^2}{2!}}$	$\frac{1+\frac{3}{5}z+\frac{3}{10}\frac{z^2}{2!}+\frac{1}{10}\frac{z^3}{3!}}{1-\frac{2}{5}z+\frac{1}{10}\frac{z^2}{2!}}$
$j = 3$	$\frac{1}{1-z+\frac{z^2}{2!}-\frac{z^3}{3!}}$	$\frac{1+\frac{1}{4}z}{1-\frac{3}{4}z+\frac{1}{2}\frac{z^2}{2!}-\frac{1}{4}\frac{z^3}{3!}}$	$\frac{1+\frac{2}{5}z+\frac{1}{10}\frac{z^2}{2!}}{1-\frac{3}{5}z+\frac{3}{10}\frac{z^2}{2!}-\frac{1}{10}\frac{z^3}{3!}}$	$\frac{1+\frac{1}{2}z+\frac{1}{5}\frac{z^2}{2!}+\frac{1}{20}\frac{z^3}{3!}}{1-\frac{1}{2}z+\frac{1}{5}\frac{z^2}{2!}-\frac{1}{20}\frac{z^3}{3!}}$

Teorem 3.2.2 Padé aproximace $R_{kj}(z)$ definovaná 3.21 je A -stabilní pokud platí $k \leq j \leq k+2$.

Podrobnější rozbor Padé aproximace exponenciální funkce a důkazy Vět 3.2.1, 3.2.2 nalezneme v [Ehl69, HW02, But08b].

3.2.1 Oblast stability řádu hvězd

When I wrote my book in 1971 I wanted to draw “relative stability domains”, but curious stars came out from the plotter. I thought of an error in the program and I threw them away ...

(C.W.Gear, in 1979)

Oblast stability řádu hvězd (“Order stars”) neporovnává $|R(z)|$ k 1, jak je tomu u oblastí stability numerických metod, ale porovnává $|R(z)|$ přímo k exaktnímu řešení $|e^z| = e^t$ a očekává více informací.

Definice 3.2.3 Množina

$$A = \{z \in \mathcal{C}; |R(z)| > |e^z|\} = \{z \in \mathcal{C}; |q(z)| > 1\},$$

kde $q(z) = R(z)/e^z$, se nazývá oblast stability řádu hvězd funkce $R(z)$.

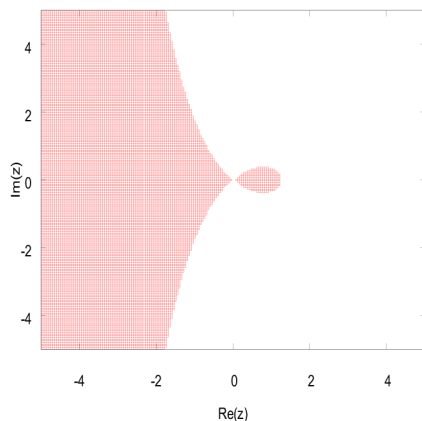
Věta 3.2.4 Metoda je I-stabilní, pokud pro její funkci stability $R(z)$ platí:

(i) $A \cap i\mathbb{R} = \emptyset$.

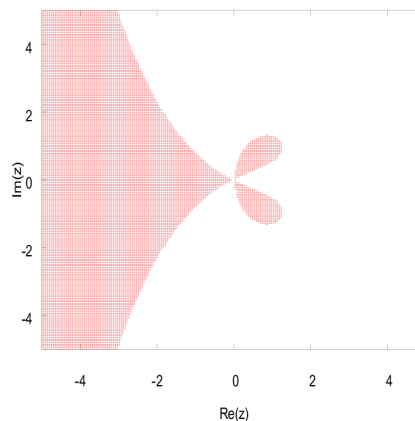
Dále je metoda A-stabilní pokud pro její funkci stability $R(z)$ platí (i) a

(ii) všechny póly $R(z)^{-1}$ (tedy póly $q(z)$) leží uvnitř pozitivní poloviny \mathbb{C}^+ .

Na Obr. 3.6 je zobrazena oblast metody řádu hvězd pro (0,1) Padé aproximaci (tedy implicitní Eulerovu metodu, což je implicitní Taylorova řada prvního řádu), Obr. 3.7 zobrazuje oblast metody řádu hvězd (0,2) Padé aproximaci, což je implicitní Taylorova řada druhého řádu. Obě oblasti jsou podle Věty 3.2.4 I-stabilní.

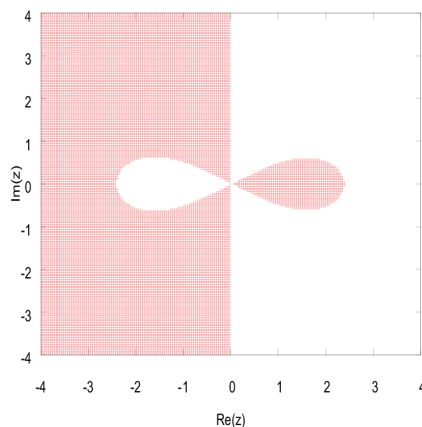


Obrázek 3.6: Oblast řádu hvězd - (0,1) Padé aproximace

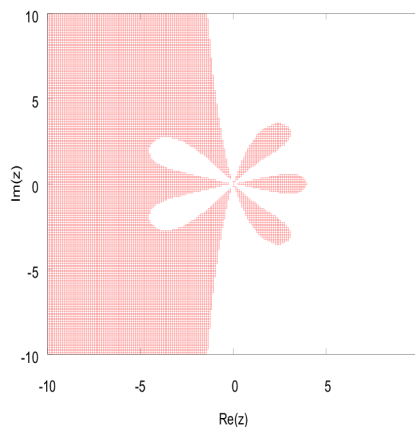


Obrázek 3.7: Oblast řádu hvězd - (0,2) Padé aproximace

Na Obr. 3.8, 3.9 jsou zobrazeny oblasti metody řádu hvězd pro (1,1) Padé aproximaci - lichoběžníková metoda a pro (2,3) Padé aproximaci, což je známá implicitní Runge-Kutta metoda Radau IIA pátého řádu.



Obrázek 3.8: Oblast řádu hvězd - (1,1) Padé aproximace



Obrázek 3.9: Oblast řádu hvězd - (2,3) Padé aproximace

Podrobnější rozbor konceptu metody řádu hvězd nalezneme např. v [HW02].

¹Pro racionální funkci $R_{k,j}(z)$ (3.21) definujeme **nuly** $z_k \in \mathbb{C}$ funkce $R_{k,j}(z)$ takové, že platí $P_{k,j}(z_k) = 0$ a **póly** $z_j \in \mathbb{C}$ funkce $R_{k,j}(z)$ takové, že platí $Q_{k,j}(z_j) = 0$.

3.3 Shrnutí

Tuhé systémy vyžadují při svém řešení numerické metody s velkou oblastí absolutní stability. Po numerické metodě vhodné pro řešení tuhých systémů požadujeme, aby byla **A-stabilní**, respektive **L-stabilní**. Tento požadavek splňují především implicitní numerické metody nižších řádů. Z požadavku stability se jako zajímavá jeví metoda explicitní a implicitní Taylorovy řady. S jejím podrobnějším rozbořem se seznámíme v podkaptitole 5.5.

Zabývali jsme se pouze základním rozdělením stability numerických metod. Existuje ještě mnoho dalších pojmů D-stabilita, B-stabilita, BN-stabilita a další. Pojem stabilita numerických metod je často diskutovaný problém v mnoha literaturách. Podrobněji se s ním můžete seznámit např. v [HNW87, HW02, sti10, Č04, Wil74].

Kapitola 4

Analýza a řešení tuhých systémů

V této kapitole se zaměříme na analýzu vzniku tuhosti a popíšeme možnosti řešení tuhých systémů na několika konkrétních příkladech. S tuhými systémy se můžeme setkat v mnoha případech běžné praxe. Vyskytují se např. v chemické kinetice, teorii řízení, mechanice pevných látek, ale také v předpovědi počasí, biologii, elektrotechnice a mnoha dalších oblastech.

Na následujících příkladech si ukážeme výpočet koeficientu tuhosti, ukážeme si nalezení analytického řešení některých problémů a pokusíme se u nich úpravami “tuhost” odstranit nalezením ekvivalentní SDR “bez tuhosti”. Zvláštní pozornost budeme věnovat vzniku tuhých systémů v elektrických obvodech. Porovnáme možnosti řešení problémů stiff soustav diferenciálních rovnic pomocí klasických numerických metod implementovaných v MATLABu a možnost řešení těchto problémů pomocí Moderní metody Taylorovy řady (programové vybavení TKSL [sof10]).

4.1 Detekce tuhých systémů

Vyděme nejprve ze základní vlastnosti tuhých systémů. Stručně lze charakterizovat tuhé systémy jako diferenciální rovnice, jejichž řešení obsahuje velmi rychle se měnící složky a složky s velmi pomalou změnou. Rychle se měnící složky bývají často nezajímavé, komplikují však výpočet pomalých složek. Kvalitní programy založené na explicitních Runge-Kutta metodách a Adamsových metodách jsou obecně velmi efektivní. Jestliže však pomocí nich chceme řešit tuhý problém, jsou najednou značně neefektivní, dokonce natolik, že jejich použití je zcela nevyhovující. Pro pevnou délku kroku začne často numerické řešení oscilovat a amplituda oscilací explozivně roste. U metod s řízením délky kroku se začne integrační krok značně zmenšovat a doba výpočtu se extrémně prodlouží. Při neúměrně velkém počtu kroků nám výpočet navíc často zkomplikuje i vliv zaokrouhlovací chyby. Proto je nezbytné předem detekovat “tuhost” v řešené SDR. Při zjištění, že se jedná o tuhý systém SDR, zvolíme k jeho řešení vhodné, často implicitní numerické metody.

Většina metod detekce pracuje s Jacobiovou maticí a vlastními čísly soustavy diferenciálních rovnic. Jedna z často uváděných definic tuhých systémů zní [Vit94]:

Definice 4.1.1 *Soustava diferenciálních rovnic je tuhá, jestliže všechna vlastní čísla matice \mathcal{J} mají zápornou reálnou část a koeficient tuhosti (angl. “stiffness ratio”) S je velký.*

Koeficient tuhosti S definujeme následovně:

Definice 4.1.2 *Mějme soustavu n obyčejných diferenciálních rovnic*

$$y'_j = f_j(t, y_1, y_2, \dots, y_n), \quad j = 1, 2, \dots, n. \quad (4.1)$$

Nechť \mathcal{J} je Jacobián SDR (4.1) a λ_j jsou vlastní čísla (obecně časově závislá) Jacobiánu \mathcal{J} . Očíslujme vlastní čísla λ_j tak, aby

$$|Re(\lambda_1)| \leq |Re(\lambda_2)| \leq \dots \leq |Re(\lambda_n)|, \quad (4.2)$$

a položíme

$$\lambda_{min} = \lambda_1, \quad \lambda_{max} = \lambda_n. \quad (4.3)$$

Koeficient tuhosti S je pak dán vztahem

$$S = \frac{|Re(\lambda_{max})|}{|Re(\lambda_{min})|}. \quad (4.4)$$

Problémem je, že není definována žádná přesná hodnota, podle níž bychom mohli rozhodnout, zda je systém tuhý či nikoliv. Pro mnoho tuhých problémů z běžné praxe je koeficient tuhosti S velmi velký (řádově $S > 10^5$).

Na následujícím demonstračním příkladu ukážeme vliv rychle se měnící složky a možnost detekce rostoucí tuhosti v soustavě diferenciálních rovnic pomocí členů Taylorovy řady.

Příklad detekce tuhosti s využitím členů explicitní Taylorovy řady

Mějme soustavu diferenciálních rovnic ve tvaru

$$\begin{aligned} y' &= -ay, & a > 0, \\ z' &= -0.0001z, \end{aligned} \quad (4.5)$$

kde počáteční podmínky jsou $y(0) = 1$, $z(0) = 1$.

Poznamenejme, že známé analytické řešení (4.5) je ve tvaru

$$\begin{aligned} y &= e^{-at}, \\ z &= e^{-0.0001t}. \end{aligned} \quad (4.6)$$

Pro detekci tuhosti v systému (4.5) je nutno vypočítat Jacobiovu matici

$$\mathcal{J} = \begin{pmatrix} -a & 0 \\ 0 & -0.0001 \end{pmatrix},$$

ze které dále určíme vlastní čísla

$$\begin{aligned} \lambda_1 &= -a, \\ \lambda_2 &= -0.0001. \end{aligned}$$

Za předpokladu, že $a > 0.0001$ je koeficient tuhosti

$$S = \frac{|Re(\lambda_{max})|}{|Re(\lambda_{min})|} = \frac{a}{0.0001}. \quad (4.7)$$

Mnoho výpočetních systémů sloužících k řešení tuhých soustav diferenciálních rovnic detekuje tuhé systémy pomocí Jacobiovu matice a vlastních čísel. Zvláště u rozsáhlých soustav

diferenciálních rovnic je však tento způsob značně výpočetně i časově náročný.

Nový způsob detekce tuhých systémů

Jak je uvedeno výše, využití Jacobiovy matice a vlastních čísel k detekci tuhých systémů je výpočetně a časově náročné. Proto **navrhuj a ověřuj k detekci tuhých systémů využití Moderní metody Taylorovy řady**. K detekci tuhých systémů není zapotřebí počítat Jacobiovu matici systému ODR ani vlastní čísla. Detekce je založena přímo na porovnávání členů Taylorovy řady¹. Podrobněji se s rekurentním výpočtem členů explicitní a implicitní Taylorovy řady seznámíme v kapitole 5.

Řešme (4.5) pomocí explicitní Taylorovy řady ve tvaru

$$y_{i+1} = y_i - ah \cdot y_i + \frac{(-ah)^2}{2!} \cdot y_i + \dots + \frac{(-ah)^n}{n!} \cdot y_i, \quad (4.8)$$

$$y_{i+1} = y_i + DY1_i + DY2_i + \dots + DYn_i, \quad (4.9)$$

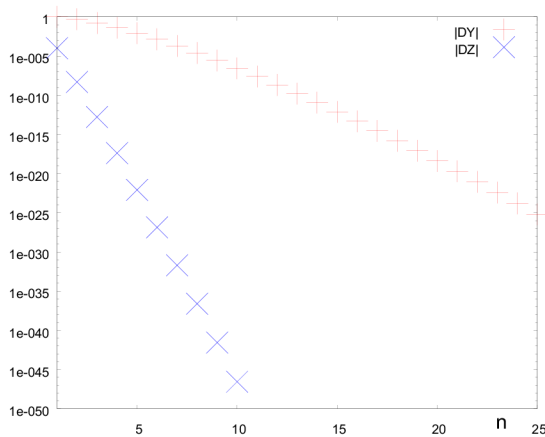
podobně

$$z_{i+1} = z_i - 0.0001h \cdot y_i + \frac{(-0.0001h)^2}{2!} \cdot z_i + \dots + \frac{(-0.0001h)^n}{n!} \cdot z_i, \quad (4.10)$$

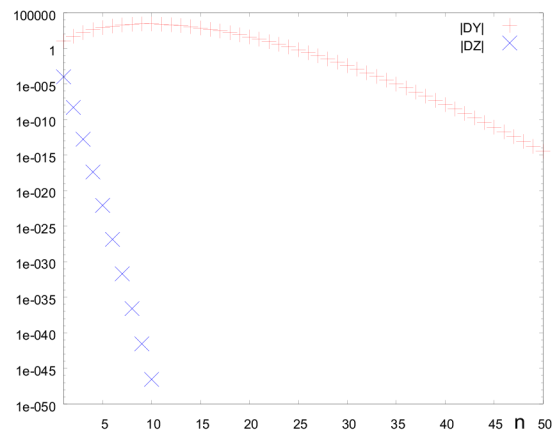
$$z_{i+1} = z_i + DZ1_i + DZ2_i + \dots + DZn_i. \quad (4.11)$$

Analyzujme členy Taylorovy řady (4.9), (4.11) v prvním kroku výpočtu. Členy Taylorovy řady DZ mají rapidně klesající trend. Problém nastává u členů DY s velkou hodnotou konstanty a .

Jak můžeme vidět v Obr. 4.1 pro konstantu $a = 1$ (respektive $S = 10000$) a integrační krok $h = 1$ potřebujeme 15 členů Taylorovy řady k obdržení přesnosti výpočtu $EPS = 10^{-10}$ ("error per step" - nastavitelná lokální diskretizační chyba vzniklá zanedbáním vyšších členů Taylorovy řady). Rostoucí hodnota konstanty a zvyšuje tuhost systému (4.5) a členy Taylorovy řady již nemají klesající charakter. Pro zajištění stability musíme počítat s více členy Taylorovy řady. V Obr. 4.2 je zobrazena velikost členů Taylorovy řady pro hodnotu konstanty $a = 10$ (tedy koeficient tuhosti $S = 100000$). K numerickému výpočtu s přesností $EPS = 10^{-10}$ již potřebujeme 45 členů Taylorovy řady.



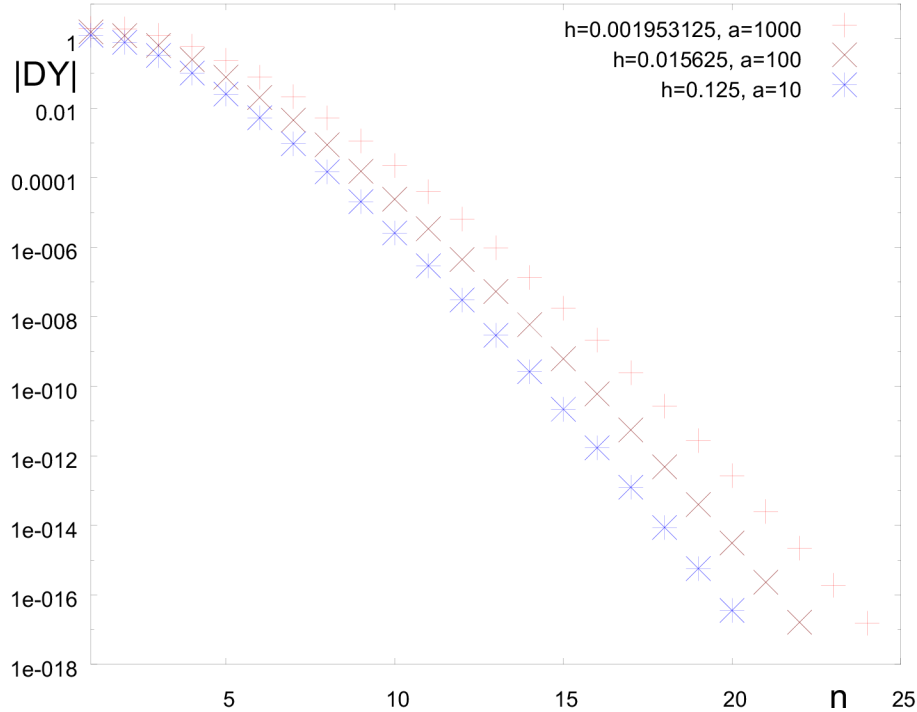
Obrázek 4.1: Členy Taylorovy řady, $a = 1$



Obrázek 4.2: Členy Taylorovy řady, $a = 10$

¹Podrobný rozbor explicitní Taylorovy řady včetně *funkce stability* omezující délku integračního kroku a *oblastí stability* pro různý řád (tedy různý počet členů Taylorovy řady) nalezneme v podkapitole 5.5.

Moderní metoda Taylorovy řady (jak je implementována v TKSL) automaticky (díky odlišným a prudce rostoucím členům Taylorovy řady) detekuje tuhost v systému (4.5). S rostoucí hodnotou konstanty a automaticky snižuje velikost integračního kroku, dokud nemají členy Taylorovy řady DY (4.9) klesající charakter viz Obr. 4.3.



Obrázek 4.3: Členy Taylorovy řady po automatickém snížení integračního kroku

Závěr a zhonocení

Detekce tuhých systémů je základním problémem většiny výpočetních zařízení řešících tuhé systémy. Problém je v přesné definici tuhých systémů, která stále neexistuje. Mnoho výpočetních zařízení řeší detekci tuhých systémů pomocí výpočtu Jacobiovy matice a vlastních čísel, což je však pro rozsáhlé soustavy nelineárních diferenciálních rovnic značně výpočetně náročné.

Zajímavý způsob detekce tuhých soustav diferenciálních rovnic poskytuje explicitní Taylorova řada, u které lze sledovat v průběhu výpočtu členy Taylorovy řady. Výpočetní software TKSL automaticky detekuje tuhost v systému (4.5) s rychle a pomalu se měnící složkou. K detekci využívá porovnání členů Taylorovy řady (4.9), (4.11). Po detekci tuhosti v systému TKSL, který využívá k výpočtu explicitní Taylorovu řadu, automaticky zmenšuje integrační krok, dokud členy Taylorovy řady nemají klesající charakter Obr. 4.3. Vhodnější je po detekci tuhého systému využít implicitní numerickou metodu např. implicitní Taylorovu řadu a její implementaci pomocí ITMRN viz kapitola 5.

4.2 Řešení tuhých systémů s využitím ekvivalentního systému

Jedním z možných přístupů odstranění tuhosti ze soustavy diferenciálních rovnic (SDR), který v práci navrhuji a ověřuji, je **nalezení nového ekvivalentního systému k původní SDR, který již “tuhost” neobsahuje**. Novou SDR, která již tuhost neobsahuje, lze poté řešit běžnými (explicitními) numerickými metodami. Tento nový přístup je možno prezentovat v následujících příkladech.

4.2.1 Problém 1

Mějme soustavou dvou obyčejných diferenciálních rovnic ve tvaru

$$\begin{aligned}y' &= z, \\z' &= -a \cdot y - (a+1) \cdot z, \quad a \in (1, \infty),\end{aligned}\tag{4.12}$$

s počátečními podmínkami $y(0) = 1, z(0) = -1$.

Při zkoumání tuhosti systému (4.12) vypočteme nejprve Jacobiovu matici

$$\mathcal{J} = \begin{pmatrix} 0 & 1 \\ -a & -a-1 \end{pmatrix}$$

a z ní následně vlastní čísla

$$\begin{aligned}\lambda_1 &= -1, \\ \lambda_2 &= -a,\end{aligned}\tag{4.13}$$

pomocí nichž určíme koeficient tuhosti systému (4.12)

$$S = \frac{|\operatorname{Re}(\lambda_{max})|}{|\operatorname{Re}(\lambda_{min})|} = \frac{a}{1}.\tag{4.14}$$

Podle koeficientu tuhosti (4.14) můžeme soudit, že tuhost systému (4.12) roste s hodnotou konstanty a .

Pozn.: Soustava (4.12) je v matematické literatuře velmi často uváděna a je matematicky pro definování tuhosti velmi populární (např. [Vit94]). Doplňme si však, že jednoduchým analytickým výpočtem lze tuhost soustavy eliminovat.

Analytické řešení

Systém (4.12) prepíšeme nejprve do jedné diferenciální rovnice druhého řádu

$$\begin{aligned}y'' &= z', \\y'' &= -a \cdot y - (a+1) \cdot z, \\y'' &= -a \cdot y - (a+1) \cdot y' .\end{aligned}$$

Sestavíme charakteristickou rovnici

$$\begin{aligned}y'' + a \cdot y - (a+1) \cdot y' &= 0, \\ \lambda^2 + (a+1) \cdot \lambda + a &= 0,\end{aligned}$$

z níž vypočteme kořeny

$$\begin{aligned}\lambda_1 &= \frac{-(a+1) + \sqrt{(a+1)^2 - 4a}}{2}, \\ \lambda_2 &= \frac{-(a+1) - \sqrt{(a+1)^2 - 4a}}{2}, \\ \lambda_1 &= \underline{\underline{-1}}, \quad \lambda_2 = \underline{\underline{-a}}.\end{aligned}$$

Kořeny charakteristické rovnice se samozřejmě shodují s vlastními čísly (4.13) Jacobiovy matice systému (4.12).

Očekávané *obecné řešení* je tedy tvaru

$$\begin{aligned} y &= C_1 \cdot e^{\lambda_1 \cdot t} + C_2 \cdot e^{\lambda_2 \cdot t}, \\ y &= C_1 \cdot e^{-t} + C_2 \cdot e^{-a \cdot t}, \\ y' &= -C_1 \cdot e^{-t} + -a \cdot C_2 \cdot e^{-a \cdot t}, \end{aligned}$$

po dosazení počátečních podmínek $y(0) = 1$, $z(0) = y'(0) = -1$

$$\begin{aligned} 1 &= C_1 + C_2, \\ -1 &= -C_1 - a \cdot C_2, \end{aligned}$$

vypočteme konstanty

$$C_1 = 1, \quad C_2 = 0.$$

Partikulární řešení systému (4.12) je tedy

$$\begin{aligned} y &= e^{-t}, \\ z &= -e^{-t}. \end{aligned} \tag{4.15}$$

Těmto rovnicím lze přiřadit následující diferenciální rovnice. Derivujeme (4.15) a přepíšeme do tvaru

$$\begin{aligned} y' &= -e^{-t}, & \Rightarrow & y' = -y, & y(0) &= 1, \\ z' &= -e^{-t} \cdot (-1), & \Rightarrow & z' = -z, & z(0) &= -1. \end{aligned}$$

Vznikla **nová soustava obyčejných diferenciálních rovnic** (4.16), která je ekvivalentní se systémem (4.12) a jeho partikulárním řešením (4.15)

$$\begin{aligned} y' &= -y, & y(0) &= 1, \\ z' &= -z, & z(0) &= -1. \end{aligned} \tag{4.16}$$

Nový systém (4.16) má koeficient tuhosti $S = 1$. Nejedná se již tedy o tuhý systém a tuto soustavu diferenciálních rovnic lze bez problémů řešit běžnými explicitními numerickými metodami.

Ukázka řešení

Řešení tuhého systému (4.12) pro $a = 1 \cdot 10^6$ a interval $t \in \langle 0, 6 \rangle$ (tedy maximální čas udávající interval řešení $TMAX = 6$) pomocí explicitních Runge-Kutta metod (např. MATLAB funkce ODE45 - nutnost nastavení velice malého integračního kroku), trvá řádově desítky minut. Při nasazení metod zpětné derivace (např. MATLAB ODE15s) již výpočet trval 0.9122 sekund.

Řešení upraveného systému s odstraněnou tuhostí (4.16) je již bezproblémové. Explicitní Runge-Kutta (MATLAB ODE45) vyřeší problém během 0.3065 sekund.

Řešení (4.16) pomocí TKSL/C je zobrazeno v Obr. 4.4.

4.2.2 Problém 2

Soustavu diferenciálních rovnic (4.17) nalezneme v [But08b] pod označením “středně tuhý systém”

$$\begin{aligned} y_1' &= -16y_1 + 12y_2 + 16 \cos(t) - 13 \sin(t), & y_1(0) &= 1, \\ y_2' &= 12y_1 - 9y_2 - 11 \cos(t) + 9 \sin(t), & y_2(0) &= 0. \end{aligned} \tag{4.17}$$



Obrázek 4.4: Příklad 1 - TKSL/C

Jacobiova matice SDR (4.17) je ve tvaru

$$\mathcal{J} = \begin{pmatrix} -16 & 12 \\ 12 & -9 \end{pmatrix},$$

vlastní čísla matice \mathcal{J} jsou $\lambda_1 = 0$, $\lambda_2 = -25$.

Analytické řešení

SDR (4.17) přepíšeme do jedné diferenciální rovnice druhého řádu

$$y_1'' = -16y_1' + 12(12y_1 - 9y_2 - 11 \cos(t) + 9 \sin(t)) - 16 \sin(t) - 13 \cos(t), \quad (4.18)$$

vyjádříme y_2 z první diferenciální rovnice SDR (4.17)

$$y_2 = \frac{y_1' + 16y_1 - 16 \cos(t) + 13 \sin(t)}{12},$$

dosadíme do (4.18)

$$y_1'' = -16y_1' - 9(y_1' + 16y_1 - 16 \cos(t) + 13 \sin(t)) + 12(12y_1 - 11 \cos(t) + 9 \sin(t)) - 16 \sin(t) - 13 \cos(t).$$

Po úpravě dostaneme

$$y_1'' = -25y_1' - \cos(t) - 25 \sin(t).$$

Hledáme tedy řešení nehomogenní diferenciální rovnice druhého řádu

$$y_1'' + 25y_1' = -\cos(t) - 25 \sin(t). \quad (4.19)$$

Řešíme nejprve homogenní diferenciální rovnici

$$y_1'' + 25y_1' = 0,$$

charakteristická rovnice je ve tvaru

$$\begin{aligned}\lambda^2 + 25\lambda &= 0, \\ \lambda(\lambda + 25) &= 0, \\ \lambda_1 &= \underline{0}, \lambda_2 = \underline{-25}\end{aligned}$$

Obecné řešení homogenní diferenciální rovnice je ve tvaru

$$\begin{aligned}y_1 &= C_1 e^{\lambda_1 t} + C_2 e^{\lambda_2 t}, \\ y_1 &= C_1 + C_2 e^{-25t}.\end{aligned}$$

Nyní hledáme partikulární integrál pro výraz $-\cos(t) - 25\sin(t)$

$$\begin{aligned}y_{par} &= A \sin(t) + B \cos(t), \\ y'_{par} &= A \cos(t) - B \sin(t), \\ y''_{par} &= -A \sin(t) - B \cos(t).\end{aligned}$$

Dosazením do původní rovnice (4.19)

$$-A \sin(t) - B \cos(t) + 25A \cos(t) - 25B \sin(t) = -\cos(t) - 25 \sin(t),$$

porovnáním koeficientů dostaneme

$$\begin{aligned}-A - 25B &= -25, \\ 25A - B &= -1,\end{aligned}$$

odkud

$$A = 0, B = 1.$$

Obecné řešení soustavy diferenciálních rovnic je ve tvaru

$$\begin{aligned}y_1 &= C_1 + C_2 e^{-25t} + \cos(t), \\ y_2 &= \frac{4}{3}C_1 - \frac{3}{4}C_2 e^{-25t} + \sin(t),\end{aligned}$$

dosazením počátečních podmínek $y_1(0) = 1$ a $y_2(0) = 0$, vypočteme $C_1 = C_2 = 0$. Dostaneme partikulární řešení

$$\begin{aligned}y_1 &= \cos(t), \\ y_2 &= \sin(t).\end{aligned}\tag{4.20}$$

Těmto rovnicím lze přiřadit následující diferenciální rovnice. Derivujeme (4.20) a přepíšeme do tvaru

$$\begin{aligned}y'_1 &= -\sin(t) \Rightarrow y'_1 = -y_2 & y_1(0) &= 1, \\ y'_2 &= \cos(t) \Rightarrow y'_2 = y_1 & y_2(0) &= 0.\end{aligned}$$

Nový systém “bez tuhosti”

Vznikl nový systém obyčejných diferenciálních rovnic (4.21), který je ekvivalentní se systémem (4.17) a jeho partikulárním řešením (4.20)

$$\begin{aligned}y'_1 &= -y_2, & y_1(0) &= 1, \\ y'_2 &= y_1, & y_2(0) &= 0.\end{aligned}\tag{4.21}$$

Nová soustava (4.21) má komplexně sdružená vlastní čísla $\lambda_{1,2} = \pm i$. Nejedná se již tedy o tuhý systém a tuto soustavu lze bez větších obtíží řešit běžnými explicitními metodami.

Ukázka řešení

Ukažme si nyní řešení SDR (4.17) a (4.21) explicitní Eulerovou metodou.

Definujme si nejprve průměrnou absolutní chybu výpočtu

$$\begin{aligned} |E_1| &= \frac{1}{N} \sum_{i=1}^N |y_{1_i} - y_1(i \cdot h)|, \\ |E_2| &= \frac{1}{N} \sum_{i=1}^N |y_{2_i} - y_2(i \cdot h)|, \end{aligned} \tag{4.22}$$

kde N je počet kroků, h je délka integračního kroku a $y_1(t) = \cos(t)$, $y_2(t) = \sin(t)$ jsou známá analytická řešení SDR (4.17) a (4.21).

Tabulka 4.1: Průměrná absolutní chyba výpočtu - explicitní Eulerova metoda

h	SDR1 (4.17)		SDR2 (4.21)	
	$ E_1 $	$ E_2 $	$ E_1 $	$ E_2 $
π	1	1.5708	1	1.5708
$\frac{\pi}{3}$	2.0026	1.82054	0.301126	0.469847
$\frac{\pi}{6}$	2904.18	2178.18	0.301767	0.359263
$\frac{\pi}{9}$	6824.6	5118.47	0.146908	0.218403
$\frac{\pi}{12}$	256592	192444	0.145873	0.153737
$\frac{\pi}{15}$	137205	102904	0.0971319	0.122799
$\frac{\pi}{18}$	501366	376024	0.094438	0.0969944
$\frac{\pi}{21}$	212360	159270	0.0801247	0.0818459
$\frac{\pi}{24}$	43003.1	32252.4	0.0696022	0.0707897
$\frac{\pi}{27}$	4266.14	3199.61	0.0614661	0.0623607
$\frac{\pi}{30}$	135.058	101.292	0.0509704	0.0571341
$\frac{\pi}{33}$	5.41458	4.06118	0.0498279	0.0503644
$\frac{\pi}{36}$	0.0765448	0.0654245	0.0427244	0.0469683

V Tab. 4.1 je zobrazena průměrná chyba výpočtu $|E_1|$, $|E_2|$ pro řešení soustavy (4.17) a (4.21). Řešení je počítáno na intervalu $t \in \langle 0, \pi \rangle$, tedy $TMAX = \pi$, pomocí explicitní Eulerovy metody. Všimněme si, že s rostoucí délkou integračního kroku h roste i průměrná absolutní chyba výpočtu u původní SDR1 (4.17) - tuhý systém. U nové SDR2 (4.21) tento problém nemáme a lze tedy řešit tuto novou SDR2 běžnými explicitními metodami - systém bez tuhosti.

Závěr

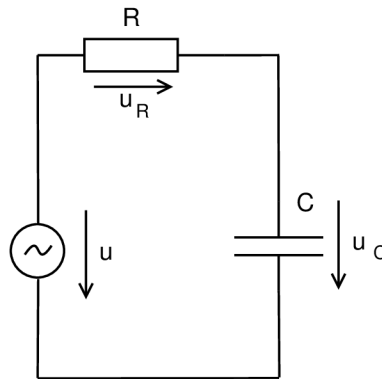
Existují určité tuhé problémy např. (4.12) nebo (4.17), které lze matematickými operacemi převést na ekvivalentní problémy, u nichž je snížena, nebo úplně odstraněna tuhost,

viz (4.16) nebo (4.21). Tento nový přístup umožňuje řešit nově vzniklé ekvivalentní SDR s odstraněnou tuhostí pomocí klasických běžně používaných explicitních numerických metod.

4.3 Příklady v elektrotechnice

V následující kapitole se budeme věnovat vzniku tuhých systémů v elektrických obvodech. Častou příčinou vzniku tuhosti v elektrických obvodech je vnášení parazitních prvků do obvodů. Tento jev si demonstrujeme na následujících příkladech.

4.3.1 RC elektrický obvod



Obrázek 4.5: RC obvod

Mějme jednoduchý elektrický RC obvod z Obr. 4.5, kde napájecí napětí je ve tvaru

$$u = \sin(\omega t).$$

Napětí na kondenzátoru lze popsat diferenciální rovnicí

$$u'_C = \frac{i}{C} \Rightarrow i = u'_C C.$$

Dále platí druhý Kirchhoffův zákon

$$u = u_C + Ri = u_C + u'_C RC,$$

odkud vyjádříme u'_C a získáme výslednou počáteční úlohu

$$u'_C = \frac{1}{RC}u - \frac{1}{RC}u_C, \quad u_C(0) = 0.$$

Hledáme tedy **analytické řešení** úlohy

$$u'_C + au_C = au, \quad u_C(0) = 0, \tag{4.23}$$

kde $a = 1/RC$. Postup řešení je podobný jako v předchozím příkladu 4.2.2 a výsledné analytické řešení je tvaru

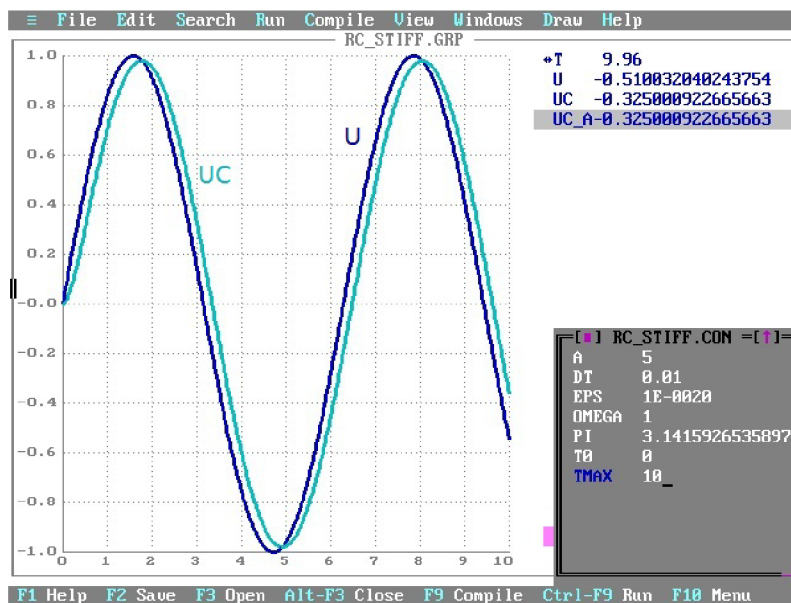
$$u_C = \frac{a\omega}{a^2 + \omega^2} e^{-at} - \frac{a\omega}{a^2 + \omega^2} \cos(\omega t) + \frac{a^2}{a^2 + \omega^2} \sin(\omega t). \tag{4.24}$$

Z analytického řešení (4.24) vyplývá

$$\begin{aligned}\lim_{a \rightarrow \infty} \left(\frac{a\omega}{a^2 + \omega^2} \right) &= 0, \\ \lim_{a \rightarrow \infty} (e^{-at}) &= 0, \\ \lim_{a \rightarrow \infty} \left(\frac{a^2}{a^2 + \omega^2} \right) &= 1,\end{aligned}$$

a tedy platí

$$\lim_{a \rightarrow \infty} (u_C) = \sin(\omega t). \quad (4.25)$$



Obrázek 4.6: RC obvod - TKSL/386: $a = 5, \omega = 1$

Závěr

Výsledek (4.25) lze odvodit z fyzikální podstaty problému. Napětí na kondenzátoru je pro $a \rightarrow \infty$ rovno vstupnímu napětí $u = \sin(\omega t)$ (4.25). **Explicitní numerické metody mají s rostoucí hodnotou konstanty a problém s výpočtem.**

Maximální velikost integračního kroku h (pro hodnotu konstanty $a = 2000$) pro některé explicitní numerické metody nalezneme v Tab. 4.2.

Tabulka 4.2: Maximální velikost h ($EPS = 10^{-20}$, $a = 2000$) - explicitní numerické metody

metoda	h
Eulerova metoda	4.076×10^{-22}
metoda Runge-Kutta 2. řádu	1.36×10^{-12}
metoda Runge-Kutta 4. řádu	5.96×10^{-8}
Taylorova metoda (63 členů)	3.125×10^{-3}

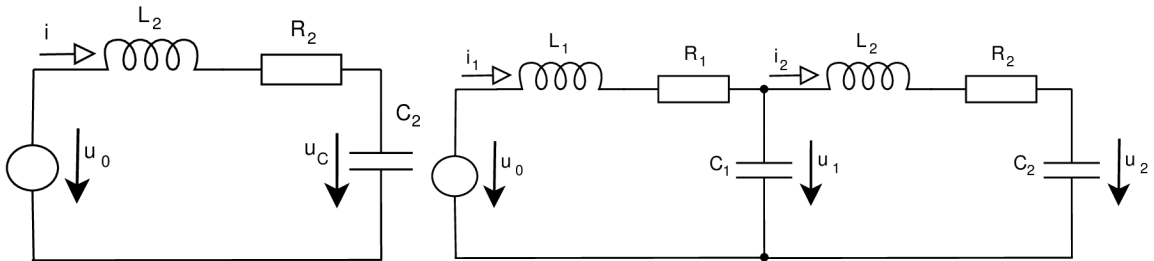
V Obr. 4.6 je zobrazen časový průběh napětí na kondenzátoru u_C , které závisí na vstupním napětí u ($\omega = 1, a = 5$). Funkce u_C (pro malé hodnoty a) je zpožděná za vstupní funkcí u .

Se zvětšováním hodnoty konstanty a se zvyšuje tuhost systému. Výpočet v MATLABu pomocí funkce ODE45 pro $a = 10^5, TMAX = 10$, trvá 259.185 sekund. Detail řešení je zobrazen v příloze B práce v Obr. B.1. Všimněme si malého integračního kroku a chyby výpočtu (Obr. B.1 - dole).

Problém pro velké hodnoty konstanty a lze efektivně řešit pomocí implicitních numerických metod. Viz podkapitola 5.6.3.

4.3.2 RLC elektrický obvod

Při simulaci elektrických obvodů se značně rozdílnými hodnotami prvků často vzniká tuhý systém diferenciálních rovnic. Jako příklad si uvedeme jednoduchý elektrický obvod s prvky L_2, R_2, C_2 (Obr. 4.7 vlevo) rozšířený (tolerance prvků v obvodu) o parazitní prvky L_1, R_1, C_1 (Obr. 4.7 vpravo).



Obrázek 4.7: Elektrický obvod RLC bez parazitních prvků (vlevo) s parazitními prvky (vpravo)

Průběh napětí a proudů v obvodu na Obr. 4.7 vlevo můžeme popsat soustavou diferenciálních rovnic

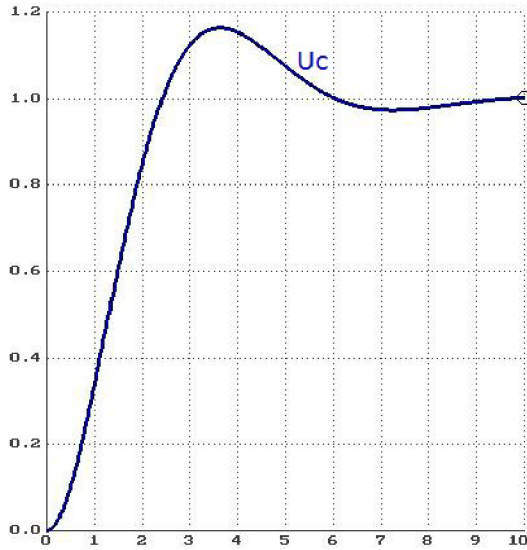
$$\begin{aligned} i' &= \frac{1}{L_2} \cdot (u_0 - u_C - R_2 \cdot i), & i(0) &= 0, \\ u_C' &= \frac{1}{C_1} \cdot i, & u_C(0) &= 0. \end{aligned} \quad (4.26)$$

Očekávaný výsledek (obvodu bez parazitních prvků) průběhu napětí u_C na kapacitě C_2 při parametrech $C_2 = 1\text{F}, L_2 = 1\text{H}, R_2 = 1\Omega$ je zobrazen na Obr. 4.8, napětí na zdroji je $u_0 = 1\text{V}$.

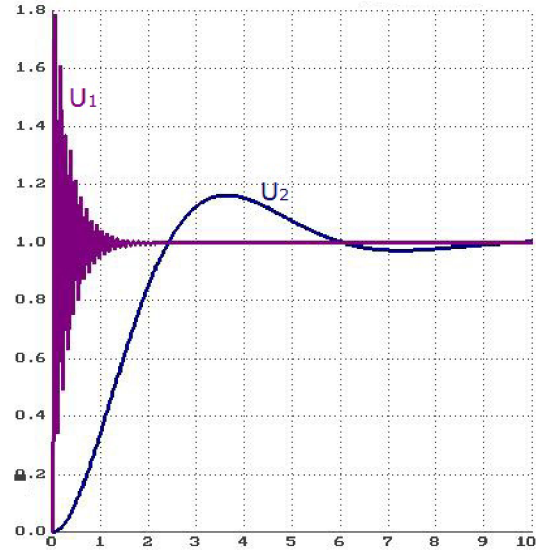
Průběh napětí a proudů v obvodu s parazitními prvky Obr. 4.7 vpravo můžeme popsat diferenciálními rovnicemi

$$\begin{aligned} i_1' &= \frac{1}{L_1} \cdot (u_0 - u_1 - R_1 \cdot i_1), \\ i_2' &= \frac{1}{L_2} \cdot (u_1 - u_2 - R_2 \cdot i_2), \\ u_1' &= \frac{1}{C_1} \cdot (i_1 - i_2), \\ u_2' &= \frac{1}{C_2} \cdot i_2, \end{aligned} \quad (4.27)$$

kde všechny počáteční podmínky jsou nulové a napětí na zdroji je $u_0 = 1\text{V}$. Při volbě značně rozdílných parazitních prvků se může systém diferenciálních rovnic (4.27) snadno stát tuhým systémem.



Obrázek 4.8: Elektrický RLC obvod bez parazitních prvků - TKSL/386, časový průběh napětí u_C



Obrázek 4.9: Elektrický obvod s parazitními prvky - TKSL/386, časový průběh napětí u_1, u_2

Zvolme nejprve parazitní prvky méně rozdílné od původních hodnot ($C_2 = 1\text{F}$, $L_2 = 1\text{H}$, $R_2 = 1\Omega$) např. $C_1 = 0.001\text{F}$, $L_1 = 0.001\text{H}$, $R_1 = 0.006\Omega$. Časový průběh napětí u_1, u_2 je v Obr. 4.9. Při takto zvolených hodnotách se prakticky nezmění průběh napětí u_2 proti původnímu napětí u_C . Rozdíl mezi napětím u_C a u_2 tedy $ERR = u_C - u_2$ je zobrazen v příloze B práce v Obr. B.2, B.3, B.4.

Při volbě značně menších parazitních prvků (např. $C_1 = 1 \cdot 10^{-12}\text{F}$, $L_1 = 1 \cdot 10^{-12}\text{H}$, $R_1 = 1 \cdot 10^{-3}\Omega$) získáme stejný průběh napětí u_1, u_2 , avšak výpočet se značně zpomalí. Elektrický obvod reprezentovaný soustavou diferenciálních rovnic (4.27) se stane tuhým systémem.

Jacobiova matice tohoto systému je

$$\mathcal{J} = \begin{pmatrix} -1 \cdot 10^9 & 0 & -1 \cdot 10^{12} & 0 \\ 0 & -1 & 1 & -1 \\ 1 \cdot 10^{12} & -1 \cdot 10^{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix},$$

Vlastní čísla pak jsou

$$\begin{aligned} \lambda_{1,2} &= -0.5 \pm 0.8657j, \\ \lambda_{3,4} &= -5 \cdot 10^8 \pm 1 \cdot 10^{12}j \end{aligned}$$

Z vlastních čísel získáme koeficient tuhosti

$$S = \frac{5 \cdot 10^8}{5 \cdot 10^{-1}} = 10^9. \quad (4.28)$$

Pro takovýto systém je již řešení pomocí běžných explicitních numerických metod značně neefektivní. Jsme opět nuceni zvolit k řešení implicitní numerickou metodu.

Závěr

Sériový elektrický obvod s prvky L_2, R_2, C_2 (Obr. 4.7 vlevo) rozšířený o parazitní prvky L_1, R_1, C_1 (Obr. 4.7 vpravo) tvoří tuhý systém (4.27), který komplikuje výpočet ustáleného stavu napětí na delším časovém intervalu. Explicitní numerické metody mají problém s výpočtem. Vhodné je použít implicitní numerickou metodu. Možné řešení pomocí implicitní Taylorovy řady nalezneme v podkapitole 5.6.5.

Poznamenejme, že stejný průběh napětí u_2 získáme i při zanedbání parazitních prvků, pak ale lze použít běžných explicitních numerických metod.

4.3.3 Van der Polův oscilátor

Problém Van der Polův oscilátor je pojmenován po svém objeviteli, holandském fyzikovi Bathasar van der Polovi (1920). Jedná se o typ oscilátoru s nelineárním tlumením, který popisuje děj v elektrických obvodech s vakuovou trubicí. Van der Polův oscilátor je popsán diferenciální rovnicí druhého řádu

$$y'' + \mu \cdot (y^2 - 1) \cdot y' + y = 0, \quad \mu > 0, \quad (4.29)$$

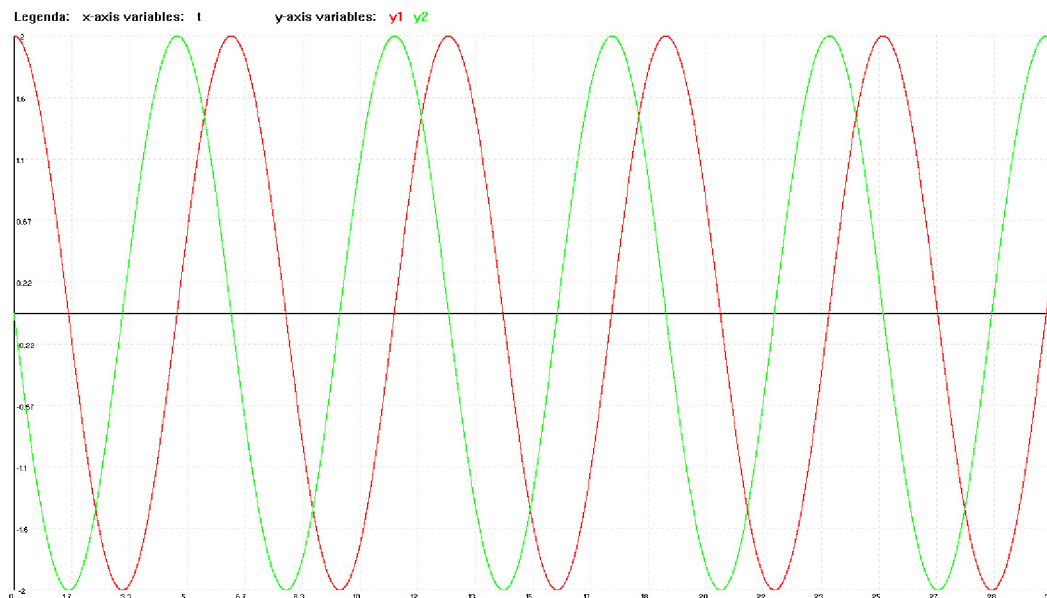
kde μ je parametr udávající sílu nelineárního tlumení. Řešením je periodická funkce, která po přechodné fázi nezávisí na počátečních podmínkách.

Diferenciální rovnice druhého řádu (4.29) lze přepsat na soustavu dvou ODR

$$\begin{aligned} y_1' &= y_2, \\ y_2' &= \mu \cdot (1 - y_1^2) \cdot y_2 - y_1, \quad \mu > 0. \end{aligned} \quad (4.30)$$

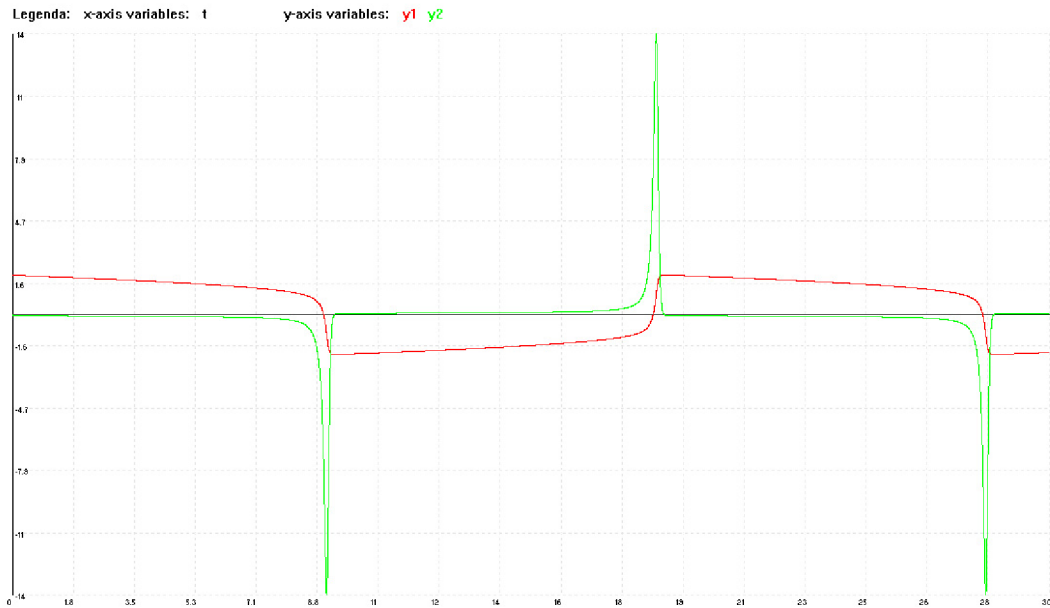
Příznačnou vlastností systému (4.30) je, že malé oscilace jsou zesilovány a velké jsou tlumeny.

Na Obr. 4.10 je zobrazeno řešení systému (4.30) pro počáteční podmínky $y_1(0) = 2$, $y_2(0) = 0$ a parametr $\mu = 0.01$. Jedná se o aplikaci generátoru harmonického signálu.



Obrázek 4.10: Van der Pol - TKSL/C, $\mu = 0.01$

Při volbě počátečních podmínek $y_1(0) = 2$, $y_2(0) = 0$ a koeficientu $\mu = 10$ vznikne tuhý systém, jehož řešení nalezneme v Obr. 4.11.



Obrázek 4.11: Van der Pol - TKSL/C, $\mu = 10$

Koeficient μ tedy značně ovlivňuje koeficient tuhosti systému (4.30). Jacobiova matice systému (4.30) je tvaru

$$\mathcal{J} = \begin{pmatrix} 0 & 1 \\ -2 \cdot \mu \cdot y_1 \cdot y_2 - 1 & \mu \cdot (1 - y_1^2) \end{pmatrix}. \quad (4.31)$$

Systém (4.30) je nelineární, tzn. vlastní čísla Jacobiovy matice (4.31) a tedy i koeficient tuhosti se s časem mění.

Závěr

Explicitní numerické metody mají opět problém s výpočtem soustavy (4.30) s rostoucí hodnotou μ . Vhodné je použít implicitních numerických metod. Zajímavou se v tomto případě jeví implicitní Taylorova řada a její implementace ITMRN viz podkapitola 5.6.6.

Poznamenejme, že Van der Polův oscilátor byl vyvinut pro generování harmonického signálu (Obr. 4.10). Pro generování obdélníkového signálu (velké hodnoty μ a tedy vysoká tuhost systému) je nevhodný (Obr. 4.11). Obdélníkový signál obdržíme snadněji např. použitím astabilního multivibrátoru.

4.4 Převod SLAR na SDR

V této kapitole se zaměříme na vznik tuhosti při převodu soustavy lineárních algebraických rovnic (SLAR) na soustavu diferenciálních rovnic (SDR) s využitím transponované matice viz podkapitola 2.1.3.

Příklad výpočtu stabilního řešení pomocí transponované matice

Řešme soustavu lineárních algebraických rovnic ve tvaru

$$\begin{aligned}x + y + z &= 19, \\2x + 3y + z &= 31, \\3x - 2y + 14z &= 52.\end{aligned}\tag{4.32}$$

SLAR (4.32) převedeme na SDR ve tvaru

$$\begin{aligned}x' &= x + y + z - 19 & x(0) &= 0, \\y' &= 2x + 3y + z - 31 & y(0) &= 0, \\z' &= 3x - 2y + 14z - 52 & z(0) &= 0.\end{aligned}\tag{4.33}$$

Při zkoumání této soustavy (4.33) z hlediska stability vypočteme vlastní čísla soustavy (4.33)

$$\begin{aligned}\lambda_1 &= 14.0425, \\ \lambda_2 &= 0.1111, \\ \lambda_3 &= 3.8464.\end{aligned}\tag{4.34}$$

Všimněme si, že všechna vlastní čísla (4.34) nespĺňují podmínku $\Re(\lambda_1, \lambda_2, \lambda_3) < 0$, tzn. systém diferenciálních rovnic (4.33) je nestabilní a nelze využít k řešení SLAR (4.32).

Provedeme tedy úpravu SDR (4.33) na stabilní systém diferenciálních rovnic pomocí transponované matice

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 2 & 3 & 1 \\ 3 & -2 & 14 \end{pmatrix}, \quad A^T = \begin{pmatrix} 1 & 2 & 3 \\ 1 & 3 & -2 \\ 1 & 1 & 14 \end{pmatrix},$$
$$A^T \cdot A = \begin{pmatrix} 14 & 1 & 45 \\ 1 & 14 & -24 \\ 45 & -24 & 198 \end{pmatrix}.$$

Obdržíme novou stabilní soustavu diferenciálních rovnic ve tvaru

$$\begin{aligned}x' &= -(14x + 1y + 45z - 237) & x(0) &= 0, \\y' &= -(x + 14y - 24z - 8) & y(0) &= 0, \\z' &= -(45x - 24y + 198z - 778) & z(0) &= 0.\end{aligned}\tag{4.35}$$

Podle vztahu

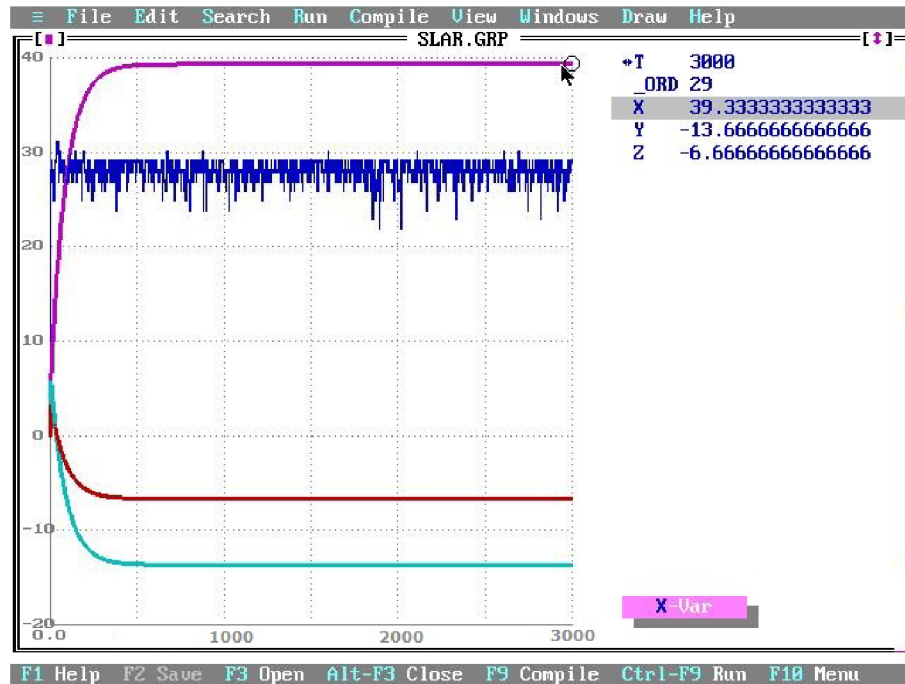
$$|-\mathbf{A}^T \cdot \mathbf{A} - \lambda \cdot \mathbf{E}| = 0,\tag{4.36}$$

obdržíme charakteristickou rovnici pro výpočet vlastních čísel

$$\begin{vmatrix} -14 - \lambda & -1 & -45 \\ -1 & -14 - \lambda & 24 \\ -45 & 24 & -198 - \lambda \end{vmatrix} = 0,$$
$$-\lambda^3 - 226\lambda^2 - 3138\lambda - 36 = 0,$$

odkud

$$\begin{aligned}\lambda_1 &= -0.01, \\ \lambda_2 &= -200.1114, \\ \lambda_3 &= -14.85.\end{aligned}\tag{4.37}$$



Obrázek 4.12: Řešení SDR (4.35) pomocí TKSL/386

Vlastní čísla (4.37) nového SDR (4.35) jsou již záporná $\Re(\lambda_1, \lambda_2, \lambda_3) < 0$ a nový systém diferenciálních rovnic (4.35) tak splňuje podmínku stability.

Nevýhodou převodu SLAR (4.32) na SDR (4.35) pomocí transponované matice je “zvýšení tuhosti” nového systému. Koeficient tuhosti nového systému diferenciálních rovnic (4.35) je ve tvaru

$$S = \frac{|\lambda_2|}{|\lambda_1|} = 20011.14. \quad (4.38)$$

Závěr

Nově vzniklá stabilní SDR (4.35) má vysoký koeficient tuhosti (4.38). Víme, že předpokládané řešení postupně konverguje k hodnotám řešení SLAR (4.32). Řešení SDR (4.35) pomocí TKSL/386 je zobrazeno v Obr. 4.12. Všimněme si nutnosti řešit SDR (4.35) na relativně velkém časovém intervalu ($TMAX = 3000$ - získáme řešení s platností na 14 desetinných míst). Zajímá nás tedy ustálený stav řešení. Vhodné je opět k řešení zvolit implicitní numerickou metodu, která umožní k nalezení ustáleného stavu řešení použít větší integrační krok.

4.5 Newtonova metoda

Všechny předcházející příklady vedou na tuhé systémy diferenciálních rovnic. K řešení tuhých systémů se využívají implicitní numerické metody, které však musí řešit soustavy rovnic v implicitním tvaru. K výpočtu soustavy nelineárních algebraických rovnic se často v praxi využívá Newtonovy iterační metody viz podkapitola 2.1.4.

V následujícím příkladu si ukážeme výpočet diferenciální rovnice pomocí implicitní lichoběžníkové metody využívající k výpočtu Newtonovu metodu.

Hledejme pomocí lichoběžníkové metody numerické řešení diferenciální rovnice [BF04]

$$y' = 5e^{5t}(y - t)^2 + 1, \quad 0 \leq t \leq 1, \quad y(0) = -1. \quad (4.39)$$

Znamé analytické řešení rovnice (4.39) lze vyjádřit ve tvaru

$$y = t - e^{-5t}. \quad (4.40)$$

Lichoběžníková metoda

Připomeňme známý tvar lichoběžníkové metody

$$y_{i+1} = y_i + \frac{1}{2}(hy'_i + hy'_{i+1}).$$

Označíme-li první derivace

$$\begin{aligned} y'_i &= f(t_i, y_i), \\ y'_{i+1} &= f(t_{i+1}, y_{i+1}), \end{aligned}$$

pak dostaneme

$$y_{i+1} = y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_{i+1})).$$

Známe y_i , t_i a t_{i+1} . Hledáme y_{i+1} , řešíme tedy rovnici ve tvaru

$$F(y_{i+1}) = y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_{i+1})) - y_{i+1} = 0. \quad (4.41)$$

Rovnici (4.41) řešíme pomocí Newtonovy metody (viz podkapitola 2.1.4)

$$\begin{aligned} y_{i+1}^{(j+1)} &= y_{i+1}^{(j)} - \frac{F(y_{i+1}^{(j)})}{F'_y(y_{i+1}^{(j)})} = \\ &= y_{i+1}^{(j)} - \frac{y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{(j)})) - y_{i+1}^{(j)}}{\frac{h}{2}f_y(t_{i+1}, y_{i+1}^{(j)}) - 1}, \end{aligned} \quad (4.42)$$

kde $f_y = \partial f(t_{i+1}, y_{i+1}) / \partial y_{i+1}$ a $y_{i+1}^{(0)} = y_i$. Běžně stačí 3 až 4 iterace pro dosažení dostatečné dané přesnosti $TOL = |y_{i+1}^{(j)} - y_{i+1}^{(j+1)}|$.

Řešení problému

Řešme nejprve rovnici (4.39) pomocí implicitní lichoběžníkové metody využívající Newtonovu metodu k výpočtu y_{i+1} dle rovnice (4.42).

$$y_{i+1}^{(j+1)} = y_{i+1}^{(j)} - \frac{y_i + \frac{h}{2}(5e^{5t}(y_i - t)^2 + 5e^{5(t+h)}(y_{i+1}^{(j)} - (t+h))^2 + 2) - y_{i+1}^{(j)}}{\frac{h}{2}(10e^{5(t+h)}y_{i+1}^{(j)} - 10e^{5(t+h)}(t+h)) - 1}, \quad (4.43)$$

Analytický výpočet vyšších derivací je však u některých funkcí složitý. Druhou možností je výpočet derivace pomocí diferenční formule. Do jmenovatele v rovnici (4.42) dosadíme $F'(y) = (F(y + h_N) - F(y))/h_N$ a obdržíme následující vztah

$$y_{i+1}^{(j+1)} = y_{i+1}^{(j)} - \frac{y_i + \frac{h}{2}(f(t_i, y_i) + f(t_{i+1}, y_{i+1}^{(j)})) - y_{i+1}^{(j)}}{F'(y_{i+1}^{(j)})}, \quad (4.44)$$

kde

$$F'(y_{i+1}^{(j)}) = \frac{[y_i + \frac{h}{2}(f(t_{i+1}, y_{i+1}^{(j)} + h_N) + f(t_i, y_i) - (y_{i+1}^{(j)} + h_N))] - [y_i + \frac{h}{2}(f(t_{i+1}, y_{i+1}^{(j)} + f(t_i, y_i)) - y_{i+1}^{(j)})]}{h_N},$$

po úpravě

$$F'(y_{i+1}^{(j)}) = \frac{\frac{h}{2}(f(t_{i+1}, y_{i+1}^{(j)} + h_N) - f(t_{i+1}, y_{i+1}^{(j)})) - (y_{i+1}^{(j)} + h_N) + y_{i+1}^{(j)}}{h_N}.$$

Porovnejme nyní množství iterací potřebných pro výpočet rovnice (4.44) - numerický výpočet derivace při konkrétní délce kroku Newtonovy metody h_N , s počtem iterací potřebných při výpočtu rovnice (4.43) - analytický výpočet derivace.

Tabulka 4.3: Impl. lichoběžníková metoda (4.43), $h = 0.25$, $TOL = 10^{-6}$

t_i	lich. metoda + anal. Newton	$ y(t_i) - y_i $	počet iterací
0.0	-1	0	0
0.25	0.00545572	0.0419605	6
0.5	0.426757	8.84216×10^{-3}	7
0.75	0.729153	2.67056×10^{-3}	8
1.0	0.99402	7.57858×10^{-4}	10

Tab. 4.3 ukazuje výpočet problému (4.39) pomocí implicitní lichoběžníkové metody s analyticky získanou derivací v Newtonově metodě (4.42). Přesnost výpočtu Newtonovy metody je $TOL = |y_{i+1}^{(j)} - y_{i+1}^{(j+1)}| = 10^{-6}$.

Tabulka 4.4: Impl. lichoběžníková metoda (4.44), $h = 0.25$, $TOL = 10^{-6}$, $h_N = 0.01$

t_i	lich. metoda + num. Newton	$ y(t_i) - y_i $	počet iterací
0.0	-1	0	0
0.25	0.00545572	0.0419605	6
0.5	0.426757	8.84216×10^{-3}	7
0.75	0.729153	2.67051×10^{-3}	8
1.0	0.994019	7.57442×10^{-4}	38

V Tab. 4.4, 4.5, 4.6 nalezneme výpočet problému (4.39) pomocí implicitní lichoběžníkové metody s numericky získanou derivací v Newtonově metodě (4.44). Přesnost výpočtu Newtonovy metody je $TOL = |y_{i+1}^{(j)} - y_{i+1}^{(j+1)}| = 10^{-6}$, délka kroku Newtonovy metody $h_N = 0.01$, $h_N = 0.001$, $h_N = 0.0001$, interval výpočtu $TMAX = 1$.

Závěr

Snižování $h_N < 0.0001$ již nemá vliv na počet iterací. Porovnáním Tab. 4.3 s Tab. 4.6

Tabulka 4.5: Impl. lichoběžníková metoda (4.44), $h = 0.25$, $TOL = 10^{-6}$, $h_N = 0.001$

t_i	lich. metoda + num. Newton	$ y(t_i) - y_i $	počet iterací
0.0	-1	0	0
0.25	0.00545572	0.0419605	6
0.5	0.426757	8.84216×10^{-3}	6
0.75	0.729153	2.67055×10^{-3}	7
1.0	0.99402	7.5786×10^{-4}	9

Tabulka 4.6: Impl. lichoběžníková metoda (4.44), $h = 0.25$, $TOL = 10^{-6}$, $h_N = 0.0001$

t_i	lich. metoda + num. Newton	$ y(t_i) - y_i $	počet iterací
0.0	-1	0	0
0.25	0.00545572	0.0419605	6
0.5	0.426757	8.84216×10^{-3}	6
0.75	0.729153	2.67056×10^{-3}	7
1.0	0.99402	7.57858×10^{-4}	8

zjistíme, že numerický výpočet derivace při vhodně malém kroku v diferenční formuli ($h_N = 0.0001$) u Newtonovy metody je výpočetně méně náročný v porovnání s analytickým řešením derivace. Celkový počet iterací u analytického výpočtu derivace Tab. 4.3 je 31. Výhodnější numerické řešení pomocí diferenční formule s $h_N = 0.0001$ je zobrazeno v Tab. 4.6 a potřebuje k výpočtu 27 iterací.

Podrobnější rozbor Newtonovy metody použité v implicitní Taylorově metodě (ITMRN) nalezneme v podkapitole 5.5.1 a 5.6.

4.6 Shrnutí

Řešení tuhých systémů diferenciálních rovnic patří v současné době stále mezi složité problémy. Sofistikované programy se snaží vypořádat s detekcí tuhých systémů (časté přepočítávání Jacobiovy matice). Většina programů však nabízí uživateli více numerických metod a zvolení vhodné numerické metody nechávají zcela na “zkušeném” uživateli. Nechceme být však jen uživateli, kteří náhodně dosadí do formule a čekají, jak to dopadne.

Rozbory a dílčí závěry kapitoly 4 práce ukazují a dávají návod, jak posílit odbornou erudici uživatele při volbě vhodné numerické metody k řešení tuhých systémů diferenciálních rovnic.

Kapitola 5

Moderní metody Taylorovy řady

There are at least two ways to combat stiffness. One is to design a better computer, the other, to design a better algorithm.

(H.Lomax, in Aiken 1985)

5.1 Úvod

Pod pojmem “Moderní metoda Taylorovy řady” si můžeme představit numerickou metodu využívající Taylorovu řadu (rekurentní výpočet členů), která automaticky kontroluje velikost integračního kroku v závislosti na stabilitě numerické metody, používá daný počet členů Taylorovy řady v závislosti na přesnosti výpočtu, lze paralelizovat na vhodné hardwarové architektuře, používá víceslovní aritmetiku atd.

Mezi Moderní metody Taylorovy řady lze zařadit i implicitní Taylorovu řadu, která je vhodná především pro řešení tuhých systémů.

Důležitou součástí metody, tak jak je implementována např. v TKSL [sof10], je automatické nastavení řádu integrační metody, to znamená, že se používá tolik členů Taylorovy řady, kolik je potřeba na požadovanou přesnost výpočtu. Je proto běžné, že se při výpočtu v jednotlivých krocích konstantní délky používá proměnný řád metody. Analogicky je možno pro daný řád metody modifikovat délku použitého integračního kroku. Tato skutečnost pozitivně ovlivňuje stabilitu a rychlost výpočtu.

“Moderní metoda Taylorovy řady” má velmi příznivé paralelní vlastnosti. Mnoho výpočetních operací je vzájemně nezávislých, takže tyto výpočty mohou být prováděny paralelně, nezávisle v oddělených procesorech paralelních výpočetních systémů. Tento paralelní přístup se experimentálně ověřuje.

Nezbytnou součástí “Moderní metody Taylorovy řady” je automatická transformace zadání. Původně zadaná soustava nelineárních diferenciálních rovnic se automaticky transformuje na polynomiální tvar, t.j. na tvar, u kterého lze snadno rekurentně vypočítávat jednotlivé členy Taylorovy řady.

Vzhledem k tomu, že se při výpočtu transformované soustavy (po provedené automatické transformaci zadání) provádějí jen základní matematické operace (+, -, *, /), lze pro jejich realizaci navrhnout jednoduché specializované elementární procesory a získat paralelní výpočetní systém s relativně jednoduchou architekturou (první experimenty již byly provedeny pomocí hradlového pole Xilinx FPGA).

Pro Moderní metodu Taylorovy řady je charakteristické, že se pro zadaný integrační krok přesnost výpočtu zvyšuje s počtem členů Taylorovy řady. Toto zvyšování přesnosti však

není neomezené. Pro daný integrační krok vždy existuje saturovaná chyba výpočtu závislá na délce slova aritmetické jednotky. V některých případech může být saturovaná chyba výpočtu zmenšena zjemněním integračního kroku, nebo rozšířením délky slova aritmetické jednotky. Pomocí mnoha experimentů bylo dokázáno, že v mnoha případech efekt rozšíření délky slova aritmetické jednotky je významnější, než efekt zjemňování integračního kroku.

Moderní metoda Taylorovy řady byla zatím zkoumána a implementována v TKSL **pouze v explicitní podobě**. Ve své práci se budu nově zabývat možností použití implicitní Taylorovy řady k řešení tuhých systému diferenciálních rovnic.

5.2 Současný stav

S rozvojem výpočetní techniky v druhé polovině 20. století se začaly vyvíjet složitější a výpočetně náročnější algoritmy. Prvními “průkopníky”, kteří naznačili možnost rekurentního výpočtu členů Taylorovy řady v kanonickém tvaru, byli Gibbons [Gib60] a Moore [R.A66]. Dalšími známými pracemi zabývajícími se automatizovaným derivováním jsou Barton [BWZ71], Corliss a Chang [CC82]. Možností proměnlivého kroku a řádu (“*variable-step variable-order*” VSVO) v průběhu výpočtu se zabývá ve svých pracích Robert Barrio [BBL03, Bar05, BBL05]. Možnost řešení tuhých systémů pomocí implicitní Taylorovy řady je nastíněna v Barton [Bar80], Kirlinger a Corliss [KC91], Scott [Sco00]. První pokusy s programováním rekurentní Taylorovy řady byly prováděny ve FORTRANu [Hal83]. V současné době se začínají implementovat jednotlivé pokusy v C/C++ [JZ05]. Paralelní výpočet a víceslovní aritmetika se využívá v programovém vybavení TKSL/C [sof10].

Dalšími zajímavými pracemi jsou např. řešení diferenciálních algebraických rovnic Nedialkov a Pryce [NP05, NP07], možnosti validace existence jednoznačného řešení pomocí Taylorovy řady a tzv. “*wrapping*” efektu Nedialkov [NJC99, NJP02, JN02], kombinace Milstein-Taylorovy numerické metody pro řešení stochastických diferenciálních rovnic [TB01], numerický výpočet vyšších derivací do čtvrtého řádu [MM03a, MM03b], kombinace lichoběžníkové metody s Taylorovou řadou [CWH06, CZG10].

5.3 Rekurentní výpočet

Nedílnou součástí Moderních metod Taylorovy řady je rekurentní výpočet členů. Před objevením digitálních počítačů bylo analytické počítání členů Taylorovy řady (především odvozování vyšších derivací) považováno v mnoha případech za příliš komplikované. Z tohoto důvodu se Taylorova řada delší dobu nevyužívala. V mnoha moderních knihách o numerické a aplikované matematice je konstatováno, že Taylorova řada je z teoretického a koncepčního hlediska velice zajímavá, avšak může být použita jen v určitých speciálních případech. Obecné použití integračního algoritmu je příliš složité. V práci se zabývám běžnými “Technickými problémy”, pro jejichž řešení lze Taylorovu řadu úspěšně využít [Mik00].

Mějme explicitní Taylorův rozvoj aproximující funkci $y(t)$ ve tvaru

$$y(t) = y(t_i + h) = y(t_i) + hy'(t_i) + \frac{h^2}{2}y''(t_i) + \dots + \frac{h^m}{m!}y^{(m)}(t_i) + \dots, \quad (5.1)$$

kde $h = t - t_i$.

Předpokládejme, že $y(t_i)$ je známá hodnota, dále používáme značení $y^{(m)}$ pro $d^m y/dt^m|_{t_i}$ ($m = 0, 1, 2, \dots$), f pro $f(t, y(t))|_{t_i}$, f_t pro $\partial f(t, y(t))/\partial t|_{t_i}$ a f_y pro $\partial f(t, y(t))/\partial y|_{t_i}$ atd. Formální vyjádření členů Taylorovy řady pak můžeme zapsat

$$\begin{aligned}
y &= \textit{konstanta} , \\
y' &= f , \\
y'' &= f_t + f_y f , \\
y''' &= f_{tt} + 2f_{ty}f + f_t f_y + f_{yy}f^2 + f_y^2 f , \\
&\vdots
\end{aligned}$$

Aplikujme nyní Taylorovu řadu na konkrétní problém $y' = y^2 + t^{10}$, parciální derivace v bodě $(t, y(t))|_{t_i}$ jsou tedy $f = y^2 + t^{10}$, $f_t = 10t^9$, $f_y = 2y$, $f_{tt} = 90t^8$, $f_{yy} = 2$, $f_{ty} = 0$, atd.

$$\begin{aligned}
y &= \textit{konstanta} , \\
y' &= y^2 + t^{10} , \\
y'' &= 10t^9 + 2y(y^2 + t^{10}) , \\
y''' &= 90t^8 + 10t^9 2y + 2(y^2 + t^{10})^2 + (2y)^2(y^2 + t^{10}) , \\
&\vdots
\end{aligned}$$

Je zřejmé, že i v takto jednoduchém příkladě je složité vyjádřit obecně vyšší derivace. Podíváme-li se však podrobněji na daný příklad, najdeme určitou možnost vyjádření pomocí předcházejících derivací

$$\begin{aligned}
y &= \textit{konstanta} , \\
y' &= y \cdot y + t^{10} , \\
y'' &= y \cdot y' + y' \cdot y + 10t^9 , \\
y''' &= y \cdot y'' + 2y' \cdot y' + y'' \cdot y + 90t^8 , \\
y^{(4)} &= y \cdot y''' + 3y' \cdot y'' + 3y'' \cdot y' + y''' \cdot y + 720t^7 , \\
&\vdots \\
y^{(m+1)} &= \sum_{j=0}^m \left(\frac{m!}{j!(m-j)!} \right) \cdot y^{(j)} \cdot y^{(m-j)} , \quad (m \geq 11) .
\end{aligned}$$

Výhodou tohoto zápisu je možnost rekurentního výpočtu.

5.4 Tvořící rovnice

Pomocí známých pravidel derivování můžeme odvodit rekurentní výpočet vyšších derivací (tedy členů Taylorovy řady) pro základní funkce jako jsou sin, cos, exp, log, atd. a jednoduché aritmetické operace.

Každé programové vybavení využívající automatického rekurentního výpočtu Taylorovy řady, musí obsahovat následující pravidla:

- Funkce $f_i(t, y_1, y_2, \dots, y_n)$ pravých stran soustavy n obyčejných diferenciálních rovnic jsou složeny z konečného počtu elementárních funkcí a konečného počtu jednoduchých aritmetických operací [Mik00].
- Všechny mezivýpočty a všechny generující funkce vedou na pomocné proměnné.
- Operandů mezivýpočtových operací a generujících funkcí využívají pomocné proměnné, x, y_1, y_2, \dots, y_n a konstanty.

Použití těchto pravidel vede k automatickému rozložení pravých stran soustavy ODR na jednodušší výrazy, na které lze aplikovat známá pravidla derivování.

Vše si demonstrujeme na následujícím příkladu

$$y' = y \cdot \exp(c \cdot y) + \frac{\cos(y)}{(t + y)}, \quad y(0) = y_0, \quad (5.2)$$

kde c je konstanta.

Pravou stranu diferenciální rovnice (5.2) rozložíme na pomocné proměnné $y_{T1}, y_{T2}, \dots, y_{T8}$

$$\begin{aligned} y_{T1} &= c \cdot y, \\ y_{T2} &= \exp(y_{T1}), \\ y_{T3} &= y \cdot y_{T2}, \\ y_{T4} &= \sin(y), \\ y_{T5} &= \cos(y), \\ y_{T6} &= t + y, \\ y_{T7} &= y_{T5}/y_{T6}, \\ y_{T8} &= y_{T3} + y_{T7}, \\ y' &= y_{T8}. \end{aligned}$$

Následně pomocí základních pravidel derivování sestrojíme “tvořící diferenciální rovnice” $y'_{T1}, y'_{T2}, \dots, y'_{T8}, y''_{T1}, y''_{T2}, \dots, y''_{T8}$

$$\begin{aligned} y'_{T1} &= c \cdot y', \\ y'_{T2} &= y'_{T1} \cdot \exp(y_{T1}) = y'_{T1} \cdot y_{T2}, \\ y'_{T3} &= y' \cdot y_{T2} + y \cdot y'_{T2}, \\ y'_{T4} &= y' \cdot \cos(y) = y' \cdot y_{T5}, \\ y'_{T5} &= y' \cdot (-\sin(y)) = -y' \cdot y_{T4}, \\ y'_{T6} &= 1 + y', \\ y'_{T7} &= \frac{y'_{T5} \cdot y_{T6} - y_{T5} \cdot y'_{T6}}{y_{T6}^2} = \frac{y'_{T5} - y_{T7} \cdot y'_{T6}}{y_{T6}}, \\ y'_{T8} &= y'_{T3} + y'_{T7}, \\ y'' &= y''_{T8}, \end{aligned}$$

$$\begin{aligned} y''_{T1} &= c \cdot y'', \\ y''_{T2} &= y''_{T1} \cdot y_{T2} + y'_{T1} \cdot y'_{T2}, \\ y''_{T3} &= y'' \cdot y_{T2} + 2 \cdot y' \cdot y'_{T2} + y \cdot y''_{T2}, \\ y''_{T4} &= y'' \cdot y_{T5} + y' \cdot y'_{T5}, \\ y''_{T5} &= -y'' \cdot y_{T4} - y' \cdot y'_{T4}, \\ y''_{T6} &= y'', \\ y''_{T7} &= \frac{y''_{T5} - 2 \cdot y'_{T7} \cdot y'_{T6} - y_{T7} \cdot y''_{T6}}{y_{T6}}, \\ y''_{T8} &= y''_{T3} + y''_{T7}, \\ y''' &= y'''_{T8}. \end{aligned}$$

Obecně lze tedy vyjádřit vyšší derivace tvořících diferenciálních rovnic $y_{T1}^{(m)}, y_{T2}^{(m)}, \dots, y_{T8}^{(m)}$

$$\begin{aligned}
m \geq 3: \quad y_{T1}^{(m)} &= c \cdot y^{(m)} \quad , \\
m \geq 3: \quad y_{T2}^{(m)} &= \sum_{j=0}^{m-1} \binom{m-1}{j} \cdot y_{T1}^{(m-j)} \cdot y_{T2}^{(j)} \quad , \\
m \geq 3: \quad y_{T3}^{(m)} &= \sum_{j=0}^m \binom{m}{j} \cdot y^{(m-j)} \cdot y_{T2}^{(j)} \quad , \\
m \geq 3: \quad y_{T4}^{(m)} &= \sum_{j=0}^{m-1} \binom{m-1}{j} \cdot y^{(m-j)} \cdot y_{T5}^{(j)} \quad , \\
m \geq 3: \quad y_{T5}^{(m)} &= - \sum_{j=0}^{m-1} \binom{m-1}{j} \cdot y^{(m-j)} \cdot y_{T4}^{(j)} \quad , \\
m \geq 3: \quad y_{T6}^{(m)} &= y^{(m)} \quad , \\
m \geq 3: \quad y_{T7}^{(m)} &= \frac{y_{T5}^{(m)} - \sum_{j=1}^m \binom{m}{j} \cdot y_{T7}^{(m-j)} \cdot y_{T6}^{(j)}}{y_{T6}} \quad , \\
m \geq 3: \quad y_{T8}^{(m)} &= y_{T3}^{(m)} + y_{T7}^{(m)} \quad , \\
m \geq 3: \quad y^{(m+1)} &= y_{T8}^{(m)} \quad .
\end{aligned}$$

Závěr

Největším problémem u řešení problémů pomocí Taylorovy metody je nalezení vyšších derivací daných funkcí. Existuje určitá skupina “Technických problémů” [Mik00], které lze pomocí “tvořících diferenciálních rovnic” a speciálních mezivýpočtů převést na konečný polynom, na který lze poté úspěšně implementovat rekurentní výpočet vyšších derivací (tedy rekurentní výpočet vyšších členů Taylorovy řady).

V následujících částech práce si rekurentní možnosti výpočtu členů Taylorovy řady budeme demontrovat na konkrétních příkladech.

5.5 Stabilita, konvergence a konzistence

Zaměřme se nyní na prozkoumání stability, konvergence a konzistence explicitní a nově také implicitní Taylorovy řady.

Připomeňme si nejprve podmínky *konzistence* viz Def. 5.5.1 a *konvergence* viz Def. 5.5.2 jednokrokové numerické metody.

Definice 5.5.1 *Jednokroková metoda s lokální chybou vzniklou zanedbáním vyšších členů Taylorovy řady (local truncat error) $\tau_i(h)$ v i -tém kroku je **konzistentní** s diferenciální rovnicí, pokud*

$$\lim_{h \rightarrow 0} \max_{1 \leq i \leq N} |\tau_i(h)| = 0.$$

Definice 5.5.2 Řekneme, že jednokroková metoda je **konvergentní**, pokud

$$\lim_{h \rightarrow 0} \max_{1 \leq i \leq N} |y_i - y(t_i)| = 0,$$

kde $y(t_i)$ značí hodnotu přesného řešení diferenciální rovnice a y_i je její aproximace získaná numerickou metodou.

5.5.1 Dahlquistův problém

Zaměříme se opět na problém z podkapitoly 3.1. Mějme diferenciální rovnici ve tvaru

$$y' = \lambda y, \quad y(0) = 1, \quad \lambda < 0, \quad (5.3)$$

známou také pod názvem *Dahlquistův problém* [HW02].

Analytické řešení rovnice (5.3) je ve tvaru

$$y = y(0)e^{\lambda t}. \quad (5.4)$$

S rostoucí hodnotou konstanty $|\lambda|$ mají numerické metody problém s řešením počáteční úlohy (5.3). Rozeberme řešení diferenciální rovnice (5.3) pomocí explicitního a implicitního Taylorova rozvoje.

Explicitní Taylorova řada

Připomeňme si tvar explicitního Taylorova rozvoje (viz podkapitola 2.2.2)

$$y_{i+1} = y_i + h y'_i + \frac{h^2}{2!} y''_i + \frac{h^3}{3!} y'''_i + \cdots + \frac{h^n}{n!} y_i^{(n)}, \quad (5.5)$$

po dosazení derivací obdržíme

$$\begin{aligned} y_{i+1} &= y_i + h \lambda y_i + \frac{h^2}{2} \lambda^2 y_i + \frac{h^3}{3!} \lambda^3 y_i + \cdots + \frac{h^n}{n!} \lambda^n y_i, \\ y_{i+1} &= \left(1 + h \lambda + \frac{h^2}{2} \lambda^2 + \frac{h^3}{3!} \lambda^3 + \cdots + \frac{h^n}{n!} \lambda^n\right) y_i. \end{aligned}$$

Funkce stability explicitní Taylorovy řady je ve tvaru

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \cdots + \frac{z^n}{n!}, \quad (5.6)$$

kde $z = h\lambda$.

Implicitní Taylorova řada

Připomeňme si tvar implicitního Taylorova rozvoje viz podkapitola 2.2.2

$$y_{i+1} = y_i + h y'_{i+1} - \frac{h^2}{2!} y''_{i+1} + \frac{h^3}{3!} y'''_{i+1} - \cdots - \frac{(-h)^n}{n!} y_{i+1}^{(n)}, \quad (5.7)$$

po dosazení derivací obdržíme

$$\begin{aligned} y_{i+1} &= y_i + h \lambda y_{i+1} - \frac{h^2}{2!} \lambda^2 y_{i+1} + \frac{h^3}{3!} \lambda^3 y_{i+1} - \cdots - \frac{(-h)^n}{n!} \lambda^n y_{i+1}, \\ y_{i+1} &= \left(\frac{1}{1 - h \lambda + \frac{h^2}{2!} \lambda^2 - \frac{h^3}{3!} \lambda^3 + \cdots + \frac{(-h)^n}{n!} \lambda^n} \right) y_i. \end{aligned}$$

Funkce stability implicitní Taylorovy řady je ve tvaru

$$R(z) = \frac{1}{1 - z + \frac{z^2}{2} - \frac{z^3}{3!} + \cdots + \frac{(-z)^n}{n!}}, \quad (5.8)$$

kde $z = h\lambda$.

Oblasti stability

Vycházíme z požadavku (3.5) konvergence numerického řešení počáteční úlohy (5.3):

$$|y_{i+1}| \leq |y_i|. \quad (5.9)$$

Tedy funkce stability $R(z) = y_{i+1}/y_i$ musí splňovat podmínku

$$|R(z)| \leq 1. \quad (5.10)$$

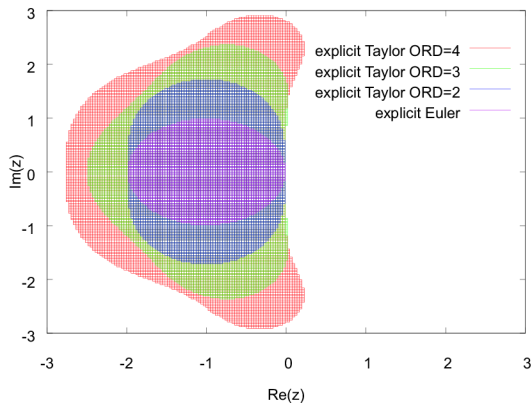
Podle (5.6) je oblast absolutní stability explicitní Taylorovy řady

$$\left| 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \dots + \frac{z^n}{n!} \right| \leq 1. \quad (5.11)$$

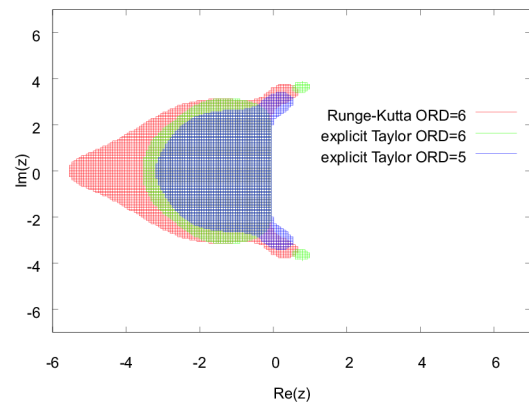
Podobně podle (5.8) je oblast absolutní stability implicitní Taylorovy řady

$$\left| \frac{1}{1 - z + \frac{z^2}{2} - \frac{z^3}{3!} + \dots + \frac{(-z)^n}{n!}} \right| \leq 1. \quad (5.12)$$

V Obr. 5.1, 5.2, B.5 (v příloze B práce) jsou zobrazeny oblasti absolutní stability explicitní Taylorovy řady (vyšrafovaná část). Obecně pokud platí pro počet členů Taylorovy řady $n \rightarrow \infty$, oblast absolutní stability (5.11) pokryje celou zápornou část komplexní poloroviny a explicitní Taylorova řada se stane A-stabilní.



Obrázek 5.1: Oblast stability explicitní Taylorova řada $ORD = 1, 2, 3, 4$

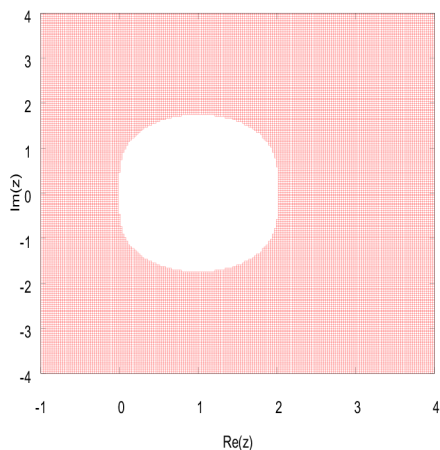


Obrázek 5.2: Oblast stability explicitní Taylorova řada $ORD = 5, 6$

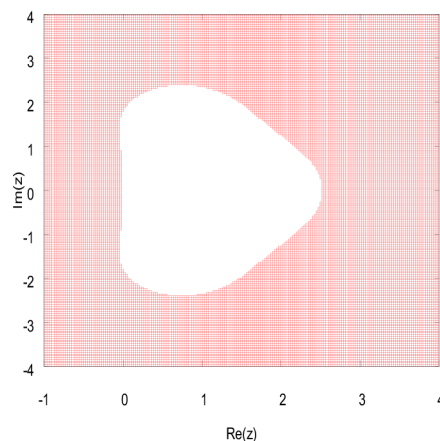
V Obr. 5.3, 5.4, 5.5, 5.6, B.6 (v příloze B práce) jsou zobrazeny oblasti absolutní stability implicitní Taylorovy řady (vyšrafovaná část). Poznamenejme, že implicitní Eulerova metoda je implicitní Taylorova řada prvního řádu ($ORD = 1$) viz Obr. 3.5. Implicitní Taylorova řada prvního a druhého řádu je L-stabilní (je splněna podmínka A-stability, kde oblast absolutní stability pokrývá celou zápornou část komplexní poloroviny). Při vyšším počtu členů implicitní Taylorovy řady se začnou objevovat menší oblasti nestability v záporné části komplexní poloroviny v blízkosti osy $Im(z)$. S rostoucím počtem členů Taylorovy řady se tyto oblasti zmenšují a vzdalují k nekonečnu viz Obr. 5.7, 5.8.

Implicitní Taylorova řada navíc splňuje limitu L-stabilní metody (5.13).

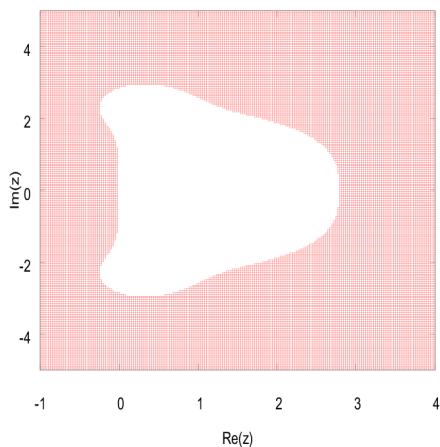
$$\lim_{z \rightarrow -\infty} \frac{1}{1 - z + \frac{z^2}{2} - \frac{z^3}{3!} + \dots + \frac{(-z)^n}{n!}} = 0. \quad (5.13)$$



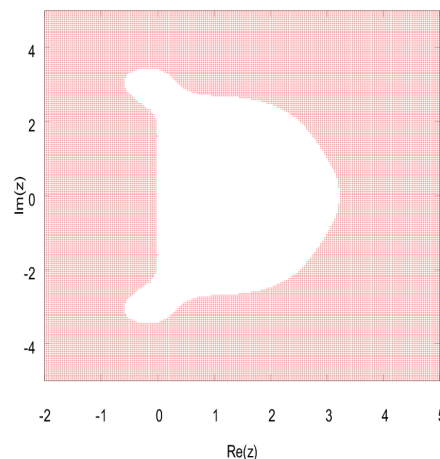
Obrázek 5.3: Oblast stability implicitní Taylorova řada $ORD = 2$



Obrázek 5.4: Oblast stability implicitní Taylorova řada $ORD = 3$



Obrázek 5.5: Oblast stability implicitní Taylorovy řady $ORD = 4$



Obrázek 5.6: Oblast stability implicitní Taylorovy řady $ORD = 5$

Porovnání explicitní a implicitní Taylorovy řady

Zaměřme se nyní na porovnání explicitní (5.14) a implicitní (5.15) Taylorovy řady z pohledu absolutní chyby výpočtu v prvním kroku řešení. Mějme

$$y(t_0 + h) = R_E(z)y_0, \quad (5.14)$$

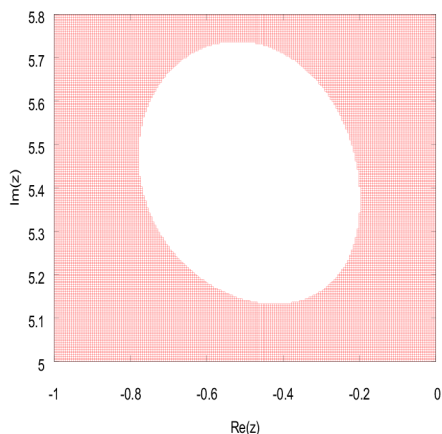
$$y(t_0 + h) = R_I(z)y_0, \quad (5.15)$$

kde

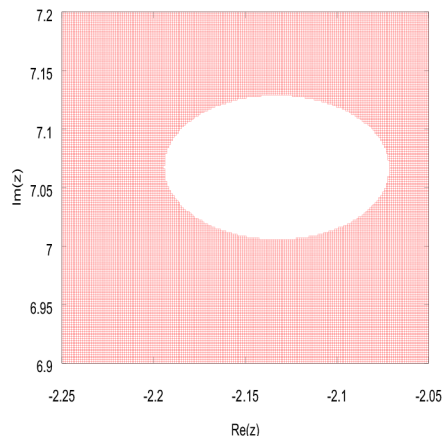
$$R_E(z) = \sum_{i=0}^{i=n} \frac{z^i}{i!}, \quad (5.16)$$

$$R_I(z) = \left(\sum_{i=0}^{i=n} \frac{(-z)^i}{i!} \right)^{-1}, \quad (5.17)$$

pro z stále platí $z = h\lambda$.



Obrázek 5.7: Oblast stability impl. Taylorovy řady $ORD = 11$ v blízkosti osy $Im(z)$



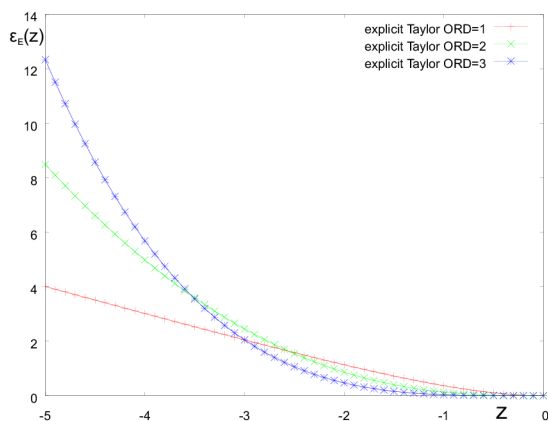
Obrázek 5.8: Oblast stability impl. Taylorovy řady $ORD = 14$ v blízkosti osy $Im(z)$

Absolutní chybu výpočtu v prvním kroku vyjádříme jako rozdíl mezi numerickým a analytickým řešením

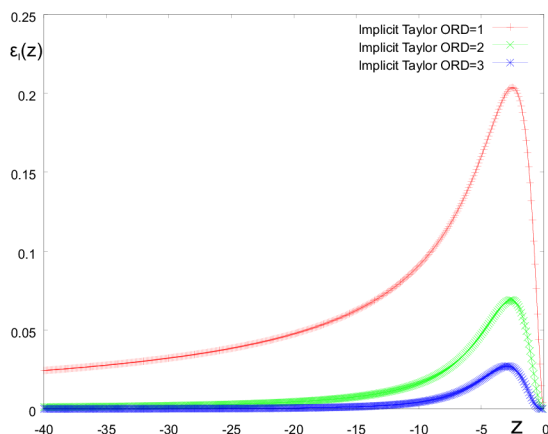
$$\varepsilon_E(z) = |R_E(z) - e^z|, \quad (5.18)$$

$$\varepsilon_I(z) = |R_I(z) - e^z|. \quad (5.19)$$

V Obr. 5.9, 5.10, 5.11, 5.12 je zobrazen vliv počáteční absolutní chyby výpočtu explicitní (5.18) a implicitní (5.19) Taylorovy řady v prvním kroku. Všimněme si, že s rostoucí hodnotou $|z|$ se u implicitní Taylorovy řady absolutní chyba výpočtu v prvním kroku zmenšuje, u explicitní Taylorovy řady je tomu naopak. Připomeňme, že při rostoucím počtu členů explicitní Taylorovy řady se zvětšuje oblast absolutní stability, absolutní chyba výpočtu je tedy malá i pro větší hodnoty $|z|$ viz Obr. 5.11.

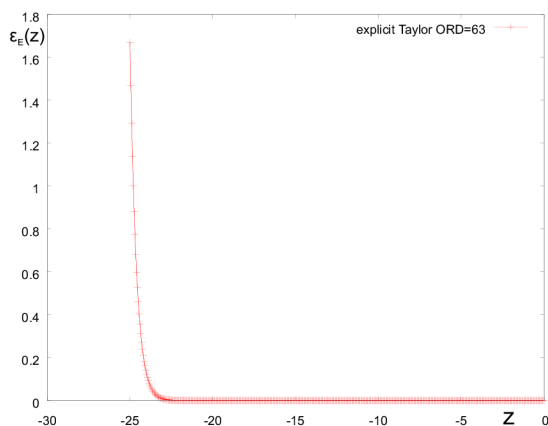


Obrázek 5.9: Absolutní chyba výpočtu, explicitní Taylorova řada $ORD = 1, 2, 3$

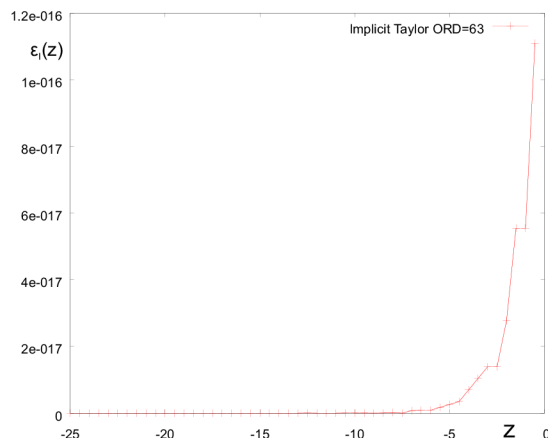


Obrázek 5.10: Absolutní chyba výpočtu, implicitní Taylorova řada $ORD = 1, 2, 3$

Víme, že implicitní Taylorova řada řádu 1,2 je L-stabilní. I přes počáteční chybu výpočtu (nejhorší případ je pro malé hodnoty $z \in (-5, 0)$) konverguje ke správnému



Obrázek 5.11: Absolutní chyba výpočtu, explicitní Taylorova řada $ORD = 63$



Obrázek 5.12: Absolutní chyba výpočtu, implicitní Taylorova řada $ORD = 63$

řešení. V Tab. 5.1 je zobrazena maximální možná absolutní chyba výpočtu pro případ kdy $z = -2.5$.

Tabulka 5.1: Implicitní Taylorova řada - absolutní chyba výpočtu, $h = 0.1$, $z = -2.5$

	Error(y)		
t	$ORD = 1$	$ORD = 2$	$ORD = 3$
0.1	0.203629	0.0688584	0.0262671
0.2	0.0748947	0.016046	0.00500224
0.3	0.0227705	0.002886	0.00071899
0.4	0.00661849	0.000473707	9.24321×10^{-5}
0.5	0.00190024	7.4629×10^{-5}	1.12077×10^{-5}
0.6	0.000543685	1.15214×10^{-5}	1.31227×10^{-6}
0.7	0.000155401	1.76014×10^{-6}	1.50223×10^{-7}
0.8	4.44055×10^{-5}	2.6741×10^{-7}	1.69365×10^{-8}

Implicitní Taylorova řada, jak lze vidět v Tab. 5.1, není vhodná pro malé hodnoty $|z|$ (avšak i pro ně konverguje ke správnému řešení). Její přesnost a tedy vhodnost použití se zvyšuje s rostoucí hodnotou $|z|$. Přesnost výpočtu se tedy projeví u velmi tuhých systémů kde $z \ll 0$ viz Tab. 5.2.

V Obr. 5.13, 5.14 je detailně znázorněna vlastnost L-stability implicitní Taylorovy řady (5.13), kdy je splněna podmínka (5.13), s rostoucí hodnotou $|z|$ pak absolutní chyba výpočtu v prvním kroku konverguje k nule.

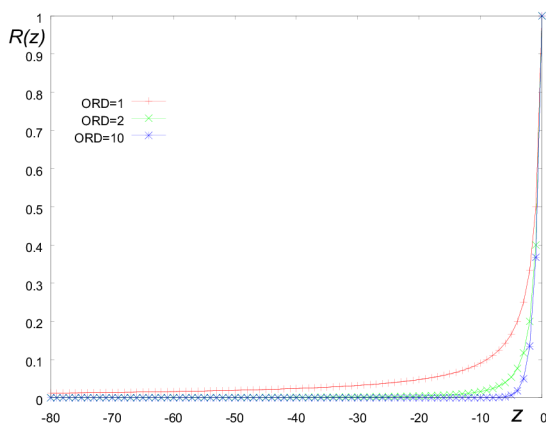
Ukázka řešení

Řešme diferenciální rovnici ve tvaru

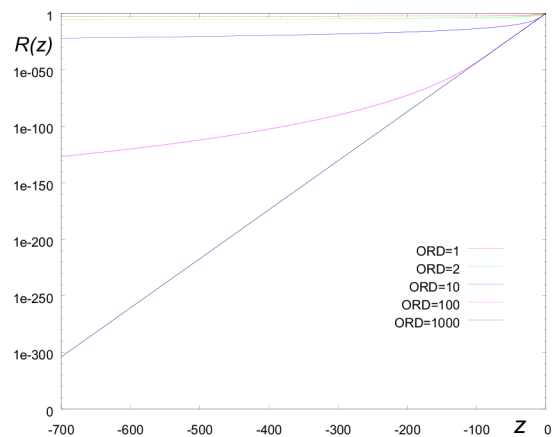
$$y' = -100 \cdot y, \quad y(0) = 1. \quad (5.20)$$

Tabulka 5.2: Implicitní Taylorova řada - absolutní chyba výpočtu v prvním kroku $\varepsilon_I(z)$

z	$\varepsilon_I(z)$		
	$ORD = 1$	$ORD = 3$	$ORD = 5$
-1	0.132121	0.00712056	0.000218718
-10	0.0908637	0.00434699	0.000631343
-50	0.0196078	4.51787×10^{-5}	3.46414×10^{-7}
-100	0.00990099	5.82182×10^{-6}	1.14062×10^{-8}
-500	0.00199601	$4.77126e \times 10^{-8}$	3.80168×10^{-12}
-1000	0.000999001	5.98202×10^{-9}	1.19401×10^{-13}



Obrázek 5.13: $R_I(z)$ (5.17)



Obrázek 5.14: $R_I(z)$ (5.17) - detail

Výpočet provedeme nejprve pomocí běžných explicitních numerických metod. Musíme použít velmi malý integrační krok pro zachování stability a přesnosti řešení. Maximální velikost integračního kroku při zachování stability dané numerické metody nalezneme v Tab. 5.3.

Z Tab. 5.3 lze usoudit, že explicitní numerické metody mají s výpočtem diferenciální rovnice (5.20) problém (nutnost použití značně malého integračního kroku). Použijeme tedy implicitní numerické metody. V Tab. 5.4 je zobrazena závislost integračního kroku h na absolutní chybě výpočtu v prvním kroku u jednotlivých implicitních numerických metod.

Pro řešení (5.20) je tedy vhodné použít implicitní numerickou metodu. Pro náš případ (kdy $\lambda = -100$) se jeví jako nejvhodnější použít implicitní Taylorovu řadu libovolného řádu, která pro $\lambda \in \mathcal{R}$ splňuje podmínku A-stabilní a L-stabilní numerické metody.

V Obr. 5.15 je zobrazeno řešení diferenciální rovnice (5.20) pomocí implicitních numerických metod.

Kombinace implicitní Taylorovy metody s Newtonovou metodou

Problémem u implicitních numerických metod je explicitní vyjádření neznámé hodnoty, která se vyskytuje i na pravé straně rovnice. Je nutno použít iteračního výpočtu viz podkapitola 2.1.4.

Tabulka 5.3: Maximální velikost h , expl. numerické metody, chyba výpočtu $EPS = 10^{-20}$

metoda	h
Eulerova metoda	4.22×10^{-11}
metoda Runge-Kutta 2. řádu	4.77×10^{-8}
metoda Runge-Kutta 4. řádu	1.52×10^{-5}
Taylorova metoda (63 členů)	0.1

Tabulka 5.4: Implicitní metody - absolutní chyba výpočtu v prvním kroku v závislosti na h

	Error(y)		
h	Lichoběžníková metoda	Eulerova metoda	Taylorova metoda (10 členů)
0.1	0.666712	9.08637×10^{-2}	3.24677×10^{-5}
1	0.960784	9.90099×10^{-3}	3.26986×10^{-14}
10	0.996008	9.99001×10^{-4}	3.59255×10^{-24}
50	0.9992	1.9996×10^{-4}	3.70846×10^{-31}
100	0.9996	9.999×10^{-5}	3.62517×10^{-34}

Řešme diferenciální rovnici (5.20) implicitní Taylorovou řadou, využívající Newtonovy metody k výpočtu y_{i+1} .

Ukázka výpočtu - implicitní Taylorova řada ($ORD = 2$):

$$\begin{aligned} y_{i+1} &= y_i + h(\lambda y_{i+1}) - \frac{h^2}{2!}(\lambda^2 y_{i+1}), \\ 0 &= y_i + (-1 + h\lambda - \frac{h^2\lambda^2}{2!})y_{i+1}. \end{aligned}$$

Použijeme Newtonovu metodu (2.22) k výpočtu y_{i+1}

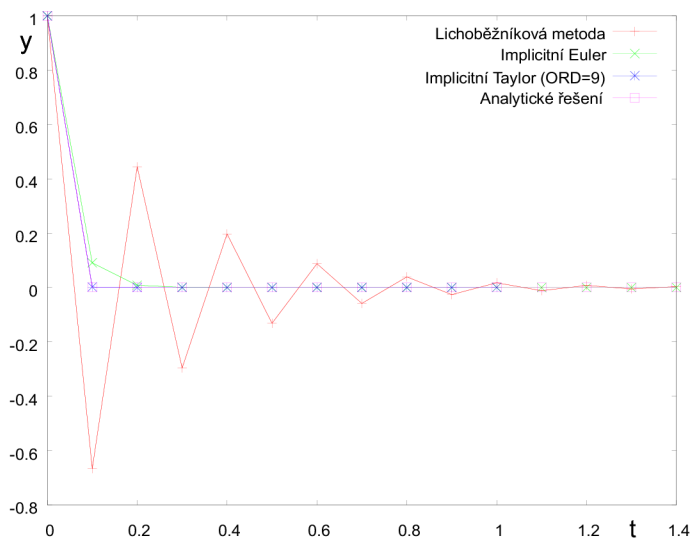
$$y_{i+1,j+1} = y_{i+1,j} - \frac{y_i + (-1 + h\lambda - \frac{h^2\lambda^2}{2!})y_{i+1,j}}{\frac{y_i + (-1 + h\lambda - \frac{h^2\lambda^2}{2!})(y_{i+1,j} + h_N) - (y_i + (-1 + h\lambda - \frac{h^2\lambda^2}{2!})y_{i+1,j})}{h_N}}. \quad (5.21)$$

U Dahlquistova problému navíc víme, že musí platit podmínka

$$|y_{i+1}| < |y_i|,$$

vhodné je tedy volit startovací hodnotu iterace $y_{i+1,0} < y_{i,j}$. Newtonovu iteraci ukončíme při dosažení zadané přesnosti výpočtu $|TOL| = |y_{i+1,j} - y_{i+1,j+1}|$.

V Tab. 5.5 nalezneme porovnání absolutní chyby výpočtu rovnice (5.20) pomocí implicitní Taylorovy řady a kombinace implicitní Taylorovy řady s Newtonovou metodou (5.21). Všimněme si stejné vypočtené hodnoty, tedy stejné absolutní chyby u obou provedených výpočtů. K výpočtu prvního kroku jsou potřeba vždy 2 iterace v Newtonově metodě. Zkratka ORD značí jako obvykle řád Taylorovy řady.



Obrázek 5.15: Řešení problému (5.20) - implicitní numerické metody, $h = 0.1$

Explicitní Taylorova metoda - rekurentní výpočet

Zaměřme se nyní na členy explicitní Taylorovy řady. Taylorovu řadu ve tvaru

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \cdots + \frac{h^n}{n!}y_i^{(n)}, \quad (5.22)$$

přepíšeme do tvaru

$$y_{i+1} = y_i + DY1_i + DY2_i + \cdots + DYn_i, \quad (5.23)$$

kde jednotlivé členy vyjádříme rekurentně ve tvaru

$$\begin{aligned} DY1_i &= hy'_i &= h\lambda y_i &, \\ DY2_i &= \frac{h^2}{2!}y''_i &= \frac{h}{2}\lambda DY1_i &= \frac{h^2}{2}\lambda^2 y_i, \\ &\vdots &\vdots &\vdots \\ DYn_i &= \frac{h^n}{n!}y_i^{(n)} &= \frac{h}{n}\lambda DY(n-1)_i &= \frac{h^n}{n!}\lambda^n y_i. \end{aligned}$$

V Obr. 5.16, 5.17, 5.18 nalezneme závislost počtu členů Taylorovy řady DY potřebných k výpočtu prvního kroku rovnice (5.3) pro dosažení dané přesnosti $EPS = 10^{-15}$, což je v TKSL/C výchozí hodnota (lokální chyba zanedbání vyšších členů Taylorovy řady “local truncation error”).

Při rostoucím λ se zvyšuje nárok na počet členů Taylorovy řady. Při zachování integračního kroku $h = 1$ potřebujeme pro výpočet prvního kroku u $\lambda = -100$ v rovnici (5.3) minimálně 312 členů DY pro dosažení dané přesnosti $EPS = 10^{-15}$, viz Obr. 5.18. Z důvodu zachování přesnosti výpočtu (v Taylorově rozvoji počítáme se členy s velkým rozptylem od 10^{-17} až po 10^{43}) je potřeba pro přesný výpočet použít víceslovní aritmetiku. Veškeré výpočty s víceslovní aritmetikou jsou prováděny pomocí knihovny GMP [GMP10].

Tabulka 5.5: Impl. Taylorova řada a Newtonova metoda - absolutní chyba výpočtu v prvním kroku $h = 1$, $\lambda = -100$, $h_N = 0.01$, $TOL = 10^{-15}$

	Error(y)	
ORD	impl. Taylorova řada	impl. Taylorova řada a Newt. iterace
1	9.90099×10^{-3}	9.90099×10^{-3}
2	1.9604×10^{-4}	1.9604×10^{-4}
3	5.82182×10^{-6}	5.82182×10^{-6}
4	2.30498×10^{-7}	2.30498×10^{-7}
5	1.14062×10^{-8}	1.14062×10^{-8}
6	6.7725×10^{-10}	6.7725×10^{-10}
7	4.69091×10^{-11}	4.69091×10^{-11}
8	3.71287×10^{-12}	3.71287×10^{-12}
9	3.30571×10^{-13}	3.30571×10^{-13}
10	3.26986×10^{-14}	3.26986×10^{-14}

Kombinace implicitní Taylorovy metody s rekurentním výpočtem a Newtonovou metodou (ITMRN)

Zaměříme se nyní na členy implicitní Taylorovy řady. Taylorovu řadu ve tvaru

$$y_{i+1} = y_i + hy'_{i+1} - \dots - \frac{(-h)^n}{n!} y_{i+1}^{(n)}, \quad (5.24)$$

přepíšeme do tvaru

$$y_{i+1} = y_i - DY1_{i+1} - DY2_{i+1} - \dots - DYn_{i+1}, \quad (5.25)$$

kde jednotlivé členy vyjádříme rekurentně ve tvaru

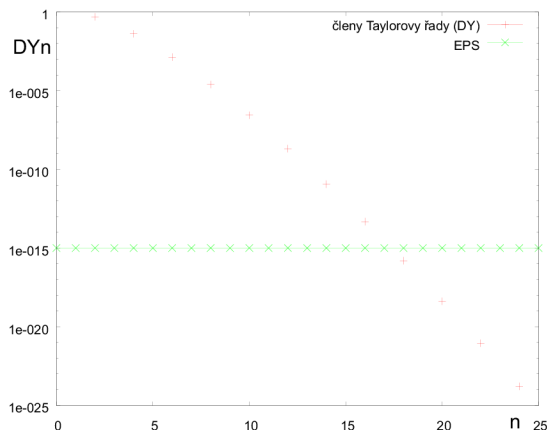
$$\begin{aligned} DY1_{i+1} &= -hy'_{i+1} = -h\lambda y_{i+1}, \\ DY2_{i+1} &= \frac{(-h)^2}{2!} y''_{i+1} = \frac{-h}{2} \lambda DY1_{i+1}, \\ &\vdots \\ DYn_{i+1} &= \frac{(-h)^n}{n!} y_{i+1}^{(n)} = \frac{-h}{n} \lambda DY(n-1)_{i+1}. \end{aligned}$$

Numerický výpočet pomocí Newtonovy metody se provádí podle vzorce

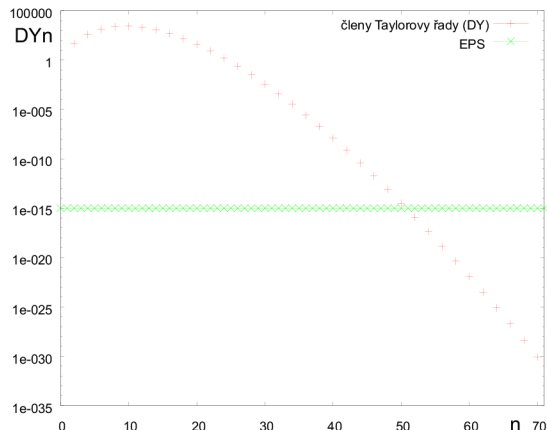
$$y_{i+1,j+1} = y_{i+1,j} - \frac{f(y_{i+1,j})}{f'(y_{i+1,j})}, \quad (5.26)$$

kde

$$\begin{aligned} f(y_{i+1,j}) &= -y_{i+1,j} + y_i - DY1_{i+1,j} - DY2_{i+1,j} - \dots - DYn_{i+1,j}, \\ f'(y_{i+1,j}) &= \frac{f(y_{i+1,j+h_N}) - f(y_{i+1,j})}{h_N} = \\ &= \frac{-(y_{i+1,j+h_N}) + y_i - DY1_{N,i+1,j} - DY2_{N,i+1,j} - \dots - DYn_{N,i+1,j} - f(y_{i+1,j})}{h_N}. \end{aligned}$$



Obrázek 5.16: Členy explicitní Taylorovy řady DY ($\lambda = -1, h = 1$)



Obrázek 5.17: Členy explicitní Taylorovy řady DY ($\lambda = -10, h = 1$)

Nové členy $DY1_{N,i+1,j}, DY2_{N,i+1,j}, \dots, DYn_{N,i+1,j}$ vyjádříme ve tvaru

$$\begin{aligned} DY1_{N,i+1,j} &= -h\lambda(y_{i+1,j} + h_N), \\ DY2_{N,i+1,j} &= \frac{(-h)^2}{2!}\lambda^2(y_{i+1,j} + h_N) = \frac{-h}{2}\lambda DY1_{N,i+1,j}, \\ &\vdots \\ DYn_{N,i+1,j} &= \frac{(-h)^n}{n!}\lambda^n(y_{i+1,j} + h_N) = \frac{-h}{n}\lambda DY(n-1)_{N,i+1,j}. \end{aligned}$$

Jako první člen iterace $y_{i+1,0}$ volíme

$$y_{i+1,0} = y_i,$$

iteraci ukončíme při splnění podmínce

$$|y_{i+1,j} - y_{i+1,j+1}| < TOL,$$

kde TOL je daná chyba výpočtu.

Výsledky výpočtu pomocí ITMRN jsou samozřejmě shodné s metodou “Kombinace implicitní Taylorovy metody s Newtonovou metodou” viz Tab. 5.5.

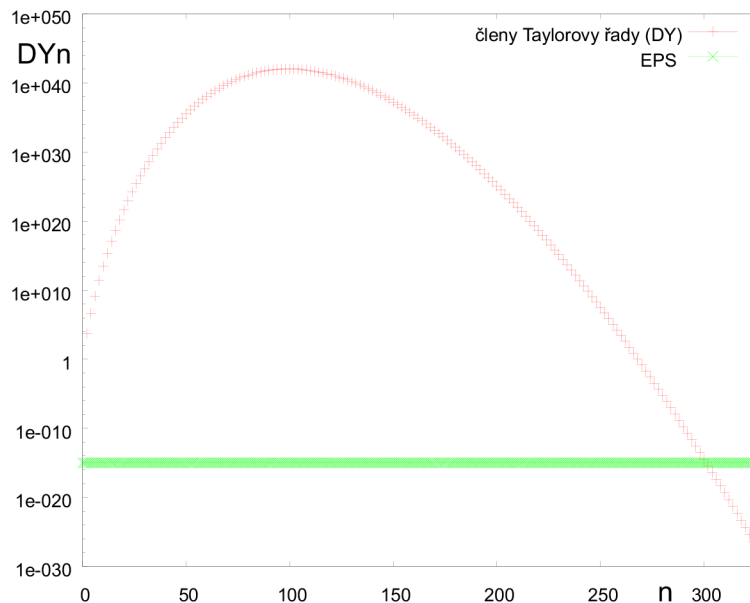
Závěr

Na předchozím příkladu jsme podrobně prozkoumali stabilitu a konvergenci explicitní a implicitní Taylorovy řady. Explicitní Taylorův polynom

$$y_{i+1} = y_i + hy'_i + \frac{h^2}{2!}y''_i + \dots + \frac{h^n}{n!}y_i^{(n)},$$

splňuje podmínku A-stabilní numerické metody pouze při $n \rightarrow \infty$. Při výpočtech prováděných pomocí explicitní Taylorovy řady (především při použití většího počtu členů) jsme často nuceni pro zajištění dostatečné přesnosti výpočtu členů použít víceslovní aritmetiku.

Implicitní Taylorova řada splňuje do druhého řádu A-stabilní a L-stabilní numerické metody (viz Obr. 3.5, Obr. 5.3). Při narůstajícím řádu implicitní Taylorovy metody se v levé polovině komplexní roviny začínou objevovat menší oblasti nestability v blízkosti



Obrázek 5.18: Členy explicitní Taylorovy řady DY ($\lambda = -100, h = 1$)

imaginární osy (viz Obr. 5.5, Obr. 5.6), které se však s rostoucím řádem zmenšují a vzdalují podél imaginární osy k nekonečnu.

Implicitní Taylorova řada je tedy vhodná pro diferenciální rovnici (5.3), kde platí $\lambda \in \mathcal{R}$ a zároveň $\lambda \ll 0$, neboť splňuje podmínky A-stabilní a L-stabilní numerické metody.

Jako zajímavý způsob implementace implicitní Taylorovy řady se jeví kombinace implicitní Taylorovy metody s rekurentním výpočtem členů Taylorovy řady a Newtonovou metodou (ITMRN).

5.6 Příklady výpočtu

Zaměřme se nyní na konkrétní problémy a jejich možnosti řešení pomocí explicitní a nově také implicitní Taylorovy řady.

5.6.1 Semi-analytický výpočet

Standardně se v literatuře uvádí, že explicitní Taylorova řada není vhodná pro řešení tuhých systémů [JZ05, Hal83, Gib60]. Ukážeme si, že díky semi-analytické podstatě výpočtu Taylorovy řady (rekurentní výpočet členů Taylorovy řady s využitím tvořících diferenciálních rovnic) má v některých případech tento proces výpočtu vliv na odstranění tuhosti systému.

Prof. Butcher ve své zvané přednášce na konferenci ICNAAM 2008 [But08a] prezentoval zajímavou diferenciální rovnici (5.27), na které demonstroval “tuhý systém”.

$$y' = L(y - \sin(t)) + \cos(t), \quad y(0) = 0, \quad L \ll 0. \quad (5.27)$$

Analytické řešení diferenciální rovnice (5.27) je ve tvaru

$$y = \sin(t). \quad (5.28)$$

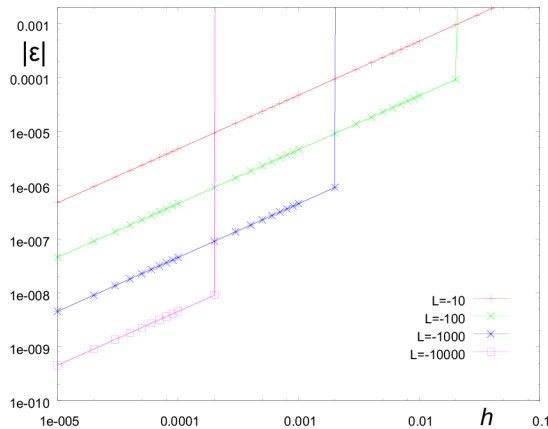
Ukažme si, jak tuto diferenciální rovnici (5.27) řeší běžné numerické metody.

Explicitní Eulerova metoda

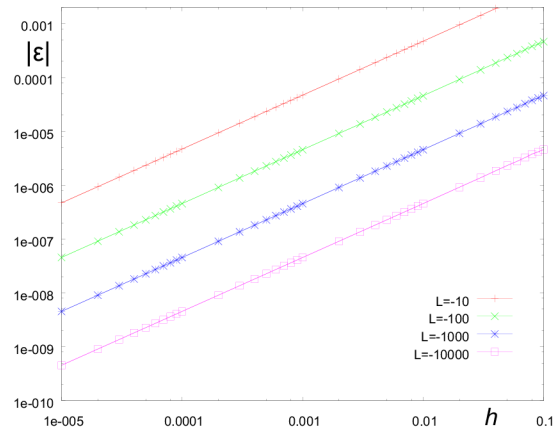
Diferenciální rovnici (5.27) řešme nejprve pomocí explicitní Eulerovy metody

$$y_{i+1} = y_i + h(L(y_i - \sin(t)) + \cos(t)). \quad (5.29)$$

V Obr. 5.19 je zobrazena závislost absolutní chyby výpočtu ($|\epsilon| = |y_i - y(i \cdot h)|$) v čase $TMAX = i \cdot h = 2$ na velikosti konstanty L a délce integračního kroku. S rostoucí absolutní hodnotou konstanty L v problému (5.27) musíme zmenšovat integrační krok (z důvodu stability výpočtu daného numerickou metodou). Explicitní numerická metoda je tedy pro řešení problému (5.27) nevhodná - jedná se o “tuhý systém” (nebo spíše nevhodný matematický popis systému).



Obrázek 5.19: Explicitní Eulerova metoda - absolutní chyba výpočtu v $TMAX = 2$



Obrázek 5.20: Implicitní Eulerova metoda - absolutní chyba výpočtu v $TMAX = 2$

V Tab. 5.6 je zobrazena absolutní chyba výpočtu explicitní Eulerovy metody v závislosti na různé velikosti konstanty L .

Implicitní Eulerova metoda

Pokusme se nyní řešit problém (5.27) pomocí implicitní Eulerovy metody

$$y_{i+1} = y_i + h(L(y_{i+1} - \sin(t+h)) + \cos(t+h)), \quad (5.30)$$

$$y_{i+1} = \frac{y_i + h(-L \sin(t+h) + \cos(t+h))}{1 - Lh}. \quad (5.31)$$

V Obr. 5.20 je zobrazena závislost chyby výpočtu v čase $TMAX = 2$ na velikosti konstanty L a délce integračního kroku. Všimněme si, že i při větší absolutní hodnotě konstanty L můžeme použít větší integrační krok (oproti Obr. 5.19) a přitom je zachována stabilita implicitní Eulerovy metody.

Při řešení “tuhého” problému (5.27) je tedy implicitní numerická metoda díky větší oblasti stability vhodnější oproti explicitní numerické metodě. Analyzujme však možnost řešení problému (5.27) pomocí explicitní Taylorovy metody využívající rekurentní výpočet členů a pomocné tvořící diferenciální rovnice.

Tabulka 5.6: Absolutní chyba výpočtu - expl. Eulerova metoda, $h = 0.1$

t	Error(y)			
	$L = -10$	$L = -100$	$L = -1000$	$L = -10000$
0.1	0.000166583	0.000166583	0.000166583	0.000166583
0.2	0.000664502	0.000834748	0.0158272	0.165752
0.3	0.00115578	0.00866851	1.56805	165.588
0.4	0.00163551	0.0763811	155.236	165422
0.5	0.0020989	0.689529	15368.3	1.65257×10^8
0.6	0.00254132	6.20322	1.52146×10^6	1.65091×10^{11}
0.7	0.00295835	55.8319	1.50625×10^8	1.64926×10^{14}
0.8	0.00334582	502.484	1.49119×10^{10}	1.64761×10^{17}
0.9	0.00369985	4522.36	1.47628×10^{12}	1.64597×10^{20}
1.0	0.00401692	40701.2	1.46151×10^{14}	1.64432×10^{23}
⋮	⋮	⋮	⋮	⋮
2.0	0.0046737	1.41916×10^{14}	1.32177×10^{34}	1.62795×10^{53}

Explicitní Taylorova metoda využívající tvořící diferenciální rovnice a rekurentní výpočet členů

Původní diferenciální rovnici ve tvaru

$$y' = \lambda(y - \sin(t)) + \cos(t), \quad y(0) = 0,$$

kde $\lambda = L$, přepíšeme do tvaru

$$y' = \lambda y - \lambda z + x, \quad y(0) = 0, \quad (5.32)$$

kde $z = \sin(t)$ a $x = \cos(t)$. K výpočtu funkcí $\sin(t)$ a $\cos(t)$ využijeme tvořící diferenciální rovnice ve tvaru

$$\begin{aligned} x' &= -z, & x(0) &= 1, \\ z' &= x, & z(0) &= 0. \end{aligned} \quad (5.33)$$

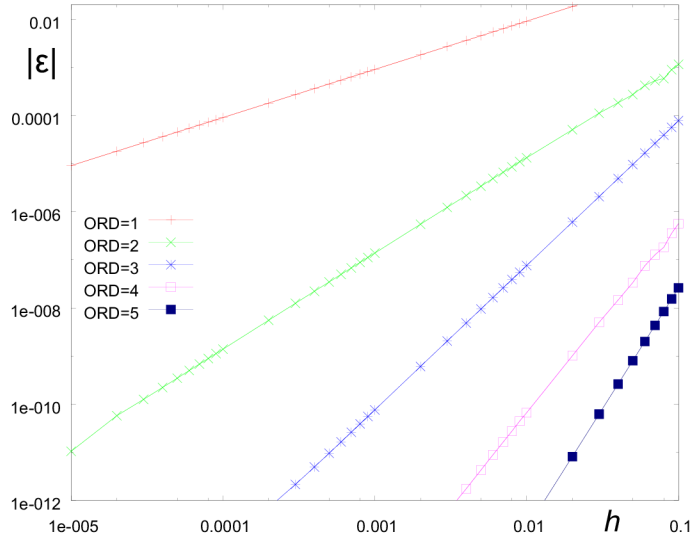
Předcházející diferenciální rovnice (5.32), (5.33) vyjádříme explicitní Taylorovou řadou ve tvaru

$$\begin{aligned} y_{i+1} &= y_i + DY1_i + DY2_i + \dots + DYn_i, \\ x_{i+1} &= x_i + DX1_i + DX2_i + \dots + DXn_i, \\ z_{i+1} &= z_i + DZ1_i + DZ2_i + \dots + DZn_i, \end{aligned}$$

kde rekurentní výpočet členů Taylorovy řady lze vyjádřit ve tvaru

$$\begin{aligned} DY1_i &= h(\lambda y_i - \lambda z_i + x_i), \\ DY2_i &= \frac{h}{2}(\lambda DY1_i - \lambda DZ1_i + DX1_i), \\ &\vdots \\ DYn_i &= \frac{h}{n}(\lambda DY(n-1)_i - \lambda DZ(n-1)_i + DX(n-1)_i), \end{aligned}$$

$$\begin{aligned}
DX1_i &= h(-z_i), \\
DX2_i &= \frac{h}{2}(-DZ1_i), \\
&\vdots \\
DXn_i &= \frac{h}{n}(-DZ(n-1)_i), \\
\\
DZ1_i &= h(x_i), \\
DZ2_i &= \frac{h}{2}DX1_i, \\
&\vdots \\
DZn_i &= \frac{h}{n}DX(n-1)_i.
\end{aligned}$$



Obrázek 5.21: Explicitní Taylorova metoda, absolutní chyba výpočtu v $TMAX = 2$

Při podrobnějším rozboru členů Taylorovy řady zjistíme, že platí $DYj_i = DZj_i$ pro $j = 1, 2, \dots, n$, pokud platí počáteční podmínka $y(0) = 0$. Pro první krok jsou tedy členy DYj_i ve tvaru

$$\begin{aligned}
DY1_0 &= h(\lambda y(0) - \lambda z(0) + x(0)) = hx(0) = DZ1_0, \\
DY2_0 &= \frac{h}{2}(\lambda DY1_0 - \lambda DZ1_0 + DX1_0) = \frac{h}{2}(\lambda hx(0) - \lambda hx(0) + DX1_0) = \frac{h}{2}DX1_0 = DZ2_0, \\
DY3_0 &= \frac{h}{3}(\lambda DY2_0 - \lambda DZ2_0 + DX2_0) = \frac{h}{3}(DX2_0) = DZ3_0, \\
&\vdots \\
DYn_0 &= \frac{h}{n}(\lambda DY(n-1)_0 - \lambda DZ(n-1)_0 + DX(n-1)_0) = \frac{h}{n}(DX(n-1)_0) = DZn_0.
\end{aligned}$$

Jedná se o semi-analytický výpočet vyplývající z Taylorovy řady využívající tvořící diferenciální rovnice.

Původní diferenciální rovnici ve tvaru

$$y' = \lambda(y - \sin(t)) + \cos(t), \quad y(0) = 0,$$

lze převést na ekvivalentní diferenciální rovnici (odstranění $\lambda(y - \sin(t))$)

$$y'_{EKV} = \cos(t), \quad y(0) = 0. \tag{5.34}$$

Tento převod provede explicitní Taylorova metoda (využívající tvořící diferenciální rovnice a rekurentní výpočet členů) automaticky. Diferenciální rovnice (5.34) má shodné analytické řešení s rovnicí (5.27) $y = y_{EKV} = \sin(t)$. U diferenciální rovnice (5.34) se však již nejedná o “tuhý systém” (konstanta $\lambda = L$ nemá vliv na průběh numerického výpočtu).

Explicitní Taylorova metoda (využívající tvořící diferenciální rovnice a rekurentní výpočet členů) počítající s $ORD = 1$ (tedy explicitní Eulerova metoda) řeší diferenciální rovnici (5.27) pro různou velikost konstanty L se stejnou absolutní chybou výpočtu $|\text{Error}(y(i \cdot h))| = |y_i - y(i \cdot h)|$. V Tab. 5.7 nalezneme absolutní chybu výpočtu v závislosti na řádu metody explicitní Taylorovy metody (konstanta L nemá vliv na absolutní chybu výpočtu).

Tabulka 5.7: Absolutní chyba výpočtu - explicitní Taylorova metoda, $h = 0.1$, $L = -10000$

t	Error(y)				
	$ORD = 1$	$ORD = 2$	$ORD = 3$	$ORD = 4$	$ORD = 5$
0.1	0.000166583	0.000166583	8.33135×10^{-8}	8.33135×10^{-8}	1.98385×10^{-11}
0.2	0.00133067	0.000330669	9.97462×10^{-7}	1.65517×10^{-7}	3.16744×10^{-10}
0.3	0.00347979	0.000487293	2.72749×10^{-6}	2.44131×10^{-7}	8.8597×10^{-10}
0.4	0.00658166	0.000631608	5.24204×10^{-6}	3.16728×10^{-7}	1.7173×10^{-9}
0.5	0.0105845	0.000758962	8.4938×10^{-6}	3.8098×10^{-7}	2.79517×10^{-9}
0.6	0.0154175	0.000864981	1.24202×10^{-5}	4.34692×10^{-7}	4.09896×10^{-9}
0.7	0.0209922	0.000945642	1.69444×10^{-5}	4.75843×10^{-7}	5.60321×10^{-9}
0.8	0.0272031	0.000997342	2.19762×10^{-5}	5.02617×10^{-7}	7.27808×10^{-9}
0.9	0.0339295	0.00101696	2.74139×10^{-5}	5.1344×10^{-7}	9.08975×10^{-9}
1.0	0.041037	0.00100193	3.3145×10^{-5}	5.07008×10^{-7}	1.10009×10^{-8}
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots
2.0	0.0981569	0.001161	7.82902×10^{-5}	5.64968×10^{-7}	2.61531×10^{-8}

V Obr. 5.21 je zobrazena absolutní chyba výpočtu v čase $TMAX = 2$ v závislosti na řádu metody explicitní Taylorovy řady ORD a délce integračního kroku h .

Závěr

Explicitní Taylorova metoda využívající rekurentní výpočet členů a tvořící diferenciální rovnice je díky své semi-analytické podstatě výpočtu v některých případech vhodná pro řešení tuhých systémů - automaticky odstraňuje tuhost ze systému (5.27) tím, že řeší ekvivalentní systém bez tuhosti (5.34).

5.6.2 Speciální testovací soustava diferenciálních rovnic

Mějme soustavu diferenciálních rovnic ve tvaru

$$\begin{aligned} x' &= -z, & x(0) &= 1, \\ z' &= x, & z(0) &= 0, \end{aligned} \tag{5.35}$$

jejíž známe analytické řešení je ve tvaru

$$\begin{aligned}x &= \cos(t), \\z &= \sin(t).\end{aligned}\tag{5.36}$$

Zaměříme se nejprve na numerické řešení SDR (5.35) pomocí explicitní Taylorovy řady.

Explicitní Taylorova metoda

Vyjádríme předcházející diferenciální rovnice (5.35) pomocí explicitní Taylorovy řady ve tvaru

$$\begin{aligned}x_{i+1} &= x_i + DX1_i + DX2_i + \cdots + DXn_i, \\z_{i+1} &= z_i + DZ1_i + DZ2_i + \cdots + DZn_i,\end{aligned}$$

kde jednotlivé členy vyjádříme rekurentně ve tvaru

$$\begin{aligned}DX1_i &= h(-z_i), \\DX2_i &= \frac{h}{2}(-DZ1_i), \\&\vdots \\DXn_i &= \frac{h}{n}(-DZ(n-1)_i), \\ \\DZ1_i &= h(x_i), \\DZ2_i &= \frac{h}{2}DX1_i, \\&\vdots \\DZn_i &= \frac{h}{n}DX(n-1)_i.\end{aligned}$$

Tabulka 5.8: Absolutní chyba výpočtu, explicitní Taylorova metoda, $h = 0.1$, $ORD = 8$

t	$ \text{Error}(x) $	$ \text{Error}(z) $
0.1	3.82567×10^{-16}	2.7478×10^{-15}
0.2	7.77156×10^{-16}	5.46785×10^{-15}
0.3	1.88738×10^{-15}	8.04912×10^{-15}
0.4	3.44169×10^{-15}	1.04916×10^{-14}
0.5	5.66214×10^{-15}	1.2601×10^{-14}
0.6	8.10463×10^{-15}	1.44329×10^{-14}
0.7	1.11022×10^{-14}	1.58762×10^{-14}
0.8	1.45439×10^{-14}	1.68754×10^{-14}
0.9	1.82077×10^{-14}	1.70974×10^{-14}
1	2.20934×10^{-14}	1.68754×10^{-14}

Tabulka 5.9: Absolutní chyba výpočtu, explicitní Taylorova metoda, $h = 0.5$, $ORD = 8$

t	$ \text{Error}(x) $	$ \text{Error}(z) $
0.5	2.68605×10^{-10}	5.37008×10^{-9}
1	5.62055×10^{-9}	9.16782×10^{-9}
1.5	1.39917×10^{-8}	8.02632×10^{-9}
2	2.15025×10^{-8}	4.47726×10^{-10}

V Tab. 5.8 resp. Tab. 5.9 je zobrazena absolutní chyba výpočtu při použití explicitní Taylorovy metody řádu 8 a délce integračního kroku $h = 0.1$ resp. $h = 0.5$.

Řešme nyní problém (5.35) implicitní Taylorovou metodou.

Implicitní Taylorova metoda

Vyjádříme předcházející SDR (5.35) pomocí implicitní Taylorovy řady ve tvaru

$$\begin{aligned}x_{i+1} &= x_i - h z_{i+1} + \frac{h^2}{2!} x_{i+1} + \frac{h^3}{3!} z_{i+1} - \frac{h^4}{4!} x_{i+1} - \dots, \\z_{i+1} &= z_i + h x_{i+1} + \frac{h^2}{2!} z_{i+1} - \frac{h^3}{3!} x_{i+1} - \frac{h^4}{4!} z_{i+1} + \dots,\end{aligned}$$

dále vyjádříme explicitně x_{i+1}, z_{i+1}

$$\begin{aligned}z_{i+1} &= z_i + x_{i+1} \cdot \left(h - \frac{h^3}{3!} + \frac{h^5}{5!} - \frac{h^7}{7!} + \dots \right) + z_{i+1} \cdot \left(\frac{h^2}{2!} - \frac{h^4}{4!} + \frac{h^6}{6!} - \frac{h^8}{8!} + \dots \right), \\z_{i+1} &= \frac{z_i + x_{i+1} \cdot \left(h - \frac{h^3}{3!} + \frac{h^5}{5!} - \frac{h^7}{7!} + \dots \right)}{1 - \frac{h^2}{2!} + \frac{h^4}{4!} - \frac{h^6}{6!} + \frac{h^8}{8!} - \dots},\end{aligned}$$

$$\begin{aligned}x_{i+1} &= x_i + z_{i+1} \cdot \left(-h + \frac{h^3}{3!} - \frac{h^5}{5!} + \frac{h^7}{7!} - \dots \right) + x_{i+1} \cdot \left(\frac{h^2}{2!} - \frac{h^4}{4!} + \frac{h^6}{6!} - \frac{h^8}{8!} + \dots \right), \\x_{i+1} &= \frac{z_i + z_{i+1} \cdot \left(-h + \frac{h^3}{3!} - \frac{h^5}{5!} + \frac{h^7}{7!} - \dots \right)}{1 - \frac{h^2}{2!} + \frac{h^4}{4!} - \frac{h^6}{6!} + \frac{h^8}{8!} - \dots},\end{aligned}$$

dosadíme z_{i+1} do x_{i+1} a obdržíme výsledný vztah ve tvaru

$$\begin{aligned}x_{i+1} &= \frac{x_i \cdot \left(1 - \frac{h^2}{2!} + \frac{h^4}{4!} - \frac{h^6}{6!} + \frac{h^8}{8!} - \dots \right) + z_i \cdot \left(-h + \frac{h^3}{3!} - \frac{h^5}{5!} + \frac{h^7}{7!} - \dots \right)}{\left(1 - \frac{h^2}{2!} + \frac{h^4}{4!} - \frac{h^6}{6!} + \frac{h^8}{8!} - \dots \right)^2 + \left(-h + \frac{h^3}{3!} - \frac{h^5}{5!} + \frac{h^7}{7!} - \dots \right)^2}, \\z_{i+1} &= \frac{z_i - x_{i+1} \cdot \left(-h + \frac{h^3}{3!} - \frac{h^5}{5!} + \frac{h^7}{7!} - \dots \right)}{1 - \frac{h^2}{2!} + \frac{h^4}{4!} - \frac{h^6}{6!} + \frac{h^8}{8!} - \dots}.\end{aligned}\tag{5.37}$$

Tabulka 5.10: Absolutní chyba výpočtu, implicitní Taylorova metoda, $h = 0.1$, $ORD = 8$

t	$ \text{Error}(x) $	$ \text{Error}(z) $
0.1	4.44089×10^{-16}	2.70617×10^{-15}
0.2	1.55431×10^{-15}	5.27356×10^{-15}
0.3	3.21965×10^{-15}	7.66054×10^{-15}
0.4	5.21805×10^{-15}	9.76996×10^{-15}
0.5	7.66054×10^{-15}	1.14908×10^{-14}
0.6	1.05471×10^{-14}	1.27676×10^{-14}
0.7	1.36557×10^{-14}	1.35447×10^{-14}
0.8	1.69864×10^{-14}	1.38778×10^{-14}
0.9	2.05391×10^{-14}	1.36557×10^{-14}
1	2.42029×10^{-14}	1.27676×10^{-14}

Tabulka 5.11: Absolutní chyba výpočtu, implicitní Taylorova metoda, $h = 0.5$, $ORD = 8$

t	$ \text{Error}(x) $	$ \text{Error}(z) $
0.5	4.37363×10^{-9}	3.12749×10^{-9}
1	1.06752×10^{-8}	1.29559×10^{-9}
1.5	1.49843×10^{-8}	5.97149×10^{-9}
2	1.37161×10^{-8}	1.65658×10^{-8}

V Tab. 5.10 resp. Tab. 5.11 je zobrazena absolutní chyba výpočtu při použití implicitní Taylorovy řady řádu 8 a délce integračního kroku $h = 0.1$ resp. $h = 0.5$. V porovnání s explicitní Taylorovou metodou se jedná o řádově shodnou absolutní chybu výpočtu.

Zavedme nyní do výpočtu implicitní Taylorovy metody iteraci pomocí Newtonovy metody.

Kombinace implicitní Taylorovy metody s rekurentním výpočtem a Newtonovou metodou (ITMRN)

Vyjádríme předcházející diferenciální rovnice (5.35) pomocí implicitní Taylorovy řady ve tvaru

$$\begin{aligned}x_{i+1} &= x_i - DX1_{i+1} - DX2_{i+1} - \dots - DXn_{i+1}, \\z_{i+1} &= z_i - DZ1_{i+1} - DZ2_{i+1} - \dots - DZn_{i+1},\end{aligned}$$

kde rekurentní výpočet členů Taylorovy řady lze vyjádřit ve tvaru

$$\begin{aligned}DX1_{i+1} &= -h(-z_{i+1}), \\DX2_{i+1} &= -\frac{h}{2}(-DZ1_{i+1}), \\&\vdots \\DXn_{i+1} &= -\frac{h}{n}(-DZ(n-1)_{i+1}), \\DZ1_{i+1} &= -h(x_{i+1}), \\DZ2_{i+1} &= -\frac{h}{2}DX1_{i+1}, \\&\vdots \\DZn_{i+1} &= -\frac{h}{n}DX(n-1)_{i+1}.\end{aligned}$$

Hledáme tedy řešení soustavy dvou rovnic f_1, f_2 o dvou neznámých x_{i+1}, z_{i+1}

$$\begin{aligned}f_1(x_{i+1}, z_{i+1}) &= -x_{i+1} + x_i - DX1_{i+1} - DX2_{i+1} - \dots - DXn_{i+1} = 0, \\f_2(x_{i+1}, z_{i+1}) &= -z_{i+1} + z_i - DZ1_{i+1} - DZ2_{i+1} - \dots - DZn_{i+1} = 0.\end{aligned}\tag{5.38}$$

K numerickému výpočtu soustavy (5.38) využijeme Newtonovu metodu

$$\begin{pmatrix} f_{1x}(x_{i+1,j}, z_{i+1,j}) & f_{1z}(x_{i+1,j}, z_{i+1,j}) \\ f_{2x}(x_{i+1,j}, z_{i+1,j}) & f_{2z}(x_{i+1,j}, z_{i+1,j}) \end{pmatrix} \begin{pmatrix} b_{i+1,j} \\ a_{i+1,j} \end{pmatrix} = \begin{pmatrix} -f_1(x_{i+1,j}, z_{i+1,j}) \\ -f_2(x_{i+1,j}, z_{i+1,j}) \end{pmatrix},$$

odkud vyjádříme $x_{i+1,j+1}$ a $y_{i+1,j+1}$ ve tvaru

$$\begin{aligned}x_{i+1,j+1} &= x_{i+1,j} + b_{i+1,j}, \\z_{i+1,j+1} &= z_{i+1,j} + c_{i+1,j},\end{aligned}$$

k výpočtu $b_{i+1,j}, c_{i+1,j}$ použijeme Cramerovo pravidlo

$$\begin{aligned}b_{i+1,j} &= -\frac{D_x}{D} = -\frac{f_{1x}f_{2z} - f_{2x}f_{1z}}{f_{1x}f_{2z} - f_{2x}f_{1z}}, \\c_{i+1,j} &= -\frac{D_z}{D} = -\frac{f_{1x}f_{2z} - f_{1x}f_{2z}}{f_{1x}f_{2z} - f_{2x}f_{1z}},\end{aligned}$$

kde D_x, D_z a D jsou příslušné determinanty, $f_{1x}, f_{1z}, f_{2x}, f_{2z}$ jsou parciální derivace - členy Jacobiovy matice soustavy (5.38).

Parciální derivace můžeme vyjádřit analyticky, nebo v praxi častěji numericky pomocí diferenční formule např. $f_{1x} = (f_1(x + h_N, z) - f_1(x, z))/h_N$.

Analyticky vyjádřená Jacobiova matice

Při známém tvaru soustavy diferenciálních rovnic (což jsou pomocné tvořící diferenciální rovnice) je vhodné počítat s analyticky vyjádřenými parciálními derivacemi. V našem případě jsou parciální derivace ve tvaru

$$\begin{aligned}
 f_1(x_{i+1,j}, z_{i+1,j}) &= -x_{i+1,j} + x_i - DX1_{i+1,j} - DX2_{i+1,j} - \dots - DXn_{i+1,j}, \\
 f_1(x_{i+1,j}, z_{i+1,j}) &= x_i + z_{i+1} \cdot (-h + \frac{h^3}{3!} - \dots) + x_{i+1} \cdot (-1 + \frac{h^2}{2!} - \dots), \\
 f_{1x}(x_{i+1,j}, z_{i+1,j}) &= -1 + \frac{h^2}{2!} - \frac{h^4}{4!} + \frac{h^6}{6!} - \frac{h^8}{8!} + \dots, \\
 f_{1z}(x_{i+1,j}, z_{i+1,j}) &= -h + \frac{h^3}{3!} - \frac{h^5}{5!} + \frac{h^7}{7!} - \frac{h^9}{9!} + \dots, \\
 \\
 f_2(x_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{i+1,j} - DZ2_{i+1,j} - \dots - DZn_{i+1,j}, \\
 f_2(x_{i+1,j}, z_{i+1,j}) &= z_i + x_{i+1} \cdot (h - \frac{h^3}{3!} + \dots) + z_{i+1} \cdot (-1 + \frac{h^2}{2!} - \dots), \\
 f_{2x}(x_{i+1,j}, z_{i+1,j}) &= h - \frac{h^3}{3!} + \frac{h^5}{5!} - \frac{h^7}{7!} + \frac{h^9}{9!} - \dots, \\
 f_{2z}(x_{i+1,j}, z_{i+1,j}) &= -1 + \frac{h^2}{2!} - \frac{h^4}{4!} + \frac{h^6}{6!} - \frac{h^8}{8!} + \dots.
 \end{aligned}$$

V Tab. 5.12, 5.13 je zobrazena absolutní chyba výpočtu implicitní Taylorovy metody řádu 8 využívající Newtonovu metodu a analyticky vypočtenou Jacobiovu matici. Počet potřebných iterací v průběhu výpočtu pro zadanou přesnost TOL (ukončení výpočtu Newtonovy metody) je značen proměnnou j . Absolutní chyba výpočtu se opět shoduje s dříve zmíněnými metodami.

Tabulka 5.12: Absolutní chyba výpočtu, implicitní Taylorova metoda (analyticky vyjádřená Jacobiova matice), $h = 0.1$, $ORD = 8$, $TOL = 10^{-20}$

t	$ \text{Error}(x) $	$ \text{Error}(z) $	j
0.1	4.44089×10^{-16}	2.70617×10^{-15}	3
0.2	1.55431×10^{-15}	5.27356×10^{-15}	2
0.3	3.10862×10^{-15}	7.66054×10^{-15}	2
0.4	5.10703×10^{-15}	9.76996×10^{-15}	3
0.5	7.54952×10^{-15}	1.14353×10^{-14}	3
0.6	1.04361×10^{-14}	1.27676×10^{-14}	2
0.7	1.35447×10^{-14}	1.35447×10^{-14}	2
0.8	1.69864×10^{-14}	1.37668×10^{-14}	2
0.9	2.06501×10^{-14}	1.35447×10^{-14}	2
1	2.43139×10^{-14}	1.26565×10^{-14}	2

Tabulka 5.13: Absolutní chyba výpočtu, implicitní Taylorova metoda (analyticky vyjádřená Jacobiova matice), $h = 0.5$, $ORD = 8$, $TOL = 10^{-10}$

t	$ \text{Error}(x) $	$ \text{Error}(z) $	j
0.5	4.3718×10^{-9}	3.13157×10^{-9}	2
1	1.06759×10^{-8}	1.30452×10^{-9}	2
1.5	1.49917×10^{-8}	5.96025×10^{-9}	2
2	1.37319×10^{-8}	1.65573×10^{-8}	2

Numericky vypočtená Jacobiova matice

V některých případech je analytické nalezení derivace složité, mnohdy nemožné. Vypočteme tedy potřebné parciální derivace v soustavě (5.38) obecně pomocí diferenčního podílu.

Mějme parciální derivace ve tvaru

$$\begin{aligned}
 f_1(x_{i+1,j}, z_{i+1,j}) &= -x_{i+1,j} + x_i - DX1_{i+1,j} - DX2_{i+1,j} - \dots - DXn_{i+1,j}, \\
 f_{1x}(x_{i+1,j}, z_{i+1,j}) &= \frac{f_1(x_{i+1,j}+h_N, z_{i+1,j}) - f_1(x_{i+1,j}, z_{i+1,j})}{h_N}, \\
 f_{1z}(x_{i+1,j}, z_{i+1,j}) &= \frac{f_1(x_{i+1,j}, z_{i+1,j}+h_N) - f_1(x_{i+1,j}, z_{i+1,j})}{h_N}, \\
 f_1(x_{i+1,j} + h_N, z_{i+1,j}) &= -(x_{i+1,j} + h_N) + x_i - DX1_{Nx, i+1,j} - \dots - DXn_{Nx, i+1,j}, \\
 f_1(x_{i+1,j}, z_{i+1,j} + h_N) &= -x_{i+1,j} + x_i - DX1_{Nz, i+1,j} - \dots - DXn_{Nz, i+1,j}, \\
 \\
 f_2(x_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{i+1,j} - DZ2_{i+1,j} - \dots - DZn_{i+1,j}, \\
 f_{2x}(x_{i+1,j}, z_{i+1,j}) &= \frac{f_2(x_{i+1,j}+h_N, z_{i+1,j}) - f_2(x_{i+1,j}, z_{i+1,j})}{h_N}, \\
 f_{2z}(x_{i+1,j}, z_{i+1,j}) &= \frac{f_2(x_{i+1,j}, z_{i+1,j}+h_N) - f_2(x_{i+1,j}, z_{i+1,j})}{h_N}, \\
 f_2(x_{i+1,j} + h_N, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{Nx, i+1,j} - \dots - DZn_{Nx, i+1,j}, \\
 f_2(x_{i+1,j}, z_{i+1,j} + h_N) &= -(z_{i+1,j} + h_N) + z_i - DZ1_{Nz, i+1,j} - \dots - DZn_{Nz, i+1,j}.
 \end{aligned}$$

Nové členy $DX(l)_{Nx, i+1,j}$, $DZ(l)_{Nx, i+1,j}$, $DX(l)_{Nz, i+1,j}$, $DZ(l)_{Nz, i+1,j}$ pro $l = 1, 2, \dots, n$ vyjádříme ve tvaru

$$\begin{aligned}
 DX1_{Nx, i+1,j} &= -h(-z_{i+1,j}), & DZ1_{Nx, i+1,j} &= -h(x_{i+1,j} + h_N), \\
 DX2_{Nx, i+1,j} &= -\frac{h}{2}(-DZ1_{Nx}), & DZ2_{Nx, i+1,j} &= -\frac{h}{2}DX1_{Nx, i+1,j}, \\
 &\vdots & &\vdots \\
 DXn_{Nx, i+1,j} &= -\frac{h}{n}(-DZ(n-1)_{Nx}), & DZn_{Nx, i+1,j} &= -\frac{h}{n}DX(n-1)_{Nx, i+1,j}, \\
 \\
 DX1_{Nz, i+1,j} &= -h(-(z_{i+1,j} + h_N)), & DZ1_{Nz, i+1,j} &= -hx_{i+1,j}, \\
 DX2_{Nz, i+1,j} &= -\frac{h}{2}(-DZ1_{Nz, i+1,j}), & DZ2_{Nz, i+1,j} &= -\frac{h}{2}DX1_{Nz, i+1,j}, \\
 &\vdots & &\vdots \\
 DXn_{Nz, i+1,j} &= -\frac{h}{n}(-DZ(n-1)_{Nz, i+1,j}), & DZn_{Nz, i+1,j} &= -\frac{h}{n}DX(n-1)_{Nz, i+1,j}.
 \end{aligned} \tag{5.39}$$

V Tab. 5.14, 5.15 je zobrazena absolutní chyba výpočtu implicitní Taylorovy metody řádu 8 využívající Newtonovu metodu a numericky vypočtenou Jacobiovu matici. Počet potřebných iterací v průběhu výpočtu pro zadanou přesnost TOL (ukončení výpočtu Newtonovy metody) je značen proměnnou j . Absolutní chyba výpočtu se shoduje s chybou u implicitní Taylorovy metody s analyticky vyjádřenou Jacobiovou maticí. Celkový počet iterací v průběhu výpočtu (v případě integračního kroku $h = 0.1$) je u numerického výpočtu derivace o 3 větší oproti analytickému vyjádření parciálních derivací viz Tab. 5.12, 5.14.

Závěr

Na základním testovacím příkladě (5.35) jsme demonstrovali numerický výpočet pomocí explicitní a implicitní Taylorovy metody. Z předchozích tabulek lze vidět, že chyba výpočtu je ve všech případech stejného řádu. U implicitní Taylorovy metody využívající Newtonovu metodu lze Jacobiovu matici vyjádřit u předem známých soustav diferenciálních rovnic

Tabulka 5.14: Absolutní chyba výpočtu, implicitní Taylorova metoda (numericky vypočtená Jacobiova matice), $h = 0.1$, $h_N = 0.1$, $ORD = 8$, $TOL = 10^{-20}$

t	$ \text{Error}(x) $	$ \text{Error}(z) $	j
0.1	4.44089×10^{-16}	2.70617×10^{-15}	3
0.2	1.55431×10^{-15}	5.27356×10^{-15}	2
0.3	3.10862×10^{-15}	7.66054×10^{-15}	3
0.4	5.10703×10^{-15}	9.76996×10^{-15}	3
0.5	7.54952×10^{-15}	1.14353×10^{-14}	3
0.6	1.04361×10^{-14}	1.27676×10^{-14}	3
0.7	1.35447×10^{-14}	1.35447×10^{-14}	2
0.8	1.69864×10^{-14}	1.37668×10^{-14}	2
0.9	2.06501×10^{-14}	1.35447×10^{-14}	3
1	2.43139×10^{-14}	1.26565×10^{-14}	2

Tabulka 5.15: Absolutní chyba výpočtu, implicitní Taylorova metoda (numericky vypočtená Jacobiova matice), $h = 0.5$, $h_N = 0.1$, $ORD = 8$, $TOL = 10^{-10}$

t	$ \text{Error}(x) $	$ \text{Error}(z) $	j
0.5	4.37364×10^{-9}	3.12749×10^{-9}	2
1	1.06752×10^{-8}	1.29559×10^{-9}	2
1.5	1.49843×10^{-8}	5.97149×10^{-9}	2
2	1.37161×10^{-8}	1.65658×10^{-8}	2

(tedy především tvořících diferenciálních rovnic) analyticky. U neznámých systémů lze počítat Jacobiovou matici automaticky s využitím diferenční formule. V daném příkladě velikost kroku h_N v diferenční formuli nemá rozhodující vliv na počet iterací. Daleko větší vliv na počet iterací v Newtonově metodě má zadaná chyba výpočtu TOL (ukončovací podmínka u Newtonovy metody).

V tomto konkrétním případě (generování funkcí $\sin(t)$ a $\cos(t)$) se nejedná o tuhý systém diferenciálních rovnic. Vhodné je tedy použít explicitní Taylorovu metodu. Implicitní Taylorova metoda využívající Newtonovu metodu musí v každém integračním kroku provádět mezivýpočty, proto není vhodná pro tento typ příkladu. Její výhody (větší oblast stability) se projeví až u systémů s “vysokou tuhostí”.

5.6.3 Problém stability

Mějme diferenciální rovnici (označená v [HW02] jako “Stability problem”) ve tvaru

$$y' = -2000(y - \cos(t)), \quad y(0) = 0, \quad t \in (0, 1.5), \quad (5.40)$$

jejíž analytické řešení je podobné jako RC obvodu viz podkapitola 4.3.1

$$y = \frac{4000000}{4000001} \cos(t) + \frac{2000}{4000001} \sin(t) - \frac{4000000}{4000001} e^{-2000t}. \quad (5.41)$$

Řešme nejprve problém (5.40) běžnými explicitními numerickými metodami. V Tab. 5.16 je zobrazena maximální velikost integračního kroku h při řešení problému (5.40) pomocí běžných explicitních metod.

Zhodnotme řešení rovnice (5.40) pomocí implicitních numerických metod.

Tabulka 5.16: Explicitní numerické metody - maximální velikost h , $EPS = 1 \times 10^{-15}$

metoda	h
Eulerova metoda	1.233816×10^{-16}
metoda Runge-Kutta 2.řádu	1.953125×10^{-11}
metoda Runge-Kutta 4.řádu	7.629394×10^{-7}
Taylorova metoda ($ORD = 15$)	1.953125×10^{-4}

Lichoběžníková metoda

Připomeňme známý tvar Lichoběžníkové metody

$$y_{i+1} = y_i + \frac{1}{2}(hy'_i + hy'_{i+1}).$$

Řešení (5.40) pomocí lichoběžníkové metody je ve tvaru

$$y_{i+1} = \frac{y_i(1 - 1000h) + 1000h(\cos(t+h) + \cos(t))}{1 + 1000h}. \quad (5.42)$$

Implicitní Eulerova metoda

Připomeňme známý tvar implicitní Eulerovy metody

$$y_{i+1} = y_i + hy'_{i+1}.$$

Řešení (5.40) pomocí implicitní Eulerovy metody bude ve tvaru

$$y_{i+1} = \frac{y_i + 2000h \cos(t+h)}{1 + 2000h}. \quad (5.43)$$

Implicitní Taylorova metoda

Připomeňme si obecný tvar implicitní Taylorovy řady

$$y_{i+1} = y_n + hy'_{i+1} - \frac{h^2}{2}y''_{i+1} + \frac{h^3}{3!}y'''_{i+1} - \dots - \frac{(-h)^n}{n!}y^{(n)}_{i+1}, \quad (5.44)$$

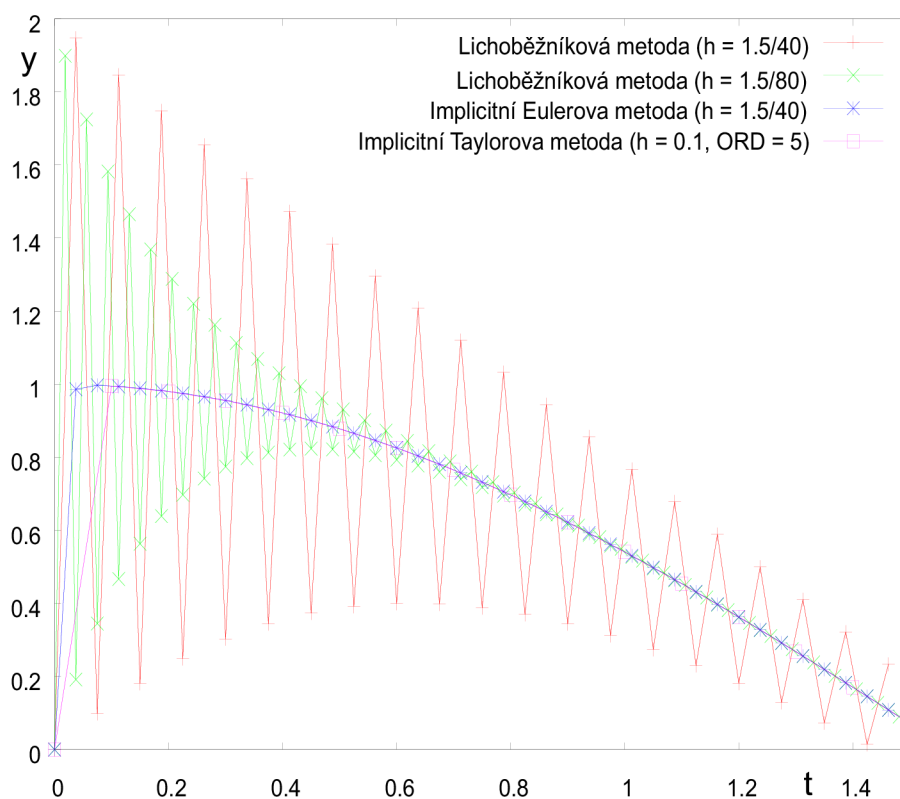
řešení (5.40) pomocí implicitní Taylorovy řady bude ve tvaru

$$\begin{aligned} y_{i+1} = & y_i + h(-2000(y_{i+1} - \cos(t+h))) - \\ & - \frac{h^2}{2}(-2000(-2000(y_{i+1} - \cos(t+h)) + \sin(t+h))) + \\ & + \frac{h^3}{3!}(-2000(-2000(-2000(y_{i+1} - \cos(t+h)) + \sin(t+h)) + \cos(t+h))) + \\ & - \frac{h^4}{4!}(-2000(-2000(-2000(-2000(y_{i+1} - \cos(t+h)) + \sin(t+h)) + \cos(t+h)) - \sin(t+h))) + \\ & + \dots \end{aligned}$$

Vyjádříme explicitně y_{i+1} ve tvaru

$$\begin{aligned} y_{i+1} = & \frac{y_i + h(-2000(-\cos(t+h))) - \frac{h^2}{2}(-2000(-2000(-\cos(t+h)) + \sin(t+h)))}{1 - h(-2000) + \frac{h^2}{2}(-2000)^2 - \frac{h^3}{3!}(-2000)^3 + \frac{h^4}{4!}(-2000)^4 - \dots} + \\ & + \frac{\frac{h^3}{3!}(-2000(-2000(-2000(-\cos(t+h)) + \sin(t+h)) + \cos(t+h)))}{1 - h(-2000) + \frac{h^2}{2}(-2000)^2 - \frac{h^3}{3!}(-2000)^3 + \frac{h^4}{4!}(-2000)^4 - \dots} - \\ & - \frac{\frac{h^4}{4!}(-2000(-2000(-2000(-2000(-\cos(t+h)) + \sin(t+h)) + \cos(t+h)) - \sin(t+h)))}{1 - h(-2000) + \frac{h^2}{2}(-2000)^2 - \frac{h^3}{3!}(-2000)^3 + \frac{h^4}{4!}(-2000)^4 - \dots} \end{aligned}$$

Na Obr. 5.22 je zobrazeno řešení problému (5.40) pomocí implicitních numerických metod. Vidíme, že nevhodnější řešení poskytuje implicitní Taylorova metoda, která umožňuje použití většího integračního kroku při větší přesnosti.



Obrázek 5.22: Řešení problému (5.40)

V Tab. 5.17 nalezneme absolutní chybu výpočtu v prvním kroku při řešení problému (5.40) pomocí implicitní Taylorovy metody různého řádu. S rostoucím řádem metody se zmenšuje absolutní chyba výpočtu, vhodné je také použití většího integračního kroku.

S rostoucím integračním krokem h roste přesnost výpočtu, což je vlastnost implicitních L-stabilních metod.

Tabulka 5.17: Absolutní chyba výpočtu v prvním kroku - implicitní Taylorova metoda

ORD	Error(y)			
	$h = 0.1$	$h = 0.5$	$h = 1.0$	$h = 1.5$
1	0.00499994	0.0011162	0.000690616	0.000522301
2	4.95031×10^{-5}	2.01115×10^{-6}	5.55226×10^{-7}	3.30287×10^{-7}
3	7.38803×10^{-7}	5.96771×10^{-9}	7.27429×10^{-10}	2.0562×10^{-10}
4	1.47015×10^{-8}	2.39015×10^{-11}	1.4877×10^{-12}	2.78513×10^{-13}
5	3.65672×10^{-10}	1.1946×10^{-13}	3.55271×10^{-15}	4.85723×10^{-16}
6	1.09144×10^{-11}	6.66134×10^{-16}	1.11022×10^{-16}	1.38778×10^{-17}

Kombinace implicitní Taylorovy metody s rekurentním výpočtem a Newtonovou metodou (ITMRN)

Původní rovnici ve tvaru

$$y' = -2000(y - \cos(t)), \quad y(0) = 0,$$

přepíšeme do tvaru

$$y' = \lambda y - \lambda x, \quad y(0) = 0, \quad (5.45)$$

kde $\lambda = -2000$ a $x = \cos(t)$. K výpočtu funkce $x = \cos(t)$ využijeme tvořící diferenciální rovnice (5.46) generující funkce $\sin(t)$ a $\cos(t)$ viz podkapitola 5.6.2

$$\begin{aligned} x' &= -z, & x(0) &= 1, \\ z' &= x, & z(0) &= 0. \end{aligned} \quad (5.46)$$

Vyjádríme předcházející diferenciální rovnice (5.45), (5.46) pomocí implicitní Taylorovy řady ve tvaru

$$\begin{aligned} y_{i+1} &= y_i - DY1_{i+1} - DY2_{i+1} - \dots - DYn_{i+1}, \\ x_{i+1} &= x_i - DX1_{i+1} - DX2_{i+1} - \dots - DXn_{i+1}, \\ z_{i+1} &= z_i - DZ1_{i+1} - DZ2_{i+1} - \dots - DZn_{i+1}, \end{aligned}$$

kde členy Taylorovy řady vyjádříme rekurentně ve tvaru

$$\begin{aligned} DY1_{i+1} &= -h(\lambda y_{i+1} - \lambda x_{i+1}), \\ DY2_{i+1} &= -\frac{h}{2}(\lambda DY1_{i+1} - \lambda DX1_{i+1}), \\ &\vdots \\ DYn_{i+1} &= -\frac{h}{n}(\lambda DY(n-1)_{i+1} - \lambda DX(n-1)_{i+1}), \\ \\ DX1_{i+1} &= -h(-z_{i+1}), \\ DX2_{i+1} &= -\frac{h}{2}(-DZ1_{i+1}), \\ &\vdots \\ DXn_{i+1} &= -\frac{h}{n}(-DZ(n-1)_{i+1}), \\ \\ DZ1_{i+1} &= -h(x_{i+1}), \\ DZ2_{i+1} &= -\frac{h}{2}DX1_{i+1}, \\ &\vdots \\ DZn_{i+1} &= -\frac{h}{n}DX(n-1)_{i+1}. \end{aligned}$$

Hledáme tedy řešení soustavy tří rovnic f_1, f_2, f_3 o třech neznámých $x_{i+1}, y_{i+1}, z_{i+1}$

$$\begin{aligned} f_1(x_{i+1}, y_{i+1}, z_{i+1}) &= -y_{i+1} + y_i - DY1_{i+1} - DY2_{i+1} - \dots - DYn_{i+1} = 0, \\ f_2(x_{i+1}, y_{i+1}, z_{i+1}) &= -x_{i+1} + x_i - DX1_{i+1} - DX2_{i+1} - \dots - DXn_{i+1} = 0, \\ f_3(x_{i+1}, y_{i+1}, z_{i+1}) &= -z_{i+1} + z_i - DZ1_{i+1} - DZ2_{i+1} - \dots - DZn_{i+1} = 0. \end{aligned} \quad (5.47)$$

Numerický výpočet $y_{i+1}, x_{i+1}, z_{i+1}$ provedeme pomocí Newtonovy metody a Cramerova pravidla (podobně jako v předchozím příkladě 5.6.2)

$$\begin{aligned} y_{i+1,j+1} &= y_{i+1,j} + a_{i+1,j}, \\ x_{i+1,j+1} &= x_{i+1,j} + b_{i+1,j}, \\ z_{i+1,j+1} &= z_{i+1,j} + c_{i+1,j}, \end{aligned}$$

kde členy $a_{i+1,j}$, $b_{i+1,j}$, $c_{i+1,j}$ vyjádříme pomocí Cramerova pravidla

$$\begin{aligned} a_{i+1,j} &= -\frac{f_1 f_2 x f_3 z + f_1 x f_2 z f_3 + f_1 z f_2 f_3 x - f_3 f_2 x f_1 z - f_3 x f_2 z f_1 - f_3 z f_2 f_1 x}{f_1 y f_2 x f_3 z + f_1 x f_2 z f_3 y + f_1 z f_2 y f_3 x - f_3 y f_2 x f_1 z - f_3 x f_2 z f_1 y - f_3 z f_2 y f_1 x}, \\ b_{i+1,j} &= -\frac{f_1 y f_2 f_3 z + f_1 f_2 z f_3 y + f_1 z f_2 y f_3 x - f_3 y f_2 f_1 z - f_3 f_2 z f_1 y - f_3 z f_2 y f_1 x}{f_1 y f_2 x f_3 z + f_1 x f_2 z f_3 y + f_1 z f_2 y f_3 x - f_3 y f_2 x f_1 z - f_3 x f_2 z f_1 y - f_3 z f_2 y f_1 x}, \\ c_{i+1,j} &= -\frac{f_1 y f_2 x f_3 z + f_1 x f_2 z f_3 y + f_1 z f_2 y f_3 x - f_3 y f_2 x f_1 z - f_3 x f_2 z f_1 y - f_3 z f_2 y f_1 x}{f_1 y f_2 x f_3 z + f_1 x f_2 z f_3 y + f_1 z f_2 y f_3 x - f_3 y f_2 x f_1 z - f_3 x f_2 z f_1 y - f_3 z f_2 y f_1 x}, \end{aligned}$$

dále f_{1x} , f_{1y} , f_{1z} , f_{2x} , f_{2y} , f_{2z} , f_{3x} , f_{3y} , f_{3z} jsou parciální derivace - členy Jacobiovy matice soustavy (5.47). Parciální derivace vyjádříme numericky pomocí diferenční formule např. $f_{1x} = (f_1(x + h_N, y, z) - f_1(x, y, z))/h_N$.

$$\begin{aligned} f_1(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= -y_{i+1,j} + y_i - DY1_{i+1,j} - DY2_{i+1,j} - \dots - DYn_{i+1,j}, \\ f_{1y}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_1(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) - f_1(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{1x}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_1(y_{i+1,j}, x_{i+1,j} + h_N, z_{i+1,j}) - f_1(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{1z}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_1(y_{i+1,j}, x_{i+1,j}, z_{i+1,j} + h_N) - f_1(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_1(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) &= -x_{i+1,j} + x_i - DY1_{Ny,i+1,j} - \dots - DYn_{Ny,i+1,j}, \\ f_1(y_{i+1,j}, x_{i+1,j} + h_N, z_{i+1,j}) &= -(x_{i+1,j} + h_N) + x_i - DY1_{Nx,i+1,j} - \dots - DYn_{Nx,i+1,j}, \\ f_1(y_{i+1,j}, x_{i+1,j}, z_{i+1,j} + h_N) &= -x_{i+1,j} + x_i - DY1_{Nz,i+1,j} - \dots - DYn_{Nz,i+1,j}, \end{aligned}$$

$$\begin{aligned} f_2(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= -x_{i+1,j} + x_i - DX1_{i+1,j} - DX2_{i+1,j} - \dots - DXn_{i+1,j}, \\ f_{2y}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_2(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) - f_2(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{2x}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_2(y_{i+1,j}, x_{i+1,j} + h_N, z_{i+1,j}) - f_2(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{2z}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_2(y_{i+1,j}, x_{i+1,j}, z_{i+1,j} + h_N) - f_2(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_2(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) &= -x_{i+1,j} + x_i - DX1_{Ny,i+1,j} - \dots - DXn_{Ny,i+1,j}, \\ f_2(y_{i+1,j}, x_{i+1,j} + h_N, z_{i+1,j}) &= -(x_{i+1,j} + h_N) + x_i - DX1_{Nx,i+1,j} - \dots - DXn_{Nx,i+1,j}, \\ f_2(y_{i+1,j}, x_{i+1,j}, z_{i+1,j} + h_N) &= -x_{i+1,j} + x_i - DX1_{Nz,i+1,j} - \dots - DXn_{Nz,i+1,j}, \end{aligned}$$

$$\begin{aligned} f_3(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{i+1,j} - DZ2_{i+1,j} - \dots - DZn_{i+1,j}, \\ f_{3y}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_3(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) - f_3(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{3x}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_3(y_{i+1,j}, x_{i+1,j} + h_N, z_{i+1,j}) - f_3(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{3z}(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}) &= \frac{f_3(y_{i+1,j}, x_{i+1,j}, z_{i+1,j} + h_N) - f_3(y_{i+1,j}, x_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_3(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{Ny,i+1,j} - \dots - DZn_{Ny,i+1,j}, \\ f_3(y_{i+1,j}, x_{i+1,j} + h_N, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{Nx,i+1,j} - \dots - DZn_{Nx,i+1,j}, \\ f_3(y_{i+1,j}, x_{i+1,j}, z_{i+1,j} + h_N) &= -(z_{i+1,j} + h_N) + z_i - DZ1_{Nz,i+1,j} - \dots - DZn_{Nz,i+1,j}. \end{aligned}$$

Nové členy $DY(l)_{Ny,i+1,j}$, $DY(l)_{Nx,i+1,j}$, $DY(l)_{Nz,i+1,j}$, $DX(l)_{Ny,i+1,j}$, $DZ(l)_{Ny,i+1,j}$ pro $l = 1, 2, \dots, n$ vyjádříme ve tvaru

$$\begin{aligned}
DY1_{Ny,i+1,j} &= -h(\lambda(y_{i+1,j} + h_N) - \lambda z_{i+1,j}), \\
DY2_{Ny,i+1,j} &= -\frac{h}{2}(\lambda DY1_{Ny,i+1,j} - \lambda DZ1_{Ny,i+1,j}), \\
&\vdots \\
DYn_{Ny,i+1,j} &= -\frac{h}{n}(\lambda DY(n-1)_{Ny,i+1,j} - \lambda DZ(n-1)_{Ny,i+1,j}), \\
\\
DY1_{Nx,i+1,j} &= -h(\lambda y_{i+1,j} - \lambda z_{i+1,j}), \\
DY2_{Nx,i+1,j} &= -\frac{h}{2}\lambda DY1_{Nx,i+1,j} - \lambda DZ1_{Nx,i+1,j}, \\
&\vdots \\
DYn_{Nx,i+1,j} &= -\frac{h}{n}(\lambda DY(n-1)_{Nx,i+1,j} - \lambda DZ(n-1)_{Nx,i+1,j}), \\
\\
DY1_{Nz,i+1,j} &= -h(\lambda(y_{i+1,j}) - \lambda(z_{i+1,j} + h_N)), \\
DY2_{Nz,i+1,j} &= -\frac{h}{2}(-DZ1_{Nz,i+1,j}), \\
&\vdots \\
DYn_{Nz,i+1,j} &= -\frac{h}{n}(-DZ(n-1)_{Nz,i+1,j}), \\
\\
DX1_{Ny,i+1,j} &= -h(-z_{i+1,j}), \\
DX2_{Ny,i+1,j} &= -\frac{h}{2}(-DZ1_{Ny,i+1,j}), \\
&\vdots \\
DXn_{Ny,i+1,j} &= -\frac{h}{n}(-DZ(n-1)_{Ny,i+1,j}),
\end{aligned}$$

$$\begin{aligned}
DX1_{Ny,i+1,j} &= -h(-z_{i+1,j}), & DZ1_{Ny,i+1,j} &= -h(x_{i+1,j}), \\
DX2_{Ny,i+1,j} &= -\frac{h}{2}(-DZ1_{Ny}), & DZ2_{Ny,i+1,j} &= -\frac{h}{2}DX1_{Ny,i+1,j}, \\
&\vdots & & \vdots \\
DXn_{Ny,i+1,j} &= -\frac{h}{n}(-DZ(n-1)_{Ny}), & DZn_{Ny,i+1,j} &= -\frac{h}{n}DX(n-1)_{Ny,i+1,j}.
\end{aligned}$$

Ostatní členy $DX(l)_{Nx,i+1,j}$, $DZ(l)_{Nx,i+1,j}$, $DX(l)_{Nz,i+1,j}$, $DZ(l)_{Nz,i+1,j}$ pro $l = 1, 2, \dots, n$ jsou shodné s předchozím příkladem (5.39).

OPTIMALIZACE: při podrobnějším prozkoumání předchozích členů zjistíme, že platí

$$\begin{aligned}
f_2(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) &= f_2(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}), \\
f_3(y_{i+1,j} + h_N, x_{i+1,j}, z_{i+1,j}) &= f_3(y_{i+1,j}, x_{i+1,j}, z_{i+1,j}),
\end{aligned}$$

a tedy pro parciální derivace platí $f_{2y} = f_{3y} = 0$. Vše vyplývá z navzájem propojených tvořících diferenciálních rovnic (5.46) generujících funkce $z = \sin(t)$ a $x = \cos(t)$, které jsou nezávislé na původní rovnici (5.45).

Nabízí se tedy nový tvar koeficientů $a_{i+1,j}$, $b_{i+1,j}$, $c_{i+1,j}$

$$\begin{aligned}
a_{i+1,j} &= -\frac{f_1 f_{2x} f_{3z} + f_{1x} f_{2z} f_3 + f_{1z} f_2 f_{3x} - f_3 f_{2x} f_{1z} - f_{3x} f_{2z} f_1 - f_{3z} f_2 f_{1x}}{f_{1y} f_{2x} f_{3z} - f_{3x} f_{2z} f_{1y}}, \\
b_{i+1,j} &= -\frac{f_{1y} f_2 f_{3z} - f_3 f_{2z} f_{1y}}{f_{1y} f_{2x} f_{3z} - f_{3x} f_{2z} f_{1y}}, \\
c_{i+1,j} &= -\frac{f_{1y} f_{2x} f_3 - f_{3x} f_2 f_{1y}}{f_{1y} f_{2x} f_{3z} - f_{3x} f_{2z} f_{1y}}.
\end{aligned}$$

Další možností je použít k výpočtu tvořících diferenciálních rovnic explicitně vyjádřený vzorec (5.37) a řešení dosadit do implicitního rozvoje Taylorovy řady. U tvořících diferenciálních rovnic tedy odpadne výpočet parciálních derivací a iterační výpočet Newtonovou metodou.

V Tab. 5.18, 5.19, 5.20, 5.21 nalezneme absolutní chybu výpočtu a počet Newtonových iterací j pro různý řád implicitní Taylorovy metody a různou délku integračního kroku.

Tabulka 5.18: Abs. chyba výpočtu ITMRN: $h = 0.25$, $h_N = 0.1$, $ORD = 5$, $TOL = 10^{-10}$

t	$ \text{Error}(y) $	$ \text{Error}(x) $	$ \text{Error}(z) $	j
0.25	3.03123×10^{-7}	3.03043×10^{-7}	1.51766×10^{-7}	2
0.5	5.12372×10^{-7}	5.1215×10^{-7}	4.44044×10^{-7}	2
0.75	5.79973×10^{-7}	5.79555×10^{-7}	8.35421×10^{-7}	2
1	4.73771×10^{-7}	4.73136×10^{-7}	1.27044×10^{-6}	2
1.25	1.80985×10^{-7}	1.80142×10^{-7}	1.68501×10^{-6}	2
1.5	2.89796×10^{-7}	2.90802×10^{-7}	2.01263×10^{-6}	2

Tabulka 5.19: Abs. chyba výpočtu ITMRN: $h = 0.5$, $h_N = 0.1$, $ORD = 5$, $TOL = 10^{-10}$

t	$ \text{Error}(y) $	$ \text{Error}(x) $	$ \text{Error}(z) $	j
0.5	1.29815×10^{-5}	1.29729×10^{-5}	1.73447×10^{-5}	2
1	6.16008×10^{-6}	6.13864×10^{-6}	4.28814×10^{-5}	2
1.5	2.2726×10^{-5}	2.27564×10^{-5}	6.0862×10^{-5}	2

Tabulka 5.20: Abs. chyba výpočtu ITMRN: $h = 0.5$, $h_N = 0.1$, $ORD = 10$, $TOL = 10^{-10}$

t	$ \text{Error}(y) $	$ \text{Error}(x) $	$ \text{Error}(z) $	j
0.5	9.99822×10^{-12}	1.00019×10^{-11}	7.02693×10^{-12}	2
1	2.42912×10^{-11}	2.42927×10^{-11}	2.74325×10^{-12}	2
1.5	3.39579×10^{-11}	3.39509×10^{-11}	1.38586×10^{-11}	2

Závěr

Implicitní Taylorova metoda se jeví jako vhodná při řešení problému (5.40). S rostoucím řádem metody jsme schopni provádět delší integrační krok při současně rostoucí přesnosti výpočtu. Jako zajímavá se jeví implementace implicitní Taylorovy metody s rekurentním výpočtem členů Newtonovou metodou a možnými optimalizacemi výpočtu u tvořících diferenciálních rovnic (není potřeba počítat parciální derivace u diferenciálních rovnic nezávislých na řešeném systému).

5.6.4 Problém stiff exponenciály

Mějme opět známou soustavu diferenciálních rovnic (viz podkapitola 4.2.1) ve tvaru

$$\begin{aligned} y' &= z, \\ z' &= -a \cdot y - (a + 1) \cdot z, \quad a \in (1, \infty), \end{aligned} \tag{5.48}$$

Tabulka 5.21: Absolutní chyba výpočtu ITMRN: $h = 1.5$, $h_N = 0.1$, $TOL = 10^{-10}$

<i>ORD</i>	Error(<i>y</i>)	Error(<i>x</i>)	Error(<i>z</i>)	<i>j</i>
1	0.236354	0.236955	0.535957	2
2	0.126078	0.12591	0.335426	2
3	0.21045	0.210475	0.0505399	2
4	0.026256	0.0262258	0.0602834	2
5	0.0144353	0.0144381	0.00558967	2
6	0.00107299	0.00107142	0.00315658	2
7	0.00059927	0.000599365	0.000189923	2
8	3.01715×10^{-5}	3.01212×10^{-5}	0.00010057	2
9	1.51633×10^{-5}	1.51655×10^{-5}	4.31964×10^{-6}	2
10	5.67345×10^{-7}	5.66307×10^{-7}	2.07689×10^{-6}	2
11	2.60475×10^{-7}	2.6051×10^{-7}	6.83692×10^{-8}	2
12	7.66305×10^{-9}	7.64798×10^{-9}	3.01443×10^{-8}	2
13	3.23702×10^{-9}	3.23741×10^{-9}	7.96952×10^{-10}	2
14	7.78821×10^{-11}	7.77199×10^{-11}	3.24391×10^{-10}	2
15	3.04601×10^{-11}	3.04637×10^{-11}	7.1223×10^{-12}	2
16	6.17006×10^{-13}	6.1566×10^{-13}	2.69207×10^{-12}	2
17	2.24515×10^{-13}	2.24543×10^{-13}	5.04041×10^{-14}	2
18	4.02456×10^{-15}	4.02456×10^{-15}	1.77636×10^{-14}	2
19	1.34615×10^{-15}	1.34615×10^{-15}	4.44089×10^{-16}	2

s počátečními podmínkami $y(0) = 1$, $z(0) = -1$.

Znamé analytické řešení je ve tvaru

$$\begin{aligned} y &= e^{-t}, \\ z &= -e^{-t}. \end{aligned} \tag{5.49}$$

Explicitní Taylorova metoda

Vyjádříme předcházející diferenciální rovnice (5.48) pomocí explicitní Taylorovy řady

$$\begin{aligned} y_{i+1} &= y_i + DY1_i + DY2_i + \dots + DYn_i, \\ z_{i+1} &= z_i + DZ1_i + DZ2_i + \dots + DZn_i, \end{aligned}$$

kde členy vyjádříme ve tvaru

$$\begin{aligned} DY1_i &= h z_i, & DZ1_i &= h(-a y_i - (a+1) z_i), \\ DY2_i &= \frac{h}{2} DZ1_i, & DZ2_i &= \frac{h}{2}(-a DY1_i - (a+1) DZ1_i), \\ &\vdots & &\vdots \\ DYn_i &= \frac{h}{n} DZ(n-1)_i, & DZn_i &= \frac{h}{n}(-a DY(n-1)_i - (a+1) DZ(n-1)_i). \end{aligned}$$

V Tab. 5.22 je uvedena absolutní chyba výpočtu explicitní Taylorovy řady $ORD = 1$ (tedy explicitní Eulerovy metody) v závislosti na velikosti konstanty a .

Tabulka 5.22: Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 0.1$, $ORD = 1$

t	Error(z)				
	$a = 10^4$	$a = 10^5$	$a = 10^6$	$a = 10^7$	$a = 10^8$
0.1	0.00483742	0.00483742	0.00483742	0.00483742	0.00483742
0.2	0.00873075	0.00873075	0.00873075	0.00873075	0.00873075
0.3	0.0118182	0.0118182	0.011818	0.011781	0.00585776
0.4	0.01422	0.0143073	0.0375021	37.2672	59604.6
0.5	0.0160768	0.856836	2328.16	3.72529×10^7	5.96046×10^{11}
0.6	0.0187225	8727.91	2.32816×10^8	3.72529×10^{13}	5.96046×10^{18}

S rostoucí hodnotou konstanty a se zvyšuje tuhost systému (5.48) - explicitní Eulerova metoda má problémy s výpočtem viz Tab. 5.22. Při řešení tuhého systému můžeme zvyšovat řád metody Taylorovy řady nebo snižovat integrační krok. V Tab. 5.23 je zobrazena absolutní chyba výpočtu v prvním kroku v závislosti na délce integračního kroku a řádu explicitní Taylorovy řady.

Tabulka 5.23: Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 0.1$, $a = 100$

t	Error(y)			
	$ORD = 1$	$ORD = 2$	$ORD = 3$	$ORD = 4$
0.1	0.00483742	0.000162582	4.0847×10^{-6}	8.1964×10^{-7}
0.2	0.00873075	0.000294247	7.39197×10^{-6}	1.48328×10^{-7}
0.3	0.0118182	0.000399404	1.00328×10^{-5}	2.0132×10^{-7}
0.4	0.01422	0.000481905	1.2104×10^{-5}	2.4302×10^{-7}
0.5	0.0160407	0.000545106	1.36903×10^{-5}	3.14994×10^{-7}
0.6	0.0173706	0.000591932	1.48514×10^{-5}	1.20207×10^{-5}

Zhodnoťme aritmetiku

Dále se nabízí možnost použití víceslovní aritmetiky, zajišťující přesný výpočet členů Taylorovy řady i pro větší hodnoty konstanty a . V Tab. 5.24 nalezneme potřebnou velikost čísla k výpočtu (5.48) pomocí explicitní Taylorovy řady s integračním krokem $h = 0.1$. Při zachování přesnosti výpočtu $EPS = 10^{-20}$ potřebujeme 9 členů Taylorovy řady.

Při řešení problému (5.48) s rostoucí konstantou a nemá vliv na stabilitu výpočtu využití vyšší oblasti stability numerické metody (tedy použití více členů explicitní Taylorovy řady). V tomto případě má rozhodující vliv na stabilitu a přesnost výpočtu rozšíření aritmetiky.

Zhodnoťme možnosti řešení problému (5.48) pomocí implicitní Taylorovy řady.

Tabulka 5.24: Explicitní Taylorova řada - víceslovní aritmetika

a	počet bitů
10^{10}	3681
10^{20}	8900
10^{50}	24000

Implicitní Taylorova metoda

Vyjádříme předcházející diferenciální rovnice (5.48) pomocí implicitní Taylorovy řady ve tvaru

$$\begin{aligned} y_{i+1} &= y_i + hy'_{i+1} - \frac{h^2}{2}y''_{i+1} + \frac{h^3}{3!}y'''_{i+1} - \dots - \frac{(-h)^n}{n!}y_{i+1}^{(n)}, \\ z_{i+1} &= z_i + hz'_{i+1} - \frac{h^2}{2}z''_{i+1} + \frac{h^3}{3!}z'''_{i+1} - \dots - \frac{(-h)^n}{n!}z_{i+1}^{(n)}, \end{aligned} \quad (5.50)$$

kde vyšší derivace vyjádříme ve tvaru

$$\begin{aligned} y'_{i+1} &= z_{i+1}, \\ z'_{i+1} &= -ay_{i+1} - (a+1)z_{i+1}, \\ y''_{i+1} &= z'_{i+1} = -ay_{i+1} - (a+1)z_{i+1}, \\ z''_{i+1} &= -ay'_{i+1} - (a+1)z'_{i+1} = -az_{i+1} - (a+1)(-ay_{i+1} - (a+1)z_{i+1}), \\ y'''_{i+1} &= z''_{i+1} = -az_{i+1} - (a+1)(-ay_{i+1} - (a+1)z_{i+1}), \\ z'''_{i+1} &= -a(-ay_{i+1} - (a+1)z_{i+1}) - (a+1)(-az_{i+1} - (a+1)(-ay_{i+1} - (a+1)z_{i+1})), \\ y_{i+1}^{(4)} &= z'''_{i+1} = -a(-ay_{i+1} - (a+1)z_{i+1}) - (a+1)(-az_{i+1} - (a+1)(-ay_{i+1} - (a+1)z_{i+1})), \\ z_{i+1}^{(4)} &= -a(-az_{i+1} - (a+1)z'_{i+1}) - (a+1)(-az'_{i+1} - (a+1)(-az_{i+1} - (a+1)z'_{i+1})), \\ &\vdots \end{aligned}$$

Ze soustavy (5.50) vyjádříme y_{i+1}, z_{i+1} pro řád metody $ORD = 1$

$$\begin{aligned} y_{i+1} &= \frac{y_i(ha+h+1)+z_i(h)}{1+h^2a+ha+h}, \\ z_{i+1} &= -\frac{y_i(ha)-z_i}{1+h^2a+ha+h}, \end{aligned}$$

obdobně pro řád metody $ORD = 2$ vyjádříme y_{i+1}, z_{i+1} ze soustavy (5.50)

$$\begin{aligned} y_{i+1} &= 2\frac{y_i(h^2a^2+h^2a+2ha+h^2+2h+2)+z_i(h^2a+h^2+2h)}{2h^2+2h^3a^2+h^4a^2+2h^3a+4+4ha+4h+4h^2a+2h^2a^2}, \\ z_{i+1} &= -2\frac{y_i(h^2a^2+2ha+h^2a)+z_i(h^2a-2)}{2h^2+2h^3a^2+h^4a^2+2h^3a+4+4ha+4h+4h^2a+2h^2a^2}, \end{aligned}$$

dále pro $ORD = 3$ vyjádříme y_{i+1}, z_{i+1} ve tvaru

$$\begin{aligned} y_{i+1} &= 6(y_i(h^3a^3+h^3a^2+h^3a+h^3+3h^2a^2+3h^2a+3h^2+6ha+6h+6)+ \\ &\quad +z_i(h^3a^2+h^3a+h^3+3h^2a+3h^2+6h))/(36+36h^2a+6h^3a^3+36ha+ \\ &\quad +3h^5a^3+3h^5a^2+h^6a^3+18h^2+18h^3a^2+36h+18h^3a+9h^4a^2+6h^4a^3+ \\ &\quad +6h^4a+6h^3+18h^2a^2), \\ z_{i+1} &= -6(y_i(3h^2a^2+6ha+3h^2a+h^3a+h^3a^2+h^3a^3)+z_i(-6+3h^2a+h^3a^2+h^3a))/ \\ &\quad / (36+36h^2a+6h^3a^3+36ha+3h^5a^3+3h^5a^2+h^6a^3+18h^2+18h^3a^2+ \\ &\quad +36h+18h^3a+9h^4a^2+6h^4a^3+6h^4a+6h^3+18h^2a^2), \end{aligned}$$

pro $ORD = 4$ vyjádříme y_{i+1}, z_{i+1} ve tvaru

$$\begin{aligned}
y_{i+1} &= 24(y_i(12h^2a^2 + 24ha + 12h^2a + 24 + 24h + 12h^2 + h^4 + 4h^3 + h^4a^4 + \\
&\quad + h^4a^3 + 4h^3a^2 + h^4a^2 + 4h^3a^2 + 4h^3a^3) + z_i(24h + 12h^2a + 12h^2 + \\
&\quad + h^4 + 4h^3 + h^4a^3 + h^4a + 4h^3a^2 + 4h^3a + h^4a^2))/(h^8a^4 + 48h^5a^3 + \\
&\quad + 48h^5a^2 + 16h^6a^3 + 96h^4a^3 + 96h^4a + 96h^4a + 576h^2a + 576h + 288h^2a^2 + \\
&\quad + 288h^3a + 144h^4a^2 + 4h^7a^3 + 24h^5a^4 + 24h^5a + 24h^4a^4 + 288h^2 + 24h^4 + \\
&\quad + 96h^3 + 96h^3a^3 + 576ha + 576 + 12h^6a^4 + 12h^6a^2 + 4h^7a^4), \\
z_{i+1} &= -24(y_i(12h^2a^2 + 24ha + 12h^2a + h^4a^4 + h^4a + h^4a^3 + 4h^3a + h^4a^2 + \\
&\quad + 4h^3a^2 + 4h^3a^3) + z_i(-24 + 12h^2a + h^4a^3 + h^4a + 4h^3a^2 + 4h^3 + h^4a^2))/ \\
&\quad (h^8a^4 + 48h^5a^3 + 48h^5a^2 + 16h^6a^3 + 96h^4a^3 + 96h^4a + 96h^4a + 576h^2a + \\
&\quad + 576h + 288h^2a^2 + 288h^3a + 144h^4a^2 + 4h^7a^3 + 24h^5a^4 + 24h^5a + 24h^4a^4 + \\
&\quad + 288h^2 + 24h^4 + 96h^3 + 96h^3a^3 + 576ha + 576 + 12h^6a^4 + 12h^6a^2 + 4h^7a^4),
\end{aligned}$$

V Tab. 5.25 je zobrazena absolutní chyba výpočtu (pro různé velikosti konstanty a) implicitní Taylorovy metody pro $ORD = 1, 2, 3, 4$. Důležité je, že velikost konstanty a nemá vliv na chybu výpočtu.

Tabulka 5.25: Absolutní chyba výpočtu - implicitní Taylorova metoda: $h = 0.1$

	Error(y) : $a = 10^4, 10^5, 10^6, 10^7, 10^8$			
t	$ORD = 1$	$ORD = 2$	$ORD = 3$	$ORD = 4$
0.1	0.00425349	0.000139958	3.48077×10^{-6}	6.93811×10^{-8}
0.2	0.00771553	0.000253297	6.29908×10^{-6}	1.25557×10^{-7}
0.3	0.0104966	0.000343816	8.54948×10^{-6}	1.70413×10^{-7}
0.4	0.0126934	0.000414829	1.03145×10^{-5}	2.05595×10^{-7}
0.5	0.0143907	0.000469227	1.16662×10^{-5}	2.32538×10^{-7}
0.6	0.0156623	0.000509528	1.26673×10^{-5}	2.52491×10^{-7}

Zaměříme se nyní na možnosti rekurentního výpočtu členů implicitní Taylorovy řady s Newtonovou metodou.

Kombinace implicitní Taylorovy metody s rekurentním výpočtem a Newtonovou metodou (ITMRN)

Vyjádříme předcházející diferenciální rovnice (5.48) pomocí rekurentní implicitní Taylorovy řady ve tvaru

$$\begin{aligned}
y_{i+1} &= y_i - DY1_{i+1} - DY2_{i+1} - \dots - DYn_{i+1}, \\
z_{i+1} &= z_i - DZ1_{i+1} - DZ2_{i+1} - \dots - DZn_{i+1},
\end{aligned}$$

kde členy vyjádříme ve tvaru

$$\begin{aligned}
DY1_{i+1} &= -hz_{i+1}, & DZ1_{i+1} &= -h(-ay_{i+1} - (a+1)z_{i+1}), \\
DY2_{i+1} &= -\frac{h}{2}DZ1_{i+1}, & DZ2_{i+1} &= -\frac{h}{2}(-aDY1_{i+1} - (a+1)DZ1_{i+1}), \\
&\vdots & &\vdots \\
DYn_{i+1} &= -\frac{h}{n}DZ(n-1)_{i+1}, & DZn_{i+1} &= -\frac{h}{n}(-aDY(n-1)_{i+1} - (a+1)DZ(n-1)_{i+1}).
\end{aligned}$$

Numerický výpočet y_{i+1}, z_{i+1} provedeme opět pomocí Newtonovy metody a Cramerova pravidla

$$\begin{aligned} y_{i+1,j+1} &= y_{i+1,j} + b_{i+1,j}, \\ z_{i+1,j+1} &= z_{i+1,j} + c_{i+1,j}, \end{aligned}$$

kde členy $b_{i+1,j}, c_{i+1,j}$ vyjádříme pomocí Cramerova pravidla

$$\begin{aligned} b_{i+1,j} &= -\frac{D_y}{D} = -\frac{f_1 f_{2z} - f_2 f_{1z}}{f_{1y} f_{2z} - f_{2y} f_{1z}}, \\ c_{i+1,j} &= -\frac{D_z}{D} = -\frac{f_1 f_2 - f_1 f_{2y}}{f_{1y} f_{2z} - f_{2y} f_{1z}}, \end{aligned}$$

$$\begin{aligned} f_1(y_{i+1,j}, z_{i+1,j}) &= -y_{i+1,j} + y_i - DY1_{i+1,j} - DY2_{i+1,j} - \dots - DYn_{i+1,j}, \\ f_{1y}(y_{i+1,j}, z_{i+1,j}) &= \frac{f_1(y_{i+1,j} + h_N, z_{i+1,j}) - f_1(y_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{1z}(y_{i+1,j}, z_{i+1,j}) &= \frac{f_1(y_{i+1,j}, z_{i+1,j} + h_N) - f_1(y_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_1(y_{i+1,j} + h_N, z_{i+1,j}) &= -(y_{i+1,j} + h_N) + y_i - DY1_{Ny,i+1,j} - \dots - DYn_{Ny,i+1,j}, \\ f_1(y_{i+1,j}, z_{i+1,j} + h_N) &= -y_{i+1,j} + y_i - DY1_{Nz,i+1,j} - \dots - DYn_{Nz,i+1,j}, \end{aligned}$$

$$\begin{aligned} f_2(y_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{i+1,j} - DZ2_{i+1,j} - \dots - DZn_{i+1,j}, \\ f_{2y}(y_{i+1,j}, z_{i+1,j}) &= \frac{f_2(y_{i+1,j} + h_N, z_{i+1,j}) - f_2(y_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_{2z}(y_{i+1,j}, z_{i+1,j}) &= \frac{f_2(y_{i+1,j}, z_{i+1,j} + h_N) - f_2(y_{i+1,j}, z_{i+1,j})}{h_N}, \\ f_2(y_{i+1,j} + h_N, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{Ny,i+1,j} - \dots - DZn_{Ny,i+1,j}, \\ f_2(y_{i+1,j}, z_{i+1,j} + h_N) &= -(z_{i+1,j} + h_N) + z_i - DZ1_{Nz,i+1,j} - \dots - DZn_{Nz,i+1,j}. \end{aligned}$$

Nové členy $DY(l)_{Ny,i+1,j}, DZ(l)_{Ny,i+1,j}, DY(l)_{Nz,i+1,j}, DZ(l)_{Nz,i+1,j}$ pro $l = 1, 2, \dots, n$ vyjádříme ve tvaru

$$\begin{aligned} DY1_{Ny,i+1,j} &= -hz_{i+1,j}, \\ DY2_{Ny,i+1,j} &= -\frac{h}{2} DZ1_{Ny,i+1,j}, \\ &\vdots \\ DYn_{Ny,i+1,j} &= -\frac{h}{n} DZ(n-1)_{Ny,i+1,j}, \end{aligned}$$

$$\begin{aligned} DY1_{Nz,i+1,j} &= -h(z_{i+1,j} + h_N), \\ DY2_{Nz,i+1,j} &= -\frac{h}{2} DZ1_{Nz,i+1,j}, \\ &\vdots \\ DYn_{Nz,i+1,j} &= -\frac{h}{n} DZ(n-1)_{Nz,i+1,j}, \end{aligned}$$

$$\begin{aligned} DZ1_{Ny,i+1,j} &= -h(-a(y_{i+1,j} + h_N) - (a+1)z_{i+1,j}), \\ DZ2_{Ny,i+1,j} &= -\frac{h}{2}(-aDY1_{Ny,i+1,j} - (a+1)DZ1_{Ny,i+1,j}), \\ &\vdots \\ DZn_{Ny,i+1,j} &= -\frac{h}{n}(-aDY(n-1)_{Ny,i+1,j} - (a+1)DZ(n-1)_{Ny,i+1,j}), \end{aligned}$$

$$\begin{aligned}
DZ1_{Nz,i+1,j} &= -h(-ay_{i+1,j} - (a+1)(z_{i+1,j} + h_N)), \\
DZ2_{Nz,i+1,j} &= -\frac{h}{2}(-aDY1_{Nz,i+1,j} - (a+1)DZ1_{Nz,i+1,j}), \\
&\vdots \\
DZn_{Nz,i+1,j} &= -\frac{h}{n}(-aDY(n-1)_{Nz,i+1,j} - (a+1)DZ(n-1)_{Nz,i+1,j}).
\end{aligned}$$

Absolutní chyby výpočtu pro ITMRN $ORD = 1, 2, 3, 4$ při nastavené chybě výpočtu $TOL = 10^{-10}$ jsou shodné s Tab. 5.25. V průběhu výpočtu (se zadanou přesností výpočtu $TOL = 10^{-10}$) jsou potřeba v každém kroku vždy pouze 2 iterace u Newtonovy metody.

V Tab. 5.26 je zobrazena absolutní chyba výpočtu ITMRN pro $ORD = 5, 6, 7, 8$ a počet iterací j v Newtonově metodě potřebných v průběhu výpočtu k dosažení přesnosti $TOL = 10^{-10}$.

Důležité je, že stále platí - velikost konstanty a (do určité hodnoty v závislosti na použité aritmetice) nemá vliv na absolutní chybu výpočtu, ani na počet potřebných iterací v Newtonově metodě.

Tabulka 5.26: Absolutní chyba výpočtu ITMRN: $h = 0.1$, $ORD = 5, 6, 7, 8$, $TOL = 10^{-10}$, $a = 10^4$

	$ORD = 5$		$ORD = 6$		$ORD = 7$		$ORD = 8$	
t	$ \text{Error}(y) $	j	$ \text{Error}(y) $	j	$ \text{Error}(y) $	j	$ \text{Error}(y) $	j
0.1	1.153×10^{-9}	3	1.644×10^{-11}	3	2.035×10^{-13}	5	2.148×10^{-12}	9
0.2	2.087×10^{-9}	3	2.976×10^{-11}	3	3.459×10^{-13}	5	2.753×10^{-14}	8
0.3	2.833×10^{-9}	3	4.04×10^{-11}	3	4.842×10^{-13}	5	6.328×10^{-14}	19
0.4	3.418×10^{-9}	3	4.874×10^{-11}	3	5.898×10^{-13}	5	2.846×10^{-12}	15
0.5	3.866×10^{-9}	3	5.513×10^{-11}	3	6.354×10^{-13}	4	2.587×10^{-12}	11
0.6	4.198×10^{-9}	3	5.986×10^{-11}	3	6.998×10^{-13}	5	4.938×10^{-12}	7

Při použití vyššího řádu $ORD > 7$ (při zachování aritmetiky - 64 bitů) se objeví nárůst počtu iterací v Newtonově metodě viz Tab. 5.26, přičemž se přesnost výpočtu již nezvýší. Nepomůže ani zvýšení přesnosti výpočtu pomocí hodnoty TOL u Newtonovy metody.

Pro zachování stability a snížení počtu iterací v Newtonově metodě, pomůže snížení integračního kroku viz Tab. 5.27.

Závěr

Při řešení problému (5.48) se nabízí možnost použití explicitní Taylorovy řady, jsme však nuceni použít víceslovní aritmetiku, která má v tomto případě rozhodující vliv na stabilitu a přesnost numerického výpočtu.

Jako vhodné se také jeví možnost použití implicitní Taylorovy metody s rekurentním výpočtem členů a Newtonovou metodou (ITMRN).

Tabulka 5.27: Absolutní chyba výpočtu ITMRN: $h = 0.05$, $ORD = 8$, $TOL = 10^{-10}$, $a = 10^4$

t	$ \text{Error}(y) $	$ \text{Error}(z) $	j
0.05	7.10543×10^{-15}	7.10543×10^{-15}	4
0.1	1.77636×10^{-15}	1.77636×10^{-15}	4
0.15	4.996×10^{-14}	5.00711×10^{-14}	4
0.2	9.18154×10^{-14}	9.17044×10^{-14}	4
0.25	8.71525×10^{-14}	8.71525×10^{-14}	5
0.3	8.23785×10^{-14}	8.23785×10^{-14}	5
\vdots	\vdots	\vdots	\vdots
0.6	1.77636×10^{-14}	1.77636×10^{-14}	4

5.6.5 Problém parazitních prvků v RLC obvodu

Řešme soustavu diferenciálních rovnic

$$\begin{aligned}
 w' &= -10^{12}y - 10^9w, & w(0) &= 0, \\
 x' &= y - z - x, & x(0) &= 0, \\
 y' &= 10^{12}w - 10^{12}x, & y(0) &= 1, \\
 z' &= x, & z(0) &= 1.
 \end{aligned} \tag{5.51}$$

Soustava diferenciálních rovnic (5.51) popisuje v čase $t = 0$ ustálený stav v RLC obvodu viz podkapitola 4.3.2 (počáteční stav - obvodem neprotéká žádný proud $w(0) = x(0) = 0$ a napětí na kondenzátorech je rovno vstupnímu napětí $y(0) = z(0) = 1$).

Analytické řešení z je v následujícím tvaru

$$\begin{aligned}
 z &= e^{-0.5t}(\cos(0.8657t) + 0.5781204548 \sin(0.8657t)) + \\
 &+ e^{-5 \cdot 10^8 t}(-0.9999987499 \cdot 10^{-24} \cos(10^{12}t) - 0.1499999311 \cdot 10^{-26} \sin(10^{12}t)).
 \end{aligned} \tag{5.52}$$

Odvození analytického řešení funkce z společně s rozborem rekurentního výpočtu členů Taylorovy řady nalezneme v příloze A.

Zhodnoťme nejprve možnosti výpočtu SDR (5.51) explicitní Taylorovou metodou.

Explicitní Taylorova metoda

Při výpočtu SDR (5.51) pomocí explicitní Taylorovy řady jsme nuceni použít vysoký počet členů a velmi malý integrační krok z důvodu zachování stability. Připomeňme, že koeficient tuhosti (4.28) u daného lineárního systému diferenciálních rovnic (5.51) je $S = 10^9$ a vlastní čísla jsou v komplexním tvaru viz podkapitola 4.3.2.

V Tab. 5.28, 5.29 jsou zobrazeny absolutní chyby výpočtu explicitní Taylorovy metody při využití maximální velikosti možného integračního kroku. Při použití integračního kroku $h > 10^{-12}$ je již metoda nestabilní.

Zaměřme se nyní na možnost výpočtu SDR (5.51) pomocí implicitní Taylorovy metody.

Tabulka 5.28: Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 10^{-12}$

	Error(z)	
t	$ORD = 5$	$ORD = 10$
10^{-7}	1.01309×10^{237}	5.10703×10^{-15}

Tabulka 5.29: Absolutní chyba výpočtu - explicitní Taylorova metoda: $h = 10^{-11}$

	Error(z)		
t	$ORD = 5$	$ORD = 15$	$ORD = 25$
10^{-10}	19203.4	105.042×10^{-15}	NULA
10^{-9}	3.97833×10^{264}	4.45933×10^{237}	NULA
10^{-8}	MIMO ROZSAH	MIMO ROZSAH	NULA
10^{-7}	MIMO ROZSAH	MIMO ROZSAH	5.10703×10^{-15}

Implicitní Taylorova metoda

Podrobné úpravy a odvození výpočtu implicitní Taylorovy řady až do řádu 3 ($ORD = 3$) nalezneme v příloze A.2.2.

Tabulka 5.30: Absolutní chyba výpočtu - impl. Taylorova metoda: $h = 10^{-5}$, $TMAX = 0.1$

	Error(z)	
t	$ORD = 1$	$ORD = 2$
10^{-5}	4.99993×10^{-11}	1.11022×10^{-16}
10^{-4}	4.99948×10^{-10}	1.44329×10^{-15}
10^{-3}	4.99497×10^{-9}	1.43219×10^{-14}
10^{-2}	4.94993×10^{-8}	1.65423×10^{-13}
10^{-1}	4.50049×10^{-7}	1.78334×10^{-11}

Tabulka 5.31: Absolutní chyba výpočtu - impl. Taylorova metoda: $h = 10^{-4}$, $TMAX = 0.1$

	Error(z)		
t	$ORD = 1$	$ORD = 2$	$ORD = 3$
10^{-4}	4.99917×10^{-9}	1.67644×10^{-13}	6.66134×10^{-16}
10^{-3}	4.99466×10^{-8}	1.6761×10^{-12}	6.43929×10^{-15}
10^{-2}	4.94962×10^{-7}	1.67549×10^{-11}	7.9603×10^{-14}
10^{-1}	4.50003×10^{-6}	1.68273×10^{-10}	2.06357×10^{-12}

V Tab. 5.30, 5.31 jsou zobrazeny absolutní chyby výpočtu pro různý řád implicitní

Taylorovy metody a různou délkou integračního kroku. Díky větší oblasti stability implicitní Taylorovy metody můžeme použít mnohem větší integrační krok oproti explicitní Taylorově metodě.

Zhodnoťme nyní možnosti řešení pomocí implicitní Taylorovy metody s rekurentním výpočtem členů a Newtonovou metodou.

Kombinace implicitní Taylorovy metody s rekurentním výpočtem a Newtonovou metodou

Ověřme možnosti výpočtu SDR (5.51) pomocí kombinace implicitní Taylorovy metody s rekurentním výpočtem a Newtonovou metodou (ITMRN). Podrobný rozbor rekurentního odvození a výpočtu členů ITMRN nalezneme v příloze A.2.3.

Tabulka 5.32: Absolutní chyba výpočtu ITMRN: $h = 10^{-5}$, $TMAX = 0.1$, $h_N = 0.1$, $TOL = 10^{-10}$

	Error(z)	
t	$ORD = 1$	$ORD = 2$
10^{-5}	4.99993×10^{-11}	3.33067×10^{-16}
10^{-4}	4.99948×10^{-10}	2.66454×10^{-15}
10^{-3}	4.99497×10^{-9}	2.45359×10^{-14}
10^{-2}	4.94993×10^{-8}	2.46914×10^{-13}
10^{-1}	4.50034×10^{-7}	2.41951×10^{-12}
$\sum(j)$	20000	20000

Tabulka 5.33: Absolutní chyba výpočtu ITMRN: $h = 10^{-4}$, $TMAX = 0.1$, $h_N = 0.1$, $TOL = 10^{-10}$

	Error(z)		
t	$ORD = 1$	$ORD = 2$	$ORD = 3$
10^{-4}	4.99917×10^{-9}	1.67644×10^{-13}	8.88178×10^{-16}
10^{-3}	4.99466×10^{-8}	1.6761×10^{-12}	7.99361×10^{-15}
10^{-2}	4.94962×10^{-7}	1.67615×10^{-11}	8.00471×10^{-14}
10^{-1}	4.50003×10^{-6}	1.66753×10^{-10}	7.59726×10^{-13}
$\sum(j)$	2000	2000	2000

V Tab. 5.32, 5.33 je zobrazena absolutní chyba výpočtu pro různý řád ITMRN a různou délkou integračního kroku. Počet iterací v Newtonově metodě byl ve všech případech v každém kroku konstantní ($j = 2$). V posledním řádku Tab. 5.32, 5.33 nalezneme celkový počet Newtonových iterací provedených během výpočtu (velikost kroku $h_N = 0.1$ v diferenciální formuli nemá zásadní vliv na celkový počet iterací, nastavitelná chyba výpočtu byla zvolena na hodnotu $TOL = 10^{-10}$).

V Tab. 5.34 je zobrazena absolutní chyba výpočtu pro různý řád ITMRN a délku integračního kroku $h = 0.01$. Počet Newtonových iterací se s rostoucí velikostí integračního kroku v jednotlivých krocích zvyšuje (pohybuje se v rozmezí $2 \leq j \leq 10$). Metoda je však stále stabilní (důvodem zvyšování Newtonových iterací v průběhu výpočtu je nedostatečná velikost aritmetiky).

Tabulka 5.34: Absolutní chyba výpočtu ITMRN s víceslovní aritmetikou: $h = 10^{-2}$, $TMAX = 1$, $h_N = 0.1$, $TOL = 10^{-10}$

t	Error(z)			
	ORD = 2		ORD = 3	
	8 bytů	64 bytů	8 bytů	64 bytů
10^{-2}	1.66802×10^{-7}	1.66801×10^{-7}	8.33156×10^{-12}	8.33788×10^{-12}
10^{-1}	1.6587×10^{-6}	1.65869×10^{-6}	4.34068×10^{-10}	4.34318×10^{-10}
1	1.09229×10^{-5}	1.09228×10^{-5}	2.22892×10^{-8}	2.229798×10^{-8}
$\sum(j)$	210	200	701	301

Hodnocení vlivu aritmetiky

Experimentálně bylo ověřeno, že je výhodné v tomto případě využít víceslovní aritmetiku. Při použití víceslovní aritmetiky Newtonova metoda konverguje rychleji a počet iterací se v jednotlivých krocích sníží ($2 \leq j \leq 3$). V posledním řádku Tab. 5.34 je zobrazen celkový počet iterací provedených v průběhu výpočtu.

Závěr

Pro řešení problému (5.51), tedy lineární soustavy obyčejných diferenciálních rovnic s vlastními čísly v komplexním tvaru, se jako vhodné jeví použít implicitní Taylorovu metodu s rekurentním výpočtem členů a Newtonovou metodou (ITMRN). Dále se ukázalo a bylo ověřeno jako vhodné využití víceslovní aritmetiky, díky níž zásadně snížíme celkový počet Newtonových iterací v průběhu výpočtu.

5.6.6 Problém Van Der Pol

Mějme soustavu ODR (viz podkapitola 4.3.3) ve tvaru

$$\begin{aligned} x' &= y, & x(0) &= 2, \\ y' &= \mu \cdot (1 - x^2) \cdot y - x, & y(0) &= 0. \end{aligned} \quad (5.53)$$

S rostoucí hodnotou μ se zvyšuje "tuhost" soustavy diferenciálních rovnic.

Zhodnotme možnosti řešení SDR (5.53) pomocí explicitní a implicitní Taylorovy metody.

Explicitní Taylorova metoda

Členy explicitní Taylorovy řady počítáme rekurentně podobně jako v předcházejících příkladech. Požadovanou vyšší derivaci součinu ($x^2 y$) lze vyjádřit rekurentním výpočtem pomocí předcházejících členů s využitím multidimenzionálních Pascalových trojúhelníků

[Eat90, Vop11]. Výpočet vychází ze vzorce

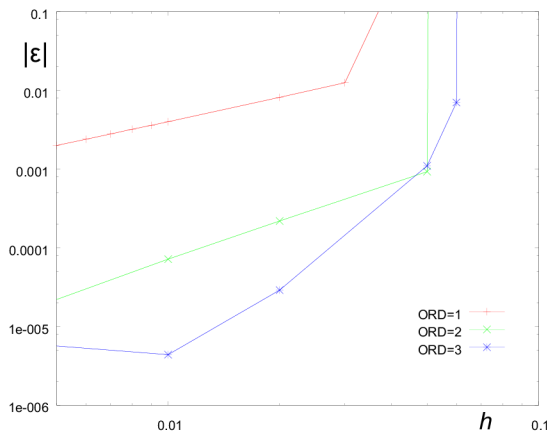
$$(y_1 \cdot y_2 \cdots y_m)^{(n)} = \sum_{i_1=0}^n \left(\sum_{i_2=0}^{i_1} \cdots \left(\sum_{i_{m-1}=0}^{i_{m-2}} \binom{n}{i_1} \binom{i_1}{i_2} \cdots \binom{i_{m-2}}{i_{m-1}} y_1^{(n-i_1)} y_2^{(i_1-i_2)} \cdots y_m^{(i_{m-1})} \right) \right). \quad (5.54)$$

Řešme nejprve soustavu ODR (5.53) pro $\mu = 10$ pomocí explicitní Taylorovy metody na intervalu $TMAX = 10$. Jako referenční hodnoty považujeme

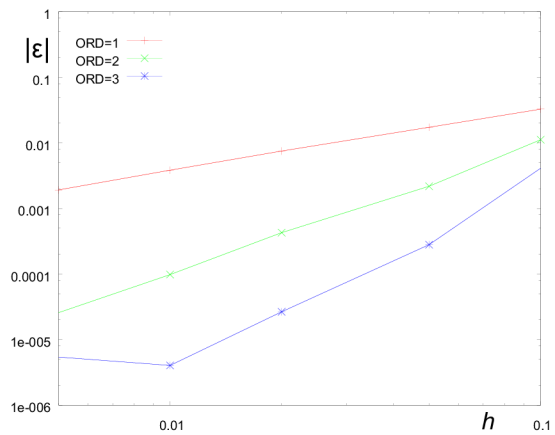
$$x(10) = -1.97120695682917, \quad y(10) = 0.0681732453104,$$

vypočteny pomocí TKSL/386 při zadané chybě výpočtu $EPS = 10^{-20}$.

V Obr. 5.23 je zobrazena absolutní chyba výpočtu $|\text{Error}(y)| = |y(10) - y_i|$, $i \cdot h = 10$ v čase $TMAX = 10$ v závislosti na délce integračního kroku. Se zvyšováním ORD (nebo snižováním h) samozřejmě klesá absolutní chyba.



Obrázek 5.23: Absolutní chyba výpočtu explicitní Taylorovy metody v $TMAX = 10$



Obrázek 5.24: Absolutní chyba výpočtu ITMRN v $TMAX = 10$

Maximální velikost integračního kroku, při kterém je ještě explicitní Taylorova řada stabilní, je $h = 0.04$ pro $ORD = 1$, resp. $h = 0.06$ pro $ORD = 2, 3$. Při větším integračním kroku je již metoda nestabilní a absolutní chyba prudce roste k nekonečnu.

U explicitních numerických metod nastává problém s výpočtem (5.53) při rostoucí hodnotě μ . V Tab. 5.35 je zobrazena maximální velikost integračního kroku pro explicitní metody v závislosti na velikosti konstanty μ . S rostoucí hodnotou μ roste tuhost systému (5.53) a jsme nuceni u explicitních numerických metod snižovat integrační krok.

Tabulka 5.35: Maximální velikost h , explicitní numerické metody, $EPS = 1 \times 10^{-5}$

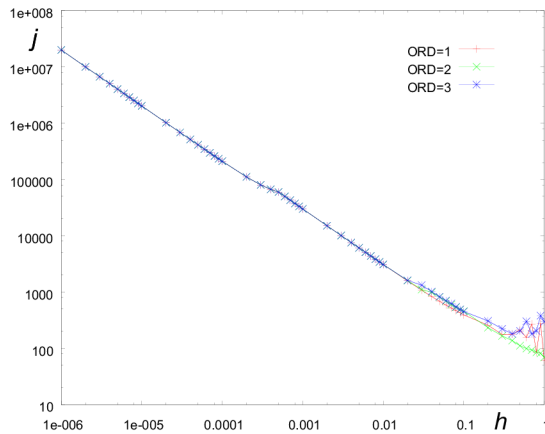
metoda	$h (\mu = 10)$	$h (\mu = 100)$	$h (\mu = 10000)$
Eulerova metoda	6.10351×10^{-7}	3.81469×10^{-8}	1.9209×10^{-9}
metoda Runge-Kutta 2.řádu	1.5625×10^{-4}	1.953125×10^{-5}	1.52587×10^{-7}
metoda Runge-Kutta 4.řádu	5×10^{-3}	3.125×10^{-4}	4.88281×10^{-6}
Taylorova metoda ($ORD = 63$)	0.1	2.5×10^{-2}	3.125×10^{-4}

Zhodnotme možnosti řešení SDR (5.53) pomocí implicitní Taylorovy metody s rekurentním výpočtem členů a Newtonovou metodou (ITMRN).

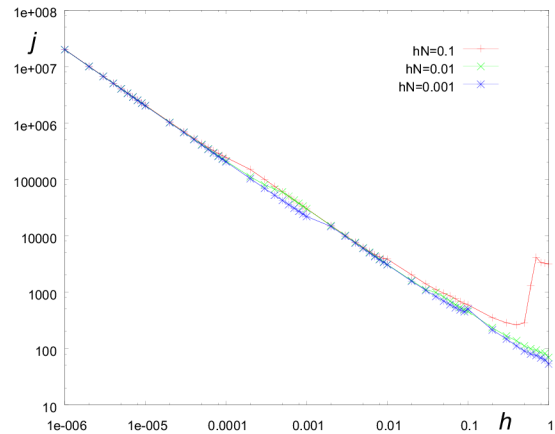
Kombinace implicitní Taylorovy metody s rekurentním výpočtem a Newtonovou metodou (ITMRN)

Členy ITMRN počítáme rekurentně podobně jako v předcházejících příkladech. K výpočtu implicitně vyjádřených členů použijeme opět Newtonovu metodu.

V Obr. 5.24 je zobrazena závislost absolutní chyby výpočtu $|\text{Error}(y)| = |y(10) - y_i|$, (kde $i \cdot h = 10$) v čase $TMAX = 10$ v závislosti na velikosti integračního kroku h . Větší stabilita implicitní Taylorovy metody umožňuje použití většího integračního kroku - např. pro explicitní Taylorovu metodu $ORD = 1$ můžeme zvolit maximální integrační krok $h = 0.03$ při absolutní chybě výpočtu $|\text{Error}(y)| = 0.0125491$, při výpočtu pomocí implicitní Taylorovy metody můžeme počítat s integračním krokem $h = 0.5$ při absolutní chybě výpočtu $|\text{Error}(y)| = 0.0961105$.



Obrázek 5.25: ITMRN - počet Newt. iterací pro $ORD = 1, 2, 3$



Obrázek 5.26: ITMRN - počet Newt. iterací v závislosti na h_N

V Obr. 5.25 je zobrazen celkový počet Newtonových iterací j potřebný k výpočtu řešení (5.53) pro $\mu = 10$ a $TMAX = 10$ s daným integračním krokem h . Rozdílný počet Newtonových iterací u různých řádů implicitní Taylorovy metody se projeví až při větší hodnotě integračního kroku.

Zajímavý je však v tomto případě celkový počet Newtonových iterací j v závislosti na velikosti kroku diferencní formule h_N viz Obr. 5.26. Optimální hodnota kroku diferencní formule v daném případě je $h_N = 0.001$, menší hodnoty se již na zmenšení celkového počtu iterací výrazně neprojeví. Při všech hodnotách h_N byl obdržen stejný výsledek. Newtonova metoda byla ukončena při dosažení nastavené přesnosti výpočtu $TOL = 10^{-10}$.

V Tab. 5.36 je zobrazena absolutní chyba výpočtu v prvním kroku pro ITMRN různého řádu. Všimněme si klesající absolutní chyby výpočtu s rostoucí hodnotou μ . Potvrzuje se opět, že ITMRN je vhodná pro “velmi tuhé” systémy diferenciálních rovnic.

S rostoucím řádem ITMRN musíme snižovat velikost kroku v diferencní formuli h_N . Optimální velikost kroku h_N je zobrazena v posledním řádku Tab. 5.36. Počet Newtonových iterací se pohybuje v průběhu výpočtu v rozmezí $4 \leq j \leq 7$.

Závěr

Při řešení nelineární soustavy ODR (5.53) se projevují pozitivní vlastnosti implicitní Taylorovy metody až při větších hodnotách konstanty μ . Opět se potvrzuje vhodnost použití implicitní Taylorovy metody pro systémy s vysokou tuhostí.

Tabulka 5.36: Absolutní chyba výpočtu v prvním kroku ITMRN: $h = 1$, $TOL = 10^{-10}$

	Error(y)			
μ	$ORD = 1$	$ORD = 2$	$ORD = 3$	$ORD = 4$
10	2.20562×10^{-3}	1.85222×10^{-4}	2.02124×10^{-5}	6.40326×10^{-6}
100	2.2221×10^{-5}	1.32989×10^{-7}	1.64021×10^{-8}	1.78864×10^{-9}
1000	2.22242×10^{-7}	1.48484×10^{-10}	2.14972×10^{-13}	6.68134×10^{-14}
h_N	10^{-1}	10^{-3}	10^{-6}	10^{-9}

Při implementaci ITMRN s numerickým výpočtem derivace v Newtonově metodě se při řešení problému (5.53) objevuje zásadní závislost na vhodné volbě kroku h_N v derivační formuli. S rostoucím řádem ITMRN jsme nuceni pro optimální počet Newtonových iterací v průběhu výpočtu snižovat hodnotu h_N .

5.7 Shrnutí

Moderní metody Taylorovy řady poskytují velice přesné řešení obyčejných diferenciálních rovnic. Hlavním problémem u metod Taylorovy řady je generování vyšších derivací pomocí tvořících diferenciálních rovnic. V současné době lze však úspěšně využít vhodných paralelních architektur. Především u rozsáhlých soustav diferenciálních rovnic lze vhodným navržením rekurentního výpočtu členů Taylorovy řady výpočet značně urychlit.

Pro řešení tuhých systémů, obzvláště systémů, kde hledáme ustálený stav řešení na delším časovém intervalu, se jeví jako vhodné použití implicitní Taylorovy metody s rekurentním výpočtem členů a Newtonovou metodou (ITMRN). V mnoha případech se ITMRN experimentálně osvědčila jako vhodná pro řešení SDR s “vysokou tuhostí”.

U mnoha tuhých problémů je nezbytné použít víceslovní aritmetiku pro dosažení dostatečné přesnosti, rychlosti a stability výpočtu.

Kapitola 6

Závěr

Předložená disertační práce se zabývá analýzou tuhých (“stiff”) soustav diferenciálních rovnic. V průběhu zpracovávání zadaného tématu disertační práce se ukázalo, že je vhodné pro komplexní řešení problému rozšířit zadané téma také o možnosti řešení stiff soustav diferenciálních rovnic.

Obsah práce bude nyní shrnut do několika kroků. Nejprve uvedu použitý přístup k řešení, dále dosažené výsledky a konečně naznačím možnosti dalšího výzkumu.

6.1 Zvolený přístup k řešení

Řešení tuhých soustav diferenciálních rovnic patří i v současné době stále mezi problémové. V praxi se k řešení běžně využívají implicitní numerické metody, které mají velkou oblast absolutní stability řešení. Díky tomu je možno s nimi provádět numerický výpočet u tuhých SDR s použitím většího integračního kroku oproti běžně používaným explicitním numerickým metodám. Velká oblast absolutní stability u těchto implicitních numerických metod je však vyvážena malým řádem těchto metod.

Ve své práci jsem se nejprve věnoval shromáždění dostupných informací o analýze a přístupech k řešení soustav diferenciálních rovnic (SDR). Zvláštní pozornost jsem věnoval především numerickému řešení SDR.

Na vybraných případech jsem kriticky zhodnotil použití a vlastnosti jednotlivých numerických metod se zaměřením na stabilitu, konvergenci a přesnost numerického výpočtu.

Dále jsem experimentálně ověřoval možnosti řešení tuhých systémů pomocí v literatuře méně zmiňované Taylorovy řady.

6.2 Dosažené výsledky

Ve své práci jsem se věnoval analýze tuhých systémů a jejich možnému vhodnému způsobu řešení především pomocí Taylorovy metody.

Problémem při řešení tuhých SDR je i nadále jejich detekce. V práci navrhuji novou možnost detekce “tuhosti” v SDR, která vychází přímo z porovnání členů explicitní Taylorovy řady (není zapotřebí počítat Jacobiovu matici ani vlastní čísla řešené soustavy). Analyzoval jsem tedy možnost **detekce tuhých SDR z rozdílných členů explicitní Taylorovy řady** v závislosti na jejich konvergenci a délce integračního kroku.

Dále byly v práci hledány a navrženy možné přístupy k odstranění tuhosti ze soustav diferenciálních rovnic. Mnoho SDR obsahuje tzv. “uměle vnášenou tuhost”. Pro tyto SDR

jsem navrhl a experimentálně ověřil nový možný přístup **odstranění tuhosti ze SDR pomocí nalezení nové ekvivalentní SDR se sníženou (nebo zcela odstraněnou) tuhostí**. Tuto novou ekvivalentní SDR lze poté bez problémů řešit běžnými explicitními numerickými metodami.

V práci byla provedena podrobná **analýza konvergence a stability explicitní a nově také implicitní Taylorovy řady**. Na vybraných demonstračních příkladech jsem se zaměřil především na možnosti přesnosti numerického výpočtu obou metod.

Jako zajímavá se jeví při řešení tuhých systémů implicitní Taylorova řada, která poskytuje vyšší řád, tedy přesnější řešení proti tradičním implicitním numerickým metodám. Na vybraných příkladech bylo experimentálně demonstrováno **řešení tuhých systémů pomocí implicitní Taylorovy řady s Newtonovou iterací a rekurentním výpočtem členů Taylorovy řady (ITMRN)**. Byl navržen algoritmus (především rekurentní výpočet členů Taylorovy řady) pro možnou budoucí paralelizaci výpočtu na vhodné hardwarové architektuře. ITMRN se z prováděných experimentů jeví jako vhodná numerická metoda především pro řešení soustav diferenciálních rovnic s vysokou tuhostí. U problémů s vysokou tuhostí se jeví jako stěžejní (k přesnému a rychlému výpočtu) rovněž **použití víceslovní aritmetiky**.

6.3 Možnosti dalšího výzkumu

V druhé části práce byla demonstrována na konkrétních příkladech méně známá metoda implicitní Taylorovy řady. Jednalo se o základní výzkum na základě mnoha experimentů. Možností dalšího výzkumu je podrobné vymezení skupiny problémů, na které je implicitní Taylorova řada “použitelná” (v technické praxi se můžeme setkat s nestandardními problémy jako jsou “detekce nespojitosti” apod.).

Zajímavé se také může jevit prozkoumání semi-implicitní Taylorovy řady, (tedy kombinace explicitní a implicitní numerické metody) využívající rekurentního výpočtu vyšších derivací a zajišťující vlastnost A-stabilní numerické metody.

Literatura

- [Arn93] M. Arnold. Stability of numerical methods for differential-algebraic equations of higher index. *Applied Numerical Mathematics*, 13:5–14, 1993.
- [Bar80] D. Barton. On Taylor Series and Stiff Equations. *ACM Transactions on Mathematical Software*, 6(3), 1980.
- [Bar05] R. Barrio. Performance of the Taylor series method for ODEs/DAEs. *Applied Mathematics and Computation*, 163:525–545, 2005. ISSN 00963003.
- [Bar06] Hans-Jochen Bartsch. *Matematické vzorce*. 4. vydání, Academia, 2006. ISBN 80-200-1448-9.
- [BBL03] R. Barrio, F. Blesa, and M. Lara. High-precision numerical solution of ODE with high-order Taylor methods in parallel. *Monografías de la Real Academia de Ciencias de Zaragoza*, 22:67–74, 2003.
- [BBL05] R. Barrio, F. Blesa, and M. Lara. VSVO Formulation of the Taylor Method for the Numerical Solution of ODEs. *Computers and Mathematics with Applications*, 50:93–111, 2005.
- [BCP96] K. E. Brenan, S. L. Campbell, and L. R. Petzold. *Numerical Solution of Initial-Value Problems Differential-Algebraic Equations*, volume 14. SIAM, Philadelphia, 1996.
- [BF04] Richard L. Burden and J. Douglas Faires. *Numerical Analysis*. 8th edition Brooks-Cole Publishing, 2004. ISBN 0-534-39200-8.
- [BKS88] Laurence A. Bales, Ohannes A. Kakakashian, and Steven M. Serbin. On the Ao-acceptability of rational approximations to the exponential function with only real poles. *BIT*, 28:70–79, January 1988.
- [But00] J. C. Butcher. Numerical methods for ordinary differential equations in the 20th century. *Journal of Computational and Applied Mathematics*, 125:1–29, 2000.
- [But08a] J. Butcher. Forty-five years of A-stability. presentation, ICNAAM Conference, Kos, Greece, 2008.
- [But08b] J. C. Butcher. *Numerical Methods for Ordinary Differential Equations*. Second Edition, John Wiley & Sons Ltd., 2008. ISBN 978-0-470-72335-7.

- [BWZ71] D. Barton, I. M. Willers, and R.V.M. Zahar. The Automatic Solution of Ordinary Differential Equations by the Method of Taylor Series. *The Computer Journal*, 14(3):243–248, 1971.
- [CC82] G.F. Corliss and Y. F. Chang. Solving Ordinary Differential Equations Using Taylor Series. *ACM Transactions on Mathematical Software*, 8(2):243–248, 1982.
- [CC94] Y. F. Chang and G.F. Corliss. ATOMFT: solving ODEs and DAEs using Taylor series. *Computers and Mathematics with Applications*, 28:209–233, 1994.
- [CGH⁺97] G. F. Corliss, A. Griewank, P. Henneberger, G. Kirlinger, F. A. Potra, and H. J. Stetter. High-order stiff ODE solvers via automatic differentiation and rational prediction. *Numerical Analysis and its Applications, Lecture Notes in Computer Science, Springer*, 1196:114–225, 1997.
- [CH52] C. F. Curtis and J. O. Hirschfelder. Integration of stiff equations. *Proc. Nat. Acad. Sci.*, 38(3):235–243, 1952.
- [Chi06] N. G. Chikurov. Stability and Accuracy of Implicit Methods for Stiff Systems of Linear Differential Equations. *Differential Equations*, 42(12):1057–1067, July 2006.
- [CS79] G. J. Cooper and A. Sayfy. Semiexplicit A-Stable Runge-Kutta Methods. *Mathematics of Computation*, 33(146):541–556, 1979.
- [Cur83] A. R. Curtis. Jacobian Matrix Properties and Their Impact on Choice of Software for Stiff ODE Systems. *IMA Journal of Numerical Analysis*, 3:397–415, 1983.
- [CWH06] X. Chang, Y. Wang, and L. Hu. An implicit Taylor series numerical calculation method for power system transient simulation. In *MIC'06 Proceedings of the 25th IASTED international conference*, pages 82–85, 2006.
- [CZG10] X. Chang, H. Zheng, and X. Gu. An A-stable improved Taylor series method for power system dynamic-stability simulation. In *Proceedings of the Power and Energy Engineering Conference Asia-Pacific*, pages 1–5, 2010.
- [Dah56] G. Dahlquist. Convergence and stability in the numerical integration of ordinary differential equations. *Math. Scand.*, 4:33–53, 1956.
- [Dal97] J. Dalík. *Numerické metody*. Akademické nakladatelství CERM, s.r.o. Brno, 1997. ISBN 80-214-0646-1.
- [Eat90] Edward A. Eaton. A multidimensional extension of Pascal's triangle. *SIGSMALL/PC Notes*, 16:34–37, May 1990.
- [Ehl68] B. L. Ehle. High order A-stable methods for the numerical solution of systems of DEs. *BIT Numerical Mathematics*, 8(4):276–278, 1968.

- [Ehl69] B. L. Ehle. *On Padé approximations to the exponential function and A-stable methods for the numerical solution of initial value problems*. PhD thesis, Department of Applied Analysis and Computer Science, Univeristy of Waterloo, Ontario, Canada, 1969.
- [Ehl73] B. L. Ehle. A-stable Methods and Padé Approximations to the Exponential. *SIAM Journal on Mathematical Analysis*, 4:671–680, 1973.
- [EHL75] W. H. Enright, T. E. Hull, and B. Lindberg. Comparing numerical methods for stiff systems of O.D.E.s. *BIT Numerical Mathematics*, 15(1):10–48, 1975.
- [Gib60] A. Gibbons. A Program for the Automatic Integration of Differential Equations using the Method of Taylor Series. *The Computer Journal*, 3:108–111, 1960.
- [GMP10] *The GNU Multiple Precision Arithmetic Library*, [cit. 10-8-2010]. URL: <http://gmpilib.org/> [online].
- [GP84] C. W. Gear and L. R. Petzold. ODE methods for the solution of differential/algebraic systems. *SIAM Journal Numerical Analysis*, 21(4):716–728, 1984.
- [Hal83] H. J. Halin. The applicability of Taylor series methods in simulation. In *Proceedings of the 1983 Summer Computer Simulation Conference*, pages 1032–1070, 1983.
- [HNW87] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*. vol. Nonstiff Problems. Springer-Verlag Berlin Heidelberg, 1987. ISBN 3-540-56670-8.
- [HW00] E. Hairer and G. Wanner. *Analysis by Its History*. corrected third printing. Springer-Verlag New York Berlin Heidelberg, 2000. ISBN 0-387-94551-2.
- [HW02] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. second revised ed. with 137 Figures, vol. Stiff and Differential-Algebraic Problems. Springer-Verlag Berlin Heidelberg, 2002. ISBN 3-540-60452-9.
- [Jel76] R. Jeltsch. Stiff stability and its relation to Ao- and A(0)-stability. *SIAM Journal Numerical Analysis*, 13(1):8–17, 1976.
- [JN02] K. R. Jackson and N. S. Nedialkov. Some recent advances in validated methods for IVPs for ODEs. *ELSEVIER, Applied Numerical Mathematics*, 42:269–284, 2002.
- [JZ05] A. Jorba and M. Zou. A software package for the numerical integration of ODE by means of high-order Taylor methods. *Exp. Math.*, 14(1):99–117, 2005.
- [KC91] G. Kirlinger and G. F. Corliss. On Implicit Taylor Series Methods for Stiff ODEs. In *The Proceedings of SCAN 91: Symposium on Computer Arithmetic and Scientific Computing*, pages 371–379, 1991.
- [KDJ05] M. Kubíček, M. Dubcová, and D. Janovská. *Numerické metody a algoritmy*. VŠCHT Praha, 2005. ISBN 80-7080-558-7.

- [Kun94] J. Kunovský. *Modern Taylor Series Method*. FEI-VUT Brno, 1994. Habilitation work.
- [LEP99] M. Lara, A. Elipe, and M. Palacios. Automatic programming of recurrent power series. *Mathematics and computers in Simulation*, 49:351–362, 1999.
- [LeV07] Randall J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems*, volume 357. SIAM, Philadelphia, 2007. ISBN 978-0-898716-29-0.
- [Map10] Maplesoft. *Math Software for Engineers, Educators and Students*, [cit. 10-6-2010]. URL: <http://www.maplesoft.com/> [online].
- [Mat10a] The MathWorks. *MATLAB and Simulink for Technical Computing*, [cit. 10-6-2010]. URL: <http://www.mathworks.com> [online].
- [Mat10b] Wolfram MathWorld. *The Web's Most Extensive Mathematics Resource*, [cit. 10-6-2010]. URL: <http://mathworld.wolfram.com> [online].
- [Mik00] K. Mikulášek. *Polynomial Transformations of Systems of Differential Equations and Their Applications*. PhD thesis, FEI VUT v Brně, 2000.
- [MKC09] R. E. Moore, R. B. Kearfott, and M. J. Cloud. *Introduction to Interval Analysis*. SIAM, Philadelphia, 2009. ISBN 978-0-898716-69-6.
- [MM03a] E. Miletics and G. Molnárka. Taylor Series Method with Numerical Derivatives for Numerical solution of ODE Initial Value Problems. *HEJ, Section Applied and Numerical Mathematics*, pages 1–16, 2003.
- [MM03b] G. Molnárka and E. Miletics. Implicit Extension of Taylor Series Method with Numerical Derivatives. *PAMM*, 3(1):569–570, 2003.
- [MUK05] Kingsley Obiajulu MUKA. *One Step Methods For Stiff Systems*, 2005. URL: <http://kingklip.aims.ac.za/resources/archive/2004/kingsley.ps> [online].
- [NJC99] N. S. Nedialkov, K. R. Jackson, and G. F. Corlis. Validated solutions of initial value problems for ordinary differential equations. *ELSEVIER, Applied Mathematics and Computation*, 105:21–68, 1999.
- [NJP02] N. S. Nedialkov, K. R. Jackson, and J. D. Pryce. An Effective High-Order Interval Method for Validating Existence and Uniqueness of the Solution of an IVP for an ODE. *Reliable Computing, Applied Numerical Mathematics*, 7(6):449–465, 2002.
- [NP05] N. S. Nedialkov and J. D. Pryce. Solving differential algebraic equations by Taylor series (I): Computing Taylor coefficients. *BIT*, 45(3):561–591, 2005.
- [NP07] N. S. Nedialkov and J. D. Pryce. Solving differential-algebraic equations by Taylor series (II): Computing the system Jacobian. *BIT*, 47(1):121–135, 2007.
- [Pet83] L. R. Petzold. Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *SIAM J. Sci. Stat. Comput.*, 4(1):136–148, 1983.

- [PR74] A. Prothero and A. Robinson. On the Stability and Accuracy of One-Step Methods for Solving Stiff Systems of Ordinary Differential Equations. *Mathematics of Computation*, 28(125):145–162, 1974.
- [PTVF07] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical Recipes 3rd Edition: The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA, 3 edition, 2007.
- [R.A66] R.A.Moore. *Interval Analysis*. Prentice-Hall, New York, 1966.
- [Ral81] Louis B. Rall. *Automatic Differentiation: Techniques and Applications*, volume 120 of *Lecture Notes in Computer Science*. Springer, Berlin, 1981. ISBN 0-540-10861-0.
- [Sco00] James R. Scott. Solving ODE Initial Value Problems With Implicit Taylor Series Methods. Technical Report NASA/TM-2000-209400, National Aeronautics and Space Administration Washington, 2000.
- [Seh07] P. Sehnalová. *Konvergence soustav algebraických rovnic*. Diplomová práce, FIT VUT v Brně, 2007.
- [Sha94] L. F. Shampine. *Numerical Solution of Ordinary Differential Equations*. Chapman & Hall, One Penn Plaza, New York, NY 10119, 1994. ISBN 0-412-05151-6.
- [sof10] TKSL software. *High Performance Computing*, [cit. 10-6-2010]. URL: <http://www.fit.vutbr.cz/~kunovsky/TKSL/index.html.en> [online].
- [sti10] *Stiff systems*, [cit. 10-6-2010]. URL: <http://www.biomatematica.it/urbino2002/programmi/Italy1.pdf> [online].
- [TB01] T. Tian and K. Burrage. Implicit Taylor methods for stiff stochastic differential equations. *Applied Numerical Mathematics*, 38:167–185, 2001.
- [Č04] L. Čermák. *Numerické metody II*. Akademické nakladatelství CERM, s.r.o. Brno, 2004. ISBN 80-214-2722-1.
- [vH05] L. Čermák and R. Hlavička. *Numerické metody*. Akademické nakladatelství CERM, s.r.o. Brno, 2005. ISBN 80-214-3071-0.
- [Vit94] E. Vitásek. *Základy teorie numerických metod pro řešení diferenciálních rovnic*. Academia, Praha, 1994. ISBN 80-200-0281-2.
- [Vop11] V. Vopěnka. *Vývojové prostředí numerických integrátorů*. Diplomová práce, FIT VUT v Brně, 2011.
- [Ř04] D. Řezáč. *Stiff Systems of Differential Equations and Modern Taylor Series Method*. PhD thesis, FIT VUT v Brně, 2004.
- [Wil74] Ralph A. Willoughby. *Stiff Differential Systems*. Plenum Press New York and London, 1974. ISBN 0-306-30797-9.

Přehled publikací autora

- [KPS06] J. Kunovský and J. Petřek, V. Šátek. Fourier Series and Symbolic Computations. In: *Proceedings of 40th Spring International Conference Mosis'06*, Ostrava, CZ, MARQ, 2006, s. 85–90, ISBN 80-86840-21-2.
- [KKS07] M. Kraus, J. Kunovský, V. Šátek. Fourier Analysis and Modern Taylor Series Method. In: *Proceedings of the 7th International Conference on Applied Mathematics*, Bratislava, SK, 2007, s. 6, ISBN 978-80-969562-5-8.
- [KPSZ07] J. Kunovský, M. Pindryč, V. Šátek, F. V. Zbořil. Stiff systems in theory and practice. In: *Proceedings of the 6th EUROSIM Congress on Modelling and Simulation*, Vídeň, AT, 2007, s. 6, ISBN 978-3-901608-32-2.
- [KKPS08] M. Kraus, J. Kunovský, M. Pindryč, V. Šátek. Taylor Series in Control Theory. In *Proceedings UKSim 10th International Conference EUROSIM/UKSim2008*, Cambridge, GB, IEEE CS, 2008, s. 378–379, ISBN 0-7695-3114-8.
- [KPS08] J. Kunovský, J. Petřek, V. Šátek. Multiple Arithmetic in Dynamic System Simulation. In *Proceedings UKSim 10th International Conference EUROSIM/UKSim2008*, Cambridge, GB, IEEE CS, 2008, s. 597–598, ISBN 0-7695-3114-8.
- [KKS08] M. Kraus, J. Kunovský, V. Šátek. Taylor Series Numerical Integrator. In *Proceedings second UKSIM European Symposium on Computer Modeling and Simulation*. Liverpool, GB, IEEE CS, 2008, s. 177–180, ISBN 978-0-7695-3325-4.
- [KKSK08] J. Kunovský, M. Kraus, V. Šátek, V. Kaluža. Accuracy and Word Width in TKSL. In *Proceedings second UKSIM European Symposium on Computer Modeling and Simulation*. Liverpool, GB, IEEE CS, 2008, s. 153–158, ISBN 978-0-7695-3325-4.
- [KSKK08] J. Kunovský, V. Šátek, M. Kraus, J. Kopřiva. Semi-analytical Computations Based on TKSL. In *Proceedings second UKSIM European Symposium on Computer Modeling and Simulation*. Liverpool, GB, IEEE CS, 2008, s. 159–164, ISBN 978-0-7695-3325-4.
- [KSK08] J. Kunovský, V. Šátek, M. Kraus. 25th Anniversary of TKSL. In *Proceedings of 7th International Conference of Numerical Analysis and Applied Mathematics*. Psalidi, Kos, GR, AIP, 2008, s. 343–346, ISBN 978-0-7354-0576-9.

- [SKKP08] V. Šátek, J. Kunovský, M. Kraus, M. Pindryč. TKSL to World Standards Comparison. In *Proceedings of 42th Spring International Conference Mosis'08*. Ostrava, CZ, MARQ, 2008, s. 21–27, ISBN 978-80-86840-40-6.
- [KKKS09] V. Kaluža, M. Kraus, J. Kunovský, V. Šátek. Initial Problems with Polynomials on Right-hand Sides. In *Proceedings of Third Asia International Conference on Modelling and Simulation*. Bandung/Bali, ID, IEEE CS, 2009, s. 182–187, ISBN 978-0-7695-3648-4.
- [KKS09] M. Kraus, J. Kunovský, V. Šátek. Taylorian initial problem. In *Proceedings MATHMOD 09 Vienna - Full Papers CD Volume*. Vienna, AT, ARGESIM, 2009, s. 1181–1186, ISBN 978-3-901608-35-3.
- [KKSK09] J. Kunovský, M. Kraus, V. Šátek, V. Kaluža. Parallel Computations Based on Analogue Principles. In *Proceedings of Eleventh International Conference on Computer Modelling and Simulation EUROSIM/UKSim2009*. Cambridge, GB, IEEE CS, 2009, s. 111–116, ISBN 978-0-7695-3593-7.
- [KSKK09] J. Kunovský, V. Šátek, M. Kraus, J. Kopřiva. Automatic Method Order Settings. In *Proceedings of Eleventh International Conference on Computer Modelling and Simulation EUROSIM/UKSim2009*. Cambridge, GB, IEEE CS, 2009, s. 117–122, ISBN 978-0-7695-3593-7.
- [KSKP09] J. Kunovský, V. Šátek, M. Kraus, M. Pindryč. Comparison of TKSL to world standards. In *International Journal of Autonomic Computing*. roč. 1, č. 2, 2009, London, GB, s. 182–191, ISSN 1741-8569.
- [KKSK09b] J. Kunovský, M. Kraus, V. Šátek, V. Kaluža. New Trends in Taylor Series Based Computations. In *Proceedings of 7th International Conference of Numerical Analysis and Applied Mathematics*. Rethymno, Crete, GR, AIP, 2009, s. 282–285, ISBN 978-0-7354-0705-3.
- [KSS10] J. Kunovský, A. Szöllös, V. Šátek. TKSL and Wave Partial Differential Equation. In *Proceeding of the 11th International Scientific Conference Electric Power Engineering*. Faculty of Electrical Engineering and Communication BUT, Brno, CZ, 2010, s. 201–205, ISBN 978-80-214-4094-4.
- [KKS10] M. Kádák, J. Kunovský, V. Šátek. Semianalytic Computation in TKSL. In *Proceedings of the 16th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists, MODERN TECHNIQUE AND TECHNOLOGIES MTT 2010*. Institute of Electrical and Electronics Engineers, Tomsk, RU, 2010, s. 116–118, ISBN 0-7803-8226-9.
- [KSV10] J. Kunovský, V. Šátek, V. Valenta. Advanced Partial Differential Equations Solutions. In *Proceedings of the 16th International Scientific and Practical Conference of Students, Post-graduates and Young Scientists, MODERN TECHNIQUE AND TECHNOLOGIES MTT 2010*. Institute of Electrical and Electronics Engineers, Tomsk, RU, 2010, s. 114–116, ISBN 0-7803-8226-9.
- [KSS10b] J. Kunovský, P. Sehnalová, V. Šátek. Stability and Convergence of the Modern Taylor Series Method. In *Proceedings of the 7th EUROSIM Congress*

on Modelling and Simulation. Czech Technical University Publishing House, Praha, CZ, 2010, s. 6, ISBN 978-80-01-04589-3.

- [KSS10c] J. Kunovský, P. Sehnalová, V. Šátek. Explicit and Implicit Taylor Series Based Computations. In *8th International Conference of Numerical Analysis and Applied Mathematics*. American Institute of Physics, Tripolis, GR, 2010, s. 587–590, ISBN 978-0-7354-0831-9.
- [KKS10b] M. Kadák, J. Kunovský, V. Šátek. Stiffness and Algebraic Manipulations in Electronic Circuits Simulations. In *Proceedings of The International Congress on Computer Applications and Computational Science*. Singapore, MY, 2010, s. 4, ISBN 978-981-08-6846-8.
- [KSS11a] J. Kunovský, P. Sehnalová, V. Šátek. Inovace výuky teorie spojitých systémů. In *Sborník Informatika 2011*. Brno, CZ, Mendelu, 2011, s. 47-48, ISBN 978-80-7375-520-1.
- [KSS11b] J. Kunovský, V. Šátek., A. Szöllös Telegraph Equation and Corresponding Wave Forms. In *Proceeding of the 12th International Scientific Conference Electric Power Engineering*. Ostrava, CZ, 2011, s. 4, ISBN 978-80-248-2393-5.
- [SKK11] V. Šátek., J. Kunovský, J. Kopřiva Advanced Stiff Systems Detection. In *Proceeding of the 11th International Scientific Conference on Informatics*. Rožňava, SK, 2011, s. 5, *bude vydáno 16. 11. 2011* .

Příloha A

RLC parazitní

Řešme soustavu diferenciálních rovnic ve tvaru

$$\begin{aligned}w' &= -10^{12}y - 10^9w, & w(0) &= 1, \\x' &= y - z - x, & x(0) &= 1, \\y' &= 10^{12}w - 10^{12}x, & y(0) &= 1, \\z' &= x, & z(0) &= 1.\end{aligned}\tag{A.1}$$

A.1 Analytické řešení

Ze soustavy diferenciálních rovnic (A.1) vyjádříme w, x, y za pomoci funkce z a jejich derivací z', z'', z'''

$$\begin{aligned}x &= z', \\y &= z'' + z' + z, \\w &= \frac{z''' + z'' + z'}{10^{12}} + z'.\end{aligned}\tag{A.2}$$

Ze soustavy (A.1) vyjádříme postupně $z'', z''', z^{(4)}$

$$\begin{aligned}z'' &= x' = y - z - x, \\z''' &= y' - z' - z'' = 10^{12}w - 10^{12}x - x - y + z + x, \\z^{(4)} &= 10^{12}w' - 10^{12}x' - y' + z' = \\&= 10^{12}(-10^{12}y - 10^9w) - 10^{12}(y - z - x) - 10^{12}w + 10^{12}x + x,\end{aligned}$$

při dosazování (A.2) získáme obyčejnou diferenciální rovnici čtvrtého řádu ekvivalentní se soustavou (A.1) ve tvaru

$$z^{(4)} + (10^9 + 1)z''' + (10^{24} + 10^9 + 10^{12} + 1)z'' + (10^{24} + 10^{21} + 10^9)z' + 10^{24}z = 0, \tag{A.3}$$

z diferenciální rovnice (A.3) vyjádříme charakteristickou rovnicí ve tvaru

$$\lambda^{(4)} + (10^9 + 1)\lambda^3 + (10^{24} + 10^9 + 10^{12} + 1)\lambda^2 + (10^{24} + 10^{21} + 10^9)\lambda + 10^{24} = 0. \tag{A.4}$$

Pro kontrolu vypočteme také charakteristickou rovnicí pomocí w a jejich derivací. Nově vypočtená charakteristická rovnice musí být ekvivalentní s (A.4).

Ze soustavy (A.1) vyjádříme x, y, z za pomoci funkce w a jejich derivací w', w'', w'''

$$\begin{aligned} y &= 1 - \frac{w'+10^9w}{10^{12}}, \\ x &= \frac{w''+10^9w'}{10^{24}} + w, \\ z &= 1 - \frac{w'+10^9w}{10^{12}} - \frac{w''+10^9w'}{10^{24}} - w - \frac{w'''+10^9w''}{10^{24}} - w'. \end{aligned} \quad (\text{A.5})$$

Ze soustavy (A.1) vyjádříme postupně $w'', w''', w^{(4)}$ ve tvaru

$$\begin{aligned} w'' &= -10^{12}y' - 10^9w' = -10^{12}(10^{12}w - 10^{12}x) - 10^9w', \\ w''' &= -10^{12}(10^{12}w' - 10^{12}x') - 10^9w'' = \\ &= -10^{12}(10^{12}w' - 10^{12}(y - z - x)) - 10^9w'', \\ w^{(4)} &= -10^{12}(10^{12}w'' - 10^{12}(y' - z' - x')) - 10^9w''' = \\ &= 10^{12}(-10^{12}w'' - 10^{12}(10^{12}w - 10^{12}x - x - y + z + x)) - 10^9w''', \end{aligned}$$

při postupném dosazování (A.5) získáme diferenciální rovnici čtvrtého řádu ekvivalentní se soustavou (A.1)

$$w^{(4)} + (10^9 + 1)w''' + (10^{24} + 10^9 + 10^{12} + 1)w'' + (10^{24} + 10^{21} + 10^9)w' + 10^{24}w = 0, \quad (\text{A.6})$$

z diferenciální rovnice (A.6) vyjádříme charakteristickou rovnici

$$\lambda^{(4)} + (10^9 + 1)\lambda^3 + (10^{24} + 10^9 + 10^{12} + 1)\lambda^2 + (10^{24} + 10^{21} + 10^9)\lambda + 10^{24} = 0. \quad (\text{A.7})$$

Charakteristické rovnice (A.7) a (A.4) jsou, jak jsme předpokládali, ekvivalentní.

Řešme charakteristickou rovnici (A.4) pomocí programu Maple

```
evalf(allvalues(solve(a^4+((1+10^9)*a^3)+((10^24+10^9+10^12+1)*a^2)
+((20^24+10^9+10^21)*a)+10^24=0,a)));
*****
```

výsledek je ve tvaru

$$\begin{aligned} &10.65222947 + 0.2686424830 \cdot 10^8 I, -11.05222947 - 0.2686424830 \cdot 10^8 I, \\ &-0.5000000004 \cdot 10^9 - 0.9999998750 \cdot 10^{12} I, -0.5000000004 \cdot 10^9 + 0.9999998750 \cdot 10^{12} I \end{aligned}$$

Řešme nyní výpočet vlastních čísel soustavy (A.1) a výpočet charakteristické rovnice (A.4) pomocí programu MATLAB

```
syms w x y
z~f=[10^12-10^12*y-10^9*w; y-z-x; 10^12*w-10^12*x; x]
Jf=jacobian(f)
eigJf=eval(eig(Jf))

eig_charEq=eval(solve('a^4+((1+10^9)*a^3)+((10^24+10^9+10^12+1)*a^2)
+((10^24+10^9+10^21)*a)+10^24=0','a'))
```

výsledek obdržíme ve tvaru

```
Jf =
[ -1000000000, 0, -1000000000000, 0]
[ 0, -1, 1, -1]
[ 1000000000000, -1000000000000, 0, 0]
[ 0, 1, 0, 0]
```

```
eigJf =
8291320,65019803 - 6,01864820431015e-05i
-8291321,71994416 + 6,01864830411525e-05i
-499999999,965066 + 999999875034,864i
-499999999,965188 - 999999875034,864i
```

```
eig_charEq =
8291320,65019803 - 6,01864820431015e-05i
-8291321,71994416 + 6,01864830411525e-05i
-499999999,965066 + 999999875034,864i
-499999999,965188 - 999999875034,864i
```

Vidíme, že výsledky z programů Maple a MATLAB se liší. Vypočteme tedy v programu MATLAB vlastní čísla přímo z Jacobiho matice soustavy (A.1)

$$\mathcal{J} = \begin{pmatrix} -1 \cdot 10^9 & 0 & -1 \cdot 10^{12} & 0 \\ 0 & -1 & 1 & -1 \\ 1 \cdot 10^{12} & -1 \cdot 10^{12} & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}.$$

Pro kontrolu si sestavíme charakteristickou rovnici

$$\begin{vmatrix} -1 \cdot 10^9 - \lambda & 0 & -1 \cdot 10^{12} & 0 \\ 0 & -1 - \lambda & 1 & -1 \\ 1 \cdot 10^{12} & -1 \cdot 10^{12} & 0 - \lambda & 0 \\ 0 & 1 & 0 & 0 - \lambda \end{vmatrix} = 0,$$

$$\lambda^4 + (10^9 + 1)\lambda^3 + (10^9 + 10^{24} + 1)\lambda^2 + (10^9 + 10^{24})\lambda + 10^{24} = 0.$$

Zdrojový kód pro program MATLAB

```
A=[ -10^9 0 -10^12 0; 0 -1 1 -1; 10^12 -10^12 0 0; 0 1 0 0]
eigA_stiff=eig(A)

eigCharEq=eval(solve('a^4+((1+10^9)*a^3)+((10^9+10^24+1)*a^2)
+((10^9+10^24)*a)+10^24=0', 'a'))
```

výsledek je ve tvaru

```

eigA_stiff=
-499999999,999512 + 999999875000,492i
-499999999,999512 - 999999875000,492i
-0,500499999999500 + 0,865736536135273i
-0,500499999999500 - 0,865736536135273i

```

```

eigCharEq=
-500000000,000000 + 999999874999,992i
-500000000,000000 - 999999874999,992i
-0,5000000000000000 + 0,866025403784439i
-0,5000000000000000 - 0,866025403784439i

```

obdrželi jsme tedy hodnoty

$$Re_1 = -0.5, Im_1 = 0.8657, Re_2 = -5 \cdot 10^8, Im_2 = 10^{12}.$$

Očekávané řešení funkce z je ve tvaru

$$z = e^{Re_1 t}(K_1 \cos(Im_1 t) + K_2 \sin(Im_1 t)) + e^{Re_2 t}(K_3 \cos(Im_2 t) + K_4 \sin(Im_2 t)), \quad (\text{A.8})$$

kde nás zajímají neznámé hodnoty konstant K_1, K_2, K_3, K_4 .

Vyjádríme vyšší derivace z ve tvaru

$$\begin{aligned} z' = & Re_1 e^{Re_1 t}(K_1 \cos(Im_1 t) + K_2 \sin(Im_1 t)) + \\ & + e^{Re_1 t}(-K_1 Im_1 \sin(Im_1 t) + K_2 Im_1 \cos(Im_1 t)) + \\ & + Re_2 e^{Re_2 t}(K_3 \cos(Im_2 t) + K_4 \sin(Im_2 t)) + \\ & + e^{Re_2 t}(-K_3 Im_2 \sin(Im_2 t) + K_4 Im_2 \cos(Im_2 t)), \end{aligned}$$

$$\begin{aligned} z'' = & (Re_1)^2 e^{Re_1 t}(K_1 \cos(Im_1 t) + K_2 \sin(Im_1 t)) + \\ & + Re_1 e^{Re_1 t}(-K_1 Im_1 \sin(Im_1 t) + K_2 Im_1 \cos(Im_1 t)) + \\ & + Re_1 e^{Re_1 t}(-K_1 Im_1 \sin(Im_1 t) + K_2 Im_1 \cos(Im_1 t)) + \\ & + e^{Re_1 t}(-K_1 (Im_1)^2 \cos(Im_1 t) - K_2 (Im_1)^2 \sin(Im_1 t)) + \\ & + (Re_2)^2 e^{Re_2 t}(K_3 \cos(Im_2 t) + K_4 \sin(Im_2 t)) + \\ & + Re_2 e^{Re_2 t}(-K_3 Im_2 \sin(Im_2 t) + K_4 Im_2 \cos(Im_2 t)) + \\ & + Re_2 e^{Re_2 t}(-K_3 Im_2 \sin(Im_2 t) + K_4 Im_2 \cos(Im_2 t)) + \\ & + e^{Re_2 t}(-K_3 (Im_2)^2 \cos(Im_2 t) - K_4 (Im_2)^2 \sin(Im_2 t)), \end{aligned}$$

$$\begin{aligned}
z''' = & (Re_1)^3 e^{Re_1 t} (K_1 \cos(Im_1 t) + K_2 \sin(Im_1 t)) + \\
& + (Re_1)^2 e^{Re_1 t} (-K_1 Im_1 \sin(Im_1 t) + K_2 Im_1 \cos(Im_1 t)) + \\
& + (Re_1)^2 e^{Re_1 t} (-K_1 Im_1 \sin(Im_1 t) + K_2 Im_1 \cos(Im_1 t)) + \\
& + Re_1 e^{Re_1 t} (-K_1 (Im_1)^2 \cos(Im_1 t) - K_2 (Im_1)^2 \sin(Im_1 t)) + \\
& + (Re_1)^2 e^{Re_1 t} (-K_1 Im_1 \sin(Im_1 t) + K_2 Im_1 \cos(Im_1 t)) + \\
& + Re_1 e^{Re_1 t} (-K_1 (Im_1)^2 \cos(Im_1 t) - K_2 (Im_1)^2 \sin(Im_1 t)) + \\
& + Re_1 e^{Re_1 t} (-K_1 (Im_1)^2 \cos(Im_1 t) - K_2 (Im_1)^2 \sin(Im_1 t)) + \\
& + e^{Re_1 t} (K_1 (Im_1)^3 \sin(Im_1 t) - K_2 (Im_1)^3 \cos(Im_1 t)) + \\
& + (Re_2)^3 e^{Re_2 t} (K_3 \cos(Im_2 t) + K_4 \sin(Im_2 t)) + \\
& + (Re_2)^2 e^{Re_2 t} (-K_3 Im_2 \sin(Im_2 t) + K_4 Im_2 \cos(Im_2 t)) + \\
& + (Re_2)^2 e^{Re_2 t} (-K_3 Im_2 \sin(Im_2 t) + K_4 Im_2 \cos(Im_2 t)) + \\
& + Re_2 e^{Re_2 t} (-K_3 (Im_2)^2 \cos(Im_2 t) - K_4 (Im_2)^2 \sin(Im_2 t)) + \\
& + (Re_2)^2 e^{Re_2 t} (-K_3 Im_2 \sin(Im_2 t) + K_4 Im_2 \cos(Im_2 t)) + \\
& + Re_2 e^{Re_2 t} (-K_3 (Im_2)^2 \cos(Im_2 t) - K_4 (Im_2)^2 \sin(Im_2 t)) + \\
& + Re_2 e^{Re_2 t} (-K_3 (Im_2)^2 \cos(Im_2 t) - K_4 (Im_2)^2 \sin(Im_2 t)) + \\
& + e^{Re_2 t} (K_3 (Im_2)^3 \sin(Im_2 t) - K_4 (Im_2)^3 \cos(Im_2 t)) .
\end{aligned} \tag{A.9}$$

Ze soustavy (A.1) vyjádříme počáteční podmínky $z'(0)$, $z''(0)$, $z'''(0)$ ve tvaru

$$\begin{aligned}
z'(0) &= x(0) = 1 , \\
z''(0) &= x'(0) = y(0) - z(0) - x(0) = -1 , \\
z'''(0) &= y'(0) - z'(0) - x'(0) = \\
&= 10^{12} w(0) - 10^{12} x(0) - x(0) - y(0) + z(0) + x(0) = 0 .
\end{aligned} \tag{A.10}$$

Dosadíme počáteční podmínky $z(0)$, $z'(0)$, $z''(0)$, $z'''(0)$ do (A.8) a (A.9)

$$\begin{aligned}
z(0) &= K_1 + K_3 , \\
z'(0) &= Re_1 K_1 + Im_1 K_2 + Re_2 K_3 + Im_2 K_4 , \\
z''(0) &= (Re_1)^2 K_1 + Re_1 Im_1 K_2 + Re_1 Im_1 K_2 - (Im_1)^2 K_1 + \\
&+ (Re_2)^2 K_3 + Re_2 Im_2 K_4 + Re_2 Im_2 K_4 + (Im_2)^2 K_3 , \\
z'''(0) &= (Re_1)^3 K_1 + (Re_1)^2 Im_1 K_2 + (Re_1)^2 Im_1 K_2 - Re_1 (Im_1) K_1 + \\
&+ (Re_1)^2 Im_1 K_2 - Re_1 (Im_1)^2 K_1 - Re_1 (Im_1)^2 K_1 - (Im_1)^3 K_2 + \\
&+ (Re_2)^3 K_3 + (Re_2)^2 Im_2 K_4 + (Re_2)^2 Im_2 K_4 - Re_2 (Im_2)^2 K_3 + \\
&+ (Re_2)^2 Im_2 K_4 - Re_2 (Im_2)^2 K_3 - Re_2 (Im_2)^2 K_3 - (Im_2)^3 K_4 .
\end{aligned} \tag{A.11}$$

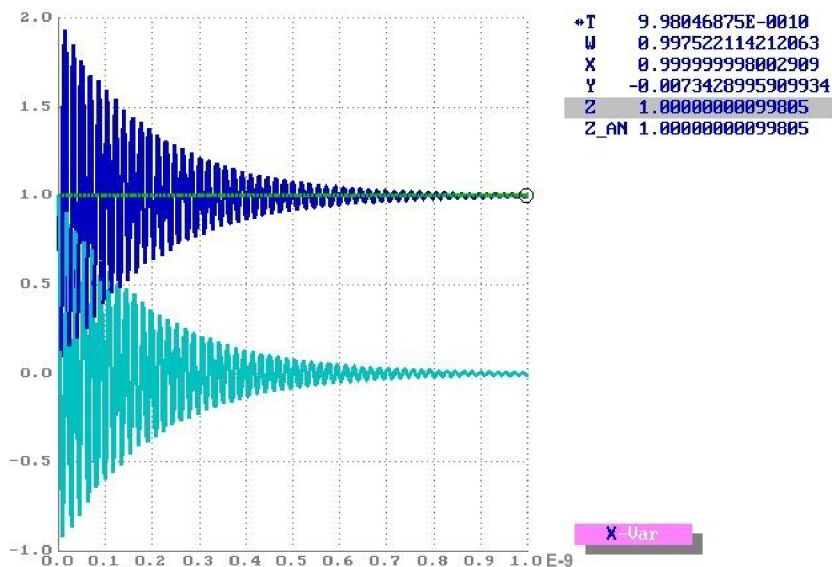
Ze soustavy (A.11) vypočteme konstanty K_1 , K_2 , K_3 , K_4

$$\begin{aligned}
K_1 &= 1 , \\
K_2 &= 1.733206279 , \\
K_3 &= -0.1000998749 \cdot 10^{-23} , \\
K_4 &= -0.1501499310 \cdot 10^{-26} .
\end{aligned}$$

Výsledné analytické řešení je ve tvaru

$$z = e^{-0.5t}(\cos(0.8657t) + 1.733206279 \sin(0.8657t)) + e^{-5 \cdot 10^8 t}(-0.1000998749 \cdot 10^{-23} \cos(10^{12}t) - 0.1501499310 \cdot 10^{-26} \sin(10^{12}t)). \quad (\text{A.12})$$

V Obr. A.1 nalezneme porovnání námi nalezené analytické řešení “Z_AN” (A.12) s numerickým řešením “Z” v TKSL/386.



Obrázek A.1: Řešení TKSL/386

Pro názornost uvedeme ukázkou analytického řešení soustavy (A.1) pomocí programu Maple. Zdrojový kód pro program Maple je ve tvaru

```
R1:=0.001; L1:=1*10^(-12); C1:=1*10^(-12); R2:=1; L2:=1; C2:=1;
eq1:= diff(w(t),t)=(1/L1)*(-y(t)-R1*w(t));
eq2:= diff(x(t),t)=(1/L2)*(y(t)-z(t)-R2*x(t));
eq3:= diff(y(t),t)=(1/C1)*(w(t)-x(t));
eq4:= diff(z(t),t)=(1/C2)*(x(t));
sol:= dsolve({eq1, w(0)=1, eq2, x(0)=1, eq3, y(0)=1,
             eq4, z(0)=1}, {w(t),x(t),y(t),z(t)});
```


řešení je poté ve tvaru

$$\begin{aligned}
sol := \{ & z(t) = (.2000000000e - 22 + .2000000000e - 9 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (.1000000000e - 12 - .1000000000e - 25 * I) * \exp(-500000000. * t) * \sin(.9999998750e12 * t) + \\
& + (.5781204550 - .5781204550e - 13 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (1.000000000 - .1000000000e - 12 * I) * \exp(-.5005000000 * t) * \cos(.8657365361 * t) \\
& + (-.6249999990e - 13 + .6249999990e - 26 * I) * \exp(-500000000. * t) * \cos(.9999998750e12 * t) \\
& + (.1000500125e - 26 + .7499291372e - 30 * I) * \exp(-500000000. * t) * \cos(.9999998750e12 * t) \\
& + (-.6249999990e - 26 - .6249999990e - 13 * I) * \exp(-500000000. * t) * \sin(.9999998750e12 * t) \\
& + (.1000000000e - 25 + .1000000000e - 12 * I) * \exp(-500000000. * t) * \cos(.9999998750e12 * t) \\
& + (.7311680150e - 33 - .4999998738e - 21 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (.1153065370e - 14 + .1155664378e - 14 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (-.4991251604e - 24 - .5002501880e - 24 * I) * \exp(-.5005000000 * t) * \cos(.8657365361 * t) \\
& + (-.9982498215e - 24 - .1000499875e - 23 * I) * \exp(-500000000. * t) * \cos(.9999998750e12 * t) \\
& + (-.5002501880e - 24 + .4991251604e - 24 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (-.1874822840e - 36 + .2501250308e - 33 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (.1155663512e - 11 + .8662325162e - 15 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (-.2501250308e - 33 - .1874822840e - 36 * I) * \exp(-.5005000000 * t) * \cos(.8657365361 * t) \\
& + (.2922827329e - 36 - .1998738000e - 24 * I) * \exp(-500000000. * t) * \sin(.9999998750e12 * t) \\
& + (.4999998738e - 21 + .7311680150e - 33 * I) * \exp(-.5005000000 * t) * \cos(.8657365361 * t) \\
& + (.1262000000e - 27 + .1845468536e - 39 * I) * \exp(-500000000. * t) * \cos(.9999998750e12 * t) \\
& + (1.155085824 + .1689124044e - 11 * I) * \exp(-.5005000000 * t) * \sin(.8657365361 * t) \\
& + (-.1000499751e - 23 - .7499288567e - 27 * I) * \exp(-500000000. * t) * \sin(.9999998750e12 * t) \\
& + (-.1497375917e - 26 - .1500751001e - 26 * I) * \exp(-500000000. * t) * \sin(.9999998750e12 * t), \\
& w(t) = \dots, x(t) = \dots, y(t) = \dots \}
\end{aligned}$$

Hledíme nyní analytické řešení z pro soustavu diferenciálních rovnic ve tvaru

$$\begin{aligned}
w' &= -10^{12}y - 10^9w, & w(0) &= 0, \\
x' &= y - z - x, & x(0) &= 0, \\
y' &= 10^{12}w - 10^{12}x, & y(0) &= 1, \\
z' &= x, & z(0) &= 1.
\end{aligned} \tag{A.13}$$

Soustava (A.13) popisuje v čase $t = 0$ ustálený stav v RLC obvodu viz 4.3.2 (obvodem neprotéká žádný proud $w(0) = x(0) = 0$ a napětí na kondenzátorech je rovno vstupnímu napětí $y(0) = z(0) = 1$).

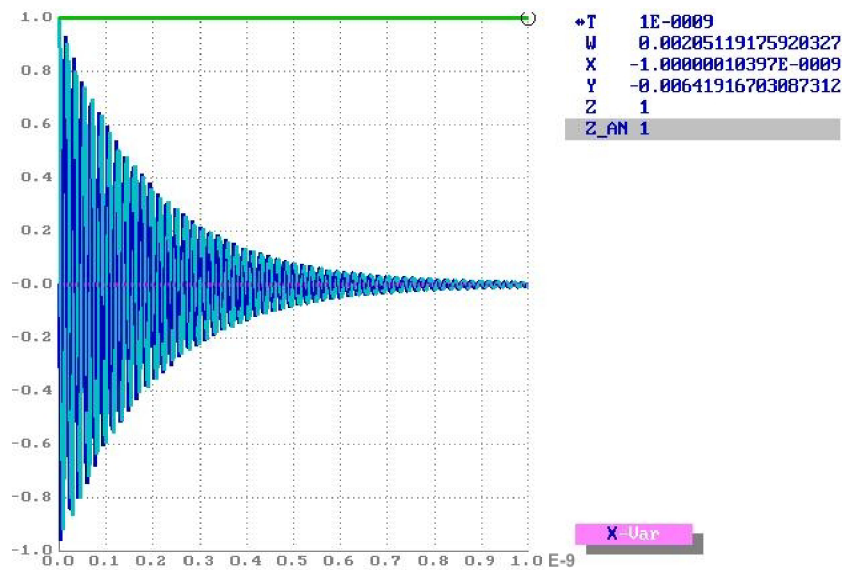
Analytické řešení funkce z vyjádříme podobně jako pro soustavu (A.13). Pro počáteční podmínky $z(0) = 1, z'(0) = 0, z''(0) = 0, z'''(0) = 0$ vypočteme ze soustavy (A.11) konstanty K_1, K_2, K_3, K_4

$$\begin{aligned}
K_1 &= 1, \\
K_2 &= 0.5781204548, \\
K_3 &= -0.9999987499 \cdot 10^{-24}, \\
K_4 &= -0.1499999311 \cdot 10^{-26}.
\end{aligned}$$

Výsledné analytické řešení je ve tvaru

$$\begin{aligned}
z &= e^{-0.5t}(\cos(0.8657t) + 0.5781204548 \sin(0.8657t)) + \\
&+ e^{-5 \cdot 10^8 t}(-0.9999987499 \cdot 10^{-24} \cos(10^{12}t) - 0.1499999311 \cdot 10^{-26} \sin(10^{12}t)).
\end{aligned} \tag{A.14}$$

V Obr. A.2 nalezneme porovnání námi nalezeného analytického řešení “Z_AN” (A.14) s numerickým řešením “Z” v TKSL/386 .



Obrázek A.2: Řešení TKSL/386

A.2 Numerické řešení

Shodnoťme možnosti řešení soustavy ODR (A.13) pomocí rekurentního výpočtu členů explicitní a implicitní Taylorovy metody.

A.2.1 Explicitní Taylorova metoda

Vyjádríme předcházející soustavu diferenciálních rovnic (A.13) pomocí explicitní Taylorovy řady ve tvaru

$$\begin{aligned}
 w_{i+1} &= w_i + DW1_i + DW2_i + \dots + DWn_i, \\
 x_{i+1} &= x_i + DX1_i + DX2_i + \dots + DXn_i, \\
 y_{i+1} &= y_i + DY1_i + DY2_i + \dots + DYn_i, \\
 z_{i+1} &= z_i + DZ1_i + DZ2_i + \dots + DZn_i,
 \end{aligned}$$

kde členy vyjádříme rekurentně ve tvaru

$$\begin{aligned}
 DW1_i &= h(-10^{12}y_i - 10^9w_i), & DX1_i &= h(y_i - z_i - x_i), \\
 DW2_i &= \frac{h}{2}(-10^{12}DY1_i - 10^9DW1_i), & DX2_i &= \frac{h}{2}(DY1_i - DZ1_i - DX1_i), \\
 &\vdots & &\vdots \\
 DWn_i &= \frac{h}{n}(-10^{12}DY(n-1)_i - 10^9DW(n-1)_i), & DXn_i &= \frac{h}{n}(DY(n-1)_i - DZ(n-1)_i - DX(n-1)_i), \\
 DY1_i &= h(10^{12}w_i - 10^{12}x_i), & DZ1_i &= h(x_i), \\
 DY2_i &= \frac{h}{2}(10^{12}DW1_i - 10^{12}DX1_i), & DZ2_i &= \frac{h}{2}(DX1_i), \\
 &\vdots & &\vdots \\
 DYn_i &= \frac{h}{n}(10^{12}DW(n-1)_i - 10^{12}DX(n-1)_i), & DZn_i &= \frac{h}{n}(DX(n-1)_i).
 \end{aligned}$$

Zaměříme se nyní na odvození členů u implicitní Taylorovy řady.

A.2.2 Implicitní Taylorova metoda

Vyjádřeme předcházející soustavu diferenciálních rovnic (A.13) pomocí implicitní Taylorovy řady ve tvaru

$$\begin{aligned}
 w_{i+1} &= w_i + hw'_{i+1} - \frac{h^2}{2}w''_{i+1} + \frac{h^3}{3!}w'''_{i+1} - \dots - \frac{(-h)^k}{k!}w_{i+1}^{(k)}, \\
 x_{i+1} &= x_i + hx'_{i+1} - \frac{h^2}{2}x''_{i+1} + \frac{h^3}{3!}x'''_{i+1} - \dots - \frac{(-h)^k}{k!}x_{i+1}^{(k)}, \\
 y_{i+1} &= y_i + hy'_{i+1} - \frac{h^2}{2}y''_{i+1} + \frac{h^3}{3!}y'''_{i+1} - \dots - \frac{(-h)^k}{k!}y_{i+1}^{(k)}, \\
 z_{i+1} &= z_i + hz'_{i+1} - \frac{h^2}{2}z''_{i+1} + \frac{h^3}{3!}z'''_{i+1} - \dots - \frac{(-h)^k}{k!}z_{i+1}^{(k)},
 \end{aligned} \tag{A.15}$$

kde vyšší derivace vyjádříme ve tvaru

$$\begin{aligned}
 w'_{i+1} &= -10^{12}y_{i+1} - 10^9w_{i+1}, \\
 x'_{i+1} &= y_{i+1} - z_{i+1} - x_{i+1}, \\
 y'_{i+1} &= 10^{12}w_{i+1} - 10^{12}x_{i+1}, \\
 z'_{i+1} &= x_{i+1}, \\
 &\vdots \\
 y''_{i+1} &= -10^{12}y'_{i+1} - 10^9w'_{i+1}, \\
 x''_{i+1} &= y'_{i+1} - z'_{i+1} - x'_{i+1}, \\
 y''_{i+1} &= 10^{12}w'_{i+1} - 10^{12}x'_{i+1}, \\
 z''_{i+1} &= x'_{i+1}, \\
 &\vdots \\
 y_{i+1}^{(n)} &= -10^{12}y_{i+1}^{(n-1)} - 10^9w_{i+1}^{(n-1)}, \\
 x_{i+1}^{(n)} &= y_{i+1}^{(n-1)} - z_{i+1}^{(n-1)} - x_{i+1}^{(n-1)}, \\
 y_{i+1}^{(n)} &= 10^{12}w_{i+1}^{(n-1)} - 10^{12}x_{i+1}^{(n-1)}, \\
 z_{i+1}^{(n)} &= x_{i+1}^{(n-1)}.
 \end{aligned}$$

Ze soustavy (A.15) po dosazování vyšších derivací vyjádříme $w_{i+1}, x_{i+1}, y_{i+1}, z_{i+1}$ ve tvaru

ORD = 1

$$\begin{aligned}jm &= 1000000001 \cdot h + 0.1001 \cdot 10^{25} \cdot h^3 + 0.1 \cdot 10^{25} \cdot h^2 + 0.1 \cdot 10^{25} \cdot h^4 + 1, \\w_{i+1} &= (0.1 \cdot 10^{25} \cdot h^2 \cdot x_i - 0.1 \cdot 10^{13} \cdot h^2 \cdot y_i - 0.1 \cdot 10^{13} \cdot h^3 \cdot y_i - \\&\quad - 0.1 \cdot 10^{25} \cdot h^3 \cdot z_i + 0.1 \cdot 10^{13} \cdot w_i \cdot h^2 + w_i \cdot h + w_i - \\&\quad - 0.1 \cdot 10^{13} \cdot h \cdot y_i)/jm, \\x_{i+1} &= (10^9 \cdot h \cdot x_i + 10^9 \cdot h^2 \cdot y_i - 10^9 \cdot h^2 \cdot z_i + 0.1 \cdot 10^{25} \cdot h^2 \cdot x_i - \\&\quad - 0.1 \cdot 10^{25} \cdot h^3 \cdot z_i + x_i + h \cdot y_i - h \cdot z_i + 0.1 \cdot 10^{13} \cdot w_i \cdot h^2)/jm, \\y_{i+1} &= (1000000001 \cdot h \cdot y_i + 10^9 \cdot h^3 \cdot y_i + 1000000001 \cdot h^2 \cdot y_i + y_i + \\&\quad + 0.1 \cdot 10^{13} \cdot w_i \cdot h^3 + 0.1 \cdot 10^{13} \cdot w_i \cdot h^2 + 0.1 \cdot 10^{13} \cdot w_i \cdot h - \\&\quad - 0.1 \cdot 10^{22} \cdot h^2 \cdot x_i + 0.1 \cdot 10^{22} \cdot h^3 \cdot z_i - 0.1 \cdot 10^{13} \cdot h \cdot x_i + \\&\quad + 0.1 \cdot 10^{13} \cdot h^2 \cdot z_i)/jm, \\z_{i+1} &= (1000000001 \cdot h \cdot z_i + 0.1001 \cdot 10^{25} \cdot h^3 \cdot z_i + 0.1 \cdot 10^{25} \cdot h^2 \cdot z_i + \\&\quad + z_i + 10^9 \cdot h^2 \cdot x_i + 10^9 \cdot h^3 \cdot y_i + 0.1 \cdot 10^{25} \cdot x_i \cdot h^3 + h \cdot x_i + \\&\quad + h^2 \cdot y_i + 0.1 \cdot 10^{13} \cdot w_i \cdot h^3)/jm,\end{aligned}$$

ORD = 2

$$\begin{aligned}
jm &= 4 + 0.2000002004 \cdot 10^{19} \cdot h^2 + 0.2 \cdot 10^{34} \cdot h^3 + 0.1 \cdot 10^{49} \cdot h^4 + \\
&\quad + 0.5005 \cdot 10^{48} \cdot h^7 + 0.501 \cdot 10^{48} \cdot h^6 + 0.25000025 \cdot 10^{48} \cdot h^8 + \\
&\quad + 0.1001 \cdot 10^{49} \cdot h^5 + 4000000004 \cdot h, \\
w_{i+1} &= -1 \cdot (0.4 \cdot 10^{13} \cdot h \cdot y_i + 0.2000000004 \cdot 10^{22} \cdot h^2 \cdot y_i - 4 \cdot w_i \cdot h - \\
&\quad - 4 \cdot w_i - 0.2 \cdot 10^{25} \cdot h^2 \cdot x_i - 0.2000000002 \cdot 10^{34} \cdot x_i \cdot h^3 + \\
&\quad + 0.2000000002 \cdot 10^{22} \cdot h^3 \cdot y_i + 0.2 \cdot 10^{25} \cdot w_i \cdot h^2 + \\
&\quad + 0.2 \cdot 10^{25} \cdot w_i \cdot h^3 + 0.1 \cdot 10^{25} \cdot h^5 \cdot w_i + \\
&\quad + 0.5 \cdot 10^{36} \cdot h^6 \cdot w_i - 0.1 \cdot 10^{49} \cdot x_i \cdot h^4 + \\
&\quad + 0.1000000001 \cdot 10^{34} \cdot h^4 \cdot z_i - 0.999999999 \cdot 10^{33} \cdot h^4 \cdot y_i - \\
&\quad - 0.1 \cdot 10^{37} \cdot h^4 \cdot w_i + 0.5 \cdot 10^{48} \cdot h^6 \cdot x_i + 0.1 \cdot 10^{49} \cdot h^5 \cdot z_i + \\
&\quad - 0.5005 \cdot 10^{48} \cdot h^6 \cdot z_i + 0.5 \cdot 10^{33} \cdot h^6 \cdot y_i + 0.1001 \cdot 10^{25} \cdot h^5 \cdot y_i) / jm, \\
x_{i+1} &= -2 \cdot (0.25 \cdot 10^{36} \cdot h^6 \cdot w_i + 0.25 \cdot 10^{48} \cdot h^6 \cdot x_i + 0.25025 \cdot 10^{48} \cdot h^6 \cdot z_i + \\
&\quad + 0.25 \cdot 10^{33} \cdot h^6 \cdot y_i + 0.5 \cdot 10^{33} \cdot h^5 \cdot x_i + 0.5 \cdot 10^{48} \cdot h^5 \cdot z_i + \\
&\quad + 0.5 \cdot 10^{24} \cdot h^5 \cdot y_i - 0.5 \cdot 10^{48} \cdot x_i \cdot h^4 + 0.1 \cdot 10^{34} \cdot h^4 \cdot z_i - \\
&\quad - 0.4999999995 \cdot 10^{33} \cdot h^4 \cdot y_i - 0.5 \cdot 10^{36} \cdot h^4 \cdot w_i + \\
&\quad + 0.1000001001 \cdot 10^{19} \cdot h^3 \cdot z_i - 0.1000000001 \cdot 10^{19} \cdot h^3 \cdot y_i - \\
&\quad - 0.1 \cdot 10^{34} \cdot x_i \cdot h^3 - 0.1000000001 \cdot 10^{22} \cdot w_i \cdot h^3 - 0.1 \cdot 10^{13} \cdot w_i \cdot h^2 - \\
&\quad - 2000000001 \cdot h^2 \cdot y_i + 2000000001 \cdot h^2 \cdot z_i - 0.1 \cdot 10^{19} \cdot h^2 \cdot x_i \\
&\quad + 2 \cdot h \cdot z_i - 2 \cdot h \cdot y_i - 2 \cdot 10^9 \cdot h \cdot x_i - 2 \cdot x_i) / jm, \\
y_{i+1} &= (-0.4 \cdot 10^{13} \cdot h \cdot x_i - 0.4000000002 \cdot 10^{22} \cdot h^2 \cdot x_i - 0.1999998 \cdot 10^{25} \cdot h^2 \cdot y_i - \\
&\quad - 0.2001998 \cdot 10^{25} \cdot h^3 \cdot y_i + 0.2 \cdot 10^{13} \cdot h^2 \cdot z_i + 0.2000000002 \cdot 10^{22} \cdot h^3 \cdot z_i + \\
&\quad + 0.2000000004 \cdot 10^{22} \cdot w_i \cdot h^2 + 0.4 \cdot 10^{13} \cdot w_i \cdot h + 4000000004 \cdot h \cdot y_i + \\
&\quad + 4 \cdot y_i - 0.2000000002 \cdot 10^{31} \cdot x_i \cdot h^3 - 0.999999999 \cdot 10^{45} \cdot x_i \cdot h^4 + \\
&\quad + 0.1000001 \cdot 10^{37} \cdot h^4 \cdot z_i - 0.1000001 \cdot 10^{31} \cdot h^4 \cdot y_i + \\
&\quad + 0.2000000002 \cdot 10^{22} \cdot w_i \cdot h^3 - 0.999999999 \cdot 10^{33} \cdot h^4 \cdot w_i + 0.5 \cdot 10^{33} \cdot h^6 \cdot w_i + \\
&\quad + 0.1 \cdot 10^{37} \cdot h^5 \cdot x_i + 0.1 \cdot 10^{46} \cdot h^5 \cdot z_i - 0.998999 \cdot 10^{24} \cdot h^5 \cdot y_i + \\
&\quad + 0.1001 \cdot 10^{25} \cdot h^5 \cdot w_i + 0.5 \cdot 10^{45} \cdot h^6 \cdot x_i + 0.5004999995 \cdot 10^{45} \cdot h^6 \cdot z_i + \\
&\quad + 0.4999995 \cdot 10^{30} \cdot h^6 \cdot y_i) / jm, \\
z_{i+1} &= (4000000004 \cdot h \cdot z_i + 0.2000002002 \cdot 10^{19} \cdot x_i \cdot h^3 + 0.2 \cdot 10^{34} \cdot x_i \cdot h^4 + \\
&\quad + 0.1000001 \cdot 10^{25} \cdot h^4 \cdot y_i + 0.1 \cdot 10^{34} \cdot h^5 \cdot y_i + 0.1 \cdot 10^{49} \cdot h^4 \cdot z_i + \\
&\quad 0.1001 \cdot 10^{49} \cdot h^5 \cdot z_i + 0.1000000001 \cdot 10^{22} \cdot h^4 \cdot w_i + 4 \cdot z_i + 2000000002 \cdot h^3 \cdot y_i + \\
&\quad 4 \cdot h \cdot x_i + 2 \cdot h^2 \cdot y_i + 0.2000002004 \cdot 10^{19} \cdot h^2 \cdot z_i + 4000000002 \cdot h^2 \cdot x_i + \\
&\quad 0.2 \cdot 10^{34} \cdot h^3 \cdot z_i + 0.1 \cdot 10^{49} \cdot h^5 \cdot x_i + 0.5005 \cdot 10^{48} \cdot h^6 \cdot x_i + \\
&\quad 0.9999999995 \cdot 10^{45} \cdot h^6 \cdot z_i + 0.5004999995 \cdot 10^{33} \cdot h^6 \cdot y_i + 0.1 \cdot 10^{37} \cdot h^5 \cdot w_i + \\
&\quad 0.5005 \cdot 10^{36} \cdot h^6 \cdot w_i) / jm,
\end{aligned}$$

ORD = 3

$$\begin{aligned}
jm &= 0.8308333333 \cdot 10^{68} \cdot h^{12} - 0.2504999998 \cdot 10^{69} \cdot h^{11} - 0.25100025 \cdot 10^{72} \cdot h^{10} - \\
&\quad - 0.10015 \cdot 10^{73} \cdot h^9 - 0.1503 \cdot 10^{73} \cdot h^8 - 0.3003 \cdot 10^{73} \cdot h^7 - 0.3 \cdot 10^{73} \cdot h^6 - \\
&\quad - 0.8999999991 \cdot 10^{58} \cdot h^5 + 0.8999982 \cdot 10^{49} \cdot h^4 - 0.1800005405 \cdot 10^{29} \cdot h^3 - \\
&\quad - 0.5400005411 \cdot 10^{20} \cdot h^2 - 0.1080000001 \cdot 10^{12} \cdot h - 108, \\
w_{i+1} &= 3 \cdot (0.36 \cdot 10^{14} \cdot h \cdot y_i - 0.1200000001 \cdot 10^{35} \cdot x_i \cdot h^3 + 0.2999994 \cdot 10^{49} \cdot x_i \cdot h^4 - \\
&\quad - 0.6002994 \cdot 10^{37} \cdot h^4 \cdot y_i - 0.3000003003 \cdot 10^{43} \cdot h^5 \cdot y_i + \\
&\quad + 0.3000000003 \cdot 10^{34} \cdot h^4 \cdot z_i - 0.2999997 \cdot 10^{49} \cdot h^5 \cdot z_i + 0.3006 \cdot 10^{37} \cdot h^4 \cdot w_i + \\
&\quad + 0.6000000018 \cdot 10^{34} \cdot w_i \cdot h^3 - 36 \cdot w_i \cdot h - 36 \cdot w_i - \\
&\quad - 0.499999985 \cdot 10^{45} \cdot h^7 \cdot w_i + 0.18 \cdot 10^{26} \cdot w_i \cdot h^2 - \\
&\quad - 0.59999994 \cdot 10^{37} \cdot h^3 \cdot y_i + 0.1800000004 \cdot 10^{22} \cdot h^2 \cdot y_i - \\
&\quad - 0.18 \cdot 10^{26} \cdot h^2 \cdot x_i + 0.6 \cdot 10^{25} \cdot h^3 \cdot z_i - 0.3 \cdot 10^{58} \cdot h^5 \cdot x_i - \\
&\quad - 0.1 \cdot 10^{73} \cdot h^6 \cdot x_i + 0.1999999999 \cdot 10^{58} \cdot h^6 \cdot z_i - \\
&\quad - 0.999999999 \cdot 10^{57} \cdot h^6 \cdot y_i - 0.3 \cdot 10^{46} \cdot h^5 \cdot w_i - \\
&\quad - 0.1 \cdot 10^{61} \cdot h^6 \cdot w_i + 0.5 \cdot 10^{60} \cdot h^8 \cdot w_i + 0.1668333333 \cdot 10^{60} \cdot h^9 \cdot w_i + \\
&\quad + 0.5000000005 \cdot 10^{57} \cdot h^7 \cdot x_i + 0.1 \cdot 10^{73} \cdot h^7 \cdot z_i + 0.9999995 \cdot 10^{48} \cdot h^7 \cdot y_i + \\
&\quad + 0.5 \cdot 10^{72} \cdot h^8 \cdot x_i + 0.1668333333 \cdot 10^{72} \cdot h^9 \cdot x_i + 0.5005 \cdot 10^{72} \cdot h^8 \cdot z_i + \\
&\quad + 0.3333333332 \cdot 10^{69} \cdot h^9 \cdot z_i + 0.1668333332 \cdot 10^{57} \cdot h^9 \cdot y_i + \\
&\quad + 0.5 \cdot 10^{57} \cdot h^8 \cdot y_i) / jm, \\
x_{i+1} &= 2 \cdot (-54 \cdot h \cdot y_i + 54 \cdot h \cdot z_i - 0.9 \cdot 10^{28} \cdot x_i \cdot h^3 + 0.4499991 \cdot 10^{49} \cdot x_i \cdot h^4 + \\
&\quad + 0.4499991004 \cdot 10^{34} \cdot h^4 \cdot y_i - 0.4499999996 \cdot 10^{43} \cdot h^5 \cdot y_i + \\
&\quad + 0.9000018014 \cdot 10^{28} \cdot h^4 \cdot z_i - 0.4499991 \cdot 10^{49} \cdot h^5 \cdot z_i + \\
&\quad + 0.4499991 \cdot 10^{37} \cdot h^4 \cdot w_i - 0.1800000002 \cdot 10^{22} \cdot w_i \cdot h^3 - \\
&\quad - 54 \cdot x_i + 0.7500000007 \cdot 10^{45} \cdot h^7 \cdot w_i - 0.27 \cdot 10^{14} \cdot w_i \cdot h^2 + \\
&\quad + 0.8999973 \cdot 10^{25} \cdot h^3 \cdot y_i - 0.54 \cdot 10^{11} \cdot h \cdot x_i - 0.5400000003 \cdot 10^{11} \cdot h^2 \cdot y_i + \\
&\quad + 0.5400000003 \cdot 10^{11} \cdot h^2 \cdot z_i - 0.27 \cdot 10^{20} \cdot h^2 \cdot x_i + \\
&\quad + 0.2700001803 \cdot 10^{20} \cdot h^3 \cdot z_i - 0.45 \cdot 10^{58} \cdot h^5 \cdot x_i - \\
&\quad - 0.15 \cdot 10^{73} \cdot h^6 \cdot x_i + 0.4499999998 \cdot 10^{58} \cdot h^6 \cdot z_i - \\
&\quad - 0.1499999999 \cdot 10^{58} \cdot h^6 \cdot y_i - 0.45 \cdot 10^{46} \cdot h^5 \cdot w_i - \\
&\quad - 0.15 \cdot 10^{61} \cdot h^6 \cdot w_i + 0.75 \cdot 10^{60} \cdot h^8 \cdot w_i + 0.25025 \cdot 10^{60} \cdot h^9 \cdot w_i + \\
&\quad + 0.2249999999 \cdot 10^{58} \cdot h^7 \cdot x_i + 0.15 \cdot 10^{73} \cdot h^7 \cdot z_i + \\
&\quad + 0.150000075 \cdot 10^{49} \cdot h^7 \cdot y_i + 0.75 \cdot 10^{72} \cdot h^8 \cdot x_i + 0.25025 \cdot 10^{72} \cdot h^9 \cdot x_i + \\
&\quad + 0.75075 \cdot 10^{72} \cdot h^8 \cdot z_i + 0.4999999998 \cdot 10^{69} \cdot h^9 \cdot z_i + \\
&\quad + 0.2502499997 \cdot 10^{57} \cdot h^9 \cdot y_i + 0.75 \cdot 10^{57} \cdot h^8 \cdot y_i) / jm,
\end{aligned}$$

$$\begin{aligned}
y_{i+1} &= -3 \cdot (0.3600000004 \cdot 10^{11} \cdot h \cdot y_i + 0.5999982 \cdot 10^{37} \cdot x_i \cdot h^3 + \\
&+ 0.2999994003 \cdot 10^{46} \cdot x_i \cdot h^4 - 0.1200899401 \cdot 10^{35} \cdot h^4 \cdot y_i - \\
&- 0.3000006006 \cdot 10^{40} \cdot h^5 \cdot y_i - 0.2999991 \cdot 10^{37} \cdot h^4 \cdot z_i + \\
&+ 0.3000003006 \cdot 10^{40} \cdot h^5 \cdot z_i - 0.6002994 \cdot 10^{37} \cdot h^4 \cdot w_i - \\
&- 0.5999994 \cdot 10^{37} \cdot w_i \cdot h^3 + 0.36 \cdot 10^{14} \cdot w_i \cdot h + 36 \cdot y_i + 0.9999995 \cdot 10^{48} \cdot h^7 \cdot w_i + \\
&+ 0.1800000004 \cdot 10^{22} \cdot w_i \cdot h^2 - 0.1199999402 \cdot 10^{35} \cdot h^3 \cdot y_i - 0.36 \cdot 10^{14} \cdot h \cdot x_i - \\
&- 0.17999982 \cdot 10^{26} \cdot h^2 \cdot y_i + 0.18 \cdot 10^{14} \cdot h^2 \cdot z_i - 0.3600000002 \cdot 10^{22} \cdot h^2 \cdot x_i + \\
&+ 0.1800000001 \cdot 10^{22} \cdot h^3 \cdot z_i - 0.2999999997 \cdot 10^{55} \cdot h^5 \cdot x_i - \\
&- 0.999999999 \cdot 10^{69} \cdot h^6 \cdot x_i + 0.1000002 \cdot 10^{61} \cdot h^6 \cdot z_i - 0.999999998 \cdot 10^{54} \cdot h^6 \cdot y_i - \\
&- 0.3000003003 \cdot 10^{43} \cdot h^5 \cdot w_i - 0.999999999 \cdot 10^{57} \cdot h^6 \cdot w_i + 0.5 \cdot 10^{57} \cdot h^8 \cdot w_i + \\
&+ 0.1668333332 \cdot 10^{57} \cdot h^9 \cdot w_i + 0.10000005 \cdot 10^{61} \cdot h^7 \cdot x_i + 0.1 \cdot 10^{70} \cdot h^7 \cdot z_i + \\
&+ 0.1999999499 \cdot 10^{46} \cdot h^7 \cdot y_i + 0.5 \cdot 10^{69} \cdot h^8 \cdot x_i + 0.1668333332 \cdot 10^{69} \cdot h^9 \cdot x_i + \\
&+ 0.5004999995 \cdot 10^{69} \cdot h^8 \cdot z_i + 0.3333333333 \cdot 10^{66} \cdot h^9 \cdot z_i + \\
&+ 0.1668333333 \cdot 10^{54} \cdot h^9 \cdot y_i + 0.5 \cdot 10^{54} \cdot h^8 \cdot y_i) / jm, \\
z_{i+1} &= -1 \cdot (0.1080000001 \cdot 10^{12} \cdot h \cdot z_i + 0.5400003605 \cdot 10^{20} \cdot x_i \cdot h^3 + \\
&+ 0.1800003603 \cdot 10^{29} \cdot x_i \cdot h^4 - 0.8999973 \cdot 10^{25} \cdot h^4 \cdot y_i + \\
&+ 0.9000009018 \cdot 10^{28} \cdot h^5 \cdot y_i - 0.8999982 \cdot 10^{49} \cdot h^4 \cdot z_i + \\
&+ 0.8999999991 \cdot 10^{58} \cdot h^5 \cdot z_i + 0.9000000009 \cdot 10^{22} \cdot h^4 \cdot w_i + \\
&+ 0.18 \cdot 10^{14} \cdot w_i \cdot h^3 + 108 \cdot z_i + 0.3 \cdot 10^{61} \cdot h^7 \cdot w_i + \\
&+ 0.5400000004 \cdot 10^{11} \cdot h^3 \cdot y_i + 108 \cdot h \cdot x_i + 54 \cdot h^2 \cdot y_i + \\
&+ 0.5400005411 \cdot 10^{20} \cdot h^2 \cdot z_i + 0.1080000001 \cdot 10^{12} \cdot h^2 \cdot x_i + \\
&+ 0.1800005405 \cdot 10^{29} \cdot h^3 \cdot z_i - 0.8999982 \cdot 10^{49} \cdot h^5 \cdot x_i + \\
&+ 0.8999999996 \cdot 10^{58} \cdot h^6 \cdot x_i + 0.3 \cdot 10^{73} \cdot h^6 \cdot z_i + \\
&+ 0.3000006 \cdot 10^{49} \cdot h^6 \cdot y_i - 0.8999991 \cdot 10^{37} \cdot h^5 \cdot w_i + \\
&+ 0.5999999998 \cdot 10^{46} \cdot h^6 \cdot w_i + 0.15015 \cdot 10^{61} \cdot h^8 \cdot w_i + \\
&+ 0.9999999995 \cdot 10^{57} \cdot h^9 \cdot w_i + 0.3 \cdot 10^{73} \cdot h^7 \cdot x_i + \\
&+ 0.3003 \cdot 10^{73} \cdot h^7 \cdot z_i + 0.3 \cdot 10^{58} \cdot h^7 \cdot y_i + \\
&+ 0.15015 \cdot 10^{73} \cdot h^8 \cdot x_i + 0.9999999995 \cdot 10^{69} \cdot h^9 \cdot x_i + \\
&+ 0.2999999999 \cdot 10^{70} \cdot h^8 \cdot z_i + 0.5 \cdot 10^{69} \cdot h^9 \cdot z_i + \\
&+ 0.999999999 \cdot 10^{54} \cdot h^9 \cdot y_i + 0.1501499999 \cdot 10^{58} \cdot h^8 \cdot y_i) / jm. \\
&\vdots
\end{aligned}$$

Zaměříme se nyní na možnosti odvození rekurentního výpočtu členů u implicitní Taylorovy metody s Newtonovou iterací (ITMRN).

A.2.3 ITMRN

Vyjádříme soustavu diferenciálních rovnic (A.13) pomocí implicitní Taylorovy řady ve tvaru

$$\begin{aligned} w_{i+1} &= w_i - DW1_{i+1} - DW2_{i+1} - \cdots - DWn_{i+1}, \\ x_{i+1} &= x_i - DX1_{i+1} - DX2_{i+1} - \cdots - DXn_{i+1}, \\ y_{i+1} &= y_i - DY1_{i+1} - DY2_{i+1} - \cdots - DYn_{i+1}, \\ z_{i+1} &= z_i - DZ1_{i+1} - DZ2_{i+1} - \cdots - DZn_{i+1}, \end{aligned}$$

kde členy vyjádříme rekurentně ve tvaru

$$\begin{aligned} DW1_{i+1} &= -h(-10^{12}y_{i+1} - 10^9w_{i+1}), \\ DW2_{i+1} &= -\frac{h}{2}(-10^{12}DY1_{i+1} - 10^9DW1_{i+1}), \\ &\vdots \\ DWn_{i+1} &= -\frac{h}{n}(-10^{12}DY(n-1)_{i+1} - 10^9DW(n-1)_{i+1}), \\ DX1_{i+1} &= -h(y_{i+1} - z_{i+1} - x_{i+1}), \\ DX2_{i+1} &= -\frac{h}{2}(DY1_{i+1} - DZ1_{i+1} - DX1_{i+1}), \\ &\vdots \\ DXn_{i+1} &= -\frac{h}{n}(DY(n-1)_{i+1} - DZ(n-1)_{i+1} - DX(n-1)_{i+1}), \\ DY1_{i+1} &= -h(10^{12}w_{i+1} - 10^{12}x_{i+1}), \\ DY2_{i+1} &= -\frac{h}{2}(10^{12}DW1_{i+1} - 10^{12}DX1_{i+1}), \\ &\vdots \\ DYn_{i+1} &= -\frac{h}{n}(10^{12}DW(n-1)_{i+1} - 10^{12}DX(n-1)_{i+1}), \\ DZ1_{i+1} &= -hx_{i+1}, \\ DZ2_{i+1} &= -\frac{h}{2}DX1_{i+1}, \\ &\vdots \\ DZn_{i+1} &= -\frac{h}{n}DX(n-1)_{i+1}. \end{aligned}$$

Hledáme tedy řešení soustavy rovnic ve tvaru

$$\begin{aligned} f_1(w_{i+1}, x_{i+1}, y_{i+1}, z_{i+1}) &= -w_{i+1} + w_i - DW1_{i+1} - DW2_{i+1} - \cdots - DWn_{i+1} = 0, \\ f_2(w_{i+1}, x_{i+1}, y_{i+1}, z_{i+1}) &= -x_{i+1} + x_i - DX1_{i+1} - DX2_{i+1} - \cdots - DXn_{i+1} = 0, \\ f_3(w_{i+1}, x_{i+1}, y_{i+1}, z_{i+1}) &= -y_{i+1} + y_i - DY1_{i+1} - DY2_{i+1} - \cdots - DYn_{i+1} = 0, \\ f_4(w_{i+1}, x_{i+1}, y_{i+1}, z_{i+1}) &= -z_{i+1} + z_i - DZ1_{i+1} - DZ2_{i+1} - \cdots - DZn_{i+1} = 0. \end{aligned} \tag{A.16}$$

Numerický výpočet $w_{i+1}, x_{i+1}, y_{i+1}, z_{i+1}$ provedeme pomocí Newtonovy iterace

$$\begin{aligned} w_{i+1,j+1} &= w_{i+1,j} + a_{i+1,j}, \\ x_{i+1,j+1} &= x_{i+1,j} + b_{i+1,j}, \\ y_{i+1,j+1} &= y_{i+1,j} + c_{i+1,j}, \\ z_{i+1,j+1} &= z_{i+1,j} + d_{i+1,j}. \end{aligned}$$

Koeficienty $a_{i+1,j}, b_{i+1,j}, c_{i+1,j}, d_{i+1,j}$ počítáme ze soustavy

$$\begin{pmatrix} f_{1w} & f_{1x} & f_{1y} & f_{1z} \\ f_{2w} & f_{2x} & f_{2y} & f_{2z} \\ f_{3w} & f_{3x} & f_{3y} & f_{3z} \\ f_{4w} & f_{4x} & f_{4y} & f_{4z} \end{pmatrix} \begin{pmatrix} a_{i+1,j} \\ b_{i+1,j} \\ c_{i+1,j} \\ d_{i+1,j} \end{pmatrix} = \begin{pmatrix} -f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) \\ -f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) \\ -f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) \\ -f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) \end{pmatrix},$$

kde parciální derivace jsou ve tvaru

$$\begin{aligned}
f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -w_{i+1,j} + w_i - DW1_{i+1,j} - DW2_{i+1,j} - \dots - DWn_{i+1,j}, \\
f_{1w}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_1(w_{i+1,j}+h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) - f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{1x}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_1(w_{i+1,j}, x_{i+1,j}+h_N, y_{i+1,j}, z_{i+1,j}) - f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{1y}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}+h_N, z_{i+1,j}) - f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{1z}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}+h_N) - f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_1(w_{i+1,j} + h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -(w_{i+1,j} + h_N) + w_i - DW1_{Nw, i+1,j} - \dots - DWn_{Nw, i+1,j}, \\
f_1(w_{i+1,j}, x_{i+1,j} + h_N, y_{i+1,j}, z_{i+1,j}) &= -w_{i+1,j} + w_i - DW1_{Nx, i+1,j} - \dots - DWn_{Nx, i+1,j}, \\
f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j} + h_N, z_{i+1,j}) &= -w_{i+1,j} + w_i - DW1_{Ny, i+1,j} - \dots - DWn_{Ny, i+1,j}, \\
f_1(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j} + h_N) &= -w_{i+1,j} + w_i - DW1_{Nz, i+1,j} - \dots - DWn_{Nz, i+1,j},
\end{aligned}$$

$$\begin{aligned}
f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -x_{i+1,j} + x_i - DX1_{i+1,j} - DX2_{i+1,j} - \dots - DXn_{i+1,j}, \\
f_{2w}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_2(w_{i+1,j}+h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) - f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{2x}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_2(w_{i+1,j}, x_{i+1,j}+h_N, y_{i+1,j}, z_{i+1,j}) - f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{2y}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}+h_N, z_{i+1,j}) - f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{2z}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}+h_N) - f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_2(w_{i+1,j} + h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -x_{i+1,j} + x_i - DX1_{Nw, i+1,j} - \dots - DXn_{Nw, i+1,j}, \\
f_2(w_{i+1,j}, x_{i+1,j} + h_N, y_{i+1,j}, z_{i+1,j}) &= -(x_{i+1,j} + h_N) + x_i - DX1_{Nx, i+1,j} - \dots - DXn_{Nx, i+1,j}, \\
f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j} + h_N, z_{i+1,j}) &= -x_{i+1,j} + x_i - DX1_{Ny, i+1,j} - \dots - DXn_{Ny, i+1,j}, \\
f_2(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j} + h_N) &= -x_{i+1,j} + x_i - DX1_{Nz, i+1,j} - \dots - DXn_{Nz, i+1,j},
\end{aligned}$$

$$\begin{aligned}
f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -y_{i+1,j} + y_i - DY1_{i+1,j} - DY2_{i+1,j} - \dots - DYn_{i+1,j}, \\
f_{3w}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_3(w_{i+1,j}+h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) - f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{3x}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_3(w_{i+1,j}, x_{i+1,j}+h_N, y_{i+1,j}, z_{i+1,j}) - f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{3y}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}+h_N, z_{i+1,j}) - f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{3z}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}+h_N) - f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_3(w_{i+1,j} + h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -y_{i+1,j} + y_i - DY1_{Nw,i+1,j} - \dots - DYn_{Nw,i+1,j}, \\
f_3(w_{i+1,j}, x_{i+1,j} + h_N, y_{i+1,j}, z_{i+1,j}) &= -y_{i+1,j} + y_i - DY1_{Nx,i+1,j} - \dots - DYn_{Nx,i+1,j}, \\
f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j} + h_N, z_{i+1,j}) &= -(y_{i+1,j} + h_N) + y_i - DY1_{Ny,i+1,j} - \dots - DYn_{Ny,i+1,j}, \\
f_3(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j} + h_N) &= -y_{i+1,j} + y_i - DY1_{Nz,i+1,j} - \dots - DYn_{Nz,i+1,j},
\end{aligned}$$

$$\begin{aligned}
f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{i+1,j} - DZ2_{i+1,j} - \dots - DZn_{i+1,j}, \\
f_{4w}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_4(w_{i+1,j}+h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) - f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{4x}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_4(w_{i+1,j}, x_{i+1,j}+h_N, y_{i+1,j}, z_{i+1,j}) - f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{4y}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}+h_N, z_{i+1,j}) - f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_{4z}(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= \frac{f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}+h_N) - f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j})}{h_N}, \\
f_4(w_{i+1,j} + h_N, x_{i+1,j}, y_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{Nw,i+1,j} - \dots - DZn_{Nw,i+1,j}, \\
f_4(w_{i+1,j}, x_{i+1,j} + h_N, y_{i+1,j}, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{Nx,i+1,j} - \dots - DZn_{Nx,i+1,j}, \\
f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j} + h_N, z_{i+1,j}) &= -z_{i+1,j} + z_i - DZ1_{Ny,i+1,j} - \dots - DZn_{Ny,i+1,j}, \\
f_4(w_{i+1,j}, x_{i+1,j}, y_{i+1,j}, z_{i+1,j} + h_N) &= -(z_{i+1,j} + h_N) + z_i - DZ1_{Nz,i+1,j} - \dots - DZn_{Nz,i+1,j},
\end{aligned}$$

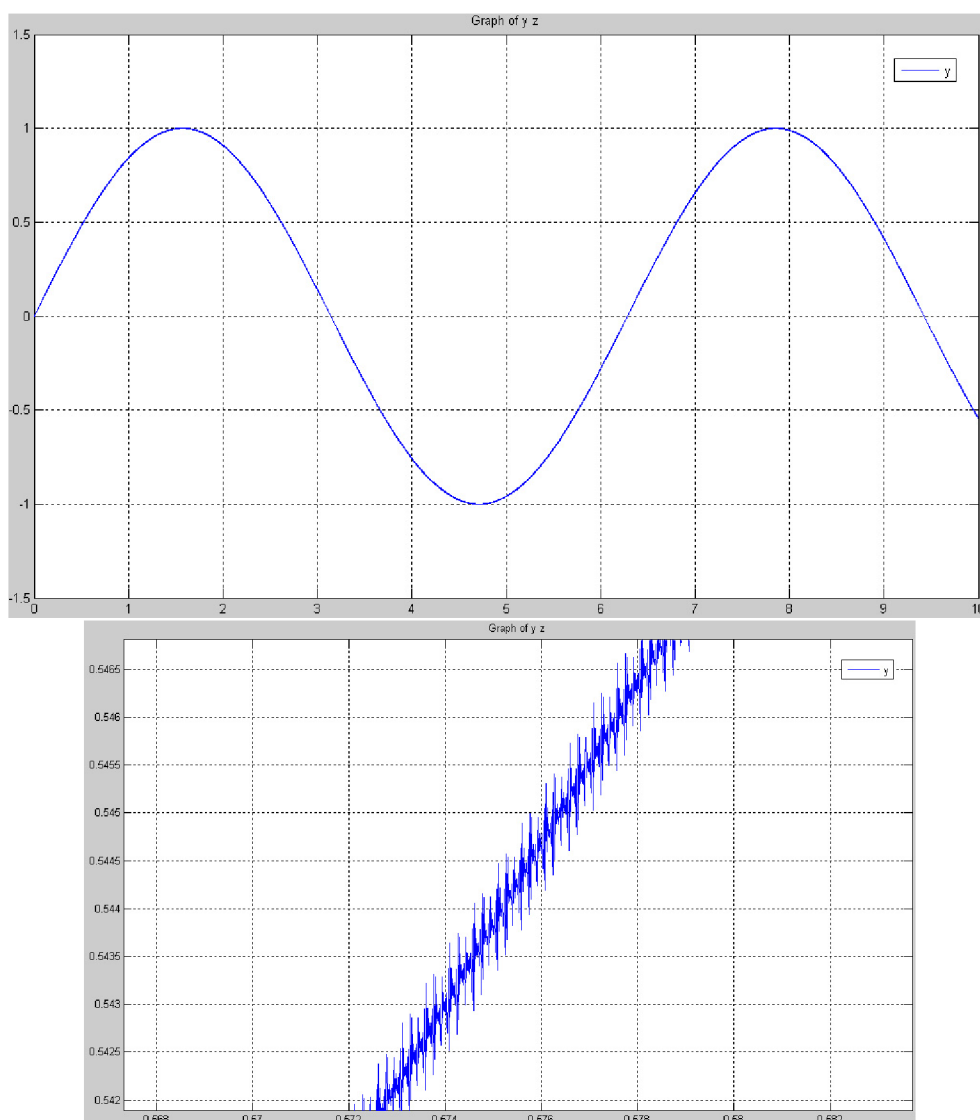
nové členy $DW(l)_{Nw,i+1,j}$, $DX(l)_{Nw,i+1,j}$, $DY(l)_{Nw,i+1,j}$, $DZ(l)_{Nw,i+1,j}$, $DW(l)_{Nx,i+1,j}$, $DX(l)_{Nx,i+1,j}$, $DY(l)_{Nx,i+1,j}$, $DZ(l)_{Nx,i+1,j}$, $DW(l)_{Ny,i+1,j}$, $DX(l)_{Ny,i+1,j}$, $DY(l)_{Ny,i+1,j}$, $DZ(l)_{Ny,i+1,j}$, $DW(l)_{Nz,i+1,j}$, $DX(l)_{Nz,i+1,j}$, $DY(l)_{Nz,i+1,j}$, $DZ(l)_{Nz,i+1,j}$ pro $l = 1, 2, \dots, n$ vyjádříme ve tvaru

$$\begin{aligned}
DW1_{Nw,i+1,j} &= -h(-10^{12}y_{i+1,j} - 10^9(w_{i+1,j} + h_N)), \\
DW2_{Nw,i+1,j} &= -\frac{h}{2}(-10^{12}DY1_{Nw,i+1,j} - 10^9DW1_{Nw,i+1,j}), \\
&\vdots \\
DWn_{Nw,i+1,j} &= -\frac{h}{n}(-10^{12}DY(n-1)_{Nw,i+1,j} - 10^9DW(n-1)_{Nw,i+1,j}), \\
\\
DW1_{Nx,i+1,j} &= -h(-10^{12}y_{i+1,j} - 10^9w_{i+1,j}), \\
DW2_{Nx,i+1,j} &= -\frac{h}{2}(-10^{12}DY1_{Nx,i+1,j} - 10^9DW1_{Nx,i+1,j}), \\
&\vdots \\
DWn_{Nx,i+1,j} &= -\frac{h}{n}(-10^{12}DY(n-1)_{Nx,i+1,j} - 10^9DW(n-1)_{Nx,i+1,j}), \\
\\
DW1_{Ny,i+1,j} &= -h(-10^{12}(y_{i+1,j} + h_N) - 10^9w_{i+1,j}), \\
DW2_{Ny,i+1,j} &= -\frac{h}{2}(-10^{12}DY1_{Ny,i+1,j} - 10^9DW1_{Ny,i+1,j}), \\
&\vdots \\
DWn_{Ny,i+1,j} &= -\frac{h}{n}(-10^{12}DY(n-1)_{Ny,i+1,j} - 10^9DW(n-1)_{Ny,i+1,j}), \\
\\
DW1_{Nz,i+1,j} &= -h(-10^{12}y_{i+1,j} - 10^9w_{i+1,j}), \\
DW2_{Nz,i+1,j} &= -\frac{h}{2}(-10^{12}DY1_{Nz,i+1,j} - 10^9DW1_{Nz,i+1,j}), \\
&\vdots \\
DWn_{Nz,i+1,j} &= -\frac{h}{n}(-10^{12}DY(n-1)_{Nz,i+1,j} - 10^9DW(n-1)_{Nz,i+1,j}), \\
\\
DX1_{Nw,i+1,j} &= -h(y_{i+1,j} - z_{i+1,j} - x_{i+1,j}), \\
DX2_{Nw,i+1,j} &= -\frac{h}{2}(DY1_{Nw,i+1,j} - DZ1_{Nw,i+1,j} - DX1_{Nw,i+1,j}), \\
&\vdots \\
DXn_{Nw,i+1,j} &= -\frac{h}{n}(DY(n-1)_{Nw,i+1,j} - DZ(n-1)_{Nw,i+1,j} - DX(n-1)_{Nw,i+1,j}), \\
\\
DX1_{Nx,i+1,j} &= -h(y_{i+1,j} - z_{i+1,j} - (x_{i+1,j} + h_N)), \\
DX2_{Nx,i+1,j} &= -\frac{h}{2}(DY1_{Nx,i+1,j} - DZ1_{Nx,i+1,j} - DX1_{Nx,i+1,j}), \\
&\vdots \\
DXn_{Nx,i+1,j} &= -\frac{h}{n}(DY(n-1)_{Nx,i+1,j} - DZ(n-1)_{Nx,i+1,j} - DX(n-1)_{Nx,i+1,j}), \\
\\
DX1_{Ny,i+1,j} &= -h((y_{i+1,j} + h_N) - z_{i+1,j} - x_{i+1,j}), \\
DX2_{Ny,i+1,j} &= -\frac{h}{2}(DY1_{Ny,i+1,j} - DZ1_{Ny,i+1,j} - DX1_{Ny,i+1,j}), \\
&\vdots \\
DXn_{Ny,i+1,j} &= -\frac{h}{n}(DY(n-1)_{Ny,i+1,j} - DZ(n-1)_{Ny,i+1,j} - DX(n-1)_{Ny,i+1,j}), \\
\\
DX1_{Nz,i+1,j} &= -h(y_{i+1,j} - (z_{i+1,j} + h_N) - x_{i+1,j}), \\
DX2_{Nz,i+1,j} &= -\frac{h}{2}(DY1_{Nz,i+1,j} - DZ1_{Nz,i+1,j} - DX1_{Nz,i+1,j}), \\
&\vdots \\
DXn_{Nz,i+1,j} &= -\frac{h}{n}(DY(n-1)_{Nz,i+1,j} - DZ(n-1)_{Nz,i+1,j} - DX(n-1)_{Nz,i+1,j}),
\end{aligned}$$

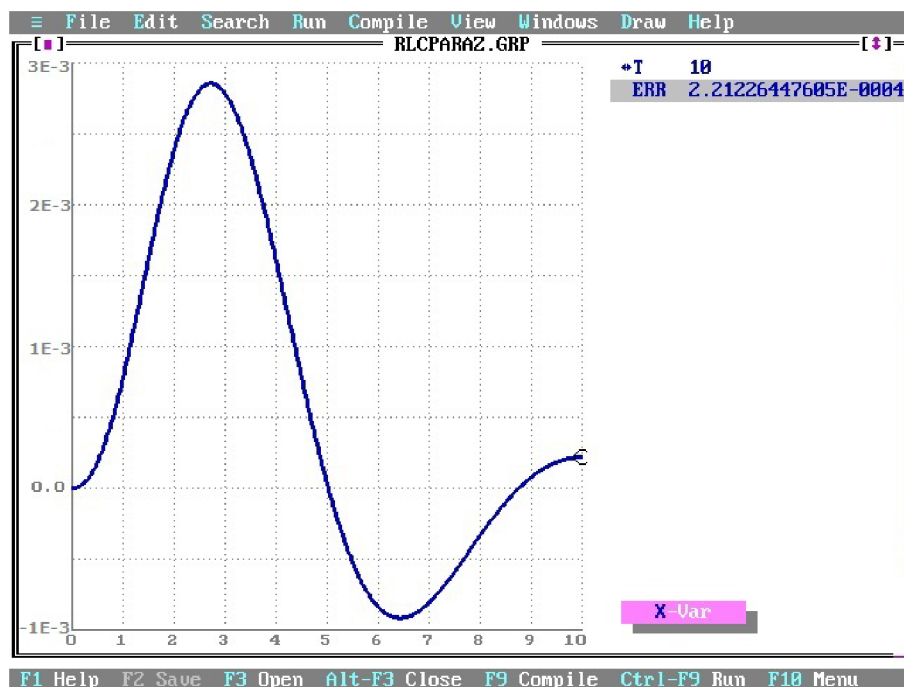
$$\begin{aligned}
DY1_{Nw,i+1,j} &= -h(10^{12}(w_{i+1,j} + h_N) - 10^{12}x_{i+1,j}), \\
DY2_{Nw,i+1,j} &= -\frac{h}{2}(10^{12}DW1_{Nw,i+1,j} - 10^{12}DX1_{Nw,i+1,j}), \\
&\vdots \\
DYn_{Nw,i+1,j} &= -\frac{h}{n}(10^{12}DW(n-1)_{Nw,i+1,j} - 10^{12}DX(n-1)_{Nw,i+1,j}), \\
\\
DY1_{Nx,i+1,j} &= -h(10^{12}w_{i+1,j} - 10^{12}(x_{i+1,j} + h_N)), \\
DY2_{Nx,i+1,j} &= -\frac{h}{2}(10^{12}DW1_{Nx,i+1,j} - 10^{12}DX1_{Nx,i+1,j}), \\
&\vdots \\
DYn_{Nx,i+1,j} &= -\frac{h}{n}(10^{12}DW(n-1)_{Nx,i+1,j} - 10^{12}DX(n-1)_{Nx,i+1,j}), \\
\\
DY1_{Ny,i+1,j} &= -h(10^{12}w_{i+1,j} - 10^{12}x_{i+1,j}), \\
DY2_{Ny,i+1,j} &= -\frac{h}{2}(10^{12}DW1_{Ny,i+1,j} - 10^{12}DX1_{Ny,i+1,j}), \\
&\vdots \\
DYn_{Ny,i+1,j} &= -\frac{h}{n}(10^{12}DW(n-1)_{Ny,i+1,j} - 10^{12}DX(n-1)_{Ny,i+1,j}), \\
\\
DY1_{Nz,i+1,j} &= -h(10^{12}w_{i+1,j} - 10^{12}x_{i+1,j}), \\
DY2_{Nz,i+1,j} &= -\frac{h}{2}(10^{12}DW1_{Nz,i+1,j} - 10^{12}DX1_{Nz,i+1,j}), \\
&\vdots \\
DYn_{Nz,i+1,j} &= -\frac{h}{n}(10^{12}DW(n-1)_{Nz,i+1,j} - 10^{12}DX(n-1)_{Nz,i+1,j}), \\
\\
DZ1_{Nw,i+1,j} &= -hx_{i+1,j}, \\
DZ2_{Nw,i+1,j} &= -\frac{h}{2}DX1_{Nw,i+1,j}, \\
&\vdots \\
DZn_{Nw,i+1,j} &= -\frac{h}{n}DX(n-1)_{Nw,i+1,j}, \\
\\
DZ1_{Nx,i+1,j} &= -h(x_{i+1,j} + h_N), \\
DZ2_{Nx,i+1,j} &= -\frac{h}{2}DX1_{Nx,i+1,j}, \\
&\vdots \\
DZn_{Nx,i+1,j} &= -\frac{h}{n}DX(n-1)_{Nx,i+1,j}, \\
\\
DZ1_{Ny,i+1,j} &= -hx_{i+1,j}, \\
DZ2_{Ny,i+1,j} &= -\frac{h}{2}DX1_{Ny,i+1,j}, \\
&\vdots \\
DZn_{Ny,i+1,j} &= -\frac{h}{n}DX(n-1)_{Ny,i+1,j}, \\
\\
DZ1_{Nz,i+1,j} &= -hx_{i+1,j}, \\
DZ2_{Nz,i+1,j} &= -\frac{h}{2}DX1_{Nz,i+1,j}, \\
&\vdots \\
DZn_{Nz,i+1,j} &= -\frac{h}{n}DX(n-1)_{Nz,i+1,j}.
\end{aligned}$$

Příloha B

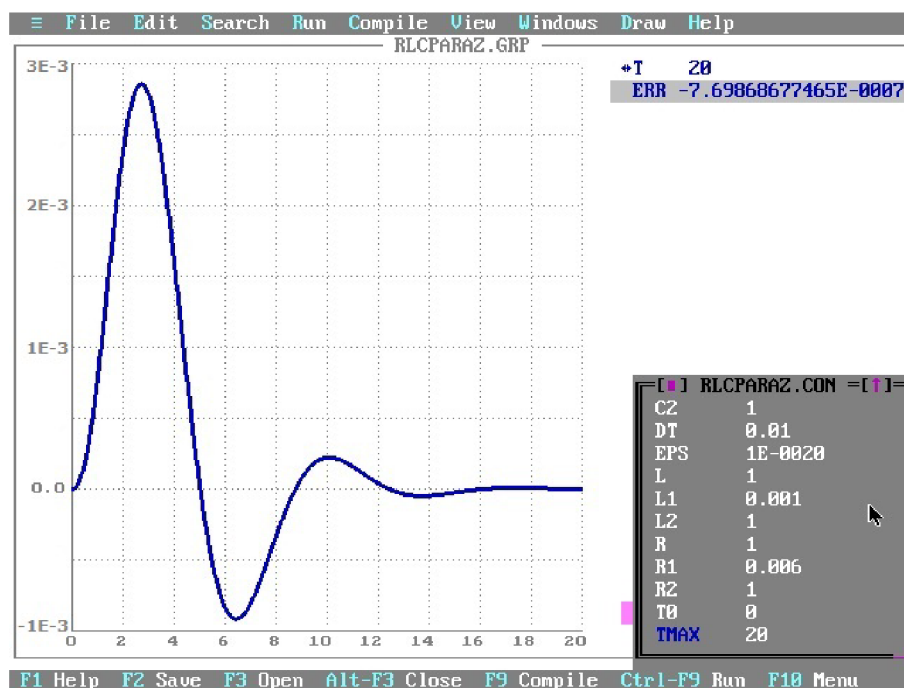
Obrázky a grafy



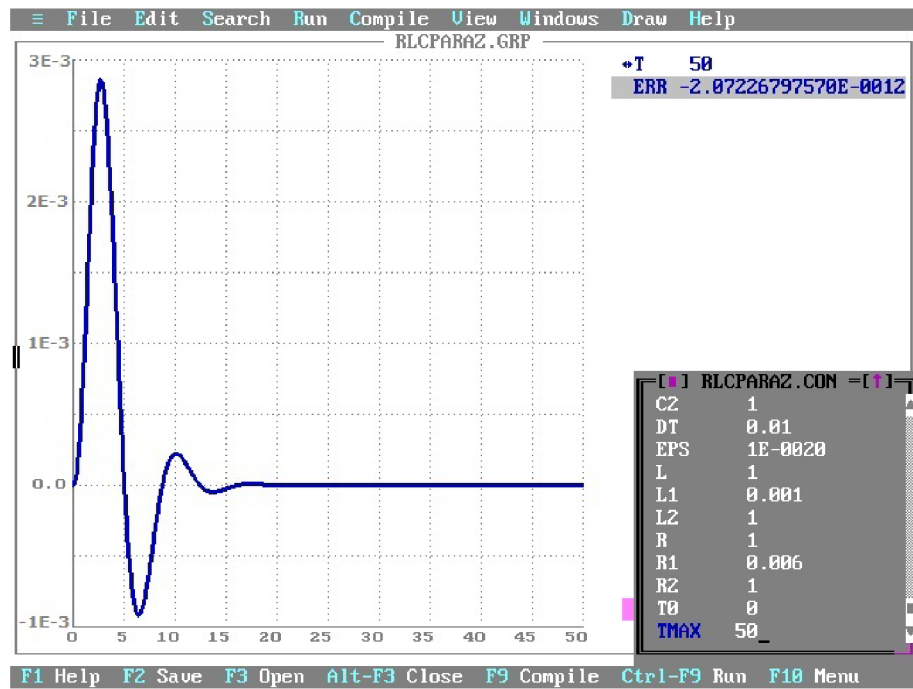
Obrázek B.1: RC elektrický obvod, napětí u_C - MATLAB (ODE45): $a = 10^5, \omega = 1$



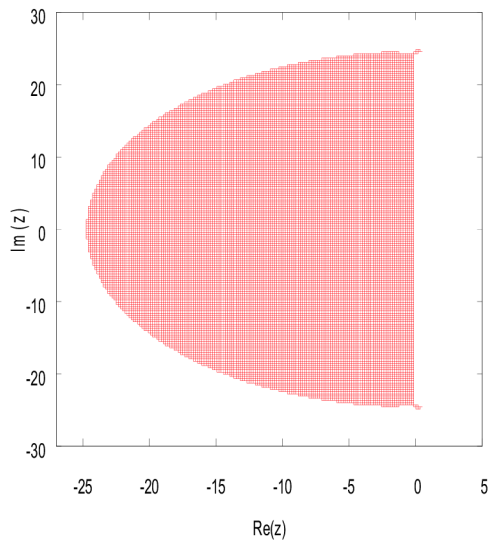
Obrázek B.2: RLC elektrický obvod - TKSL/386: $ERR = u_C - u_2$, $TMAX = 10$



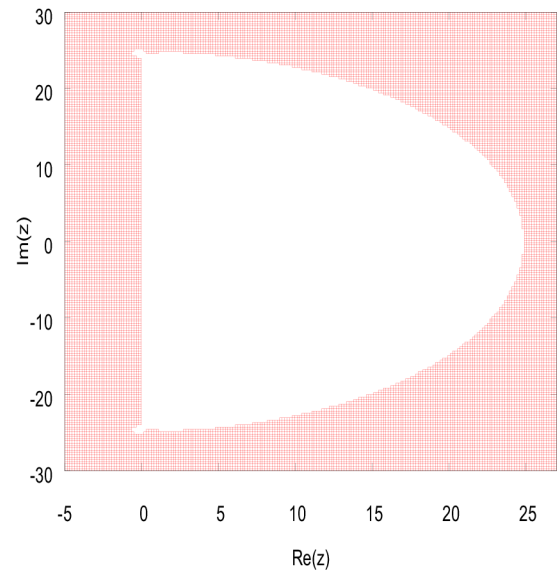
Obrázek B.3: RLC elektrický obvod - TKSL/386: $ERR = u_C - u_2$, $TMAX = 20$



Obrázek B.4: RLC elektrický obvod - TKSL/386: $ERR = u_C - u_2$, $TMAX = 50$



Obrázek B.5: Oblast stability explicitní Taylorovy řady $ORD = 63$



Obrázek B.6: Oblast stability implicitní Taylorovy řady $ORD = 63$