Numerické řešení diferenciálních rovnic s využitím diferenciální transformace

Lukáš Hurtík

Bakalářská práce
2019

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky

Univerzita Tomáše Bati ve Zlíně
Fakulta aplikované informatiky
akademický rok: 2018/2019

ZADÁNÍ BAKALÁŘSKÉ PRÁCE
(PROJEKTU, UMĚLECKÉHO DÍLA, UMĚLECKÉHO VÝKONU)

Jméno a příjmení:	**Lukáš Hurtík**
Osobní číslo:	**A16059**
Studijní program:	**B3902 Inženýrská informatika**
Studijní obor:	**Informační a řídicí technologie**
Forma studia:	**prezenční**
Téma práce:	**Numerické řešení diferenciálních rovnic s využitím diferenciální transformace**
Téma anglicky:	**The Numerical Solving of Differential Equations Using Differential Transformation**

Zásady pro vypracování:

1. **Vypracujte literární rešerši na téma základní metody numerického řešení diferenciálních rovnic a jejich implementace v systému Matlab a na téma diferenciální transformace.**
2. **Ilustrujte metodu řešení obyčejných diferenciálních rovnic pomocí diferenciální transformace v řešených příkladech.**
3. **Navrhněte implementaci diferenciální transformace do programovacího prostředí systému Matlab.**
4. **Uvedený návrh realizujte naprogramováním funkce, kde na vstupu bude počáteční úloha pro lineární obyčejné diferenciální rovnice a požadovaná přesnost a výstupem bude přibližné řešení počáteční úlohy splňující zadanou přesnost.**
5. **Naprogramujte funkci pro řešení některých typů nelineárních obyčejných diferenciálních rovnic pomocí diferenciální transformace.**
6. **Porovnejte na několika příkladech přesnost a výpočetní rychlost získaných výsledků s výsledky jiných metod standardně dostupných v Matlabu (které budou prostudovány na základě bodu 1).**

Rozsah bakalářské práce:

Rozsah příloh:

Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam odborné literatury:

1. **Horová, I. a Zelinka, J.: Numerické metody. 2. rozš. vyd. Praha: Masarykova univerzita, 2008 dotisk. ISBN 80-210-3317-7.**

2. **Kubíček. M. Dubcová. M. Janovská. D.: Numerické metody a algoritmy. 2. vyd. Vysoká škola chemicko-technologická v Praze, Praha, 2005. ISBN 80-7080-558-7.**

3. **Hairer, E., S. P N?rsett a Gerhard Wanner. Solving ordinary differential equations I: nonstiff problems. 2nd rev. ed. London: Springer, c2009. ISBN 978-3-540-56670-0.**

4. **Butcher J.C., Numerical Methods for Ordinary Differential Equations, John Wiley & Sons,Chichester, 2003, ISBN 0-471-96758-0.**

5. **Rebenda, J.: An application of bell polynomials in numerical solving of nonlinear differential equations, 17th Conference on Applied Mathematics, APLIMAT 2018 - Proceedings 2018-February, pp. 891-900. ISBN 978-802274765-3.**

Vedoucí bakalářské práce: **Mgr. Zuzana Pátíková, Ph.D.**
Ústav matematiky

Datum zadání bakalářské práce: **21. prosince 2018**

Termín odevzdání bakalářské práce: **15. května 2019**

Ve Zlíně dne 21. prosince 2018

UNIVERZITA TOMÁŠE BATI VE ZLÍNĚ
Fakulta aplikované informatiky
L.S.

doc. Mgr. Milan Adámek, Ph.D.
děkan

prof. Ing. Vladimír Vašek, CSc.
ředitel ústavu

Jméno, příjmení: LUKÁŠ HURTÍK

Název bakalářské práce: NUMERICKÉ ŘEŠENÍ DIFERENCIÁLNÍCH ROVNIC S VYUŽITÍM DIFERENCIÁLNÍ TRANSFORMACE

Prohlašuji, že

- beru na vědomí, že odevzdáním bakalářské práce souhlasím se zveřejněním své práce podle zákona č. 111/1998 Sb. o vysokých školách a o změně a doplnění dalších zákonů (zákon o vysokých školách), ve znění pozdějších právních předpisů, bez ohledu na výsledek obhajoby;
- beru na vědomí, že bakalářská práce bude uložena v elektronické podobě v univerzitním informačním systému dostupná k prezenčnímu nahlédnutí, že jeden výtisk bakalářské práce bude uložen v příruční knihovně Fakulty aplikované informatiky Univerzity Tomáše Bati ve Zlíně a jeden výtisk bude uložen u vedoucího práce;
- byl/a jsem seznámen/a s tím, že na moji bakalářskou práci se plně vztahuje zákon č. 121/2000 Sb. o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon) ve znění pozdějších právních předpisů, zejm. § 35 odst. 3;
- beru na vědomí, že podle § 60 odst. 1 autorského zákona má UTB ve Zlíně právo na uzavření licenční smlouvy o užití školního díla v rozsahu § 12 odst. 4 autorského zákona;
- beru na vědomí, že podle § 60 odst. 2 a 3 autorského zákona mohu užít své dílo – diplomovou/bakalářskou práci nebo poskytnout licenci k jejímu využití jen připouští-li tak licenční smlouva uzavřená mezi mnou a Univerzitou Tomáše Bati ve Zlíně s tím, že vyrovnání případného přiměřeného příspěvku na úhradu nákladů, které byly Univerzitou Tomáše Bati ve Zlíně na vytvoření díla vynaloženy (až do jejich skutečné výše) bude rovněž předmětem této licenční smlouvy;
- beru na vědomí, že pokud bylo k vypracování bakalářské práce využito softwaru poskytnutého Univerzitou Tomáše Bati ve Zlíně nebo jinými subjekty pouze ke studijním a výzkumným účelům (tedy pouze k nekomerčnímu využití), nelze výsledky bakalářské práce využít ke komerčním účelům;
- beru na vědomí, že pokud je výstupem bakalářské práce jakýkoliv softwarový produkt, považují se za součást práce rovněž i zdrojové kódy, popř. soubory, ze kterých se projekt skládá. Neodevzdání této součásti může být důvodem k neobhájení práce.

Prohlašuji,

- že jsem na bakalářské práci pracoval samostatně a použitou literaturu jsem citoval. V případě publikace výsledků budu uveden jako spoluautor.
- že odevzdaná verze bakalářské práce a verze elektronická nahraná do IS/STAG jsou totožné.

Ve Zlíně, dne 13.5.2019

Hurtík
....
podpis diplomanta

ABSTRAKT

Cieľom tejto práce je priblížiť čitateľom numerické metódy riešenia diferenciálnych rovníc a zlepšiť im povedomie o metóde diferenciálnej transformácie. Táto práca tiež oboznamuje čitateľov o tom ako implementovať danú metódu v systéme MATLAB. Okrem iného obsahuje príklady riešené pomocou danej metódy a tiež porovnanie implementácie metódy diferenciálnej transformácie a metódy Runge-Kutta 4. rádu v systéme MATLAB.

Kľúčové slová: matlab, numerické metódy, diferenciálna transformácia

ABSTRACT

The primary focus of this thesis is to introduce numerical methods of solving differential equations to readers and improve their awareness of the differential transform method. This thesis also informs readers about how to implement the considered method in MATLAB. Among other issues this thesis includes examples of problems solved by the differential transform method and also comparison of the implementation of the differential transform method and the Runge-Kutt method of the 4th order in MATLAB.

Keywords: matlab, numerical methods, differential transform

Rád by som poďakoval rodine, priateľom, vyučujúcim, ostatným pracovníkom fakulty a tiež všetkým ľuďom, ktorí ma v priebehu štúdia podporovali. Avšak osobitné poďakovanie patrí pani Mgr. Zuzane Pátíkovej, Ph.D. za odborné vedenie, trpezlivosť, cenné rady a pripomienky a čas, ktorý mi venovala počas vypracovávania bakalárskej práce.

Prehlasujem, že odovzdaná verzia bakalárskej práce a verzia elektronická nahraná do IS/STAG sú totožné.

„Robiť politiku je rovnako ľahké, ako riešiť diferenciálnu rovnicu.

Keď vám to dobre vyjde dostanete funkciu.“

Ing. Zoltán Kafka

OBSAH

ÚVOD

Neustály rozvoj informačných technológií a ich stále sa rozširujúce spektrum použitia majú za následok, že numerické metódy majú a aj budú mať stále väčší význam v matematických aplikáciách. V súčasnosti je možné vykonávať v krátkom čase až milióny numerických operácií, ktorých náročnosť sa stále zvyšuje. Môže za to zvyšujúca sa rýchlosť výpočtových prístrojov a relatívne malé finančné náklady potrebné k dosiahnutiu výsledku. Vďaka týmto faktorom je možné výpočty, ktoré by ľuďom trvalo vypočítať niekoľko desiatok hodín, vyriešiť pomocou výpočtovej techniky v priebehu niekoľkých sekúnd. Hlavnou témou tejto práce sú numerické metódy riešenia obyčajných diferenciálnych rovníc, ktoré sú základom skoro každej vednej disciplíny a vedú k riešeniu mnohých fyzikálnych, technických, ba dokonca i spoločenských problémov. Pozornosť bude venovaná najmä metóde využívajúcu diferenciálnu transformáciu a jej implementáciu v systéme MATLAB.

Teoretická časť tejto práce má za cieľ objasniť čo vlastne diferenciálne rovnice sú, ako vznikli a ako sa delia. V tejto časti práce sa tiež čitateľ dozvie, čo vlastne numerické metódy sú, aké základné numerické metódy existujú a ako prebieha ich výpočet. Následne sa v práci oboznamujeme s definíciou Taylorovho polynómom a s metódou diferenciálnej transformácie.

V praktickej časti sa nachádzajú riešené príklady pomocou metódy diferenciálnej transformácie. Praktická časť tiež oboznamuje čitateľa so systémom MATLAB. Následne práca pojednáva o algoritme vytvorenom v tomto systéme, s popisom funkcií použitých pri implementácii metódy diferenciálnej transformácie v systéme MATLAB.

I. TEORETICKÁ ČASŤ

1 DIFERENCIÁLNE ROVNICE

Diferenciálna rovnica je matematická rovnica, v ktorej neznámou je funkcia a v ktorej sa vyskytujú derivácie tejto funkcie. Rovnako ako ostatné typy rovníc, môžu aj diferenciálne rovnice tvoriť tzv. sústavy rovníc. V tejto práci je premenná značená t, hľadaná funkcia spl-ňujúca diferenciálnu rovnicu $u(t)$ a jej derivácie od prvého rádu do n-tého rádu postupne $u'(t), u''(t), \dots, u^{(n)}(t)$. V prípade, že nehrozí riziko nepochopenia, sa niekedy argument t pre funkciu a jej derivácie vynecháva. Diferenciálna rovnica môže byť vyjadrená v explicit-nom alebo implicitnom tvare, pričom explicitný tvar má podobu:

$$u^{(n)} = f\left(t, u, u', u'', \dots, u^{(n-1)}\right). \tag{1}$$

Vyjadrenie v explicitnom tvare znamená, že z rovnice je možné vyjadriť najvyššiu deriváciu $u^{(n)}$ ako funkciu premennej t a všetkých nižších derivácií veličiny u. Ak rovnicu nemožno zapísať v tvare (1), hovoríme že je diferenciálna rovnica vyjadrená v implicitnom tvare, ktorý má podobu:

$$f\left(t, u, u', u'', \dots, u^{(n)}\right) = 0. \tag{2}$$

Riešenie rovnice (1), resp. (2) nazývame:

- **obecné riešenie**, ak ho možno vyjadriť v tvare:

  $$\Phi(t, u, C_1, C_2, \dots, C_n) = 0 \text{ alebo } u = \varphi(t, C_1, C_2, \dots, C_n), \tag{3}$$

  kde $C_1, C_2, \dots, C_n$ sú konštanty,
- **partikulárne riešenie**, ak je ho možné získať z obecného riešenia pre konkrétne hod-noty konštánt $C_1, C_2, \dots, C_n$, ktoré vypočítame alebo zvolíme,
- **výnimočné riešenie**, ak ho nie je možné získať z obecného riešenia pre žiadnu voľbu hodnôt $C_1, C_2, \dots, C_n$.

Pri jednoduchých DR je možné vypočítať partikulárne riešenie analyticky, avšak pri nároč-nejších DR je analytické riešenie zdĺhavé, alebo nedosiahnuteľné štandardnými technikami, práve vtedy používame pre výpočet numerické metódy.

Dôležitou vlastnosťou diferenciálnej rovnice je jej rád. Rád diferenciálnej rovnice je určený najvyššou deriváciou, ktorú daná rovnica obsahuje. [4] [12] [13]

1.1 Rozdelenie diferenciálnych rovníc

Základné rozdelenie DR vychádza z typov derivácií použitých v rovnici. Podľa najvyššieho rádu sa DR tiež delia na diferenciálne rovnice 1. rádu a diferenciálne rovnice vyšších rádov.

- **Obyčajné diferenciálne rovnice** sú rovnice, v ktorých neznámou je funkcia jednej nezávislej premennej a v ktorých sa vyskytuje aspoň jedna derivácia tejto funkcie.
- **Parciálne diferenciálne rovnice** sú rovnice, v ktorých sa vyskytujú parciálne derivácie hľadanej funkcie, t. j. derivácie podľa viac premenných. V obecnom tvare pre neznámu funkciu $u(t_1, t_2, t_3 \dots t_n)$ n premenných majú tvar:

$$F\left(t_1, t_2, t_3 \dots t_n, u, \frac{\partial u}{\partial t_1}, \dots, \frac{\partial u}{\partial t_n}, \frac{\partial^2 u}{\partial {t_1}^2}, \frac{\partial^2 u}{\partial t_1 t_2}, \dots, \frac{\partial^2 u}{\partial t_1 t_n}, \frac{\partial^2 u}{\partial {t_2}^2}, \dots, \frac{\partial^k u}{\partial {t_n}^k}, \dots\right) = 0. \quad (4)$$

Oba základné typy diferenciálnych rovníc možno ďalej rozdeliť podľa ich tvaru na ďalšie typy. V obyčajných diferenciálnych rovniciach 1. rádu sa môžeme stretnúť napríklad s týmito typmi rovníc: homogénne, nehomogénne, separovateľné, lineárne, nelineárne, exaktné atď. Parciálne diferenciálne rovnice sa tiež delia na ďalšie typy: 1.rádu, vyšších rádov, kvazilineárne, nelineárne atď.

Diferenciálne rovnice, v ktorých sa hľadaná funkcia vyskytuje iba lineárne, pričom sa nikde nevyskytujú ani súčiny hľadanej funkcie s jej deriváciami, ani súčiny derivácií tejto funkcie, označujeme ako lineárne diferenciálne rovnice. Ak jedna z uvedených podmienok nie je splnená, hovoríme o nelineárnych diferenciálnych rovniciach.

Ku každému zadaniu rovnice sa v teórii DR pristupuje odlišným spôsobom. Je preto vhodné poznať všetky používané matematické metódy, ktoré boli pre riešenie DR vytvorené. [12]

1.2 História diferenciálnych rovníc

Samotný názov „diferenciálne rovnice“ pochádza od nemeckého matematika G. W. Leibniza (1646 - 1716), ktorý sa počas svojho života zaoberal riešením diferenciálnych rovníc so separovateľnými premennými. Spoločne s ďalšími matematikmi, bratmi Jacobom Bernoullim (1654 - 1705) a Johannom Bernoullim (1667 - 1748) pôvodom zo Švajčiarska sa pokúšali previesť diferenciálne rovnice 1. rádu na rovnice so separovateľnými premennými a následne ich riešiť.

Ďalšia etapa rozvoja diferenciálnych rovníc sa začala v období 18. storočia, kde sa o rozvoj tejto problematiky podstatnou mierou zaslúžili najmä L. Euler (1707 - 1783), J. d´Alembert

(1717 - 1783) a J. L. Lagrange (1736 - 1813). Švajčiarsky matematik Euler už v roku 1740 použil metódu variácie konštánt, ktorá sa však neprávom nazýva Lagrangeova metóda. Až v priebehu rokov 1766 až 1777 Lagrange pravdepodobne rozpracoval túto metódu a použil ju na riešenie DR n-tého rádu s pravou stranou. Euler je tiež považovaný za zakladateľa teórie diferenciálnych rovníc.

Začiatkom 19. storočia rozvoj matematickej analýzy ovplyvnil aj rozvoj teórie diferenciálnych rovníc. V tomto období sa začali študovať otázky existencie a jednoznačnosti riešení diferenciálnych rovníc. Prvú presnú formuláciu a prvé riešenie tejto úlohy podal francúzsky matematik A. L. Cauchy (1789 - 1857), všeobecnejšiu podmienku sformuloval nemecký matematik R. Lipschitz (1832 - 1903).

Z českých matematikov dosiahol v oblasti diferenciálnych rovníc svetových úspechov matematik Otakar Borůvka (1899 - 1995), ktorý sa zaoberal prevažne teóriou lineárnych diferenciálnych rovníc s premennými koeficientami. [1] [2]

1.3 Počiatočná úloha

V praktických úlohách je často potrebné nájsť také riešenie DR, ktoré splňuje vopred stanovené podmienky. Ak sa podmienky vzťahujú na jeden bod, hovoríme o počiatočných podmienkach, ak sa ale vzťahujú k viac bodom, hovoríme o okrajových podmienkach.

Uvažujme Cauchyho počiatočnú úlohu:

$$u'(t) = f\big(t, u(t)\big), \qquad u(t_0) = u_0, \tag{5}$$

kde funkcia $f\big(t, u(t)\big)$ má na oblasti:

$$\Omega = \{[t, u] \in E_2 : t \in (t_0 - a, t_0 + a), u \in (u_0 - b, u_0 + b)\},$$

kde a, b sú kladné čísla, resp. ∞ , nasledovné vlastnosti:

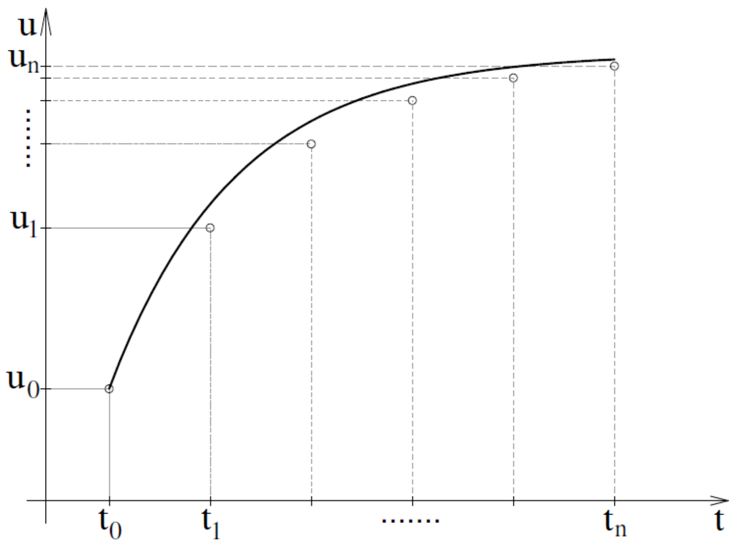
- je **spojitá**,
- je **ohraničená**, tzn. existuje taká kladná konštanta K, že pre každý bod $[t, u] \in \Omega$ platí:
$$|f(t, u)| \leq K,$$
- $\frac{\partial}{\partial u} f(t, u)$ je **ohraničená**, tzn. existuje taká kladná konštanta L, že pre každý bod $[t, u] \in \Omega$ platí:
$$\left|\frac{\partial}{\partial u} f(t, u)\right| \leq L.$$

Potom DR $u'(t) = f\big(t, u(t)\big)$ má práve jedno riešenie $u = f(t)$, ktoré prechádza bodom $[t_0, u_0]$, tzn. spĺňa podmienku $u(t_0) = u_0$ a je riešením aspoň na intervale $J = (t_0 - c, t_0 + c)$, kde $c = min\left\{a, \frac{b}{K}\right\}$.

Ak existuje riešenie takejto úlohy, hovoríme o existencii daného riešenia. Ak existuje práve jedno riešenie takejto úlohy, hovoríme o jednoznačnosti daného riešenia. [5]

2 NUMERICKÉ METÓDY RIEŠENIA DR

Numerické metódy riešenia DR sa používajú v prípade, keď dosiahnutie analytického riešenia DR je náročné alebo nemožné. Spoločným znakom základných numerických metód je, že riešenie nehľadáme ako spojitú funkciu, definovanú na celom skúmanom intervale $\langle a, b \rangle$, ale hodnoty približného riešenia počítame iba v konečnom počte bodov $a = t_0 < t_1 < \cdots < t_n = b$. Tieto body sa nazývajú uzlové body, alebo tiež uzly siete. Množinu $\{t_0, t_1, \dots, t_n\}$ nazývame sieť. Krokom siete v uzle t_i rozumieme rozdiel $h_i = t_{i+1} - t_i$. Približné hodnoty riešenia v uzlových bodoch, vypočítané nejakou numerickou metódou sa zvyknú označovať $u_0, u_1, \dots, u_n$, hodnoty presného riešenia označujeme $u(t_0), u(t_1), \dots, u(t_n)$. Na obrázku môžeme badať presné riešenie DR, ktoré je vykreslené plnou čiarou a približné hodnoty riešenia v uzlových bodoch, vyznačené krúžkami.



Obrázok 1 – Presné a približné riešenie diferenciálnej rovnice

V príklade z obrázka 1 je použitá pravidelná (ekvidištantná) sieť, tzn. krok h medzi jednotlivými uzlami je konštantný. [11]

Numerické metódy môžeme rozdeliť podľa počtu použitých predchádzajúcich krokov k dosiahnutiu novej hodnoty:

- **jednokrokové metódy** k výpočtu novej hodnoty používajú iba jednu predchádzajúcu hodnotu,
- **viackrokové metódy** k výpočtu novej hodnoty používajú k predchádzajúcich krokov.

Hovoríme, že metóda je konvergentná, ak pre ľubovoľnú počiatočnú úlohu (5) platí pre každé $t \in \langle a, b \rangle$:

$$\lim_{\substack{h \to 0 \\ n \to \infty}} u_n = u(t),$$

kde $t = t_0 + nh$.

Pre popis rýchlosti konvergencie metódy sa používa pojem rád metódy. Rád metódy je vlastne prirodzené číslo p, pre ktoré platí že pre malé h je lokálna diskretizačná chyba d_i rádovo veľkosti h^{p+1}. [11]

2.1 Chyby pri numerických výpočtoch

Keďže základom numerických metód je získavanie približných výsledkov, je potrebné vždy vedieť, aký je rozdiel medzi presným riešením danej úlohy a riešením získaným danou numerickou metódou. Okrem chýb ktorých sa môže dopustiť riešiteľ danej numerickej metódy, môžeme chyby základne rozdeliť na:

- **chyby matematického modelu**, ktoré zväčša vznikajú pri nahradení reálnej fyzikálnej situácie matematickým modelom, napr. popis daného fyzikálneho deja diferenciálnou rovnicou,
- **chyby vstupných údajov**, ktoré sú spôsobené nepresným meraním fyzikálnych veličín,
- **chyby numerickej metódy** zväčša vznikajú pri nahradzovaní matematickej úlohy jednoduchšou numerickou úlohou. Najčastejšie sa jedná o náhradu nekonečného procesu konečným procesom, napríklad pri výpočte danej elementárnej funkcie pomocou súčtu niekoľko prvých členov jej nekonečného Taylorovho polynómu. Odhad tejto chyby je veľmi podstatnou súčasťou riešenia každej numerickej úlohy,
- **chyby zaokrúhľovania** vznikajú vtedy, keď pri výpočtoch pracujeme s číslami zaokrúhlenými na určitý počet miest. Týmto spôsobom dochádza ku kumulácii, resp. rušení chýb vo výpočte. Pri veľkom počte operácii je posúdenie ich vplyvu na výsledok náročný. [11]

Absolútnou chybou aproximácie nazývame rozdiel medzi presnou hodnotou čísla $\hat{t}$ a jeho aproximáciou t, ktorá sa zväčša vyjadruje v percentách:

$$E(t) = \hat{t} - t.$$

Relatívnou chybou aproximácie používame v prípade, keď je presná hodnota veľmi veľká alebo naopak veľmi malá. Táto chyba sa tiež vyjadruje v percentách a má tvar:

$$RE(t) = \frac{\hat{t}-t}{t}.$$

Keďže pracujeme na sieti je potrebné poznať rozdiel medzi presným a približným riešením v bodoch siete, t. j. ako veľká je globálna diskretizačná chyba, ktorá má tvar:

$$e_i = u(t_i) - u_i.$$

Pre získanie predstavy o globálnej diskretizačnej chybe býva často veľmi užitočné poznať tzv. lokálnu diskretizačnú chybu danej metódy. Je to chyba, ktorej sa dopustíme v jednom kroku danej metódy za predpokladu, že všetky hodnoty, ktoré sme pri výpočte použili, boli presné. Pre označenie lokálnej diskretizačnej chyby je zaužívané značenie d_i.

Odhadom absolútnej chyby aproximácie t alebo tiež medznej absolútnej chyby nazývame každé nezáporné číslo $ME(t)$, pre ktoré platí:

$$|\hat{t} - t| \leq ME(t), \text{ t. j. } \hat{t} \in \langle t - ME(t), t + ME(t) \rangle.$$

Odhadom relatívnej chyby alebo tiež medznej relatívnej chyby nazývame každé nezáporné číslo $MR(t)$, pre ktoré platí:

$$\frac{|\hat{t}-t|}{|t|} \leq MR(t), t \neq 0.$$

Zväčša sa používa zápis:

$$\hat{t} = t \pm ME(t), \text{ resp. } \hat{t} = t\big(1 \pm MR(t)\big).$$

[6] [11]

Zaokrúhľovanie môžeme parafrázovať ako šírenie chýb pri výpočte. Uvažujme reálne číslo t, ktoré má obecne nekonečné dekadické vyjadrenie, a potom číslo $t^{(d)}$, ktoré má d desatinných miest, je správne zaokrúhlenou hodnotou čísla t, ak platí:

$$\left|t - t^{(d)}\right| \leq \frac{1}{2} 10^{-d}.$$

Napríklad, ak má byť $t^{(1)}$ správne zaokrúhlená hodnota čísla t na jedno desatinné miesto, nemôže sa od t líšiť o viac ako o $\frac{1}{2} 10^{-1}$ čiže o $0{,}05$.

Pri numerických metódach pracujeme so zaokrúhlenými číslami, pričom výsledky početných operácií s týmito číslami sú opäť zaokrúhľované a ďalej sa s nimi pracuje, čím sa zaokrúhľovacie chyby šíria. [11]

2.2 Eulerova metóda

Eulerova metóda patrí medzi najjednoduchšie jednokrokové metódy. Uvažujme DR (5) a pravidelnú sieť $\{t_0, t_1, \dots, t_n\}$ s krokom h. Vo všetkých bodoch siete platí:

$$u'(t_i) = f\big(t_i, u(t_i)\big).$$

Ak deriváciu na ľavej strane tejto rovnice nahradíme diferenciou podľa vzťahu:

$$f'(t_1) = \frac{f(t_1) - f(t_0)}{h} - \frac{h}{2} f''(\xi),$$

kde $\xi \in \langle t_0, t_1 \rangle$, po dosadení dostaneme:

$$\frac{u(t_{i+1}) - u(t_i)}{h} \doteq f\big(t_i, u(t_i)\big).$$

Ak nahradíme $u(t_i)$ približnou hodnotou u_i, môžeme vyjadriť približnú hodnotu $u(t_{i+1})$ ako:

$$u_{i+1} = u_i + h\, f(t_i, u_i). \tag{6}$$

Následne je možné vypočítať približnú hodnotu riešenia v ďalšom uzlovom bode na základe predchádzajúceho, pričom hodnotu riešenia v bode t_0 poznáme z počiatočnej podmienky. [3] [8] [11]

Príklad 2.2.1

Uvažujme DR $u' = t^2 - u$ s počiatočnou podmienkou $u(0) = 1$ a s krokom $h = 0{,}1$ na intervale $\langle 0; 0{,}5 \rangle$. V tomto prípade je $t_0 = 0$, $u_0 = 1$ a $f(t, u) = t^2 - u$. Približné hodnoty riešenia určíme podľa vzťahu (5):

$$u_{i+1} = u_i + 0{,}1\, f({t_i}^2 - u_i),$$

kde $i = 0, \dots, 4$.

Do tabuľky vynesieme približné hodnoty riešenia spolu s presným riešením $u = -e^{-t} + t^2 - 2t + 2$ v prvých piatich uzlových bodoch. Všetky hodnoty v tabuľke sú zaokrúhlené na 4 desatinné miesta. [11]

Tabuľka 1 – Výsledky výpočtu EM

i	t_i	u_i	$u(t_i)$
0	0	1	1
1	0,1	0,9	0,9052
2	0,2	0,811	0,8213
3	0,3	0,7339	0,7492
4	0,4	0,6695	0,6897
5	0,5	0,6186	0,6435

2.3 Modifikácie Eulerovej metódy

Výpočet prebieha podobne ako pri Eulerovej metóde. Najskôr vypočítame pomocné hodnoty k_1 a k_2 s pomocou ktorých následne určíme približnú hodnotu riešenia v nasledujúcom uzlovom bode. Pri prvej modifikácii postupujeme podľa vzťahov:

$$k_1 = f(t_i, u_i),$$

$$k_2 = f\left(t_i + \frac{h}{2}, u_i + \frac{h}{2}k_1\right),$$

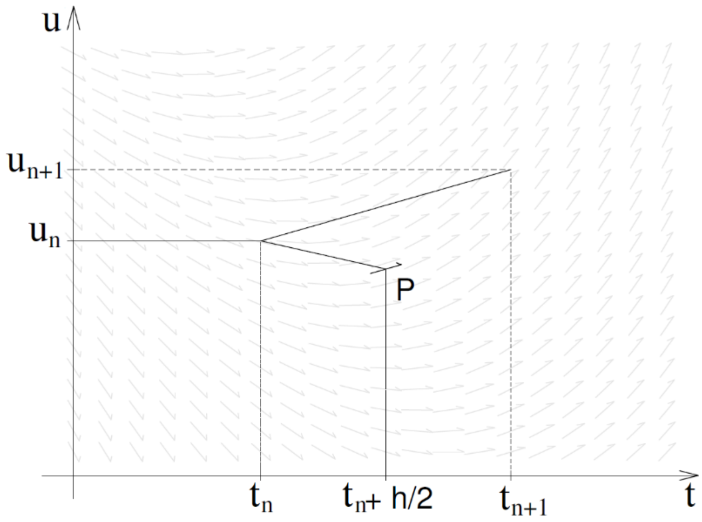
$$u_{i+1} = u_i + hk_2.$$

Pri druhej modifikácii postupujeme podľa vzťahov:

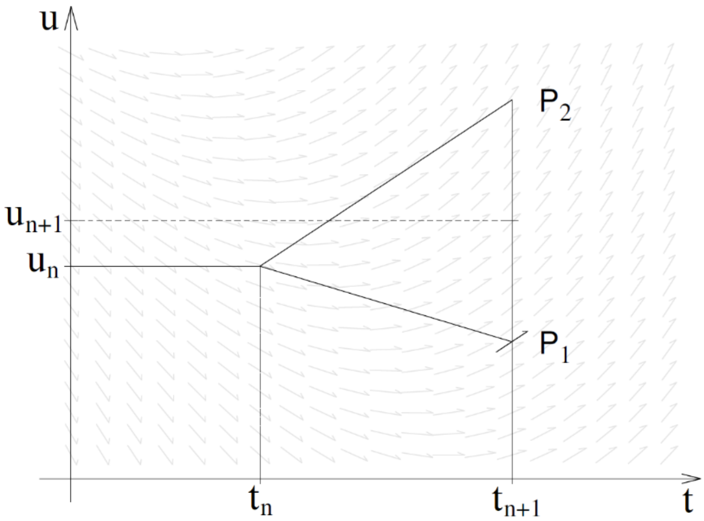
$$k_1 = f(t_i, u_i),$$

$$k_2 = f(t_i + h, u_i + hk_1),$$

$$u_{i+1} = u_i + \frac{h}{2}(k_1 + k_2).$$



Obrázok 2 – Prvá modifikácia Eulerovej metódy



Obrázok 3 – Druhá modifikácia Eulerovej metódy

Prvá a druhá modifikácia Eulerovej metódy sú vlastne jednoduchými príkladmi metódy Runge-Kutta. [3] [11]

2.4 Runge-Kuttove metódy

Runge-Kuttove metódy vychádzajú z vylepšovania ďalších modifikácií Eulerovej metódy a patria k jednej z najdôležitejších skupín jednokrokových metód. Obecný tvar Runge-Kuttovej metódy je:

$$u_{i+1} = u_i + h(\omega_1 k_1 + \cdots + \omega_s k_s),$$

kde:

$$k_1 = f(t_i, u_i),$$

$$k_n = f\left(t_i + \alpha_n h, u_i + h \sum_{j=1}^{n-1} \beta_{nj}\, k_j\right),$$

kde: $n = 2, \dots, s$ a ω_n, α_n a β_{nj} sú konštanty zvolené tak, aby mala daná metóda čo najvyšší rád. [3] [8] [11]

V praxi sa najviac používa Runge-Kuttova metóda 4. rádu, zväčša keď sa hovorí o Runge-Kuttovej metóde, myslí sa tým práve táto metóda, ktorá má tvar:

$$u_{i+1} = u_i + \frac{h}{6}(k_1 + 2k_2 + 3k_3 + k_4),$$

$$k_1 = f(t_i, u_i),$$

$$k_2 = f\left(t_i + \frac{h}{2}, u_i + \frac{h}{2}k_1\right),$$

$$k_3 = f\left(t_i + \frac{h}{2}, u_i + \frac{h}{2}k_2\right),$$

$$k_4 = f(t_i + h, u_i + hk_3). \qquad (7)$$

Príklad 2.4.1

Uvažujme DR $u' = t^2 - u$ s počiatočnou podmienkou $u(0) = 1$ s krokom $h = 0{,}1$ na intervale $\langle 0; 0{,}5 \rangle$. V každom kroku výpočtu musíme vypočítať čísla k_1, k_2, k_3, k_4 a pomocou nich následne približnú hodnotu riešenia v nasledujúcom uzlovom bode podľa vzťahu (7). Potrebné údaje vynesieme do tabuľky. V stĺpcoch označených t a u sa nachádzajú súradnice bodov, v ktorých vyčíslujeme funkciu $f(t, u) = t^2 - u$ pri výpočte k_i. Do tabuľky vynesieme aj hodnoty presného riešenia $u = -e^{-t} + t^2 - 2t + 2$. Všetky hodnoty v tabuľke sú zaokrúhlené na 7 desatinných miest. [11]

Tabuľka 2 – Výsledky výpočtu R-KM

i	t_i	u_i	$u(t_i)$	t	u	
0	0	1	1	0 0,05 0,05 0,1	1 0,95 0,952625 0,9049875	$k_1 = -1$ $k_2 = -0{,}9475$ $k_3 = -0{,}950125$ $k_4 = -0{,}8949875$
1	0,1	0,9051627	0,9051626	0,1 0,15 0,15	0,9051627 0,8604046 0,8632675	$k_1 = -0{,}8951627$ $k_2 = -0{,}8379046$ $k_3 = -0{,}8407675$

				0,2	0,8210860	$k_4 = -0{,}7810860$
2	0,2	0,8212695	0,8212693	0,2 0,25 0,25 0,3	0,8212695 0,7822060 0,7852842 0,7489911	$k_1 = -0{,}7812695$ $k_2 = -0{,}7197060$ $k_3 = -0{,}7227842$ $k_4 = -0{,}6589911$
3	0,3	0,7491822	0,7491818	0,3 0,35 0,35 0,4	0,7491822 0,7162230 0,7194960 0,6894826	$k_1 = -0{,}6591822$ $k_2 = -0{,}5937230$ $k_3 = -0{,}5969960$ $k_4 = -0{,}5294826$
4	0,4	0,6896804	0,6896800	0,4 0,45 0,45 0,5	0,6896804 0,6631964 0,6666456 0,6432659	$k_1 = -0{,}5296804$ $k_2 = -0{,}4606964$ $k_3 = -0{,}4641456$ $k_4 = -0{,}3932659$
5	0,5	0,6434699	0,6434693			

Výsledné údaje môžeme porovnať s hodnotami približného riešenia vypočítanými pomocou Eulerovej metódy v kapitole 2.3. Z porovnania je možné badať, že riešenie získané metódou Runge-Kutta 4. rádu je podstatne presnejšie.

2.5 Viackrokové metódy

U viackrokových metód určujeme približnú hodnotu riešenia v nasledujúcom uzlovom bode siete pomocou niekoľkých prechádzajúcich uzlov. V dôsledku toho, že pri výpočte sa používajú nielen hodnoty približného riešenia, ale taktiež hodnoty pravej strany $f(t,u)$, sa v týchto bodoch používa kvôli jednoduchšiemu zápisu označenie $f_j = (t_j, u_j)$. Lineárna k-kroková metóda ma obecný tvar:

$$u_{i+1} = a_1 u_i + a_2 u_{i-1} + \cdots + a_k u_{i-k+1} + h(b_0 f_{i+1} + b_1 f_i + \cdots + b_k f_{i-k+1}), \qquad (8)$$

kde k je prirodzené číslo a aspoň jedna z konštánt a_k, b_k je rôzna od nuly.

Ak platí, že $b_0 = 0$, metóda (8) sa nazýva explicitná. Ak platí, že $b_0 \neq 0$, metóda (8) sa nazýva implicitná. [11]

2.6 Numerické metódy riešenia DR v systéme MATLAB

V tejto práci sme porovnávali výsledky presného riešenia daných diferenciálnych rovníc, približného riešenia pomocou metódy diferenciálnej transformácie a približného riešenia pomocou funkcie implementovanej v systéme MATLAB. V systéme MATLAB je hneď niekoľko implementovaných funkcií pre numerické riešenie diferenciálnych rovníc, napríklad ode45, ode23, ode113 atď. V našom prípade sme pracovali s funkciou **ode45**, ktorá funguje na princípe metódy Runge-Kutta 4. rádu. Vstupom do tejto funkcie je pravá strana explicitnej diferenciálnej rovnice 1. rádu alebo vektor z pravých strán sústavy takých rovníc, interval na ktorom prebieha výpočet a počiatočné podmienky. V prípade že je daná diferenciálna rovnica vyššieho rádu, je potrebné ju transformovať na systém diferenciálnych rovníc. Uvažujme diferenciálnu rovnicu druhého rádu $u'' + u = 0$. Transformácia na systém diferenciálnych rovníc bude mať potom tvar:

$$u_1{}' = u_2,$$

$$u_2{}' = -u_1.$$

3 DIFERENCIÁLNA TRANSFORMÁCIA

Pre riešenie obyčajných diferenciálnych rovníc, funkčných diferenciálnych rovníc a parciálnych diferenciálnych rovníc sú k dispozícii rôzne presné alebo približné numerické metódy. Väčšina z týchto metód pre zložitejšie úlohy je výpočtovo náročná, pretože fungujú na princípe pokus-omyl, alebo vyžadujú komplikované symbolické výpočty. Metóda diferenciálnej transformácie je jednou z numerických metód pre obyčajné diferenciálne rovnice a funkčné diferenciálne rovnice. Koncepcia diferenciálnej transformačnej metódy bola prvýkrát navrhnutá čínskym matematikom J. K. Zhouom v roku 1986 a bola aplikovaná na riešenie lineárnych a nelineárnych problémov počiatočnej úlohy v analýze elektrických obvodov.

Táto metóda vytvára semi-analytickú numerickú techniku, ktorá využíva Taylorovu radu na riešenie diferenciálnych rovníc vo forme polynómu. Odlišuje sa od metódy Taylorovho radu vysokých rádov, ktorá vyžaduje symbolický výpočet potrebných derivátov dátových funkcií. Výpočet pomocou metódy Taylorovho radu je časovo náročný najmä pre rovnice vysokého rádu. Diferenciálna transformácia je iteratívny postup na získanie analytických riešení diferenciálnych rovníc vo forme Taylorovho radu (Taylorovho polynómu). [16]

3.1 Taylorov polynóm

Uvažujme funkciu f s vlastnosťami:

- je možné určiť funkčnú hodnotu a hodnotu derivácii až do n-tého rádu v určitom bode t_0, pričom musia byť derivácie v tomto bode spojité. V prípade odhadu zvyšku je tento predpoklad potrebný až do rádu (n+1) vrátane,
- je náročné nájsť dostatočne efektívny algoritmus pre výpočet funkčných hodnôt v ostatných bodoch $t \neq t_0$.

Pre výpočet funkčných hodnôt v bodoch v okolí bodu t_0 je potrebné funkciu aproximovať jednoduchšou funkciou, v tomto prípade polynómom n-tého stupňa. Najlepší polynóm, ktorý funkciu f aproximuje v okolí bodu t_0 je polynóm, ktorý má s danou funkciu v okolí bodu t_0 totožné derivácie až do n-tého rádu. Taký polynóm sa nazýva Taylorov polynóm n-tého stupňa funkcie f, ktorý je definovaný vzťahom:

$$T_n(t) = f(t_0) + \frac{f'(t_0)}{1!}(t - t_0) + \frac{f''(t_0)}{2!}(t - t_0)^2 + \cdots + \frac{f^{(n)}(t_0)}{n!}(t - t_0)^n, \qquad (9)$$

kde bod t_0 sa nazýva stred Taylorovho polynómu. V prípade, že je stred polynómu rovný nule, hovoríme o Maclaurinovom polynóme. Pre vybrané typy elementárnych funkcií má Maclaurinov polynóm tvar:

- $e^t = \sum_{n=0}^{\infty} \frac{t^n}{n!} = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots,\ t \in \mathbb{R},$
- $\sin t = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n+1)!} t^{2n+1} = t - \frac{t^3}{3!} + \frac{t^5}{5!} + \cdots,\ t \in \mathbb{R},$
- $\cos t = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} t^{2n} = 1 - \frac{t^2}{2!} + \frac{t^4}{4!} + \cdots,\ t \in \mathbb{R},$
- $\tan t = \sum_{n=1}^{\infty} \frac{B_{2n}(-4)^n(1-4^n)}{(2n)!} t^{2n-1} = t + \frac{t^3}{3} + \frac{2t^5}{15} + \cdots,\ |t| \in \frac{\pi}{2}.$

kde premenná B_k označuje tzv. Bernoulliho čísla. V matematike sú Bernoulliho čísla sled racionálnych čísel, ktoré sa často vyskytujú v teórii čísel. Pre každé k okrem nuly platí, že B_k je záporné ak k je deliteľné 4, inak je B_k kladné. Pre každé nepárne k okrem 1 je B_k rovné nule. [9]

Nech má funkcia u v bode t_0 a nejakom jeho okolí $O(t_0)$ spojité derivácie do rádu $n+1$, vrátane. Potom pre všetky $t \in O(t_0)$ platí:

$$u(t) = T_n(t) + R_{n+1}(t). \tag{10}$$

Za určitých predpokladov o funkcii $u(t)$ v okolí daného bodu možno túto funkciu vyjadriť (rozvinúť) ako mocninovú radu. Toto vyjadrenie funkcie prostredníctvom Taylorovej rady sa označuje ako Taylorov rozvoj. Ak sa jedná o rozvoj v okolí bodu 0, hovoríme o Maclaurinovej rade. V prípade existencie všetkých konečných derivácii funkcie u v bode t_0 možno Taylorovu radu zapísať ako:

$$u(t) = \sum_{k=0}^{\infty} \frac{f^{(k)}(t_0)}{k!} (t - t_0)^k.$$

[7] [9]

3.1.1 Odhad chyby Taylorovho polynómu

Vo vzťahu (10) $R_{n+1}(t)$ označuje zvyšok, ktorý môže mať tvar:

- **Cauchyho tvar zvyšku** vychádza z Lagrangeovej vety o strednej hodnote diferenciálneho počtu. Uvažujme číslo c, ktoré sa nachádza medzi t a stredom Taylorovho polynómu t_0, čiže ak je $t < t_0$, tak $c \in (t, t_0)$ a naopak ak platí, že $t > t_0$, tak $c \in (t_0, t)$. Tvar zvyšku je teda možné zapísať ako:

$$R_{n+1}(t) = \frac{f^{(n+1)}(c)}{n!}(t-c)^n(t-t_0),$$

- **Lagrangeov tvar zvyšku** je zovšeobecnením Cauchyho tvaru, vychádza z Cauchyho vety o strednej hodnote diferenciálneho počtu. Má jednoduchšiu formu, podobá sa na (n+1)-vý člen Taylorovho polynómu. Tvar zvyšku je možné zapísať ako:

$$R_{n+1}(t) = \frac{f^{(n+1)}(c)}{(n+1)!}(t-t_0)^{n+1},$$

- **integrálny tvar zvyšku** má tvar:

$$R_{n+1}(t) = \int_{t_0}^{t} \frac{f^{(n+1)}(z)}{n!}(t-z)^n \, dz.$$

[15]

Príklad 3.1.1.1

Uvažujme najjednoduchší Maclaurinov polynóm (Taylorov polynóm čo najnižšieho rádu so stredom v nule) funkcie:

$$f(t) = e^t.$$

Pokúsme sa odhadnúť hodnotu Eulerovho čísla e s chybou menšou ako $\frac{1}{100}$. Pre ľubovolné číslo $n \in \mathbb{N}_0$ platí $f^{(n)}(t) = e^t$ a $f^{(n)}(0) = e^0 = 1$. Maclaurinov polynóm n-tého stupňa má teda podobu:

$$M_n(t) = \sum_{i=0}^{n} \frac{f^{(i)}(0)}{i!}(t-0)^i = \sum_{i=0}^{n} \frac{1}{i!} t^i = 1 + t + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots + \frac{t^n}{n!}.$$

Lagrangeov zbytok má tvar:

$$R_n(t) = \frac{f^{(n+1)}(c)}{(n+1)!}(t-0)^{n+1} = \frac{e^c}{(n+1)!} t^{n+1},$$

kde číslo c sa nachádza medzi 0 a t. Pre odhad e platí:

$$e = f(1) \doteq M_n(1) = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \cdots + \frac{1}{n!},$$

s chybou v tvare:

$$|R_n(1)| = \left| \frac{e^c}{(n+1)!} \right|,$$

kde číslo $c \in (0,1)$.

Musí platiť, že $|R_n(1)| < \frac{1}{100}$. Keďže je kladná aj exponenciálna funkcia aj faktoriál, môžeme absolútnu hodnotu z výpočtov vynechať a riešiť nerovnicu v tvare:

$$\frac{e^c}{(n+1)!} < \frac{1}{100},$$

$$100e^c < (n+1)!,$$

Následne je možné buď pravú stranu nerovnice zmenšiť, alebo naopak ľavú stranu rovnice zväčšiť, pretože ak bude platiť nová nerovnosť, bude platiť aj pôvodná. Keďže $c \in (0,1)$, potom $e^c < 3$ a $100e^c < 300!$. Z nerovnosti $300 < (n+1)!$ plynie platnosť aj pôvodnej nerovnice. Pretože $5! = 120$ nestačí pre splnenie nerovnice a $6! = 720$ už stačí, musí platiť $n \geq 5$. Keďže je potrebné získať polynóm čo najmenšieho stupňa, $n = 5$. Odhad e má potom tvar:

$$e \doteq M_5(1) = 1 + 1 + \frac{1}{2!} + \frac{1}{3!} + \frac{1}{4!} + \frac{1}{5!} = \frac{163}{60} = 2{,}7167.$$

[17]

3.2 Diferenciálna transformácia

Hlavnou výhodou tejto metódy je, že sa dá aplikovať priamo na nelineárne obyčajné diferenciálne rovnice a funkčné diferenciálne rovnice bez potreby linearizácie alebo diskretizácie. Ďalšou dôležitou výhodou je, že táto metóda je schopná výrazne zmenšiť veľkosť výpočtovej práce a pritom presne poskytovať sériové riešenie s vysokou mierou konvergencie.

Diferenciálna transformácia k-tej derivácie funkcie $u(t)$ má tvar:

$$U(k) = \frac{1}{k!}\left[u^{(k)}\right]_{t=t_0}. \qquad (11)$$

kde u je funkcia pôvodná a $U(k)$ je funkcia transformovaná. Inverzná diferenciálna transformácia funkcie $U(k)$ má tvar:

$$u(t) = \sum_{k=0}^{\infty} U(k)(t-t_0)^k, \qquad (12)$$

kde namiesto $U(k)$ dosadíme vzťah získaný v (11):

$$u(t) = \sum_{k=0}^{\infty} \frac{(t-t_0)^k}{k!}\left[u^{(k)}\right]_{t=t_0}. \qquad (13)$$

Z inverznej transformácie vyplýva, že koncepcia diferenciálnej transformácie je odvodená z expanzie Taylorovho radu. Hoci pomocou metódy diferenciálnej transformácie nie je možné vypočítať derivácie symbolicky, je možné iteratívnym spôsobom vypočítať relatívne deriváty. Spôsob tohto výpočtu je opísaný transformovanými rovnicami pôvodnej funkcie. Pretože metóda diferenciálnej transformácie je založená na Taylorovom rade, je jasné, že podmienky konvergencie pre metódu diferenciálnej transformácie budú rovnaké ako pre Taylorovu radu. V praxi je funkcia $u(t)$ vyjadrená ako $u_{N+1}(t) + R_{N+1}$, kde $u_{N+1}(t)$ má tvar:

$$u_n(t) = \sum_{k=0}^{n} U(k)(t - t_0)^k,$$

a R_{N+1} je zvyšok vyjadrený Lagrangeovým tvarom zbytku:

$$R_{N+1}(t) = \frac{u^{(n+1)}(c)}{(n+1)!}(t - t_0)^{n+1},$$

kde c leží niekde medzi t_0 a t. Ak (n+1)-vá derivácia funkcie $u(t)$ je ohraničená pre $t \in (0,1)$ platí:

$$\left|u^{(n+1)}(t)\right| \leq K,$$

kde K je daná nezáporná konštanta. Potom je možné maximálnu chybu pre $u_n(t)$ na tomto intervale vyjadriť ako:

$$e_{max} = \frac{K}{(N+1)!}. \tag{14}$$

V praxi sa pre zastavenie výpočtu s danou presnosťou ε používa jednoduchšie overiteľná podmienka:

$$|u_{n+1}(t) - u_n(t)| < \varepsilon. \tag{15}$$

Teda výpočet ďalších členov Taylorovho rozvoja riešenia u bežia tak dlho, pokiaľ rozdiel približných riešení nie je dostatočne malý. V algoritme pre diferenciálnu transformáciu pre $t_0 = 0$ možno vzťah (15) prepísať ako:

$$|U(n + 1)t^{n+1}| < \varepsilon,$$

kde t leží v uvažovanom intervale. Pre $t \in \langle 0, b\rangle$ platí odhad:

$$|U(n + 1)t^{n+1}| \leq |U(n + 1)b^{n+1}| < \varepsilon.$$

Zo vzťahov (11) a (13) pre $t_0 = 0$ je možné odvodiť základné pravidlá diferenciálnej transformácie. Uvažujme, že $F(k)$, $G(k)$, $H(k)$ a $U_i(k)$, $i = 1, \dots, n$, sú diferenciálnymi transformáciami funkcií $f(t)$, $g(t)$, $h(t)$ a $u_i(t)$, $i = 1, \dots, n$. Potom platí:

- Ak $f(t) = u^{(n)}$, potom $F(k) = \frac{(k+n)!}{k!}U(k + n)$. (16)
- Ak $f(t) = u(t)h(t)$, potom $F(k) = \sum_{l=0}^{k} U(l)H(k - l)$. (17)
- Ak $f(t) = t^n$, potom $F(k) = \delta(k - n)$, kde δ je Kroneckerova delta. (18)

- Ak $f(t) = e^{\lambda t}$, potom $F(k) = \frac{\lambda^k}{k!}$. (19)
- Ak $f(t) = \sin(at + b)$, potom $F(k) = \frac{a^k}{k!}\sin\left(\frac{k\pi}{2} + b\right)$. (20)
- Ak $f(t) = \cos(at + b)$, potom $F(k) = \frac{a^k}{k!}\cos\left(\frac{k\pi}{2} + b\right)$. (21)
- Ak $f(t) = \prod_{i=1}^{n} u_i(t)$, potom

  $F(k) = \sum_{s_1=0}^{k} \sum_{s_2=0}^{k-s_1} \dots \sum_{s_n=0}^{k-s_1-\dots-s_{n-1}} U_1(s_1) \dots U_{n-1}(s_{n-1}) U_n(k - s_1 - \dots - s_n)$.

  (22)

Okrem základných pravidiel diferenciálnej transformácie sú odvodené aj ďalšie pravidlá napríklad pre špeciálne typy zložených funkcii alebo pre funkcie so stredom inde ako v nule. [10] [16]

II. PRAKTICKÁ ČASŤ

4 RIEŠENÉ PRÍKLADY POMOCOU DIFERENCIÁLNEJ TRANFORMÁCIE

4.1 Príklad č. 1

Uvažujme diferenciálnu rovnicu $u' - u = e^t$ s danou počiatočnou podmienkou $u(0) = 0$ na intervale $t \in (0{,}1)$, s presnosťou $\frac{1}{100}$. Pomocou základných pravidiel diferenciálnej transformácie popísaných v 3. kapitole, konkrétne podľa vzťahu (16) a (19), sme preložili danú diferenciálnu rovnicu do tvaru:

$$(k+1)U(k+1) - U(k) = \frac{1}{k!},$$

kde $k \geq 0$. Rovnicu sme následne upravili do tvaru potrebného pre dosadzovanie za k:

$$U(k+1) = \frac{1}{k+1}\left[U(k) + \frac{1}{k!}\right]. \tag{23}$$

V počiatočnej podmienke je dané, že $u(0) = 0$, takže $U(0) = 0$. Následne dosadzujeme do vzťahu (23):

$$U(1) = 1$$

$$U(2) = 1$$

$$U(3) = \frac{1}{2!} = 0{,}5$$

$$U(4) = \frac{1}{3!} = \frac{1}{6} \doteq 0{,}1667$$

$$U(5) = \frac{1}{4!} = \frac{1}{24} \doteq 0{,}0417$$

...

Výsledok zapíšeme v tvare:

$$\begin{aligned} u(t) &= t + t^2 + \frac{1}{2!}t^3 + \frac{1}{3!}t^4 + \frac{1}{4!}t^5 + \cdots + \frac{1}{k!}t^{k-1} \ldots \\ &= t\left(1 + t + \frac{1}{2!}t^2 + \frac{1}{3!}t^3 + \frac{1}{4!}t^4 + \cdots + \frac{1}{k!}t^k \ldots\right). \end{aligned}$$

V prípade, že $k \to \infty$, môžeme pozorovať, že riešenie získané pomocou metódy diferenciálnej transformácie konverguje k riešeniu:

$$u(t) = te^t.$$

[11]

Výpočet presnosti je daný vzťahom (15) a je možné ho zapísať v tvare:

$$U(N+1) = \frac{1}{N!}.$$

Podľa podmienky uvedenej v zadaní príkladu vypočítame k podľa vzťahu:

$$\frac{1}{N!} \leq \frac{1}{100}.$$

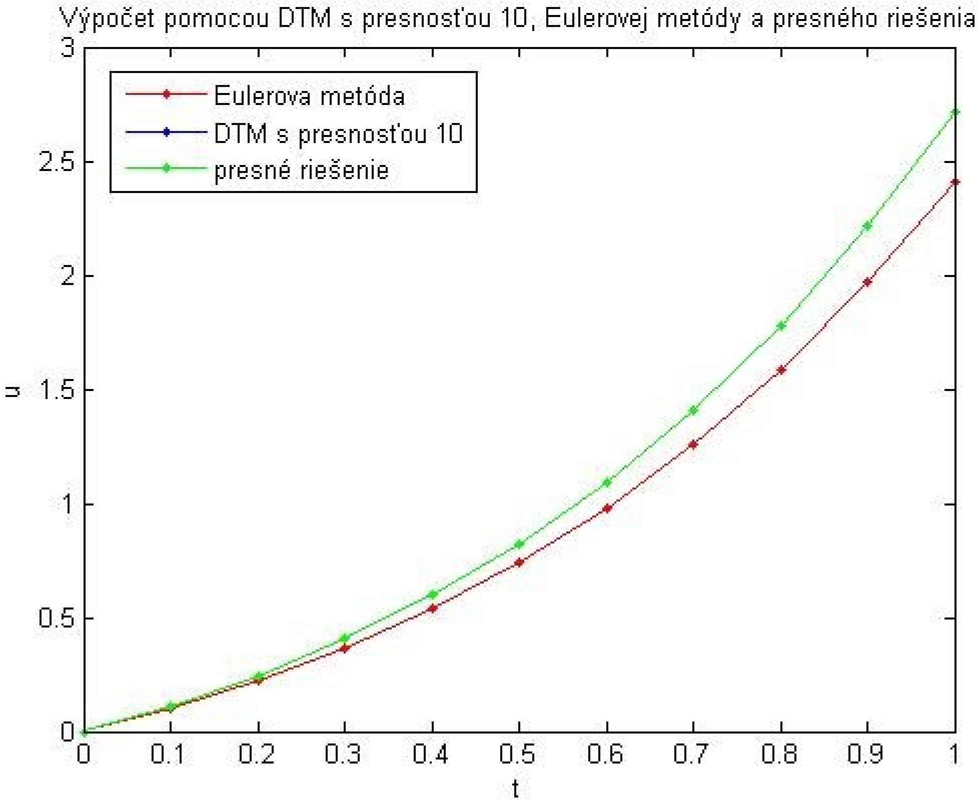
Keďže N nám vyšlo 5, pre splnenie danej presnosti nám stačí určiť prvých 5 členov numerického riešenia.

V systéme MATLAB sme porovnali výsledky získané použitím Eulerovej metódy, funkcie ode45, nami vytvorenou funkciou pre riešenie pomocou DTM a presným riešením. Výsledky sme vyniesli do tabuľky a do grafu.

Tabuľka 3 – Výsledky príkladu č. 1

Eulerova m.	DTM (5)	DTM (10)	ode45	presné r.
0	0	0	0	0
0,1	0,11051708	0,11051709	0,11051709	0,11051710
0,2205171	0,24428000	0,24428055	0,24428055	0,24428056
0,3647091	0,40495125	0,40495764	0,40495764	0,40495765
0,5361659	0,59669333	0,59672988	0,59672988	0,59672989
0,7389649	0,82421875	0,82436064	0,82436064	0,82436065
0,9777335	1,09284000	1,09327128	1,09327128	1,09327129
1,2577188	1,40851958	1,40962689	1,40962690	1,40962691
1,5848659	1,77792000	1,78043272	1,78043274	1,78043276
1,9659066	2,20845375	2,21364271	2,21364280	2,21364282
2,4084576	2,70833333	2,71828153	2,71828183	2,71828184

Pri tomto príklade sme vykonávali výpočty na intervale $t \in \langle 0; 1 \rangle$ s krokom 0,1. Z tabuľky č. 3 je možné badať, že najpresnejšia metóda je ode45 a najmenej presná je Eulerova metóda. Metóda DTM s rádom výsledného polynómu 5 sa líši od presného riešenia v poslednom kroku o jednu stotinu, zatiaľ čo metóda DTM s presnosťou 10 sa líši len o 3 desaťmilióntiny.



Obrázok 4 – Výsledky príkladu č. 1

4.2 Príklad č. 2

Uvažujme diferenciálnu rovnicu $u'' + u = 0$ s danými počiatočnými podmienkami $u(0) = 1$ a $u'(0) = 1$. Pomocou základných pravidiel diferenciálnej transformácie popísaných v 3. kapitole, konkrétne podľa vzťahu (16), sme preložili danú diferenciálnu rovnicu do tvaru:

$$(k+1)(k+2)U(k+2) + U(k) = 0,$$

Rovnicu sme následne upravili do tvaru potrebného pre dosadzovanie za k:

$$U(k+2) = -\frac{U(k)}{(k+1)(k+2)}. \tag{24}$$

Z počiatočných podmienok vyplýva, že $U(0) = 1$ a $U(1) = 1$. Následne dosadzujeme do vzťahu (24):

$$U(2) = -\frac{1}{2!} = -0{,}5$$

$$U(3) = -\frac{1}{3!} = -\frac{1}{6} \doteq -0{,}166667$$

$$U(4) = \frac{1}{4!} = \frac{1}{24} \doteq 0{,}041667$$

$$U(5) = \frac{1}{5!} = \frac{1}{120} \doteq 0{,}008333$$

$$U(6) = -\frac{1}{6!} = -\frac{1}{720} \doteq -0{,}001388$$

$$U(7) = -\frac{1}{7!} = -\frac{1}{5040} \doteq -0{,}000198$$

...

Výsledok zapíšeme v tvare:

$$u(t) = 1 + t - \frac{1}{2!}t^2 - \frac{1}{3!}t^3 + \frac{1}{4!}t^4 + \frac{1}{5!}t^5 - \frac{1}{6!}t^6 - \frac{1}{7!}t^7 + \cdots$$
$$= \left(1 - \frac{1}{2!}t^2 + \frac{1}{4!}t^4 - \frac{1}{6!}t^6 + \cdots\right) + \left(1 - \frac{1}{3!}t^3 + \frac{1}{5!}t^5 - \frac{1}{7!}t^7 + \cdots\right).$$

V prípade, že $k \to \infty$, môžeme pozorovať, že riešenie získané pomocou metódy diferenciálnej transformácie konverguje k riešeniu:

$$u(t) = \cos t + \sin t.$$

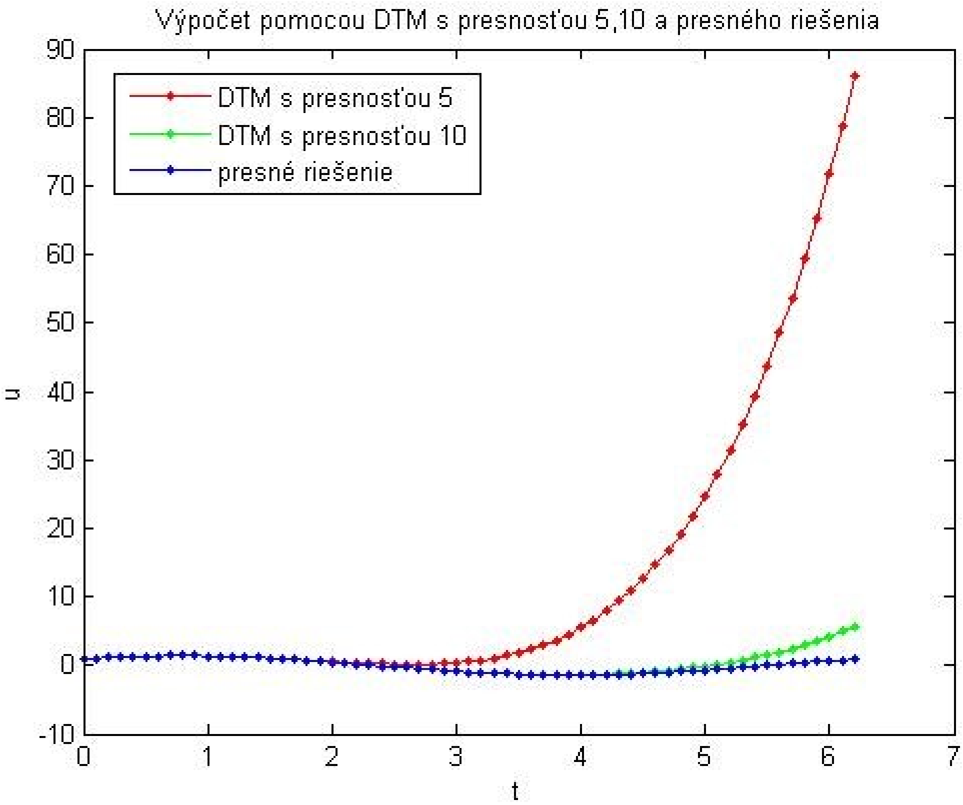
[11]

V systéme MATLAB sme porovnali výsledky získané použitím Eulerovej metódy, funkcie ode45, nami vytvorenou funkciou pre riešenie pomocou DTM a presným riešením. Výsledky sme vyniesli do tabuľky a do grafu, pričom do tabuľky sme vyniesli prvých 10 krokov výpočtu.

Tabuľka 4 – Výsledky príkladu č. 2

Eulerova m.	DTM (5)	DTM (10)	ode45	presné r.
1	1	1	1	1
1,1	1,09483758	1,09483758	1,09483758	1,09483758
1,19	1,17873600	1,17873591	1,17873591	1,17873591
1,269	1,25085775	1,25085670	1,25085669	1,25085670
1,3361	1,31048533	1,31047934	1,31047933	1,31047934
1,39051	1,35703125	1,35700810	1,35700810	1,35700810
1,431559	1,39004800	1,38997809	1,38997809	1,38997809
1,4587029	1,40923808	1,40905987	1,40905987	1,40905987
1,47153121	1,41446400	1,41406280	1,41406280	1,41406280
1,46977249	1,40575825	1,40493689	1,40493687	1,40493688

1,45329846	1,38333333	1,38177331	1,38177329	1,38177329



Obrázok 5 – Výsledky príkladu č. 2

Pri tomto príklade sme vykonávali výpočty na intervale $t \in \langle 0; 6{,}28 \rangle$ s krokom 0,1. Z obrázka č. 5 je možné badať, že metóda diferenciálnej transformácie s presnosťou 5 sa začína odchyľovať od presného riešenia už približne po 20 krokoch, zatiaľ čo metóda diferenciálnej transformácie s presnosťou 10 až po približne 40 krokoch. Aby bolo riešenie pomocou metódy diferenciálnej transformácie zhodné s presným riešením na danom intervale, musela by byť zadaná hodnota presnosti minimálne 17, čo je to dané tým, že pravý kraj intervalu je relatívne ďaleko od stredu rady v nule.

4.3 Príklad č. 3

Uvažujme diferenciálnu rovnicu $u'' - 3u' + 2u = 2t - 3$ s danými počiatočnými podmienkami $u(0) = 1$ a $u'(0) = 2$. Pomocou základných pravidiel diferenciálnej transformácie popísaných v 3. kapitole, konkrétne podľa vzťahu (16) a (18), sme preložili danú diferenciálnu rovnicu do tvaru:

$$(k+1)(k+2)U(k+2) - 3(k+1)U(k+1) + 2U(k) = 2\delta(k-1) - 3\delta(k).$$

kde $k \geq 0$. Rovnicu sme následne upravili do tvaru potrebného pre dosadzovanie za k:

$$U(k+2) = \frac{1}{(k+1)(k+2)}[3(k+1)U(k+1) - 2U(k) + 2\delta(k-1) - 3\delta(k)]. \tag{25}$$

Z počiatočných podmienok vyplýva, že $U(0) = 1$ a $U(1) = 2$. Následne dosadzujeme do vzťahu (25):

$$U(2) = \frac{1}{2!} = 0{,}5$$

$$U(3) = \frac{1}{3!} = \frac{1}{6} \doteq 0{,}166667$$

$$U(4) = \frac{1}{4!} = \frac{1}{24} \doteq 0{,}041667$$

$$U(5) = \frac{1}{5!} = \frac{1}{120} \doteq 0{,}008333$$

...

Výsledok zapíšeme v tvare:

$$u(t) = 1 + 2t + \frac{1}{2!}t^2 + \frac{1}{3!}t^3 + \frac{1}{4!}t^4 + \cdots + \frac{1}{k!}t^k + \cdots =$$

$$= t\left(1 + t + \frac{1}{2!}t^2 + \frac{1}{3!}t^3 + \frac{1}{4!}t^4 + \cdots + \frac{1}{k!}t^k \ldots\right).$$

V prípade, že $k \to \infty$, môžeme pozorovať, že riešenie získané pomocou metódy diferenciálnej transformácie konverguje k riešeniu:

$$u(t) = t + e^t.$$

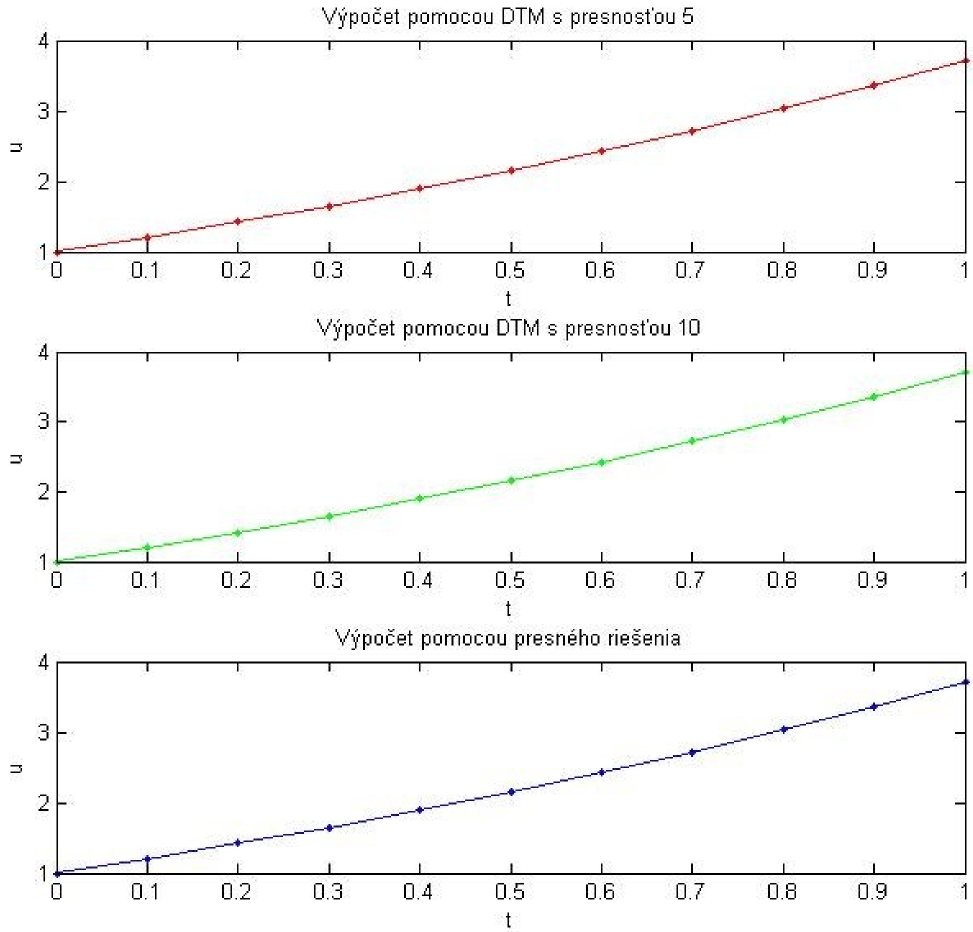
[11]

V systéme MATLAB sme porovnali výsledky získané použitím Eulerovej metódy, funkcie ode45, nami vytvorenou funkciou pre riešenie pomocou DTM a presným riešením. Výsledky sme vyniesli do tabuľky a do grafu.

Tabuľka 5 – Výsledky príkladu č. 3

Eulerova m.	DTM (5)	DTM (10)	ode45	presné r.
1	1	1	1	1
1,2	1,20517092	1,20517092	1,20517092	1,20517092
1,35	1,42140267	1,42140276	1,42140276	1,42140276
1,488	1,64985775	1,64985881	1,64985881	1,64985881
1,616	1,89181867	1,89182470	1,89182470	1,89182470

1,736	2,14869792	2,14872127	2,14872127	2,14872127
1,85	2,42204800	2,42211880	2,42211880	2,42211880
1,96	2,71357142	2,71375271	2,71375271	2,71375271
2,068	3,02513067	3,02554093	3,02554093	3,02554093
2,176	3,35875825	3,35960310	3,35960312	3,35960311
2,286	3,71666667	3,71828180	3,71828183	3,71828183



Obrázok 6 – Výsledky príkladu č. 3

Pri tomto príklade sme vykonávali výpočty na intervale $t \in \langle 0; 1 \rangle$ s krokom 0,1. Z obrázka č. 6 je možné badať, že výsledky sa takmer zhodujú, avšak z tabuľky č. 5 pozorujeme, že výpočet pomocou Eulerovej metódy je veľmi nepresný. Metóda DTM s presnosťou 5 sa líši od presného riešenia v poslednom kroku o jednu tisícinu, zatiaľ čo metóda DTM s presnosťou 10 sa líši len o 3 stomilióntiny.

4.4 Príklad č. 4

Uvažujme diferenciálnu rovnicu $u' - 2tu = 0$ s danou počiatočnou podmienkou $u(0) = 1$. Pomocou základných pravidiel diferenciálnej transformácie popísaných v 3. kapitole, konkrétne podľa vzťahu (16) a (17), sme preložili danú diferenciálnu rovnicu do tvaru:

$$(k+1)U(k+1) - 2\sum_{m=0}^{k} \delta(m-1)\, U(k-m) = 0,$$

kde δ označuje Kroneckerovo delta. Kroneckerovo delta je matematická funkcia dvoch premenných, obvykle celých čísel. Je pomenovaná po Leopoldovi Kroneckerovi (1823-1891). Táto funkcia sa rovná 1 v prípade keď je premenná rovná nule, v ostatných prípadoch sa funkcia rovná nule.

Rovnicu sme následne upravili do tvaru potrebného pre dosadzovanie za k:

$$U(k+1) = \frac{2}{k+1}\sum_{m=0}^{k} \delta(m-1)\, U(k-m). \tag{26}$$

Z počiatočnej podmienky vyplýva, že $U(0) = 1$. Následne dosadzujeme do vzťahu (26):

$$k = 0\colon U(1) = 2\sum_{m=0}^{0} \delta(m-1)\, U(-m) = 0U(0) = 0$$

$$k = 1\colon U(2) = \sum_{m=0}^{1} \delta(m-1)\, U(1-m) = 0U(1) + 1U(0) = 1$$

$$k = 2\colon U(3) = \frac{2}{3}\sum_{m=0}^{2} \delta(m-1)\, U(2-m) = 0U(2) + 1U(1) + 0U(0) = 0$$

$$k = 3\colon U(4) = \frac{1}{2}\sum_{m=0}^{3} \delta(m-1)\, U(3-m) = \frac{1}{2}[0U(3) + 1U(2) + 0U(1) + 0U(0)]$$
$$= \frac{1}{2!} = 0{,}5$$

$$k = 4\colon U(5) = \frac{2}{5}\sum_{m=0}^{4} \delta(m-1)\, U(4-m)$$
$$= 0U(4) + 1U(3) + 0U(2) + 0U(1) + 0U(0) = 0$$

$$k = 5: U(6) = \frac{1}{3}\sum_{m=0}^{5} \delta(m-1)\,U(5-m)$$

$$= \frac{1}{3}[0U(5) + 1U(4) + 0U(3) + 0U(2) + 0U(1) + 0U(0)] = \frac{1}{3!} = \frac{1}{6}$$

...

Výsledok zapíšeme v tvare:

$$u(t) = 1 + t^2 + \frac{1}{2!}t^4 + \frac{1}{3!}t^6 + \frac{1}{4!}t^8 + \cdots = 1 + t^2 + \frac{(t^2)^2}{2!} + \frac{(t^2)^3}{3!} + \frac{(t^2)^4}{4!} + \cdots$$

V prípade, že $k \to \infty$, môžeme pozorovať, že riešenie získané pomocou metódy diferenciálnej transformácie konverguje k riešeniu:

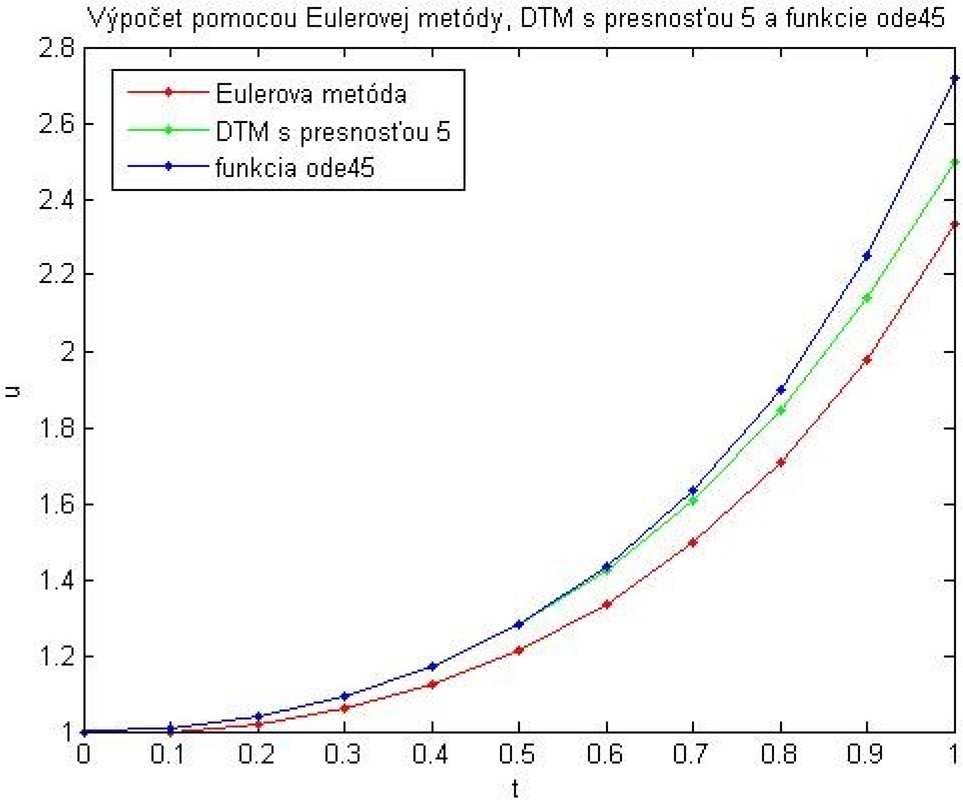
$$u(t) = e^{t^2}.$$

[11]

V systéme MATLAB sme porovnali výsledky získané použitím Eulerovej metódy, funkcie ode45, nami vytvorenou funkciou pre riešenie pomocou DTM a presným riešením. Výsledky sme vyniesli do tabuľky a do grafu.

Tabuľka 6 – Výsledky príkladu č. 4

Eulerova m.	DTM (5)	DTM (10)	ode45	presné r.
1	1	1	1	1
1	1,01005	1,01005017	1,01005017	1,01005017
1,02	1,0408	1,04081077	1,04081077	1,04081077
1,0608	1,09405	1,09417428	1,09417428	1,09417428
1,124448	1,1728	1,17351085	1,17351087	1,17351087
1,21440384	1,28125	1,28402507	1,28402542	1,28402542
1,33584422	1,4248	1,43332623	1,43332941	1,43332941
1,49614553	1,61005	1,63229556	1,63231622	1,63231622
1,70560591	1,8448	1,89637596	1,89648089	1,89648088
1,97850285	2,13805	2,24746529	2,24790801	2,24790799
2,33463336	2,5	2,71666667	2,71828188	2,71828183



Obrázok 7 – Výsledky príkladu č. 4

Pri tomto príklade sme vykonávali výpočty na intervale $t \in \langle 0; 1 \rangle$ s krokom 0,1. Z obrázka č. 7 je možné badať, že výpočet pomocou Eulerovej metódy je najmenej presný. Metóda DTM s presnosťou 5 sa líši od presného riešenia v poslednom kroku o dve desatiny, zatiaľ čo metóda DTM s presnosťou 10 sa líši o dve tisíciny a funkcia ode45 iba o 5 stomilióntin.

4.5 Príklad č. 5

Uvažujme diferenciálnu rovnicu $u' - u^2 = 1$ s danou počiatočnou podmienkou $u(0) = 0$. Pomocou základných pravidiel diferenciálnej transformácie popísaných v 3. kapitole, konkrétne podľa vzťahu (16) a (17), sme preložili danú diferenciálnu rovnicu do tvaru:

$$(k+1)U(k+1) - \sum_{m=0}^{k} U(k)\,U(k-m) = \delta(k)$$

Rovnicu sme následne upravili do tvaru potrebného pre dosadzovanie za k:

$$U(k+1) = \frac{\delta(k) + \sum_{m=0}^{k} U(m)\,U(k-m)}{k+1} \tag{27}$$

Z počiatočnej podmienky vyplýva, že $U(0) = 0$. Následne dosadzujeme do vzťahu (27):

$$k = 0: U(1) = \frac{1 + \sum_{m=0}^{0} U(m)\, U(0-m)}{0+1} = \frac{1+0}{1} = 1$$

$$k = 1: U(2) = \frac{0 + \sum_{m=0}^{1} U(m)\, U(1-m)}{1+1} = \frac{0 + [(0 \cdot 1) + (1 \cdot 0)]}{2} = \frac{0+0}{2} = 0$$

$$k = 2: U(3) = \frac{0 + \sum_{m=0}^{2} U(m)\, U(2-m)}{2+1} = \frac{0 + [(0 \cdot 0) + (1 \cdot 1) + (0 \cdot 0)]}{3} =$$

$$= \frac{0+1}{3} = \frac{1}{3}$$

$$k = 3: U(4) = \frac{0 + \sum_{m=0}^{3} U(m)\, U(3-m)}{3+1} =$$

$$= \frac{0 + \left[\left(0 \cdot \frac{1}{3}\right) + (0 \cdot 0) + (0 \cdot 1) + \left(\frac{1}{3} \cdot 0\right)\right]}{4} = \frac{0+0}{4} = 0$$

$$k = 4: U(5) = \frac{0 + \sum_{m=0}^{4} U(m)\, U(4-m)}{4+1} =$$

$$= \frac{0 + \left[(0 \cdot 0) + \left(1 \cdot \frac{1}{3}\right) + (0 \cdot 0) + \left(\frac{1}{3} \cdot 1\right) + (0 \cdot 0)\right]}{5} = \frac{0 + \left[\frac{1}{3} + \frac{1}{3}\right]}{5}$$

$$= \frac{\frac{2}{3}}{5} = \frac{2}{15}$$

...

Výsledok zapíšeme v tvare:

$$u(t) = t + \frac{t^3}{3} + \frac{2t^5}{15} + \cdots$$

V prípade, že $k \to \infty$, môžeme pozorovať, že riešenie získané pomocou metódy diferenciálnej transformácie konverguje k riešeniu:

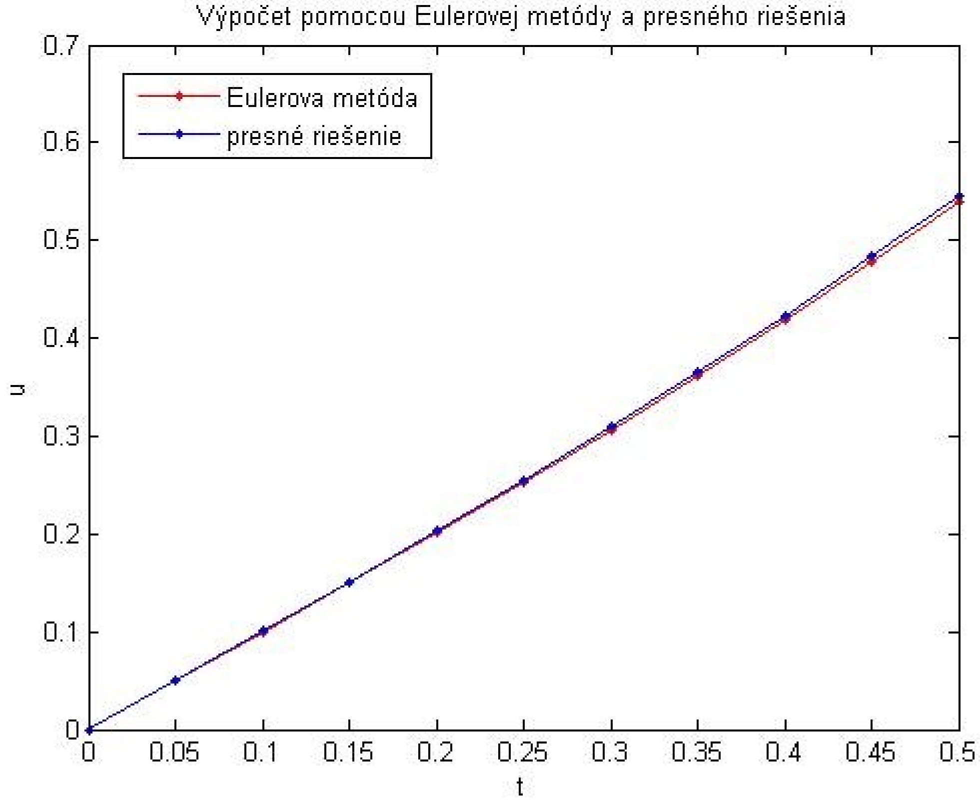
$$u(t) = \tan t.$$

V systéme MATLAB sme porovnali výsledky získané použitím Eulerovej metódy, funkcie ode45, nami vytvorenou funkciou pre riešenie pomocou DTM a presným riešením. Výsledky sme vyniesli do tabuľky a do grafu.

Tabuľka 7 – Výsledky príkladu č. 5

Eulerova m.	DTM (5)	DTM (10)	ode45	presné r.
0	0	0	0	0
0,05	0,05004171	0,05004171	0,05004171	0,05004171

0,100125	0,10033467	0,10033467	0,10033467	0,10033467
0,15062625	0,15113513	0,15113522	0,15113522	0,15113522
0,20176066	0,20270933	0,20271004	0,20271004	0,20271004
0,25379603	0,25533854	0,25534192	0,25534192	0,25534192
0,30701665	0,30932400	0,30933623	0,30933625	0,30933625
0,36172962	0,36499196	0,36502840	0,36502850	0,36502849
0,41827203	0,42269867	0,42279282	0,42279322	0,42279322
0,47701961	0,48283538	0,48305359	0,48305507	0,48305507
0,53839699	0,54583333	0,54629767	0,54630249	0,54630249



Obrázok 8 – Výsledky príkladu č. 5

Pri tomto príklade sme vykonávali výpočty na intervale $t \in \langle 0; 0{,}5 \rangle$ s krokom 0,05. Z obrázka č. 8 je možné pozorovať, že Eulerova metóda je v tomto prípade pomerne presná. Metóda DTM s presnosťou 5 sa líši od presného riešenia v poslednom kroku o jednu tisícinu, zatiaľ čo metóda DTM s presnosťou 10 sa líši o jednu stotisícinu.

V tejto práci sme sa zaoberali aj rýchlosťou výpočtu pomocou danej metódy. Rýchlosť sme merali pomocou príkazov tic a toc. Tieto príkazy fungujú tak, že výstupom je čas v sekundách, ktorý reprezentuje čas ubehnutý od momentu kedy systém prečítal príkaz tic, do momentu kedy prečítal príkaz toc. Namerané údaje sme vyniesli do tabuľky.

Tabuľka 8 – Prehľad nameraných rýchlostí výpočtu

	Eulerova m.	DTM (5)	DTM (10)	ode45	presné r.
1. príklad	0,0123	0,1659	0,3716	0,019	0,0127
2. príklad	0,011	0,1378	0,3016	0,0163	0,0123
3. príklad	0,0107	0,1262	0,305	0,0181	0,014
4. príklad	0,0101	0,1719	0,3082	0,0195	0,0103
5. príklad	0,0108	0,155	0,3494	0,0167	0,0196
priemer	0,01098	0,1514	0,32716	0,0179	0,01378

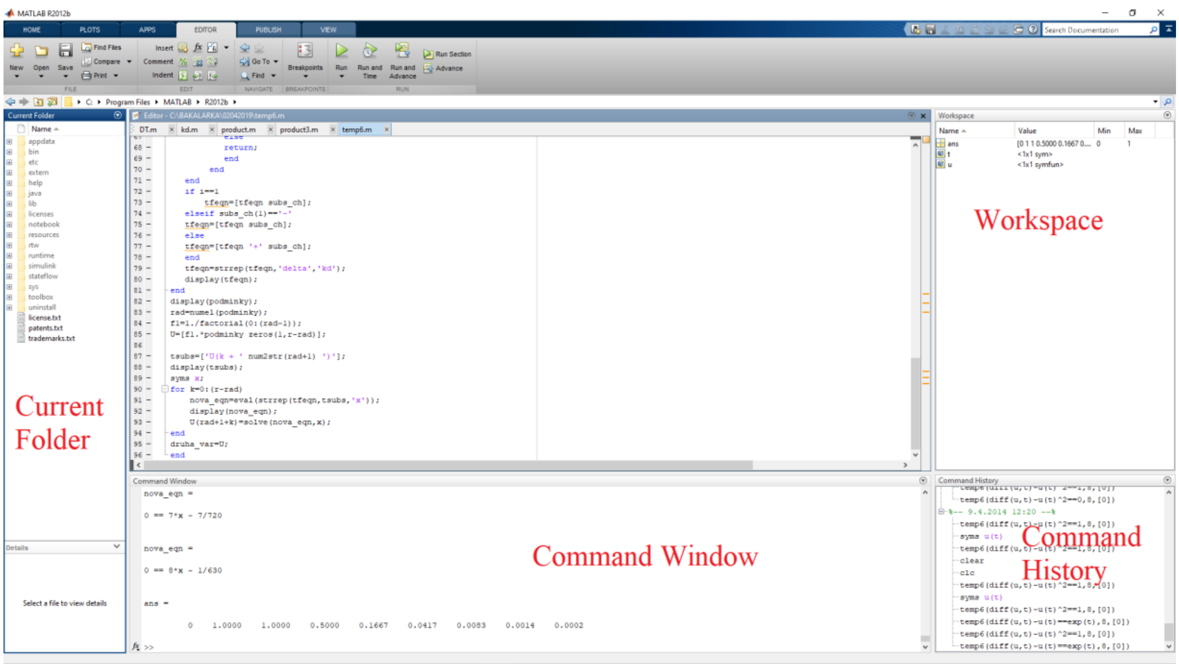
Z tabuľky č. 9 môžeme pozorovať, že najrýchlejšou metódou je Eulerova metóda s priemerným časom výpočtu 0,011 sekundy. Naopak najpomalšou je nami implementovaná metóda DTM v systéme MATLAB. Z tabuľky č. 9 pozorujeme, že rýchlosť výpočtu značne ovplyvňuje žiadaná presnosť výpočtu. Vysoký čas výpočtu je spôsobený najmä nedostatočnou optimalizáciou algoritmu.

5 MATLAB

MATLAB od firmy MATHWORKS prvý krát vydaný v roku 1984 jednoznačne patrí medzi najpoužívanejšie výpočtové systémy na svete. Názov MATLAB vznikol spojením a skrátením anglických slov *matrix* a *laboratory*. Pomocou tohto systému možno vytvárať napríklad algoritmy pre analýzu a vizualizáciu dát, numerické výpočty a pod. Špecifickým nástrojom sú tzv. toolboxy, ktoré rozširujú základnú verziu systému pre rôzne konkrétne oblasti, napríklad toolbox Simulink slúži pre simuláciu a modelovanie dynamických systémov. V tejto práci pracujeme so systémom MATLAB vo verzii 2012b. [14]

5.1 Popis prostredia

Ihneď po spustení programu MATLAB sa na monitore zobrazí pracovná plocha vývojového prostredia.



Obrázok 9 – Pracovná plocha vývojového prostredia MATLAB

Pracovná plocha je rozdelená na niekoľko častí:

- **Current Folder** zobrazuje aktuálne otvorený adresár a jeho podadresáre a súbory,
- **Command Window** je najdôležitejšou časťou pracovnej plochy, pretože slúži pre komunikáciu medzi systémom, ktorý sem vypisuje výsledky výpočtov a užívateľom, ktorý zapisuje do tohto okna príkazy,

- **Workspace** informuje užívateľa a momentálne definovaných premenných a hodnotách v nich uložených,
- **Command History** je časť pracovnej plochy, ktorá informuje o použitých príkazoch v minulosti. Primárne teda slúži pre rýchlejší prístup k už použitým príkazom,
- **Hlavná lišta** slúži pre jednoduchý prehľad možností práce s programom. Pomocou hlavnej lišty je možné napríklad exportovať alebo importovať dáta, vytvárať nové súbory a funkcie a pod. [14]

5.2 Algoritmus diferenciálnej transformácie

Algoritmus pre implementáciu výpočtu približného riešenia diferenciálnej rovnice s počiatočnou podmienkou pomocou diferenciálnej transformácie má dve hlavné časti:

- preklad rovnice pro neznámu funkciu $u(t)$ pomocou „slovníka" (vzťahy (16) až (22)), ktorého výsledkom je transformovaná rovnica pre $U(n)$,
- riešenie tejto rekurentnej rovnice. Výsledné zložky vektoru U potom reprezentujú koeficienty polynómu dávajúceho približné riešenie.

Vstupom do algoritmu je diferenciálna rovnica, presnosť vyjadrená maximálnym stupňom polynómu a počiatočné podmienky. Na úvod je potrebné overiť či boli zadané všetky údaje potrebné pre výpočet. Následne je potrebné zistiť všetky členy rovnice aby bolo možné vykonať transformáciu. Potom je potrebné o každom členovi rovnice rozhodnúť či je daný člen číslom, výrazom, alebo súčinom viacerých výrazov. Podľa výsledku rozhodnutia prebehne preklad podľa pravidiel diferenciálnej transformácie popísaných v 3. kapitole. Po preložení celej rovnice je potrebné osamostatniť transformovanú funkciu s najvyšším rádom a substituovať ju za novú neznámu. Na záver sa vykoná dosadzovanie do novej substituovanej rovnice, ktorej výsledok predstavuje polynóm určený vektorom jeho koeficientov.

5.3 Implementácia algoritmu v programe MATLAB

Hlavná časť algoritmu prebieha vo funkcií **DTM**, kde prebieha rozklad rovnice na členy a ich následná identifikácia. Vo funkcii DTM tiež prebieha prípadné oddelenie konštánt daných členov, nahradenie pôvodných členov funkcie preloženými, substituovanie rovnice a následné dosadzovanie. Volanie funkcie DTM má odlišnú podobu ako volanie funkcie ode45, ktorá je implementovaná v systéme MATLAB. Porovnanie volania funkcií DTM a ode45 pre príklady riešené v 4. kapitole tejto práce sme vyniesli do tabuľky.

Tabuľka 9 – Porovnanie volania funkcií

	funkcia DTM	funkcia ode45
príklad č. 1	DTM(diff(u,t)-u(t)==exp(t),5,[0])	[t,u] = ode45(@(t,u) exp(t)+u, tspan, u0)
príklad č. 2	DTM(diff(u,2,t)+u(t)==0,5,[1 1])	[t,y] = ode45(@(t,y) odefcn1(t,y), tspan, y0)
príklad č. 3	DTM(diff(u,2,t)-3*diff(u,t)+2*u(t)==2*t-3,5,[1 2])	[t,y] = ode45(@(t,y) odefcn2(t,y), tspan, y0)
príklad č. 4	DTM(diff(u,t)-2*t*u(t)==0,5,[1])	[t,u] = ode45(@(t,u) 2*t*u, tspan, u0)
príklad č. 5	DTM(diff(u,t)-u(t)*u(t)==1,5,[0])	[t,u] = ode45(@(t,u) u^2+1, tspan, u0)

Funkcia **DT** slúži pre preklad podľa pravidiel diferenciálnej transformácie. Funkcia DT dokáže preložiť funkcie $\sin(at+b)$, $\cos(at+b)$, e^{at}, t^n pre $n \geq 1$, $u(t)$, $u', \dots, u^{(n)}$. Na úvod funkcie DT sa vykoná kontrola či funkcia na vstupe náhodou nie je prázdna. Následne sa rovnica prevedie do textového reťazca, v ktorom je potrebné vyhľadať symboly určujúce vzťahy pre preklad (sin, cos, exp, mocnina a pod.). Výstupom je nový transformovaný výraz.

Funkcia **product** slúži pre preklad ak sú v jednom člene rovnice dva výrazy, pracuje podľa vzťahu (15) základných pravidiel diferenciálnej transformácie. Funkcia **product3** slúži pre preklad ak sú v jednom člene rovnice tri výrazy a pracuje podľa vzťahu (20) základných pravidiel diferenciálnej transformácie.

Funkcia **kd** slúži pre definovanie Kroneckerovej delty, keďže systém MATLAB vo verzii s ktorou sme pracovali nemá implementovanú obdobnú funkciu.

Nami vytvorený algoritmus dokáže pracovať s funkciami sínus, kosínus, exponenciálou, s deriváciou a s mocninou. Jeden člen rovnice môže obsahovať maximálne súčin troch vyššie spomenutých funkcií. Algoritmus dokáže riešiť aj nelineárne diferenciálne rovnice, v podobe mocniny neznámej. Presnosť je riadená zadaním stupňa výsledného polynómu. Ovládanie presnosti pracuje na princípe vzťahu (15) popísaného detailnejšie v kapitole 3.2 v teoretickej časti tejto práce.

5.3.1 Prehľad funkcií použitých v algoritme

Veľkou výhodou systému MATLAB je to, že má zabudované niektoré funkcie umožňujúce prácu so symbolickými výrazmi. Z toho hľadiska je pre nás dôležitou funkciou pri rozdelení rovnice na členy funkcia **children**. Pomocou tejto funkcie a funkcií **numel** a **coeffs** sme rovnicu rozdelili na členy a členy z pravej strany rovnice presunuli na ľavú. Výstupom funkcie children je vektor obsahujúci členy rovnice, výstupom funkcie coeffs je vektor, ktorý obsahuje koeficienty daných členov rovnice. Funkcia numel vracia číslo reprezentujúce počet prvkov vo vektore.

```
deti=children(eqn);

lhs=deti(1);
rhs=deti(2);

if (numel(coeffs(lhs))==1)&&(numel(coeffs(rhs))==1)
    vyrazy=[lhs, -1.*rhs];
elseif (numel(coeffs(lhs))==1)&&(numel(coeffs(rhs))~=1)
    vyrazy=[lhs, -1.*children(rhs)];
elseif (numel(coeffs(lhs))~=1)&&(numel(coeffs(rhs))==1)
    vyrazy=[children(lhs), -1.*rhs];
else
    vyrazy=[children(lhs),-1.*children(rhs)];
end
display(vyrazy);
pocet_vyrazu=numel(vyrazy);
```

Obrázok 10 – Ukážka použitia funkcií children, coeffs a numel v algoritme.

Funkcie **isempty**, **isequal** a **isnumeric** pracujú na podobnom princípe, ich výstupom je totiž logická hodnota buď 0, alebo 1. Ak je vstup do funkcie isempty prázdny, výstupom je logická 1. Funkcia isequal porovnáva vektory, ak sú totožné, výstupom je logická 1. Funkcia isnumeric vráti logickú 1, ak je na vstupe numerický dátový typ. Numerickým dátovým typom v systéme MATLAB rozumieme niektorý z dátových typov: int8, int16, int32, int64, uint8, uint16, uint32, uint64, single a double.

```
if isempty(funkcia)
  DT = {};
  return
end
```

Obrázok 11 – Ukážka použitia funkcie isempty v algoritme.

```
if ((isempty(strfind(t_vyraz,'t')))&& (isnumeric(eval(vyraz))))
    subs_ch=[num2str(eval(vyraz)) '*delta(k)'];
```

Obrázok 12 – Ukážka použitia funkcie isnumeric v algoritme.

```
if (~isequal(char(ch(1)),'t'))&& (isnumeric(eval(ch(2))))
    for ii=1:eval(ch(2))
        sucinitele=[sucinitele ch(1)];
    end
elseif (isequal(char(ch(1)),'t'))&& (isnumeric(eval(ch(2))))
    sucinitele=[sucinitele rozklad(j)];
end
```

Obrázok 13 – Ukážka použitia funkcie isequal v algoritme.

Funkcie **char**, **num2str**, **sym2poly** a **cell2mat** slúžia pre pretypovanie medzi dátovými typmi. Výstupom funkcie char je vektor znakov. Funkcia num2str konvertuje číslo na pole znakov, funkcia sym2poly vypíše vektor všetkých numerických koeficientov vrátane núl zo symbolického polynómu. Funkcia cell2mat konvertuje pole buniek na obyčajné pole pôvodného dátového typu.

```
for i=1:pocet_vyrazu
   vyraz=vyrazy(i);
   t_vyraz=char(vyraz);
```

Obrázok 14 – Ukážka použitia funkcie char v algoritme.

```
tsubs=['U(k + ' num2str(rad+1) ')'];
```

Obrázok 15 – Ukážka použitia funkcie num2str v algoritme.

```
pol=sym2poly(argument_funkcie);
```

Obrázok 16 – Ukážka použitia funkcie sym2poly v algoritme.

```
hladaj_mocninu=cell2mat(strfind(t_funkcia,'^'));
```

Obrázok 17 – Ukážka použitia funkcie cell2mat v algoritme.

Funkcie **strfind** a **strrep** pracujú s textovými reťazcami. Výstupom funkcie strfind je vektor, ktorý obsahuje čísla reprezentujúce počet znakov pred hľadaným výrazom od začiatku reťazca. Vstupom do funkcie strrep sú tri textové reťazce a to konkrétne: textový reťazec v ktorom prebieha vyhľadávanie a náhrada, textový reťazec, ktorý bude nahradený a textový reťazec, ktorým sa nahradí hľadaný reťazec. Výstupom je textový reťazec po náhrade.

```
hladaj_sin=cell2mat(strfind(t_funkcia,'sin'));
```

Obrázok 18 – Ukážka použitia funkcie strfind v algoritme.

```
nova_eqn=eval(strrep(tfeqn,tsubs,'x'));
```

Obrázok 19 – Ukážka použitia funkcie strrep v algoritme.

Funkcia **solve** vyrieši rovnicu na vstupe do funkcie podľa premennej, ktorá je tiež na vstupe funkcie.

```
U(rad+1+k)=solve(nova_eqn,x);
```

Obrázok 20 – Ukážka použitia funkcie solve v algoritme.

Funkcia **eval** vyhodnotí výraz v textovom reťazci, ktorý je na vstupe funkcie. Napríklad ak na vstup funkcie eval vložíme výraz „sin(y)“, funkcia vyhodní „y“ ako premennú a ak v nej bola zadaná hodnota, tak na výstupe funkcie bude sínus danej hodnoty.

```
if ((isempty(strfind(t_vyraz,'t')))&& (isnumeric(eval(vyraz))))
    subs_ch=[num2str(eval(vyraz)) '*delta(k)'];
```

Obrázok 21 – Ukážka použitia funkcie eval v algoritme.

Funkcia **length** zisťuje dĺžku vektora. Na jej výstupe je číslo reprezentújuce počet prvkov vo vektore.

```
l=length(sucinitele);
```

Obrázok 22 – Ukážka použitia funkcie length v algoritme.

Funkcia **factorial** vracia číslo, ktoré reprezentuje faktoriál čísla na vstupe funkcie.

```
fl=1./factorial(0:(rad-1));
```

Obrázok 23 – Ukážka použitia funkcie factorial v algoritme.

Funkcia **zeros** vracia maticu, ktorej všetky prvky sú rovné nule. Rozmer výslednej matice určuje vstup funkcie. Ak je na vstupe iba jedna hodnota, vytvorí sa štvorcová matica.

```
U=[fl.*podminky zeros(1,r-rad)];
```

Obrázok 24 – Ukážka použitia funkcie zeros v algoritme.

ZÁVER

Cieľom tejto bakalárskej práce bolo objasniť problematiku metódy diferenciálnej transformácie a implementovanie danej metódy v systéme MATLAB. Podarilo sa mi tiež priblížiť viaceré numerické metódy riešenia diferenciálnych rovníc, konkrétne Eulerovu metódu a metódy Runge-Kutta. Postup pri výpočte s použitím metódy diferenciálnej transformácie sa mi podarilo ukázať na piatich riešených príkladoch. Dva príklady boli obyčajnými diferenciálnymi rovnicami druhého rádu, zvyšné tri príklady prvého rádu. Tým som objasnil princíp fungovania DTM pri obyčajných diferenciálnych rovniciach vyšších rádov. Jeden z riešených príkladov bol ukážkou riešenia nelineárnych diferenciálnych rovníc pomocou DTM.

Pri vytváraní algoritmu som vychádzal z matematických definícií pre metódu diferenciálnej transformácie. K porovnaniu výsledkov najlepšie poslúžili grafy a tabuľky, kde sú rozdiely vo výsledkoch jasne viditeľné. Z výsledkov som dokázal, že Eulerova metóda je síce najrýchlejšia, avšak je najmenej presná. Rýchlosť výpočtu pomocou DTM značne ovplyvňuje žiadaná presnosť výpočtu. Pri detailnejšej optimalizácii algoritmu sa dá očakávať, že by metóda DTM bola podstatne rýchlejšia. Funkcia ode45, pracujúca na základe metódy Runge-Kutta 4. rádu, dosahuje výbornú presnosť a je tiež veľmi rýchla.

Systém MATLAB je výbornou pomôckou pri podobných výpočtoch. Má veľké množstvo implementovaných funkcií, ktoré sú užitočné pri práci so symbolickými výrazmi. Na webovej stránke spoločnosti Mathworks sa tiež nachádza rozsiahla dokumentácia k systému MATLAB spolu s užívateľským fórom, čo rovnako prispieva k zjednodušeniu práce pri riešení danej problematiky. Implementácia algoritmu DTM do programovacích jazykov nepodporujúcich symboliku by teda bola náročná, keďže by bolo nutné naprogramovať aj celú logiku vyhodnocovania symbolických výrazov. Možná by bola implementácia algoritmu DTM do systému Mathematica. Tento systém rovnako ako systém MATLAB dokáže pracovať so symbolickými výrazmi, avšak myslím si že príkazová syntax v systéme MATLAB je podstatne jednoduchšia a užívateľsky prístupnejšia.

ZOZNAM POUŽITEJ LITERATÚRY

[1] BEZÁKOVÁ, Anna. *Diferenciálne rovnice*. Zvolen: Matcentrum, 2001. ISBN 80-968057-9-7.

[2] BORŮVKA, Otakar. Diferenciální rovnice v rámci dějin matematiky: Plenární přednáška na Konferenci československých matematiků. *Matematické obzory* [online]. Ostrava, 1977, 1974, **1977**(11), 1-10. Dostupné z: http://hdl.handle.net/10338.dmlcz/500267

[3] BUTCHER, J. C. *Numerical methods for ordinary differential equations*. Hoboken, NJ: J. Wiley, c2003. ISBN 0-471-96758-0.

[4] DOLEŽALOVÁ, Jarmila. *Diferenciální rovnice*. Dostupné také z: http://homen.vsb.cz/~kre40/matem2/prehled-difrovnice.pdf

[5] HALABRÍN, Marián. *Lineárna algebra*. Bratislava: Slovenská technická univerzita v Bratislave, 2004. Edícia skrípt. ISBN 80-227-2126-3.

[6] HOROVÁ, Ivana a Jiří ZELINKA. *Numerické metody*. 2., rozš. vyd. Brno: Masarykova univerzita v Brně, 2004. ISBN 80-210-3317-7.

[7] HUTNÍK, PHD., doc. RNDr. Ondrej. *Matematická analýza II*. Dostupné také z: https://umv.science.upjs.sk/analyza/texty/predmety/MANb/MANb16.pdf

[8] KUBÍČEK, Milan, Miroslava DUBCOVÁ a Drahoslava JANOVSKÁ. *Numerické metody a algoritmy*. Vyd. 2., opr. Praha: Vysoká škola chemicko-technologická, 2005. ISBN 80-7080-558-7.

[9] MAŘÍK, Robert. *Tayloruv polynom* [online]. Dostupné z: http://user.mendelu.cz/marik/mat-web/mat-webse4.html

[10] REBENDA, J. *An application of bell polynomials in numerical solving of nonlinear differential equations: 17th Conference on Applied Mathematics*. APLIMAT 2018, 2018, s. 891-900. ISBN 978-802274765-3.

[11] RŮŽIČKOVÁ, Mgr. Irena a RNDr. Rudolf HLAVIČKA, CSC. Numerické metody [online]. Brno: Vysoké učení technické v Brně. Dostupné z: http://physics.ujep.cz/~jskvor/NME/DalsiSkripta/Numerika.pdf

[12] ŘEZNÍČKOVÁ, Jana. *Diferenciální rovnice* [online]. Zlín: Ústav matematiky FAI UTB ve Zlíně, 2015. Dostupné z: http://msc.utb.cz/wp-content/uploads/2016/11/A3DRI_stud_text.pdf

[13] VYBÍRAL, Bohumil a Miroslava JAREŠOVÁ. *Diferenciální rovnice* [online]. Dostupné z: http://fyzikalniolympiada.cz/texty/matematika/difro.pdf

[14] ZAPLATÍLEK, Karel a Bohuslav DOŇAR. *MATLAB pro začátečníky*. 2. vyd. Praha: BEN - technická literatura, 2005. ISBN 80-7300-175-6.

[15] *Aproximace funkcí polynomy*. Dostupné také z: http://people.fjfi.cvut.cz/pelanedi/taylorN.pdf

[16] *Differential Transformation Method* [online]. Brno: Department of Mathematics FEEC BUT, 2014. Dostupné z: http://www.umat.feec.vutbr.cz

[17] *Taylorův polynom a chyba odhadu*. Dostupné také z: http://www.math.muni.cz/~xjuranekj/mb102/Taylor.pdf

ZOZNAM POUŽITÝCH SYMBOLOV A SKRATIEK

a pod.	a podobne
atď.	a tak ďalej
DR	diferenciálna rovnica
DTM	differential transform method
ODE	ordinary differential equation
napr.	napríklad
resp.	respektíve
t. j.	to jest
tzn.	to znamená
tzv.	takzvané

ZOZNAM OBRÁZKOV

ZOZNAM TABULIEK

ZOZNAM PRÍLOH

DT	Zdrojový kód pre preklad rovnice podľa pravidiel DTM.
DTM	Zdrojový kód pre vyhodnocovanie výrazov a konečný výpočet.
kd	Zdrojový kód pre Kroneckerovo delta.
Príklad1_euler	Zdrojový kód pre výpočet 1. riešeného príkladu pomocou Eulerovej metódy.
Príklad1_ode45	Zdrojový kód pre výpočet 1. riešeného príkladu pomocou funkcie ode45.
Príklad1_presne	Zdrojový kód pre výpočet 1. riešeného príkladu pomocou presného riešenia.
Príklad2_euler	Zdrojový kód pre výpočet 2. riešeného príkladu pomocou Eulerovej metódy.
Príklad2_ode45	Zdrojový kód pre výpočet 2. riešeného príkladu pomocou funkcie ode45.
Príklad2_presne	Zdrojový kód pre výpočet 2. riešeného príkladu pomocou presného riešenia.
Príklad3_euler	Zdrojový kód pre výpočet 3. riešeného príkladu pomocou Eulerovej metódy.
Príklad3_ode45	Zdrojový kód pre výpočet 3. riešeného príkladu pomocou funkcie ode45.
Príklad3_presne	Zdrojový kód pre výpočet 3. riešeného príkladu pomocou presného riešenia.
Príklad4_euler	Zdrojový kód pre výpočet 4. riešeného príkladu pomocou Eulerovej metódy.
Príklad4_ode45	Zdrojový kód pre výpočet 4. riešeného príkladu pomocou funkcie ode45.
Príklad4_presne	Zdrojový kód pre výpočet 4. riešeného príkladu pomocou presného riešenia.
Príklad5_euler	Zdrojový kód pre výpočet 5. riešeného príkladu pomocou Eulerovej metódy.
Príklad5_ode45	Zdrojový kód pre výpočet 5. riešeného príkladu pomocou funkcie ode45.

Príklad5_presne	Zdrojový kód pre výpočet 5. riešeného príkladu pomocou presného riešenia.
product	Zdrojový kód pre prácu s členom rovnice obsahujúcim dva výrazy.
product3	Zdrojový kód pre prácu s členom rovnice obsahujúcim tri výrazy.