BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

*Department of Intelligent Systems*

# STIFF SYSTEMS OF DIFFERENTIAL EQUATIONS AND MODERN TAYLOR SERIES METHOD

# STIFF SYSTÉMY DIFERENCIÁLNÍCH ROVNIC A MODERNÍ METODA TAYLOROVA ROZVOJE

SHORT VERSION OF PH.D. THESIS

by David Řezáč
M.Sc., Brno University of Technology, 2001

| | |
|---|---|
| Study field: | Information technology |
| Supervisor: | Doc. Ing. Jiří Kunovský, CSc. |
| Opponents: | Prof. Ing. Ivo Serba, CSc.<br>Doc. Ing. Miroslav Přeučil, DrSc. |
| Presentation date: | October 20$^{\text{th}}$, 2004 |

## Keywords

Stiff systems detection, solving stiff systems, Modern Taylor series method, TKSL.

## Klíčová slova

Detekce stiff systémů, řešení stiff systémů, Moderní metoda Taylorova rozvoje, TKSL.

## Archive

Department of Intelligent Systems, Faculty of Information Techonology, Brno University of Technology, Božetěchova 2, 612 66 Brno


ISBN 80–214–2783–3
ISSN 1213–4198

Department of Intelligent Systems
Faculty of Information Techonology
Brno University of Technology

# Contents

# 1 Introduction

This thesis deals with numerical solution of stiff systems of ordinary differential equations (ODE), which appear often in simulation of continuous dynamic systems. In such a simulation (of electronic circuits, chemical reactions, physical phenomenons etc.) a system of differential equations is used to create an abstract mathematical model of the examined system.

However common in practice the stiff systems are, they stay in the background of the interest. It is a real challenge to collect up-to-date information on the topic: the literature mentions this problem marginally, there are few articles scattered in proceedings and journals.

## 1.1 Stiff systems

The actual definition of stiffness differs in many publications. There are vague definitions like "problems for which explicit methods don't work" [14] as well as other definitions that try to specify the characteristics in a more precise way. One of the most frequently mentioned definition is from [23]:

**Definition** Let

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}, t) \qquad \mathbf{y}(0) = \mathbf{y_0} \tag{1.1}$$

be a system of $n$ ordinary differential equations. Let $\mathcal{J}$ be the Jacobian of the (1.1) and $\lambda_i$ the eigenvalues of $\mathcal{J}$. The eigenvalues $\lambda_i$ are generally time-dependent.

A stiff system stable on the integration interval $I$ is characterized by the following features:

1. $\mathrm{Re}\lambda_i < 0$ for $i = 1, \ldots, n$ and $t \in I$
2. $\max_i |Re\lambda_i| \geq \min_i |Re\lambda_i|, \quad i = 1, \ldots n$

Let the eigenvalues $\lambda_i$ be arranged so as $|\mathrm{Re}\lambda_{\max}| \geq |\mathrm{Re}\lambda_i| \geq |\mathrm{Re}\lambda_{\min}|$, where $i = 1, \ldots, n$. The *stiffness ratio*

$$r = \left| \frac{\mathrm{Re}\lambda_{\max}}{\mathrm{Re}\lambda_{\min}} \right| \tag{1.2}$$

is a coefficient that helps to decide whether a problem is stiff or not. A higher $r$ indicates a more stiff system.

However, there is no exact value of the stiffness ratio $r$ that would distinguish the non-stiff problems from the stiff-problems. For many problems in common practice the stiffness ratio $r$ "is very high" (say $1 \cdot 10^6$ or higher).

### 1.1.1 Features of a stiff system

1. Let us consider the case we have to find out the steady state of system (1.1). A small $|\mathrm{Re}\lambda_{\min}|$ implies a longer integration interval the system has to be solved on, because $|\mathrm{Re}\lambda_{\min}|$ determines the speed the slowest component of the solution is damped with.
2. However, in order to keep the stability of the computation a reasonably short integration step $h$ has to be used – so as $|h\lambda_{\max}| \leq K$ where $K$ is a constant dependent on both the method and the problem. This may impose immense restrictions on the size of the integration step $h$.

## 1.2 Stiffness Detection Methods

It is important that the solver is equipped with a cheap means of detecting stiffness so as an explicit method does not waste too much effort when encountering stiffness and to enable a switch to a more suitable method. There are two basic methods of stiffness detection shown here.

**Eigenvalues of Jacobian** The basic method uses the definition of stiffness directly. That means the Jacobian $\mathcal{J}$ of the system has to be computed and its eigenvalues found. Having $\lambda_{\min}$ and $\lambda_{\max}$ we can now compute the stiffness ratio and take a proper action.

The great disadvantage of this method is the evaluation of the eigenvalues of $\mathcal{J}$. This operation is very computational power consuming. When it comes to large scale problems it is very difficult to perform the computation in a reasonable time. The other thing is that the computation has to be done repeatedly when solving a nonlinear problem.

**Special Error Estimator** A reliable but not too costly stiffness detection technique is the method of the "Low-order comparison formula pair" [12, 14, 21]. The idea is to use a comparison formula pair in addition to the basic formula pair during the integration process. The two error estimates obtained from the two formulas are used to assess the degree of stiffness in the problem.

In the context of Runge-Kutta methods the comparison formula pair is designed to use the same stages as the basic formula, thus no extra function evaluations are required. The comparison formula has lower order of accuracy than the basic formula, hence when the problem is nonstiff, it is expected to yield a larger error estimate than the basic formula pair. However, if the comparison formula pair has a

larger stability region than the basic formula pair, we would expect its error estimate to become smaller than that of the basic formula pair when the problem is stiff.

The prevalent methods have a common unpleasant feature – they are very expensive in the terms of computational costs and are seldom applicable to large scale stiff systems of ODE's. The stiffness detection process usually consists in computing the extreme eigenvalues of the Jacobian of the system. This is acceptable only in the case the examined system is of a low order.

Apart from the naive and inapt method of direct computation of stiffness ratio $r$ of the system, there are methods that try to harness the intermediate results of the particular ODE method. This often requires to integrate additional formulas into the solver. Thus the detection algorithms are often ODE method-dependent.

## 1.3 Solving Stiff Systems

Once the system is declared to be stiff and it is decided that the stiff-solver would be more suitable one should choose a convenient method to use in further computation.

Recalling the fact that the explicit methods are not usable for the stiff systems (it is one of the definitions of the stiffness) only implicit methods are considered further. It is obvious that there are many various stiff-solvers. Those interested are encouraged to have a look in a bibliography, starting probably with [13, 14].

Some methods require an autonomous system of ODE's, i.e. a system that does not explicitly contain the independent variable (it does not depend on time). A non-autonomous system (1.1) can be converted to autonomous form by adding a new equation $t' = 1$ with initial condition $t_0 = 0$.

$$\mathbf{y}' = \mathbf{f}(\mathbf{y}) \qquad \mathbf{y}(0) = \mathbf{y_0} \tag{1.3}$$

**Partitioning Method** One of the oldest ideas [14] is to partition a stiff system (1.3) into a (hopefully) small stiff system and a large nonstiff part

$$\begin{array}{rcll} y_a' & = & f_a(y_a, y_b) & (\text{stiff}) \\ y_b' & = & f_b(y_a, y_b) & (\text{nonstiff}) \end{array} \tag{1.4}$$

so that the two systems can be treated by two different methods, one implicit and the other explicit. The obvious difficulty of this approach is to decide which equations should be the stiff ones.

**Implicit Runge-Kutta Methods** (IRK Methods) The Runge-Kutta methods are very favored and often used. More information on the IRK methods – their con-

struction, error estimation and stability issues – are discussed for example in [14, 8, 9, 4, 3, 6, 5, 7, 2, 1].

The general formula of implicit Runge-Kutta method of $s-$stages is

$$\begin{aligned} k_i &= f\left(t_n + c_i h, y_n + h\sum_{j=1}^{s} a_{ij}k_j\right) \qquad i = 1, \ldots, s \\ y_{n+1} &= y_n + h\sum_{i=1}^{s} b_i k_i \end{aligned} \tag{1.5}$$

There is also a tabular notation – the Butcher tableau. An $s-$stage IRK in this notation is shown in table 1.

$$\begin{array}{c|c} c & A \\ \hline & b^T \end{array} \quad = \quad \begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \cdot\ \cdot & a_{1s} \\ c_2 & a_{21} & a_{22} & \cdot\ \cdot & a_{2s} \\ \cdot & \cdot & \cdot & \cdot\ \cdot & \cdot \\ c_s & a_{s1} & a_{s2} & \cdot\ \cdot & a_{ss} \\ \hline & b_1 & b_2 & \cdot\ \cdot & b_s \end{array}$$

Table 1: General IRK method in Butcher tableau notation

Obviously the coefficients $b_i$, $c_i$, $a_{ij}$ in (1.5) have to meet several conditions so as the method is usable. The details are stated for example in [14].

The solution of a general system of $n$ ODE's using an $s-$stage IRK method requires the solution of $s \cdot n$ generally non-linear equations to determine the stage derivatives $k_i$ of (1.5). For large ODE systems the solution of this system is usually very costly. In order to reduce this cost several modification of the general IRK were created.

A method with $a_{ij} = 0$ for $i < j$ and at least one $a_{ii} \neq 0$ is called diagonal implicit Runge-Kutta method (DIRK). If in addition all diagonal elements are identical ($a_{ii} = \gamma$ for $i = 1, \ldots, s$) we speak of singly diagonal implicit Runge-Kutta method (SDIRK) – see table 2.

An example of well-known IRK methods is the implicit Euler method

$$y_{n+1} = y_n + hf(t_{n+1}, y_{n+1}) \tag{1.6}$$

**General Linear Methods** An unifying framework to both one-step and multistep methods provides the concept of general linear methods. General linear methods were introduced (refer to [6, 24]) as a unifying framework for the traditional methods to study the properties of consistency, stability and convergence and to formulate new methods with clear advantages over the traditional methods.

| | | | | | |
|---|---|---|---|---|---|
| $c_1$ | $a_{11}$ | $0$ | $0$ | $\dots$ | $0$ |
| $c_2$ | $a_{21}$ | $a_{22}$ | $0$ | $\dots$ | $0$ |
| $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ | $\vdots$ |
| $c_s$ | $a_{s1}$ | $a_{s2}$ | $a_{s3}$ | $\dots$ | $a_{ss}$ |
| | $b_1$ | $b_2$ | $b_3$ | $\dots$ | $b_s$ |

| | | | | |
|---|---|---|---|---|
| $c_1$ | $\gamma$ | | | |
| $c_2$ | $a_{21}$ | $\gamma$ | | |
| $\vdots$ | $\vdots$ | $\ddots$ | | |
| $c_s$ | $a_{s1}$ | $a_{s2}$ | $\dots$ | $\gamma$ |
| | $b_1$ | $b_2$ | $\dots$ | $b_s$ |

Table 2: Butcher tableau for DIRK and SDIRK methods.

A general linear method used for numerical solution of an autonomous system of ODE's (1.3) is both multistage and multivalued. Denote the internal stage values of step number $n$ by $Y_1^{[n]}, Y_2^{[n]}, \dots, Y_s^{[n]}$ and the derivatives evaluated at these steps by $f(Y_1^{[n]}), f(Y_2^{[n]}), \dots, f(Y_s^{[n]})$. At the start of step $n$, there are $r$ quantities denoted by $y_1^{[n-1]}, y_2^{[n-1]}, \dots, y_r^{[n-1]}$ available from approximations computed in step $n-1$. Corresponding quantities $y_1^{[n]}, y_2^{[n]}, \dots, y_r^{[n]}$ are evaluated in the step $n$.

A general linear method is then characterized by four matrices which can be arranged in a partitioned $(s+r) \times (s+r)$ matrix as follows

$$M = \left[\begin{array}{c|c} A & U \\ \hline B & V \end{array}\right] \tag{1.7}$$

The stage values $Y_i$ and outgoing values $y_i$ from step number $n$ are defined by

$$\begin{aligned} Y_i &= h\sum_{j=1}^{s} a_{ij} f(Y_j^{[n]}) + \sum_{j=1}^{r} u_{ij} y_j^{[n-1]} \quad i = 1, 2, \dots, s \\ y_i^{[n]} &= h\sum_{j=1}^{s} b_{ij} f(Y_j^{[n]}) + \sum_{j=1}^{r} v_{ij} y_j^{[n-1]} \quad i = 1, 2, \dots, r \end{aligned} \tag{1.8}$$

General linear methods can be represented by the following string of numbers, $pqrs$, where $p$ is the order of the methods, $q$ is the stage order, $r$ is the number of output approximations and $s$ is the number of internal stages. The construction of various $pqrs-$methods is shown for example in [10].

## 1.4 Implementation issues

When a numerical method is constructed it is important to compare it with existing numerical solvers. Such a comparison makes possible to verify the advantages and disadvantages the method is claimed to have. Before doing so, it is essential to discuss certain choices regarding how the method is implemented. The five main issues with regards to implementation of the IRKs methods (they are focused here because of their wide usage) are:

- *Starting procedure*
  One-step methods are easy to use. The procedure is all the same during the computation from the very beginning. However, the situation is quite different in the case of multistep methods. The implementation of multistep methods has usually two parts – the starting procedure which provides $y_1, \ldots, y_{k-1}$ (approximations to the exact solution at the points $t_0 + h, \ldots, t_0 + (k-1)h$) and, secondly, a multistep formula to obtain an approximation to the exact solution $y(t_0 + kh)$. This is then applied recursively, based on the numerical approximations of $k$ successive steps, to compute $y(t_0 + (k+1)h)$, etc. Another problem is stopping because the equal steps are unlikely to land directly on the desired termination point.

- *Error estimation*
  It must be possible to estimate the local truncation error for any general linear method to be implemented in an adaptive fashion, as this allows a measure of how accurate the approximations are, and how much the stepsize should be varied. The formula representing the truncation error differs from method to method. The principal term of the local truncation error is calculated for the sake of usability usually.

- *Variable stepsize*
  It is obvious that the step size selection depends on the current step error and the predefined acceptable error range. With a given initial stepsize the solver code must be capable to select the next stepsize automatically to gain as much efficiency as possible.

- *Variable order*
  It is well known that to obtain high efficiency of the computational process not only the stepsize is varied but also the order of the method. To appreciate this statement one looks at possible problems which occur with fixed order methods.

  Rigorous instructions on how to change the order of the used methods are presented for example in [13, 14, 2, 24]. However, [2] offers an algorithm expressed in both mathematical and human words. The advices are:

  - The order multiplied by the error per work done should be maintained constant from step to step.
  - The order $p$ selected for any step should be the one which minimizes the product of $p + 1$ and the error per unit step.

- *Computing the stages*
  Implicit methods are much more complicated to use than explicit methods,

because the stages are defined implicitly. It is generally regarded, that to solve the stages accurately and efficiently the modified Newton's method should be used. The modified Newton's method uses the Jacobian matrix $\mathcal{J}$ evaluated at the beginning of the step for each iteration in one step. Often to further increase efficiency the same Jacobian is used over several steps.

There are many (implicit) methods for solving stiff systems of ODE's, from the most simple such as implicit Euler method to more sophisticated (implicit Runge-Kutta methods) and finally the general linear methods. The mathematical formulation of the methods often looks clear, however the implicit nature of those methods implies several implementation problems. Usually a quite complicated auxiliary system of equations has to be solved in each step. The modified versions of Newton algorithm is employed in common practice. Another computational costs are caused by helping transformations (such as LU factorization) used during the computation. Both the stepsize and method order should be controlled to achieve good results, too. These facts lead to immense amount of work to be done in each step of the computation.

That is why on has to judge well the actual necessity of using stiff solver.

# 2 Modern Taylor Series Method

The Modern Taylor Series Method (MTSM) is the cornerstone for this thesis. It is an original mathematical method which uses the Taylor series method for solving differential equations in a non-traditional way. Even though this method is not much preferred in the literature, experimental calculations done at the Department of Intelligent Systems of the Faculty of Information Technology of TU Brno have shown and theoretical analyses at the Department of Mathematics of the Faculty of Electrical Engineering and Communication of TU Brno have verified that the accuracy and stability of the MTSM exceeds the currently used algorithms for numerically solving differential equations. It has been verified that the computation quite naturally uses the full hardware accuracy of the computer and is not restricted to the usual accuracies of $10^{-5}$ to $10^{-6}$.

The main idea behind the MTSM is an automatic integration method order setting, i.e. using as many Taylor series terms for computing as needed to achieve the required accuracy [15]. The $\mathrm{p}^{\mathrm{th}}$ order method (ORD=$p$) uses Taylor series terms up to the $\mathrm{p}^{\mathrm{th}}$ power of the integration step $h$.

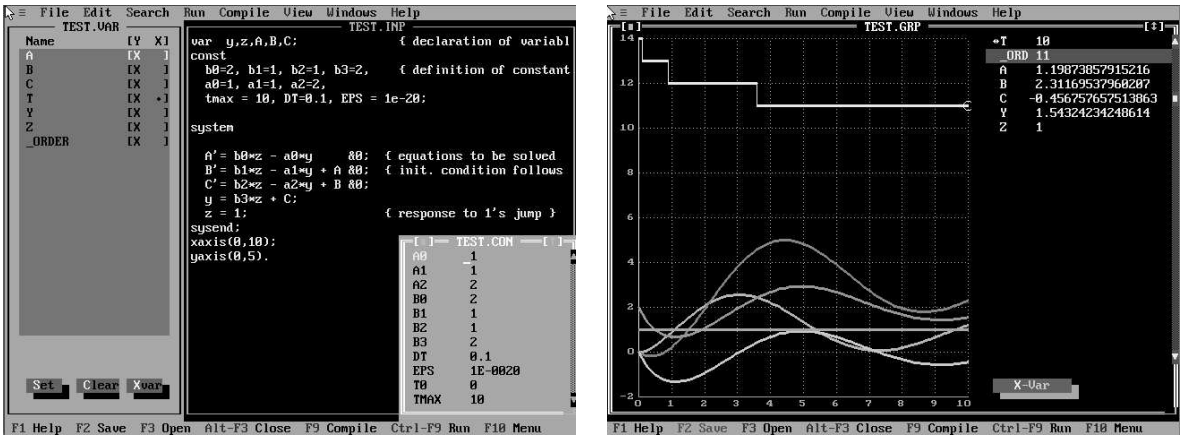
The "Modern Taylor Series Method" also has some properties very favourable for parallel processing. Many calculation operations are independent making it pos-

sible to perform the calculations independently using separate processors of parallel computing systems [19, 11]. This parallel approach has been tested using the available parallel transputer system – an original methodology for parallel computation of systems of differential equations in a transputer system has been defined.

Since the calculations of the transformed system (after the automatic transformation of the initial problem) use only the basic mathematical operations ($+,-,\times,\div$), simple specialized elementary processors can be designed for their implementation thus creating an efficient parallel computing system with a relatively simple architecture [19].

MTSM been successfully applied to a number of various problems [15]:

- Systems of linear algebraic equations (including detection of linear dependency) [17, 15]
- Partial differential equations (parabolic, elliptical, hyperbolic) [18]
- Definite integrals [15]
- Fourier series coefficients [15]
- Algebraic and transcendental equations [16, 15]




(a) User interface, editor

(b) User interface, result browser

Figure 1: TKSL/386

## 2.1 TKSL/386

The simulation system TKSL/386 has been created in order to test algorithms which utilize MTSM for solving the sets of differential equations. The user friendly environment TurboVision has been used (Fig 1(a)). User is able to control the accuracy

of the computation, the order of the method and some other parameters of the computation. The system is able to detect discontinuities precisely. The integration step changes during the computation.

The program features an integrated editor, a compiler of the equations, a simulation engine and result displayer – all in one program (Fig. 1(a) and 1(b)). Once you successfully compile the source code which describes the system of equations it is possible to start the simulation. The results are plotted into a graph which you can explore using the cursor. The values of solution can be read in this way.

# 3 Goals

The availability of the "Modern Taylor series method" and the presence of the stiff systems of ordinary differential equations lead to a straightforward formulation of the problems to solve.

- **To find a cheap way of *detection of stiff systems* of ordinary differential equations using Modern Taylor series method.**

  The commonly used methods of stiffness detection are quite complicated and quite expensive in the terms of computational power. It is especially true in case of large systems because many methods use eigenvalues of the Jacobian $\mathcal{J}$ of the system. The goal is to utilize MTSM as much as possible during the stiffness detection.

- **To adapt the Modern Taylor series method so as it is able to *solve stiff systems* of ordinary differential equations.**

  There are several methods theoretically suitable for the stiff systems. Unfortunately, because of their complexity and high specialization few of them appear in widely spread applications. Large systems (as they commonly appear for example in simulations of electrical circuits) usually cause more problems. The goal is to suggest a method based on MTSM which makes it possible to deal with stiff systems in a reasonable way. It should be as much consistent with the MTSM as it is suitable for hardware implementation and this 'improvement' would extend the area of usage of the method.

The Modern Taylor series method is used to achieve the goals because of its unique features: it is both very fast and accurate even when dealing with large systems of ordinary differential equations. Such systems appear very often in simulations. The stiff systems are a hard nut to crack and the potential MTSM promised to be the right tool to solve the problem.

# 4 Test Examples

In order to be able to compare several different numerical methods suitable for solving stiff systems a set of problems has been compiled. This test set consists of equation proposed in [13, 14, 15, 23, 20]. Some other interesting problems have been added to this set, too.

As there are stiff systems of various nature the test set tries to cover the most of the types so as one can judge the most suitable method for his needs. The selected problems are taken from the common practice – they represent physical or chemical phenomenons, simulation of electrical circuits etc.

Both linear and nonlinear examples were examined. Among them were well-known stiff problems such as the reaction of Robertson, Belusov–Zhabotinky reaction, reaction-diffusion (brusselator) or van der Pol oscillator. There are only two examples introduced in this short version of the thesis.

**Linear problem**
An example of linear stiff problem is system:

$$\begin{aligned} y' &= -2.7 \cdot 10^6 y + 2.7 \cdot 10^6 z + 1.08 \cdot 10^6 \\ z' &= -3.5651205 z + 19.60816275 \end{aligned} \tag{4.9}$$

with initial conditions $y(0) = 4.2$ and $z(0) = 0.3$ is shown in Fig. 2. The solution of $y$ drops rapidly at first but then it copies the solution of $z$.

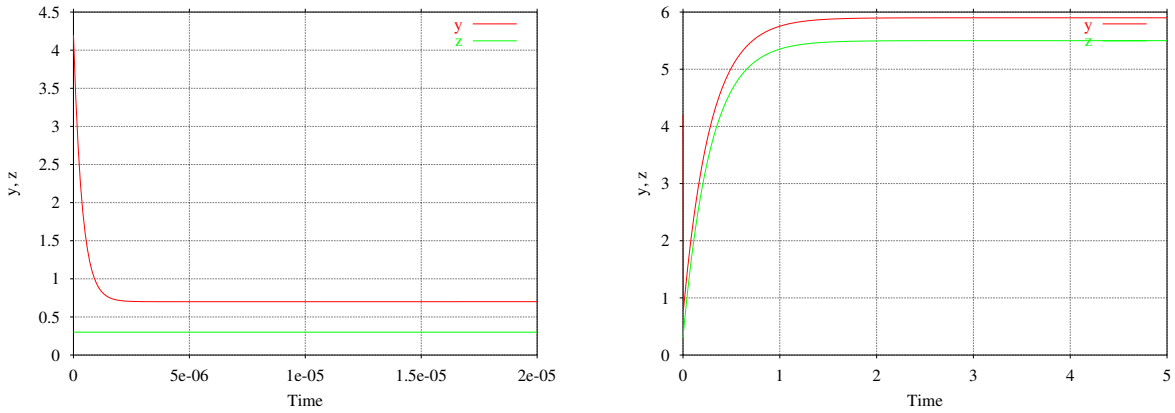



Figure 2: Linear problem, solution of (4.9)

**Van der Pol Oscillator**
The van der Pol problem is described by system

$$\begin{aligned} y_1' &= y_2 \\ y_2' &= ((1 - y_1^2) y_2 - y_1)/\varepsilon, \quad \varepsilon > 0 \end{aligned} \tag{4.10}$$

The solution of (4.10) for initial conditions $y_1(0) = 2$, $y_2(0) = 0$ and parameter $\varepsilon = 10^{-6}$ is shown in Fig. 3.

The significant feature of (4.10) is that the small oscillations are amplified and the large oscillations are damped. The coefficient $\varepsilon$ influences the stiffness of the system.

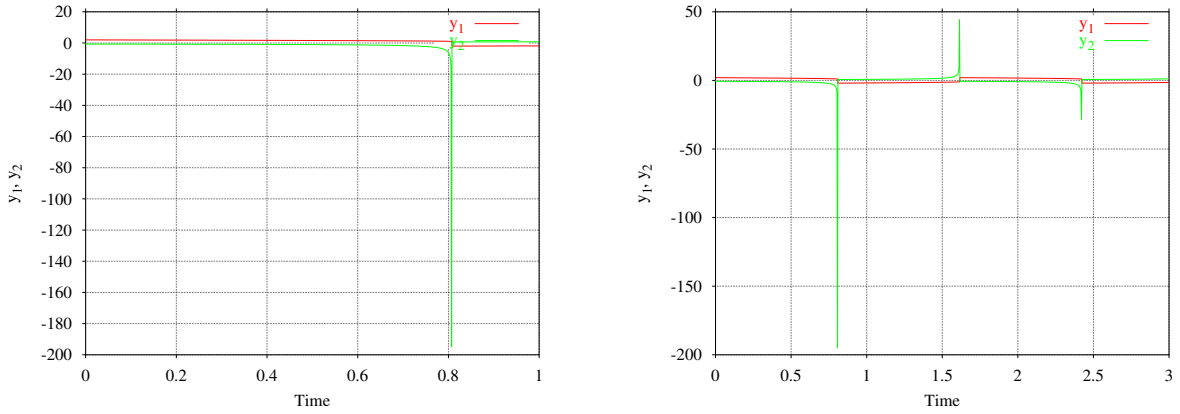



Figure 3: Van der Pol oscillator, solution of (4.10)

# 5 Detecting Stiff Systems Using Modern Taylor Series Method

There are experiments with the famous van der Pol system in TKSL in the background of the idea. The striking fact is that the used order of MTSM varies on the integration interval. The more noticeable thing is that the changes in $\varepsilon$ coefficient of the system (which affects its stiffness) impacted the number of used Taylor series terms used, too.

The situation is shown in Fig. 4. The integration step is fixed $h = 0.01$ in all cases. In order to meet the precision requirements, the MTSM is forced to use more Taylor series terms (see the line labeled as "ORD"). The usage of the Taylor series terms is enormous on intervals where the solution changes rapidly. This is especially true if the local stiffness ratio of the system is high.

### Mathematical background

Let $\mathcal{J}$ be the Jacobian of system (1.1) and $\lambda_i(t) = \lambda_i^t$ its eigenvalues in time $t$ (local eigenvalues). We are able to compute the local stiffness ratio $r_t = \left|\frac{\mathrm{Re}\lambda^t_{\max}}{\mathrm{Re}\lambda^t_{\min}}\right|$.

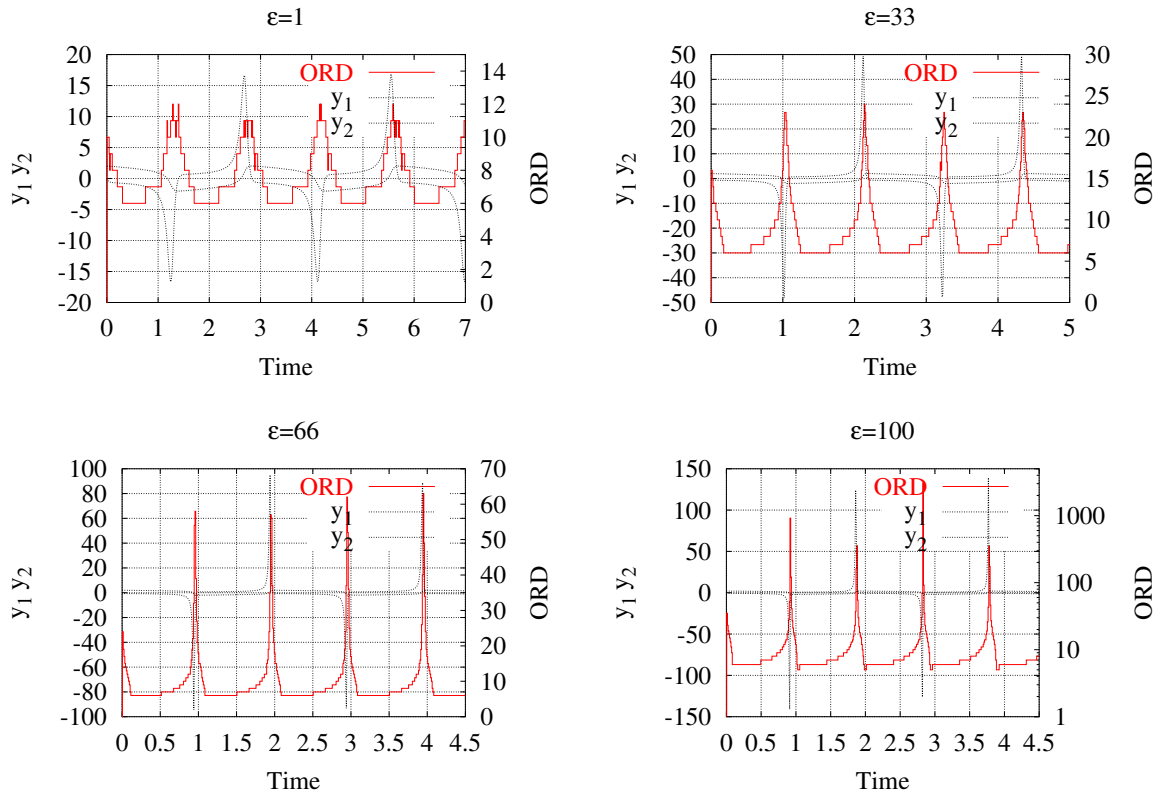



Figure 4: Order of the MTSM used depending on stiffness of system (4.10)

Consider now a linear system of ODE's

$$\mathbf{z}' = \mathbf{g}(t, \mathbf{z}) \qquad \mathbf{z}(t_0) = \mathbf{z_0} \tag{5.11}$$

of $m$ equations such that its eigenvalues are equal to the local eigenvalues $\lambda_i^t$ of the $\mathcal{J}$ which belongs to the system (1.1). The solution of (5.11) is

$$\mathbf{z}(t) = \sum_{i=1}^{m} c_i e^{\lambda_i t} \mathbf{u_i} + \mathbf{p}(t) \tag{5.12}$$

where $\mathbf{p(t)}$ is the particular solution and $\mathbf{u_i}$ is the eigenvector corresponding to the negative eigenvalue $\lambda_i$.

The crucial are the solutions with $\lambda_{\max}$ and $\lambda_{\min}$ – the fast and the slow component of the solution. If we now create a system

$$\begin{aligned} y_1 &= e^{\lambda_{\max} t} \\ y_2 &= e^{\lambda_{\min} t} \end{aligned} \tag{5.13}$$

processed by the MTSM which uses up to $p$ Taylor series terms we can write the $p^{th}$ terms

$$\begin{aligned} y_{1p} &= \frac{h^p}{p!} y_1^{(p)} = \frac{h^p}{p!} \left(e^{\lambda_{\max} t}\right)^{(p)} = \frac{h^p}{p!} \lambda_{\max}^p e^{\lambda_{\max} t} \\ y_{2p} &= \frac{h^p}{p!} y_2^{(p)} = \frac{h^p}{p!} \left(e^{\lambda_{\min} t}\right)^{(p)} = \frac{h^p}{p!} \lambda_{\min}^p e^{\lambda_{\min} t} \end{aligned} \tag{5.14}$$

Let us focus on the terms (5.14) without the known factor $h^p/p!$, transforming them to the form used in many common computers, i.e. in form

$$x = a \cdot 2^b, \qquad 1/2 \leq |a| < 1 \tag{5.15}$$

Supposing the examined system is stiff, the difference between the maximum and minimum eigenvalue is significant ($\lambda_{\max}$ is a large number and $\lambda_{\min}$ is a small number).

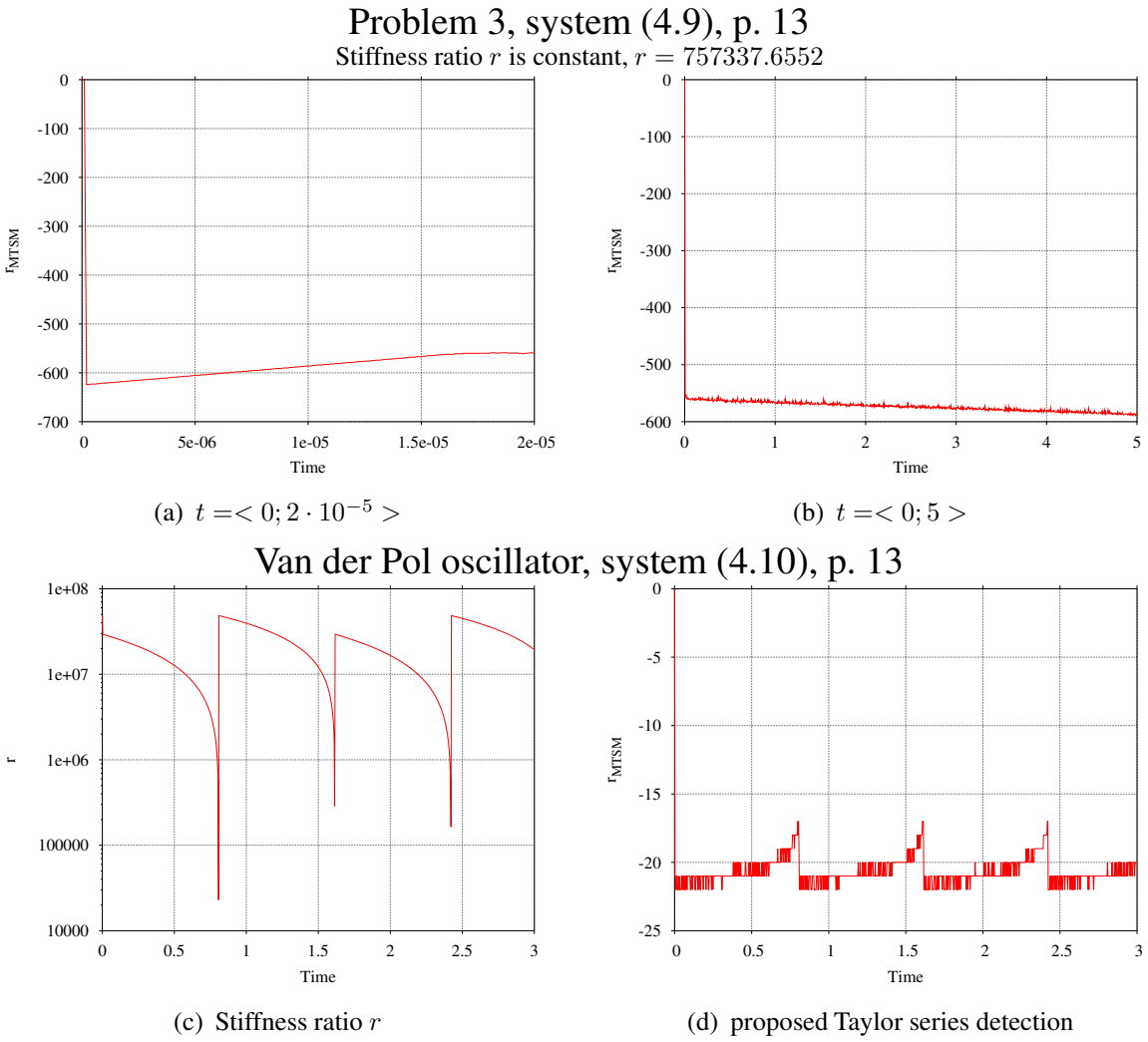
Let the $p^{th}$ Taylor series terms $y_{1p}$ and $y_{2p}$ be written in the form given by (5.15). The part $b$ in this notation is

$$\begin{aligned} \text{for } y_{1p} &: \lceil \log_2 \lambda_{\max}^p + \log_2 e^{\lambda_{\max} t} \rceil = \lceil p \log_2 \lambda_{\max} + \lambda_{\max} t \log_2 e \rceil \\ &\quad b_1 \approx t\lambda_{\max} \\ \text{for } y_{2p} &: \lceil \log_2 \lambda_{\min}^p + \log_2 e^{\lambda_{\min} t} \rceil = \lceil p \log_2 \lambda_{\min} + \lambda_{\min} t \log_2 e \rceil \\ &\quad b_2 \approx p + t\lambda_{\min} \end{aligned} \tag{5.16}$$

This shows us that only a comparison of the exponents of the $p^{th}$ Taylor series terms is needed to decide whether a system is stiff or not. The Taylor series terms of order $p$ used in the computation may be very different in size across the system. In other words the slow and stiff components are easily guessed by the rate the sizes of the Taylor series terms of the particular equation decrease.

If the difference is very significant the system may be declared (locally) stiff. Unlike the commonly used stiffness ratio, the analysis of the Taylor series terms provides more information – it also indicates whether the system in the examined time is changing rapidly (small difference of the Taylor series terms) or not. Using the information a proper action might be taken.

The detection method has been implemented in TKSL/C and put to a test on a set of various stiff systems with positive results (see Fig. 5).




(a) $t =< 0; 2 \cdot 10^{-5} >$

(b) $t =< 0; 5 >$

(c) Stiffness ratio $r$

(d) proposed Taylor series detection

Figure 5: Stiffness ratio $r$ versus proposed stiffness detection method.

# 6 Adaptation of Modern Taylor Series Method for Stiff Systems

The second major goal of the thesis is to suggest a modification of the MTSM such that would make it suitable for stiff systems.

## 6.1 High-order MTSM and Multiple Word Arithmetics

The more Taylor series terms are used, the larger is the region of absolute stability. Being aware of this fact one can decide to use more terms to ensure the stability of solution of a stiff system while keeping a sufficiently large integration step. How-

ever, the rounding error often ruins the computation in such a case.

Nevertheless, by employing a multiple word arithmetics in the MTSM the computation can be done with an extremely large integration step, because a greater number of Taylor series terms can be used without the loss of precision due to the rounding error.

Let us examine the Dahlquist's test problem

$$y' = \lambda y \tag{6.17}$$

The $k-$th Taylor series term for equation (6.17) is $y_{(k)} = \frac{h^k}{k!}\lambda^k e^{kt}$

We have to ensure that starting from a term $k > n$ the Taylor series term factor $\lambda^k$ grow slowlier than the problem-independent factor $k!/h^k$. This ensures that the Taylor series terms diminish starting with the $n-$th term.

The Table 4 shows the number of Taylor series terms needed for the computation should the last used term be less than $\varepsilon = 10^{-20}$. Various stepsizes $h$ and values of $\lambda$ are examined. Very similar results were achieved for $\varepsilon = 10^{-10} \ldots 10^{-50}$.

| | $\lvert\lambda\rvert$ | | | | | | |
|---|---|---|---|---|---|---|---|
| $h$ | 1 | 10 | 100 | $10^3$ | $10^4$ | $10^5$ | $10^6$ |
| $10^{-8}$ | 3 | 3 | 4 | 4 | 5 | 6 | 8 |
| $10^{-7}$ | 3 | 4 | 4 | 5 | 6 | 8 | 12 |
| $10^{-6}$ | 4 | 4 | 5 | 6 | 8 | 12 | 22 |
| $10^{-5}$ | 4 | 5 | 6 | 8 | 12 | 22 | 58 |
| $10^{-4}$ | 5 | 6 | 8 | 12 | 22 | 58 | 312 |
| $10^{-3}$ | 6 | 8 | 12 | 22 | 58 | 312 | 2760 |
| $10^{-2}$ | 8 | 12 | 22 | 58 | 312 | 2760 | 27223 |
| 0.1 | 12 | 22 | 58 | 312 | 2760 | 27223 | 271868 |
| 1 | 22 | 58 | 312 | 2760 | 27223 | 271868 | 2718320 |
| 10 | 58 | 312 | 2760 | 27223 | 271868 | 2718320 | 27182855 |

Table 4: Order of MTSM used to solve problem (6.17) depends on both $\lambda$ and stepsize $h$.

The obvious fact shown in the Table 4 is that for a certain number of Taylor series terms computed a pair $(h, \lambda)$ can be chosen. However, $\lambda$ is given by the problem and so a proper stepsize $h$ for the particular maximum number of Taylor series terms (or vice versa) has to be used. A thing to consider is the fact that the number of required Taylor series terms grows quickly as $h$ and/or $\lambda$ increases.

**Stepsize Versus Order of MTSM**

The number of operations needed to compute the Taylor series terms up to order $p$ is $\mathcal{O}(p^2)$. Considering the remainder term $R = \frac{h^{p+1}}{(p+1)!} f^{(p+1)}(t)$ of the Taylor expansion of order $p$ we often speak of accuracy of order $\mathcal{O}(h^{p+1})$. Let $\varepsilon$ be the truncation error when MTSM of order $p$ and integration step $h$ is used. The options to increase the accuracy to $\varepsilon^k,\ k \geq 2$ are:

- to reduce the integration step size (while maintaining the order of MTSM equal to $p$).

  The new stepsize $\bar{h} = h^k$. According to the formula of the remainder Taylor series term

  $$\varepsilon :\ \mathcal{O}(h^p) \qquad \varepsilon^k :\ \mathcal{O}(h^{pk}) \qquad \mathcal{O}(\bar{h}^p),\ \bar{h} = h^k$$

  In this case we have to use $\frac{1}{h^{k-1}}$ steps of size $\bar{h}$. The total number of operations is thus increased by a factor $\frac{1}{h^{k-1}}$.

- to increase the order of MTSM to $kp$ (the stepsize remains unchanged).

  There is a quite complicated result in [22] that the number of Taylor series terms $p$ needed to achieve accuracy $\varepsilon$ is $p = -\frac{1}{2}\log\varepsilon$. Using this conclusion, it is clear that the increase of the accuracy to $\varepsilon^k$ corresponds to increase of the MTSM order to $kp$. By increasing the order of MTSM to $kp$ the total number of operations is increased by a factor $k^2$.

In order to produce results with accuracy increased to $\varepsilon^k$ one has to either increase the the order of MTSM to $kp$ or reduce the stepsize to $h^k$. The former usually leads to much smaller increase of operations than the latter option.

The qualities of the used arithmetics, available memory, computer architecture have to be considered to obtain optimal results.

The test problems were solved by the suggested technique (i.e. using MTSM of high-order and multiple word arithmetics). The parameters for the method were all the same in all the cases in the sake of simplicity and easy comparison. The requested accuracy was $\varepsilon = 1{\cdot}10^{-10}$ in every case. A multiple word arithmetics with 1024 bits used for the mantissa was employed. This is obviously a too exaggerated value, but it ensured all the calculations were done correctly even when using a very high MTSM order $p$. The results are summarized in the Fig. 6 There are two pictures for each of the problems:

- The relation between the largest possible integration step $h$ and the used MTSM order $p$ is shown first. There are minimum and maximum values of $h$, the average integration step $h$ and the median of $h$ shown in the figure.

- The second picture of the series presented for each of the problems shows the used MTSM order $p$ and the corresponding integration step $h$ as time functions (along with the solution of the problem). One can see that both the order $p$ and the integration step $h$ vary over the time.

Problem 3,
system (4.9), p. 13

Van der Pol oscillator,
system (4.10), p. 13

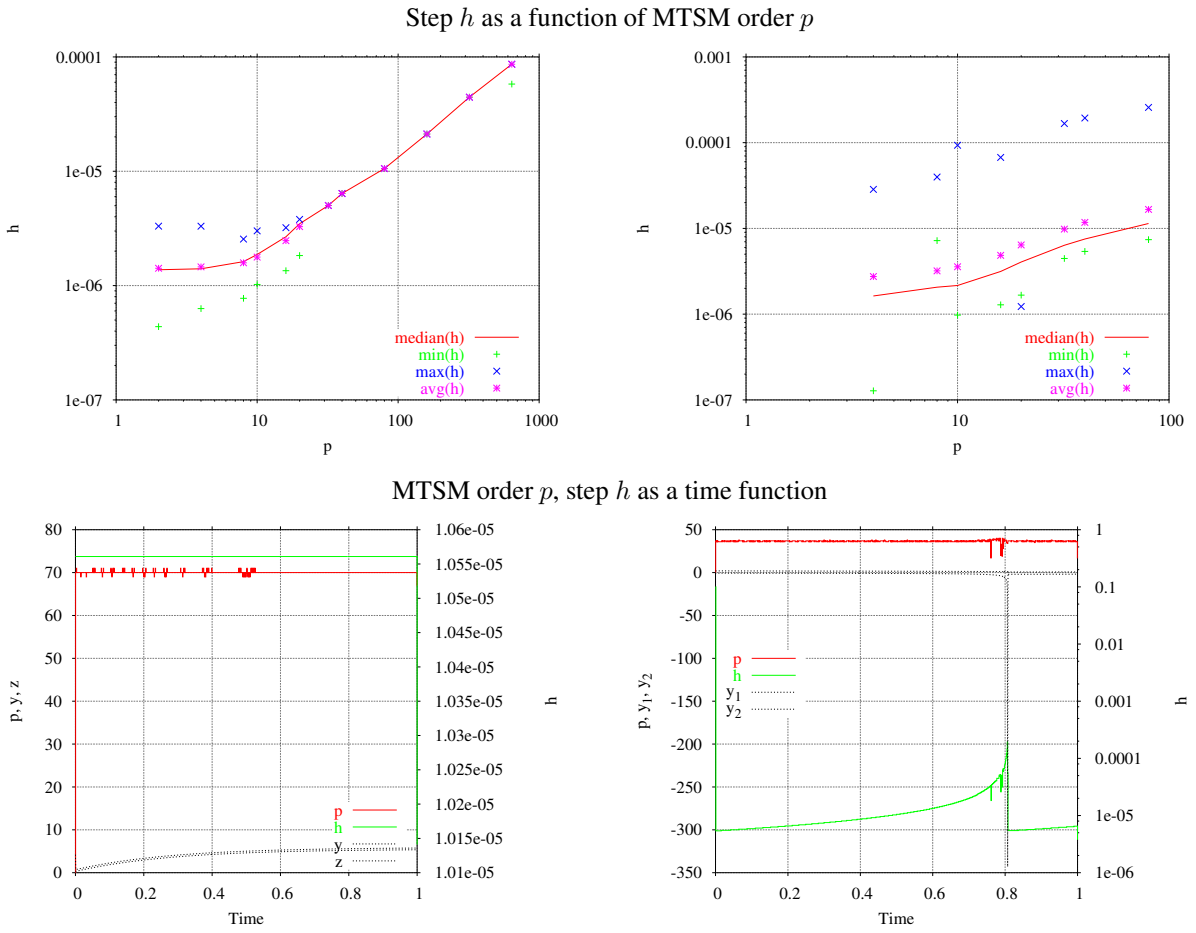



Figure 6: High-order MTSM and multiple word arithmetics, features of the solution.

According to [13]: *The predictor-corrector Adams code DEABM of Shampine & Watts solves van der Pol problem with* $\mu = 500$, $Rtol = 1 \cdot 10^{-2}$, $Atol = 5 \cdot 10^{-8}$ *on interval* $t \in < 0, 0.001 >$ *in 450 steps. By using the IRK RADAU5 method, the problem is solved in 11 steps.* The suggested method solves this problem with $\varepsilon = 1 \cdot 10^{-10}$ using MTSM order $p = 90$ in 119 steps.

The MTSM order $p$ in ranges 20–60 gives the best performance. A very low order makes the method unusable as the integration step must be very small. A very high order leads to a great overhead caused by evaluation of the Taylor series terms.

The MTSM order $p$ may be much larger in the case of some linear problems which results in an large integration step $h$.

For many problems the suggested method is quite competitive with the commonly used methods. The advantage over the common methods is that the computation needed by the implicit method is completely avoided, which makes the method usable even for large systems of ODE's.

## 6.2 Implicit MTSM

Unfortunately, the MTSM is A-stable only if all the Taylor series terms are used during the computation. The explicit methods are not very efficient (or even unusable) for stiff problems, so let us focus on the implicit methods.

Dahlquist's problem (6.17) is used for linear stability analysis. Its solution is $y = e^{\lambda t}$ and this is the reason to make use of the $(k, j)$-Padé approximation to $e^z$ given by $R_{kj}(z) = \frac{P_{kj}(z)}{Q_{kj}(z)}$ where

$$\begin{aligned} P_{kj}(z) &= 1 + \frac{k}{j+k}z + \frac{k(k-1)}{(j+k)(j+k-1)} \cdot \frac{z^2}{2!} + \cdots + \frac{k(k-1)\dots 1}{(j+k)\dots(j+1)} \cdot \frac{z^k}{k!} \\ Q_{kj}(z) &= P_{jk}(-z) \end{aligned}$$

with error

$$e^z - R_{kj}(z) = (-1)^j \frac{j!k!}{(j+k)!(j+k+1)!} z^{j+k+1} + \mathcal{O}(z^{j+k+2})$$

**Theorem 6.1**
*[14]: A method is A-stable if and only if its stability function is $(k, j)$-Padé approximation $R_{kj}(z)$ such that $k \leq j \leq k + 2$ holds.*

Coupling this knowledge with the nature of the MTSM we arrive to the following idea: The MTSM lacks the A-stability because its stability function $R(z)$ doesn't hold theorem 6.1. This suggests that we try to change the degree of the numerator or denominator of the Padé approximation $R_{kj}(z) = \frac{P_{kj}(z)}{Q_{kj}(z)}$ corresponding to the stability function so as the condition $k \leq j \leq k + 2$ holds.
The stability function of a one-step methods $R(z)$ is such that

$$y_{i+1} = R(z)y_i$$

The stability function of the MTSM of order $p$ is

$$R(z) = 1 + z + \frac{z^2}{2!} + \frac{z^3}{3!} + \cdots + \frac{z^p}{p!} \tag{6.18}$$

According to the theorem 6.1 a way of assuring the method is A-stable is that we increase the degree of the denominator of the function $R(z)$. The stability function for $k = j$ is

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

The corresponding A-stable one-step method which uses the Taylor series:

$$\begin{aligned} y_{i+1} = y_i + \tfrac{1}{2}h\left(f(y_i)+f(y_{i+1})\right) + \tfrac{(k-1)}{2(2k-1)}\tfrac{h^2}{2!}\left(f'(y_i)-f'(y_{i+1})\right) + \\ +\cdots+\tfrac{(k-1)...1}{2(2k-1)...(k+1)}\tfrac{h^k}{k!}\left(f^{(k-1)}(y_i)+(-1)^k f^{(k-1)}(y_{i+1})\right) \end{aligned} \tag{6.19}$$

Let us call it *Implicit Modern Taylor series method (iMTSM)*.

**Remark** Using iMTSM formula (6.19) with $k = 1$ we get the well-known trapezoidal rule $y_{i+1} = y_i + \frac{1}{2}h\left(f(y_i)+f(y_{i+1})\right)$. The iMTSM formula (6.19) with $k = 2$ we get a method with stability function equal to that of Lobatto IIIA (IRK method of order 4; its description can be found for example in [13]). Similarly, for $k = 3$ the stability function of the iMTSM method is equal to that of Kuntzmann & Butcher method (IRK method of order 6).

**L-stable iMTSM**

However, the iMTSM (6.19) is not an L-stable method, because $\lim_{z\to\infty} R_{kk}(z) = (-1)^k \neq 0$ To make the method L-stable, the denominator $Q_{kj}(z)$ of (6.19) must be of a higher degree than the numerator $P_{kj}(z)$.

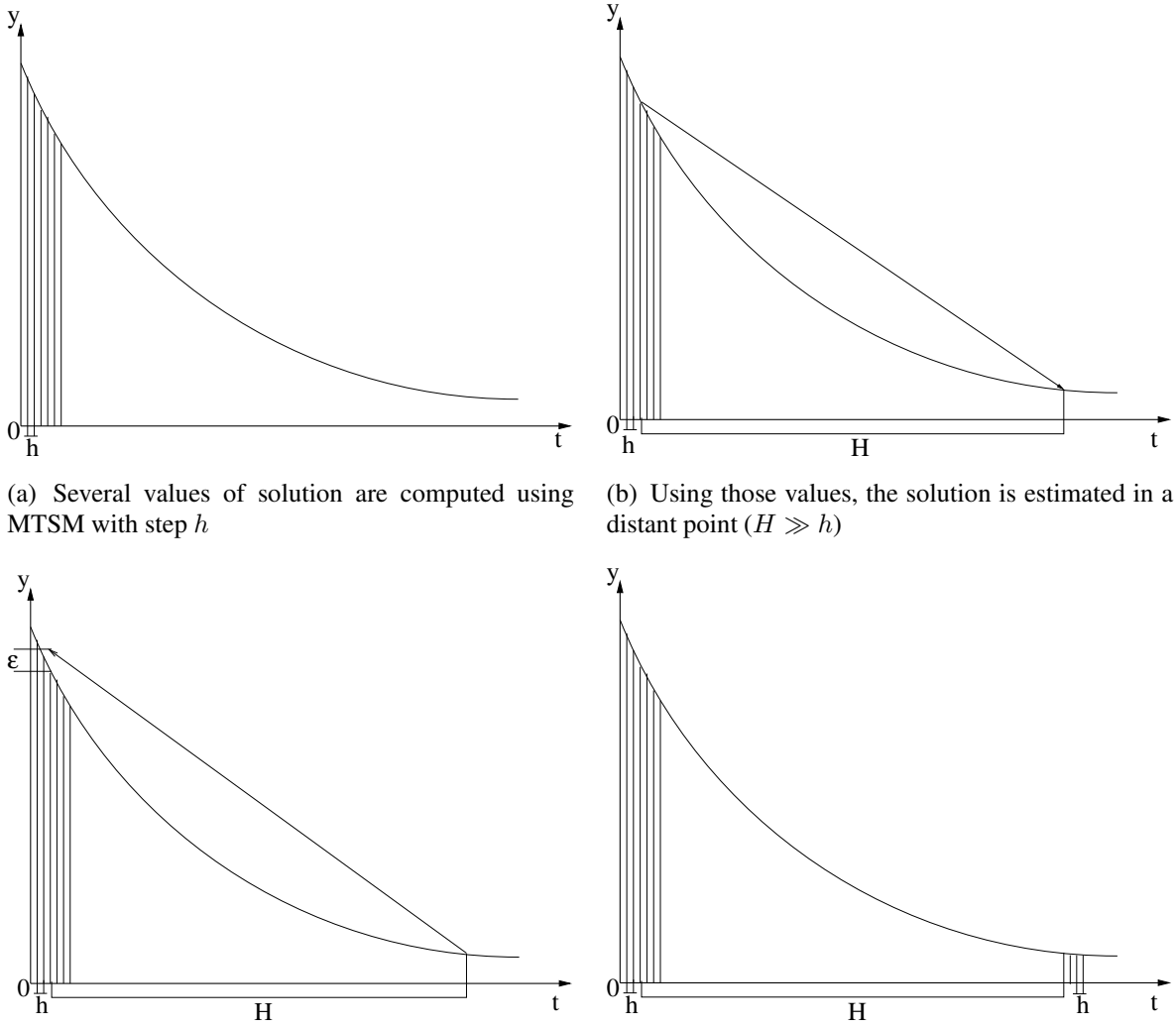
The L-stable version of the iMTSM, for which the condition $\lim_{z\to\infty} R_{k,k+1}(z) = 0$ holds, uses stability function $R_{k,j}(z)$ where $j = k+1$.

$$\begin{aligned} y_{i+1} = \;& y_i + \tfrac{1}{2k+1}h\left(kf(y_i)+(k+1)f(y_{i+1})\right) + \\ &+\tfrac{1}{2(2k+1)}\tfrac{h^2}{2!}\left((k-1)f'(y_i)-(k+1)f'(y_{i+1})\right)+ \\ &+\tfrac{k-1}{2(2k+1)(2k-1)}\tfrac{h^3}{3!}\left((k-2)f''(y_i)+(k+1)f''(y_{i+1})\right)+ \\ &+\cdots+ \\ &+\tfrac{(k-1)(k-2)...2}{2(2k+1)(2k-1)...(k+2)}\tfrac{h^k}{k!}\left(1f^{(k-1)}(y_i)+(-1)^k(k+1)f^{(k-1)}(y_{i+1})\right)+ \\ &+\tfrac{(k-1)(k-2)...1}{2(2k+1)(2k-1)...(k+1)}\tfrac{h^{k+1}}{(k+1)!}\left((-1)^{k+1}(k+1)f^{(k)}(y_{i+1})\right) \end{aligned} \tag{6.20}$$

**Remark** Using the L-stable variant of iMTSM (6.20) with $k = 0$ we get the implicit Euler method $y_{i+1} = y_i + hf(y_{i+1})$.

## 6.3 Outline of the Suggested iMTSM Algorithm

The algorithm described here is also shown in Fig. 7.




(a) Several values of solution are computed using MTSM with step $h$

(b) Using those values, the solution is estimated in a distant point ($H \gg h$)

(c) The usage of $H$ is veryfied using formula (6.19).

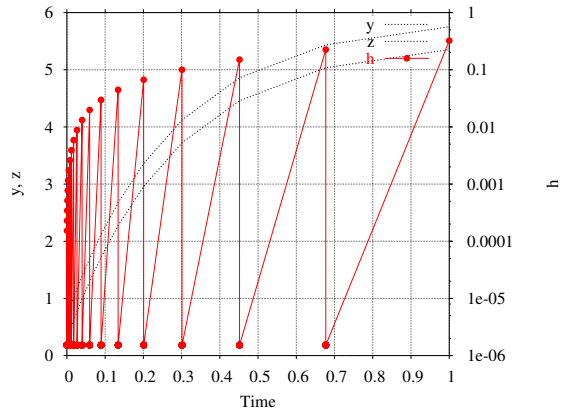
(d) Provided the accuracy meets the requirements, the computation continues using $h$.

Figure 7: Outline of the implicit MTSM algorithm.

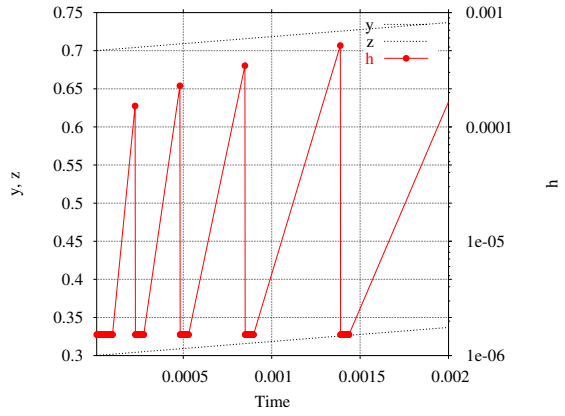
1. At the beginning, the MTSM is used in the traditional way. If the system is declared to be stiff the stiff solver starts its job.
2. $n$ values of the solution are computed in equidistant points (with stepsize $h$) using the classical MTSM. Using the $n$ values, derivatives of the solution up to the order $n-1$ are evaluated (see the full version of the thesis for details).
3. The computed derivatives in conjuction with the Taylor series are used in order to estimate the value of the solution $y_{i+1}$ with step $H \gg h$.

4. The derivatives in the distant point $f^{(k)}(y_{i+1})$ are computed using the MTSM again and the formula (6.19) is used in order to estimate the error.
5. The MTMS computation proceeds from the new point $y_{i+1}$ if a good result was obtained. The value of $H$ may be increased for further usage.
6. The computation continues with MTMS from the original point, if the test fails. The length of $H$ may be decreased.

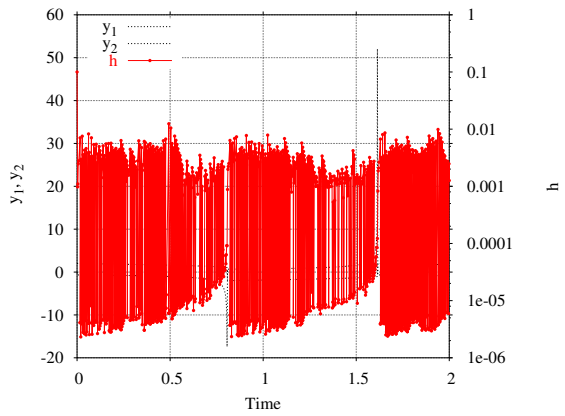
The successful application of the iMTSM is shown in Figure 8. The usage of steps $h$ and $H$ is especially clear from the Fig. 8(b).



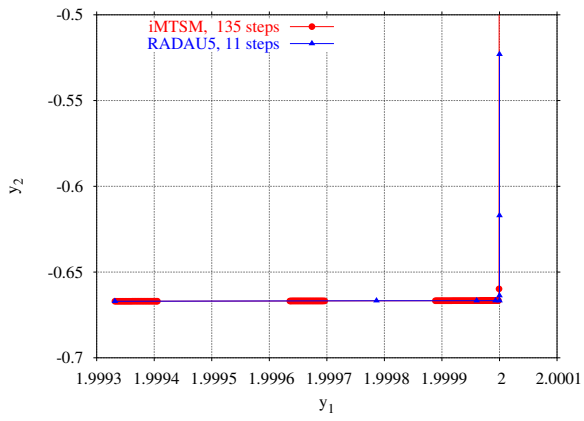

(a) Problem 3, system (4.9)– $t \in < 0, 1 >$




(b) Problem 3, system (4.9)– $t \in < 0, 0.002 >$




(c) Van der Pol oscillator, system (4.10) – $t \in < 0; 2 >$




(d) Van der Pol oscillator, system (4.10) – $t \in < 0; 0.001 >$

Figure 8: Stepsize control of the new method which combines usage of multiple word arithmetics and iMTSM (as computed by TKSL/C)

The multiple word arithmetics and iMTSM method complement one another well. If the iMTSM shows that the attempted large step $H$ fails, there is still a chance the usage of high order MTSM together with the multiple word arithmetics will be successful.

The new approach was compared to commonly used methods and old implementation of MTSM – TKSL/386 which lacks the ability to solve stiff problems. The results obtained using the new simulation system TKSL/C are superior to those provided by TKSL/386 which was to be expected.

The new method is quite comparable to those commonly used. Although the number of used integration steps made by TKSL/C is usually higher compared to the other tested methods, there is still quite an advantage — no need to solve nonlinear systems or to make an LU decomposition. This feature may be crucial when dealing with large systems. Refer to the full version of the thesis for detailed comparison.

# 7 Conclusions

- In order to achieve the goals of the thesis *a simulator TKSL/C* based on the MTSM has been created. It can handle very large scale systems, makes use of multiple arithmetics and is highly configurable.
- *A completely new approach in stiffness detection* is proposed in this thesis. It is based on the analysis of the Taylor series terms. The Taylor series are computed in order to implement the MTSM and thus almost no extra work is needed. The new stiffness detection method has been implemented and incorporated into TKSL/C and put to the test on various examples of stiff systems — with success.
- *A new method of solving stiff systems of ODE's* based on the Modern Taylor series method is suggested. The advantages of usage of high-order Taylor series, multiple word arithmetics and implicit variant of MTSM were scrutinized. The suggested algorithm was implemented into a new simulator TKSL/C and then put to a test on the problems. The new approach was compared to commonly used methods and old implementation of MTSM – TKSL/386 which lacks the ability to solve stiff problems. The results obtained using the new simulation system TKSL/C are superior to those provided by TKSL/386, which was to be expected.

  The new method is competitive with those commonly used. Although the number of used integration steps made by TKSL/C is usually higher compared to the other tested methods, there is still quite an advantage — no need to solve nonlinear systems or to make an LU decomposition. This feature may be crucial when dealing with large systems.

The fact that the new proposed methods of stiffness detection and stiff system solving are based on the MTSM implies that they are very suitable for parallel and/or

hardware implementation. The parallelism occurs at several levels and the method uses only the basic mathematical operations.

Several parts of this thesis have been presented on internation and national conferences home and abroad.

### Further Research

There is still a room for further research – one could analyze the algorithms in order to find its optimally parallelizable parts or suggest a suitable hardware implementation. Another possibility is to find rigorous proofs of the algorithms.

## 8 Přehled

Předkládaná dizertační práce se zabývá numerickým řešením stiff soustav obyčejných diferenciálních rovnic, které se často vyskytují při vytváření abstraktního matematického modelu spojitých dynamických systémů (popisy dějů v elektrických obvodech, chemických reakcí, fyzikálních jevů atp.).

Jakkoliv jsou stiff soustavy při modelování obvyklé, zůstávají na pokraji zájmu. Ucelená literatura, která by se věnovala této problematice, v podstatě neexistuje a je nutno sbírat kusé informace roztroušené v různých sbornících a časopisech.

Zvládnutí stiff systémů je v podstatě dvoufázové. Vzhledem k tomu, že numerické metody řešení běžných (ne-stiff) soustav obyčejných diferenciálních rovnic jsou pro stiff systémy nepoužitelné, je nejprve nutno zjistit, zda je soustava předkládaná k řešení stiff systémem. Teprve v případě, kdy jedná o stiff systém, můžeme nasadit metodu specializovanou na tento typ soustav.

Úvodní kapitoly obsahují stručný přehled současného stavu detekce a numerického řešení stiff systémů. Běžně používané metody mají společnou nepříjemnou vlastnost – jsou výpočetně náročné a v případě rozsáhlých soustav diferenciálních rovnic v podstatě nepoužitelné. V případě detekce stiff systému se jedná o výpočty vlastních čísel, při řešení stiff systému pak o řešení pomocných soustav nelineárních rovnic, LU rozklad atp.

V nedávné době byla představena [15] originální numerická metoda pro řešení soustav obyčejných diferenciálních rovnic – Moderní metoda Taylorova rozvoje (MTSM). Metoda při výpočtu používá proměnný počet členů Taylorova rozvoje, což má pozitivní dopad na rychlost a přesnost výpočtu. MTSM používá pouze základní aritmetické operace ($+,-,\times,\div$) a je snadno paralelizovatelná na různých úrovních.

V uvedených souvislostech jsou formulovány cíle práce – nalezení vhodného způsobu detekce stiff soustav a jejich numerického řešení. Nové metody by měly být konzistentní se stávající MTSM a také by měly být aplikovatelné na rozsáhlé soustavy diferenciálních rovnic.

Má-li být nově navržená metoda zhodnocena, je potřeba provést testy pro srovnání se stávajícími metodami. Byla představena řada příkladů; uvedeny jsou informace o výchozí situaci pro formulaci problému, jeho řešení a některé charakteristiky. Tato sada příkladů je posléze použita při testování navržených metod.

**Dosažené výsledky**

- K dosažení vytyčených cílů byl vytvořen simulátor TKSL/C založený na MTSM. TKSL/C dokáže řešit rozsáhlé systémy obyčejných diferenciálních rovnic prvního řádu, využívá víceslovní aritmetiky a disponuje řadou různých nastavení ovlivňujících výpočet a výpis výsledků.

- V předložené dizertační práci je navržena zcela nová metoda detekce stiff systémů, která se odvíjí od MTSM. Metoda analyzuje jednotlivé členy rozvoje použité během výpočtu a na základě této analýzy vyhodnotí, zda je řešen stiff systém. Vzhledem k tomu, že jsou využity číselné informace, které jsou součástí vlastního výpočtu MTSM, je metoda výpočetně úsporná a vhodná pro nasazení i v případě rozsáhlých soustav.

- Dále je navržena originální nová metoda řešení stiff soustav obyčejných diferenciálních rovnic, která je rovněž (v souladu se zadáním) založena na MTSM. Byly prozkoumány výhody použití vysokého počtu členů Taylova rozvoje a víceslovní aritmetiky; zkoumány byly rovněž možnosti navržené implicitní varianty MTSM. Byl navržen a implementován algoritmus kombinující výhody obou přístupů. Výsledná metoda byla úspěšně použita při řešení testovacích příkladů.

  Nově navržená metoda řešení stiff systémů byla porovnána s původní MTSM a také s několika běžně používanými metodami. Výsledně bylo zjištěno, že nová metoda je s nimi zcela srovnatelná. Výhodou je navíc její velice snadná paralelizovatelnost a zejména pak též použitelnost v případě rozsáhlých systémů (narozdíl od stávajících metod není třeba řešit pomocné nelineární soustavy apod.).

- Nově navržené metody byly implementovány – byl vytvořen simulační systém TKSL/C. Mnohé nové poznatky uvedené v této dizertační práci byly prezentovány na celé řadě zahraničních i domácích konferencí. Simulační systém TKSL/C je využíván také při výuce v několika předmětech na FIT VUT.

# References


[1] BURRAGE, K., BUTCHER, J. C.: Non-linear stability of a general class of differential equation methods. *BIT*, (20):185–203, 1980.

[2] BUTCHER, J. C.: Optimal order and stepsize sequences. *IMA Journal of Numerical Analysis*, (6):433–438, 1986.

[3] BUTCHER, J. C.: Diagonally-implicit multi-stage integration methods. *Applied Numerical Mathematics*, (11):347–363, 1993.

[4] BUTCHER, J. C.: An introduction to DIMSIMs. *Journal of Computational and Applied Mathematics*, 59–72, 1995.

[5] BUTCHER, J. C.: Order and stability of parallel methods for stiff problems. *Advances in Computational Mathematics*, 77–96, 1997.

[6] BUTCHER, J. C.: General linear methods for stiff differential equations. *Numerical Mathematics*, (41):240–264, 2001. ISSN 0006-3835.

[7] BUTCHER, J. C., CASH, J. R.: Some recent developments on numerical initial value problems: A survey. *Applied Numerical Mathematics*, (5):3–18, 1989.

[8] BUTCHER, J. C., CASH, J. R.: Towards efficient Runge–Kutta methods for stiff systems. *Society for Industrial and Applied Mathematics*, 1990.

[9] BUTCHER, J. C., JOHNSTON, P. B.: Estimating local truncation errors for Runge–Kutta methods. *Journal of Computational and Applied Mathematics*, 1992.

[10] BUTCHER, J. C., SINGH, A. D.: The choice of parameters in parallel general linear methods for stiff problems. *Applied Numerical Mathematics*, (34):59–84, 2000.

[11] DVOŘÁK, V., KUNOVSKÝ, J., SCHWARZ, J., ZBOŘIL, F.: Problem solving with parallel processing. *Proceedings of the Workshop'95*, 215–216, 1995.

[12] EKELAND, K., OWREN, B., ØINES, E.: Stiffness detection and estimation of dominant spectrum with explicit Runge-Kutta methods. *ACM Transactions on Mathematical Software (TOMS)*, 24:368–382, 1997.

[13] HAIRER, E., NØRSETT, S. P., WANNER, G.: *Solving Ordinary Differential Equations I*, vol. Nonstiff Problems. Springer-Verlag Berlin Heidelberg, 1987. ISBN 3-54017145-2.

[14] HAIRER, E., WANNER, G.: *Solving Ordinary Differential Equations II*, second revised ed., vol. Stiff and Differential-Algebraic Problems. Springer-Verlag Berlin Heidelberg, 1996. ISBN 3-540-60452-9.

[15] KUNOVSKÝ, J.: *Modern Taylor Series Method*. FEI, VUT, Brno, 1994. Habilitační práce.

[16] KUNOVSKÝ, J., MATÚŠ, J.: Algebraic equations roots and multiple arithmetic. *Proceedings of Conference MOSIS '96*, 27–32, 1996. ISBN 80-85988-02-X.

[17] KUNOVSKÝ, J., MELKES, F., MIKULÁŠEK, K.: Extremely large systems of algebraic equations. *Proceedings of ASIS 2001*, 117–120, 2001. ISBN 80-85988-61-5.

[18] KUNOVSKÝ, J., MIKULÁŠEK, K., ZBOŘIL, F.: Modern Taylor series method and partial differential equations. *Proceedings of the CISS 94, Zurich*, 134–136, 1994.

[19] KUNOVSKÝ, J., ŘEZÁČ, D.: Parallel computations based on Taylor series method. *Proceedings 4th MATHMOD Vienna*, 316–317, 2003. ISBN 3-901608-24-9.

[20] MAZZIA, F., IAVERNARO, F.: *Test Set for Initial Value Problems Solvers*. Department of Mathematics, University of Bari, 2003. `http://pitagora.dm.uniba.it/~testset/`.

[21] SHAMPINE, L. F.: Error estimation and control for ODEs. *Mathematics Department, Southern Methodist University*, 2004.

[22] SIMÓ, C.: *Global dynamics and fast indicators, In Broer, et al.: Global analysis of dynamical systems.* Bristol IOP Publishing, 2001. Chapter 15. ISBN 0750308036.

[23] VITÁSEK, E.: *Základy teorie numerických metod pro řešení diferenciálních rovnic*. Academia, Praha, 1994. ISBN 80-200-0281-2.

[24] WRIGHT, W.: *General linear methods with inherent Runge-Kutta stability*. PhD thesis, The University of Auckland, 2002.

# Author's C.V.

**Personal data**

| | |
|---|---|
| First and last name: | David Řezáč |
| Birth: | March 18th, 1978, Přerov, Czech Republic |
| Nationality: | Czech |
| Marital status: | married |
| E-mail: | rezacd@fit.vutbr.cz |
| Web: | http://www.fit.vutbr.cz/ rezacd |

**Education**

2001-2004: Post-graduate study, Faculty of Information Technology, Brno University of Technology

1996-2001: M.Sc. in computer science, Dept. of Computer Science and Engineering, Faculty of Electrical Engineering and Computer Science Brno University of Technology

**Memberships**

- 2002-2004 – Member of organizing committee of Student conference and competition EEICT
- High Performance Computing Research Group
- The Moravian Genealogical and Heraldic Society
- Mensa Czech Republic